A Project entitled

Motivating Students to Learn Concepts on Relational Databases via Gamification:

A Learning Kit for HKDSE ICT Curriculum

Submitted by

Chung Sai Yuk

submitted to the Education University of Hong Kong

for the degree *Bachelor of Education (Honours) (Secondary) - Information and Communication Technology*

in *April 2023*

Declaration

I, *Chung Sai Yuk* declare that this capstone project report represents my own work under the supervision of Associate Professor Dr Cheng Kwok Shing Gary, and that it has not been submitted previously for examination to any tertiary institution.

Signed _____

Chung Sai Yuk

11 April 2023

Acknowledgement

First and foremost, I would like to thank Almighty God for His grace and blessings, enabling me to successfully accomplish this capstone project.

I would like to express my deepest gratitude to my supervisor, Dr Cheng Kwok Shing Gary, for his continuous guidance and constructive advice in completing this project. My completion of this project would not have been accomplished without his assistance. I am particularly grateful for the valuable feedback and help given by Mr Hui, my ICT teacher in the Alma Mater, and Mr Godfrey Lai, the in-service ICT teacher. This endeavour would not have been possible without their insights and support. Lastly, I would like to acknowledge my friends who have given me encouragement.

Table of Content

Introduction

In the Information and Communication Technology (ICT) curriculum of the Hong Kong Diploma of Secondary Education (HKDSE), concepts of relational databases and Structured Query Language (SQL) are included in both the compulsory part and the elective options (CDC & HKEAA, 2015). Learning concepts on relational databases could be challenging for students. In view of the limited HKDSE ICT learning resources and the importance of the topic, the Project provides a quality course package for teachers to motivate students to learn SQL with the gamification approach.

Background and Need Analysis

With the development of big data in the last decade, the importance of information systems rising (Mills et al., 2015). SQL and the relational database model are the dominant and the foundation of information systems deployed. However, the relational database model could challenge students of its complex and abstract nature (Hamzah et al., 2019; Connolly & Begg, 2006). It could lead to students believing they are not capable of handling challenging topics. Considering the motivation is the self-power to push an adolescent to learn, students having low self-efficacy toward relational databases could be demotivated (Bandura, 1977; Schunk & Zimmerman, 2012).

The available teaching resources designed for HKDSE ICT are very limited or outdated. The current ICT textbook has not been updated for more than ten years (Tong, 2020). It would require teachers to prepare their teaching resources. Due to the limitation of class time, students are expected to be presented with teaching content and homework. It would be difficult for students to analyse complex problems but to complete simple exercises (Connolly & Begg, 2006). The drill-and-practice mode may overwhelm students when they face challenges.

Therefore, the Project could provide a ready-to-use course package for students to learn the relational database concepts with motivation.

Literature Review

Difficulty of Learning Relational Databases

In previous works, database analysis and design for solving problems are identified to be challenging as its complexity (DeGrace & Stahl, 1990). As suggested in Mitrovic's (1998) study, memorising the database, table and field names could block students from constructing SQL queries. Prior & Lister (2004) discussed that the lack of visuals of the query execution could generate difficulty. Recent research has shown that logical operations like subqueries and confusion between non-exist record and the NOT operator hindered students' learning (Taipalus et al., 2018; Migler & Dekhtyar, 2020). The learning difficulties found in different studies share the same pattern of errors. Summarily students are challenged with invisible logical operations. When designing logical operations like joining tables, students are required to visualise the process mentally. Otherwise, students would be unable to interpret the operation logic or rules behind the database. Meanwhile, students would be required to recall the database's schema when constructing the query. As suggested in the cognitive load theory, visualising with heavy memory retrieval could generate a massive burden on students' cognitive load (Shin, 2020; Sweller, 1984; 2011). It may be concluded that learning SQL and relational databases requires a substantial cognitive load, which may challenge secondary school students.

Motivation

Petri (1996) defines motivation as the power sources which push an individual to perform a specific action. Motivation is the deciding factor of students' engagement in learning activities (Brophy, 1987). Vroom (1964) identifies two elements of motivation: an individual's self-perception of the possibility of success and the incentive value of success. That means when the students believe that it is likely for them to succeed in learning something with high value, they will be motivated to learn (Wigfield & Cambria, 2010). Students could be possibly challenged by the deliberated complex and cognitive-demanding nature of learning SQL. Implying that once students found it challenging to learn or succeed, their self-efficacy would be degraded with the decreasing incentive value. With the lowered motivation, students would be less motivated to learn the concepts of relational databases.

Gamification

Deterding et al. (2011) define gamification as 'the use of game design elements in non-game contexts'. Gamification means using elements like points and leaderboards, which intended to promote specific behaviour (Mekler et al., 2015). Applying in instructional design, the method is sought as an effective approach to stimulate students' motivation (Plass et al., 2015). Alsawaier (2017) summarised research suggesting that applying gamification in class could positively affect students' motivation and engagement in learning activities. As gamification using the game elements brought joy to the students, the perceived value of participating in the learning activities could be increased with the elements. Students would be expected to be extrinsically motivated. By increasing extrinsic motivation with the use of gamification, students' motivation in engaging learning activities will be higher (Lister, 2015).

Teaching Relational Databases with Gamification

There are attempts to integrate gamification with relational database teaching. Lai (2020) finds that using gamification, even during the pandemic, could engage students and provide better academic results. Tuparov & Keremedchiev's (2020) research shares similar findings on increasing academic results. Dæhli et al. (2021) study revealed that students found the motivation to complete the tasks as they perceived them as challenges, which gave them a sense of accomplishment. The case is coherent with the suggestion that manageable tasks with moderate difficulty tasks could motivate students in the Dettmers et al. (2010) work. Referring to the prior works, this project may create a course package tailored for the HKDSE ICT curriculum context with gamification.

Project Output

Deliverables

The package is designed for learning SQL within the HKDSE ICT curriculum in 10 1-hour lessons with the following deliverables:

1. lesson plans;
2. presentation slides, worksheets, and game materials; and
3. online game learning platform ('SQLGo').

Objectives

The package is aimed to:

1. allow students to master the fundamental concepts of relationships and skills of data manipulation with SQL;
2. motivate students to learn relational database concepts through gamification-based design; and
3. maximise the students' learning effectiveness on relational databases by reducing the cognitive load with multimedia and game elements.

Project Design

Instructional Design

Scaffolding with Information Processing Theory

The lessons are designed with a scaffolding approach: the difficulty will be elevated while the guidance will be reduced progressively (see Figure 1.1). Each lesson begins by recalling students' memory of the last lesson. Students will be asked a question related to the learning objectives and discuss the need to solve similar issues. The teacher will then present the solution and demonstrate the example, and students will be guided to construct SQL statements. After instruction, there will be group activities or classroom games. Students will apply their learning, and their short-term memory will be strengthened. By completing the online tasks after class, students may consolidate their long-term memory (see Figure 1.2). With the scaffolding approach and information processing theory applied, students' learning effectiveness could be maximised (Lachman et al., 2015).

Figure 1.1

Diagram on the Package Experience Design

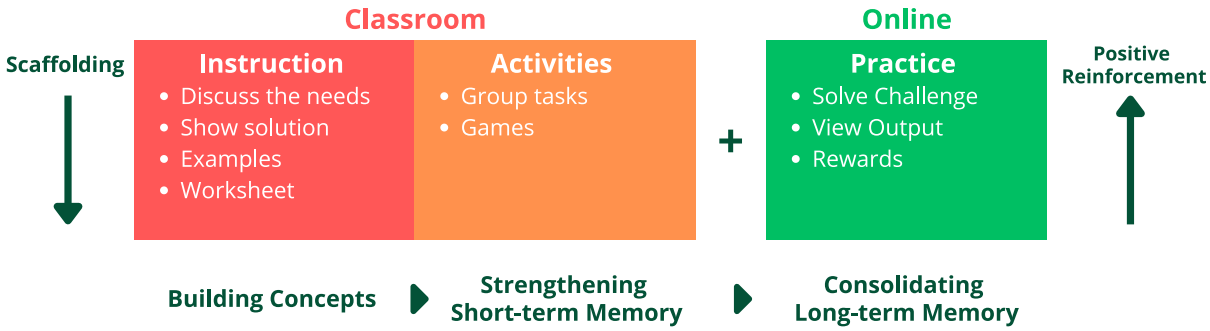
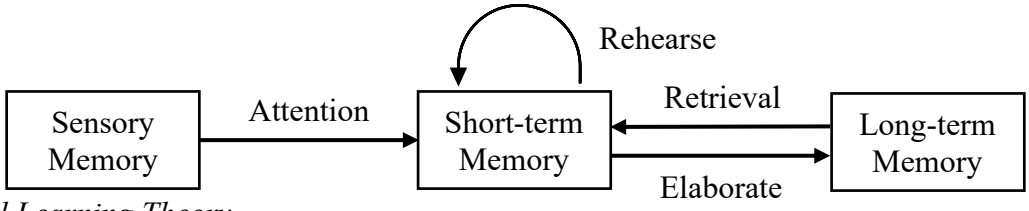


Figure 1.2

Diagram on Information Processing Model



Social Learning Theory

Social Learning Theory identifies students' natural learning from their peers through observation and modelling (Bandura & Walters, 1963). In addition to constructing a motivating environment by promoting social learning, it assists in building a belief that they are making progress altogether.

Positive Reinforcement with Gamification

Positive Reinforcement is applied in *SQLGo*. The platform assigns tasks according to the progress in class (see Figures 2.1 and 2.2). It enables students a sense of accomplishment when completing challenges with certain difficulties perceived. With iterations of this process, students may develop their competence and confidence along with a gamified rewarding experience.

Figure 2.1

SQLGo Progress Management for Teachers

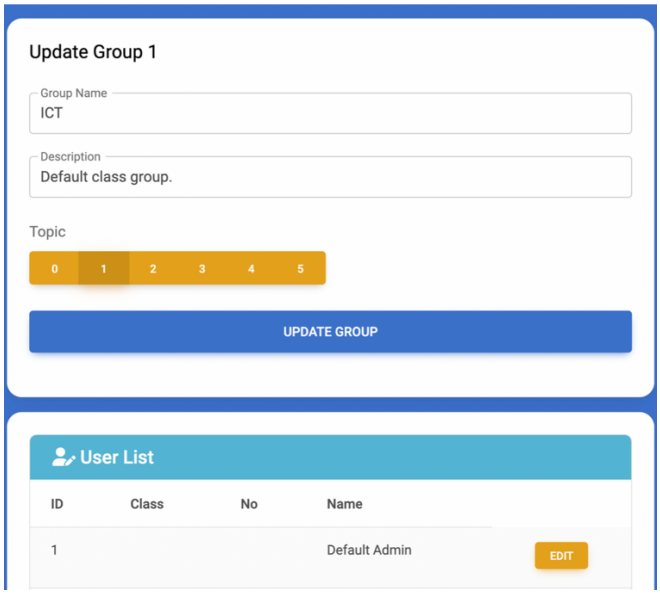
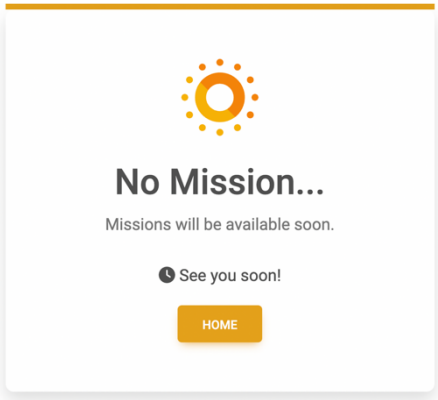


Figure 2.2

View on Completing Available Missions



SQLGo – Online Game Learning Platform

Story and Challenge Mode: Assessment as Learning

SQLGo is a game system designed for students to practice and learn SQL with motivation. SQLGo consists of two game modes: story and challenge mode. In story mode, students will follow a storyline to help solve problems (see Figure 3.1). In challenge mode, the system provides three random questions per day for students to exert themselves with a time limit (see Figures 3.2 and 3.3). When students complete the missions, they learn from their attempts which fosters assessment as learning.

Figure 3.1

Storyline in the Story Mode

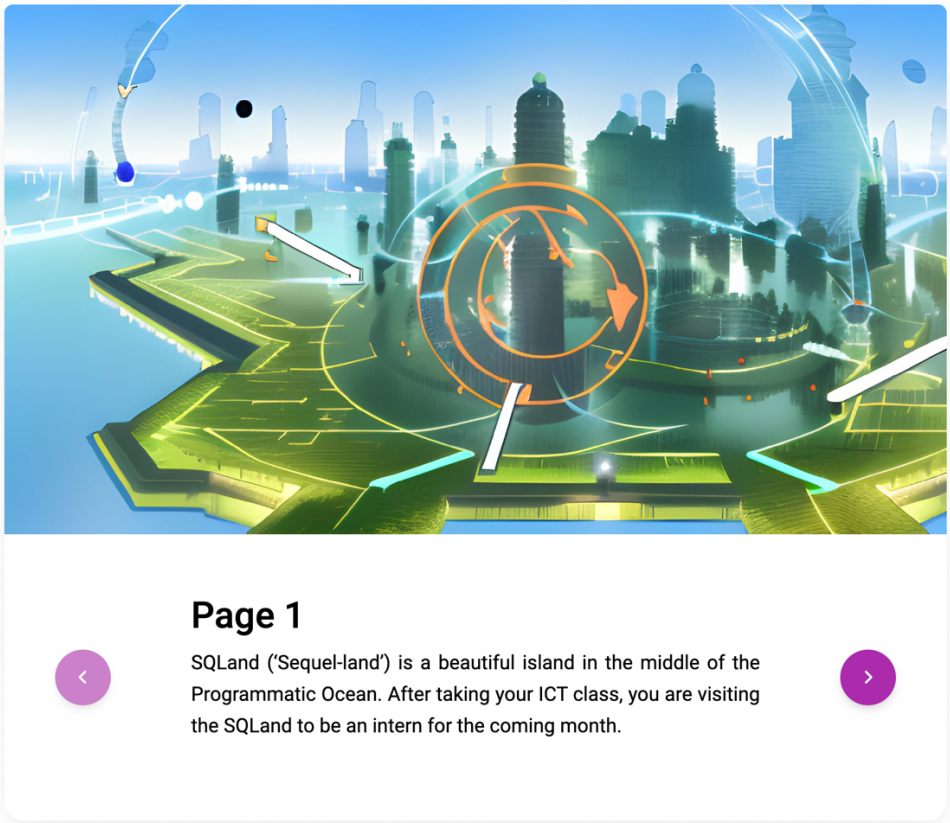


Figure 3.2

Mission in the Challenge Mode

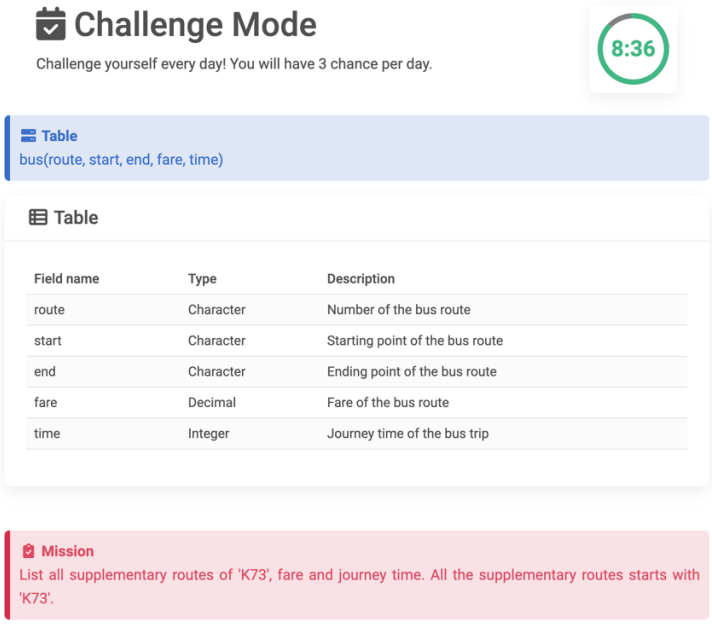
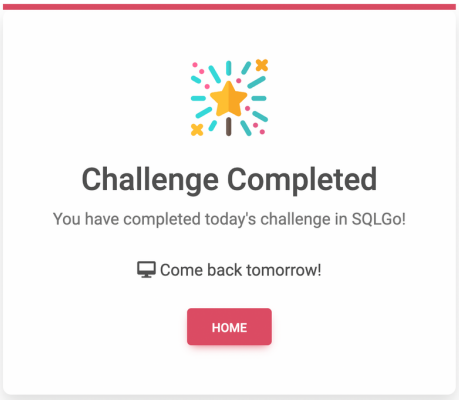


Figure 3.3

Message of Used up All Chances



Points and Leader Board: Incentives

Students will be awarded points and chips for completing the mission or challenge as incentives to encourage them to keep up their excellent work (see Figure 4.1). The leaderboard provides recognition to the top 3 leading students only to maintain an encouraging environment (see Figure 4.2).

Figure 4.1

Progress Shown on the Home Page

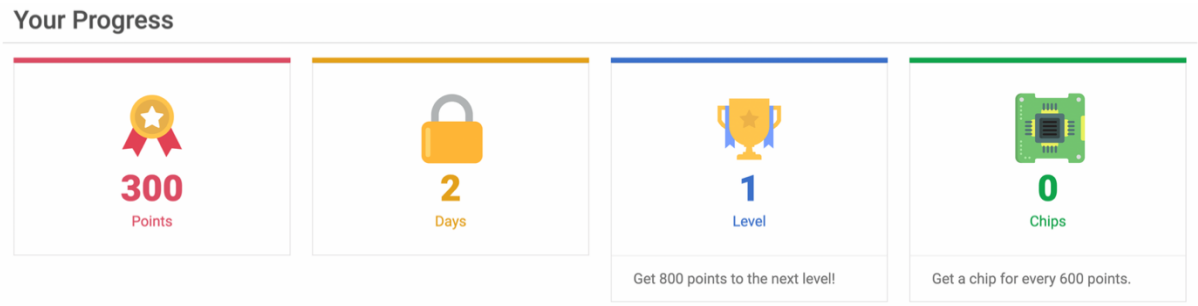
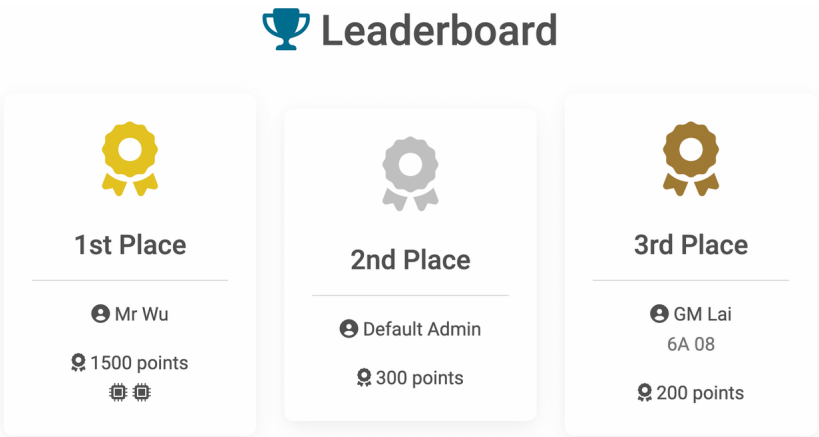


Figure 4.2

Leaderboard



SQL Practice: Learning Bit by Bit

In the platform, students will be presented with a mission or challenge which requires SQL statements (see Figure 5.1). The platform verifies the answer by executing the query and comparing the results, which allows all equivalent statements to complete the mission. Students will see the output from the database if the answer is correct (see Figure 5.2).

Figure 5.1

Example of the Mission in Story Mode

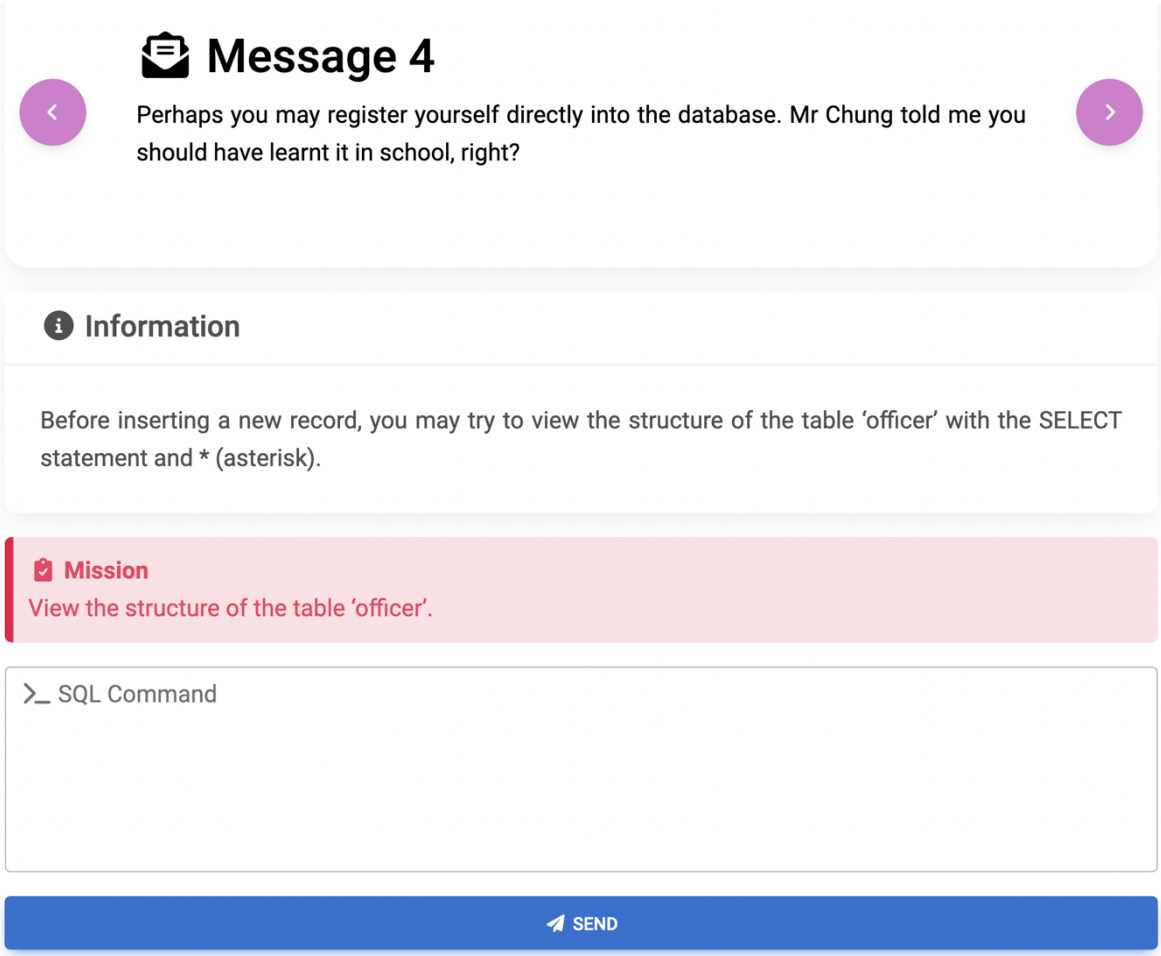
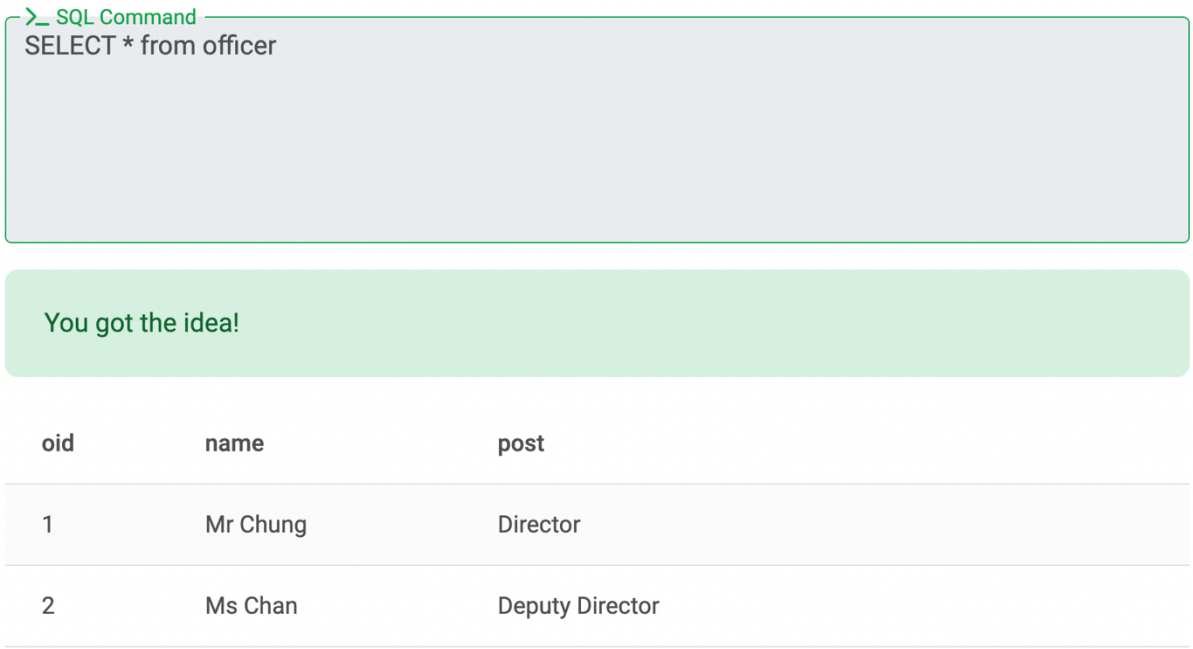


Figure 5.2

Example of Success Attempt



oid	name	post
1	Mr Chung	Director
2	Ms Chan	Deputy Director

If the statement is incorrect, students will see the error message returned from the database to help them understand the reason for the mistakes (see Figure 5.3 and 5.4). At the same time, a link to relevant topic resources will be shown to assist students in learning (see Figures 5.5 and 5.6).

Figure 5.3

Example of Syntax Error

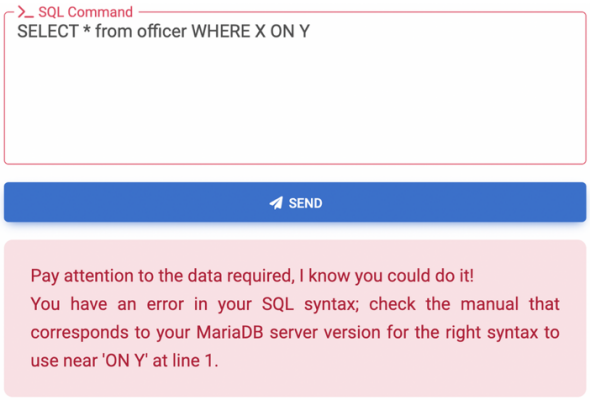


Figure 5.4

Example of Unknown Field

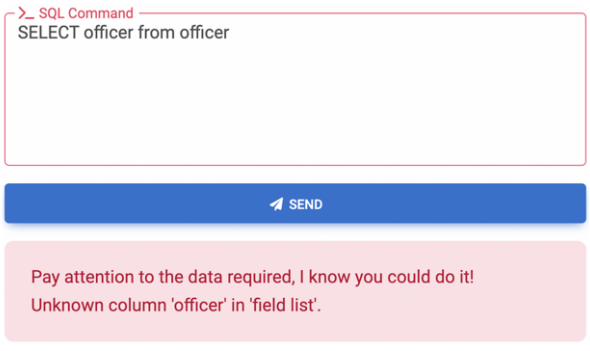


Figure 5.5

Assistance on Incorrect Answer

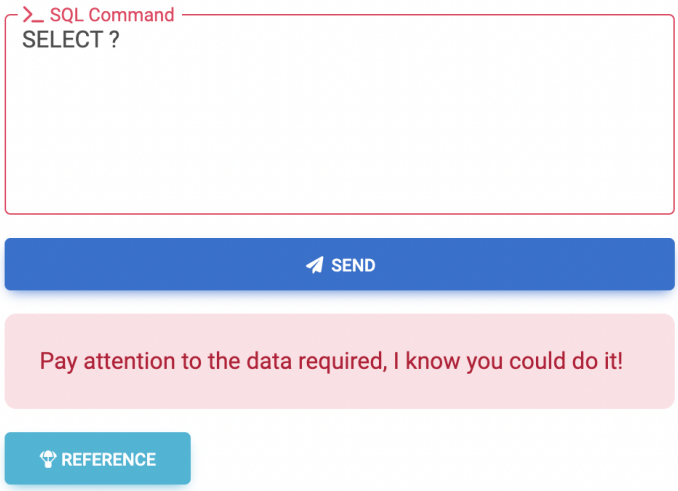
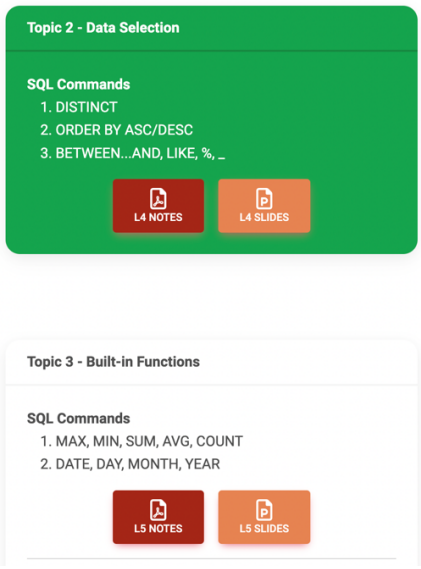


Figure 5.6

Material Page



The missions in the Story are designed to scaffold students' learning. Several tasks require students to construct statements based on the last query (see Figure 5.7). Students will be levelled up when completing enough points (see Figure 5.8).

