SAPTHAGIRI NPS UNIVERSITY

UNMATCHED EXCELLENCE, UNLIMITED POTENTIAL

School of Engineering and Technology

Department of CSE-Artificial Intelligence and Machine Learning

Laboratory Manual

Course Name: Database Management Systems

Course Code: 24BEAML306

Semester III

Prepared By

Dr. Abdul Khadar A

Professor, Department of CSE-AI&ML

Sapthagiri NPS University

14/5, Chikkasandra, Hesaraghatta Rd, Bengaluru, Karnataka 560057

Contents		
Sl. No	**Topic**	**Page No**

Vision of the University

Offer a transformative impact on society through unique learning experience in engineering technology, Medicine, Applied sciences, Business studies, Management studies and other areas of scholarship to the stakeholders to an unparalleled educational journey to serve the world and betterment of mankind. To establish Sapthagiri NPS university as a leading institution fostering academic excellence and wisdom for success in a dynamic world.

Mission of the University

- To provide a student Centric-learning environment focused on deep disciplinary knowledge, problem solving, leadership, and communication, interpersonal skills through innovative pedagogy and education reforms.
- To generate outstanding leaders in the field of health sciences and to provide optimum human patient-centered health care of the highest quality.
- To create and sustain a community of lifelong learners in an environment that emphasizes literacy, critical and innovative thinking, humanity, scientific inquiry and to promote patriotism and moral values.
- To impact in a transformative way regionally, nationally and globally to face the economic, social and health related challenges for nation building.
- To accomplish quality assurance, enhancement and sustenance in academics and research for a fair and social justice by providing equal opportunity.

Vision of the Department

To cultivate a center of excellence in Artificial Intelligence and Machine Learning, shaping the next generation of innovators and thought leaders who will drive technological advancement, solve real-world problems, and contribute to the global AI landscape.

Mission of the Department

- To deliver world-class education and training in Artificial Intelligence and Machine Learning, fostering deep theoretical knowledge and practical expertise.
- To advance cutting-edge research that addresses critical challenges across industries and society through interdisciplinary collaboration and accelerate the real-world impact of AI solutions.
- To promote innovation, entrepreneurship, and the ethical application of AI technologies for the betterment of the society.

Program Outcomes (Pos)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles ofmathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEOs):

PEO1: Graduates will develop strong foundational knowledge in Artificial Intelligence, Machine Learning, Data Science, and related computing fields to solve real-world problems and pursue successful careers in industry, research, or entrepreneurship.

PEO2: Graduates will engage in lifelong learning and innovation through higher studies, research, or professional certifications to adapt to the rapidly evolving AI/ML landscape.

PEO3: Graduates will exhibit professional ethics, leadership qualities, teamwork, and societal responsibility
while applying AI/ML solutions for sustainable development.

PEO4: Graduates will be capable of applying AI/ML across domains such as healthcare, finance, agriculture, cybersecurity, and automation with an interdisciplinary perspective.

Program Specific Outcomes (PSOs):

PSO1: Apply core principles of Artificial Intelligence, Machine Learning, Deep Learning, and Data Analytics to design and develop intelligent systems and predictive models.

PSO2: Utilize programming languages (Python, R), frameworks (TensorFlow, PyTorch), and simulation tools for real-time problem-solving in AI and ML domains.

PSO3: Demonstrate capabilities to innovate AI-based solutions, participate in research, and develop prototypes, contributing to product development or start-ups.

General Lab Guidelines

Do's

1. Maintain discipline in the Laboratory.
2. Before entering the Laboratory, keep the footwear on the shoe rack.
3. Proper dress code has to be maintained while entering the Laboratory.
4. Students should carry a lab observation book, student manual and record book completed in all aspects.
5. Read and understand the logic of the program thoroughly before coming to the laboratory.
6. Enter the login book before switching on the computer.
7. Enter your batch member names and other details in the slips for hardware kits.
8. Students should be at their concerned places; unnecessary movement is restricted.
9. Students should maintain the same computer until the end of the semester.
10. Report any problems in computers/hardware kits to the faculty member in-charge/laboratory technician immediately.
11. The practical result should be noted down into their observation and the result must be shown to the faculty member in-charge for verification.
12. After completing the experiments, students should switch off the computers, enter logout Time and keep the chairs properly.

Don'ts

1. Do not come late to the Laboratory.
2. Do not enter the laboratory without an ID card, lab dress code, observation book and record.
3. Do not leave the laboratory without the permission of the faculty in-charge.
4. Never eat, drink while working in the laboratory.
5. Do not handle any equipment before reading the instructions/instruction manuals.
6. Do not exchange the computers with others and hardware kits also.
7. Do not misbehave in the laboratory.
8. Do not alter computer settings/software settings.
9. External Disk/drives should not be connected to computers without permission, doing so will attract fines.
10. Do not remove anything from the kits/experimental set up without permission. Doing so will attract fines.
11. Do not mishandle the equipment / Computers.
12. Do not leave the laboratory without verification of hardware kits by the lab instructor.
13. Usage of Mobile phones, tablets and other portable devices are not allowed in restricted places.

