Hong Kong Baptist University

Department of Computer Science

COMP 7990 Principles and Practices of data analytics (2022-23)

Lab 3a: Structured Query Language (SQL)

Introduction

Every organization has data, which requires some organized method or mechanism for maintaining the data. This mechanism is referred to as a **database management system (DBMS),** which is a program that stores, retrieves, and modifies data in the database.

A relational database uses a structure that allows us to identify and access data in relation to another piece of data in the database. Often, data in a relational database is organized into tables. There are some popular relational databases such as Oracle, SQL Server, and MS Access.

SQL, Structured Query Language, is the standard language used to communicate with a relational database. SQL is either pronounced as the letters S-Q-L or "Sequel". SQL is a comprehensive database language. It is also a Data Manipulation Language (DML), a Data Definition Language (DDL) and a Data Control Language (DCL).

DML: work with the data in the database

- Select, Insert, Update, Delete

DDL: define the database, tables, user-defined data types and the way data is stored

- Create, Alter, Drop, Rename etc.

DCL: used to change permission associated with DB role

- Revoke, Grant

A query is an inquiry into the database using SELECT statement. A query is used to extract data from the database in a readable format according to the user's request. For instance, if you have an employee table, you might issue a SQL statement that returns the employee who is paid the most.

The **SELECT** statement is not a standalone statement, which means other clauses are required. The **FROM** clause is a mandatory clause which must always be used in conjunction with the SELECT statement. Some other clauses, like **WHERE**, can be used together with SELECT to increase its functionalities, e.g. to retrieve some rows in a table which satisfy certain conditions or to bring together different data stored in different tables.

Learning Outcomes

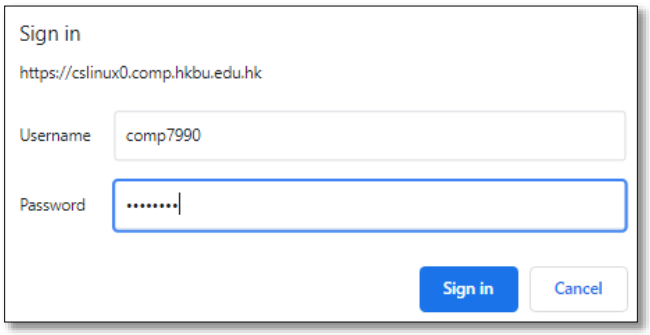
By finishing this lab session, you should be able to

- Learn how to use MySQL
- Execute some SQL statements

Connecting to a MySQL Database

Follow the steps below to login:

1. Open a browser and go to https://cslinux0.comp.hkbu.edu.hk/7990_phpmyadmin
2. There is a two-level password protection. Enter the first credential:
   a. Level 1: Username: comp7990 / Password: 7990mysql



   b. You should see a login screen of **phpMyAdmin**. Please change the **Language**. Enter the second credential as follows.
   Level 2: For student "20123456" your credentials are as follows:
   -account name: f0123456 (replace the first digit "2" by "f")
   -account password: dbf0123456

Alternatively, you may also want to run the database locally. There are many possible solutions and we recommend **Docker**. Follow the guide in the file below to use your database with Docker. The only thing is Docker is not installed in our lab machine, so you need to run them in your own machine. Use MySQLwithDocker.zip

Data Definition Language (DDL):

Syntax for creating tables

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

Syntax for dropping tables

```
DROP TABLE table_name;
```

Data Manipulation Language (DML)

Syntax for inserting records:

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

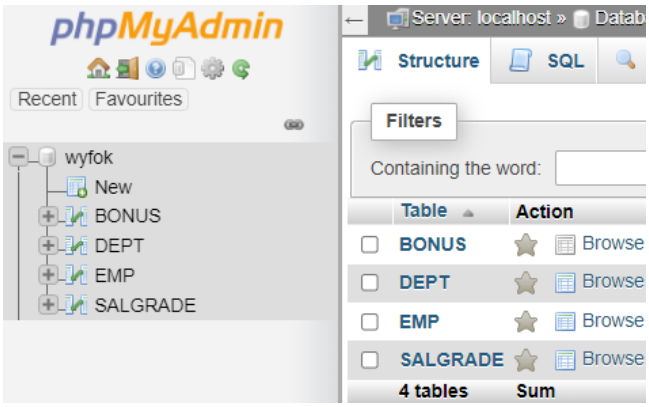
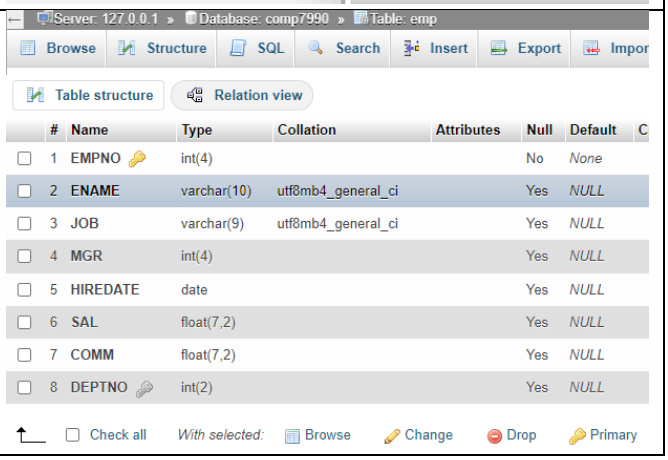
Syntax for updating records:

```
UPDATE table_name
SET column1 = value1, column2 = value2
WHERE condition;
```

Syntax for deleting records:

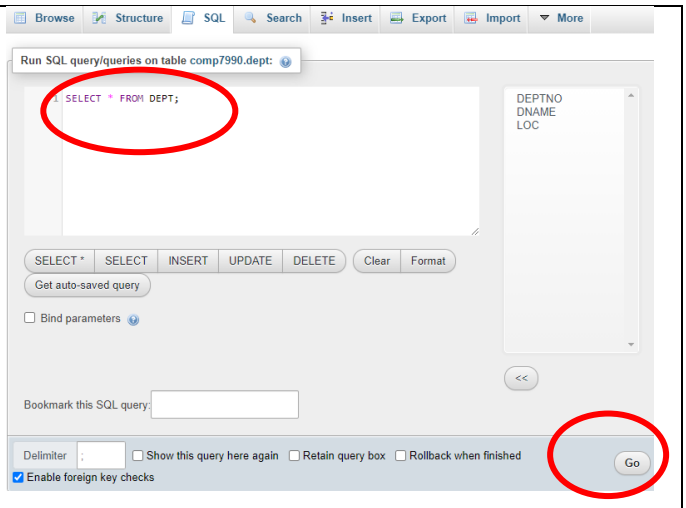
```
DELETE FROM table_name
WHERE condition;
```

Create tables and insert records

1. Four tables are created - EMP, DEPT, BONUS and SALGRADE. Use **DESCRIBE** command to display the structure of the tables.	Check **Structure** of each Table to display their structures, 
2. Here is the structure of **EMP** table.	

SELECT and WHERE clause

1. To execute a SQL Statement: • Click the table **DEPT** and click **SQL** from the menu. • Enter the command in the box and click **GO** to run the query. • **Semicolon (;)** at the end of the last clause is optional. • Table name is case sensitive while the keyword **SELECT** and **FROM** are not	To display all columns of data in a table by using asterisk (*): SELECT * FROM DEPT;

<table>
<tr>
<td></td>
<td> 

SELECT * FROM EMP;</td>
</tr>
<tr>
<td>2. Use the SELECT statement to display specific columns of the table by specifying the column names, separated by commas. (How to retrieve distinct deptno using EMP table?)</td>
<td>To display all the department numbers and locations from the DEPT table:

SELECT DEPTNO, LOC FROM DEPT;</td>
</tr>
<tr>
<td>3. Use the WHERE clause to select specific rows or records.