Figure 5.7

Showing the Last Query

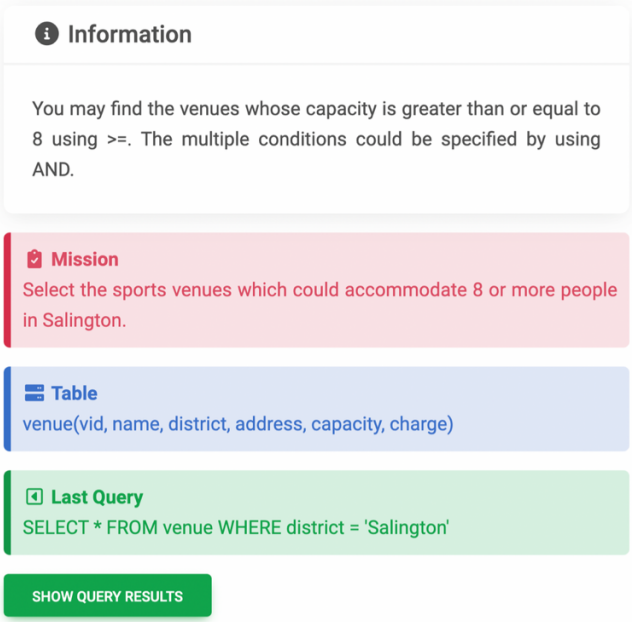
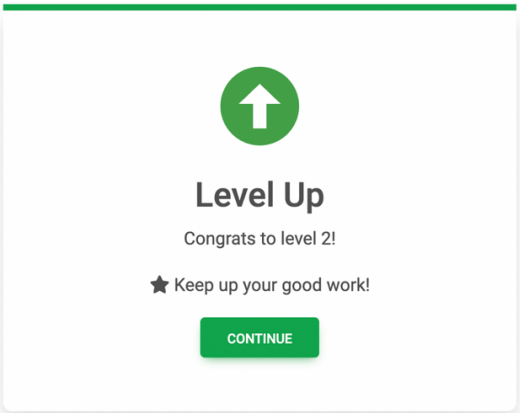


Figure 5.8

Message of Level Up



Point Adder: Connected Experience

The elements designed in the package formulate a connected experience for students. When students engage in class or exhibit positive behaviour, the teacher may reward them by boosting the points on SQLGo (see Figure 5.9). Students may see in their Point History record and increase students' perceived value on classroom engagement (see Figure 5.10).

Figure 5.9

Boosting Points to a Student

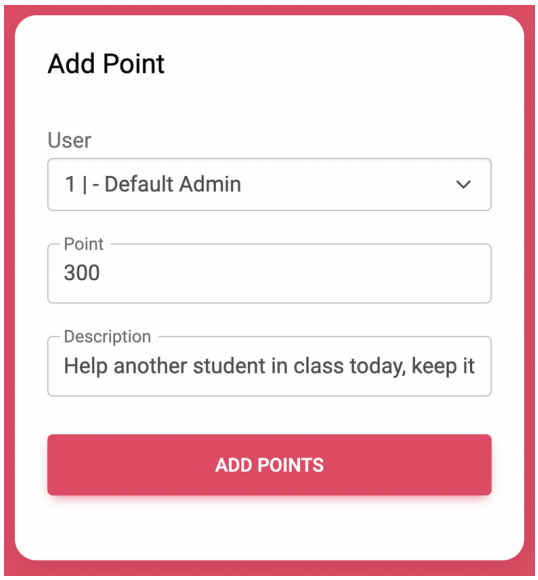
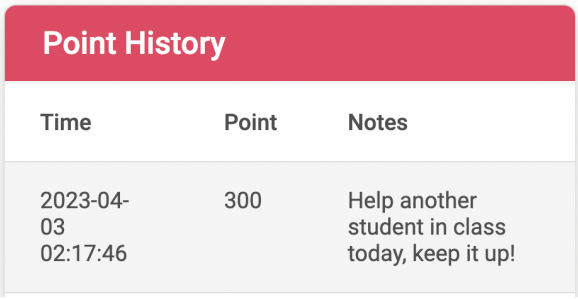


Figure 5.10

Record Shown on Point History



Evaluation and Further Development

The instructional elements in the package create an encouraging and engaging environment blended with various pedagogical approaches. Building the SQLGo system provides context-based online judge by verifying authentic query results, which pioneers a breakthrough in SQL resources for the HKDSE ICT curriculum. Meanwhile, the instruction in the story relies on text, which may create a barrier for students who are weak in English.

To further unleash the full potential of the SQLGo, error analysis could be the next milestone to reach. The data collected, like students' answers and time required, could provide solid evidence to the teacher, which is a powerful tool for fostering assessment for learning.

Conclusion

With the dominant role of relational databases in information systems, the demand of graduates to master relevant skills and knowledge is rising. Considering the limited available learning resources, Project built a quality course package of the challenging HKDSE ICT topic – relational databases, aimed to assist students in learning with motivation and maximising learning effectiveness.

References

Alsawaier, R. S. (2018). The effect of gamification on motivation and engagement. *The International Journal of Information and Learning Technology, 35*(1), 56-79.

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological Review, 84*(2), 191.

Bandura, A., & Walters, R.H. (1963). *Social learning and personality development*. Holt Rinehart and Winston.

Branson, R. K. (1978). The interservice procedures for instructional systems development. *Educational Technology, 18*(3), 11-14.

Brophy, J. (1987) Synthesis of research on strategies for motivating students to learn. *Educational Leadership, 45*(2), 40–48.

Connolly, T. M., & Begg, C. E. (2006). A constructivist-based approach to teaching database analysis and design. *Journal of Information Systems Education, 17*(1), 43-53.

Curriculum Development Council [CDC], & Hong Kong Examinations and Assessment Authority [HKEAA]. (2015). *Information and Communication Technology Curriculum and Assessment Guide (Secondary 4 - 6)*. https://www.edb.gov.hk/attachment/en/curriculum-development/kla/technology-edu/curriculum-doc/ICT_CAGuide_e_2015.pdf

DeGrace, P., & Stahl, L. H. (1990). *Wicked problems, righteous solutions*. Yourdon Press.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September). From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments* (pp. 9-15).

Dettmers, S., Trautwein, U., Lüdtke, O., Kunter, M., & Baumert, J. (2010). Homework works if homework quality is high: Using multilevel modeling to predict the development of achievement in mathematics. *Journal of Educational Psychology, 102*(2), 467.

Dæhli, O., Kristoffersen, B., Lauvås jr, P., & Sandnes, T. (2021). Exploring feedback and gamification in a data modeling learning tool. *Electronic Journal of e-Learning, 19*(6), 559-574.

Hamzah, M. L., Rukun, K., Rizal, F., Purwati, A. A., Hamzah, H., & Zarnelly, Z. (2019). A review of increasing teaching and learning database subjects in computer science. *Revista Espacios, 40*(26).

Kurt, S. (2018, December 16). *ADDIE model: Instructional design*. Educational Technology. https://educationaltechnology.net/the-addie-model-instructional-design/

Lachman, R., Lachman, J. L., & Butterfield, E. C. (2015). *Cognitive psychology and information processing: An introduction*. Psychology Press.

Lai, P. P. (2020, December). Engaging Students in SQL Learning by Challenging Peer during the Pandemic. In *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering* (pp. 205-212). IEEE.

Lister, M. (2015). Gamification: The effect on student motivation and performance at the post-secondary level. *Issues and Trends in Educational Technology*, *3*(2), 1-22.

Mekler, E. D., Brühlmann, F., Tuch, A. N., & Opwis, K. (2017). Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior*, *71*, 525-534.

Migler, A., & Dekhtyar, A. (2020). Mapping the SQL learning process in introductory database courses. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 619–625.

Mills, R. J., Dupin-Bryant, P. A., Johnson, J. D., & Beaulieu, T. Y. (2015). Examining learning styles and perceived benefits of analogical problem construction on SQL knowledge acquisition. J*ournal of Information Systems Education, 26*(3), 203–217.

Mitrovic, A. (1998). Learning SQL with a computerized tutor. *ACM SIGCSE Bulletin, 30*(1), 307–311.

Petri, H. L. (1996). *Motivation: theory, research and applications*. Brooks.

Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist*, *50*(4), 258-283.

Prior, J. C., & Lister, R. (2004). The backwash effect on SQL skills grading. *ACM SIGCSE Bulletin, 36*(3), 32–36.

Schunk, D. H., & Zimmerman, B. J. (Eds.). (2012). *Motivation and self-regulated learning: Theory, research, and applications*. Routledge.

Shin, S. S. (2020). Structured Query Language learning: Concept map-based instruction based on cognitive load theory. *IEEE Access, 8*, 100095-100110.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction, 4*(4), 295–312.

Sweller, J. (2011). Cognitive load theory. In *Psychology of learning and motivation* (pp. 37–76). Academic Press.

Taipalus, T., Siponen, M., & Vartiainen, T. (2018). Errors and complications in SQL query formulation. *ACM Transactions on Computing Education, 18*(3), 1-29.

Tong, H. M., (2020, June 12). ICT jap daai hok mou jau sai hok saang m gaan [Students are not choosing to take ICT as no advantage for university entry]. *Ta Kung Pao*, A8. https://dw-media.tkww.hk/epaper/tkp/20200612/A8_Screen.pdf

Tuparov, G., & Keremedchiev, D. (2020). Assessing students' SQL knowledge and skills in gamification manner. In *2020 43rd International Convention on Information, Communication and Electronic Technology* (pp. 1531-1536). IEEE.

Vroom, V. H. (1964). *Work and motivation.* Wiley.

Wigfield, A., & Cambria, J. (2010). Achievement motivation. In I. B. e & W. E. Craighead (Eds.), *The Corsini Encyclopedia of Psychology* (4th ed., Vol. 1, pp. 14–15). Wiley.

Wiggins, G. P., & McTighe, J. (2005). *Understanding by design*. Association for Supervision & Curriculum Development.

Appendices

List

The Education University of Hong Kong
INT4901 Honours Project II
Scheme of Work

Module Elective Option A – Database **Lesson Duration** 1 hour per lesson
Unit SQL **Teaching Hours** 10 hours

Lesson	Learning Objectives in C&A Guide	Learning Outcomes *By completing the lessons, students will be able to…*	Activity	Assignment
1	1.1. Create simple relational database (p.38) 1.2. Use SQL to maintain a simple relational database (p.39)	**Database Definition (1)** 1. explain the steps of creating a database 2. list data types available in MySQL 3. create table with appropriate data types and constraints 4. modify the structure of the tables SQL 1. CREATE TABLE, INT, CHAR 2. NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY 3. ALTER/ DROP TABLE, ADD/ MODIFY/ DROP COLUMN	1. Mini-database (1) - Database Design Students will be instructed to work in groups. Each group will have to create a database for collecting specific information for their classmate (e.g., birthdays, favourite food or singer, etc.) with at least 5 fields and 2 tables.	1. Database [Classwork] Students will be instructed to submit the database file. 2. Worksheet [Classwork] Students will be instructed to put their database design on the worksheet.
2	2.1. Use SQL to manipulate a simple relational database's data (p.39)	**Database Definition (2) and Data Insertion** 1. define an index for a table by using SQL statements 2. add, delete and modify the data in the tables by using SQL statements SQL 1. CREATE INDEX 2. INSERT INTO…VALUES, UPDATE…SET…WHERE, DELETE…FROM…WHERE	2. Mini-database (2) - Data Collection Students continue working on their database in the last lesson. Students will be invited to collect the specified data. Students will be instructed to insert their data into the database.	1. Database [Classwork] Students will be instructed to submit the database file. 2. Worksheet [Classwork] Students will be instructed to write down the SQL statements to insert the data.

Lesson	Learning Objectives in C&A Guide	Learning Outcomes *By completing the lessons, students will be able to…*	Activity	Assignment
3	3.1. Use SQL to retrieve the required information (p.39) 3.2. Use appropriate operators and expressions such as arithmetic operators and expressions, and comparison operators to perform specific operations (p.39) 3.3. view and select contents by filtering (p.39)	**Data Selection (1)** 1. view all records in the tables by using SQL statements 2. select data by filtering and performing arithmetic operations by using SQL statements SQL 1. SELECT…FROM…WHERE, * 2. >, <, =, >=, <=, <>, NOT, AND, OR, IS NULL 3. +, -, *, /, % [mod]	3. Mini-database (3) - Presentation Students will be invited to present the data collected in the last lesson, using the SQL statement to select relevant records.	1. Online SQL Game System [Homework] Students will be instructed to login to complete the task.
4	4.1. Use appropriate operators and expressions such as the logical operators and the between and like operators to perform specific operations (p.39) 4.2. view, sort and select contents by filtering (p.39)	**Data Selection (2)** 1. filter for unique results 2. sort query results with fields 3. select data in a specified pattern SQL 1. DISTINCT 2. ORDER BY ASC/DESC 3. BETWEEN…AND, LIKE, %, _	1. Data Searcher Students will get into groups. Students will be given a sample database. Students' mission is to find the specified data using the skills learnt as fast as possible.	1. Online SQL Game System [Homework] Students will be instructed to login to complete the task.
5	5.1. Use simple built-in functions such as aggregate functions (p.39)	**Built-in Functions (1)** 1. retrieve information with aggregate functions 2. retrieve information with date functions SQL 1. MAX, MIN, SUM, AVG, COUNT 2. DATE, DAY, MONTH, YEAR	1. Utilising Functions Students will get into groups. Students will be instructed to construct SQL statements for their database using aggregate and date functions.	

Lesson	Learning Objectives in C&A Guide	Learning Outcomes *By completing the lessons, students will be able to…*	Activity	Assignment
6	6.1. Use simple built-in functions such as string functions (p.39)	**Built-in Functions (2)** 1. retrieve information with numeric functions 2. retrieve information with string functions 3. perform queries for information in terms of groups <u>SQL</u> 1. ABS 2. LEN, UPPER, LOWER, TRIM, SUBSTRING 3. GROUP BY, HAVING	1. Guess-and-try Students' mission is to guess the function used from the results given as quick as possible.	1. Online SQL Game System [Homework] Students will be instructed to login to complete the task.
7	7.1. Perform queries on multiple tables, including the use of natural join and equi-join (p.39)	**Multi-table Query (1)** 1. describe the use of inner join 2. perform queries using equi-join 3. perform queries using natural join <u>SQL</u> 1. INNER JOIN 2. NATURAL JOIN	1. Accept or Reject? Students will be given SQL statements with sample records. Students' mission is to determine whether the record would be selected by shouting 'accept' or 'reject'. 2. The Librarian Students will get into groups. Students' mission is to construct a SQL statement to find the information according to the instructions as fast as possible.	
8	8.1. Perform queries on multiple tables, including the use of outer join (p.39)	**Multi-table Query (2)** 1. describing the use of outer join 2. performing queries using left outer join 3. performing queries using right outer join 4. performing queries using full outer join <u>SQL</u> 1. LEFT [OUTER] JOIN, RIGHT [OUTER] JOIN, FULL [OUTER] JOIN	1. Speed-reader Students will be given various Venn diagram or query results. Students' mission is to match the Venn diagram with the results or the join type as fast as possible.	1. Online SQL Game System [Homework] Students will be instructed to login to complete the task.

Lesson	Learning Objectives in C&A Guide	Learning Outcomes *By completing the lessons, students will be able to…*	Activity	Assignment
9	9.1. Perform sub-query (for one sub-level only) (p.39) 9.2. Use appropriate operators and expressions such as the logical operators and the in operators to perform specific operations (p.39)	**Sub-query** 1. performing queries with sub-query utilising testing operators 2. performing queries using set operators <u>SQL</u> 1. IN, EXISTS, ALL, ANY 2. UNION, INTERSECT, MINUS	1. SQL King Students will be given various usage of sub-queries. Students' mission is to match the operators as fast as possible. Students may choose the options by moving to the left, middle or right of the classroom.	1. Online SQL Game System [Homework] Students will be instructed to login to complete the task.
10	10.1. Use access rights to achieve data privacy (p.40)	**Access Rights and Summary** 1. create different views to limit access rights <u>SQL</u> 1. CREATE VIEW…AS	4. Mini-database (4) – System Design Students will get back into the groups for the mini-database activity. Students may design 3 system function with 3 types of SQL statement.	1. Worksheet [Homework] Students will be instructed to write down the SQL statements using specified statement.

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 1

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Database Definition (1)

A. Students' Prior Knowledge

1. Usage of Structured Query Language (SQL) in Database Management System (DBMS)
2. Hierarchical organisation of data (Database, Table, Record, Field)
3. Simple data types (Integer, Real, Character, Boolean)
4. Basic relational database concepts (Entity, Relationship, Attribute, Domain, Integrity, Index)

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. explain the steps of creating a database;
2. list data types available in MySQL;
3. create table with appropriate data types and constraints by using SQL statements; and
4. modify the structure of the tables by using SQL statements.

SQL

1. CREATE TABLE, INT, CHAR
2. NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY
3. ALTER/ DROP TABLE, ADD/ MODIFY/ DROP COLUMN

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedures

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (5 mins)				
5		**Recalling Prior Knowledge** Students' prior knowledge will be recalled by mentioning the use of SQL in DBMS and hierarchical organisation of data. Students will be invited to share a real-life example to describe the functions of SQL for databases. SQL is a computer language to give instructions to manipulate DBMS for query. The organisation of data in computer system could be identified in the following order: (1) Database > (2) Table > (3) Record > (4) Field **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including the steps of creating a database, data types in MySQL, and how to create table with appropriate constraints using SQL statements.	1. Questioning Students will be invited to describe the use of SQL with an example.	Textbook (Page 116 – 117)
Direct Instruction 1 (10 mins)				
10	LO1 LO2	**Steps of Creating a Database** The following steps of creating a database will be introduced to students: 1. Defining a new database 2. Defining entities (tables) and attributes (fields) 3. Defining appropriate constraints and relationships 4. Creating index for the tables Available data types will be shown with the projector. An example topic (library borrow record) will be illustrated to demonstrate the steps of creating a database with 3 tables. Students will be invited to suggest attributes and data types.	2. Questioning Students will be invited to suggest appropriate attributes and data types of the given example.	Textbook (Page 118 – 122) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 1 (10 mins)				
10	LO1 LO2	**Mini-database (1) - Database Design** Students will be instructed to work in groups. Each group will have to create a database for collecting specific information from their classmates (e.g., birthdays, favourite food or singer, etc.) with at least 2 tables and 5 fields each. Students will be instructed to brainstorm a topic for their group within 1 minute. Then, students will be instructed to discuss the design of the entities and attributes. Their design should be written on the worksheet.	3. Worksheet Students will be instructed to draft their database design on the worksheet. 4. Observation The teacher will walk around the classroom and check students' progress.	Textbook (Page 118 – 122) Worksheet
Direct Instruction 2 (15 mins)				
5	LO3	**Creating Table** The syntax of the CREATE statement will be introduced to the students. The naming practice for tables and fields will be reminded. The example database discussed in *Direct Instruction 1* will be used to demonstrate how to construct a CREATE statement.		Textbook (Page 121, 125)
10	LO3	**Constraints** The constraints available (including NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY) will be introduced to students. Students will be guided to discuss the necessity of adding constraints into tables to ensure data validity and integrity within 3 minutes. Then, each group will be invited to share 1 problem that might occur. The syntax of the following constraints will be introduced to students: 1. NULL 2. UNIQUE 3. PRIMARY KEY 4. FOREIGN KEY	5. Group Discussion and Sharing Students will be invited to discuss 3 problems might possibly occur without appropriate constraints.	Textbook (Page 127 – 131)

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 2 (15 mins)				
15	LO4	**Mini-database (1) - Database Design** Students will be instructed to create a database on an online SQL complier. After students successfully creating the database, the syntax of ALTER statement will be introduced to the students. Students will be instructed to add/modify/drop the columns. After completing the given tasks, students will be instructed to download the database file and submit it.	6. Database File Students will be required to submit the database file exported from the online platform.	Textbook (Page 135 – 136) Worksheet
Conclusion (5 mins)				
5	LO1 LO2 LO3 LO4	Students will be asked the following questions to strengthen their short-term memory. 1. Steps of creating a database 2. Data types available in MySQL 3. Syntax of the CREATE and ALTER statements and constraints	7. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 116 – 136) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 2

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Database Definition (2) and Data Insertion

A. Students' Prior Knowledge

1. Use of constraints in a database table
2. Concept of index in database

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. define an index for a table by using SQL statements; and
2. add, delete and modify the data in the tables by using SQL statements.

SQL

1. CREATE INDEX
2. INSERT INTO…VALUES, UPDATE…SET…WHERE, DELETE…FROM…WHERE

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedures

Time (mins)	LO	Activity	Assessment	Material
Introduction (10 mins)				
10	LO1	**Recalling Prior Knowledge** Students have learnt steps of creating a database, data types in MySQL, and how to create table with appropriate constraints. Students will be asked about the following items to consolidate their memory. 1. Steps of creating a database 2. Data types available in MySQL 3. Syntax of the CREATE and ALTER statements and constraints Students' prior knowledge will be recalled by mentioning the use of index. **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including defining an index, and how to add, delete and modify the data in the tables using SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 116 – 136) Worksheet 1
Direct Instruction 1 (10 mins)				
10	LO1	**Creating Index** The use of index file will be introduced to students. An index is a small file that keeps the sorted values and the actual location of the records in a table. Students will be guided to discuss the selection of fields as an index key. The example raised in last lesson will be used as a discussion topic. Students should aware that the selection should base on the needs of searching for the database, and the primary key will be indexed automatically. The syntax of the CREATE INDEX statement will be introduced to the students.	2. Questioning Students will be invited to suggest which field should be used as an index key.	Textbook (Page 132 – 133)
Activity (10 mins)				
10	LO2	**Mini-database (2a) - Data Collection** Students will be instructed to work in groups. Students will continue to work on their database created in the last lesson. Students will be invited to collect the specified data from their classmates.	3. Worksheet Students will be instructed to write down the collected data on the worksheet.	Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Direct Instruction 2 (10 mins)				
10	LO2	**Inserting Data** The syntax of the INSERT INTO statement will be introduced to the students. The example raised in last lesson will be used to demonstrate how to construct an INSERT INTO statement. After demonstrating inserting a single record, the syntax for inserting multiple records and inserting data into specific column will be demonstrated.		Textbook (Page 133 – 134)
Activity (15 mins)				
15	LO2	**Mini-database (2b) - Data Collection** Students will be instructed to upload the database created in the last lesson to an online SQL complier. Students will be instructed to insert their data into the database. After students successfully inserting the values, the syntax of the UPDATE and DELETE statements will be introduced to the students. Students will be instructed to update/delete records. After completing the given tasks, students will be instructed to download the database file and submit it.	4. Database File Students will be instructed to submit the database file exported from the online platform.	Textbook (Page 133 – 134) Worksheet
Conclusion (5 mins)				
5	LO1 LO2	Students will be asked about the following items to strengthen their short-term memory. 1. Syntax of the CREATE INDEX statement 2. Syntax of the INSERT INTO, UPDATE, DELETE statements	5. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 132 – 134)

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 3

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Data Selection (1)

A. Students' Prior Knowledge

1. Practise data extraction and manipulation by querying
2. Tracing and interpreting simple SQL statements

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. view all records in the tables by using SQL statements; and
2. select data by filtering and performing arithmetic operations by using SQL statements.

SQL

1. SELECT…FROM…WHERE, *
2. >, <, =, >=, <=, <>, NOT, AND, OR, IS NULL
3. +, -, *, /, % [mod]

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedures

Time (mins)	LO	Activity	Assessment	Material
Introduction (10 mins)				
10	LO1	**Recalling Prior Knowledge** Students have learnt the definition of index, and how to add, delete and modify the data in the tables with SQL statements. Students will be asked about the following items to consolidate their memory. 1. Syntax of the CREATE INDEX statement 2. Syntax of the INSERT INTO, UPDATE, DELETE statements Students' prior knowledge will be recalled by mentioning the use of basic SQL statements. **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including viewing all records and selecting data by filtering and performing arithmetic operations using SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 132 – 134, 147 – 148) Worksheet 2
Direct Instruction 1 (15 mins)				
5	LO1	**Viewing All Records** The use of SELECT statement will be reminded. Students will be invited to share how to construct a statement to select all data. The syntax of SELECT statement for selecting specific fields will be recalled.	2. Questioning Students will be asked what operators could be used to view all records.	Textbook (Page 148 – 150) Worksheet
10	LO2	**Filtering Records** Students will be invited to share how to construct a statement to filter some records under certain condition. The usage of the following operators will be introduced one by one: 1. >, <, =, >=, <=, <> 2. NOT, AND, OR, IS NULL 3. +, -, *, /, % [mod] The example raised in Lesson 1 will be used to demonstrate how to construct the SQL statements. Students will be guided to discuss the meaning of the query results.	3. Questioning Students will be asked what operators could be used for selecting records under certain condition.	Textbook (Page 152 – 154) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 1 (15 mins)				
15	LO1 LO2	**Mini-database (3) - Data Presentation** Students will be instructed to work in groups. Students will be invited to present the data collected in the last lesson, using the operators just taught to construct SQL statements with meaningful results to filter the records. Each student should construct 1 statement, and each group should have used all three types of operators. Each group will present one of the constructed SQL statements and its purpose. For example, students may have constructed a statement to find out whose favourite singer is 'Taylor Swift'. Students should aware that SQL statements are designed for a meaningful purpose in the information systems.	4. Worksheet Students will be instructed to construct SQL statements to find meaningful results from their database. 5. Verbal Presentation Students will present their SQL statements and its purpose.	Textbook (Page 148 – 154) Worksheet
Direct Instruction 2 (10 mins)				
10		**Introducing Online SQL Game Platform** As a learning component of the course package, the online SQL game platform will be introduced to students. The features and the storyline of the game platform will be introduced to students. Students will be encouraged to complete the tasks daily.		Worksheet
Activity 2 (5 mins)				
5		**Trial Login and Changing Password** Students will be instructed to login the online SQL game platform. After signing in the system, students will be requested to change the password immediately.		Worksheet
Conclusion (5 mins)				
5	LO1 LO2	Students will be asked about the following items to strengthen their short-term memory. 1. Syntax of the SELECT statement for all and specified fields 2. Usage of the comparison, logical and arithmetic operators	6. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 147 – 154) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 4

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Data Selection (2)

A. Students' Prior Knowledge

1. Selecting data by filtering and performing arithmetic operations

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. filter for unique results by using SQL statements;
2. sort query results with fields by using SQL statements; and
3. select data in a specified pattern by using SQL statements.

