(1)	**GHRCEM , Practical Exam. 2025** **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Design and develop SQL statements for below problem statement. 1.Create Table: i) Teacher_ info{Teacher_ id, Teacher_ Name, Dept_ Name, Salary, Subject _taught} ,ii) Student_ info{Roll_ No, Student_ Name, Dept_ Name, Class, Result} 2. Insert 5 records in both above collections. 3. Find student name whose result is above 60% 4. Find student name whose name ends with "i" 5. Display name of teacher whose salary is between 20,000 to 50000. 6. Add "Join_ Date" field in Teacher_ info table. 7. Add "Address" field in Student_ info table. 8. Use "DROP" command 9. Find Name of students who lives in city "Pune" 10. Find Average salary of all teachers

```
CREATE DATABASE College_DB;

USE College_DB;


-- 1 Create tables

CREATE TABLE Teacher_info (

   Teacher_id INT PRIMARY KEY,

   Teacher_Name VARCHAR(50),

   Dept_Name VARCHAR(50),

   Salary DECIMAL(10,2),

   Subject_taught VARCHAR(50)

);


CREATE TABLE Student_info (

   Roll_No INT PRIMARY KEY,

   Student_Name VARCHAR(50),

   Dept_Name VARCHAR(50),

   Class VARCHAR(20),

   Result DECIMAL(5,2)

);


-- 2 Insert records

INSERT INTO Teacher_info VALUES

(1, 'Ramesh Patil', 'Computer', 45000, 'DBMS'),
```

```
(2, 'Sneha Kulkarni', 'IT', 30000, 'Networking'),
(3, 'Amit Deshmukh', 'Mechanical', 52000, 'Thermodynamics'),
(4, 'Pooja Joshi', 'ENTC', 25000, 'Digital Circuits'),
(5, 'Suresh Pawar', 'Civil', 18000, 'Structural Engg');

INSERT INTO Student_info VALUES
(1, 'Rahul Jadhav', 'Computer', 'SY', 75.5),
(2, 'Anjali Patil', 'IT', 'TY', 58.0),
(3, 'Rohit Joshi', 'Mechanical', 'FY', 62.3),
(4, 'Priya Nair', 'Civil', 'TY', 85.0),
(5, 'Sakshi More', 'ENTC', 'SY', 45.5);

-- 3 Find student name whose result is above 60%
SELECT Student_Name, Result
FROM Student_info
WHERE Result > 60;

-- 4 Find student name whose name ends with “i”
SELECT Student_Name
FROM Student_info
WHERE Student_Name LIKE '%i';

-- 5 Display name of teacher whose salary is between 20,000 to 50,000
SELECT Teacher_Name, Salary
FROM Teacher_info
WHERE Salary BETWEEN 20000 AND 50000;

--  6 Add “Join_Date” field in Teacher_info table
ALTER TABLE Teacher_info
ADD Join_Date DATE;

-- (Optional: Add values)
UPDATE Teacher_info SET Join_Date = '2020-06-15' WHERE Teacher_id = 1;
UPDATE Teacher_info SET Join_Date = '2019-08-01' WHERE Teacher_id = 2;
```

```
UPDATE Teacher_info SET Join_Date = '2021-02-10' WHERE Teacher_id = 3;
UPDATE Teacher_info SET Join_Date = '2022-01-20' WHERE Teacher_id = 4;
UPDATE Teacher_info SET Join_Date = '2018-11-05' WHERE Teacher_id = 5;

-- 7 Add “Address” field in Student_info table
ALTER TABLE Student_info
ADD Address VARCHAR(100);

-- (Optional: Add addresses)
UPDATE Student_info SET Address = 'Pune' WHERE Roll_No IN (1, 3);
UPDATE Student_info SET Address = 'Mumbai' WHERE Roll_No = 2;
UPDATE Student_info SET Address = 'Nashik' WHERE Roll_No = 4;
UPDATE Student_info SET Address = 'Pune' WHERE Roll_No = 5;

-- 8 Use “DROP” command
CREATE TABLE Temp_Table (id INT);
DROP TABLE Temp_Table;

-- 9 Find Name of students who live in city “Pune”
SELECT Student_Name, Address
FROM Student_info
WHERE Address = 'Pune';

-- 10 Find Average salary of all teachers
SELECT AVG(Salary) AS Average_Salary
FROM Teacher_info;
```

(2)	**GHRCEM, Practical Exam. 2025** **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Design and develop SQL DML statements for below problem statement. 1. Create Table: Teacher_ info{Teacher_ id, Teacher_ Name, Dept_ Name, Sal, status} 2. Insert below values in respective fields of 'Teacher_ info' collection (Pic001, Ravi, IT, 30000, A) (Pic002, Aarti, IT, 40000, A) (Pic003, Narendra , COMP, 20000, B) (Pic004, Swati, ENTC, 15000, C) (Pic005, Sampada, IT, 35000, A) (Pic006, Mihir, IT, 25000, B) (Pic007, Malati, ENTC, 36000, A) 3. Display all contents of 'Teacher_ info'. 4. Display information of teacher whose salary is 35000. 5. Display information of teacher whose status is not equal to C. 6. Display information of teacher whose status is either A or Salary is 15000. 7. Find teacher name whose name ends with letter 'i'. 8. Create index for Teacher_Name column. 9. Find Average salary of all teachers 10. Change Department name as "COMP" of teachers who are teaching in IT department.

```
CREATE DATABASE IInd;
USE IInd;

-- 1CREATE TABLE
CREATE TABLE Teacher_info (
   Teacher_id VARCHAR(10) PRIMARY KEY,
   Teacher_Name VARCHAR(50),
   Dept_Name VARCHAR(50),
   Sal DECIMAL(10,2),
   Status CHAR(1)
);

-- 2) Insert below values
INSERT INTO Teacher_info VALUES
('Pic001', 'Ravi', 'IT', 30000, 'A'),
('Pic002', 'Aarti', 'IT', 40000, 'A'),
```

```
('Pic003', 'Narendra', 'COMP', 20000, 'B'),
('Pic004', 'Swati', 'ENTC', 15000, 'C'),
('Pic005', 'Sampada', 'IT', 35000, 'A'),
('Pic006', 'Mihir', 'IT', 25000, 'B'),
('Pic007', 'Malati', 'ENTC', 36000, 'A');

--  3 Display all contents
SELECT * FROM Teacher_info;

-- 4. Display information of teacher whose salary is 35000.
SELECT *
FROM Teacher_info
WHERE Sal = 35000;

-- 5. Display information of teacher whose status is not equal to C
SELECT *
FROM Teacher_info
WHERE Status <> 'C';

-- 6. Display information of teacher whose status is either A or Salary is 15000.
SELECT *
FROM Teacher_info
WHERE Status = 'A' OR Sal = 15000;

-- 7. Find teacher name whose name ends with letter ‘i’
SELECT Teacher_Name
FROM Teacher_info
WHERE Teacher_Name LIKE '%i';

-- 8. Create index for Teacher_Name column.
CREATE INDEX idx_teacher_name
ON Teacher_info (Teacher_Name);

-- 9. Find Average salary of all teachers
```

```
SELECT AVG(Sal) AS Average_Salary
FROM Teacher_info;

-- 10. Change Department name as “COMP
SET SQL_SAFE_UPDATES = 0;

UPDATE Teacher_info
SET Dept_Name = 'COMP'
WHERE Dept_Name = 'IT';
```