Note: <u>Characters MUST BE enclosed by single quote ‘ ’</u>. For number, no need ‘ ’</td>
<td>To display the name, job title, and department number of all employees whose job title is CLERK

SELECT ENAME, JOB, DEPTNO
FROM EMP
WHERE JOB= 'CLERK';

<table>
<tr><th>ENAME</th><th>JOB</th><th>DEPTNO</th></tr>
<tr><td>SMITH</td><td>CLERK</td><td>20</td></tr>
<tr><td>ADAMS</td><td>CLERK</td><td>20</td></tr>
<tr><td>JAMES</td><td>CLERK</td><td>30</td></tr>
<tr><td>MILLER</td><td>CLERK</td><td>10</td></tr>
</table>

SELECT EMPNO, ENAME, JOB
FROM EMP
WHERE EMPNO= 7839;

<table>
<tr><th>EMPNO</th><th>ENAME</th><th>JOB</th></tr>
<tr><td>7839</td><td>KING</td><td>PRESIDENT</td></tr>
</table></td>
</tr>
</table>

Arithmetic Expression and Operators

You may need to modify the way in which data is displayed, perform calculations, or look at what-if scenarios. This is possible by using arithmetic expressions. An arithmetic expression may contain column name, constant numeric values, and the arithmetic operators.

You can use **arithmetic operators** in any clause except the FROM clause.

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

Operator Precedence: If an arithmetic expression contains more than one operator, *multiplication and division will take priority over addition and subtraction*. If operators within an expression are of same priority, then evaluation is done from left to right. You can use parentheses () to force the expression within parentheses to be evaluated first.

1. **Addition** Operator Note: ➢ SQL ignores spaces before and after the arithmetic operator ➢ SAL+300 is *not a new column, for display only*	To calculate a salary increase of $300 for all employees and display a new column (SQL+300) SELECT ENAME, SAL, SAL+300 FROM EMP;
2. Use Multiple Operators and Parentheses.	SELECT ENAME, SAL, 12*SAL+100 FROM EMP; SELECT ENAME, SAL, 12*(SAL+100) FROM EMP;

Column Aliases

When displaying the result of a query, MYSQL normally uses the name of the selected column as the column heading. In many cases, this heading may not be descriptive and is difficult to understand. You can change a column heading by using a column alias.

ENAME	SAL	12*(SAL+100)

Not understandable

We can specify the alias after the column in the SELECT list using a space as a separator. If the alias contains space, special characters (such as # or $), enclose the alias in quotation marks.

1. Use **AS** and a space as a separator to change to alias heading. (optional **AS** keyword has been used)	SELECT ENAME AS NAME, SAL SALARY FROM EMP;
2. Use **single/double quotation marks** to explicitly name the column. Because "Basic Annual Income" contains spaces, it has been enclosed quotation marks. **AS** keyword is **optional**.	SELECT ENAME "Name", SAL*12 AS "Basic Annual Income" FROM EMP; Name | Basic Annual Income SMITH | 9600.00 ALLEN | 19200.00 WARD | 15000.00 JONES | 35700.00

NULL VALUE

- A null value is a value that is unavailable, unassigned, unknown or inapplicable.
- A null value is NOT the same as zero or a blank space.
- Columns can contain null values unless the column was defined as NOT NULL or as PRIMARY KEY when the column was created.
- If any column value in an arithmetic expression is null, the result is null.

1. Select all the employees **where commission is NULL**	SELECT ENAME, JOB, SAL, COMM FROM EMP WHERE COMM IS NULL;
2. Calculate the total annual salary of all employees, any problem?	SELECT ENAME, 12*(SAL+COMM) FROM EMP;
3. **COALESCE(COMM, 0)** substitutes all NULL entries in COMM column to 0	SELECT ENAME, JOB, 12*(SAL+COALESCE(COMM, 0)) FROM EMP;

Comparison Operators

Comparison operators are used in conditions that compare one expression to another. =, >, >=, <, <=, <> are all comparison operators. Other comparison operators are **LIKE**, **BETWEEN.. AND**

1. **LIKE** Operator: You may not always know the exact value to search for. LIKE operator	To select employee name from the EMP table where employee's name begins with an "S":

<table>
<tr>
<td>allows you to select rows that match a character pattern.