Rules For Maintaining Laboratory Record

● Write your name, SRN and subject on the outside front cover of the record. Put that same information on the first page inside.
● Update Table of Contents every time you start each new experiment or topic
● Always use pen and write neatly and clearly
● Start each new topic (experiment, notes, calculation, etc.) on a right-side (odd numbered) page
● Obvious care should be taken to make it readable, even if you have bad handwriting
● Date to be written every page on the top right side corner
● On each right-side page
    - Title of experiment
    - Aim/Objectives
    - Components Required
    - Theory
    - Procedure described clearly in steps
    - Result
● On each left side page
    - Schema diagrams
    - Tables
    - Graphs
● Use labels and captions for figures and tables
● Strictly observe the instructions given by the Teacher/ Lab Instructor.

Syllabus

Program	B.E. CSE-Artificial Intelligence and Machine Learning				
Semester	III				
Course	Database Management Systems			Course Code	24BEAML306
L	T	P	Credits	Total Hours	60
3	0	2	4		

Course Objective:

This course introduces the fundamental concepts of database systems, query processing and normalization, transaction management and recovery techniques for database management. The course also covers modern paradigms like NoSQL and Big Data storage and prepares students to handle contemporary data challenges in large-scale environments.

Module	Topics	Hours
I	**Introduction to Databases and ER Model** **Database Systems:** Need for DBMS, Applications of Databases. DBMS Architecture, Data Independence, Data Models (Hierarchical, Network, Relational, Object-Oriented). Three-Schema Architecture. **ER Model:** Entities, Attributes, Relationships, Cardinality, Participation. **Enhanced ER Model:** Specialization, Generalization, and Aggregation. Functional Dependencies and Anomalies.	9
II	**Relational Model and Structured Query Language (SQL)** **Introduction to Relational Model:** Attributes, Tuples, Domains, and Relations Keys. **Relational Integrity Constraints:** Key Constraints, Referential Integrity, Domain Constraints. **Relational Algebra:** Selection, Projection, Joins, Set Operations. **SQL:** Basics, Data Types, DDL (CREATE, ALTER, DROP), DML (INSERT, UPDATE, DELETE), DCL (GRANT, REVOKE), TCL (COMMIT, ROLLBACK).	9
III	**Query Processing and Normalization** **SQL Query Processing**: Joins, Nested Queries, Views, Aggregation. Integrity Constraints, Triggers and stored procedures, query optimization techniques. **Normalization:** 1NF, 2NF, 3NF, BCNF. Schema Refinement for Eliminating Redundancy.	9
IV	**Transaction Management and Database Recovery Techniques** **Transactions:** ACID Properties, States of Transactions. Serializability and Recoverability. **Database Recovery Techniques:** Log-Based, Check points, Concurrency Control **Application Design and Development:** Complex Data Types, Application Development	9
V	**NoSQL, and Big Data Storage System**	9

	NoSQL Databases: Key-Value Stores, Document Stores, Column Stores (MongoDB, Cassandra). **Big Data Storage Techniques and BASE Properties.**	
VI	**DBMS Lab** The programs will include topics in Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, MongoDB or any other DBMS under Linux/ Windows environment. Student will develop a mini project incorporating all topics covered in the syllabus.	**15**

Course Outcomes:

On successful completion of this course, students will be able to:

CO1: Explain the basic concepts in database management systems and ER models.
CO2: Describe transaction management & recovery mechanisms, NoSQL & Big Data Storage System**.**
CO3: Design efficient & normalized database schemas using ER relations& normalization techniques
CO4: Apply SQL queries for data retrieval and manipulation
CO5: Apply Integrity constraints and schemas, Basics of MongoDB and evaluate the project developed for an application.

Text books:

1. **Database System Concepts**, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 7th Edition, McGraw Hill, 2021.

Reference books:

1. **Fundamentals of Database Systems,** Ramez Elmasri, Shamkant B. Navathe, 7th Edition, Pearson, 2016.
2. **Database Management Systems**, Raghu Ramakrishnan, Johannes Gehrke, 3rd Edition, McGraw Hill, 2014.
3. **SQL for Data Scientists: A Beginner's Guide for Building Datasets for Analysis**, Renee M. P. Teate, 1st Edition, Wiley, 2021.
4. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence**, Pramod J. Sadalage, Martin Fowler, 1st Edition, Addison-Wesley, 2012.

