

**JSS MAHAVIDYAPEETHA**



## **LABORATORY MANUAL**

**Subject Name: DBMS LAB**

**Subject Code: KCS 551**

**COURSE: B.Tech.**

**SEMESTER: V SEM**

<b>Name</b>	
<b>Roll No.</b>	
<b>Section-Batch</b>	

**Department of Computer Science and Engineering**  
**JSS ACADEMY OF TECHNICAL EDUCATION**  
**C-20/1, SECTOR-62, NOIDA**

### **VISION OF THE INSTITUTE**

JSS Academy of Technical Education, Noida aims to become an Institution of excellence in imparting quality Outcome Based Education that empowers the young generation with Knowledge, Skills, Research, Aptitude and Ethical values to solve Contemporary Challenging Problems.

### **MISSION OF THE INSTITUTE**

Develop a platform for achieving globally acceptable level of intellectual acumen and technological competence.

Create an inspiring ambience that raises the motivation level for conducting quality research.

Provide an environment for acquiring ethical values and positive attitude.

### **VISION OF THE DEPARTMENT**

To spark the imagination of the Computer Science Engineers with values, skills and creativity to solve the real world problems.

### **MISSION OF THE DEPARTMENT**

To inculcate creative thinking and problem solving skills through effective teaching, learning and research.

To empower professionals with core competency in the field of Computer Science and Engineering.

To foster independent and lifelong learning with ethical and social responsibilities.

## **PROGRAM EDUCATIONAL OUTCOMES (PEOs)**

**PEO1:** To empower students with effective computational and problem solving skills.

**PEO2:** To enable students with core skills for employment and entrepreneurship.

**PEO3:** To imbibe students with ethical values and leadership qualities.

**PEO4:** To foster students with research oriented ability which helps them in analyzing and solving real life problems and motivates them for pursuing higher studies.

## **PROGRAM OUTCOMES (POs)**

**Engineering Graduates will be able to:**

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

**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**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.

**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.

**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**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.

**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.

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

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

**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.

**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.

**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 SPECIFIC OUTCOMES (PSOs)**

**PSO1:** An ability to apply foundation of Computer Science and Engineering, algorithmic principles and theory in designing and modeling computation based systems.

**PSO2:** The ability to demonstrate software development skills.

**COURSE OUTCOMES (COs)**

SL.NO	DESCRIPTION	COGNITIVE LEVEL (BLOOMS TAXONOMY)
<b>C308.1</b>	Understand and apply oracle 11 g products for creating tables, views, indexes, sequences and other database objects.	K2, K4
<b>C308.2</b>	Design and implement a database schema for company data base, banking data base, library information system, payroll processing system, student information system.	K3, K5, K6
<b>C308.3</b>	Write and execute simple and complex queries using DDL, DML, DCL and TCL	K4, K5
<b>C308.4</b>	Write and execute PL/SQL blocks, procedure functions, packages and triggers, cursors.	K4, K5
<b>C308.5</b>	Enforce entity integrity, referential integrity, key constraints, and domain constraints on database.	K3, K4

**CO-PO MAPPING**

Cos	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
<b>C308.1</b>	3	3	3	1	3	0	0	2	2	1	2	3	2	2
<b>C308.2</b>	3	3	3	3	3	0	0	2	2	1	2	2	2	2
<b>C308.3</b>	3	2	2	3	3	0	0	2	2	1	2	2	2	2
<b>C308.4</b>	3	3	3	3	3	0	0	2	2	1	2	2	2	2
<b>C308.5</b>	3	3	3	3	3	0	0	2	2	1	2	3	2	2
<b>C308</b>	<b>3.00</b>	<b>2.80</b>	<b>2.80</b>	<b>2.60</b>	<b>3.00</b>	<b>0.00</b>	<b>0.00</b>	<b>2.00</b>	<b>2.00</b>	<b>1.00</b>	<b>2.00</b>	<b>2.40</b>	<b>2.00</b>	<b>2.00</b>