<table>
<tr><th>Symbol</th><th>Description</th></tr>
<tr><td>%</td><td>Represents any sequence of zero or more characters</td></tr>
<tr><td>_</td><td>Represents any single character</td></tr>
</table>
</td>
<td>SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'S%'

SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'S___';
(with four underscores [_])</td>
</tr>
<tr>
<td>2. Try BETWEEN……AND operator.</td>
<td>SELECT * FROM DEPT WHERE DEPTNO BETWEEN 10 and 30</td>
</tr>
</table>

ORDER BY Clause

ORDER BY clause can be used to sort the rows. You must place the ORDER BY clause as the last clause, and specify an expression or an alias to sort. **ASC** means **ascending** order while **DESC** means **descending** order.

1. ORDER BY: To sort the results by a certain column in ascending/descending order. Note: default is ASC	SELECT ENAME, JOB, DEPTNO, HIREDATE FROM EMP ORDER BY HIREDATE ASC;

Logical Operators (AND, OR, NOT)

A logical operator combines the result of two component conditions to produce a single result based on them or to invert the result of a single condition. Three logical operators are available in SQL: **AND, OR** and **NOT**.

Order Evaluated	Operator
1	All comparison operators like =, >, >=, <, <=, <>/!= (not equal to)
2	NOT
3	AND
4	OR

1. **AND** Operator: both conditions must be true for any record to be selected. Note: No row is returned if CLERK is not in uppercase.	SELECT EMPNO, ENAME, JOB, SAL FROM EMP WHERE SAL>=1100 AND JOB= 'CLERK';
2. **OR** Operator: either condition can be true for any record to be selected.	SELECT EMPNO, ENAME, JOB, SAL FROM EMP WHERE SAL>=1100

	OR JOB= 'CLERK';
3. **NOT** Operator: express negative conditions. Employee whose job title *is not* CLERK or ANALYST will be selected.	SELECT EMPNO, ENAME, JOB FROM EMP WHERE NOT (JOB= 'CLERK' OR JOB= 'ANALYST');
4. Combination of AND and OR (AND first, then OR)	SELECT EMPNO, ENAME, JOB, SAL FROM EMP WHERE JOB='SALESMAN' OR JOB='PRESIDENT' AND SAL>1500;

GROUP BY

Group by functions operate on sets of rows to **give one result per group**. These sets may be the whole table or the table split into groups. Some guidelines for GROUP BY clause:

a. GROUP BY statement is often used with aggregate functions (MIN, MAX, AVG, COUNT, SUM)
b. Using WHERE clause, you can pre-exclude rows before dividing them into groups.
c. If you include a group function in a SELECT clause, you cannot select individual results as well unless the individual column appears in the GROUP BY clause.
d. By default, rows are sorted by ascending order of the columns include in the GROUP BY list. You can override this using ORDER BY clause.

1. **COUNT(*)** function returns the number of rows in a table, including duplicate rows and rows containing null values. (the whole table is a group)	SELECT COUNT(*) FROM EMP; COUNT(*) 14
2. To return the number of rows that satisfies a certain condition.	SELECT COUNT(*) FROM EMP WHERE DEPTNO=30;
3. GROUP BY clause divides the rows into smaller groups. Then use the group functions to return summary information, like calculating minimum.	To select the minimum salary for the whole company: SELECT MIN(SAL) FROM EMP; To select the minimum salary in each department: SELECT DEPTNO, MIN(SAL)

	FROM EMP GROUP BY DEPTNO;
4. Group by more than one column	To display total salary being paid to each job title, within each department: SELECT DEPTNO, JOB, SUM(SAL) FROM EMP GROUP BY DEPTNO, JOB;

HAVING clause

HAVING clause will filter the records that work on summarized GROUP BY results.

1. Place the HAVING and GROUP BY clauses after the WHERE clause in a statement. Then, place the ORDER BY clause at the end.	To display department numbers and maximum salary for those departments (less than 40) whose maximum salary is greater than $2900: SELECT DEPTNO, MAX(SAL) FROM EMP WHERE DEPTNO < 40 GROUP BY DEPTNO HAVING MAX(SAL)>2900 ORDER BY DEPTNO DESC;

JOIN

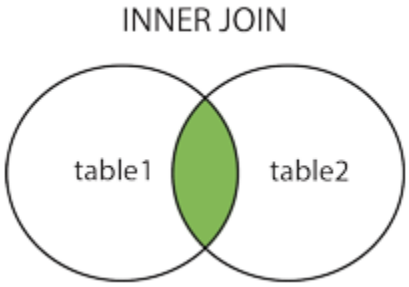
Sometime, you need to use data from more than one table, Join condition can be used. Rows in one table can be joined to rows in another table according to common values existing in corresponding columns, that is usually primary and foreign key columns.