(3)	**GHRCEM , Practical Exam. 2025** **Sub: DBMS Class: TY BTECH AI/AIML Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Design and develop SQL DML statements for below problem statement. 1 .Create Table: Teacher_ info{Teacher_ id, Teacher_ Name, Dept_ Name, Sal, status} 2. Insert below values in respective fields of 'Teacher_ info' collection (Pic001, Ravi, IT, 30000, A) (Pic002, Aarti, IT, 40000, A) (Pic003, Narendra, COMP, 20000, B) (Pic004, Swati, ENTC, 15000, C) (Pic005, Sampada, IT, 35000, A) (Pic006, Mihir, IT, 25000, B) (Pic007, Malati, ENTC, 36000, A) 3. Display all contents of 'Teacher_ info'. 4. Display information of teacher whose salary is greater than 20000. 5. Sort Teachers according to descending order of their salary. 6. Increment the salary of all teachers by 1000 whose status is 'A'. 7. Find teacher name whose name contains letter 'a'. 8. Create index on Teacher_Name column. 9. Find Teacher name who has Highest Salary. 10. Find name of teachers whose salary is greater than average salary of all teachers.

```
CREATE DATABASE IIIrd;
USE IIIrd;

-- 1) CREATE TABLE
CREATE TABLE Teacher_info (
   Teacher_id VARCHAR(10) PRIMARY KEY,
   Teacher_Name VARCHAR(50),
   Dept_Name VARCHAR(50),
   Sal DECIMAL(10,2),
   Status CHAR(1)
);

-- 2) Insert below values
INSERT INTO Teacher_info VALUES
('Pic001', 'Ravi', 'IT', 30000, 'A'),
('Pic002', 'Aarti', 'IT', 40000, 'A'),
('Pic003', 'Narendra', 'COMP', 20000, 'B'),
('Pic004', 'Swati', 'ENTC', 15000, 'C'),
```

```
('Pic005', 'Sampada', 'IT', 35000, 'A'),
('Pic006', 'Mihir', 'IT', 25000, 'B'),
('Pic007', 'Malati', 'ENTC', 36000, 'A');

-- 3) Display all contents
SELECT * FROM Teacher_info;

-- 4) Display information of teachers whose salary is greater than 20000
SELECT *
FROM Teacher_info
WHERE Sal > 20000;

-- 5) Sort teachers according to descending order of their salary
SELECT *
FROM Teacher_info
ORDER BY Sal DESC;

-- 6) Increment the salary of all teachers by 1000 whose status is ‘A’
SET SQL_SAFE_UPDATES = 0;

UPDATE Teacher_info
SET Sal = Sal + 1000
WHERE Status = 'A';

SET SQL_SAFE_UPDATES = 1;

-- 7) Find teacher name whose name contains letter ‘a’
SELECT Teacher_Name
FROM Teacher_info
WHERE Teacher_Name LIKE '%a%';

-- 8) Create index on Teacher_Name column
CREATE INDEX idx_teacher_name
ON Teacher_info (Teacher_Name);
```

```
-- 9) Find teacher name who has highest salary
SELECT Teacher_Name
FROM Teacher_info
WHERE Sal = (SELECT MAX(Sal) FROM Teacher_info);

-- 10) Find name of teachers whose salary is greater than average salary of all teachers
SELECT Teacher_Name, Sal
FROM Teacher_info
WHERE Sal > (SELECT AVG(Sal) FROM Teacher_info);
```

(4)	**GHRCEM , Practical Exam. 2025** **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Create table for following Diagram. Books Book_Name Author Price Publication_Year 1. Java Robert 150 2005 2. .Net John 180 2006 3. XML Robert,Peter 200 2003 4. XML Kevin 150 2004 Implement queries using SQL expression. 1. List the name of Books whose price is greater than 180. 2. List the name of Books in descending order of Price. 3. List the titles of books published Robert. 4. Find the average price of all Book. 5. For each book whose price is greater than the average price, return the title of the book price exceeds the average price. 6. Execute all aggregate functions on book table. 7. Update book price of book XML to 225. 8. Add new column name "Publisher" in above table and also add publisher for the same. 9. Apply any two string operations on column Book_Name. 10. Create view named "BookInfo" containing column Book name and author only.

```
CREATE DATABASE IVth;
USE IVth;

-- 1) CREATE TABLE
CREATE TABLE Books (
   Book_Name VARCHAR(50),
   Author VARCHAR(50),
   Price DECIMAL(10,2),
   Publication_Year INT
);

-- 2) Insert values into Books table
INSERT INTO Books VALUES
('Java', 'Robert', 150, 2005),
('.Net', 'John', 180, 2006),
('XML', 'Robert,Peter', 200, 2003),
```

```
('XML', 'Kevin', 150, 2004);

-- 3) List the name of Books whose price is greater than 180
SELECT Book_Name
FROM Books
WHERE Price > 180;

-- 4) List the name of Books in descending order of Price
SELECT Book_Name, Price
FROM Books
ORDER BY Price DESC;

-- 5) List the titles of books published by Robert
SELECT Book_Name
FROM Books
WHERE Author LIKE '%Robert%';

-- 6) Find the average price of all Books
SELECT AVG(Price) AS Average_Price
FROM Books;

-- 7) For each book whose price is greater than the average price,
--    return the title of the book and how much its price exceeds the average price
SELECT
   Book_Name,
   Price,
   Price - (SELECT AVG(Price) FROM Books) AS Exceeds_By
FROM Books
WHERE Price > (SELECT AVG(Price) FROM Books);

-- 8) Execute all aggregate functions on book table
SELECT
   COUNT(*) AS Total_Books,
   MAX(Price) AS Max_Price,
```

```
    MIN(Price) AS Min_Price,
    SUM(Price) AS Total_Price,
    AVG(Price) AS Average_Price
FROM Books;

-- 9) Update book price of book XML to 225
SET SQL_SAFE_UPDATES = 0;

UPDATE Books
SET Price = 225
WHERE Book_Name = 'XML';

SET SQL_SAFE_UPDATES = 1;

-- 10) Add new column “Publisher” in Books table
ALTER TABLE Books
ADD Publisher VARCHAR(50);

SET SQL_SAFE_UPDATES = 0;

-- 11) Update publisher for the existing books
UPDATE Books
SET Publisher = 'TechWorld'
WHERE Book_Name = 'Java';

UPDATE Books
SET Publisher = 'CodePress'
WHERE Book_Name = '.Net';

UPDATE Books
SET Publisher = 'BookHub'
WHERE Book_Name = 'XML';

-- 12) Apply any two string operations on column Book_Name
```

-- Example 1: Convert all Book names to upper case

```
SELECT UPPER(Book_Name) AS UpperCase_Name FROM Books;
```

-- Example 2: Display first 3 letters of each Book name

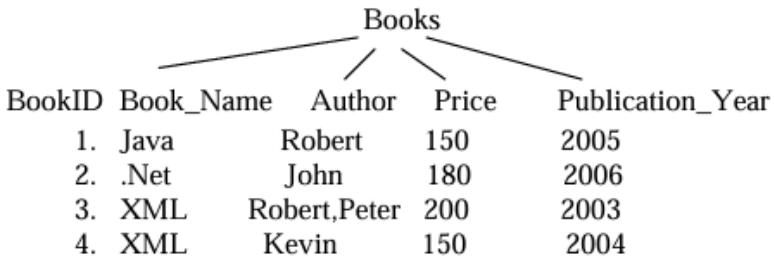
```
SELECT LEFT(Book_Name, 3) AS Short_Name FROM Books;
```