**CO-PSO MAPPING**

Cos	PSO 1	PSO 2
<b>C308.1</b>	2	2
<b>C308.2</b>	2	2
<b>C308.3</b>	2	2
<b>C308.4</b>	2	2
<b>C308.5</b>	2	2
<b>C308</b>	<b>2.00</b>	<b>2.00</b>

**LIST OF EXPERIMENTS**

<b>Sr. No.</b>	<b>Title of experiment</b>	<b>Corresponding CO</b>
1	Write the queries for Data Definition and Data Manipulation Language.	C308.3
2	To explore ‘select’ statement using various clauses like where, order by, between, like, group by, having and logical/relational operator. To implement various DML statements	C 308.3
3	To implement nested and correlated queries and to create various views and indexes	C 308.1
4	To enforce various integrity constraint and to implement various DCL statements.	C 308.3, C 308.5
5	To implement Cursors and triggers using PL/SQL.	C 308.4
6	To implement Procedures and Functions using PL/SQL.	C 308.4
7	Design and implementation of payroll processing systems.	C 308.2
8	Design and implementation of Library Information systems.	C 308.2
9	Design and implementation of Student Information systems.	C 308.2
10	Automatic Backup of Files and Recovery of Files	C308.1,C308.2, C308.3, C308.4, C 308.5
11	Mini Project (Design and Development of Data & Application) as Per AKTU Curriculum – Refer Appendix	C308.1,C308.2, C308.3, C308.4, C 308.5
<b>Content beyond syllabus (not limited to two)</b>		
12	To implement various TCL commands	C308.3

## **INTRODUCTION**

A Database Management System (DBMS) is a set of computer programs that controls the creation, maintenance, and the use of a database. It allows organizations to place control of database development in the hands of database administrators(DBAs) and other specialists. A DBMS is a system software package that helps the use of integrated collection of data records and files known as databases. It allows different user application programs to easily access the same database. DBMSs may use any of a variety of database models, such as the network model or relational model.

In large systems, a DBMS allows users and other software to store and retrieve data in a structured way. Instead of having to write computer programs to extract information, user can ask simple questions in a query language. Thus, many DBMS packages provide Fourth-generation programming language(4GLs) and other application development features. It helps to specify the logical organization for a database and access and use the information within a database. It provides facilities for controlling data access, enforcing data integrity, managing concurrency, and restoring the database from backups.

A DBMS also provides the ability to logically present database information to users. The various functions of DBMS are Data Definition, Data manipulation, Data security and integrity, Data recovery and concurrency and Data dictionary maintenance.

We can remove the unwanted data in the database, is NORMALIZATION, elimination of redundant data. We have types in normalization like 0NF,1NF,2NF,3NF, Boyce-codd normal form.

The benefits of normalization are:

1. Less storage space
2. Quicker updates
3. Less data inconsistency
4. Easier to add data
5. Clearer data relationships

A trigger is an application which consists of event, condition, and action in an organized manner. It will be activated only when there is some insertion, deletion or modification of content present in database. There are 2 types of triggers row-level trigger and statement level trigger.

### Applications of DBMS

1. Data independence
2. Efficient data access
3. Data integrity
4. Data security
5. Concurrent access
6. Crash recovery.

### Some Examples of DBMS are:

1. Banking
2. Finance
3. Human resource
4. Tele communication
5. Airlines
6. Online ticket booking
7. E-seva

We can hide the data in the database i.e. the Data Abstraction. There are certain levels in the data abstraction, they are Physical level, Logical level and View level. The overall design of a database is called as SCHEMA. The data in the database is stored in the form of entities, the tables. We can perform certain operations of DDL, DML commands for the entities in the database like create, insert, delete, rename, alter, truncate, drop, update. A VIEW is a table whose rows are not explicitly stored in the database but are computed as needed, it can be used just like a base table. We can perform operations like create, update, delete, drop.

An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called ER diagrams.