E-Learning Sources:

1. https://nptel.ac.in/courses/106106220
2. https://learn.mongodb.com/courses/mongodb-logging-basics
3. https://www.udemy.com/topic/database-management/?srsltid=AfmBOoraF28bRhnv8l_r2JUO0f72ehtRNxzAgABK9nadPTrL9htxXF7T
4. https://www.coursera.org/learn/introduction-to-relational-databases

Assessment Components

a)	**Continuous Internal Assessment**	IA – I	Each **Internal Assessment (IA)** will be conducted for 50 marks and scaled down to **15 (Total Test Marks 30 = IA-I + IA-II)**
		IA – II	
		Lab	**Lab Internal Assessment (IA)** will be conducted for **15 marks.** (Lab conduction **10 marks** + Lab Test **05 marks**)
		Quiz / Assignment	**05 Marks**
		Total CIA	**50 Marks**
b)	**Semester End Examinations**		**50 Marks**

Laboratory Evaluation Components

Sl. No	Evaluation Component	Weightage	Details	Marks
1	Lab Conduction	20%	CIE shall be conducted for 100 Marks and scaled down to 10 Marks	Execution – 4 Observation – 2 Viva – 2 Record – 2 No of Experiments = 7 Total = 07 x 10 = 70
2	Lab internals	10%	Internal (CIA) shall be conducted for 50 Marks and scaled down to 5 Marks	Write up -20 Execution – 20 Viva – 10

Teaching-Learning Process

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) needs not to be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding
9. Use any of these methods: Chalk and board, Active Learning, Case Studies

List of Lab Experiments

>

Exp. No	Problem Statement
1.	Create a table called Employee & execute the following. Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION) 1. Create a user and grant all permissions to the user. 2. Insert the any three records in the employee table contains attributes EMPNO,ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.
2.	Create a table called Employee that contain attributes EMPNO,ENAME,JOB, MGR,SAL & execute the following. 1. Add a column commission with domain to the Employee table. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee whose Empno is 105.
3.	Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Groupby, Orderby. Employee(E_id, E_name, Age, Salary) 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employee table 3. Find the Maximum age from employee table. 4. Find the Minimum age from employee table. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.
4	Create a table called department(DEPT_ID,DEPT_NAME,DEPT_LOCATION) Create a table called employee(EMP_ID,EMP_NAME,EMP_SALARY,EMP_EXP,EMP_DEPT_ID) 1. Insert any five records into department table. 2. Insert at least 2 records in each department. 3. Display the employees who are getting more than 40000 salaries along with dept name. 4. Display the employee who are having more than 5 years of experience along with dept name.
5.	**Create a tables STUDENT1 and STUDENT2 with attributes (SRN, Name, Branch)** (a) Display all students from both tables (remove duplicates) (b) Display all students from both tables (including duplicates) (c) Display students common to both tables (d) Display students present in STUDENTS_A but not in STUDENTS_B (e) Display students present in STUDENTS_B but not in STUDENTS_A
6.	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)
7.	Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

Experiment-1

Create a table called Employee & execute the following.
Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION)
1. Create a user and grant all permissions to the user.
2. Insert the any three records in the employee table contains attributes EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result.
3. Add primary key constraint and not null constraint to the employee table.
4. Insert null values to the employee table and verify the result.

Solutions:
1. Create a user and grant all permissions to the user.
//connect to MySql database first
//create user
Create user "SRN" identified by "password";
User created.
//grant the permission
Grant all privileges on *.* to 'SRN' with grant option;
Flush privileges;

Now user SRN can log in with full privileges:
mysql -u SRN –p;

Create a table employee with the given attributes:

SQL> create table employee (empno int, ename varchar(10), job varchar(10), mgr_no int, sal int, commission int);

SQL>desc employee;

2. Insert the any three records in the employee table and use rollback.
insert into Employee values(101,'abhi','manager',1234,10000,'70');
insert into employee values(102,'rohith','analyst',2345,9000,'65');
insert into employee values(103,'david','analyst',3456,9000,'65');
insert into employee values(104,'rahul','clerk',4567,7000,'55');
insert into employee values(105,'pramod','salesman',5678,5000,'50');

SQL>select * from Employee;

SQL>rollback;

Check the result

3. Add primary key constraint and not null constraint to the employee table.

SQL> alter table employee add primary key (empno);
SQL> alter table employee modify(ename varchar(10) not null);

SQL>desc Employee;

4. Insert null values to the employee table and verify the result.

SQL>insert into employee values(106, 'shashi' , 'HR' , 5509, ' ‘ ,80);
SQL> select * from Employee;

Viva Question:
1. What is data?
Data is a collection of information gathered by observations, measurements, research or analysis.

2. What is database?
A database is an electronically stored, systematic collection of data. It can contain any type of data, including words, numbers, images, videos, and files.

3. What is DBMS?
Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data.

4. What is a Database system?
A database is an organized collection ofstructured information, or data, typically stored electronically in a computer system.

5. What are the advantages of DBMS?
The advantages of database management include improved data integrity, consistency, and security, efficient data access and sharing, and reduced data redundancy and inconsistency.

6. What is relational database?
A relational database is a collection of information that organizes data in predefined relationships where data is stored in one or more tables (or "relations") of columns and rows.

7. What is Table?
A table is an arrangement of data in rows and columns, or possibly in a more complex structure.

8. What is a Tuple?
A tuple is an ordered sequence of values. The values can be repeated, but their number is always finite.

9. What is Columns?
column or pillar in architecture and structural engineering is a structural element that transmits, through compression, the weight of the structure above to other structural elements below.

10. What is a query?

A query is a question or a request for information expressed in a formal manner.

Experiment-2

Create a table called Employee that contain attributes EMPNO,ENAME,JOB, MGR,SAL) execute the following.
1. Add a column commission with domain to the Employeetable.
2. Insert any five records into the table.
3. Update the column details of job
4. Rename the column of Employee table using alter command.
5. Delete the employee whose Empno is 105.