-- 13) Create view named “BookInfo” containing Book name and Author only

```
CREATE VIEW BookInfo AS
SELECT Book_Name, Author
FROM Books;
```

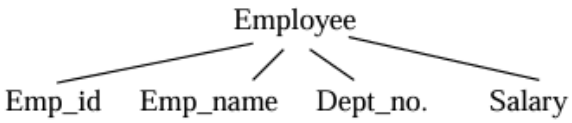
-- To display data from view

```
SELECT * FROM BookInfo;
```

(5)	**GHRCEM , Practical Exam. 2025** **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Create Table for following Diagram.  Implement Following queries using SQL expression. 1. Create book table with primary key as Bookid and Book_name. 2. Find all book titles published after 2003. 3. Find all books with more than 1 authors: 4. Compute a list of (author, title) pairs 5. List the Books order by its Name. 6. For each book whose price is greater than the average price, return the title of the book price exceeds the average price. 7. Update publication year as 2008 for book “Java”. 8. Fing highest and lowest price book name from table. 9. Add column “Date_of_publication” in book.

```
-- ✅ Create and use database
CREATE DATABASE Vth;
USE Vth;
-- ✅ 1) Create Book table with Primary Key as BookID and Book_Name
CREATE TABLE Books (
   BookID INT,
   Book_Name VARCHAR(50),
   Author VARCHAR(50),
   Price DECIMAL(10,2),
   Publication_Year INT,
   PRIMARY KEY (BookID, Book_Name)
);
-- ✅ 2) Insert records into Books table
INSERT INTO Books VALUES
(1, 'Java', 'Robert', 150, 2005),
(2, '.Net', 'John', 180, 2006),
(3, 'XML', 'Robert,Peter', 200, 2003),
(4, 'XML', 'Kevin', 150, 2004);
-- ✅ 3) Find all book titles published after 2003
SELECT Book_Name
FROM Books
WHERE Publication_Year > 2003;
-- ✅ 4) Find all books with more than 1 author
SELECT *
FROM Books
WHERE Author LIKE '%,%';
-- ✅ 5) Compute a list of (Author, Title) pairs
SELECT Author, Book_Name
FROM Books;
-- ✅ 6) List the books ordered by their Name
SELECT *
FROM Books
ORDER BY Book_Name;
```

```
-- ✅ 7) For each book whose price is greater than the average price,
--    return the title of the book and how much its price exceeds the average
SELECT
   Book_Name,
   Price,
   Price - (SELECT AVG(Price) FROM Books) AS Exceeds_By
FROM Books
WHERE Price > (SELECT AVG(Price) FROM Books);
-- ✅ 8) Update publication year as 2008 for book “Java”
SET SQL_SAFE_UPDATES = 0;
UPDATE Books
SET Publication_Year = 2008
WHERE Book_Name = 'Java';
SET SQL_SAFE_UPDATES = 1;
-- ✅ 9) Display book(s) with highest and lowest price
SELECT Book_Name, Price
FROM Books
WHERE Price = (SELECT MAX(Price) FROM Books)
UNION ALL
SELECT Book_Name, Price
FROM Books
WHERE Price = (SELECT MIN(Price) FROM Books);
-- ✅ 10) Add column “Date_of_publication” in Books table
ALTER TABLE Books
ADD Date_of_publication DATE;
```

(6)	GHRCEM , Practical Exam. 2025 Sub: DBMS Class: TY BTECH AI/AIML Duration : 2 Hrs
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Create table for following Diagram & Enter any 5 records.  Implement Following queries using SQL expression. 1. Find Average salary of all employees. 2. Find the Id and Name of Employees whose salary is Greater Than 10,000. 3. Find the employees with Highest salary. 4. Compute the list of Emp_name, Dept_no. order by Dept_no. 5. Compute the List of Emp_name, salary. 6. Count the no of employees department wise. 7. Find the name of employees whose salary is greater than the average salary of all employees.

```
-- Step 0: Create and Use Database
CREATE DATABASE EmployeeDB;
USE EmployeeDB;

-- Table creation
CREATE TABLE Employee (
    Emp_id INT PRIMARY KEY,
    Emp_name VARCHAR(50),
    Dept_no INT,
    Salary INT
);

-- Insert records
INSERT INTO Employee VALUES (101, 'Rahul', 1, 12000);
INSERT INTO Employee VALUES (102, 'Priya', 2, 9500);
INSERT INTO Employee VALUES (103, 'Amit', 1, 15000);
INSERT INTO Employee VALUES (104, 'Neha', 3, 8000);
INSERT INTO Employee VALUES (105, 'Sanjay', 2, 13500);

-- 1. Find Average salary of all employees.
SELECT AVG(Salary) AS Average_Salary
FROM Employee;
```

```
-- 2. Find the Id and Name of Employees whose salary is Greater Than 10,000.
SELECT Emp_id, Emp_name
FROM Employee
WHERE Salary > 10000;

-- 3. Find the employees with Highest salary.
SELECT *
FROM Employee
WHERE Salary = (SELECT MAX(Salary) FROM Employee);

-- 4. Compute the list of Emp_name, Dept_no. order by Dept_no.
SELECT Emp_name, Dept_no
FROM Employee
ORDER BY Dept_no;

-- 5. Compute the List of Emp_name, salary.
SELECT Emp_name, Salary
FROM Employee;

-- 6. Count the no of employees department wise.
SELECT Dept_no, COUNT(*) AS Employee_Count
FROM Employee
GROUP BY Dept_no;

-- 7. Find the name of employees whose salary is greater than the average salary of all employees.
SELECT Emp_name
FROM Employee
WHERE Salary > (SELECT AVG(Salary) FROM Employee);
```

(7)	**GHRCEM , Practical Exam. 2025** **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
.	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Execute DDL statements which demonstrate the use of views. Try to update the base table using its corresponding view. Also consider restrictions on updatable views .

```
-- Step 0: Create and Use Database
CREATE DATABASE View_DB;
USE View_DB;

-- Step 1: Create the Employee Table
CREATE TABLE Employee (
Emp_id INT PRIMARY KEY,
Emp_name VARCHAR(50),
Dept_no INT,
Salary INT
);

-- Step 2: Insert Sample Data
INSERT INTO Employee VALUES (101, 'Rahul', 1, 12000);
INSERT INTO Employee VALUES (102, 'Priya', 2, 9500);
INSERT INTO Employee VALUES (103, 'Amit', 1, 15000);
INSERT INTO Employee VALUES (104, 'Neha', 3, 8000);
INSERT INTO Employee VALUES (105, 'Sanjay', 2, 13500);

-- Step 3: Create a View to Show Only Employees with Salary Above 10,000
CREATE VIEW HighSalaryEmployees AS
SELECT Emp_id, Emp_name, Salary
FROM Employee
WHERE Salary > 10000;

-- Step 4: Query the View
```

```
SELECT * FROM HighSalaryEmployees;

-- Step 5: A empt to Update the Base Table Using the View
UPDATE HighSalaryEmployees
SET Salary = 16000
WHERE Emp_id = 101;

-- Step 6: Check the Result in Base Table
SELECT * FROM Employee WHERE Emp_id = 101;

-- Step 7: Demonstrate Restric on on an Updatable View
-- This view is NOT updatable because of the aggrega on func on (AVG)
CREATE VIEW DeptwiseAvgSalary AS
SELECT Dept_no, AVG(Salary) AS Avg_Salary
FROM Employee
GROUP BY Dept_no;

-- This will FAIL: trying to update a non-updatable view
UPDATE DeptwiseAvgSalary
SET Avg_Salary = 14000
WHERE Dept_no = 1;
```