SQL

1. DISTINCT
2. ORDER BY ASC/DESC
3. BETWEEN…AND, LIKE, %, _

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)
4. Mission Card

D. Teaching Procedures

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (10 mins)				
10	LO1	**Recalling Prior Knowledge** Students have learnt the how to select data by filtering and performing arithmetic operations. Students will be asked about the following items to consolidate their memory. 1. Syntax of the SELECT statement for all and specified fields 2. Usage of the comparison, logical and arithmetic operators **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including filtering for unique results, sorting query results with fields, and selecting data in a specified pattern with the use SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 147 – 154) Worksheet 3
Direct Instruction 1 (15 mins)				
5	LO1	**Filtering for Unique Results** The use of SELECT statement and operators will be reminded. Students should aware that the design of SQL statements should facilitate certain purposes in the information systems. Students will be guided to discuss scenarios that unique results are necessary. The syntax of DISTINCT statement will be introduced to students.	2. Questioning Students will be asked under given scenarios whether using the operators learnt in Lesson 3 are desirable and its reason.	Textbook (Page 159 – 160) Worksheet
10	LO2	**Sorting Query Results** Students will be guided to discuss scenarios that returning unsorted query results could be a problem. For example, the data returned from a 100-metre sprint required to be sorted in descending order for further processing in an Athletic Meet. The syntax of the ORDER BY statement will be introduced to the students. The example raised in previous lessons will be used to demonstrate how to construct an ORDER BY statement.	3. Questioning Students will be invited to share real-life scenarios when using ORDER BY statement is necessary.	Textbook (Page 161 – 163) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Direct Instruction 2 (15 mins)				
5	LO3	**Selecting Data Within a Range** Students will be asked how to construct a SQL statement which returns the values within a specified range. Students should be able to suggest using the AND operator with two expressions to limit the range. Students will be guided to discuss the risks. Students should aware that the missing of 'equal or' is a common mistake. The syntax of the BETWEEN statement will be introduced to the students.	4. Questioning Students will be asked how to construct a SQL statement for selecting data within a specified range.	Textbook (Page 153, 155) Worksheet
10	LO3	**Selecting Data in a Specified Pattern** Students will be guided to discuss how to construct a SQL statement returns the values under similar pattern. For example, how to retrieve records of all form 1 students. Students should be able to suggest using the AND operator with many expressions to specify the required values. The syntax of the LIKE statement and the wildcards (_, %) will be introduced to the students.	5. Questioning Students will be asked how to construct a SQL statement for selecting data in a specified pattern.	Textbook (Page 153, 156 – 157) Worksheet
Activity (15 mins)				
15	LO1 LO2 LO3	**Game – Data Searcher** Students will be instructed to work in groups. Students will be given a sample database and a mission card. Students' mission is to find the specified data using the skills learnt as fast as possible. With each correct attempt, the group will be awarded with 100 points.	6. Game Students will be instructed to find query results of the SQL statements on the worksheet. 7. Observation The teacher will walk around the classroom and check students' progress.	Mission Card Worksheet
Conclusion (5 mins)				
5	LO1 LO2 LO3	Students will be asked about the following items to strengthen their short-term memory. 1. Syntax of the DISTINCT statement and its usage 2. Syntax of the ORDER BY statement and the ordering 3. Syntax of the BETWEEN and LIKE statements	8. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 153, 155 – 157, 159 – 163)

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 5

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Built-in functions (1)

A. Students' Prior Knowledge

1. Use of spreadsheets functions
2. Data types available in MySQL
3. Selecting data by filtering and performing arithmetic operations

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. retrieve information with aggregate functions by using SQL statements; and
2. retrieve information with date functions by using SQL statements.

SQL

1. MAX, MIN, SUM, AVG, COUNT
2. DATE, DAY, MONTH, YEAR

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedures

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (10 mins)				
10	LO1	**Recalling Prior Knowledge** Students have learnt how to filter for unique results, sort query results and select data in a specific pattern using SQL statements. Students will be asked about the following items to consolidate their memory. 1. Syntax of the DISTINCT statement and its usage 2. Syntax of the ORDER BY statement and the ordering 3. Syntax of the BETWEEN and LIKE statements Students' prior knowledge will be recalled by mentioning the use of spreadsheet functions. **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including how to retrieve information with aggregate and date functions by using SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 153, 155 – 157, 159 – 163) Worksheet 4
Direct Instruction 1 (15 mins)				
5	LO1	**Built-in Functions** The use of built-in functions in SQL will be introduced. Students will be invited to suggest example use of spreadsheets functions. Then, the syntax of built-in functions in SQL will be introduced.	2. Questioning Students will be invited to share spreadsheets functions in real-life context.	Textbook (Page 157) Worksheet
10	LO1	**Aggregate Functions** The use of functions in spreadsheets will be related to database functions. The usage of following functions will be introduced one by one: 1. MAX, MIN 2. SUM 3. AVG 4. COUNT The example raised in previous lessons will be used to demonstrate how to construct a statement utilising the aggregate functions.		Textbook (Page 158 – 160) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 1 (15 mins)				
15	LO1	**Utilising Aggregate Functions** Students will be instructed to work in groups. Students will be instructed to construct SQL statements for their database using aggregate functions on the worksheet. Students should write down the purposes of the SQL statement.	3. Worksheet Students will be instructed to construct SQL statements with the use of the functions.	Textbook (Page 158 – 160) Worksheet
Direct Instruction 2 (5 mins)				
5	LO2	**Date Functions** The data type of date and time will be reminded. Students will be guided to discuss how to compare dates in database. Students should aware that dates are similar to strings while it could be converted to numerical data with built-in functions. The usage of following functions will be introduced one by one: 1. DATE 2. DAY, MONTH, YEAR The example raised in previous lessons will be used to demonstrate how to construct a statement utilising the date functions.	4. Questioning Students will be asked about the how to compare dates in database.	Textbook (Page 158 – 160) Worksheet
Activity 2 (10 mins)				
10	LO2	**Utilising Date Functions** Students will be instructed to work in groups. Students will be instructed to construct SQL statements for their database using date functions on the worksheet. Students should write down the purpose of the SQL statement.	5. Worksheet Students will be instructed to construct SQL statements with the use of the functions.	Textbook (Page 158 – 160) Worksheet
Conclusion (5 mins)				
5	LO1 LO2	Students will be asked about the following items to strengthen their short-term memory. 1. Syntax of using aggerate functions 2. Syntax of using date functions	6. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 157 – 160) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 6

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Built-in functions (2)

A. Students' Prior Knowledge

1. Data types available in MySQL
2. Selecting data by filtering and performing arithmetic operations
3. Retrieving information with aggregate and date functions

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. retrieve information with numeric functions by using SQL statements;
2. retrieve information with string functions by using SQL statements; and
3. perform queries for information in terms of groups.

SQL

1. ABS
2. LEN, UPPER, LOWER, TRIM, SUBSTRING
3. GROUP BY, HAVING

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedures

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (5 mins)				
5		**Recalling Prior Knowledge** Students have learnt how to retrieve information with aggregate and date functions by using SQL statements. Students will be asked about the following items to consolidate their memory. 1. Syntax of using aggerate functions 2. Syntax of using date functions **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including how to retrieve information with numeric and string functions by using SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 157 – 160) Worksheet 5
Direct Instruction 1 (15 mins)				
5	LO1	**Numeric Functions** The usage of following functions will be introduced one by one: 1. ABS The example raised in previous lessons will be used to demonstrate how to construct a statement utilising the numeric functions.	2. Worksheet Students will be instructed to construct SQL statements with the use of the functions.	Worksheet
10	LO2	**String Functions** The usage of following functions will be introduced one by one: 1. LEN 2. UPPER, LOWER 3. TRIM 4. SUBSTRING The example raised in previous lessons will be used to demonstrate how to construct a statement utilising the string functions.	3. Worksheet Students will be instructed to construct SQL statements with the use of the functions.	Textbook (Page 160 – 161)

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 1 (5 mins)				
5	LO1 LO2	**Game – Guess-and-try** Students' mission is to guess the function used from the results given as quick as possible. Students should recall their memory on the built-in functions learnt in these two lessons. The activity aims to assist students to organise their learning on the usage, syntax, and the results of the built-in functions.	4. Activity Students understanding on the function usages will be assessed through the functions matching activity.	Textbook (Page 157 – 161) Worksheet
Direct Instruction 2 (20 mins)				
10	LO3	**Queries for Grouped Information** Students will be guided to discuss the needs of grouping information in database. Students should aware that listing summarised information of all instances (of an entity object) cannot be achieved by the WHERE statement. For example, listing the personal best records one-by-one of ALL the athletes in the database. The syntax of the GROUP BY statement will be introduced to the students. The example raised in previous lessons will be used to demonstrate how to construct queries utilising the built-in functions and the GROUP BY statement.	5. Questioning Students will be invited to suggest other example scenarios which grouping records are necessary. 6. Worksheet Students will be instructed to construct the SQL statements on the worksheet.	Textbook (Page 163 – 164) Worksheet
10	LO3	**Filtering Grouped Information** Students will be guided to discuss the usage of filtering the grouped records. Students will be asked how to construct a SQL statement which filter the results from grouped information. For example, how to find the classes which scores higher than 50 marks in average. Students should be able to suggest using the WHERE statement to specify a condition. Students will be guided to discuss the logic flow of the SQL statement interpretation. The WHERE statement may filter the records prepared to be grouped, while it could not be used to filter the results after grouping. The syntax of the HAVING statement will be introduced to the students. The example raised in previous lessons will be used to demonstrate how to construct a statement to filter grouped information.	7. Questioning Students will be asked how to construct a SQL statement which filter the results from grouped information. 8. Worksheet Students will be instructed to construct the SQL statements on the worksheet.	Textbook (Page 165 – 166) Worksheet Textbook (Page 167 – 168)

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 2 (10 mins)				
10	LO3	**Utilising Grouped Information** Students will be instructed to work in groups. Students will be instructed to construct SQL statements for their database using grouped information and filtering on the worksheet. Students should write down the purposes of the SQL statement.	9. Worksheet Students will be instructed to construct SQL statements on the worksheet.	Textbook (Page 157 – 168) Worksheet
Conclusion (5 mins)				
5	LO1 LO2 LO3	Students will be asked about the following items to strengthen their short-term memory. 1. Syntax of using numeric functions 2. Syntax of using string functions 3. Syntax of using the GROUP BY and HAVING statements and its usage	10. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 157 – 168) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 7

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Multi-table Query (1)

A. Students' Prior Knowledge

1. Data redundancy, data integrity and keys
2. Selecting data by filtering and performing arithmetic operations
3. Performing queries for information in terms of groups

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. describe the use of inner join;
2. perform queries using equi-join with SQL statements; and
3. perform queries using natural join with SQL statements.

SQL

1. INNER JOIN
2. NATURAL JOIN

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedures

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (20 mins)				
10		**Recalling Prior Knowledge** Students have learnt how to retrieve information with numeric and string functions with SQL statements and perform queries for information in terms of groups. Students will be asked about the following items to consolidate their memory. 1. Syntax of using numeric functions 2. Syntax of using string functions 3. Syntax of using the GROUP BY and HAVING statements and its usage **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including perform queries on multiple tables with SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 157 – 168) Worksheet 6
10	LO1	**Multi-table Query** Students will be guided to discuss the needs of retrieving information from multiple tables. The example from the textbook (p.188) will be used to facilitate discussion. Students should aware that it is necessary to perform multi-table queries as the data spread across different tables. Students will be guided to discuss with their groups the reasons of not storing all data in the same table. Students should aware that relational database are usually designed with multiple tables, which avoids data redundancy and ensures data integrity. *[The concept of UNF or NoSQL should not be covered to avoid confusions at this early stage.]* Students will be reminded the Primary Key will be used to create linkage between tables, which identifies as Foreign Key.	2. Questioning Students will be invited to share their thoughts on the needs of retrieving information from multiple tables. 3. Group Discussion Students will be invited to discuss the reasons of not storing all data in the same table.	Textbook (Page 187 – 189) Worksheet Textbook (Page 190 – 193)

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Direct Instruction 1 (10 mins)				
5	LO1	**Inner Join and Outer Join** Two major types of joins will be introduced to students: inner join and outer join. The usage of inner join will be introduced to students. Students should aware that only the values matched between two specified fields in two tables will be selected.		Textbook (Page 194)
5	LO2	**Equi-join** The use of equi-join (filtering the joined results base on the fields with equal sign) will be introduced to students. The syntax of using an equi-join in a SELECT statement will be introduced to the students.	4. Worksheet Students will be instructed to construct the SQL statements using equi-join.	Textbook (Page 194 – 197) Worksheet
Activity 1 (5 mins)				
5	LO2	**Game – Accept or Reject?** Students will be given SQL statement using equi-join with sample records. The records will be presented one by one. Students' mission is to determine whether the record would be selected by the query. If students believe the record will be selected under the statement, students may shout 'accept'; or 'reject' otherwise. With each correct choice, the class will be awarded with 100 points. Students will be invited to share their thoughts on the selection.	5. Game Students' understanding of the topic will be assessed through the game.	
Direct Instruction 2 (5 mins)				
5	LO3	**Natural Join** The use of natural join (matching the values of the fields with the same name and eliminating the repeated fields) will be introduced to students. The syntax of using a natural join in a SELECT statement will be introduced to the students.	6. Worksheet Students will be instructed to construct the SQL statements using natural join.	Textbook (Page 197 – 198) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity 2 (15 mins)				
15	LO2 LO3	**Game – The Librarian** Students will be instructed to get into groups. In this game, students will become librarian assistants who assist for finding books, tracking circulation records and looking up fines. Students' mission is to construct a SQL statement to find the information according to the instructions as fast as possible. For example, students will be asked to find 'which students owe library for more than $10'. With each correct attempt, students will be awarded with 500-100 points according to their speed.	7. Game Students' understanding of the topic will be assessed through the game.	
Conclusion (5 mins)				
5	LO1 LO2 LO3	Students will be asked about the following items to strengthen their short-term memory. 1. Use of inner join 2. Syntax of SQL statements using equi-join 3. Syntax of SQL statements using natural join	8. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 187 – 198) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 8

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Multi-table Query (2)

A. Students' Prior Knowledge

1. Data redundancy, data integrity and keys
2. Selecting data by filtering and performing arithmetic operations
3. Performing queries on multiple tables using inner join

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. describe the use of outer join;
2. perform queries using left outer join with SQL statements;
3. perform queries using right outer join with SQL statements; and
4. perform queries using full outer join with SQL statements.

SQL

1. LEFT [OUTER] JOIN, RIGHT [OUTER] JOIN, FULL [OUTER] JOIN

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedure

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (10 mins)				
10		**Recalling of Prior Knowledge** Students have learnt how to perform queries on multiple tables using inner join. Students will be asked about the following items to consolidate their memory. 1. Use of inner join 2. Syntax of SQL statements using equi-join 3. Syntax of SQL statements using natural join The use of inner join will be reminded. **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including perform queries using outer join with SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 187 – 198) Worksheet 7
Direct Instruction 1 (30 mins)				
5	LO1	**Outer Join** Students will be guided to discuss the needs of retrieving information which is not always appear from both tables. Students will be invited to share scenarios of using outer join. The usage of outer join will be introduced to students. Students should aware that the values will be returned as NULL if no matching record could be found.	2. Questioning Students will be invited to share scenarios of using outer join.	Textbook (Page 199) Worksheet
10	LO2	**Left Outer Join** The use of left outer join (selecting all records of the left table regardless of the existence of matching values) will be introduced to students. The example from the textbook (p.199 – 200) will be used to facilitate discussion. The syntax of the LEFT JOIN statement will be introduced to the students.	3. Worksheet Students will be instructed to construct the SQL statements using left outer join.	Textbook (Page 199 – 201) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
5	LO3	**Right Outer Join** The use of left outer join (selecting all records of the right table regardless of the existence of matching values) will be introduced to students. The example from the textbook (p.202 – 203) will be used to facilitate discussion. The syntax of the RIGHT JOIN statement will be introduced to the students.	4. Worksheet Students will be instructed to construct the SQL statements using right outer join.	Textbook (Page 202 – 203) Worksheet
10	LO4	**Full Outer Join** The use of full outer join (selecting all records from both tables regardless of the existence of matching values) will be introduced to students. The example from the textbook (p.204 – 205) will be used to facilitate discussion.	5. Worksheet Students will be instructed to construct the SQL statements using full outer join.	Textbook (Page 203 – 206) Worksheet
Activity 2 (15 mins)				
15	LO2 LO3 LO4	**Game – Speed-reader** Students will be given various Venn diagram or query results. Students' mission is to match the Venn diagram with the results or the join type as fast as possible. Students will be given a red, yellow, green, and blue cards. Students may indicate their choice with the coloured cards. With each correct attempt, students will be awarded with 500-100 points according to their speed.	6. Game Students' understanding of the topic will be assessed through the game.	
Conclusion (5 mins)				
5	LO1 LO2 LO3 LO4	Students will be asked about the following items to strengthen their short-term memory. 1. Use of outer join 2. Syntax of the LEFT OUTER JOIN statement 3. Syntax of the RIGHT OUTER JOIN statement 4. Syntax of the FULL OUTER JOIN statement	7. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 199 – 206) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 9

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Sub-query

A. Students' Prior Knowledge

1. Using nested functions (e.g. IF) in spreadsheets
2. Selecting data by filtering and performing arithmetic operations
3. Performing queries on multiple tables

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. perform queries with sub-query utilising testing operators; and
2. perform queries using set operators.

SQL

1. IN, EXISTS, ALL, ANY
2. UNION, INTERSECT, MINUS

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

D. Teaching Procedure

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (10 mins)				
10		**Recalling of Prior Knowledge** Students have learnt how to perform queries on multiple tables using outer join. Students will be asked about the following items to consolidate their memory. 1. Use of outer join 2. Syntax of the LEFT OUTER JOIN statement 3. Syntax of the RIGHT OUTER JOIN statement 4. Syntax of the FULL OUTER JOIN statement **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including perform sub-query utilising testing operators, and perform queries utilising set operators with SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 199 – 206) Worksheet 8
Direct Instruction 1 (20 mins)				
10	LO1	**Sub-query with Single Value** A mathematical expression with round brackets, 6 ÷ 2(1 + 2), will be shown to students. Students will be guided to discuss answer of the expression and the steps of calculation. Students should aware that the expression inside the round brackets should be calculated first. The use of nested functions (like IF) in spreadsheet will be reminded. The use of sub-query with single value will be introduced to students. Students should aware that a sub-query is a query nested in another query. The example from the textbook (p. 207) will be used to facilitate discussion.	2. Discussion Students will be invited to discuss how to calculate and share the steps of his/her calculations. 3. Worksheet Students will be instructed to construct the SQL statements using sub-query with single value.	Textbook (Page 207) Worksheet
10	LO1	**Sub-query with Multiple Values** The use of sub-query with multiple values will be introduced to students. Students should aware that the results returned with multiple values will be identified as set. The example from the textbook (p. 208 – 209) will be used to facilitate discussion. The usage of the following testing operators will be introduced one by one: 1. IN 2. EXISTS, ALL, ANY	4. Worksheet Students will be instructed to construct the SQL statements using sub-query with multiple values and testing operators.	Textbook (Page 208 – 209) Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Activity (10 mins)				
10	LO1	**Game – SQL King** Students will be invited to stand up to play this game. Students will be given various usage of sub-queries. Students' mission is to match the operators as fast as possible. Students may choose the options by moving to the left, middle or right of the classroom. With each correct attempt, students will be awarded with 300-100 points according to their speed.	5. Game Students' understanding of the topic will be assessed through the game.	
Direct Instruction 2 (15 mins)				
15	LO2	**Set Operators** The set operations will be introduced to students. Students should aware that the set operators are used to join the results of the sub-queries. The usage of the following set operators will be introduced one by one: 1. UNION 2. INTERSECT 3. MINUS	6. Worksheet Students will be instructed to construct the SQL statements using the set operators.	Textbook (Page 86 – 89) Worksheet
Conclusion (5 mins)				
5	LO1 LO2	Students will be asked about the following items to strengthen their short-term memory. 1. Usage of sub-queries 2. Syntax of the sub-queries with testing operators (IN, EXISTS, ALL, ANY) 3. Syntax of the queries with set operators (UNION, INTERSECT, MINUS)	7. Questioning Students will be asked about the relevant concepts taught in this lesson.	Textbook (Page 86 – 89, 207 – 209) Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Lesson Plan 10

Module	Elective Option A – Database	**Lesson Duration**	1 hour
Unit	Structured Query Language (SQL)	**Topic**	Access Rights and Summary

A. Students' Prior Knowledge

1. Creating table with appropriate data types and constraints
2. Needs of enforcing access rights

B. Learning Objectives (LO)

By completing this lesson, students will be able to…

1. create different views to limit access rights.

SQL

1. CREATE VIEW…AS

C. Resources

1. Textbook (Longman New Senior Secondary Information and Communication Technology: Elective A1)
2. Presentation Slides
3. Worksheet (with notes)

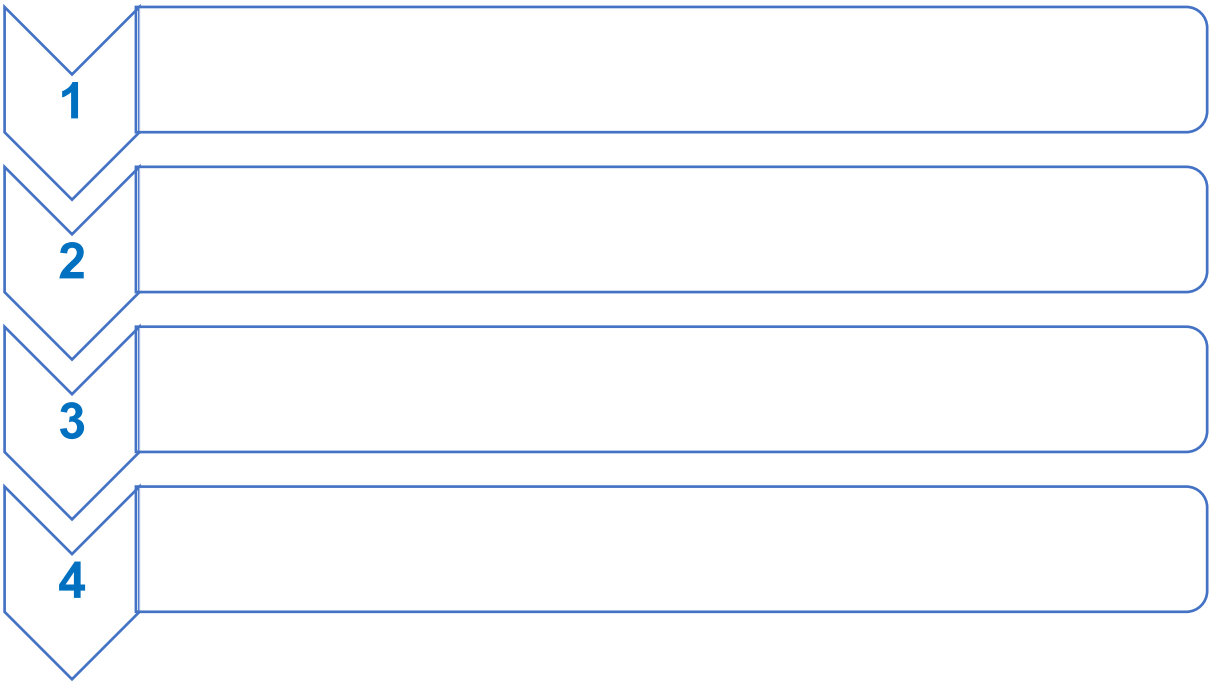
D. Teaching Procedure

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
Introduction (10 mins)				
10		**Recalling of Prior Knowledge** Students have learnt how to perform sub-query utilising testing operators and perform queries utilising set operators. Students will be asked about the following items to consolidate their memory. 1. Usage of sub-queries 2. Syntax of the sub-queries with testing operators (IN, EXISTS, ALL, ANY) 3. Syntax of the queries with set operators (UNION, INTERSECT, MINUS) **Informing Learning Objectives** Students will be informed the learning objectives of this lesson, including how to limit access rights by creating different views with SQL statements.	1. Questioning Students will be asked about the concepts taught in the last lesson.	Textbook (Page 86 – 89, 207 – 209) Worksheet 9
Direct Instruction 1 (10 mins)				
10	LO1	**Access Control** Students will be invited to discuss the needs of enforcing access rights. Students should aware that views may enforce data security by restricted users from accessing confidential data. The syntax of the CREATE VIEW statement will be introduced to students.	2. Discussion Students will be invited to discuss the needs of enforcing access rights. 3. Worksheet Students will be instructed to construct a SQL statement to create view.	Textbook (Page 173 – 174) Worksheet
Activity (30 mins)				
15		**Mini-database (4) - System Design** Students will be instructed to work in groups for the mini-database activity in Lesson 1 – 3. Students' mission is to enhance their initial design to a functional system with 3 applications. Students will be reminded that SQL statements are designed for a meaningful purpose in the information systems. Students in Lesson 3 have created 3 meaningful queries. In this lesson, the functions should be designed for specified users and the completeness of a functional system.	4. Observation The teacher walks around the classroom and check students' progress.	Worksheet

Time (mins)	**LO**	**Activity**	**Assessment**	**Material**
		Students will be instructed to brainstorm 3 applications, the functions created should be meaningful and related. Each group should utilise the SQL statements learnt in previous lessons.		
15		Each group will present their system in 3 minutes. Audience will be invited to share their thoughts on the system and provide suggestions.	5. Verbal Presentation Students will present their design and its implementation with SQL statements.	Worksheet
Conclusion (10 mins)				
10		**Unit Summary** The module will be summarised with the assistance of the SQL list on the worksheet. Students will be asked the usage of the following items to consolidate their understanding memory: 1. Define Table 2. Modify Table 3. Create Index 4. Create View 5. Insert Record 6. Select Record 7. Comparison Operators 8. Arithmetic Operators 9. Removing Duplicates 10. Specified Pattern 11. Sort Results 12. Aggregate Functions 13. Date Functions 14. Numeric Functions 15. String Functions 16. Grouping Records 17. Inner Join 18. Outer Join 19. Sub-query 20. Set Operators	6. Questioning Students will be asked about the relevant concepts taught in this module.	Worksheet

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 1

STEPS OF DRAFTING DATABASE DESIGN



DATA TYPES

Data Type	MySQL Syntax	Example Data
		2147483647
		32.8
		+852 2948 8000
	`VARCHAR(50)`	CHAN PUI YI
		TRUE
		2022-08-31
		2022-06-12 15:00:00

CONSTRAINTS

Constraint	Usage	Syntax
		…fullName VARCHAR(12) _____
		…phone CHAR(8) _____
		…SID VARCHAR(12) _____
		_____ (SID) _____ _____(SID)

CONSTRUCTING SQL STATEMENT

CREATE TABLE

```
CREATE TABLE [Table_Name] (
          [Field1_Name] [Data_Type],
          [Field2_Name] [Data_Type])
```

CREATE TABLE

CREATE TABLE

ADD COLUMN

```
ALTER TABLE     [Table_Name]
ADD COLUMN      [New_Field_Name] [New_Data_Type]
```

MODIFY COLUMN

```
ALTER TABLE     [Table_Name]
MODIFY COLUMN   [Field_Name] [New_Data_Type]
```

DROP COLUMN

```
ALTER TABLE     [Table_Name]
DROP COLUMN     [Field_Name]
```

DROP TABLE

```
DROP TABLE      [Table_Name]
```

GROUP ACTIVITY – MINI-DATABASE (1) DATABASE DESIGN

Your group will create a database for your classmates.

1. Think and discuss a topic.
2. Design a database with at least 2 tables and 5 fields.

TOPIC

Fields	Data Types	Constraints

Fields	Data Types	Constraints

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 2

INDEX FILE

USAGE

1

SELECTION CONSIDERATIONS

1

2

CONSTRUCTING SQL STATEMENT

CREATE INDEX

```
CREATE INDEX [Index_Name] ON [Table_Name] (Column_Name)
```

INSERT INTO

INSERT INTO

INSERT INTO (SPECIFIED FIELD)

INSERT INTO

INSERT INTO (MULTIPLE ROWS)

INSERT INTO

UPDATE

UPDATE _____

_____ [Field_1] **=** [New_Value], [Field_2] **=** [New_Value]

_____ _____

DELETE

DELETE _____

_____ [ ]

_____ _____

GROUP ACTIVITY – MINI-DATABASE (2) DATA COLLECTION

Your group will work the database created in the last lesson.

1. Collect data from your classmates and mark it on this worksheet.
2. Construct a SQL statement to insert data to the database.

DATA

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 3

DATABASE QUERY

SELECT

SELECT (SPECIFIED FIELDS)

OPERATORS

COMPARISON

Operator	Syntax	Example
=	`Age = 20`	Age is equal to 20.

LOGICAL

Operator	Syntax	Example

ARITHMETIC

Operator	Syntax	Example

SQLGO - ONLINE SQL GAME PLATFORM

SQLGo is an online SQL game platform built for you to learn and make progress on SQL.

OVERVIEW

Story Mode

Visit the platform after class and try your best to help SQLand!

Daily Challenge

The system provides daily challenge for you to exert yourself. Complete the daily challenge and see you on the Leaderboard!

ACTIVATE YOUR ACCOUNT

1. Visit https://sql.eduhk.tk/
2. Sign-in with your student number
3. Change your password **immediately**

Step 1	Step 2	
https://sql.eduhk.tk/	**Username**	Student email
	Password	Student number

GROUP ACTIVITY – MINI-DATABASE (3) DATA PRESENTATION

Your group will work on the database created in previous lessons.

1. Construct a SQL statement with the operators learnt to find out meaningful results.
2. Use all three types of operators in your group.
3. Present one of the constructed SQL statements and its purpose to the class.

