<u>RKIT Training Notes</u>
<u>Week - 1</u>

INTRODUCTION:

What is SQL?

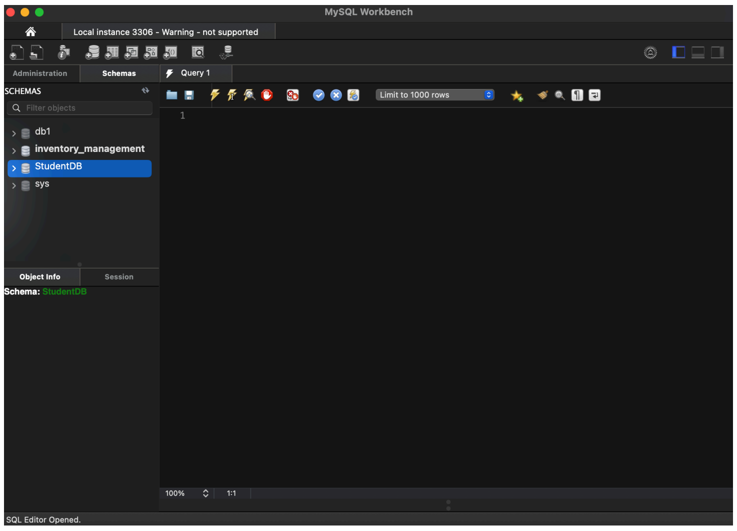
- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL is an ANSI (American National Standards Institute) standard

MySQL Workbench is a visual software tool for MySQL databases.
It combines SQL development, database design, administration, and migration into a single application for database architects, developers, and administrators.

This is for Database Administrators (DBAs).
Workbench provides a graphical interface to manage the MySQL server, simplifying tasks that would otherwise require command-line operations.

What SQL can do?

- SQL is used to query and manipulate data by retrieving, inserting, updating, or deleting records.
- It also defines and manages the database's structure by creating new databases, tables, views, and procedures.
- Finally, SQL controls access to the database by setting user permissions on various database objects.



RDMS stands for Relational Database Management System.
It’s a basic for SQL, and all other database systems like MS SQL server, IBM DB2, Oracle, MySQL and Microsoft Access.

SQL Basics:

There are 4 types of queries:

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)
3. DQL (Data Query Language)
4. DCL (Data Control Language)

DDL AND DML:

Data Definition Language (DDL) commands are used in SQL to define and manage the structure of your database and its objects, like tables.
They don't deal with the actual data, but rather the blueprint that holds the data.
Ex: CREATE, ALTER, DROP, TRUNCATE, RENAME, COMMENT

DML (Data Manipulation Language) commands are used in SQL to manage and manipulate the actual data stored within database tables.
While DDL commands build the structure, DML commands work with the information inside that structure.
Ex: INSERT, UPDATE, DELETE, MERGE

DQL is used to retrieve data from the database. It's technically a subset of DML, but often categorized separately.
Ex: SELECT

DCL commands are used to manage user permissions and access levels to the database objects.
Ex: GRANT, REVOKE

Create:

```
CREATE TABLE TableName (
   Column1 DataType,
   Column2 DataType,
   ...
);
```

Alter:

```
ALTER TABLE TableName
ADD ColumnName DataType;
```

Drop:
DROP TABLE TableName;

Truncate:
TRUNCATE TABLE table_name;   //deletes all rows quickly and resets the table's
                                             AUTO_INCREMENT, keeps the structure

—---

INSERT:
INSERT INTO table_name VALUES (value1, value2, ...);