## **PREFACE**

In order to develop the database systems, this lab enables the students to practice the efficient and effective designing of databases by using SQL Queries on Oracle. A DBMS also provides the ability to logically present database information to users. The purpose of this laboratory manual is to introduce undergraduate students to understand the benefits of database designing for effective retrieval of information from the records. A DBMS is a system software package that helps the use of integrated collection of data records and files known as databases. It includes SQL Query writing, create views, Indexing, procedures and functions and transaction control language. The manual contains procedures, and pre-experiment questions to help students prepare for experiments.

This practical manual will be helpful for students of Computer Science & Engineering for understanding the course from the point of view of applied aspects. Though all the efforts have been made to make this manual error free, yet some errors might have crept in inadvertently.

**Dr. Kakoli Banerjee**

Associate Professor, Dept. of CSE

## **DO'S AND DON'TS**

### **DO's**

1. Conform to the academic discipline of the department.
2. Enter your credentials in the laboratory attendance register.
3. Read and understand how to carry out an activity thoroughly before coming to the laboratory.
4. Ensure the uniqueness with respect to the methodology adopted for carrying out the experiments.
5. Shut down the machine once you are done using it.

### **DON'TS**

1. Eatables are not allowed in the laboratory.
2. Usage of mobile phones is strictly prohibited.
3. Do not open the system unit casing.
4. Do not remove anything from the computer laboratory without permission.
5. Do not touch, connect or disconnect any plug or cable without your faculty/laboratory technician's permission.

## **GENERAL SAFETY INSTRUCTIONS**

1. Know the location of the fire extinguisher and the first aid box and how to use them in case of an emergency.
2. Report fires or accidents to your faculty /laboratory technician immediately.
3. Report any broken plugs or exposed electrical wires to your faculty/laboratory technician immediately.
4. Do not plug in external devices without scanning them for computer viruses.

**INDEX**

<b>S.No</b>	<b>DATE OF CONDUCTION</b>	<b>EXPT. No</b>	<b>TITLE OF THE EXPERIMENT</b>	<b>PAGE No.</b>	<b>MARKS AWARDED (20)</b>	<b>FACULTY SIGNATURE WITH REMARK</b>


## **GUIDELINES FOR LABORATORY RECORD PREPARATION**

While preparing the lab records, the student is required to adhere to the following guidelines:

Contents to be included in Lab Records:

1. Cover page
2. Vision
3. Mission
4. PEOs
5. POs
6. PSOs
7. COs
8. CO-PO-PSO mapping
9. Index
10. Experiments
  - Aim
  - Source code
  - Input-Output

A separate copy needs to be maintained for pre-lab written work

The student is required to make the Lab File as per the format given on the next two pages.

**JSS ACADEMY OF TECHNICAL EDUCATION**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**DATABASE MANAGEMENT SYSTEM LAB FILE (KCS-551)**

<b>Name</b>	
<b>Roll No.</b>	
<b>Section- Batch</b>	

**GUIDELINES FOR ASSESSMENT**

Students are provided with the details of the experiment (Aim, pre-experimental questions, procedure etc.) to be conducted in next lab and are expected to come prepared for each lab class.

Faculty ensures that students have completed the required pre-experiment questions and they complete the in-lab programming assignment(s) before the end of class. Given that the lab programs are meant to be formative in nature, students can ask faculty for help before and during the lab class.

Students' performance will be assessed in each lab based on the following Lab Assessment Components:

**AC1:** Written Work (Max. marks = 4)

**AC2:** Fundamental Knowledge to conduct Experiment (Max. marks = 4)

**AC3:** Experiment Completed Successfully (Max. marks = 4)

**AC4:** Questions Answered (Max. marks = 4)

**AC5:** Punctuality (Max. marks = 4)

In each lab class, students will be awarded marks out of 4 under each component head, making it total out of 20 marks.