PURPOSE

QUERY

	Comparison		Logical		Arithmetic

PURPOSE

QUERY

	Comparison		Logical		Arithmetic

PURPOSE

QUERY

	Comparison		Logical		Arithmetic

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 4

FILTERING FOR UNIQUE RESULTS

USAGE

1

SYNTAX

SORTING QUERY RESULTS

SCENARIOS

1

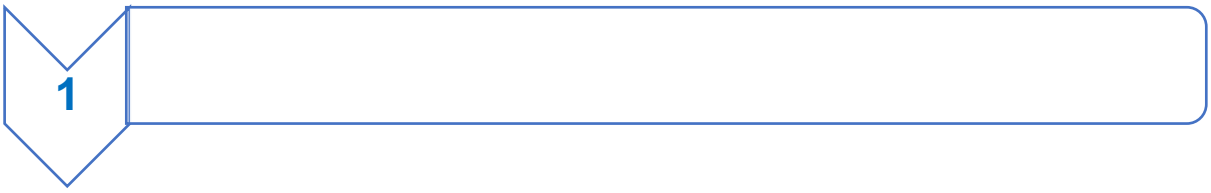
2

SYNTAX

SYNTAX

SELECTING DATA WITHIN A RANGE

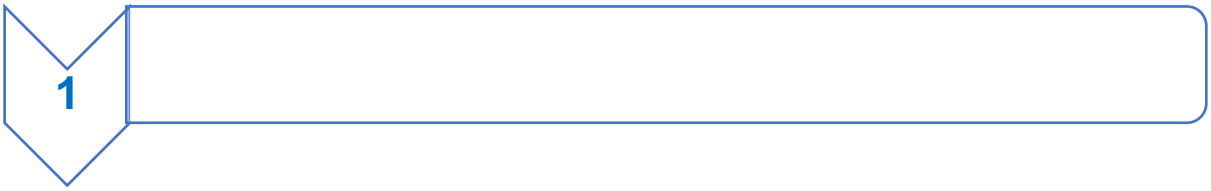
USAGE



SYNTAX

SELECTING DATA IN A SPECIFIED PATTERN

USAGE



WILDCARDS

Wildcard	Syntax	Example

QUERY

QUERY

GROUP ACTIVITY – DATA SEARCHER

Your group will complete the missions given.

1. Read SQL statements to solve the problems with the skills you have learnt.
2. Write down the query results on this worksheet as soon possible.

QUERY 1

QUERY 2

QUERY 3

QUERY 4

QUERY 5

QUERY 6

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 5

BUILT-IN FUNCTIONS

AGGREGATE FUNCTIONS

Syntax	Usage

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

DATE FUNCTIONS

Syntax	Usage

QUERY

Purpose	

QUERY

Purpose	

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 6

BUILT-IN FUNCTIONS

NUMERIC FUNCTIONS

Syntax	Usage

QUERY

Purpose	

STRING FUNCTIONS

Syntax	Usage

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

ACTIVITY

GUESS-AND-TRY

Guess the function used from the results given as quick as possible.

Database table `ICTEXAM`

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

MAX	MIN	SUM	AVG	COUNT

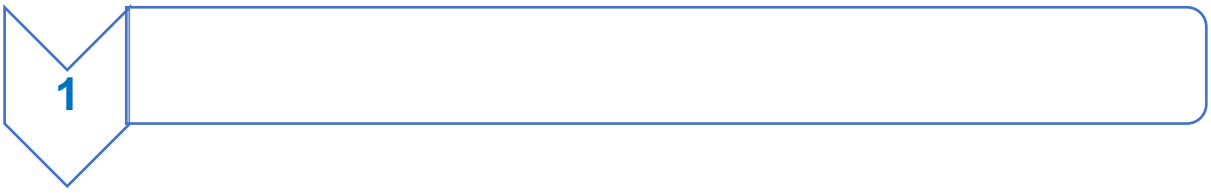
Results	Function

LEN	UPPER	LOWER	TRIM	SUBSTRING

Results	Function(s)

GROUPED INFORMATION

USAGE



SYNTAX

Database table `ICTEXAM`

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	50

QUERY

Purpose	

RESULTS

FILTERING GROUPED INFORMATION

USAGE

1

SYNTAX

Database table `ICTEXAM`

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	50

QUERY

Purpose	

RESULTS

GROUP ACTIVITY – UTILISING GROUPED INFORMATION

Your group will work the database created in lesson 1.

1. Construct a SQL statement utilising the grouped information for your database.
2. Draw a rectangle outside each group in your database.

DATABASE

QUERY

Purpose	

RESULTS

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 7

MULTI-TABLE QUERY

NEEDS

REASONS FOR STORING IN MULTIPLE TABLES

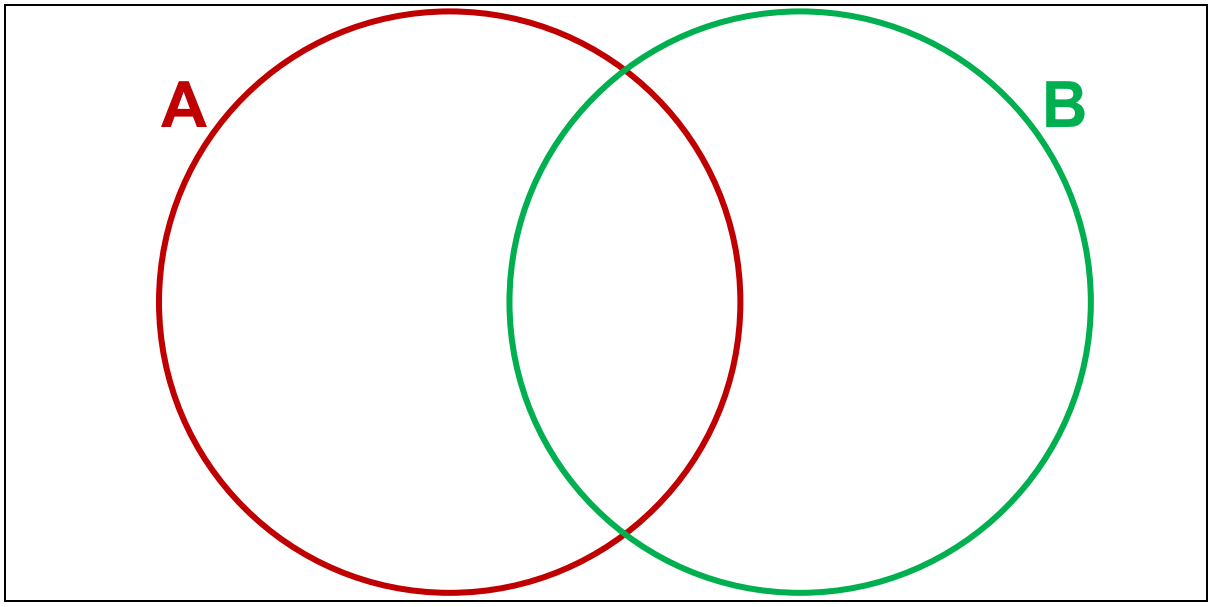
LINKAGE CREATION

EXAMPLE SCENARIOS

INNER JOIN

USAGE

DIAGRAM



TYPES

Type	Usage
1.	
2.	

QUERY 1

QUERY 2

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 8

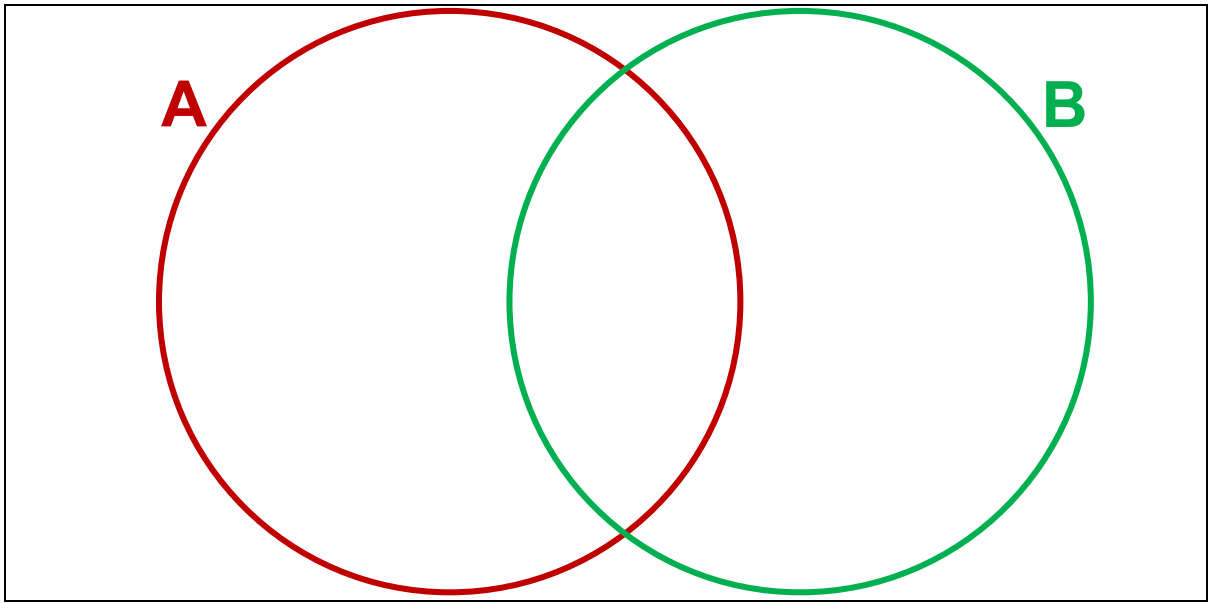
OUTER JOIN

USAGE

LEFT OUTER JOIN

USAGE

DIAGRAM

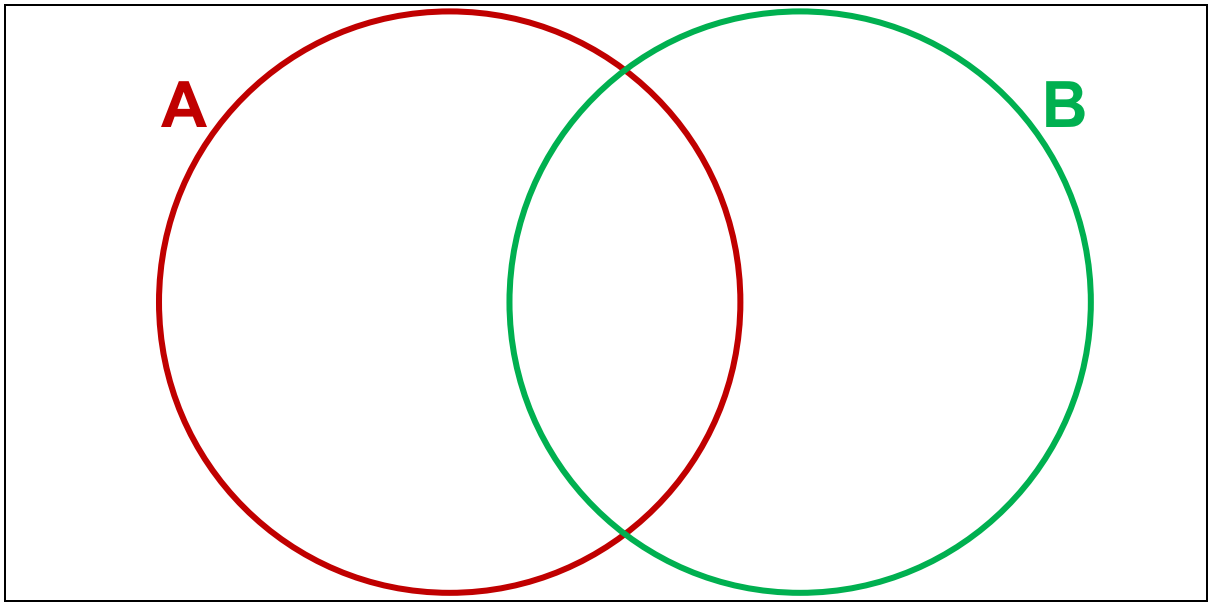


SYNTAX

RIGHT OUTER JOIN

USAGE

DIAGRAM

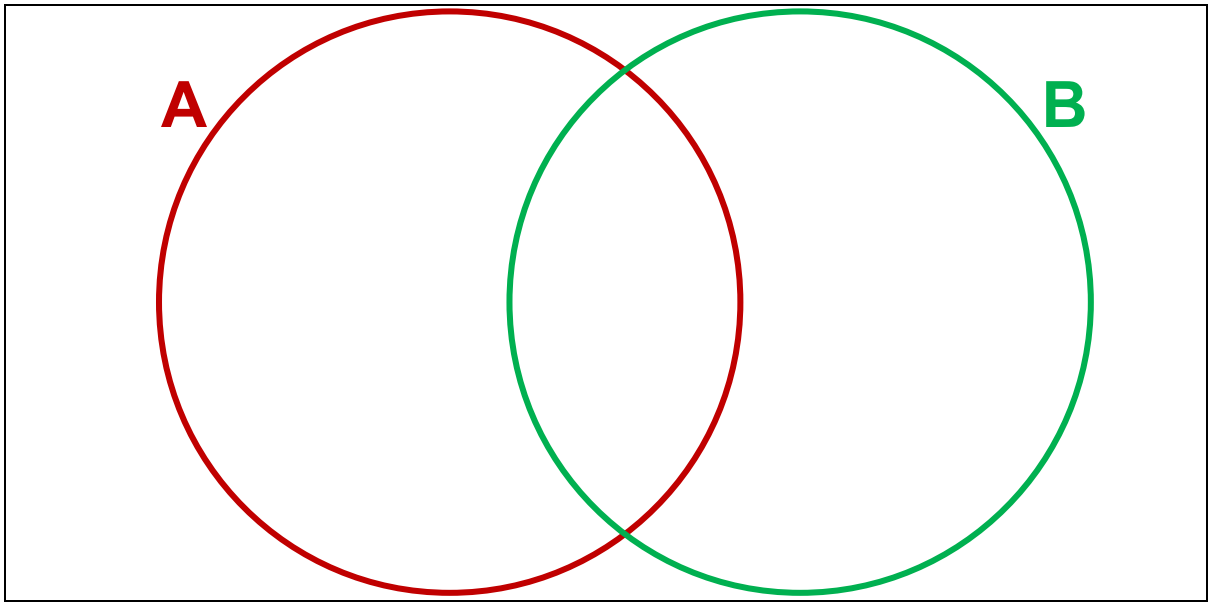


SYNTAX

FULL OUTER JOIN

USAGE

DIAGRAM

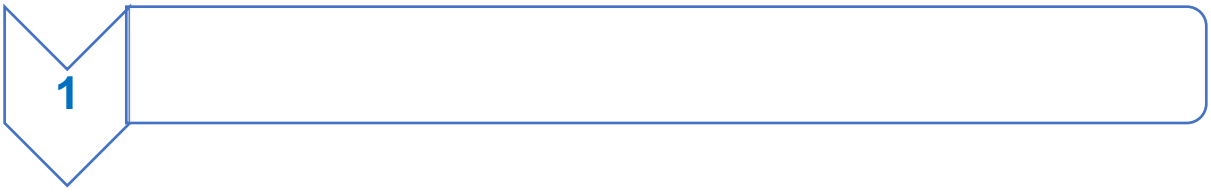


SYNTAX

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 9

SUB-QUERIES

USAGE



SUB-QUERIES OF SINGLE VALUE

QUERY

SUB-QUERIES OF MULTIPLE VALUE

OPERATORS

Operator	Usage

QUERY

QUERY

QUERY

QUERY

SET OPERATORS

USAGE

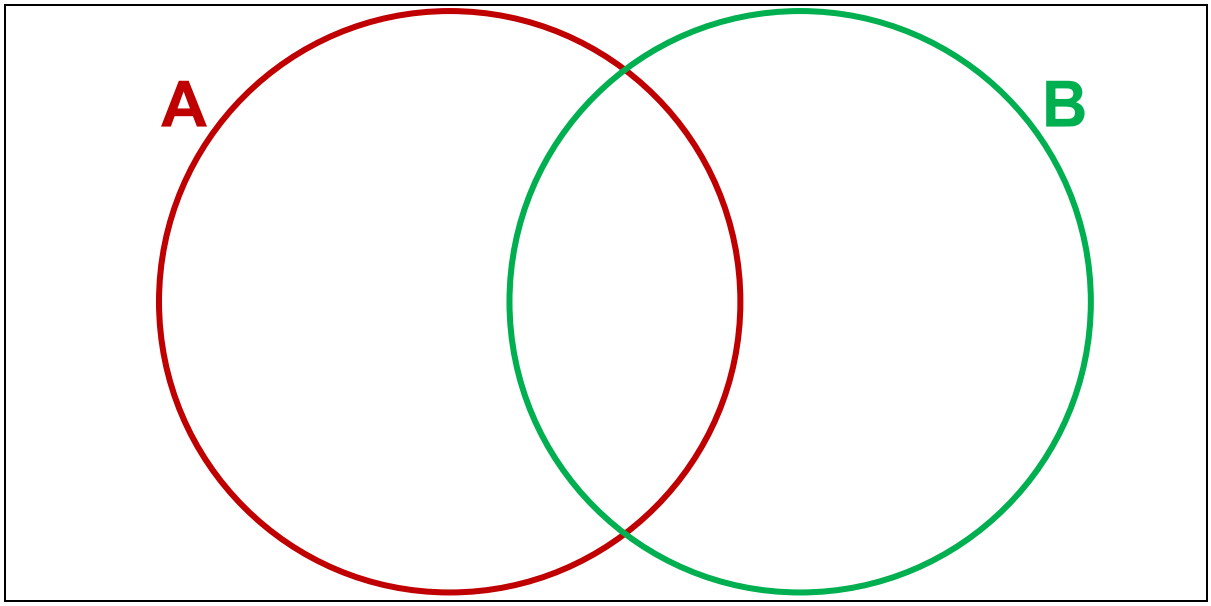
1

UNION

USAGE

1

DIAGRAM



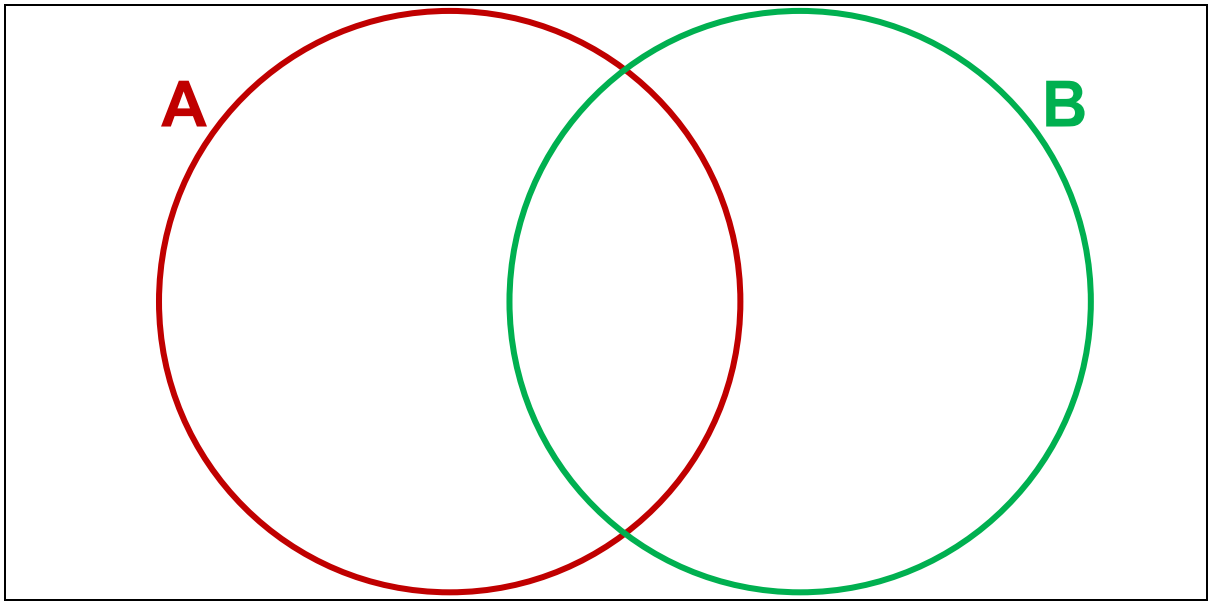
SYNTAX

INTERSECT

USAGE

1

DIAGRAM



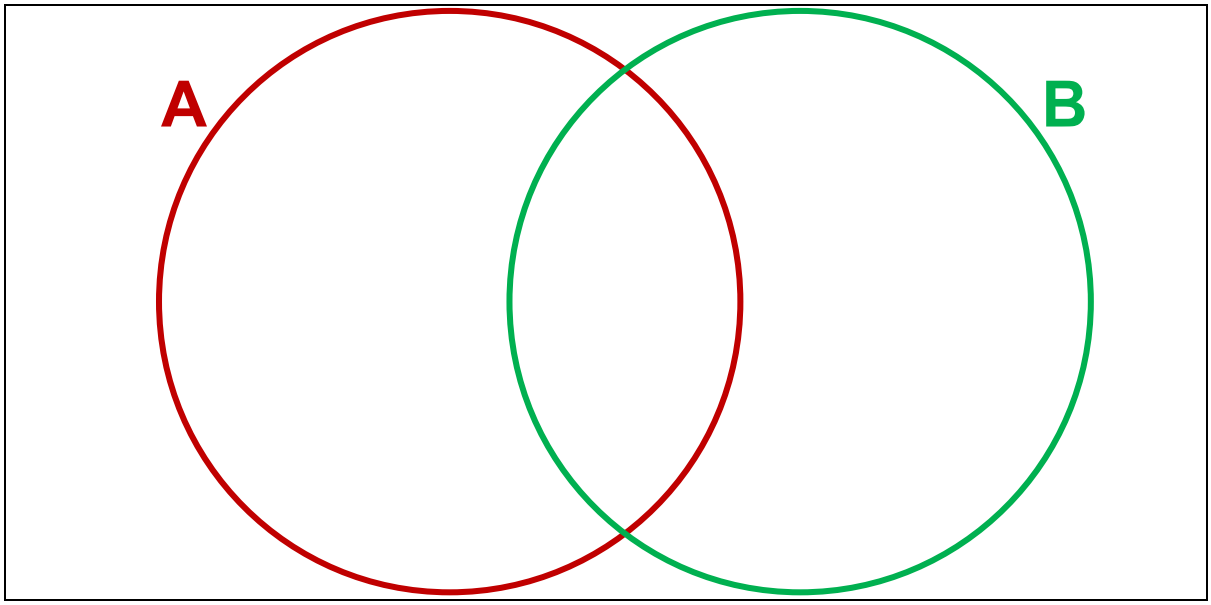
SYNTAX

MINUS

USAGE

1

DIAGRAM

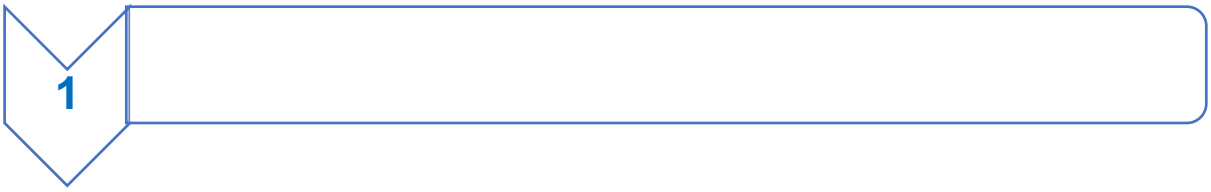


SYNTAX

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 10

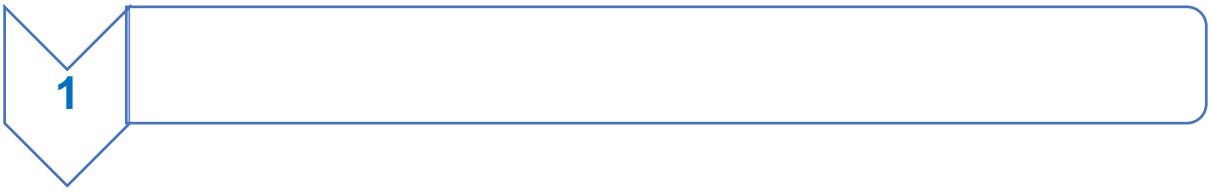
ACCESS RIGHTS

NEEDS



VIEWS

USAGE



SYNTAX

SUMMARY

Usage	SQL	
Define Table	CREATE TABLE, NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY	
Modify Table	ALTER/ DROP TABLE, ADD/ MODIFY/ DROP COLUMN	
Create Index	CREATE INDEX	
Create View	CREATE VIEW	
Insert Record	INSERT INTO, UPDATE, DELETE	
Select Record	SELECT	
Comparison Operators	>, <, =, >=, <=, <>, NOT, AND, OR, IS NULL	
Arithmetic Operators	+, -, *, /, % [mod]	
Removing Duplicates	DISTINCT	
Specified Pattern	BETWEEN…AND, LIKE, %, _	
Sort Results	ORDER BY ASC/DESC	
Aggregate Functions	MAX, MIN, SUM, AVG, COUNT	
Date Functions	DATE, DAY, MONTH, YEAR	
Numeric Functions	ABS	
String Functions	LEN, UPPER, LOWER, TRIM, SUBSTRING	
Grouping Records	GROUP BY, HAVING	
Inner Join	INNER JOIN, NATURAL JOIN	
Outer Join	LEFT [OUTER] JOIN, RIGHT [OUTER] JOIN, FULL [OUTER] JOIN	
Sub-query	IN, EXISTS, ALL, ANY	
Set Operators	UNION, INTERSECT, MINUS	

GROUP ACTIVITY – MINI-DATABASE (4) SYSTEM DESIGN

Your group will work the database created in previous lessons. Your mission is to enhance their initial design to a functional system with 3 applications.

1. Brainstorm 3 applications which form a mini-system.
2. Utilise the SQL statements you have learnt in previous lessons.
3. Present the system to the class.

SYSTEM FUNCTIONS

DATABASE SCHEMA

Function 1	**Users**	

USAGE

QUERY

Function 2	**Users**	

USAGE

QUERY

Function 3	**Users**	

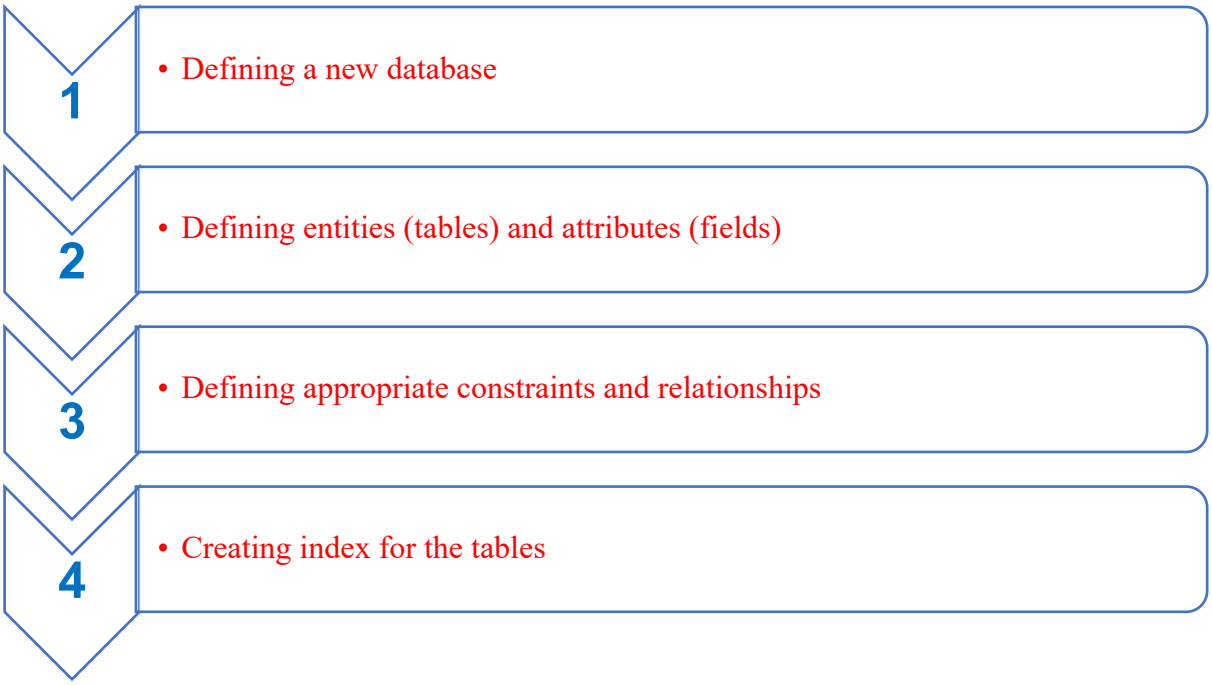
USAGE

QUERY

FEEDBACK RECEIVED

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 1

STEPS OF DRAFTING DATABASE DESIGN



DATA TYPES

Data Type	MySQL Syntax	Example Data
Integer	`INT`	2147483647
Real	`DECIMAL(3,1)`	32.8
Character	`CHAR(14)`	+852 2948 8000
Character	`VARCHAR(50)`	CHAN PUI YI
Boolean	`BOOLEAN`	TRUE
Date	`DATE`	2022-08-31
Time	`DATETIME`	2022-06-12 15:00:00

CONSTRAINTS

Constraint	Usage	Syntax
Not Null	Ensure the field is filled.	…fullName VARCHAR(12) NOT NULL
Unique	Ensure the values are unique on the field.	…phone CHAR(8) UNIQUE
Primary Key	Define the field(s) which may uniquely identify each of the records.	…SID VARCHAR(12) PRIMARY KEY
Foreign Key	Define the field(s) relates to the primary key of another table.	FOREIGN KEY (SID) REFERENCES STUDENT(SID)

CONSTRUCTING SQL STATEMENT

CREATE TABLE

```
CREATE TABLE [Table_Name] (
          [Field1_Name] [Data_Type],
          [Field2_Name] [Data_Type])
```

CREATE TABLE

CREATE TABLE

ADD COLUMN

```
ALTER TABLE     [Table_Name]
ADD COLUMN      [New_Field_Name] [New_Data_Type]
```

MODIFY COLUMN

```
ALTER TABLE     [Table_Name]
MODIFY COLUMN   [Field_Name] [New_Data_Type]
```

DROP COLUMN

```
ALTER TABLE     [Table_Name]
DROP COLUMN     [Field_Name]
```

DROP TABLE

```
DROP TABLE      [Table_Name]
```

GROUP ACTIVITY – MINI-DATABASE (1) DATABASE DESIGN

Your group will create a database for your classmates.