Preparation:

```
SQL> create table employee (empno int, ename varchar(10), job varchar(10), mgr int,
        sal int);

SQL> desc employee;
```

1. Add a column commission with domain to the Employee table.

```
SQL> ALTER TABLE employee ADD commission int;
SQL> desc employee;
```

2. Insert any five records into the table.

```
insert into Employee values(101,'abhi','manager',1234,10000,'70');
insert into employee values(102,'rohith','analyst',2345,9000,'65');
insert into employee values(103,'david','analyst',3456,9000,'65');
insert into employee values(104,'rahul','clerk',4567,7000,'55');
insert into employee values(105,'pramod','salesman',5678,5000,'50');

SQL>select * from Employee;
```

3. Update the column details of job

```
SQL> update employee set job='trainee' where empno=103;

SQL> select * from employee;
```

4. Rename the column of Employee table using alter command.

```
SQL> ALTER TABLE employee CHANGE  mgr   manager_no INT;
SQL>desc employee;
```

5. Delete the employee whose Empno is 105

```
SQL> delete employee where empno=105;
```

Viva Questions

1. What is an Attribute?
A quality, character, or characteristic ascribed to someone or something has leadership attributes.

2. What is Single valued Attributes ?
Single-valued attributes Single-valued attributes accept only one value. For single-valued attributes, the syntax is: attribute = value attribute = "value with spaces" Multi-valued attributes.

3. What is Multi valued Attributes?
A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity.

4. What is Compound /Composite Attribute?
A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity.

5. What is Simple/Atomic Attributes?
A simple, or atomic, attribute is one that cannot be decomposed into meaningful components.

6. What is Stored Attribute?
Stored attributes are those attributes that are stored in the physical database for e.g date of birth.

7. What is Derived Attribute ?
A derived attribute is one that can be figured out from other information. An example is "age". A person's age can be derived from date of birth.

8. What is Complex Attributes?
Complex attributes are formed by grouping together the attributes of composite and multi-valued attributes.

9. What is Key Attribute ?
In DBMS, key attributes refer to the specific fields or columns in a table that are used to uniquely identify each record in the table.

10. What is Non Key Attributes ?
The values of a primary key cannot be duplicated. Non-prime (non-key) attributes are those that are not the primary key attributes.

Experiment-3

Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Orderby. Employee(E_id, E_name, Age, Salary)
1. Create Employee table containing all Records E_id, E_name, Age, Salary.
2. Count number of employee names from employee table
3. Find the Maximum age from employee table.
4. Find the Minimum age from employee table.
5. Find salaries of employee in Ascending Order.
6. Find grouped salaries of employees.

Preparation:
1. Create Employee table containing all Records E_id, E_name, Age, Salary.

SQL> create table employee (E_id int, E_name varchar(10), age int, sal int);

SQL>desc Employee;

Insert any five records into Employee table:
insert into Employee values(10,'abhi',25 ,10000);
insert into employee values(20,'rohith',30,9000);
insert into employee values(30,'david',28,9000);
insert into employee values(40,'rahul',29,7000);
insert into employee values(50,'pramod',31,8000);

SQL>select * from employee;

2. Count number of employee names from employee table

SQL> select count(E_name) from Employee;

3. Find the Maximum age from employee table.

SQL>select max(age) from Employee;

4. Find the Minimum age from employee table

SQL>select min(age) from Employee;

5. Find salaries of employee in Ascending Order.

SQL> SELECT * FROM Students ORDER BY salary ASC;

6. Find grouped salaries of employees.

SQL> select salary from employee group by salary;

Viva Questions

1. What is an Attribute?

A table consists of several records(row), each record can be broken down into several smaller parts of data known as Attributes. The above Employee table consist of four attributes, ID, Name, Age and Salary.

2. What is Single valued Attributes ?

An attribute, that has a single value for a particular entity. For example, age of a employee entity.

3. What is Multi valued Attributes?

An attributes that may have multiple values for the same entity. For example colors of a car entity.

4. What is Compound /Composite Attribute?

Attribute can be subdivided into two or more other Attribute. For Example, Name can be divided into First name, Middle name and Last name.

5. What is Simple/Atomic Attributes?

The attributes which cannot be divided into smaller subparts are called simple or atomic attributes. For example, age of employee entity

6. What is Stored Attribute?

An attribute, which cannot be derived from other attribute, is known as stored attribute. For example, BirthDate of employee.

7. What is Derived Attribute ?

Attributes derived from other stored attribute. For example age from Date of Birth and Today's date.

8. What is Complex Attributes?

If an attribute of an entity, is built using composite and multivalued attributes, then these attributes are called complex attributes. For example, a person can have more than one residence and each residence can have multiple phones, an addressphone for a person entity can be specified as – {Addressphone (phone {(Area Code, Phone Number)}, Address(Sector Address (Sector Number,House Number), City, State, Pin))}.Here {} are used to enclose multivalued attributes and () are used to enclose composite attributes with comma separating individual attributes.