(8)	**GHRCEM , Practical Exam. 2025** **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.
	Implement following Queries in SQL database. 1. Create Teacher dB. 2. Create table: Teacher_info{Teacher_id, Teacher_Name, Dept_Name, Sal, status} 3. Insert below values in respective fields of 'Teacher_info' collection (Pic001, Ravi, IT, 30000, A) (Pic002, Aarti, IT, 40000, A) (Pic003, Narendra, COMP, 20000, B) (Pic004, Swati, ENTC, 15000, C) (Pic005, Sampada, IT, 35000, A) (Pic006, Mihir, IT, 25000, B) (Pic007, Malati, ENTC, 36000, A) 4. Display all contents of 'Teacher_info'. 5. Display information of teacher whose salary is 35000. 6. Update Salary 25000 of teacher whose id is Pic004. 7. Display information of teacher whose status is either A and Salary is 15000. 8. Insert column Teacher_phone as new column. 9. Enter 2 phone no for any one row of table. 10. Create Index on teacher_Name column.

```
-- 1. Create Teacher database (database creation syntax depends on the system)
CREATE DATABASE TeacherDB;

-- Use the new database
USE TeacherDB;

-- 2. Create table: Teacher_info
CREATE TABLE Teacher_info (
Teacher_id VARCHAR(10) PRIMARY KEY,
Teacher_Name VARCHAR(50),
Dept_Name VARCHAR(20),
Sal INT,
status CHAR(1)
);

-- 3. Insert the given values into Teacher_info
INSERT INTO Teacher_info VALUES ('Pic001', 'Ravi', 'IT', 30000, 'A');
INSERT INTO Teacher_info VALUES ('Pic002', 'Aar ', 'IT', 40000, 'A');
```

```
INSERT INTO Teacher_info VALUES ('Pic003', 'Narendra', 'COMP', 20000, 'B');
INSERT INTO Teacher_info VALUES ('Pic004', 'Swa ', 'ENTC', 15000, 'C');
INSERT INTO Teacher_info VALUES ('Pic005', 'Sampada', 'IT', 35000, 'A');
INSERT INTO Teacher_info VALUES ('Pic006', 'Mihir', 'IT', 25000, 'B');
INSERT INTO Teacher_info VALUES ('Pic007', 'Mala ', 'ENTC', 36000, 'A');

-- 4. Display all contents of Teacher_info
SELECT * FROM Teacher_info;

-- 5. Display information of teacher whose salary is 35000
SELECT * FROM Teacher_info WHERE Sal = 35000;

-- 6. Update Salary to 25000 for teacher whose id is Pic004
UPDATE Teacher_info
SET Sal = 25000
WHERE Teacher_id = 'Pic004';

-- 7. Display information of teacher whose status is 'A' and Salary is 15000
SELECT * FROM Teacher_info
WHERE status = 'A' AND Sal = 15000;

-- 8. Insert column Teacher_phone as new column
ALTER TABLE Teacher_info
ADD Teacher_phone VARCHAR(20);

-- 9. Enter 2 phone numbers for any one row (use comma separation or array/json if supported)
UPDATE Teacher_info
SET Teacher_phone = '9999999999,8888888888'
WHERE Teacher_id = 'Pic001';

-- 10. Create Index on teacher_Name column
CREATE INDEX idx_teacher_name
ON Teacher_info(Teacher_Name);
```

(9)

GHRCEM , Practical Exam. 2025

Sub: DBMS **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**

Implement following Queries in SQL database.

1. Create the database as student.
2. Create table stud as Roll. No., Name, Batch,marks, Address
3. Insert all the rows as documents into the tables.

Roll No.	Name	Batch	Marks	Address
1	Anita	T1	90	Pune
2	Om	T2	85	Nasik
3	Mini	T1	80	Mumbai
4	Rahul	T1	83	Raigad
5	Reena	T2	87	Pune

4. Select Roll No. and Name of students belong to city Pune?
5. Display the entire student from T1 batch.
6. Update marks by 5 of roll no 4.
7. Add column as Phone No. into table?
8. Insert one row in table with 2 phone numbers?

```
-- 1. Create the database as student
CREATE DATABASE student;

-- Use the new database
USE student;

-- 2. Create table stud as Roll_No, Name, Batch, Marks, Address
CREATE TABLE stud (
    Roll_No INT PRIMARY KEY,
    Name VARCHAR(50),
    Batch VARCHAR(10),
    Marks INT,
    Address VARCHAR(50)
);

-- 3. Insert all the rows as documents into the table
INSERT INTO stud VALUES (1, 'Anita', 'T1', 90, 'Pune');
INSERT INTO stud VALUES (2, 'Om', 'T2', 85, 'Nasik');
INSERT INTO stud VALUES (3, 'Mini', 'T1', 80, 'Mumbai');
INSERT INTO stud VALUES (4, 'Rahul', 'T1', 83, 'Raigad');
```

```
INSERT INTO stud VALUES (5, 'Reena', 'T2', 87, 'Pune');

-- 4. Select Roll No. and Name of students belong to city Pune
SELECT Roll_No, Name
FROM stud
WHERE Address = 'Pune';

-- 5. Display the entire student from T1 batch
SELECT *
FROM stud
WHERE Batch = 'T1';

-- 6. Update marks by 5 of roll no 4
UPDATE stud
SET Marks = Marks + 5
WHERE Roll_No = 4;

-- 7. Add column as Phone No. into table
ALTER TABLE stud
ADD Phone_No VARCHAR(50);

-- 8. Insert one row in table with 2 phone numbers (comma separated)
INSERT INTO stud
VALUES (6, 'Vijay', 'T2', 88, 'Aurangabad', '9999988888,8888877777');
```

(11)

GHRCEM , Practical Exam. 2025

Sub: DBMS **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**

N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail
2. Assume suitable data if necessary.

Implement following Queries in SQL database.

1. Create Customer table as Cust_id, Cust_name, Address, City, State, Zipcode.
2. Insert following items into table:

Cust_id	Cust_name	Address	City	State	Zipcode
100	Ravi	Sinhgad Road	Pune	Maharashtra	411041
101	Sanket	Wagholi	PUNE	Maharashtra	412207
102	Poonam	Kharadi	Pune	Maharashtra	411014
104	Madhuri	swargate	Pune	Maharashtra	411041