1. Think and discuss a topic.
2. Design a database with at least 2 tables and 5 fields.

TOPIC

Fields	Data Types	Constraints

Fields	Data Types	Constraints

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 2

INDEX FILE

USAGE

1. Allow faster searching in a database table

SELECTION CONSIDERATIONS

1. Searching frequency of the field
2. Primary key or foreign key will be created automatically

CONSTRUCTING SQL STATEMENT

CREATE INDEX

```
CREATE INDEX [Index_Name] ON [Table_Name] (Column_Name)
```

INSERT INTO

```
INSERT INTO [Table_Name]
VALUES ('Value_1', 'Value_2', 'Value_3')
```

INSERT INTO (SPECIFIED FIELD)

```
INSERT INTO [Table_Name] ('Field_1', 'Field_2')
VALUES ('Value_1', 'Value_2')
```

INSERT INTO (MULTIPLE ROWS)

```
INSERT INTO [Table_Name] ('Field_1', 'Field_2')
VALUES ('Value_A1', 'Value_A2'),
('Value_B1', 'Value_B2')
```

UPDATE

```
UPDATE  [Table_Name]
SET     [Field_1] = [New_Value], [Field_2] = [New_Value]
WHERE   [Field_X] = [Criterion]
```

DELETE

```
DELETE  FROM
[Table_Name]
WHERE   [Field_X] = [Criterion]
```

GROUP ACTIVITY – MINI-DATABASE (2) DATA COLLECTION

Your group will work the database created in the last lesson.

1. Collect data from your classmates and mark it on this worksheet.
2. Construct a SQL statement to insert data to the database.

DATA

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 3

DATABASE QUERY

SELECT

```
SELECT *
FROM [Table]
```

SELECT (SPECIFIED FIELDS)

```
SELECT [Field_1], [Field_2]
FROM [Table]
WHERE [Field_X] = [Criterion1] AND [Field_Y] = [Criterion2]
```

OPERATORS

COMPARISON

Operator	Syntax	Example
=	`Age = 20`	Age is equal to 20.
<	`Age < 18`	Age is less than 18.
>	`Age > 20`	Age is greater than 20.
>=	`Age >= 65`	Age is greater than or equal to 65.
<=	`Age <= 12`	Age is less than or equal to 12.
<>	`Age <> 11`	Age is not equal to 11.

LOGICAL

Operator	Syntax	Example
NOT	`WHERE NOT Age = 11`	Age is not equal to 11.
AND	`Age >= 18` `AND Class = '6A'`	Age is greater than 18 and the class is '6A'.
OR	`Class = '6A'` `OR Class = '6B'`	Class is either '6A' or '6B'.

ARITHMETIC

Operator	Syntax	Example
+	`Balance + Credit`	Summing up Balance and Credit.
–	`Balance - Fine`	Subtracting Fine from Balance.
*	`Price * Product`	Multiplying Price with Product.
/	`Mark / Score`	Dividing Mark by Score.
% [mod]	`StudentNo % 4`	Finding the remainder of StudentNo divding by 4.

SQLGO - ONLINE SQL GAME PLATFORM

SQLGo is an online SQL game platform built for you to learn and make progress on SQL.

OVERVIEW

Story Mode

Visit the platform after class and try your best to help SQLand!

Daily Challenge

The system provides daily challenge for you to exert yourself. Complete the daily challenge and see you on the Leaderboard!

ACTIVATE YOUR ACCOUNT

1. Visit https://sql.eduhk.tk/
2. Sign-in with your student account
3. Change your password **immediately**

<table>
<tr><th>Step 1</th><th colspan="2">Step 2</th></tr>
<tr><td rowspan="2">https://sql.eduhk.tk/</td><td>Username</td><td>Student email</td></tr>
<tr><td>Password</td><td>Student number</td></tr>
</table>

GROUP ACTIVITY – MINI-DATABASE (3) DATA PRESENTATION

Your group will work on the database created in previous lessons.

1. Construct a SQL statement with the operators learnt to find out meaningful results.
2. Use all three types of operators in your group.
3. Present one of the constructed SQL statements and its purpose to the class.

PURPOSE

QUERY

	Comparison		Logical		Arithmetic

PURPOSE

QUERY

	Comparison		Logical		Arithmetic

PURPOSE

QUERY

	Comparison		Logical		Arithmetic

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 4

FILTERING FOR UNIQUE RESULTS

USAGE

1. • Return distinct values of the specific field.

SYNTAX

```
SELECT DISTINCT   [Field_1]
FROM              [Table_Name]
WHERE             [Field_X] = [Criterion]
```

SORTING QUERY RESULTS

SCENARIOS

1. • Sorting exam results descendingly.

2. • Sorting the rank in atheletic meet ascendingly.

SYNTAX

```
SELECT     [Field_1]
FROM       [Table_Name]
ORDER BY   [Field_X] ASC
```

SYNTAX

```
SELECT     [Field_1]
FROM       [Table_Name]
ORDER BY   [Field_X] DESC
```

SELECTING DATA WITHIN A RANGE

USAGE

1

- Selecting results within two numbers where the start and end are inclusive.

SYNTAX

```
SELECT      [Field_1]
FROM        [Table_Name]
WHERE       [Field_X] BETWEEN [Num_1] AND [Num_2]
```

SELECTING DATA IN A SPECIFIED PATTERN

USAGE

1

- Selecting results simular to a pattern.
- Could be starting with, ending with, or include a word.

WILDCARDS

Wildcard	Syntax	Example
_	`Class LIKE '6_'`	Class starts with '6' followed by a character.
%	`Word LIKE '%ish%'`	Word contain 'ish'.

QUERY

QUERY

GROUP ACTIVITY – DATA SEARCHER

Your group will complete the missions given.

1. Read SQL statements to solve the problems with the skills you have learnt.
2. Write down the query results on this worksheet as soon possible.

QUERY 1

QUERY 2

QUERY 3

QUERY 4

QUERY 5

QUERY 6

The Education University of Hong Kong
INT4901 Honours Project II
Data Searcher - Database

TEACHER

Field name	Description	Example
TID	Teacher Identity Code	T01
TNAME	Teacher Name	Mr Au

STUDENT

Field name	Description	Example
SID	Student Identity Code	171194
SEX	Sex	F
SNAME	Name	Amy
CLASS	Class	6A
CNO	Class Number	1

CLUB

Field name	Description	Example
CID	Club Identity Code	A01
CNAME	Club Name	Editorial Board
TID	Teacher Identity Code of the Teacher-in-charge	032
QUOTA	Quota of the Club	17

TEACHER

TID	NAME
T01	Mr Au
T02	Mr Yeung
T03	Mr Chan
T04	Miss Cheung
T07	Miss Ho
T08	Miss Kwan
T10	Miss Kong
T11	Miss Lee
T12	Mr Leung
T13	Mr Chow
T14	Mr Mak
T15	Mr Wong

CLUB

CID	CNAME	TID	QUOTA
C01	Animation and Comics Club	T11	65
C02	Music Club	T10	65
C03	Christian Fellowship	T07	10
C04	Dancing Club	T15	50
C05	Debate Club	T14	50
C06	Dram Club	T08	35
C07	Film Club	T11	25
C08	Gaming Club	T10	40
C09	Culture Club	T10	15
C10	Photography Club	T14	10
C11	Computer Club	T14	65
C12	English Club	T08	70
C13	Mathematics Club	T07	15

STUDENT

SID	SEX	SNAME	CLASS	CNO
191134	F	Denny	4A	1
191189	F	Eva	4A	2
191174	M	Frank	4B	1
191146	M	Leo	4B	2
191185	F	Jasmine	4B	3
191123	M	Boris	4C	1
191125	F	Kristina	4C	2
181073	M	Martin	5A	1
181184	M	Matthew	5A	2
181005	M	Michael	5A	3
181043	F	Chloe	5B	1
181124	F	Charlie	5B	2
171194	F	Amy	6A	1
171076	M	Ben	6A	2
171104	F	Anson	6B	1
171124	M	Alex	6B	2

Mission Card

Query 1

```
SELECT DISTINCT TID
FROM CLUB
```

Query 2

```
SELECT CID, QUOTA
FROM CLUB
ORDER BY QUOTA DESC
```

Query 3

```
SELECT SID, SNAME
FROM STUDENT
WHERE SEX = 'M'
ORDER BY SNAME
```

Query 4

```
SELECT CID, QUOTA
FROM CLUB
WHERE QUOTA BETWEEN 15 AND 35
```

Query 5

```
SELECT *
FROM TEACHER
WHERE TNAME LIKE 'Mr%'
```

Query 6

```
SELECT CID, CNAME
FROM CLUB
WHERE NOT CNAME LIKE '%Club'
```

Answers

Query 1

```
SELECT DISTINCT TID
FROM CLUB
```

TID
T11
T10
T07
T15
T14
T08
T13
T02

Query 2

```
SELECT CID, QUOTA
FROM CLUB
ORDER BY QUOTA DESC
```

CID	QUOTA
C17	80
C12	70
C11	65
C01	65
C02	65
C05	50
C14	50
C04	50
C08	40
C06	35
C07	25
C09	15
C13	15
C03	10
C10	10

Query 3

```
SELECT SID, SNAME
FROM STUDENT
WHERE SEX = 'M'
ORDER BY SNAME
```

SID	SNAME
171124	Alex
171076	Ben
191123	Boris
191174	Frank
191146	Leo
181073	Martin
181184	Matthew
181005	Michael

Query 4

```
SELECT CID, QUOTA
FROM CLUB
WHERE QUOTA BETWEEN 15 AND 35
```

CID	QUOTA
C06	35
C07	25
C09	15
C13	15

Query 5

```
SELECT * FROM TEACHER
WHERE TNAME LIKE 'Mr%'
```

TID	TNAME
T01	Mr Au
T02	Mr Yeung
T03	Mr Chan
T12	Mr Leung
T13	Mr Chow
T14	Mr Mak
T15	Mr Wong

Query 6

```
SELECT CID, CNAME
FROM CLUB
WHERE NOT CNAME LIKE '%Club'
```

CID	CNAME
C03	Christian Fellowship
C14	Campus TV
C17	Choir

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 5

BUILT-IN FUNCTIONS

AGGREGATE FUNCTIONS

Syntax	Usage
`MAX(Score)`	Find the highest Score.
`MIN(Score)`	Find the lowest Score.
`AVG(Score)`	Find the average Score.
`SUM(Price)`	Find the total Price.
`COUNT(*)`	Find the number of records.

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

DATE FUNCTIONS

Syntax	Usage
`DATE`	Get the date from a timestamp
`YEAR`	Get the year from a date field
`MONTH`	Get the month from a date field
`DAY`	Get the day of month from a date field

QUERY

Purpose	

QUERY

Purpose	

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 6

BUILT-IN FUNCTIONS

NUMERIC FUNCTIONS

Syntax	Usage
ABS	Get the absolute (positive) value

QUERY

Purpose	

STRING FUNCTIONS

Syntax	Usage
`LEN`	Get the number of characters
`UPPER`	Coverts into CAPITAL letters
`LOWER`	Coverts into small letters
`TRIM`	Remove extra spaces
`SUBSTRING(Name, k, n)`	Get the n characters in Name from k-th character

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

QUERY

Purpose	

ACTIVITY

GUESS-AND-TRY

Guess the function used from the results given as quick as possible.

Database table `ICTEXAM`

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

MAX	MIN	SUM	AVG	COUNT

Results	Function
100	MAX
80	AVG
480	SUM
6	COUNT

LEN	UPPER	LOWER	TRIM	SUBSTRING

Results	Function(s)
10	LEN
HO	UPPER, SUBSTRING
lee hiu laam	LOWER

GROUPED INFORMATION

USAGE

1

- Listing the summarised results for each group one by one.

SYNTAX

```
SELECT      [Field_1], [Function]
FROM        [Table_Name]
GROUP BY    [Field_X]
```

Database table `ICTEXAM`

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	50

QUERY

Purpose	

RESULTS

FILTERING GROUPED INFORMATION

USAGE

1

- Filtering the results **after** summarising the data with GROUP BY.

SYNTAX

```
SELECT      [Field_1], [Function]
FROM        [Table_Name]
GROUP BY    [Field_X]
HAVING      [Condition]
```

Database table `ICTEXAM`

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	50

QUERY

Purpose	

RESULTS

GROUP ACTIVITY – UTILISING GROUPED INFORMATION

Your group will work the database created in lesson 1.

1. Construct a SQL statement utilising the grouped information for your database.
2. Draw a rectangle outside each group in your database.

DATABASE

QUERY

Purpose	

RESULTS

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 7

MULTI-TABLE QUERY

NEEDS

- Data are spread across different tables in a database.

REASONS FOR STORING IN MULTIPLE TABLES

- To avoid data redundancy and save storage space.
- To ensure data integrity.

LINKAGE CREATION

- Creating foreign keys for referencing another table.

EXAMPLE SCENARIOS

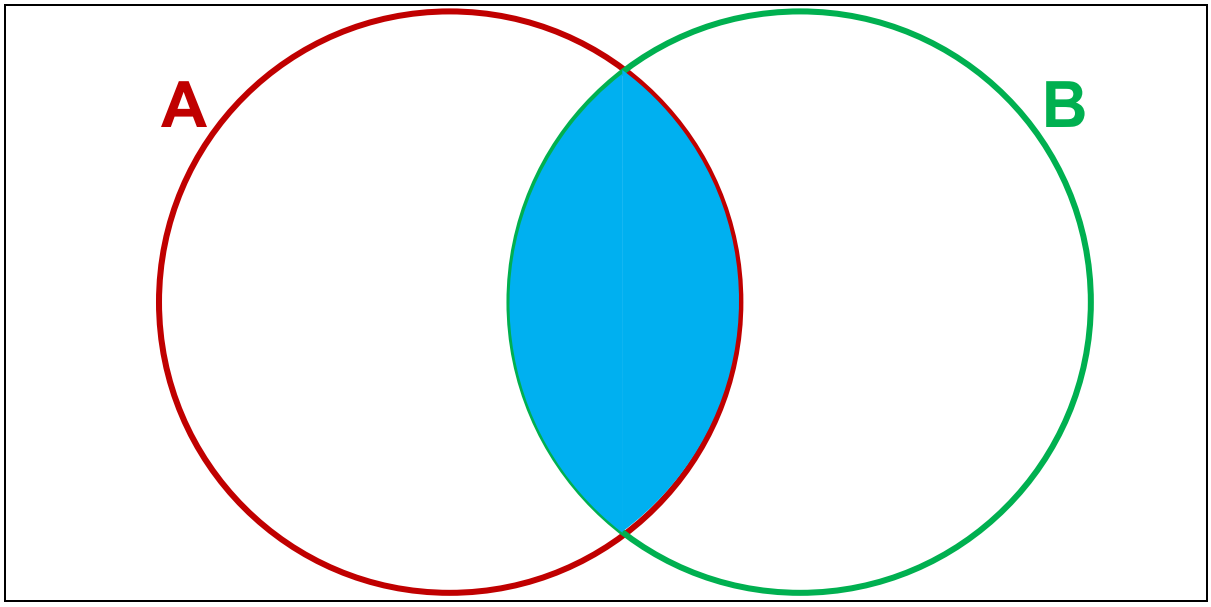
- Student information and exam results are stored in two tables.

INNER JOIN

USAGE

Joining results when the values is matched in both tables.

DIAGRAM



TYPES

Type	Usage
1. Equi-join	Filtering the joined results base on the fields with equal sign.
2. Natural Join	Matching the values of the fields with the same name and eliminating the repeated fields.

QUERY 1

QUERY 2

The Education University of Hong Kong
INT4901 Honours Project II
Accept or Reject? - Database

TEACHER

Field name	Description	Example
TID	Teacher Identity Code	T01
TNAME	Teacher Name	Mr Au

CLUB

Field name	Description	Example
CID	Club Identity Code	A01
CNAME	Club Name	Editorial Board
TID	Teacher Identity Code of the Teacher-in-charge	032
QUOTA	Quota of the Club	17

TEACHER

TID	NAME
T01	Mr Au
T02	Mr Yeung
T03	Mr Chan
T04	Miss Cheung
T07	Miss Ho
T08	Miss Kwan
T10	Miss Kong
T11	Miss Lee
T12	Mr Leung
T13	Mr Chow
T14	Mr Mak
T15	Mr Wong

CLUB

CID	CNAME	TID	QUOTA
C01	Animation and Comics Club	T11	65
C02	Music Club	T10	65
C03	Christian Fellowship	T07	10
C04	Dancing Club	T15	50
C05	Debate Club	T14	50
C06	Dram Club	T08	35
C07	Film Club	T11	25
C08	Gaming Club	T10	40
C09	Culture Club	T10	15
C10	Photography Club	T14	10
C11	Computer Club	T14	65
C12	English Club	T08	70
C13	Mathematics Club	T07	15

Mission

Query 1

```
SELECT     T.TID, T.TNAME
FROM       TEACHER T, CLUB C
WHERE      T.TID = C.TID
```

TID	TNAME
T11	Miss Lee
T10	Miss Kong
T07	Miss Ho
T15	Mr Wong
T14	Mr Mak
T08	Miss Kwan
T11	Miss Lee
T10	Miss Kong
T10	Miss Kong
T14	Mr Mak
T14	Mr Mak
T08	Miss Kwan
T07	Miss Ho
T13	Mr Chow
T02	Mr Yeung

Query 2

```
SELECT     T.TID, T.TNAME
FROM       TEACHER T, CLUB C
WHERE      T.TID = C.TID AND QUOTA > 50
```

TID	TNAME
T11	Miss Lee
T10	Miss Kong
T14	Mr Mak
T08	Miss Kwan
T02	Mr Yeung

The Education University of Hong Kong
INT4901 Honours Project II
The Librarian - Database

STUDENT

Field name	Description	Example
SID	Student Identity Code	171194
SEX	Sex	F
SNAME	Name	Amy
CLASS	Class	6A
CNO	Class Number	1

BOOK

Field name	Description	Example
BID	Book Identity Code	EB002021
BTITLE	Book Title	Survival Guide

RECORD

Field name	Description	Example
RID	Borrow Record Identity Code	2023000012
SID	Student Identity Code	171194
BID	Book Identity Code	CA001952
BORROW	Date of Borrow	2023-01-14
DUE	Due Date	2023-01-28
RETURN	Actual Date of Return	NULL
FINE	Amount of Punishment for late return	15.0

Mission 1

Find the name of the students who have borrowed books. The list should be sorted by the class and class number in ascending order.

Mission 2

Find the book titles of the books which have yet to be returned. The list should be sorted by the due date in descending order.

Mission 3

List the book titles and the number of borrowings.

Mission 4

List the names of the students and the corresponding fine for those who have owned more than $10 in 2023.

Answer

Mission 1

Find the name of the students who have borrowed books. The list should be sorted by the class and class number in ascending order.

```
SELECT     S.SNAME
FROM       STUDENT S, RECORD R
WHERE      S.SID = R.SID
ORDER BY   S.CLASS, S.CNO
```

Mission 2

Find the book titles of the books which have yet to be returned. The list should be sorted by the due date in descending order.

```
SELECT     B.BTITLE
FROM       BOOK B, RECORD R
WHERE      B.BID = R.BID AND
           RETURN IS NULL
ORDER BY   DUE DESC
```

Mission 3

List the book titles and the number of borrowings.

```
SELECT     B.BTITLE, COUNT(*)
FROM       BOOK B, RECORD R
WHERE      B.BID = R.BID
GROUP BY   B.BID
```

Mission 4

List the names of the students and the corresponding fine for those who have owned more than $10 in 2023.

```
SELECT     S.SNAME, SUM(R.FINE)
FROM       STUDENT S, RECORD R
WHERE      S.SID = R.SID AND
           YEAR(DUE) = 2023
GROUP BY   R.SID
HAVING     SUM(R.FINE) > 10
```

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 8

OUTER JOIN

USAGE

Joining results when the values is matched one of the tables.

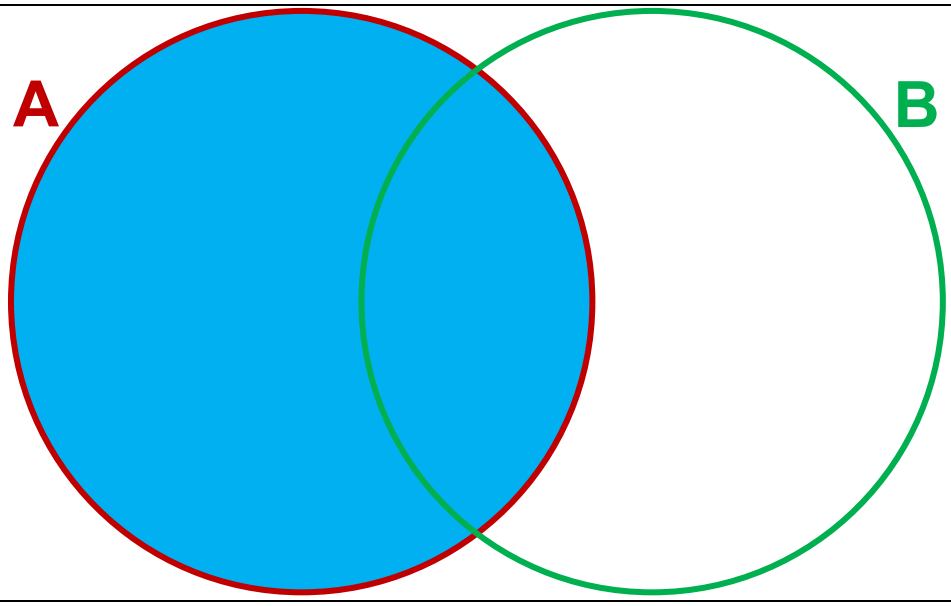
LEFT OUTER JOIN

USAGE

Selecting all records of **the left table** regardless of the existence of matching values.

NULL will be assigned if no matching value is found.

DIAGRAM



SYNTAX

```
SELECT          [Field_1], [Field_2]
FROM            [Table_1] A
LEFT JOIN       [Table_2] B
ON              A.[Field_X] = B.[Field_Y]
```

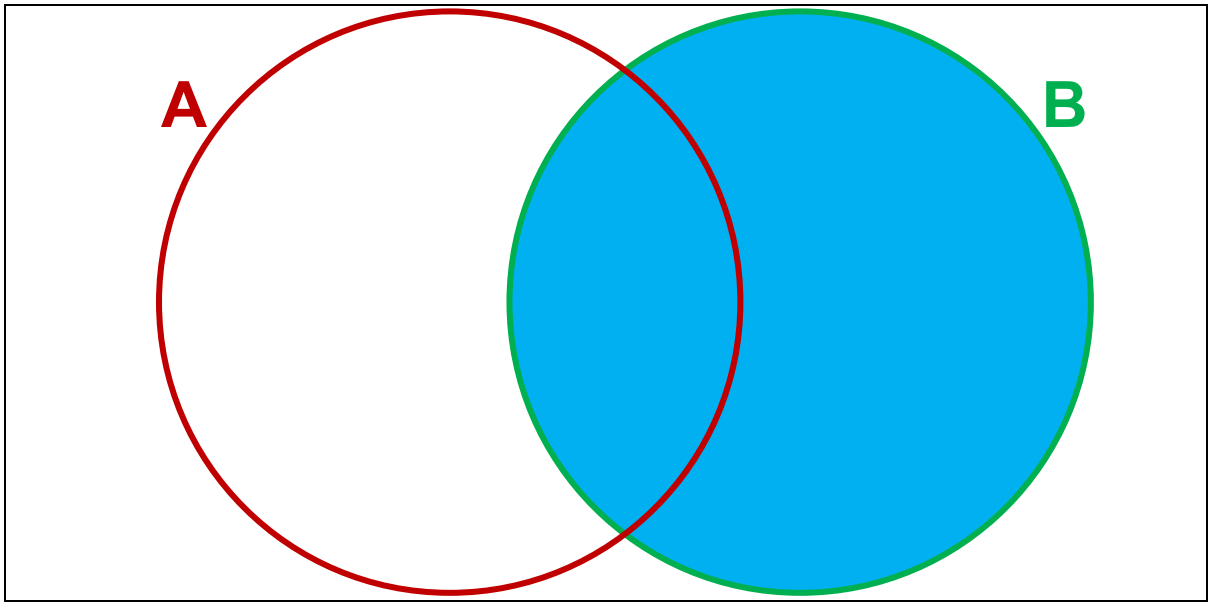
RIGHT OUTER JOIN

USAGE

Selecting all records of **the right table** regardless of the existence of matching values.

NULL will be assigned if no matching value is found.

DIAGRAM



SYNTAX

```
SELECT          [Field_1], [Field_2]
FROM            [Table_A] A
RIGHT JOIN      [Table_B] B
ON              A.[Field_X] = B.[Field_Y]
```

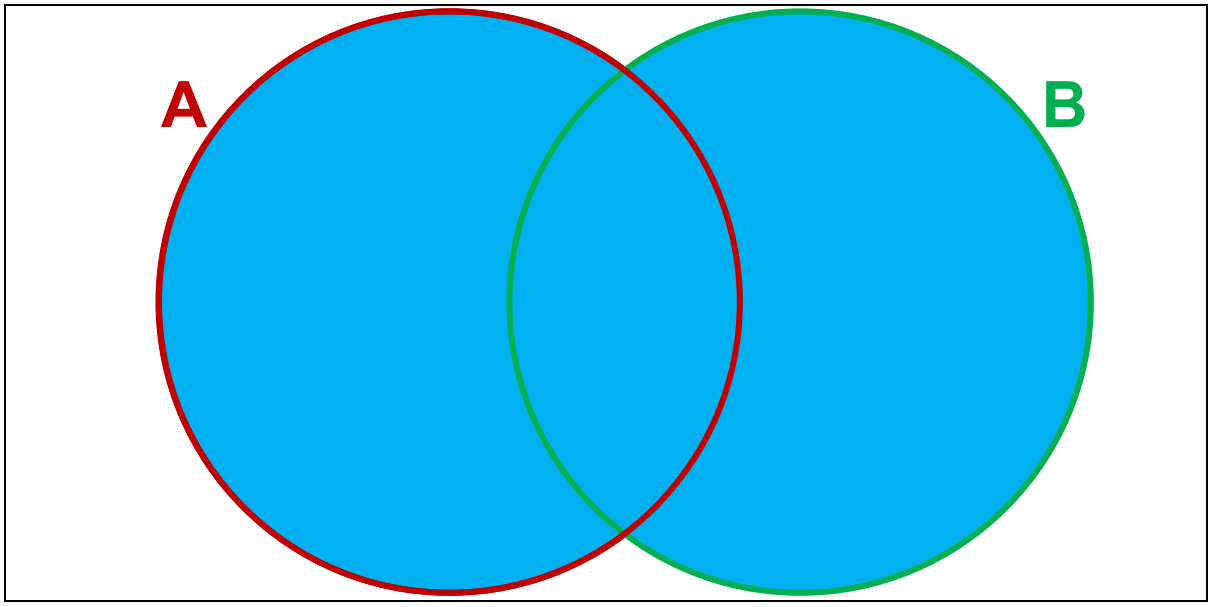
FULL OUTER JOIN

USAGE

Selecting all records **from both tables** regardless of the existence of matching values.