9. What is Key Attribute ?

It represents primary key. It is an attribute, that has distinct value for each entity/element in an entity set. For example, Roll number in a Student Entity Type.

10. What is Non Key Attributes ?

These are attributes other than candidate key attributes in a table. For example Firstname is a non key attribute as it does not represent the main characteristics of the entity.

Experiment 4:

Create a table called department(DEPT_ID,DEPT_NAME,DEPT_LOCATION)
Create a table called
employee(EMP_ID,EMP_NAME,EMP_SALARY,EMP_EXP,EMP_DEPT_ID)

1. Insert any five records into department table.
2. Insert at least 2 records in each department.
3. Display the employees who are getting more than 40000 salaries along with dept name.
4. Display the employee who are having more than 5 years of experience along with dept name.

Preparation:

Department table

create table dept(DEPT_ID int primary key,DEPT_NAME varchar(10) not null,DEPT_LOCATION varchar(10) not null);

Query OK, 0 rows affected (0.12 sec)

desc dept;

Employee table

create table employee(EMP_ID int primary key,EMP_NAME varchar(10) not null,EMP_EXP int not null,EMP_SALARY int not null,EMP_DEPT_ID int,foreign key(EMP_DEPT_ID) references dept(DEPT_ID));

Query OK, 0 rows affected (0.05 sec)

Insert any five records into department table.

insert into dept values(1001,'testing','ECITY');

Query OK, 1 row affected (0.01 sec)

insert into dept values(2001,'HR','ECITY');

Query OK, 1 row affected (0.01 sec)

insert into dept values(3000,'Service','KMangala');

Query OK, 1 row affected (0.01 sec)

insert into dept values(4000,'Develop','WField');

Query OK, 1 row affected (0.01 sec)

Insert at least 2 records in each department.

insert into employee values(100,'Ashwini',5,45000,1001);

Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(101,'Ranjan',1,25000,1001);

Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(201,'Asha',10,75000,2001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(202,'Ambika',6,45000,2001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(301,'Amarnath',2,35000,3000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(302,'Gokul',8,85000,3000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(401,'Gowtham',3,50000,4000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(402,'Gomathi',13,150000,4000);
Query OK, 1 row affected (0.01 sec)

Display the employees who are getting more than 40000 salaries along with dept name.

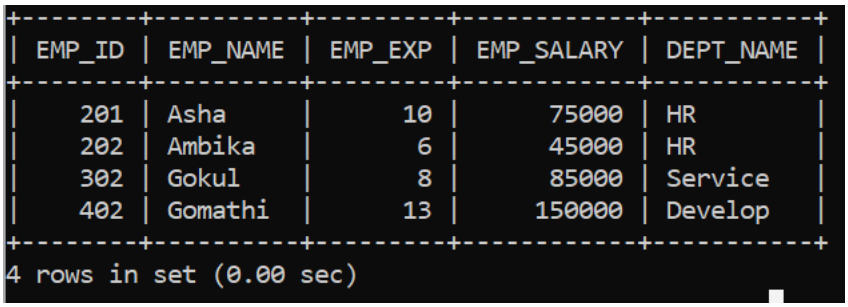
select E.EMP_ID,E.EMP_NAME,E.EMP_EXP,E.EMP_SALARY,D.DEPT_NAME from employee E,dept D where E.EMP_DEPT_ID=D.DEPT_ID and E.EMP_SALARY>40000;

EMP_ID	EMP_NAME	EMP_EXP	EMP_SALARY	DEPT_NAME
100	Ashwini	5	45000	testing
201	Asha	10	75000	HR
202	Ambika	6	45000	HR
302	Gokul	8	85000	Service
401	Gowtham	3	50000	Develop
402	Gomathi	13	150000	Develop

6 rows in set (0.00 sec)

Display the employee who are having more than 5 years of experience along with dept name.

select E.EMP_ID,E.EMP_NAME,E.EMP_EXP,E.EMP_SALARY,D.DEPT_NAME from employee E,dept D where E.EMP_DEPT_ID=D.DEPT_ID and E.EMP_EXP>5;



EMP_ID	EMP_NAME	EMP_EXP	EMP_SALARY	DEPT_NAME
201	Asha	10	75000	HR
202	Ambika	6	45000	HR
302	Gokul	8	85000	Service
402	Gomathi	13	150000	Develop

4 rows in set (0.00 sec)

Experiment **5**

Aim: To perform Set Operations in SQL using multiple tables to understand how to combine and compare datasets.