3. Display all inserted items form customer.
4. Display details of customer whose customer id is 103 and name “sanket”
5. Give the address and zipcode of customer whose customer id is 103 and name “Rajesh Kumar.
6. Give the city of customer whose customer id is 102.
7. List the customer id,customer name and city of customer whose customer id is 100,101,104.
8. Display all details of customer whose who stays at Wagholi.
9. Update customer table by adding one attribute Product.
10. Delete customer with customer id 104 and Name Madhuri.

```
-- Create database and use it
CREATE DATABASE customer_db;


-- 1) Create Customer table
CREATE TABLE Customer (
  Cust_id INT PRIMARY KEY,
  Cust_name VARCHAR(50),
  Address VARCHAR(100),
  City VARCHAR(50),
  State VARCHAR(50),
  Zipcode INT
);


-- 2) Insert records
INSERT INTO Customer VALUES
(100, 'Ravi', 'Sinhgad Road', 'Pune', 'Maharashtra', 411041),
```

```
(101, 'Sanket', 'Wagholi', 'Pune', 'Maharashtra', 412207),
(102, 'Poonam', 'Kharadi', 'Pune', 'Maharashtra', 411014),
(104, 'Madhuri', 'Swargate', 'Pune', 'Maharashtra', 411041);

-- 3) Display all inserted items
SELECT * FROM Customer;


-- 4) Display details of customer whose id=103 and name='Sanket'
SELECT * FROM Customer
WHERE Cust_id=103 AND Cust_name='Sanket';


-- 5)Address and Zipcode of customer whose id=103 and name='Rajesh Kumar'
SELECT Address, Zipcode FROM Customer
WHERE Cust_id=103 AND Cust_name='Rajesh Kumar';


-- 6)City of customer whose id=102
SELECT City FROM Customer WHERE Cust_id=102;

-- 7)List id, name and city of customers with id 100, 101, 104
SELECT Cust_id, Cust_name, City
FROM Customer
WHERE Cust_id IN (100, 101, 104);

-- 8)Display customers who stay at Wagholi
SELECT * FROM Customer WHERE Address='Wagholi';

-- 9) Add new column Product
ALTER TABLE Customer ADD Product VARCHAR(50);
-- 10)Delete customer with id=104 and name='Madhuri'
DELETE FROM Customer
WHERE Cust_id=104 AND Cust_name='Madhuri';
```

(13)	GHRCEM , Practical Exam. 2025 Sub: DBMS Class: TY BTECH AI/AIML Duration : 2 Hrs
	Consider the following tables with appropriate data types and Constraints. Sales_order (ordNo, ordDate, clientNo) Client (clientNo, ClientName, addr) Constraints: - Primary Key, ordDate should not be NULL 1. Add 5 rows in each table. 2. Add column amount into Sales_order table with data type int. 3. Delete the details of the clients whose names start with 'A' character. 4. Delete sales order details of client whose name is "Patil" and order date is "09/08/2023". 5. Delete all sales_record having order date is before '10 /02/2018'. 6. Display date wise sales_order given by clients. 7. Update the address of client to "Pimpri" whose name is 'Mr. Roy'

```
-- Create database and use it
CREATE DATABASE sales_db;
USE sales_db;

-- (1) Create Client table
CREATE TABLE Client (
  clientNo INT PRIMARY KEY,
  ClientName VARCHAR(50),
  addr VARCHAR(100)
);

-- (1) Create Sales_order table
CREATE TABLE Sales_order (
  ordNo INT PRIMARY KEY,
  ordDate DATE NOT NULL,
  clientNo INT,
  FOREIGN KEY (clientNo) REFERENCES Client(clientNo)
);

-- (1) Insert 5 rows in each table
INSERT INTO Client VALUES
```

```
(1, 'Amit', 'Pune'),
(2, 'Patil', 'Mumbai'),
(3, 'Mr. Roy', 'Delhi'),
(4, 'Anand', 'Nashik'),
(5, 'Kiran', 'Pimpri');

INSERT INTO Sales_order VALUES
(101, '2023-08-09', 2),
(102, '2023-08-09', 1),
(103, '2023-02-10', 4),
(104, '2018-02-05', 3),
(105, '2018-01-10', 5);

-- (2) Add column amount into Sales_order table
ALTER TABLE Sales_order ADD amount INT;

-- (3) Delete details of clients whose names start with ‘A’
SET SQL_SAFE_UPDATES = 0;

-- First delete from Sales_order
DELETE FROM Sales_order
WHERE clientNo IN (SELECT clientNo FROM Client WHERE ClientName LIKE 'A%');

-- Then delete from Client
DELETE FROM Client WHERE ClientName LIKE 'A%';

-- (4) Delete sales order details of client “Patil” and order date “09/08/2023”
DELETE FROM Sales_order
WHERE clientNo IN (SELECT clientNo FROM Client WHERE ClientName='Patil')
AND ordDate='2023-08-09';
```

```
-- (5) Delete all sales_record having order date before ‘10/02/2018’
DELETE FROM Sales_order WHERE ordDate < '2018-02-10';

-- (6) Display date wise sales_order given by clients
SELECT ordDate, clientNo FROM Sales_order ORDER BY ordDate;

-- (7) Update address of client to “Pimpri” whose name is ‘Mr. Roy’
UPDATE Client SET addr='Pimpri' WHERE ClientName='Mr. Roy';

SET SQL_SAFE_UPDATES = 1;
```

(14)	GHRCEM , Practical Exam. 2025 Sub: DBMS Class: TY BTECH AI/AIML Duration : 2 Hrs
	Consider the following entities and their relationships. Create a RDB in 3 NF with appropriate data types and Constraints. Hospital (hno ,hname , city, Est_year, addr, dno) Dor (dno , dname , addr, Speciality) Constraints: - Primary Key, Est_year should be greater than 1990. 1. Create above table and insert any 5 records in it. 2. Delete addr column from Hospital table. 3. Display dor name, Hospital name and specialty of dors from "Pune City". 4. Display the names of the hospitals which are located at "Pimpri" city. 5. Display the specialty of the dors who are working in "Ruby" hospital.

```
-- 1) Create database and use it
CREATE DATABASE hospital_db;
USE hospital_db;


-- 1) Create Doctor table
CREATE TABLE Doctor (
  dno INT PRIMARY KEY,
  dname VARCHAR(50),
  addr VARCHAR(100),
  speciality VARCHAR(50)
);


-- 1) Create Hospital table
CREATE TABLE Hospital (
  hno INT PRIMARY KEY,
  hname VARCHAR(50),
  city VARCHAR(50),
  Est_year INT CHECK (Est_year > 1990),
  addr VARCHAR(100),
  dno INT,
  FOREIGN KEY (dno) REFERENCES Doctor(dno)
);
```

```
-- 1) Insert 5 records into Doctor table
INSERT INTO Doctor VALUES
(1, 'Dr. Mehta', 'Pune', 'Cardiology'),
(2, 'Dr. Roy', 'Mumbai', 'Neurology'),
(3, 'Dr. Patil', 'Pimpri', 'Orthopedic'),
(4, 'Dr. Deshmukh', 'Pune', 'ENT'),
(5, 'Dr. Rao', 'Nashik', 'Dermatology');

-- 1) Insert 5 records into Hospital table
INSERT INTO Hospital VALUES
(101, 'Ruby', 'Pune', 2000, 'Shivajinagar', 1),
(102, 'Aditya', 'Pimpri', 2005, 'Nehru Nagar', 3),
(103, 'Yashoda', 'Mumbai', 2010, 'Andheri', 2),
(104, 'SaiCare', 'Pune', 1995, 'Viman Nagar', 4),
(105, 'Wellness', 'Nashik', 2001, 'Main Road', 5);

SELECT * FROM Doctor;
SELECT * FROM Hospital;

-- 2) Delete addr column from Hospital table
ALTER TABLE Hospital DROP COLUMN addr;

-- 3) Display doctor name, hospital name and specialty of doctors from “Pune City”
SELECT d.dname AS Doctor_Name, h.hname AS Hospital_Name, d.speciality AS Speciality
FROM Doctor d
JOIN Hospital h ON d.dno = h.dno
WHERE h.city = 'Pune';

-- 4) Display names of the hospitals located at “Pimpri” city
SELECT hname AS Hospital_Name
FROM Hospital
WHERE city = 'Pimpri';

-- 5) Display specialty of the doctors who are working in “Ruby” hospital
```

```
SELECT d.speciality AS Speciality
FROM Doctor d
JOIN Hospital h ON d.dno = h.dno
WHERE h.hname = 'Ruby';
```