NULL will be assigned if no matching value is found.

DIAGRAM



SYNTAX

```
SELECT          [Field_1], [Field_2]
FROM            [Table_A] A
FULL JOIN       [Table_B] B
ON              A.[Field_X] = B.[Field_Y]
```

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 9

SUB-QUERIES

USAGE

1

- Using the results from a query to obtain other results

SUB-QUERIES OF SINGLE VALUE

QUERY

SUB-QUERIES OF MULTIPLE VALUE

OPERATORS

Operator	Usage
IN	Check if the field equals to any of the values in the sub-query.
EXISTS	Check if the sub-query returns results.
ALL	Check if the condition matches with all the sub-query results.
ANY	Check if the condition matches with any of the sub-query results.

QUERY

QUERY

QUERY

QUERY

SET OPERATORS

USAGE

1

- Joining the results of the sub-queries under same structure.

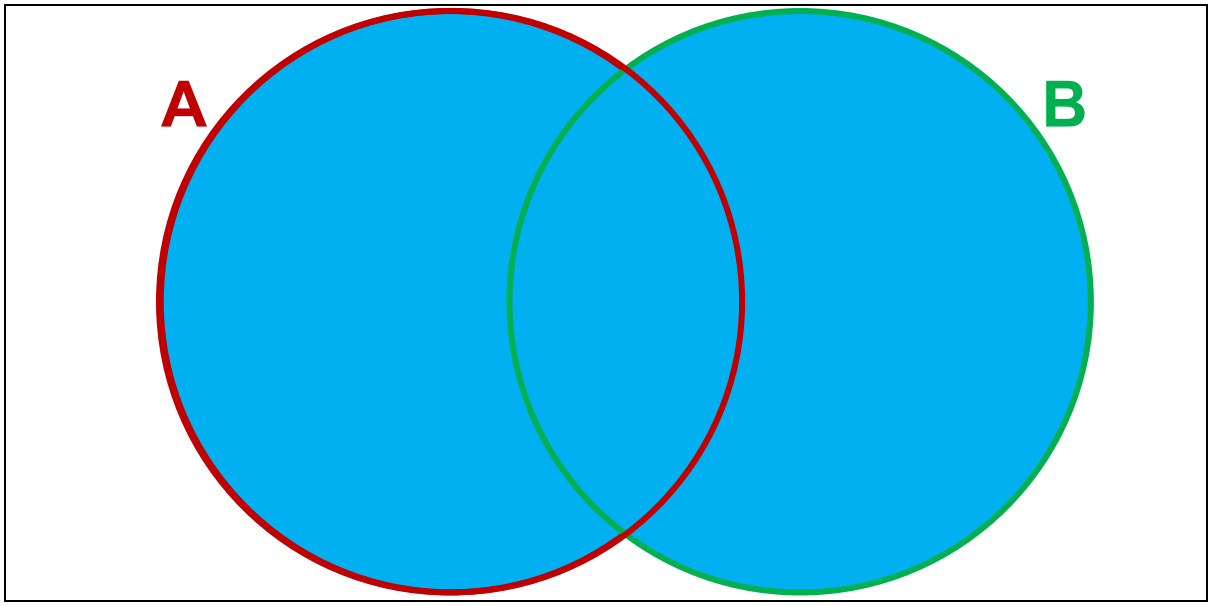
UNION

USAGE

1

- Combining two query results.

DIAGRAM



SYNTAX

```
SELECT          [Field_X], [Field_Y]
FROM            [Table_A]
UNION
SELECT          [Field_X], [Field_Y]
FROM            [Table_B]
```

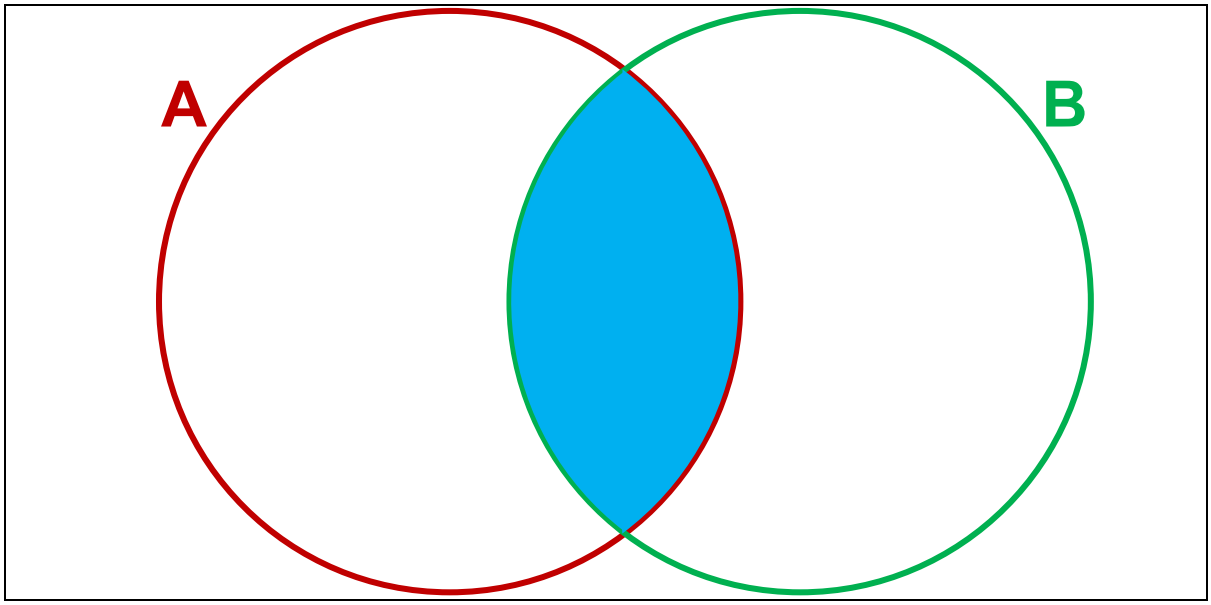
INTERSECT

USAGE

1

- Getting the records exist in **both query results**.

DIAGRAM



SYNTAX

```
SELECT          [Field_X], [Field_Y]
FROM            [Table_A]
INTERSECT
SELECT          [Field_X], [Field_Y]
FROM            [Table_B]
```

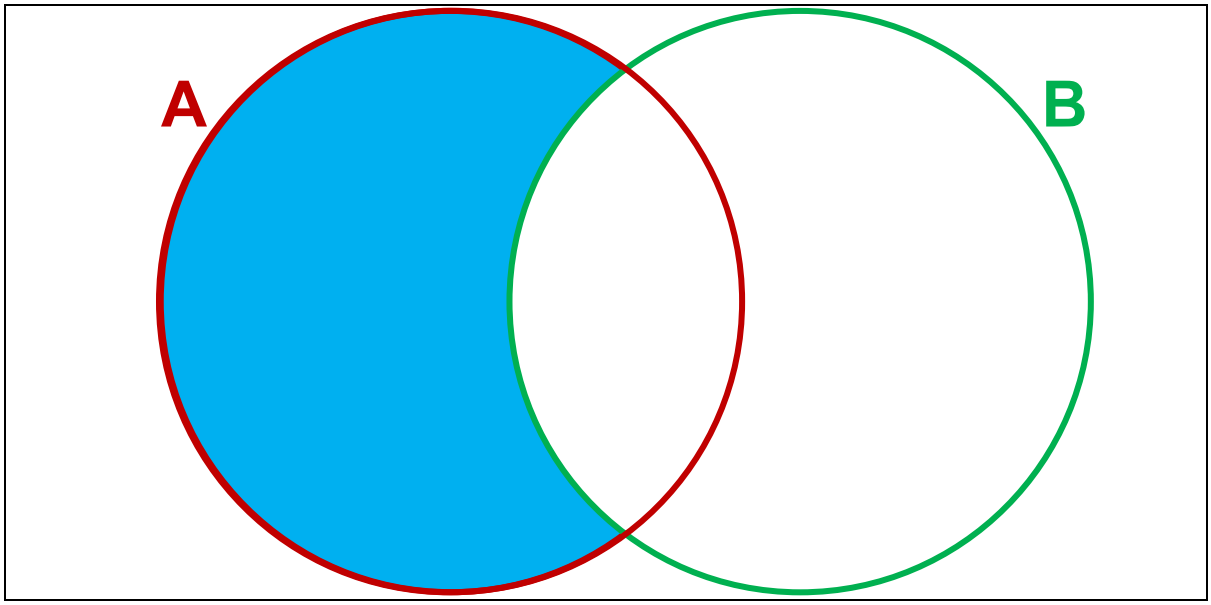
MINUS

USAGE

1

- Getting the records only exist in the **first query results**.

DIAGRAM



SYNTAX

```
SELECT          [Field_X], [Field_Y]
FROM            [Table_A]
MINUS
SELECT          [Field_X], [Field_Y]
FROM            [Table_B]
```

The Education University of Hong Kong
INT4901 Honours Project II
Worksheet 10

ACCESS RIGHTS

NEEDS

1

- A database stores data centrally, while not all data are intended for end users' access.

VIEWS

USAGE

1

- Restricting users from accessing confidential data by specifying the items could be viewed in a virtual table.

SYNTAX

```
CREATE VIEW [View_Name] AS
SELECT    …
FROM      …
WHERE     …
```

SUMMARY

Usage	SQL	
Define Table	CREATE TABLE, NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY	
Modify Table	ALTER/ DROP TABLE, ADD/ MODIFY/ DROP COLUMN	
Create Index	CREATE INDEX	
Create View	CREATE VIEW	
Insert Record	INSERT INTO, UPDATE, DELETE	
Select Record	SELECT	
Comparison Operators	>, <, =, >=, <=, <>, NOT, AND, OR, IS NULL	
Arithmetic Operators	+, -, *, /, % [mod]	
Removing Duplicates	DISTINCT	
Specified Pattern	BETWEEN…AND, LIKE, %, _	
Sort Results	ORDER BY ASC/DESC	
Aggregate Functions	MAX, MIN, SUM, AVG, COUNT	
Date Functions	DATE, DAY, MONTH, YEAR	
Numeric Functions	ABS	
String Functions	LEN, UPPER, LOWER, TRIM, SUBSTRING	
Grouping Records	GROUP BY, HAVING	
Inner Join	INNER JOIN, NATURAL JOIN	
Outer Join	LEFT [OUTER] JOIN, RIGHT [OUTER] JOIN, FULL [OUTER] JOIN	
Sub-query	IN, EXISTS, ALL, ANY	
Set Operators	UNION, INTERSECT, MINUS	

GROUP ACTIVITY – MINI-DATABASE (4) SYSTEM DESIGN

Your group will work the database created in previous lessons. Your mission is to enhance their initial design to a functional system with 3 applications.

1. Brainstorm 3 applications which form a mini-system.
2. Utilise the SQL statements you have learnt in previous lessons.
3. Present the system to the class.

SYSTEM FUNCTIONS

DATABASE SCHEMA

Function 1	**Users**	

USAGE

QUERY

Function 2	**Users**	

USAGE

QUERY

Function 3	**Users**	

USAGE

QUERY

FEEDBACK RECEIVED

Information and Communication Technology

ICT
1

Lesson 1
Database Definition

1

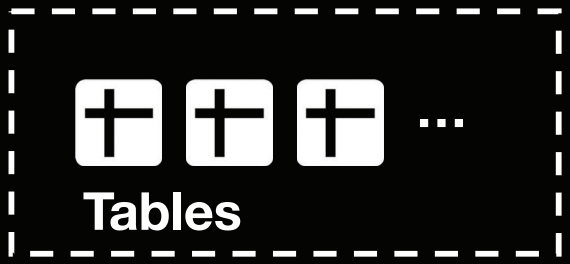
Recalling your memory…

2

Database

where we **store** the data

3



Database

4

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Table

5

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Record

6

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Field

7

Structured Query Language

instructions to access the data

8

SQL

instructions to access the data

9

TODAY

Creating database and tables
with SQL statements

Database Creation | Data Types | Constraints

10

1

Steps of Creating Database

11

Steps of Creating Database

1. Defining a new **database**
2. **Entities** (tables) and **attributes** (fields)
3. **Constraints** and **relationships**
4. **Index** for the tables

12

13

2

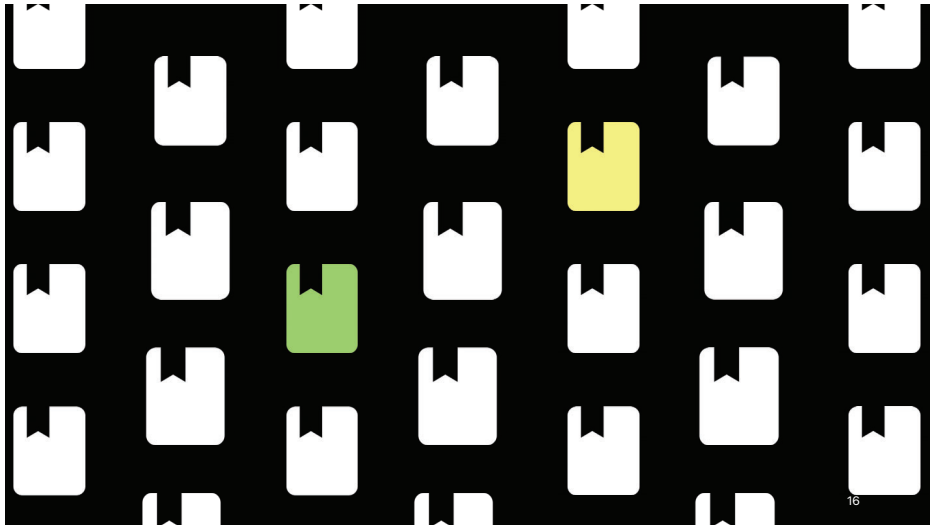
Data Types

14

Data Types

1	**Integer** INT	2147483647	5	**Date** DATE	2022-08-31
2	**Real** DECIMAL(3,1)	32.8	6	**Time** DATETIME	2022-06-12 15:00:00
3	**Character** CHAR(14)	+852 2948 8000			
4	**Boolean** BOOLEAN	TRUE			

15



16

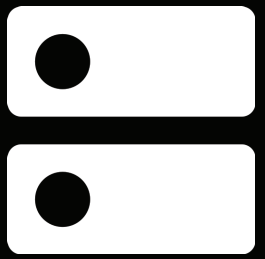
Library Borrow Record

17

Group Activity

Database Design

10 minutes



18

Database Design

Design a database for our class on daily topics
(e.g., favourite singer)

1. Get into a **group of 4**
2. Think and discuss a **topic**
3. Design the **entities** and **attributes**

19

20

3

Creating Table
with SQL Statement

21

CREATE TABLE Table

(Field Type Constraints)

22

```
CREATE TABLE Student (
    Name  CHAR(32),
    Class CHAR(2)          )
```

Name	Class
Jason	6D
Alexis	4A

Student

23

Constraints

1. **Not Null**
   Must-filled
2. **Unique**
   No repetition
3. **Primary Key**
   Identify each record
4. **Foreign Key**
   Reference another table's PK

Name*

Email

24

Constraints

1. **Not Null**
   Must-filled

   fullName VARCHAR(12)
   NOT NULL

2. **Unique**
   No repetition

   phone CHAR(8)
   UNIQUE

3. **Primary Key**
   Identify each record

   SID VARCHAR(12)
   PRIMARY KEY

4. **Foreign Key**
   Reference another table's PK

   FOREIGN KEY (SID)
   REFERENCES
   STUDENT(SID)

25

CREATE TABLE Student (

```
SID     CHAR(12) PRIMARY KEY,
Name    CHAR(12) NOT NULL,
Class   CHAR(12)                )
```

26

CREATE TABLE Borrow (

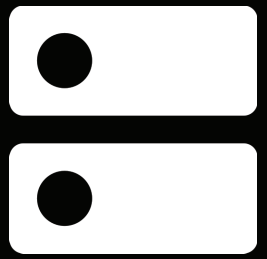
```
SID     CHAR(12) PRIMARY KEY,
BID     CHAR(12),
DATE    DATETIME NOT NULL,
FOREIGN KEY (SID)
REFERENCES
Student (SID)                   )
```

27

Group Activity

Database Design

15 minutes



28

SQLite

sqliteonline.com

29

ALTER TABLE Borrow

DROP COLUMN SID

ADD COLUMN SID CHAR(12)

MODIFY COLUMN SID INT

DROP TABLE Student

30

31

SUMMARY

Creating database and tables

with SQL statements

Database Creation | Data Types | Constraints

32

Steps of Creating Database

1. Defining a new **database**
2. **Entities** (tables) and **attributes** (fields)
3. **Constraints** and **relationships**
4. **Index** for the tables

33

Data Types

1	**Integer** INT	**2147483647**	5	**Date** DATE	**2022-08-31**
2	**Real** DECIMAL(3,1)	**32.8**	6	**Time** DATETIME	**2022-06-12 15:00:00**
3	**Character** CHAR(14)	**+852 2948 8000**			
4	**Boolean** BOOLEAN	**TRUE**			

34

Constraints

1	**Not Null** Must-filled	fullName VARCHAR(12) **NOT NULL**
2	**Unique** No repetition	phone CHAR(8) **UNIQUE**
3	**Primary Key** Identify each record	SID VARCHAR(12) **PRIMARY KEY**
4	**Foreign Key** Reference another table's PK	**FOREIGN KEY** (SID) **REFERENCES** STUDENT(SID)

35

CREATE TABLE Table

(Field Type Constraints)

36

```
CREATE TABLE Borrow (
    SID     CHAR(12) PRIMARY KEY,
    BID     CHAR(12),
    DATE    DATETIME NOT NULL,
    FOREIGN KEY (SID)
    REFERENCES
    Student (SID)                    )
```

37

```
ALTER TABLE Borrow
DROP COLUMN     SID
ADD COLUMN      SID  CHAR(12)
MODIFY COLUMN   SID  INT

DROP TABLE Student
```

38

See you in the next lesson!

39

Icons created by Google – Flaticon

40

Information and Communication Technology

ICT 2

Lesson 2
Database Definition & Insertion

1

LAST LESSON

Creating database and tables
with SQL statements

Database Creation | Data Types | Constraints

2

Steps of Creating Database

1. Defining a new **database**
2. **Entities** (tables) and **attributes** (fields)
3. **Constraints** and **relationships**
4. **Index** for the tables

3

Data Types

1	**Integer** INT	**2147483647**	5	**Date** DATE	**2022-08-31**
2	**Real** DECIMAL(3,1)	**32.8**	6	**Time** DATETIME	**2022-06-12 15:00:00**
3	**Character** CHAR(14)	**+852 2948 8000**			
4	**Boolean** BOOLEAN	**TRUE**			

4

Constraints

1. **Not Null**
   Must-filled
   `fullName VARCHAR(12) NOT NULL`
2. **Unique**
   No repetition
   `phone CHAR(8) UNIQUE`
3. **Primary Key**
   Identify each record
   `SID VARCHAR(12) PRIMARY KEY`
4. **Foreign Key**
   Reference another table's PK
   `FOREIGN KEY (SID) REFERENCES STUDENT(SID)`

5

CREATE TABLE Table

(Field Type Constraints)

6

CREATE TABLE Borrow (

```
SID     CHAR(12) PRIMARY KEY,
BID     CHAR(12),
DATE    DATETIME NOT NULL,
FOREIGN KEY (SID)
REFERENCES
Student (SID)                    )
```

7

ALTER TABLE Borrow

```
DROP COLUMN     SID
ADD COLUMN      SID  CHAR(12)
MODIFY COLUMN   SID  INT
```

DROP TABLE Student

8

9

TODAY

Manipulating data
with SQL statements

Index Definition

Data Manipulation

10

1

Defining Index

11

Index
allows **faster searching**

12

Index [Noun]

a small file that keeps the sorted values
and the actual location of the records in a table

13

SID	BID	DATE
1001	C012	2023-01-02 15:34:02
1001	C195	2023-01-02 15:34:08
1002	C075	2023-01-04 10:04:06
1003	C045	2023-01-04 13:12:24

Borrow

14

CREATE INDEX IND
ON Borrow (BID)

	SID	BID	DATE
1	1001	C012	2023-01-02 15:34:02
3	1001	C195	2023-01-02 15:34:08
2	1002	C075	2023-01-04 10:04:06

Borrow

15

Should we create index on ALL fields?

16

Index Considerations

Searching frequency

Primary Key
Foreign Key
will be auto created

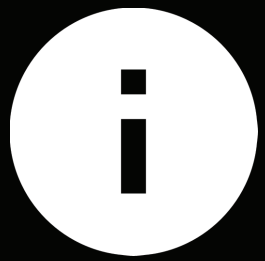
17

18

Group Activity

Data Collection

10 minutes



19

Data Collection

Collect data from your classmates

1. Get into **your group**
2. **Assign a number** for each member
3. Go to other groups and **collect data**

20

21

2

Manipulating Data

22

Insert Record

creating a new record

23

INSERT INTO Table

VALUES (Data)

24

```
INSERT INTO Borrow
VALUES
('1004', 'C186', '2023-01-15 14:37:15')
```

SID	BID	DATE
1004	C186	2023-01-15 14:37:15

Borrow

25

```
INSERT INTO   Table
              (Field)
VALUES        (Data)
```

26

```
INSERT INTO Borrow
            ('SID', 'BID')
VALUES      ('3018', 'E123')
```

SID	BID	DATE
3018	E123	NULL

Borrow

27

```
INSERT INTO Borrow ('SID', 'BID')
VALUES             ('2085', 'E099'),
                   ('2085', 'E853')
```

SID	BID	DATE
2085	E099	NULL
2085	E853	NULL

Borrow

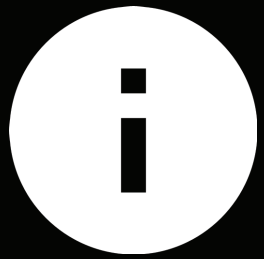
28

29

Group Activity

Data Collection

15 minutes



30

SQLite

sqliteonline.com

31

Data Collection

Collect data from your classmates

1. **Upload** your database
2. **Insert** the data collected

32

33

Update Record

updating data under condition

34

UPDATE **Table**

SET **Column** = **Data**

WHERE **Condition**

35

UPDATE Student

SET Phone = '2948 8001'
WHERE SID = '1001'

SID	Name	Class	Phone
1001	Jason	6D	2948 8081
1002	Alexis	4A	2948 0000

Student

36

UPDATE Student

SET Phone = '2948 8001'
WHERE SID = '1001'

SID	Name	Class	Phone
1001	Jason	6D	2948 8081
1002	Alexis	4A	2948 0000

Student

37

38

Delete Record

deleting record under condition

39

DELETE

FROM Table

WHERE Condition

40

DELETE

FROM Student
WHERE SID = '2047'

SID	Name	Class	Phone
~~2047~~	~~Lily~~	~~3A~~	~~2948 8081~~

Student

41

42

SUMMARY

Manipulating data
with SQL statements

Index Definition

Data Manipulation

43

Index
allows **faster searching**

44

CREATE INDEX IND
ON Borrow (BID)

Searching frequency

Primary Key
Foreign Key
will be auto created

45

INSERT INTO Table

(Field)

VALUES (Data)

46

UPDATE Table

SET Column = Data

WHERE Condition

47

DELETE

FROM Table

WHERE Condition

48

See you in the next lesson!

49

Icons created by Google – Flaticon

50

Information and Communication Technology

ICT
3

Lesson 3
Database Selection (1)

1

LAST LESSON

Manipulating data
with SQL statements

Index Definition

Data Manipulation

2

Index
allows **faster searching**

3

CREATE INDEX IND
ON Borrow (BID)

Searching frequency

Primary Key
Foreign Key
will be auto created

4

INSERT INTO Table

(Field)

VALUES (Data)

5

UPDATE Table

SET Column = Data

WHERE Condition

6

DELETE

FROM Table

WHERE Condition

7

Recalling your memory…

8

SELECT Column

FROM Table

WHERE Condition

9

10

TODAY

Selecting data
with SQL statements

Record Filtering

Operators

11

1

Selecting Data

12

How to select
ALL columns?

13

```
SELECT  *
FROM    Student
```

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Student

14

How to select
a specific column?

15

```
SELECT  Name
FROM    Student
```

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Student

16

How to select
data with condition?

17

```
SELECT   Name
FROM     Student
WHERE    Class = '6D'
```

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Student

18

SELECT Column

FROM Table

WHERE Condition

19

20

2

Operators

Comparison Logical Arithmetic

21

Operators
Comparison

=	**Equal to** Age = 20	<>	**Not equal** Age <> 11
<	**Less than** Age < 18	<=	**Less than or equal to** Age <= 12
>	**Greater than** Age < 20	>=	**Greater than or equal to** Age >= 65

22

Operators
Logical

- NOT — **Not** NOT Age = 11
- AND — **And** Age >= 18 AND Class = '6A'
- OR — **Or** Class = '6A' OR Class = '6B'

23

Operators
Arithmetic

- \+ — **Add** Balance + Credit
- – — **Subtract** Balance - Fine
- * — **Multiply** Price * Product
- / — **Divide** Mark / Score
- % — **Find Remainder [mod]** StudentNo mod 4

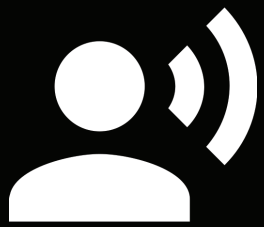
24

25

Group Activity

Data Presentation

15 minutes



26

Data Presentation

Present what you could find in the data

1. Get into **your group**
2. **Create statements**
   All 3 types of operators should be used
3. **Present** the **purpose** and **findings**

27

28

Game Exercise

SQLGo Online Platform

At Home

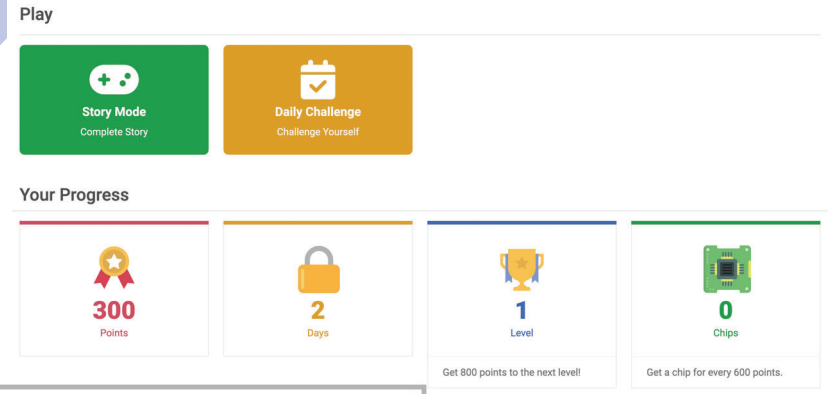
29

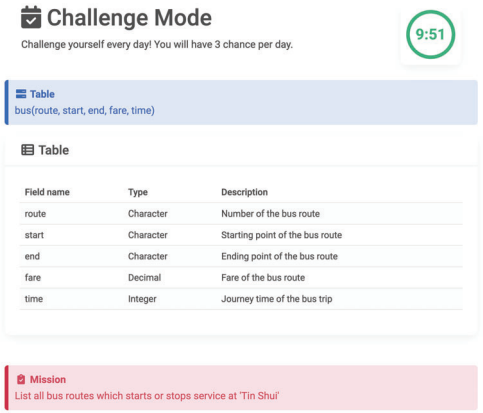
Activate SQLGo

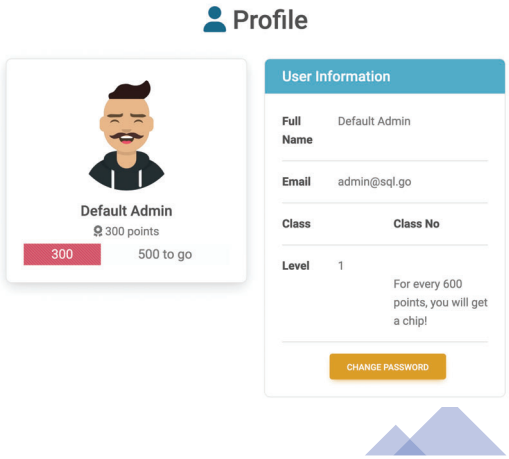
Work on SQL after school

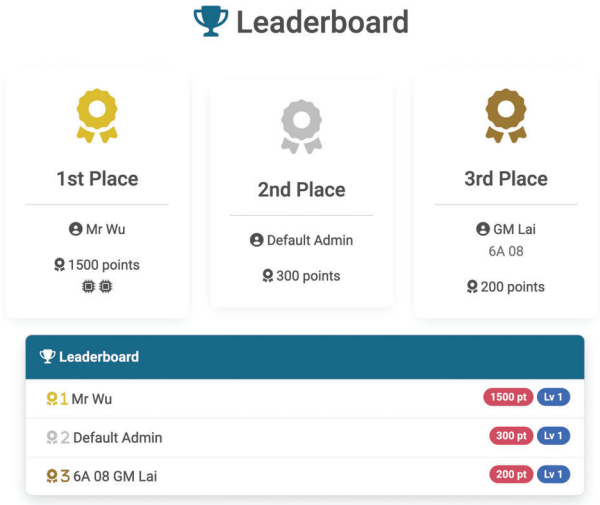
1. Sign-in with your **student account**
2. **Change password**
3. Explore the **story** and **challenge**

30









SUMMARY

Selecting data
with SQL statements

Record Filtering

Operators

SELECT Column

FROM Table

WHERE Condition

37

SELECT *
FROM Student
WHERE Class = '6D'

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Student

38

Operators

Comparison

=	**Equal to** Age = 20	<>	**Not equal** Age <> 11
<	**Less than** Age < 18	<=	**Less than or equal to** Age <= 12
>	**Greater than** Age < 20	>=	**Greater than or equal to** Age >= 65

39

Operators

Logical

NOT	**Not** NOT Age = 11
AND	**And** Age >= 18 AND Class = '6A'
OR	**Or** Class = '6A' OR Class = '6B'

40

Operators

Arithmetic

+	**Add**	Balance + Credit
–	**Subtract**	Balance - Fine
*	**Multiply**	Price * Product
/	**Divide**	Mark / Score
%	**Find Remainder [mod]**	StudentNo mod 4

41

See you in the next lesson!