Create a tables STUDENT1 and STUDENT2 with attributes (SRN, Name, Branch)

(a) Display all students from both tables (remove duplicates)
(b) Display all students from both tables (including duplicates)
(c) Display students common to both tables
(d) Display students present in STUDENTS_A but not in STUDENTS_B
(e) Display students present in STUDENTS_B but not in STUDENTS_A

Create Table: STUDENTS_A

```
CREATE TABLE STUDENTS_A (
   SRN  INT PRIMARY KEY,
   Name VARCHAR(30),
   Branch VARCHAR(20)
);
```

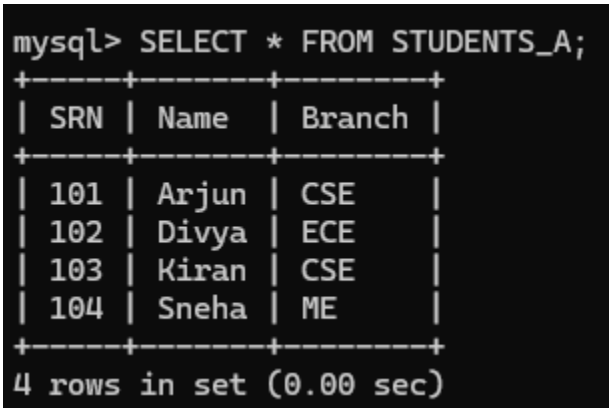
Create Table: STUDENTS_B

```
CREATE TABLE STUDENTS_B (
   SRN INT PRIMARY KEY,
   Name VARCHAR(30),
   Branch VARCHAR(20)
);
INSERT INTO STUDENTS_A (SRN, Name, Branch) VALUES (101, 'Arjun', 'CSE'),
(102, 'Divya', 'ECE'), (103, 'Kiran', 'CSE'), (104, 'Sneha', 'ME');

INSERT INTO STUDENTS_B (SRN, Name, Branch) VALUES (103, 'Kiran', 'CSE'),
(104, 'Sneha', 'ME'), (105, 'Rahul', 'EEE'), (106, 'Priya', 'CSE');

SELECT * FROM STUDENTS_A;
```

OUTPUT:



```
mysql> SELECT * FROM STUDENTS_A;
+-----+-------+--------+
| SRN | Name  | Branch |
+-----+-------+--------+
| 101 | Arjun | CSE    |
| 102 | Divya | ECE    |
| 103 | Kiran | CSE    |
| 104 | Sneha | ME     |
+-----+-------+--------+
4 rows in set (0.00 sec)
```

```
SELECT * FROM STUDENTS_B;
```

OUTPUT:

```
mysql> SELECT * FROM STUDENTS_B;
+-----+-------+--------+
| SRN | Name  | Branch |
+-----+-------+--------+
| 103 | Kiran | CSE    |
| 104 | Sneha | ME     |
| 105 | Rahul | EEE    |
| 106 | Priya | CSE    |
+-----+-------+--------+
4 rows in set (0.00 sec)
```

(a) Display all students from both tables (remove duplicates)

```
SELECT * FROM STUDENTS_A
UNION
SELECT * FROM STUDENTS_B;
```

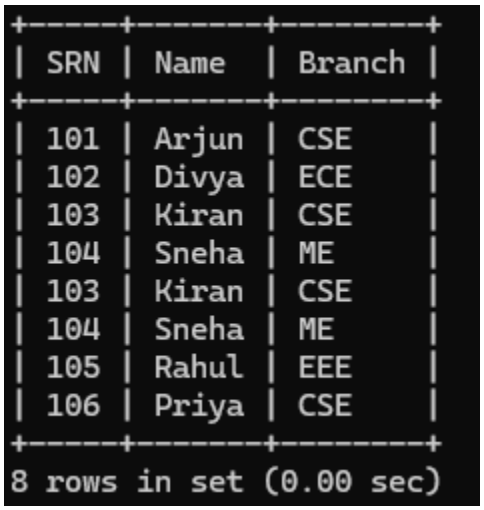
OUTPUT:

```
+-----+-------+--------+
| SRN | Name  | Branch |
+-----+-------+--------+
| 101 | Arjun | CSE    |
| 102 | Divya | ECE    |
| 103 | Kiran | CSE    |
| 104 | Sneha | ME     |
| 105 | Rahul | EEE    |
| 106 | Priya | CSE    |
+-----+-------+--------+
6 rows in set (0.01 sec)
```

(b) Display all students from both tables (including duplicates)

```
SELECT * FROM STUDENTS_A
UNION ALL
SELECT * FROM STUDENTS_B;
```

OUTPUT:



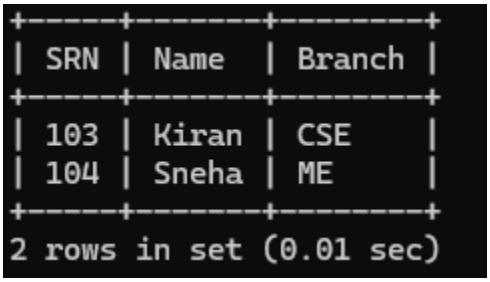
SRN	Name	Branch
101	Arjun	CSE
102	Divya	ECE
103	Kiran	CSE
104	Sneha	ME
103	Kiran	CSE
104	Sneha	ME
105	Rahul	EEE
106	Priya	CSE

8 rows in set (0.00 sec)

(c) Display students common to both tables

```
SELECT * FROM STUDENTS_A
INTERSECT
SELECT * FROM STUDENTS_B;
```

OUTPUT:



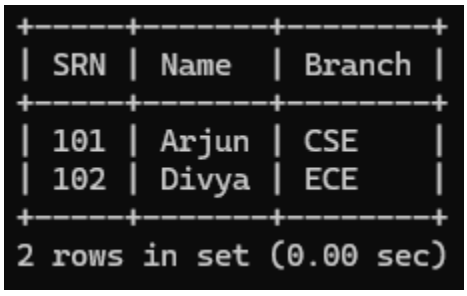
SRN	Name	Branch
103	Kiran	CSE
104	Sneha	ME

2 rows in set (0.01 sec)

(d) Display students present in STUDENTS_A but not in STUDENTS_B

```
SELECT * FROM STUDENTS_A
EXCEPT
SELECT * FROM STUDENTS_B;
```

OUTPUT:



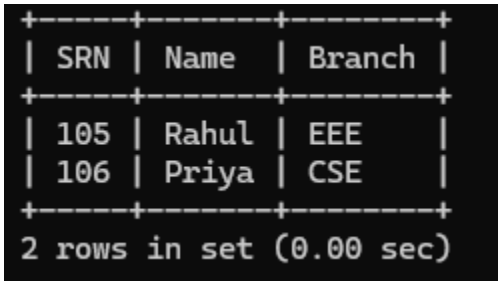
SRN	Name	Branch
101	Arjun	CSE
102	Divya	ECE

2 rows in set (0.00 sec)

(e) Display students present in STUDENTS_B but not in STUDENTS_A

```
SELECT * FROM STUDENTS_B
EXCEPT
SELECT * FROM STUDENTS_A;
```

OUTPUT:



```
+-----+-------+--------+
| SRN | Name  | Branch |
+-----+-------+--------+
| 105 | Rahul | EEE    |
| 106 | Priya | CSE    |
+-----+-------+--------+
2 rows in set (0.00 sec)
```

Experiment 6:

Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary.
CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)

Preperation:
1. Create Customer Table:

SQL> create table Customers (id int, name varchar(10), age int, sal int, address varchar(50));

insert into Customers values(10,'abhi',25 ,10000,"Bangalore");
insert into Customers values(20,'rohith',30,9000, "Delhi");
insert into Customers values(30,'david',28,9000, "Pune");
insert into Customers values(40,'rahul',29,7000, "Mumbai");
insertinto Customers values(50,'pramod',31,8000,"Mysore");

SQL> Select * from Customer;
2. Creating trigger

```
SQL> CREATE TRIGGER sal_diff
                BEFORE INSERT OR UPDATE OR DELETE ON Customer
                FOR EACH ROW
                BEGIN
                   DECLARE sal_difference DECIMAL(10,2);
                   -- Handle INSERT or UPDATE when id > 0
                   IF (NEW.id > 0) THEN
                      IF (OLD.sal IS NOT NULL) THEN
                         SET sal_difference = NEW.sal - OLD.sal;
                         INSERT INTO salary_log (previous_salary, current_salary,
                                  salary_difference)
                         VALUES (OLD.sal, NEW.sal, sal_difference);
                      ELSE
                         -- If it's a fresh insert, log only new salary
                         INSERT INTO salary_log (previous_salary, current_salary,
                                  salary_difference)
                         VALUES (NULL, NEW.sal, NULL);
                      END IF;
                   END IF;
                   -- Handle DELETE (log old salary only)
                   IF (OLD.sal IS NOT NULL AND NEW.id IS NULL) THEN
                      INSERT INTO salary_log (previous_salary, current_salary,
                                  salary_difference)
                      VALUES (OLD.sal, NULL, NULL);
                   END IF;
                END$$
```

Finding salary difference
SQL>UPDATE CUSTOMERS SET salary ='10000' where id='50';

Viva Questions

1. What is a primary key?

A primary key is a column whose values uniquely identify every row in a table.

2. What are the conditions for a field to be a primary key?

- No two rows can have the same primary key value.
- Every row must have a primary key value.
- The primary key field cannot be null.
- Value in a primary key column can never be modified or updated, if any foreign key refers to that primary key.

3. What is a Foreign Key ?

When a "one" table's primary key field is added to a related "many" table in order to create the common field which relates the two tables, it is called a foreign key in the "many" table. For example, the salary of an employee is stored in salary table. The relation is established via foreign key column “Employee_ID_Ref” which refers “Employee_ID” field in the Employee table.

4. What is Super Key?

A set of attributes (one or more) that collectively identifies an entity in an entity set.

5. What is Candidate Key

A minimal super key is called a candidate key. An entity set may have more than one candidate key.

6. What is a query?

A query with respect to DBMS relates to user commands that are used to interact with a data base. The query language can be classified into data definition language and data manipulation language.

7. Define SQL Insert Statement ?

SQL INSERT statement is used to add rows to a table.

8. Define SQL Update Statement ?

SQL Update is used to update data in a row or set of rows specified in the filter condition.

9. Define SQL Delete Statement ?

SQL Delete is used to delete a row or set of rows specified in the filter condition.

10. What is order by clause?

ORDER BY clause helps to sort the data in either ascending order to descending

Viva Questions

1. What is Mapping Cardinalities

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

2. What are the different types of Mapping

- One to one
- One to many
- Many to one
- Many to many

3. What is One-to-one mapping?

One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

4. What is One-to-many mapping?

One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

5. What is Many-to-one mapping?

More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

6. What is Many-to-many mapping?

One entity from A can be associated with more than one entity from B and vice versa.