## EXPERIMENT - 1

**Aim:** Write the queries for Data Definition and Data Manipulation Language.

**Query 1:** Display the structure of the tables.

**Output:**

```
mysql> desc empl018;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | varchar(5) | YES  |     | NULL    |       |
| empname | char(10)   | YES  |     | NULL    |       |
| job    | char(10)   | YES  |     | NULL    |       |
| city   | char(10)   | YES  |     | NULL    |       |
| salary | int        | YES  |     | NULL    |       |
| deptno | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.04 sec)
```

```
mysql> desc projo18;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| pno   | varchar(5) | YES  |     | NULL    |       |
| pname | char(10)   | YES  |     | NULL    |       |
| eno   | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> desc dept018;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dno        | varchar(5)    | YES  |     | NULL    |       |
| dname      | char(10)      | YES  |     | NULL    |       |
| dlocation  | char(10)      | YES  |     | NULL    |       |
| pno        | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

**Query 2:** Insert the data into the tables using three different methods.

```
mysql> insert into empl018 values('e1', 'Charu', 'HR', 'Agra', 50000, 'd1');
Query OK, 1 row affected (0.01 sec)

mysql> insert into empl018(empno, empname, job, city, salary, deptno)values('e2',
-> 'Meenamkshi', 'IT', 'Jhansi', 40000, 'd2');
Query OK, 1 row affected (0.01 sec)

mysql> insert into empl018 values('e3','Ankita','Marketting','Delhi','60000','d3');
Query OK, 1 row affected (0.01 sec)

mysql> insert into empl018 values('e4','jai','Finance','Bangalore','40000','d4');
Query OK, 1 row affected (0.01 sec)

mysql> insert into empl018 values('e5','Anil','Sales','Bangalore','10000','d5');
Query OK, 1 row affected (0.01 sec)

mysql> insert into empl018 values('e6','Shivani','Sales','Bangalore','70000','d6');
Query OK, 1 row affected (0.03 sec)
```

**Query 3:**Select and display the records of the tables employee, project and department.

```
mysql> select * from empl018;
+-----+-----+-----+-----+-----+-----+
| empno | empname | job      | city      | salary | deptno |
+-----+-----+-----+-----+-----+-----+
| e1    | Charu   | HR       | Agra      | 50000  | d1     |
| e2    | Meenamkshi | IT      | Jhansi    | 40000  | d2     |
| e3    | Ankita  | Marketting | Delhi    | 60000  | d3     |
| e4    | jai     | Finance  | Bangalore | 40000  | d4     |
| e5    | Anil    | Sales    | Bangalore | 10000  | d5     |
| e6    | Shivani | Sales    | Bangalore | 70000  | d6     |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from projo18;
+-----+-----+-----+
| pno | pname | eno |
+-----+-----+-----+
| p1  | DBMS  | e1  |
| p1  | DBMS  | e5  |
| p2  | NW    | e3  |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from dept018;
+-----+-----+-----+-----+
| dno | dname | dlocation | pno |
+-----+-----+-----+-----+
| d1  | cse   | Delhi     | p1  |
| d1  | cse   | Delhi     | p2  |
| d2  | IT    | delhi     | p1  |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

**Query 4:** Display only employee no. and employee name from the table employee.

```
mysql> select empno,empname from empl018;
+-----+-----+
| empno | empname |
+-----+-----+
| e1    | Charu   |
| e2    | Meenamkshi |
| e3    | Ankita  |
| e4    | jai     |
| e5    | Anil    |
| e6    | Shivani |
+-----+-----+
6 rows in set (0.00 sec)
```

**Query 5:** Add a column age to table employee.

```
mysql> alter table empl018 add(age integer(3));
Query OK, 0 rows affected, 1 warning (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 1
```

```
mysql> desc empl018;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | varchar(5) | YES  |     | NULL    |       |
| empname | char(10)   | YES  |     | NULL    |       |
| job    | char(10)   | YES  |     | NULL    |       |
| city   | char(10)   | YES  |     | NULL    |       |
| salary | int        | YES  |     | NULL    |       |
| deptno | varchar(5) | YES  |     | NULL    |       |
| age    | int        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)
```