42

Icons created by Google – Flaticon

43

Information and Communication Technology

ICT 4

Lesson 4
Database Selection (2)

1

LAST LESSON

Selecting data
with SQL statements

Record Filtering

Operators

2

SELECT Column

FROM Table

WHERE Condition

3

SELECT *
FROM Student
WHERE Class = '6D'

Name	Class	Date of Birth
Jason	6D	16/09/2007
Alexis	4A	05/09/2007

Student

4

Operators

Comparison

=	**Equal to** Age = 20	<>	**Not equal** Age <> 11
<	**Less than** Age < 18	<=	**Less than or equal to** Age <= 12
>	**Greater than** Age < 20	>=	**Greater than or equal to** Age >= 65

5

Operators

Logical

NOT	**Not** NOT Age = 11
AND	**And** Age >= 18 AND Class = '6A'
OR	**Or** Class = '6A' OR Class = '6B'

6

Operators

Arithmetic

+	**Add** Balance + Credit	%	**Find Remainder [mod]** StudentNo mod 4
–	**Subtract** Balance - Fine		
*	**Multiply** Price * Product		
/	**Divide** Mark / Score		

7

8

TODAY

Selecting data
with SQL statements

Unique Results | Sorting Results | Matching Pattern

9

1

Filtering Unique Results

10

Is it always that the results are unique?

11

SELECT Class
FROM Student

Class
1A
1A
1A
…

12

SELECT DISTINCT Column

FROM Table

WHERE Condition

13

SELECT DISTINCT Class
FROM Student

Class
1A
1B
...
6E

14

15

2 Sorting Results

16

Is that a problem with
unsorted results?
Share your views!

17

```
SELECT  SID, Score
FROM    Exam
```

SID	Score
1001	43
1002	84
1003	65
…	…

18

```
SELECT    Column
FROM      Table
WHERE     Condition
ORDER BY  Column  Order
```

19

```
SELECT    SID, Score
FROM      Exam
ORDER BY  Score ASC
```

SID	Score
1001	43
1003	65
1002	84

20

SELECT SID, Score
FROM Exam
ORDER BY Score DESC

SID	Score
1002	84
1003	65
1001	43

21

22

3

Within Range & Matching Pattern

23

How to limit range of the data?

24

```
SELECT   Name
FROM     Exam
WHERE    Score > 40
         AND Score < 60
```

25

```
SELECT   Name
FROM     Exam
WHERE    Score >= 40
         AND Score <= 60
```

26

```
SELECT   Name
FROM     Exam
WHERE    Score BETWEEN
         40 AND 60
```

27

How to search
all form 1 students?

28

```
SELECT   Name
FROM     Student
WHERE    Class = '1A'
         OR Class = '1B'
         OR ...
```

29

```
SELECT   Name
FROM     Student
WHERE    Class LIKE '1_'
```

30

Wildcards

_	**Single character** Must have 1 character only	'Ch_' Cha, Chu
%	**0-N characters** Allow no character	'Ch%' Ch, Cha, Chan, Ching

31

32

Game

Data Searcher

15 minutes

33

Data Searcher

Find the results as soon as possible!

1. Get into **your group**
   with the mission card and database
2. **Write down the results**
   as soon as possible
3. **Raise your hand**

100 pt

34

35

SUMMARY

Selecting data

with SQL statements

Unique Results	Sorting Results	Matching Pattern

36

SELECT DISTINCT Column

FROM Table

WHERE Condition

37

SELECT Column

FROM Table

WHERE Condition

ORDER BY Column Order

38

SELECT Name

FROM Exam

WHERE Score BETWEEN 40 AND 60

39

SELECT Name

FROM Student

WHERE Class LIKE '1_'

40

Wildcards

Single character
Must have 1 character only

'Ch_'
Ch**a**, Ch**u**

0-N characters
Allow no character

'Ch**%**'
Ch, Ch**a**, Ch**an**, Ch**ing**

41

See you in the next lesson!

42

Icons created by Google – Flaticon

43

Information and Communication Technology

ICT 5

Lesson 5
Built-in functions (1)

1

LAST LESSON

Selecting data
with SQL statements

Unique Results

Sorting Results

Matching Pattern

2

SELECT DISTINCT **Column**

FROM **Table**

WHERE **Condition**

3

SELECT **Column**

FROM **Table**

WHERE **Condition**

ORDER BY **Column** **Order**

4

```
SELECT   Name
FROM     Exam
WHERE    Score BETWEEN
         40 AND 60
```

5

```
SELECT   Name
FROM     Student
WHERE    Class LIKE '1_'
```

6

Wildcards

_ **Single character**
Must have 1 character only

'Ch_'
Ch**a**, Ch**u**

% **0-N characters**
Allow no character

'Ch**%**'
Ch, Ch**a**, Ch**an**, Ch**ing**

7

8

TODAY

Retrieving Information
with Database Built-in Functions

Aggregate Functions

Date Functions

9

Recalling your memory…

10

What are functions in spreadsheets?

11

1

Aggregate Functions

MAX
MIN

AVG

SUM

COUNT

12

```
SELECT  Function( Column)
FROM    Table
```

13

```
SELECT  MAX(Score),
        MIN(Score)
FROM    Exam
```

MAX(Score)	MIN(Score)
84	28

14

```
SELECT  AVG(Score)
FROM    Exam
```

AVG(Score)
65.42

15

```
SELECT  SUM(Price)
FROM    Product
```

SUM(Price)
30956

16

```
SELECT  COUNT(*)
FROM    Product
```

COUNT(*)
214

17

Aggregate Functions

1. **MAX** Highest value
2. **MIN** Lowest value
3. **AVG** Average value
4. **SUM** Total the values added
5. **COUNT** Number of records

18

19

Activity

Utilising Aggregate Functions

15 minutes

20

Utilising Aggregate Functions

Construct SQL statements using the functions

1. Get into **your group**
2. **Constructs statements**
3. **Write down** the **purpose**

21

22

2

Date Functions

DATE DAY MONTH YEAR

23

SELECT Function(Column)

FROM Table

WHERE Function(Column)

Condition

24

Date Functions

1. **DATE**
   2023-01-15
2. **YEAR**
   2023
3. **MONTH**
   1
4. **DAY**
   15

25

```
SELECT YEAR('2023-01-15'),
       MONTH('2023-01-15'),
       DAY('2023-01-15')
```

YEAR(…)	MONTH(…)	DAY(…)
2023	1	15

26

27

Group Activity

Utilising Date Functions

10 minutes

28

Utilising Date Functions

Construct SQL statements using the functions

1. Get into **your group**
2. **Constructs statements**
3. **Write down** the **purpose**

29

30

SUMMARY

Retrieving Information

with Database Built-in Functions

Aggregate Functions

Date Functions

31

Aggregate Functions

1. **MAX** Highest value
2. **MIN** Lowest value
3. **AVG** Average value
4. **SUM** Total the values added
5. **COUNT** Number of records

32

Date Functions

1. **DATE**
   2023-01-15
2. **YEAR**
   2023
3. **MONTH**
   1
4. **DAY**
   15

33

SELECT Function(Column)

FROM Table

WHERE Function(Column)

Condition

34

See you in the next lesson!

35

Icons created by Google – Flaticon

36

Information and Communication Technology

ICT 6

Lesson 6
Built-in functions (2)

1

LAST LESSON

Retrieving Information
with Database Built-in Functions

Aggregate Functions

Date Functions

2

Aggregate Functions

1. **MAX** Highest value
2. **MIN** Lowest value
3. **AVG** Average value
4. **SUM** Total the values added
5. **COUNT** Number of records

3

Date Functions

1. **DATE** 2023-01-15
2. **YEAR** 2023
3. **MONTH** 1
4. **DAY** 15

4

SELECT Function(Column)

FROM Table

WHERE Function(Column)

Condition

5

6

TODAY

Retrieving Information
with Database Built-in Functions

Numeric Function | String Functions | Grouped Information

7

1

Numeric Function

ABS

8

Numeric Function

1 **ABS**
Positive value
(ignore the negative sign)

9

SELECT ABS(4.99), ABS(-4.99)

ABS(4.99)	ABS(-4.99)
4.99	4.99

10

11

2 String Functions

LEN UPPER LOWER TRIM SUBSTRING

12

String Functions

1. **LEN**
   Number of characters
2. **UPPER**
   Coverts to CAPITAL letters
3. **LOWER**
   Coverts to small letters
4. **TRIM**
   Remove extra spaces
5. **SUBSTRING**
   Get the n characters
   in the field
   from the k-th character

13

SELECT LEN(Name), UPPER(Name), LOWER(Name)

LEN(Name)	UPPER(Name)	LOWER(Name)
12	CHAN LOK YAN	chan lok yan

14

SELECT SUBSTRING (Name, 1, 4)

Start Length

SUBSTRING(…)
Chan

15

SELECT TRIM(‘ I love ICT ’)

TRIM(…)
I love ICT

16

17

Game

Guess-and-try

15 minutes

18

Guess-and-try

Guess the function used in the output

1. **Read** the data
2. **Results** show on screen
3. **Guess** the function(s) used as soon as possible

100 pt

19

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

?

MAX MIN SUM AVG COUNT

20

Ready?
Round 1

21

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

MAX | MIN | SUM | AVG | COUNT

22

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

100

MAX | MIN | SUM | AVG | COUNT

23

MAX

24

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

MAX MIN SUM AVG COUNT

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

80

MAX MIN SUM AVG COUNT

AVG

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

MAX MIN SUM AVG COUNT

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

480

MAX MIN SUM AVG COUNT

29

SUM

30

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

MAX MIN SUM AVG COUNT

31

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

6

MAX MIN SUM AVG COUNT

32

COUNT

33

34

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

?

LEN UPPER LOWER TRIM SUBSTRING

Ready?
Round 2

36

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

LEN | UPPER | LOWER | TRIM | SUBSTRING

37

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

10

LEN | UPPER | LOWER | TRIM | SUBSTRING

38

LEN

39

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

LEN | UPPER | LOWER | TRIM | SUBSTRING

40

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

HO

LEN UPPER LOWER TRIM SUBSTRING

TRIM SUBSTRING

42

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

LEN UPPER LOWER TRIM SUBSTRING

ICTEXAM

CLASS	NAME	MARKS
5A	Chan Lok Yan	70
5A	Cheng Tsz Ki	80
5A	Chung Yuek Sum	90
5B	Ho Man Tik	60
5B	Lee Hiu Laam	80
5C	Lee Ka Yeung	100

Output

lee hiu laam

LEN UPPER LOWER TRIM SUBSTRING

LOWER

45

46

3

Grouped Information

47

How to get
the highest score
for each class?

48

Recalling your memory…

49

```
SELECT  MAX(Score),
        MIN(Score)
FROM    Exam
```

MAX(Score)	MIN(Score)
84	28

50

```
SELECT  SID, Class, Score
FROM    Exam
```

SID	Class	Score
1001	1A	43
1002	1B	84
1003	1A	65
1004	1B	28

51

```
SELECT    Column
FROM      Table
GROUP BY  Column
```

52

```
SELECT    Class, MAX(Score)
FROM      Exam
GROUP BY  Class
```

Class	Score
1A	43
1B	84
1A	65
1B	28

53

```
SELECT    Class, MAX(Score)
FROM      Exam
GROUP BY  Class
```

Class	Score
1A	43
1B	84
1A	65
1B	28

54

```
SELECT    Class, MAX(Score)
FROM      Exam
GROUP BY  Class
```

1A	43
1A	65

1B	84
1B	28

55

```
SELECT    Class, MAX(Score)
FROM      Exam
GROUP BY  Class
```

Class	MAX(Score)
1A	65
1B	84

56

57

How to get the classes having >= 55 marks in average?

58

```
SELECT   Class, AVG(Score)
FROM     Exam
GROUP BY Class
```

Class	AVG(Score)
1A	54
1B	56
1C	58

59

```
SELECT   Class, AVG(Score)
FROM     Exam
WHERE    Score >= 55
GROUP BY Class
```

60

```
SELECT    Class, AVG(Score)
FROM      Exam
① WHERE   Score >= 55
② GROUP BY Class
```

Class	Score
~~1A~~	~~43~~
1B	84
1A	65
~~1B~~	~~28~~

61

SELECT	Column
FROM	Table
GROUP BY	Column
HAVING	Condition

62

```
SELECT    Class, AVG(Score)
FROM      Exam
GROUP BY  Class
HAVING    AVG(Score) >= 55
```

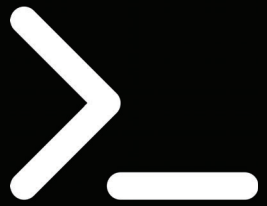
Class	AVG(Score)
~~1A~~	~~54~~
1B	56
1C	58

63

64

Group Activity

Utilising Grouped Information



10 minutes

65

Utilising Grouped Information

Construct SQL statements using groups

1. Get into **your group**
2. **Constructs statements**
3. **Write down** the **purpose**

66

67

SUMMARY

Retrieving Information

with Database Built-in Functions

Numeric Function	String Functions	Grouped Information

68

Numeric Function

1. **ABS**
   Positive value
   (ignore the negative sign)

69

String Functions

1. **LEN**
   Number of characters
2. **UPPER**
   Coverts to CAPITAL letters
3. **LOWER**
   Coverts to small letters
4. **TRIM**
   Remove extra spaces
5. **SUBSTRING**
   Get the n characters
   in the field
   from the k-th character

70

```
SELECT    Column
FROM      Table
GROUP BY  Column
HAVING    Condition
```

71

```
SELECT    Class, AVG(Score)
FROM      Exam
GROUP BY  Class
```

Class	AVG(Score)
1A	54
1B	56
1C	58

72

```
SELECT    Class, AVG(Score)
FROM      Exam
GROUP BY  Class
HAVING    AVG(Score) >= 55
```

Class	AVG(Score)
~~1A~~	~~54~~
1B	56
1C	58

73

See you in the next lesson!

74

Icons created by Google – Flaticon

75

Information and Communication Technology

ICT
7

Lesson 7
Multi-table Query (1)

1

LAST LESSON

Retrieving Information
with Database Built-in Functions

Numeric Function | String Functions | Grouped Information

2

Numeric Function

1 **ABS**
Positive value
(ignore the negative sign)

3

String Functions

1 **LEN**
Number of characters

2 **UPPER**
Coverts to CAPITAL letters

3 **LOWER**
Coverts to small letters

4 **TRIM**
Remove extra spaces

5 **SUBSTRING**
Get the n characters
in the field
from the k-th character

4

SELECT Column

FROM Table

GROUP BY Column

HAVING Condition

5

SELECT Class, AVG(Score)
FROM Exam
GROUP BY Class

Class	AVG(Score)
1A	54
1B	56
1C	58

6

SELECT Class, AVG(Score)
FROM Exam
GROUP BY Class
HAVING AVG(Score) >= 55

Class	AVG(Score)
~~1A~~	~~54~~
1B	56
1C	58

7

8

TODAY

Retrieving Information from Multiple Tables

Types of Join Equi-join Natural Join

9

Is it always the data stored in a single table?

10

Why storing data in multiple tables?

11

Why storing data in multiple tables?

Avoid data redundancy
Save storage

Ensure data integrity

12

1

Types of Join

Inner Join Outer Join

13

Joins

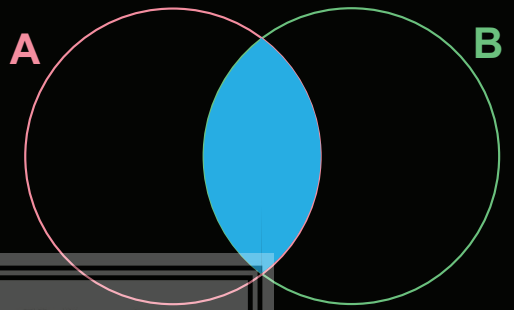
1 **Inner Join**
Joining results when the values is matched in both tables

2 **Outer Join**
Joining results even if there is no match values

14

Inner Join

Joining results when the values is matched in both tables



15

Inner Join

Joining results when the values is matched in both tables

1 **Equi-join**
- Filtering the results by fields with equal sign

2 **Natural Join**
- Matching the values of the fields with the same name
- Eliminating the repeated fields

16

2

Equi-join

17

=

Equi-join

filtering the joined results with =

18

```
SELECT   S.Name, E.Score
FROM     Student S, Exam E
WHERE    S.SID = E.SID
```

SID	Name
1001	Jason
1002	Alexis

Student

SID	Score
1001	43
1002	84

Exam

19


```
SELECT   S.Name, E.Score
FROM     Student S, Exam E
WHERE    S.SID = E.SID
```

S.Name	E.Score
Jason	43
Alexis	84

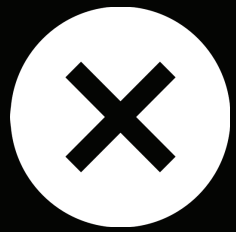
20

21

Game

Accept or Reject?

5 minutes



22

Accept or Reject?

Decide if the record should be in the results

1. **Read** the SQL statement
2. **Record** will be shown
3. **Say Accept or Reject**

whole class will be as a team

100 pt

23

Query

```
SELECT   ?
FROM     A, B
WHERE    A.Field_X = B.Field_Y
```

Value

?

Accept | **Reject**

24

Ready?

Query 1

25

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID
```

Value

Accept | Reject

26

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID
```

Value

T14 Mr Mak

Accept | Reject

27

Accept

28

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID
```

Value

Accept | Reject

29

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID
```

Value

T07 Miss Ho

Accept | Reject

30

Accept

31

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID
```

Value

Accept | Reject

32

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID
```

Value

T12 Mr Leung

Accept | Reject

33

Reject

34

35

Ready?

Query 2

36

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID AND QUOTA > 50
```

Value

Accept | Reject

37

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID AND QUOTA > 50
```

Value

T10 Miss Kong

Accept | Reject

38

Accept

39

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID AND QUOTA > 50
```

Value

Accept | Reject

40

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID AND QUOTA > 50
```

Value

T15 Mr Wong

Accept | Reject

41

Reject

42

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID AND QUOTA > 50
```

Value

Accept | Reject

43

Query

```
SELECT   T.TID, T.TNAME
FROM     TEACHER T, CLUB C
WHERE    T.TID = C.TID AND QUOTA > 50
```

Value

T13 Mr Chow

Accept | Reject

44

Reject

45

46

3

Natural Join

47

Natural Join
auto match the fields
with the same name

48

SELECT *
FROM Student
NATURAL JOIN Exam

SID	Name
1001	Jason
1002	Alexis

Student

SID	Score
1001	43
1002	84

Exam

49

SELECT *
FROM Student
NATURAL JOIN Exam

SID	Name	Score
1001	Jason	43
1002	Alexis	84

50

51

Game
The Librarian

15 minutes

52

The Librarian

Help the Librarian find the information!

1. Get into **your group**
2. **Requests** will be given
3. **Constructs statement**
   as soon as possible

500 pt max

53

54

SUMMARY

Retrieving Information from Multiple Tables

Types of Join Equi-join Natural Join

55

Why storing data in multiple tables?

Avoid data redundancy
Save storage

Ensure data integrity

56

Joins

1. **Inner Join**
   Joining results when the values is matched in both tables
2. **Outer Join**
   Joining results even if there is no match values

57

Inner Join

Joining results when the values is matched in both tables

1. **Equi-join**
   - Filtering the results by fields with equal sign
2. **Natural Join**
   - Matching the values of the fields with the same name
   - Eliminating the repeated fields

58

```
SELECT   S.Name, E.Score
FROM     Student S, Exam E
WHERE    S.SID = E.SID
```

S.Name	E.Score
Jason	43
Alexis	84

59


```
SELECT   *
FROM     Student
NATURAL JOIN Exam
```

SID	Name	Score
1001	Jason	43
1002	Alexis	84

60

See you in the next lesson!

61

Icons created by Google – Flaticon

62

Information and Communication Technology

ICT 8

Lesson 8
Multi-table Query (2)

1

LAST LESSON

Retrieving Information from Multiple Tables

Types of Join Equi-join Natural Join

2

Why storing data in multiple tables?

Avoid data redundancy
Save storage

Ensure data integrity

3

Joins

1. **Inner Join**
   Joining results when the values is matched in both tables
2. **Outer Join**
   Joining results when the values is matched one of the tables

4

Inner Join

Joining results when the values is matched in both tables

1 Equi-join

- Filtering the results by fields with equal sign

2 Natural Join

- Matching the values of the fields with the same name
- Eliminating the repeated fields

5

```
SELECT   S.Name, E.Score
FROM     Student S, Exam E
WHERE    S.SID = E.SID
```

S.Name	E.Score
Jason	43
Alexis	84

6

```
SELECT   *
FROM     Student
NATURAL JOIN Exam
```

SID	Name	Score
1001	Jason	43
1002	Alexis	84

7

8

TODAY

Retrieving Information from Multiple Tables

Left Outer Join

Right Outer Join

Full Outer Join

9

Joins

1. **Inner Join**
   Joining results when the values is matched in both tables
2. **Outer Join**
   Joining results even if there is no match values

10

Outer Join

Joining results even if there is no match values

1. **Left Outer Join**
   all records from the **left**
2. **Right Outer Join**
   all records from the **right table**
3. **Full Outer Join**
   all records from **both tables**

NULL will be assigned if **no matching** value is found

11

12

1

Left Outer Join

13

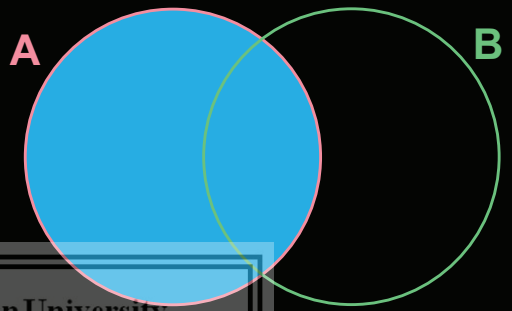
Left Outer Join

getting all records on **left table**

14

Left Outer Join

Selecting all records of **the left table** regardless of the existence of matching values.



15

```
SELECT  *  FROM  Citizen C
LEFT JOIN  Fine F
ON  C.ID = F.ID
```

ID	Name
Y01	John
Y02	Kiki

Citizen

ID	Fine
Y01	10
X08	20

Fine

16

```
SELECT  *  FROM  Citizen C
LEFT JOIN     Fine F
ON            C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
Y02	Kiki	**NULL**	**NULL**

17

18

2

Right Outer Join

19

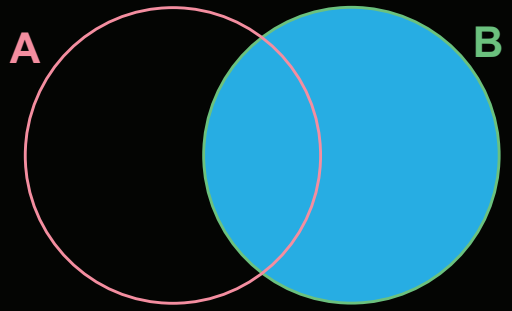
Right Outer Join

getting all records on **right table**

20

Right Outer Join

Selecting all records of **the right table** regardless of the existence of matching values.



21

```
SELECT * FROM Citizen C
RIGHT JOIN Fine F
ON C.ID = F.ID
```

ID	Name
Y01	John
Y02	Kiki

Citizen

ID	Fine
Y01	10
X08	20

Fine

22

```
SELECT * FROM Citizen C
RIGHT JOIN Fine F
ON C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
NULL	NULL	X08	20

23

24

3

Full Outer Join

25

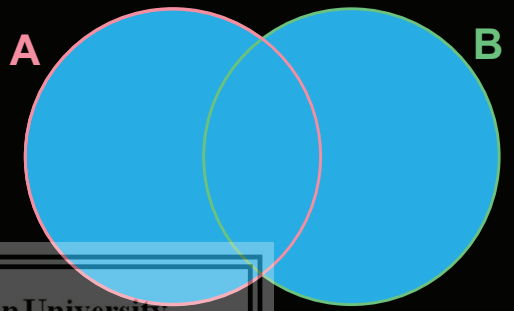
Full Outer Join

getting all records from **both tables**

26

Full Outer Join

Selecting all records from **both tables** regardless of the existence of matching values.



27

```
SELECT  *  FROM  Citizen C
FULL JOIN  Fine F
ON  C.ID = F.ID
```

ID	Name
Y01	John
Y02	Kiki

Citizen

ID	Fine
Y01	10
X08	20

Fine

28

```
SELECT * FROM Citizen C
FULL JOIN Fine F
ON C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
Y02	Kiki	NULL	NULL
NULL	NULL	X08	20

29

30

Game

Speed-reader

15 minutes

31

Speed-reader

Match the join type with the results

1. **Read** the results
2. **Think** of which type of join
3. **Raise your card**
   for the respective answer

500 pt max

32

Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

33

Ready?

