Prerequisite

Mysql Workbench
Mysql server

Class 1:
Create connection
Create database

Drawing ER Diagram for different Scenario
SQL Query on DDL Commands

Query:

Database creation:

```
    CREATE DATABASE databasename;
```

Create table:

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

```
USE DATABASENAME;
```

This command will redirect to our database

Drop table:

```
DROP TABLE table_name;
```

Deletes entire table

Truncate table:

```
TRUNCATE TABLE table_name;
```

Deletes the data inside the table

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype;
```

ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

ALTER TABLE - RENAME COLUMN

To rename a column in a table, use the following syntax:

```
ALTER TABLE table_name
RENAME COLUMN old_name  new_name data_type;
```

```
ALTER TABLE student.student_details CHANGE  column id
usn int;
```

ALTER TABLE - ALTER/MODIFY DATATYPE

To change the data type of a column in a table, use the following syntax:

```
ALTER database_name.table_name MODIFY column_name;
```

```
ALTER TABLE  student. student_details MODIFY column id  int;
```

DML COMMANDS:

SELECT , INSERT , DELETE , and UPDATE

SELECT:

```
SELECT column1, column2, ...
FROM table_name;
```

```
SELECT * FROM Customers;
```

```
SELECT name FROM Customers;
```

The `SELECT DISTINCT` statement is used to return only distinct (different) values.

```
Select distinct column_name from table_name;
```

Queries on multiple relation:

```
Select column_name, database_name.column_name, from tablename,tablename where
database_name.column_name>or<or=another database_name.column_name;
```

```
Can use AND operator to check multiple condition using where clause
```

EXAMPLE:

I have department table and faculty table

```
SELECT name, faculty.dept_name, block FROM faculty, department WHERE
faculty.dept_name=department.dept_name;
```

STRING OPERATIONS:

‘Intro%’ matches any string beginning with “Intro”.

‘%Comp%’ matches any string containing “Comp”.

‘---’matches any string exactly three characters.

‘---%’ matches any string of atleast three characters.

Like is the comparison operator

```
SELECT dept_name FROM department where building like ‘%ablock%’;
```

Ordering the display of table:

```
Order by is the operator
```

```
SELECT name from faculty WHERE det_name=’CA’ ORDER BY name;
```

```
This will return us the ordered table or arranged table in alphabetical order
```

INSERT:

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

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode,
Country)
VALUES
('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway'),
('Greasy Burger', 'Per Olsen', 'Gateveien 15', 'Sandnes', '4306', 'Norway'),
('Tasty Tee', 'Finn Egan', 'Streetroad 19B', 'Liverpool', 'L1 0AA', 'UK');
```

Where clause can be used in insert statement also:

```
INSERT INTO faculty SELECT id, name,dept_name,18000 FROM student where
dept_name=’CA’ and tot_cred>144;
```

This will first select the values from student table such as id, name ,dept_name which satisfies the where conditions and will insert those data’s into the faculty table along with 18000

The `UPDATE` statement is used to modify the existing records in a table.

UPDATE Syntax

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

DELETE Syntax

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

Delete All Records

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact:

```
DELETE FROM table_name;
```

Delete a Table

To delete the table completely, use the `DROP TABLE` statement:

```
DROP TABLE Customers;
```

Aggregate Function:

- `COUNT` counts how many rows are in a particular column.
- `SUM` adds together all the values in a particular column.
- `MIN` and `MAX` return the lowest and highest values in a particular column, respectively.
- `AVG` calculates the average of a group of selected values.

COUNT():

The COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

AVG() :

The AVG() function returns the average value of a numeric column.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

SUM() :

The `SUM()` function returns the total sum of a numeric column.

```
SELECT SUM(column_name) FROM table_name WHERE condition;
```

The SQL MIN() and MAX() Functions

The MIN() function returns the smallest value of the selected column.

The MAX() function returns the largest value of the selected column.

```
SELECT MIN(Column_name) FROM table_ name WHERE condition;

SELECT MAX(Column_name) FROM table_name WHERE condition;
```

```
SELECT MIN(Price) FROM Products;

SELECT MAX(Price) FROM Products;
```

Dayananda Sagar University Bengaluru

Dayananda Sagar
University

School of Engineering

Kudlu Gate, Hosur Road, Bengaluru 560068

DEPARTMENT OF COMPUTER APPLICATIONS

DATABASE SYSTEMS LAB

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:	22CA2406	**Credits**	:	01
Hours / Week	:	02 Hours	**Total Hours**	:	24 Hours
L–T–P	:	0–0–2			

Course Learning Objectives:

This Course will enable students to:

1. **Apply** fundamentals of data models, conceptualize and depict a database system using ER diagram.
2. **Apply** an effective relational database and write SQL queries to store data in database systems
3. **Apply** an effective relational database and write SQL queries to retrieve data from database systems
4. **Apply** constraints on fields to understand the importance of keys and other constraints
5. **Explain** basic database storage structures, access techniques and query processing.

List of Experiments

1. Draw E-R diagram and convert entities and relationships to relation table for a given scenario.
a. Two assignments shall be carried out i.e. consider two different scenarios

2. Write SQL query on various DDL programs. (5 query statements)

3. Write SQL query on various DML programs. (5 query statements)

4. Write SQL query on Aggregate functions (8-10 query statements)

5. Write a sql query using functions using Create()

6. Write a sql query using functions using Join()

7. Write a sql query using functions - Usage

8. Write a sql query using functions- User Defined Function

9. Write a sql query using functions -User Defined Function

10. Write a sql query using functions- User Defined Function

DEPARTMENT OF COMPUTER APPLICATIONS

Course Outcom e	**Description**	**Bloom's Taxonomy Level**
At the end of the course the student will be able to:		
1	**Apply** to Model an application's data requirements using conceptual modelling and design database schemas based on the conceptual model.	L3
2	**Formulate** solutions to a broad range of query problems using to store data in database	L5
3	**Formulate** solutions to a broad range of query problems using to retrieve data in database	L2
4	**Apply** constraints on fields to understand the importance of keys and other constraints	L3
5	**Explain** basic database storage structures, access techniques and query processing.	L2

****

DEPARTMENT OF COMPUTER APPLICATIONS

****