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

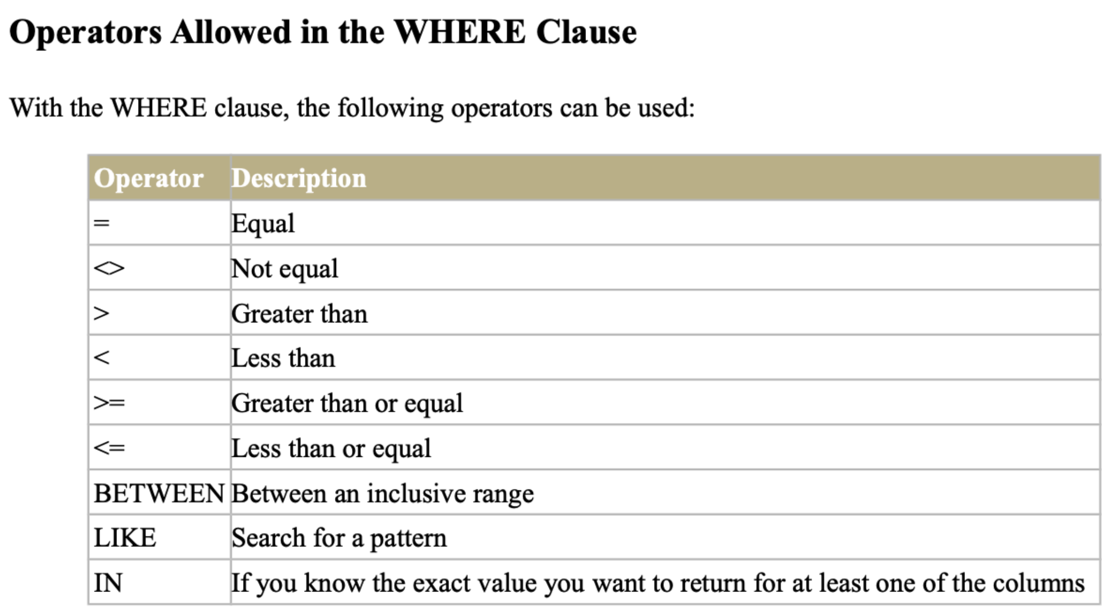
DELETE:
```
DELETE FROM table_name
WHERE condition;
```

MERGE:
```
MERGE INTO target_table t
USING source_table s
ON (t.id = s.id)
WHEN MATCHED THEN
   UPDATE SET t.col1 = s.col1
WHEN NOT MATCHED THEN
   INSERT (id, col1)
   VALUES (s.id, s.col1);
```
—---

SELECT *FROM table_name;

DISTINCT:
This keyword can be used to return only distinct (different) values.
SELECT DISTINCT column_name(s) FROM table_name

WHERE Clause:
The WHERE clause is used to filter records.
The WHERE clause is used to extract only those records that fulfill a specified criterion.



Operators Allowed in the WHERE Clause

With the WHERE clause, the following operators can be used:

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	If you know the exact value you want to return for at least one of the columns

ORDER BY:

- The ORDER BY keyword is used to sort the result-set.
- The ORDER BY keyword is used to sort the result-set by a specified column.
- The ORDER BY keyword sorts the records in ascending order by default.
- If you want to sort the records in a descending order, you can use the DESC keyword, ASC for ascending

Ex: select * from Students ORDER BY name ASC;

GROUP BY:

This clause in SQL is used to collect and arrange identical data into groups. It's almost always used with **aggregate functions** like COUNT(), MAX() or MIN() to perform a calculation on each of these groups.

Ex: select course_id, COUNT(*) AS num_students FROM Students GROUP BY course_id;

SQL JOINS:

- SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables.
- The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.
- Tables in a database are often related to each other with keys.
- A primary key is a column (or a combination of columns) with a unique value for each row.
- Each primary key value must be unique within the table.

- The purpose is to bind data together, across tables, without repeating all of the data in every table.

Types of joins:
- Inner Join
- Right Join
- Left Join
- Full Join
- Cross Join
- Self Join

INNER JOIN:
Returns only the records that have matching values in **both** tables. This is the most common type of join

Ex:
```
select s.name, c.course_name
from students s
inner join courses c on s.course_id = c.course_id;
```

RIGHT JOIN:
Returns **all** records from the right table, and the matched records from the left table. If there is no match, the result is `NULL` on the left side.

Ex:
```
select c.course_id, c.course_name, s.student_id, s.name as student_name
from students s
right join courses c on s.course_id = c.course_id;
```

LEFT JOIN:
Returns **all** records from the left table, and the matched records from the right table. If there is no match, the result is NULL on the right side.

Ex:
```
select s.name, c.course_name
from students s
left join courses c on s.course_id = c.course_id;
```

FULL JOIN:
Returns **all** records when there is a match in either the left or the right table. It's essentially the combination of LEFT JOIN and RIGHT JOIN.

Ex:
```
select s.student_id, s.name, c.course_name
from Students s
FULL JOIN Courses c
on s.student_id = c.student_id;
```