(15)	GHRCEM , Practical Exam. 2025 Sub: DBMS Class: TY BTECH AI/AIML Duration : 2 Hrs

Create following Pateint and Bed table

PCODE	NAME	ADDR	DISEASE
11	Raghav	pimple gurav	listeria
12	Abhay	pune	norovirus
13	Mr.Roy	mumbai	cholera
14	Sachin	pimple gurav	dengue
15	Priya	nashik	listeria

Bed Table

BNO	RNO	LOC	PCODE
1	105	3rd floor	11
2	102	2nd floor	12
3	103	4th floor	13
4	104	1st floor	11
5	105	3rd floor	14
6	106	2nd floor	15

Implement following queries:

1. Display the details of patients who are from “Pimple Gurav”
2. Delete the details of patient whose Bed_No is 1 and RoomNo is 105.
3. Display unique disease of patients.
4. Find the name of patient whos name contains letter “a”
5. Find total number of beds.

```
CREATE DATABASE IF NOT EXISTS dbms_exam;

USE dbms_exam;


-- Create Patient table

CREATE TABLE Patient (

  PCODE INT PRIMARY KEY,
```

```
  NAME VARCHAR(100) NOT NULL,
  ADDR VARCHAR(100),
  DISEASE VARCHAR(100)
);

-- Create Bed table (BNO primary key, PCODE references Patient)
CREATE TABLE Bed (
  BNO INT PRIMARY KEY,
  RNO INT,
  LOC VARCHAR(100),
  PCODE INT,
  FOREIGN KEY (PCODE) REFERENCES Patient(PCODE)
);

-- Insert patients
INSERT INTO Patient (PCODE, NAME, ADDR, DISEASE) VALUES
(11, 'Raghav', 'pimple gurav', 'listeria'),
(12, 'Abhay',  'pune',         'norovirus'),
(13, 'Mr.Roy', 'mumbai',       'cholera'),
(14, 'Sachin', 'pimple gurav', 'dengue'),
(15, 'Priya',  'nashik',       'listeria');

-- Insert beds
INSERT INTO Bed (BNO, RNO, LOC, PCODE) VALUES
(1,  105, '3rd floor', 11),
(2,  102, '2nd floor', 12),
(3,  103, '4th floor', 13),
(4,  104, '1st floor', 11),
(5,  105, '3rd floor', 14),
(6,  106, '2nd floor', 15);

SELECT * FROM Patient;
SELECT * FROM Bed;
```

```
SELECT * FROM Patient
WHERE LOWER(ADDR) = 'pimple gurav';

SELECT PCODE FROM Bed
WHERE BNO = 1 AND RNO = 105;
DELETE FROM Bed
WHERE BNO = 1 AND RNO = 105;

SELECT DISTINCT DISEASE FROM Patient;

SELECT NAME FROM Patient
WHERE LOWER(NAME) LIKE '%a%';

SELECT COUNT(*) AS total_beds FROM Bed;
```

16	GHRCEM , Practical Exam. 2025 Sub: DBMS Class: TY BTECH AI/AIML Duration : 2 Hrs

Create following customer and loan table

CNO	CNAME	ADDR	CITY
101	Dhiraj	kharadi	pune
102	Patil	kalptaru	pimpri
103	Abhay	west	pimpri
104	Raghav	rt	nashik
105	Dhanu	bvh	pune

LNO	LAMT	CNO
1	120000	101
2	100000	102
3	30000	103
4	120	104
5	1000000	105

Implement following queries:

1. **Add Phone_No column in customer table with data type int.**
2. **Delete the details of customer whose loan_amt<1000.**
3. **Find details of all customers who are staying at Nashik**
4. **List all customers whose name starts with 'D' character.**
5. **Find loan taken by customer "102"**
6. **Finad total loan given by bank.**

```
-- Step 1: Create and use database
CREATE DATABASE IF NOT EXISTS dbms_exam1;
USE dbms_exam1;

-- Step 2: Create Customer table
CREATE TABLE Customer (
```

```
  CNO INT PRIMARY KEY,
  CNAME VARCHAR(100) NOT NULL,
  ADDR VARCHAR(100),
  CITY VARCHAR(100)
);

-- Step 3: Create Loan table
CREATE TABLE Loan (
  LNO INT PRIMARY KEY,
  LAMT INT,
  CNO INT,
  FOREIGN KEY (CNO) REFERENCES Customer(CNO)
);

-- Step 4: Insert sample records into Customer
INSERT INTO Customer (CNO, CNAME, ADDR, CITY) VALUES
(101, 'Dhiraj', 'kharadi', 'pune'),
(102, 'Patil', 'kalptaru', 'pimpri'),
(103, 'Abhay', 'west', 'pimpri'),
(104, 'Raghav', 'rt', 'nashik'),
(105, 'Dhanu', 'bvh', 'pune');

-- Step 5: Insert records into Loan
INSERT INTO Loan (LNO, LAMT, CNO) VALUES
(1, 120000, 101),
(2, 100000, 102),
(3, 30000, 103),
(4, 120, 104),
(5, 1000000, 105);

-- Step 6: Verify inserted data
SELECT * FROM Customer;
SELECT * FROM Loan;
```

```
-- ----------------------------------------
-- REQUIRED QUERIES
-- ----------------------------------------

-- Q1. Add Phone_No column in customer table with data type int
ALTER TABLE Customer ADD Phone_No BIGINT;

-- Q2. Delete details of customer whose loan_amt < 1000

-- First delete from Loan (child)
DELETE FROM Loan
WHERE LAMT < 1000;

-- Then delete from Customer (parent)
DELETE FROM Customer
WHERE CNO NOT IN (SELECT CNO FROM Loan);


-- (Also delete corresponding loan record to maintain referential integrity)
DELETE FROM Loan WHERE LAMT < 1000;

-- Q3. Find details of all customers who are staying at Nashik
SELECT * FROM Customer
WHERE LOWER(CITY) = 'nashik';

-- Q4. List all customers whose name starts with 'D'
SELECT * FROM Customer
WHERE CNAME LIKE 'D%';

-- Q5. Find loan taken by customer “102”
SELECT * FROM Loan
WHERE CNO = 102;
-- Q6. Find total loan given by bank
SELECT SUM(LAMT) AS Total_Loan_Amount FROM Loan;
```

(17)	GHRCEM , Practical Exam. 2025 **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 hrs**
	Create following table and insert 5 records in it. **Emp(eno ,ename ,designation ,salary, Date_Of_Joining)** **Dept(dno,dname ,loc)** **The relationship between Dept & Emp is one-to-many. Constraints: - Primary Key, ename should not be NULL, salary must be greater than 0.** **Implement following queries:** 1. **Find maximum salary of employee.** 2. **Find name of employee working in "Sales" department.** 3. **Delete the details of Employee whose designation is 'Manager'.** 4. **Implement any 3 aggregate function on salary column** 5. Create index on column eno.