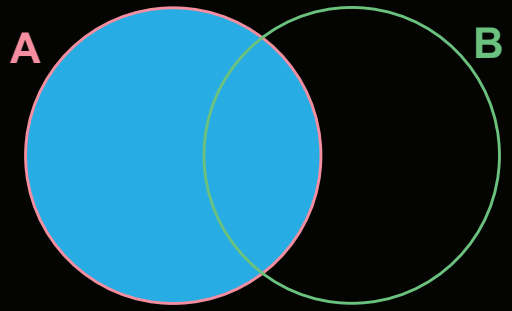
Round 1

34

Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

35



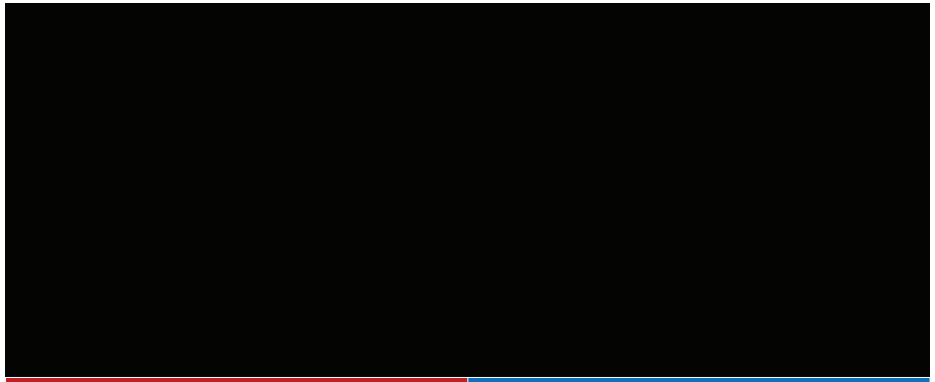
Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

36

Left Outer Join

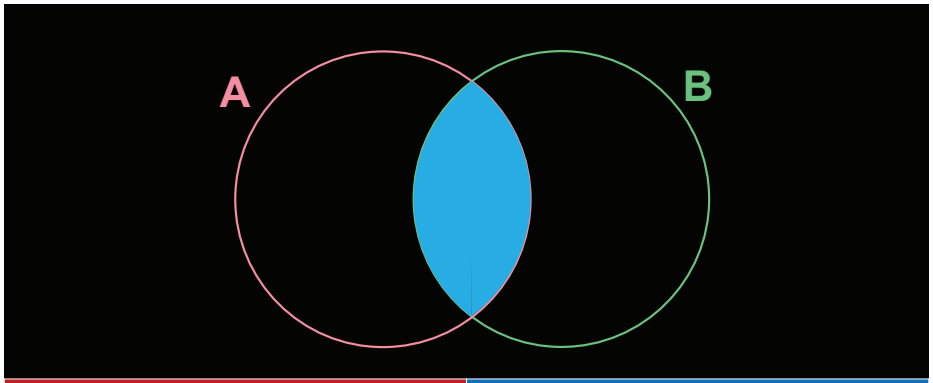
37



Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

38



Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

39

Inner Join

40

41

Ready?
Round 2

42

CID	Name
C01	Jack
C02	Polly

Candidate

SID	Subj
C01	ICT
C04	Econ

Elective

Inner Join | **Full Outer Join**
Left Outer Join | **Right Outer Join**

43

CID	Name
C01	Jack
C02	Polly

Candidate

SID	Subj
C01	ICT
C04	Econ

Elective

CID	Name	SID	Subj
C01	Jack	C01	ICT
NULL	NULL	C04	Econ

Inner Join | **Full Outer Join**
Left Outer Join | **Right Outer Join**

44

Right Outer Join

45

CID	Name
C01	Jack
C02	Polly

Candidate

SID	Subj
C01	ICT
C04	Econ

Elective

Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

46

CID	Name
C01	Jack
C02	Polly

Candidate

SID	Subj
C01	ICT
C04	Econ

Elective

CID	Name	SID	Subj
C01	Jack	C01	ICT

Inner Join | Full Outer Join

Left Outer Join | Right Outer Join

47

Inner Join

48

49

SUMMARY

Retrieving Information from Multiple Tables

Left Outer Join | Right Outer Join | Full Outer Join

50

Outer Join

Joining results even if there is no match values

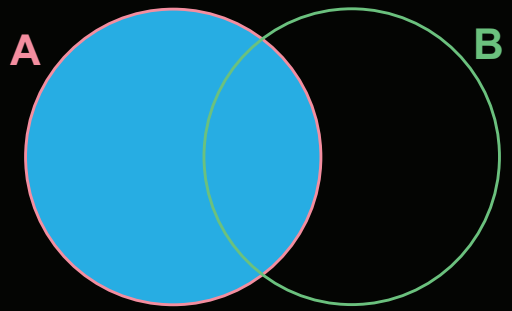
1. **Left Outer Join** all records from the **left**
2. **Right Outer Join** all records from the **right table**
3. **Full Outer Join** all records from **both tables**

NULL will be assigned if **no matching** value is found

51

Left Outer Join

Selecting all records of **the left table** regardless of the existence of matching values.



52

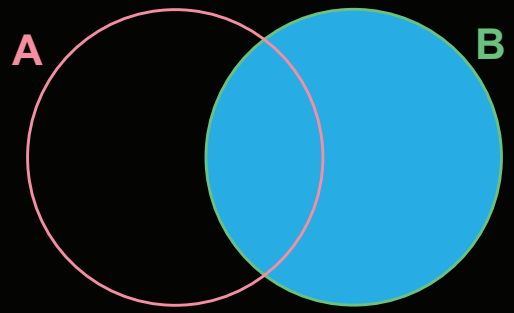
```
SELECT  *  FROM  Citizen C
LEFT JOIN      Fine F
ON             C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
Y02	Kiki	**NULL**	**NULL**

53

Right Outer Join

Selecting all records of **the right table** regardless of the existence of matching values.



54

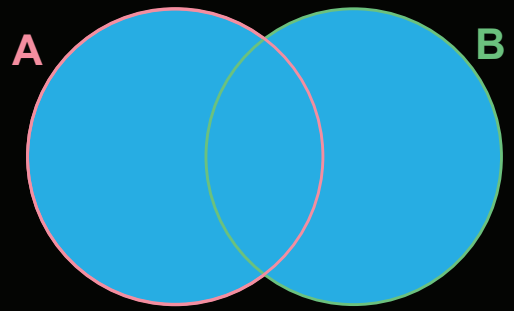
```
SELECT  *  FROM  Citizen C
RIGHT JOIN     Fine F
ON             C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
NULL	**NULL**	X08	20

55

Full Outer Join

Selecting all records from **both tables** regardless of the existence of matching values.



56

```
SELECT * FROM Citizen C
FULL JOIN Fine F
ON C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
Y02	Kiki	NULL	NULL
NULL	NULL	X08	20

57

See you in the next lesson!

58

Icons created by Google – Flaticon

59

Information and Communication Technology

ICT 9

Lesson 9
Sub-query

1

LAST LESSON

Retrieving Information from Multiple Tables

Left Outer Join

Right Outer Join

Full Outer Join

2

Outer Join

Joining results even if there is no match values

1 **Left Outer Join**
all records from the **left**

2 **Right Outer Join**
all records from the **right table**

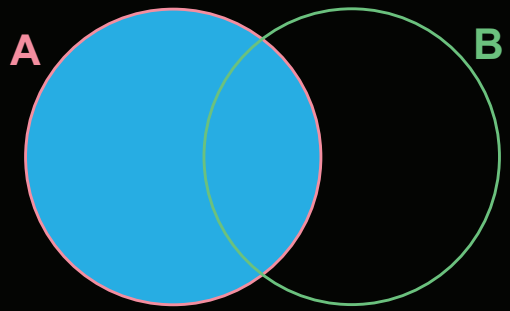
3 **Full Outer Join**
all records from **both tables**

NULL will be assigned if **no matching** value is found

3

Left Outer Join

Selecting all records of **the left table** regardless of the existence of matching values.



4

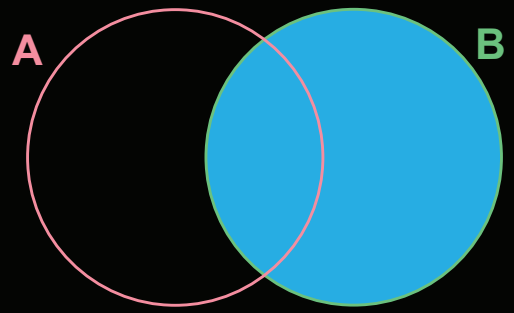
```
SELECT  *  FROM  Citizen C
LEFT JOIN      Fine F
ON             C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
Y02	Kiki	**NULL**	**NULL**

5

Right Outer Join

Selecting all records of **the right table** regardless of the existence of matching values.



6

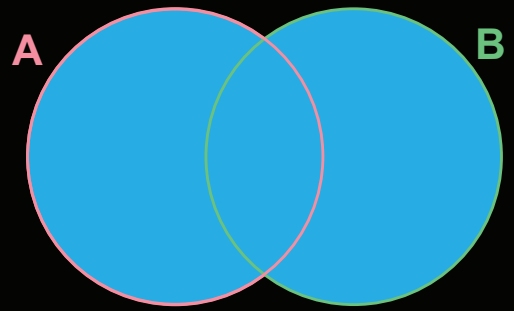
```
SELECT  *  FROM  Citizen C
RIGHT JOIN     Fine F
ON             C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
NULL	**NULL**	X08	20

7

Full Outer Join

Selecting all records from **both tables** regardless of the existence of matching values.



8

```
SELECT * FROM Citizen C
FULL JOIN Fine F
ON C.ID = F.ID
```

C.ID	C.Name	F.ID	F.Fine
Y01	John	Y01	10
Y02	Kiki	NULL	NULL
NULL	NULL	X08	20

9

10

TODAY

Performing Sub-queries

Testing Operators

Set Operators

11

12

1

Sub-query
with Testing Operators

13

6 ÷ 2 (1 + 2)

14

Do you remember?

=IF(A2>C1, 'A', **IF(…))**

15

#

Sub-query
with Single Value

16

```
SELECT *          FROM   Exam
WHERE  Score >
       (SELECT AVG(Score)
        FROM   Exam       )
```

17

Sub-query
with Multiple Values

18

Testing Operators
for Sub-query

1 **IN**
equals to any of the values in the sub-query

2 **EXISTS**
if sub-query returns results

3 **ALL**
condition matches with all the sub-query results

4 **ANY**
condition matches with any of the sub-query results

19

```
SELECT *          FROM   Student
WHERE  SID IN
       (SELECT SID
        FROM   Borrow
        WHERE  FINE > 0          )
```

20

```
SELECT *          FROM    Student S
WHERE  EXISTS
          (SELECT SID
           FROM   Borrow B
           WHERE  S.SID = B.SID      )
```

21

```
SELECT SID        FROM    Borrow
WHERE  Fine >= ALL
          (SELECT Fine
           FROM   Borrow             )
```

22

```
SELECT BID        FROM    Book
WHERE  Year = ANY
          (SELECT Year
           FROM   Tablet             )
```

23

24

Game

SQL King

10 minutes

25

SQL King

Choose operators by moving around

1. **Stand up**
2. **Usages** will be shown
3. **Stay or Move to Left / Right**
   speed counts

300 pt max

26

Usage

?

IN | EXISTS | ALL

27

Ready?

Query 1

28

Usage

ANY ALL

29

Usage

Find student with the maximum borrowing records

ANY ALL

30

ALL

31

Usage

EXISTS ALL

32

Usage

Find the students who has borrowed book from the Library

EXISTS | ALL

33

EXISTS

34

35

2 Sub-query with Set Operators

36

```
SELECT *          FROM   Exam
WHERE  Score >
       (SELECT AVG(Score)
        FROM   Exam       )
```

37

Set Operators

for results (sets) with same structure

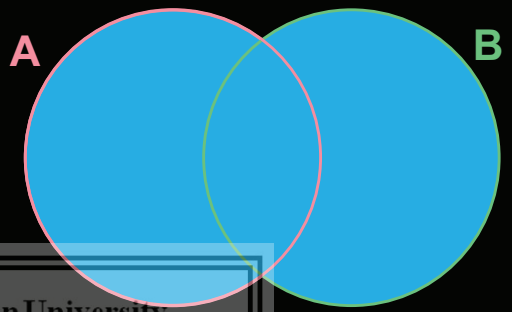
1. **UNION**
   combine two query results
2. **INTERSECT**
   get the records exist in **both query results**
3. **MINUS**
   get the records only exist in **the 1st query results**

38

UNION

Combine two query results.

Query_1 UNION Query_2

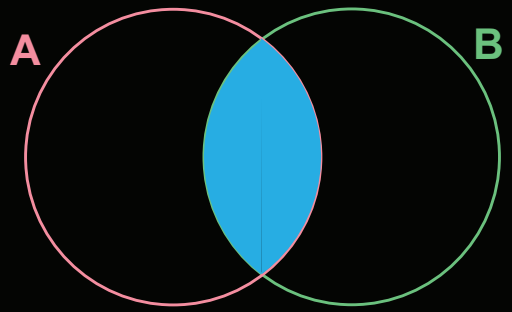


39

INTERSECT

Get the records exist in **both query results**.

Query_1 INTERSECT Query_2

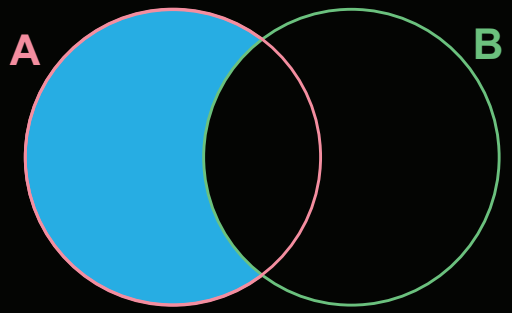


40

MINUS

Get the records only exist in the **first query results**.

Query_1 MINUS Query_2



41

42

SUMMARY

Performing Sub-queries

Testing Operators

Set Operators

43

Testing Operators

for Sub-query

1. **IN** equals to any of the values in the sub-query
2. **EXISTS** if sub-query returns results
3. **ALL** condition matches with all the sub-query results
4. **ANY** condition matches with any of the sub-query results

44

Set Operators

for results (sets) with same structure

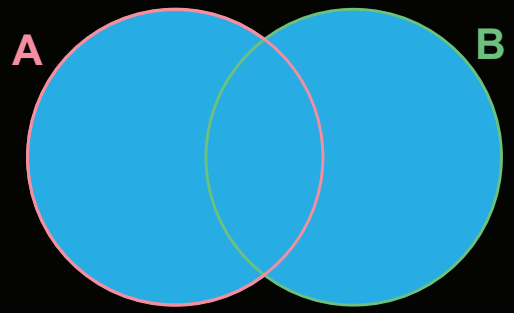
1. **UNION**
   combine two query results
2. **INTERSECT**
   get the records exist in **both query results**
3. **MINUS**
   get the records only exist in **the 1st query results**

45

UNION

Combine two query results.

Query_1 UNION Query_2

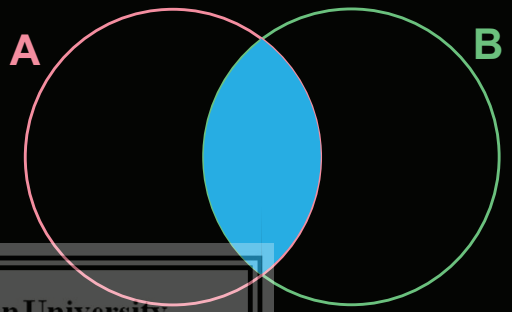


46

INTERSECT

Get the records exist in **both query results**.

Query_1 INTERSECT Query_2

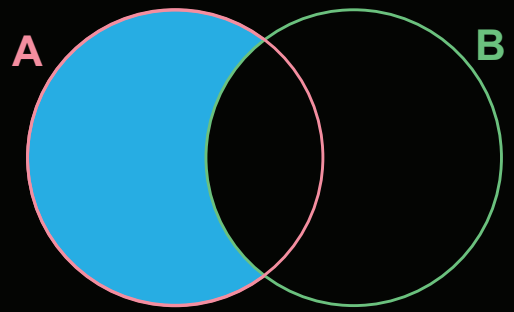


47

MINUS

Get the records only exist in the **first query results**.

Query_1 MINUS Query_2



48

See you in the next lesson!

49

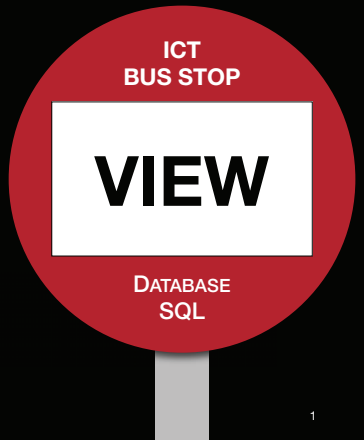
Icons created by Google – Flaticon

50

Information and Communication Technology

ICT 10

Lesson 10
Access Rights and Summary



1

LAST LESSON

Performing Sub-queries

Testing Operators

Set Operators

2

Testing Operators

for Sub-query

1 **IN**
equals to any of the values in the sub-query

2 **EXISTS**
if sub-query returns results

3 **ALL**
condition matches with all the sub-query results

4 **ANY**
condition matches with any of the sub-query results

3

Set Operators

for results (sets) with same structure

1 **UNION**
combine two query results

2 **INTERSECT**
get the records exist in **both query results**

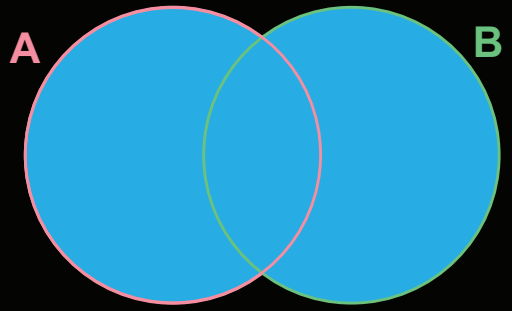
3 **MINUS**
get the records only exist in **the 1st query results**

4

UNION

Combine two query results.

Query_1 UNION Query_2

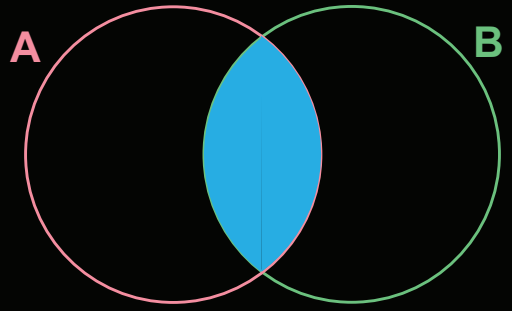


5

INTERSECT

Get the records exist in **both query results**.

Query_1 INTERSECT Query_2

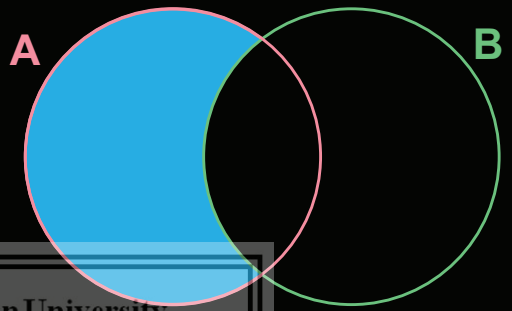


6

MINUS

Get the records only exist in the **first query results**.

Query_1 MINUS Query_2



7

TODAY

Limiting Access Rights with Views

8

9

1

Access Control
with Views

10

Should we allow the end users accessing the database?

11

Views
Restricting users by specifying the items could be viewed

12

CREATE VIEW View Name

AS SELECT Column

FROM Table

WHERE Condition

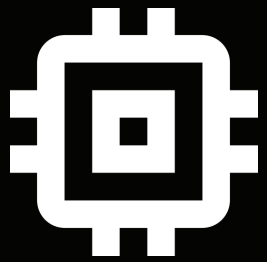
13

14

Group Activity

System Design

15 minutes



15

System Design

Upgrade your system design with 3 applications

1. Get into **your group**
2. **Brainstorm 3 applications**
   based on your queries designed
3. **Present** your system for 3 minutes

16

Presentation

Share your amazing system

1. **Listen**
2. **Support**
3. **Feedback**

17

18

Unit Summary

Use of SQL

19

Thank you!

20

The Education University of Hong Kong
INT4901 Honours Project II
Package Guide

OVERVIEW

Motivating Students to Learn Concepts on Relational Databases via Gamification is created aligned with the HKDSE ICT C&A Guide.

The package is designed for learning SQL 10 1-hour lessons with the following deliverables:

1. lesson plans;
2. presentation slides, worksheets, and game materials; and
3. online game learning platform ('SQLGo').

OBJECTIVES

The package is aimed to:

1. allow students to master the fundamental concepts of relationships and skills of data manipulation with SQL;
2. motivate students to learn relational database concepts through gamification-based design; and
3. maximise the students' learning effectiveness on relational databases by reducing the cognitive load with multimedia and game elements.

SQLGO INSTALLATION

Environment

Item	Version
HTTP	Apache 2.4
Database	MariaDB 10.3.32
PHP	7.4

Database Accounts

Account	Description	Permission
sqlapp	System account for user and game management.	ALL for sqlgo_app
sqlgame	Game database allow students' query to be executed.	SELECT for sqlgo_game and sqlgo_challenge

Database Installation

1. Create the two accounts in the above table.
2. Execute all 3 SQL files in the 'Installation SQL.zip'.

Configuration

Item	Setting	Value
Website Root Directory	sqlgo/public	
Writable Directory	sqlgo/writable	
sqlgo/.env (hidden file)	app.baseURL (line 23)	Application URL
sqlgo/Config/App.php	public string $baseURL (line 20)	Application URL
sqlgo/Config/Database.php	$default password (line 31)	Password for 'sqlapp'
sqlgo/Config/Database.php	$gamedb password (line 51)	Password for 'sqlgame'
sqlgo/Config/Database.php	$challengedb password (line 71)	Password for 'sqlgame'

SQLGo Admin Account

Username	Default Password	Description
admin@sql.go	WelcomeSQL	Default administrator account will be generated if no administrator is found.

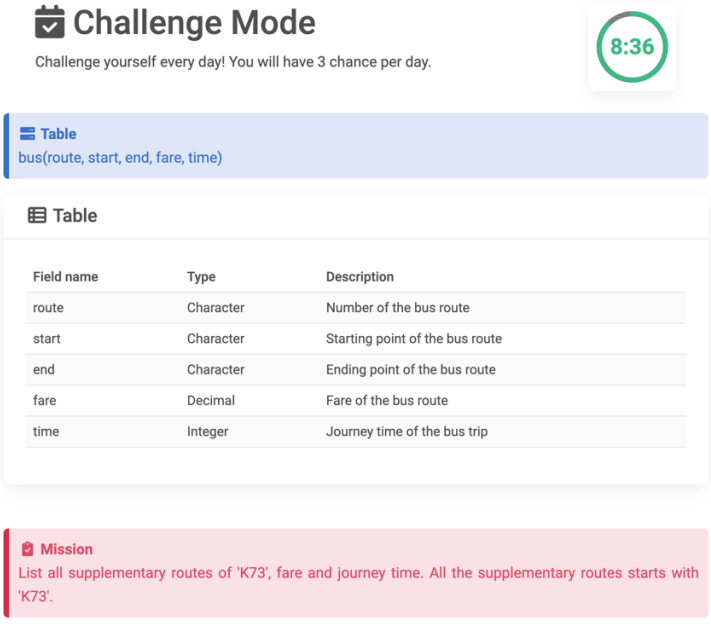
SQLGo Features

Story Mode

Students will follow a storyline to help solve problems.

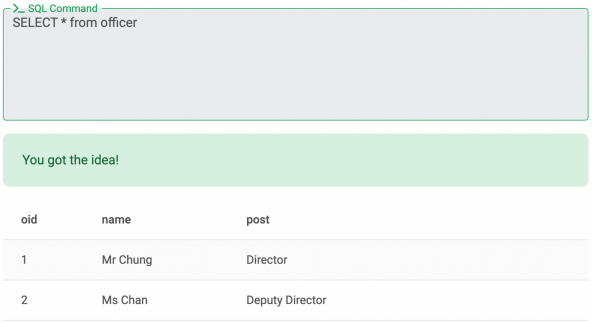
Challenge Mode

System provides three random questions per day for students to exert themselves with a time limit.

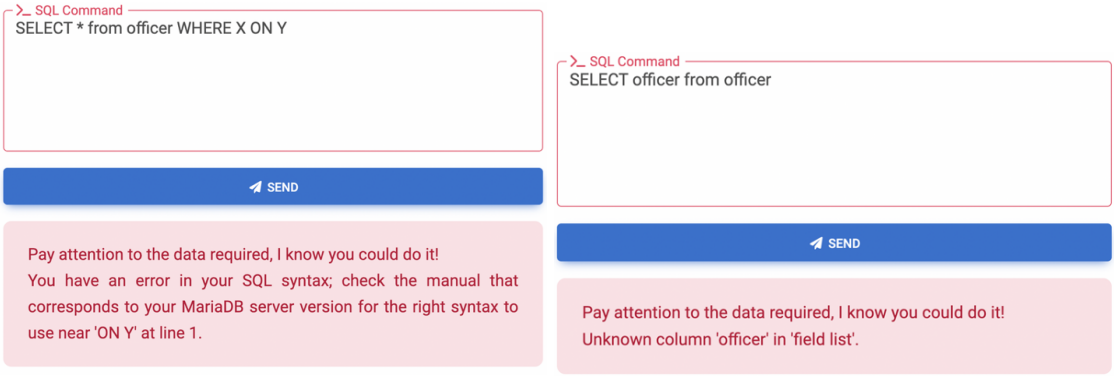


SQL Statement Verification

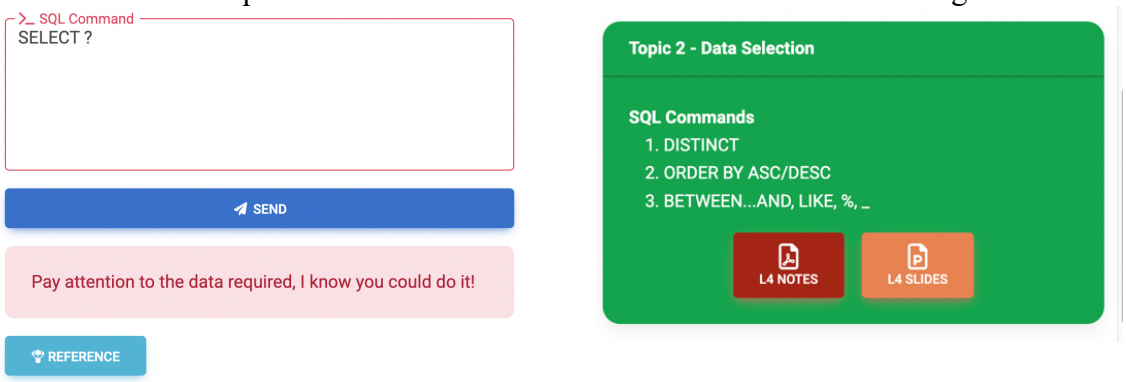
SQLGo verifies the answer by executing the query and comparing the results, which allows all equivalent statements to complete the mission.



Students will see the error message returned from the database to help them understand the reason for the mistakes.

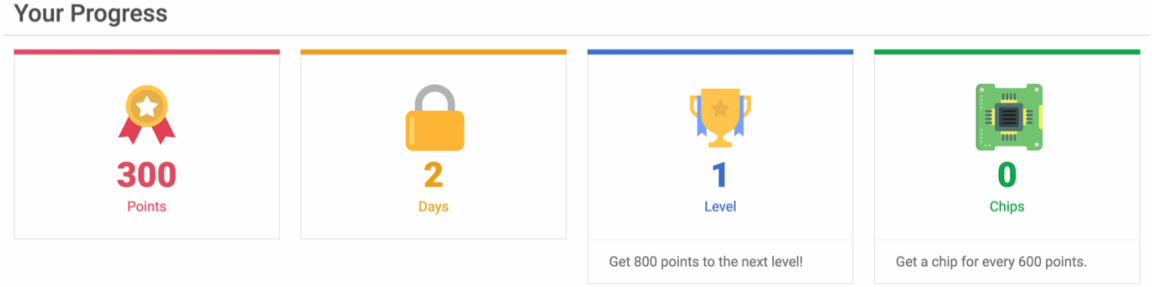


A link to relevant topic resources will be shown to assist students in learning.



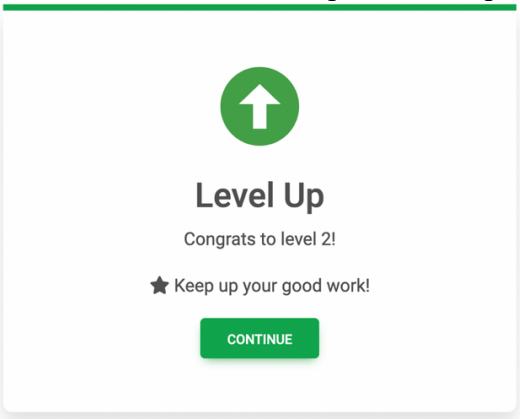
Progress

Students will be awarded points and chips for completing the mission or challenge as incentives to encourage them.

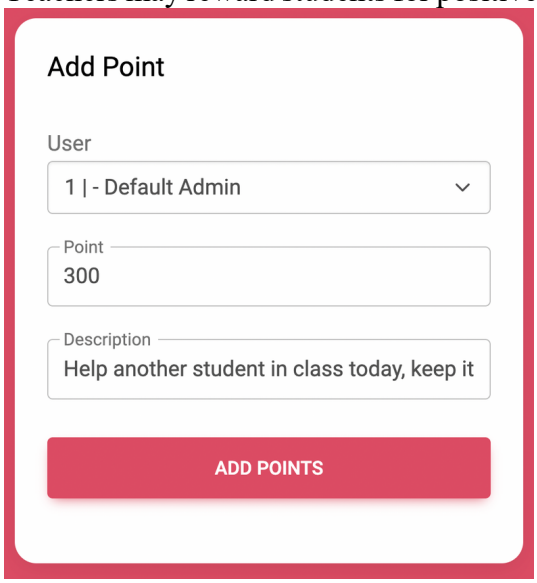


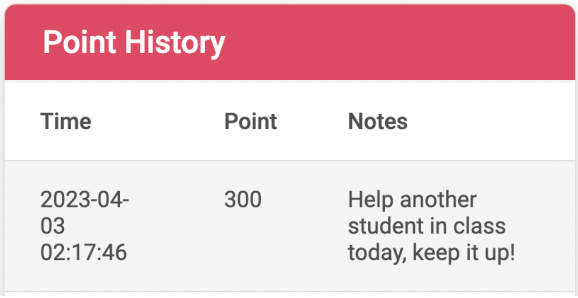
Point System

Students will be levelled up when completing enough points.



Teachers may reward students for positive behaviour in class by boosting the points on SQLGo.





Leaderboard

Leaderboard provides recognition to the top 3 leading students.