Experiment-7:

Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read,Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

How to Install and Configure MongoDB in Windows?

Step 1: Download MongoDB

1. Go to the official MongoDB download page: https://www.mongodb.com/try/download/community
2. Select:
    - **Version** → Latest (e.g., 7.0 or higher)
    - **Platform** → Windows
    - **Package** → MSI (Windows Installer)
3. Click **Download**.

Step 2: Install MongoDB

1. Run the **MSI Installer** you downloaded.
2. Choose **Complete Setup**.
3. In **Service Configuration**:
    - Keep the default settings (MongoDB will run as a Windows service).
    - Optionally, change the data directory location if needed.
4. Finish installation.

Step 3: Setup MongoDB Folders (if needed)

MongoDB uses two main folders:

- **Data files** → C:\Program Files\MongoDB\Server\<version>\data\db
- **Log files** → C:\Program Files\MongoDB\Server\<version>\log

If these folders are not created automatically, create them manually.

Step 4: Start MongoDB

If installed as a service, MongoDB starts automatically.

To check:

1. Open **Command Prompt (cmd)** as Administrator.
2. Run:

net start MongoDB

If you didn't install as a service, you can start it manually:

"C:\Program Files\MongoDB\Server\<version>\bin\mongod.exe" --dbpath "C:\data\db"

(Replace <version> with your MongoDB version, e.g., 7.0)

Step 5: Connect to MongoDB

1. Open a new **Command Prompt**.
2. Run:

"C:\Program Files\MongoDB\Server\<version>\bin\mongosh.exe"
This opens the **MongoDB shell**. You're now connected! 🎉

Step 6: (Optional) Add MongoDB to PATH
So you don't need to type the full path every time:

1. Search **Environment Variables** in Windows.
2. Edit **Path** → Add:

C:\Program Files\MongoDB\Server\<version>\bin
Now you can just run:
mongod
mongosh

CRUD Operations:
1. Create (Insert)
To create or insert data into a MongoDB collection, you use the insertOne() or insertMany() methods.

Insert a single document:
db.collection('yourCollection').insertOne({ key: value });

Insert multiple documents:

```
db.collection('yourCollection').insertMany([
 { key1: value1 },
 { key2: value2 },
// more documents ]);
```

2. Read (Query)
To read or retrieve data from a MongoDB collection, you use the find() method.

Find all documents:
db.collection('yourCollection').find();

Find documents with a specific condition:
db.collection('yourCollection').find({ key: value });

3. Update
To update existing documents in a MongoDB collection, you use the updateOne() or updateMany()

methods.

Update a single document:

```
db.collection('yourCollection').updateOne(
 { key: value }, // filter
 { $set: { newField: newValue } } // update operation
);
```

Update multiple documents:

```
db.collection('yourCollection').updateMany(
 { key: value }, // filter
 { $set: { newField: newValue } } // update operation
);
```

4. Delete

To delete documents from a MongoDB collection, you use the deleteOne() or deleteMany() methods.

Delete a single document:

```
db.collection('yourCollection').deleteOne({ key: value });
```

Delete multiple documents:

```
db.collection('yourCollection').deleteMany({ key: value });
```

Viva Questions

1. How do you perform CRUD operations create, read, update, deleteMongoDB?

MongoDB CRUD Operations The Create operation is used to insert new documents in the MongoDB database.The Read operation is used to query a document in the database. The Update operation is used to modify existing documents in the database.The Delete operation is used to remove documents in the database.

2. What are the CRUD operations in NoSQL database?

CRUD is the acronym for CREATE, READ, UPDATE and DELETE. These terms describe the four essential operations for creating and managing persistentdata elements, mainly in relational and NoSQL databases.

3. How to update in CRUD operations?

You can perform update, insert and delete operation in the Grid. While performing these operations, the corresponding event is invoked. In that event SQL query is used to update the database. The events for performing CRUD operation are declared.

4. How can we create updating and deleting documents in MongoDB?

The MongoDB shell provides the following methods to update documents in acollection:

1. To update a single document, use db. collection. updateOne() .
2. To update multiple documents, use db. collection. updateMany() .
3. To replace a document, use db. collection. replaceOne() .

5. What is the full form of CRUD in MongoDB?

The basic methods of interacting with a MongoDB server are called CRUD operations. CRUD stands for Create, Read, Update, and Delete. These CRUDmethods are the primary ways you will manage the data in your databases.

6. What are the CRUD methods in REST API?

CRUD stands for Create, Read, Update, and Delete. These are the four fundamental operations of persistent storage. In the context of RESTful APIs , they correspond to the HTTP methods POST, GET, PUT/PATCH, and DELETE.

7. How to create a collection in MongoDB?

Several ways can be employed to create and remove collections in MongoDB. Of which one way is by using db. Create Collection (name, options). MongoDB creates a collection for an inserted command automatically if no similar collectionalready exists in the MongoDB database.