-- Step 1: Create / use database

CREATE DATABASE IF NOT EXISTS dbms_exam;

USE dbms_exam;

-- Step 2: Drop old tables if they already exist (safety)

DROP TABLE IF EXISTS Emp;

```
DROP TABLE IF EXISTS Dept;

-- Step 3: Create Department table
CREATE TABLE Dept (
  DNO INT PRIMARY KEY,
  DNAME VARCHAR(50) NOT NULL,
  LOC VARCHAR(50)
);

-- Step 4: Create Employee table
CREATE TABLE Emp (
  ENO INT PRIMARY KEY,
  ENAME VARCHAR(50) NOT NULL,
  DESIGNATION VARCHAR(50),
  SALARY DECIMAL(10,2) CHECK (SALARY > 0),
  DATE_OF_JOINING DATE,
  DNO INT,
  FOREIGN KEY (DNO) REFERENCES Dept(DNO)
);

-- Step 5: Insert 5 records into Dept
INSERT INTO Dept (DNO, DNAME, LOC) VALUES
(10, 'Sales', 'Pune'),
(20, 'HR', 'Mumbai'),
(30, 'IT', 'Nashik'),
(40, 'Finance', 'Pune'),
(50, 'Admin', 'Delhi');

-- Step 6: Insert 5 records into Emp
INSERT INTO Emp (ENO, ENAME, DESIGNATION, SALARY, DATE_OF_JOINING, DNO) VALUES
(101, 'Raghav', 'Manager', 55000, '2020-01-15', 10),
(102, 'Abhay', 'Salesman', 30000, '2021-03-10', 10),
(103, 'Priya', 'Clerk', 25000, '2022-02-01', 20),
(104, 'Sachin', 'Developer', 45000, '2023-06-05', 30),
```

```
(105, 'Dhanu', 'Analyst', 40000, '2022-11-11', 40);

-- Step 7: Verify inserted data
SELECT * FROM Dept;
SELECT * FROM Emp;

-- ----------------------------------------
-- REQUIRED QUERIES
-- ----------------------------------------

-- Q1. Find maximum salary of employee
SELECT MAX(SALARY) AS Max_Salary FROM Emp;

-- Q2. Find name of employee working in “Sales” department
SELECT E.ENAME, D.DNAME
FROM Emp E
JOIN Dept D ON E.DNO = D.DNO
WHERE D.DNAME = 'Sales';

-- Q3. Delete the details of Employee whose designation is ‘Manager’
DELETE FROM Emp
WHERE LOWER(DESIGNATION) = 'manager';

-- Q4. Implement any 3 aggregate functions on salary column
SELECT
  COUNT(*) AS Total_Employees,
  AVG(SALARY) AS Average_Salary,
  SUM(SALARY) AS Total_Salary
FROM Emp;

-- Q5. Create index on column ENO
CREATE INDEX idx_emp_eno ON Emp(ENO);
-- Verify index creation (optional)
SHOW INDEX FROM Emp;
```

18

GHRCEM , Practical Exam. 2025

Sub: DBMS **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**

1. Consider the following table:

Order:

Order_ID	CustomerID	OrderDate
10308	2	2018-09-18
10309	3	2018-09-19
10310	4	2018-09-20

Customer

CustomerID	CustomerName	ContactNo	State
1	Vijay	9873456782	Maharashtra
2	Sanket	1234567890	Maharashtra
3	Abin	9812345607	Kerla
4	Ram	1209876543	Chennai

1. Create Above order and Customer table.
2. Perform Inner Join, Left Outer Join, Right Outer Join.
3. Find Name, and Order date of Customer who live in state Maharashtra.
4. Add column contact no in customer table and insert record for all customers.
5. Find name of states which start with letter 'M'.
6. Find Orderid and Customer name of Customers who live in state "Maharashtra"
7. Display customer table state wise in descending order.
8. Find the contact no which consist of letter "45".

```
-- Step 1: Create / use database
CREATE DATABASE IF NOT EXISTS dbms_exam;
USE dbms_exam;

-- Step 2: Clean old data if rerun
DROP TABLE IF EXISTS Orders;
DROP TABLE IF EXISTS Customer;

-- Step 3: Create Customer table
CREATE TABLE Customer (
  CustomerID INT PRIMARY KEY,
  CustomerName VARCHAR(50) NOT NULL,
```

```
  ContactNo BIGINT,
  State VARCHAR(50)
);

-- Step 4: Create Order table
CREATE TABLE Orders (
  Order_ID INT PRIMARY KEY,
  CustomerID INT,
  OrderDate DATE,
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

-- Step 5: Insert data into Customer
INSERT INTO Customer (CustomerID, CustomerName, ContactNo, State) VALUES
(1, 'Vijay', 9873456782, 'Maharashtra'),
(2, 'Sanket', 1234567890, 'Maharashtra'),
(3, 'Abin', 9812345607, 'Kerala'),
(4, 'Ram', 1209876543, 'Chennai');

-- Step 6: Insert data into Orders
INSERT INTO Orders (Order_ID, CustomerID, OrderDate) VALUES
(10308, 2, '2018-09-18'),
(10309, 3, '2018-09-19'),
(10310, 4, '2018-09-20');

-- Step 7: Verify tables
SELECT * FROM Customer;
SELECT * FROM Orders;

-- ----------------------------------------
-- REQUIRED QUERIES
-- ----------------------------------------

-- Q1. Perform INNER JOIN (Customers having orders)
```

```
SELECT C.CustomerName, O.Order_ID, O.OrderDate
FROM Customer C
INNER JOIN Orders O ON C.CustomerID = O.CustomerID;

-- Q2. Perform LEFT OUTER JOIN (All customers + matching orders)
SELECT C.CustomerName, O.Order_ID, O.OrderDate
FROM Customer C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID;

-- Q3. Perform RIGHT OUTER JOIN (All orders + matching customers)
SELECT C.CustomerName, O.Order_ID, O.OrderDate
FROM Customer C
RIGHT JOIN Orders O ON C.CustomerID = O.CustomerID;

-- Q4. Find Name and Order date of Customers who live in state 'Maharashtra'
SELECT C.CustomerName, O.OrderDate
FROM Customer C
JOIN Orders O ON C.CustomerID = O.CustomerID
WHERE C.State = 'Maharashtra';

-- Q5. Add column contact_no (already exists; if missing add it)
-- ALTER TABLE Customer ADD ContactNo BIGINT;

-- Q6. Find names of states which start with letter ‘M’
SELECT DISTINCT State
FROM Customer
WHERE State LIKE 'M%';

-- Q7. Find Order_ID and CustomerName of customers who live in state “Maharashtra”
SELECT O.Order_ID, C.CustomerName
FROM Orders O
JOIN Customer C ON O.CustomerID = C.CustomerID
WHERE C.State = 'Maharashtra';
```

```
-- Q8. Display customer table state-wise in descending order
SELECT * FROM Customer
ORDER BY State DESC;

-- Q9. Find contact numbers which contain '45'
SELECT * FROM Customer
WHERE ContactNo LIKE '%45%';
```