The **SELECT** clause specific the column name to retrieve:

- Employee name, employee number, and department number, which are columns in the EMP table.
- Department number, department name, and location, which are columns in the DEPT table.

The **FROM** clause specifies the two tables that the database must access:

- EMP table
- DEPT table



The **WHERE** clause specifies how the tables are to be joined:

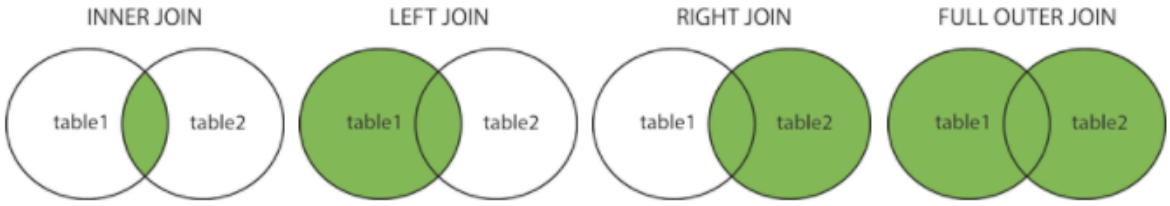
EMP.DEPTNO=DEPT.DEPTNO

1. Because the DEPTNO column is common to both tables, it must be prefixed by the table name to avoid ambiguity. (Inner join)	SELECT EMP.EMPNO, EMP.ENAME, EMP.DEPTNO, DEPT.DEPTNO, DEPT.LOC FROM EMP, DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO; SELECT EMPNO, ENAME, EMP.DEPTNO, LOC FROM EMP, DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO; SELECT EMP.EMPNO, EMP.ENAME, EMP.DEPTNO, DEPT.DEPTNO, DEPT.LOC FROM EMP inner join DEPT on EMP.DEPTNO=DEPT.DEPTNO
2. Select empno, ename, deptno, loc from the two tables, also include deptno that does not appear in emp table. (Right join) [empno 40 is included in the result]	SELECT EMP.EMPNO, EMP.ENAME, EMP.DEPTNO, DEPT.DEPTNO, DEPT.LOC FROM EMP right join DEPT on EMP.DEPTNO=DEPT.DEPTNO
3. Self-join: join a table to itself	To find the name of each employee's manager: SELECT E.ENAME AS WORKER, M.ENAME AS MANAGER FROM EMP E, EMP M WHERE E.MGR=M.EMPNO ORDER BY M.ENAME;

Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table



(Ref: https://www.w3schools.com/sql/sql_join.asp)

SUBQUERY

A subquery is a SELECT statement that is embedded in a clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries

Suppose you want to find out who earns a salary greater than Jones' salary. To solve this problem, you can combine two queries, placing one query inside the other:

- One query to find what Jones earns, and
- Second query to find who earns more than that amount.

Some Guidelines:

- Enclose subqueries in parentheses ().
- Place subqueries on the RIGTH side of the comparison operator.
- Do not add ORDER BY clause to a subquery.

1. Single row subquery (the subquery returns only one value)	SELECT * FROM EMP E WHERE E.SAL = (SELECT MIN(SAL) FROM EMP);
2. Inner query determines the salary of an employee (returns only one value), the outer query takes the result of the inner query and uses this result to display someone who earn more than this amount.	To display the employees who earn more than the salary of employee 7566 (Jone's salary) SELECT ENAME, EMPNO, SAL FROM EMP WHERE SAL > (SELECT SAL FROM EMP WHERE EMPNO=7566);

Reference

- https://www.w3schools.com/sql/
- https://www.w3schools.com/MySQL/default.asp
- https://www.tutorialspoint.com/mysql/index.htm

Take home assignment

Download the file **Lab3a-SQL.zip**. Open the file **lab3a-assignment-ans.docx**, complete it individually.

Submission

Submit the following file to buelearning website:

- lab3a-assignment-ans.docx