19	GHRCEM , Practical Exam. 2025		
	Sub: DBMS	**Class: TY BTECH AI/AIML**	**Duration : 2 Hrs**
	Write and execute suitable database triggers .Consider row level and statement level triggers.		

```
-- Step 1: Create and use database
CREATE DATABASE IF NOT EXISTS dbms_exam;
USE dbms_exam;

-- Step 2: Drop old tables & triggers if exist
DROP TABLE IF EXISTS Employee;
DROP TABLE IF EXISTS Emp_Audit;
DROP TRIGGER IF EXISTS before_emp_insert;
DROP TRIGGER IF EXISTS after_emp_delete;

-- Step 3: Create Employee table
CREATE TABLE Employee (
  EmpID INT PRIMARY KEY AUTO_INCREMENT,
  EmpName VARCHAR(50) NOT NULL,
  Dept VARCHAR(50),
  Salary DECIMAL(10,2)
);

-- Step 4: Create Audit table to record trigger actions
CREATE TABLE Emp_Audit (
  Audit_ID INT PRIMARY KEY AUTO_INCREMENT,
  Action_Type VARCHAR(50),
  EmpName VARCHAR(50),
  Action_Time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- ----------------------------------------
-- CREATE TRIGGERS
-- ----------------------------------------
```

```
-- Q1. Row-level BEFORE INSERT trigger
DELIMITER $$
CREATE TRIGGER before_emp_insert
BEFORE INSERT ON Employee
FOR EACH ROW
BEGIN
  -- Check salary > 0 before insert
  IF NEW.Salary <= 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Salary must be greater than zero';
  END IF;
END$$
DELIMITER ;

-- Q2. Statement-level AFTER DELETE trigger
DELIMITER $$
CREATE TRIGGER after_emp_delete
AFTER DELETE ON Employee
FOR EACH ROW
BEGIN
  -- Log deleted employee in audit table
  INSERT INTO Emp_Audit (Action_Type, EmpName)
  VALUES ('Employee Deleted', OLD.EmpName);
END$$
DELIMITER ;

-- ----------------------------------------
-- TEST THE TRIGGERS
-- ----------------------------------------

-- Step 1: Insert valid employees
INSERT INTO Employee (EmpName, Dept, Salary) VALUES
('Raghav', 'Sales', 30000),
('Abhay', 'IT', 40000),
```

('Priya', 'HR', 35000);

-- Step 2: Try inserting invalid employee (should fail)

-- This will show an error intentionally (to test BEFORE trigger)

-- INSERT INTO Employee (EmpName, Dept, Salary) VALUES ('Test', 'Finance', 0);

-- Step 3: Delete one employee (to test AFTER DELETE trigger)

DELETE FROM Employee WHERE EmpName = 'Abhay';

-- Step 4: View results

SELECT * FROM Employee;

SELECT * FROM Emp_Audit;

(21)	GHRCEM , Practical Exam. 2025 **Sub: DBMS** **Class: TY BTECH AI/AIML** **Duration : 2 Hrs**
	N.B.:-1.write problem statement on Answer sheet, then write implementation steps in detail 2. Assume suitable data if necessary.

Implement following Queries in SQL database.

1. Create Bookstore table as Book_id, Book_Name, Author_Name, Price, Publication_year.
2. Insert following items into table:

Book_id	Book_Name	Author_Name	Price	Publication_year
201	DBMS	S. Korth	400	2004
202	ADBMS	A Navathe	450	2003
203	Data Mining	M. Kamber	530	2005
204	Web mining	Springer	420	2004

3. Display inserted items from bookstore.
4. Display details of Books whose id is 203 and name “ADBMS”
5. Give the Book_name and Author_Name of Books whose id is 201
6. Give the Price of Book whose Author_Name is Springer.
7. List the Book_id, Book_ name and Publication_ year of Book whose customer id is 202,204
8. Display all details of Books whose the customer id is 203.
9. Delete Book with Book id 204.
10. Find Average price of all books.

```
-- 1) Create database and use it
CREATE DATABASE bookstore_db;
USE bookstore_db;

-- 1) Create Bookstore table
CREATE TABLE Bookstore (
  Book_id INT PRIMARY KEY,
  Book_Name VARCHAR(100),
  Author_Name VARCHAR(100),
  Price INT,
  Publication_year INT
);

-- 2) Insert given records
INSERT INTO Bookstore VALUES
(201, 'DBMS', 'S. Korth', 400, 2004),
```

```
(202, 'ADBMS', 'A. Navathe', 450, 2003),
(203, 'Data Mining', 'M. Kamber', 530, 2005),
(204, 'Web Mining', 'Springer', 420, 2004);

-- 3) Display all inserted items
SELECT * FROM Bookstore;


-- 4) Display details of Books whose id is 203 and name “ADBMS”
SELECT * FROM Bookstore WHERE Book_id = 203 AND Book_Name = 'ADBMS';


-- 5) Give the Book_name and Author_Name of Books whose id is 201
SELECT Book_Name, Author_Name FROM Bookstore WHERE Book_id = 201;


-- 6) Give the Price of Book whose Author_Name = 'Springer'
SELECT Price FROM Bookstore WHERE Author_Name = 'Springer';


-- 7) List the Book_id, Book_Name, Publication_year of Books whose id is 202,204
SELECT Book_id, Book_Name, Publication_year
FROM Bookstore
WHERE Book_id IN (202, 204);


-- 8) Display all details of Books whose id = 203
SELECT * FROM Bookstore WHERE Book_id = 203;


-- 9) Delete Book with Book_id = 204
DELETE FROM Bookstore WHERE Book_id = 204;
```

-- 10) Find Average price of all Books

SELECT AVG(Price) AS Average_Price FROM Bookstore;

PRACTICAL 12 — (Equivalent SQL Implementation of DynamoDB Bookstore)

Code :

```
-- Q1) Create Bookstore table as (Book_id, Book_Name, Author_Name, Price, Publication_year)
CREATE DATABASE bookstore_nosql;
USE bookstore_nosql;

CREATE TABLE Bookstore (
  Book_id INT PRIMARY KEY,
  Book_Name VARCHAR(100),
  Author_Name VARCHAR(100),
  Price INT,
  Publication_year INT
);

-- Q2) Insert the given items into table
INSERT INTO Bookstore VALUES
(201, 'DBMS', 'S. Korth', 400, 2004),
(202, 'ADBMS', 'A. Navathe', 450, 2003),
(203, 'Data Mining', 'M. Kamber', 530, 2005),
(204, 'Web Mining', 'Springer', 420, 2004);

-- Q3) Display all inserted items from Bookstore
SELECT * FROM Bookstore;

-- Q4) List details of Books whose id is 203 and name “ADBMS”
SELECT * FROM Bookstore
WHERE Book_id = 203 AND Book_Name = 'ADBMS';

-- Q5) Give the Book_name and Author_Name of Books whose id is 201
```

```
SELECT Book_Name, Author_Name FROM Bookstore
WHERE Book_id = 201;

-- Q6) Give the Price of Book whose Author_Name is “Springer”
SELECT Price FROM Bookstore
WHERE Author_Name = 'Springer';

-- Q7) List the Book_id, Book_name and Publication_year of Books whose id is 202,204
SELECT Book_id, Book_Name, Publication_year
FROM Bookstore
WHERE Book_id IN (202, 204);

-- Q8) Display all details of Books whose id is 203
SELECT * FROM Bookstore
WHERE Book_id = 203;

-- Q9) Update Books table by adding one attribute ISSN_NO
ALTER TABLE Bookstore ADD COLUMN ISSN_NO VARCHAR(20);

-- Q10) Delete Book with Book id 204
DELETE FROM Bookstore WHERE Book_id = 204;
```