**Query 6:** Insert value into the new column age.

## JSS Academy of Technical Education – NOIDA

### Department of Computer Science and Engineering

```
mysql> insert into empl018(age)values(20);
Query OK, 1 row affected (0.01 sec)

mysql> select * from empl018;
+-----+-----+-----+-----+-----+-----+-----+
| empno | empname | job      | city      | salary | deptno | age |
+-----+-----+-----+-----+-----+-----+-----+
| e1    | Charu   | HR       | Agra      | 50000  | d1     | NULL |
| e2    | Meenamkshi | IT      | Jhansi    | 40000  | d2     | NULL |
| e3    | Ankita  | Marketting | Delhi    | 60000  | d3     | NULL |
| e4    | jai     | Finance  | Bangalore | 40000  | d4     | NULL |
| e5    | Anil    | Sales    | Bangalore | 10000  | d5     | NULL |
| e6    | Shivani | Sales    | Bangalore | 70000  | d6     | NULL |
| NULL  | NULL    | NULL     | NULL      | NULL   | NULL   | 20  |
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

**Query 7:** Delete a column age from the table employee.

```
mysql> alter table empl018 drop column age;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc empl018;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | varchar(5) | YES  |     | NULL    |       |
| empname | char(10)   | YES  |     | NULL    |       |
| job    | char(10)   | YES  |     | NULL    |       |
| city   | char(10)   | YES  |     | NULL    |       |
| salary | int        | YES  |     | NULL    |       |
| deptno | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

**Query 8:** Modify the attribute employee name by changing its datatype to varchar.

```
mysql> alter table empl018 modify column empname varchar(15);
Query OK, 7 rows affected (0.07 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> desc empl018;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | varchar(5) | YES  |     | NULL    |       |
| empname | varchar(15) | YES  |     | NULL    |       |
| job    | char(10)   | YES  |     | NULL    |       |
| city   | char(10)   | YES  |     | NULL    |       |
| salary | int        | YES  |     | NULL    |       |
| deptno | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

**Query 9:** Rename a table empl018 to emp018.

```
mysql> alter table empl018
-> rename to emp018;
Query OK, 0 rows affected (0.06 sec)
```

**Query 10:** Create a fake table with a row in it and truncate it.

```
mysql> create table faketable(sno varchar(2), name char(10));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into faketable values('1','mini');
Query OK, 1 row affected (0.01 sec)

mysql> select * from faketable;
+-----+-----+
| sno  | name |
+-----+-----+
| 1    | mini |
+-----+-----+
1 row in set (0.00 sec)

mysql> truncate table faketable;
Query OK, 0 rows affected (0.04 sec)
```

**Query 11:** Delete the data as well as the structure of the table.

```
mysql> drop table faketable;
Query OK, 0 rows affected (0.03 sec)
```

**Query 12:** Update the salary of employee by 1.25%

```
mysql> update emp018 set salary=salary*1.25 where salary=40000;
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2  Changed: 2  Warnings: 0
mysql> select * from emp018;
+-----+-----+-----+-----+-----+-----+
| empno | empname | job      | city      | salary | deptno |
+-----+-----+-----+-----+-----+-----+
| e1    | Charu   | HR       | Agra      | 50000  | d1     |
| e2    | Meenamkshi | IT      | Jhansi    | 50000  | d2     |
| e3    | Ankita  | Marketing | Delhi     | 60000  | d3     |
| e4    | jai     | Finance  | Bangalore | 50000  | d4     |
| e5    | Anil    | Sales    | Bangalore | 10000  | d5     |
| e6    | Shivani | Sales    | Bangalore | 70000  | d6     |
| NULL  | NULL    | NULL     | NULL      | NULL   | NULL   |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

**Query 13:** Delete the record from employee whose employee no. is e2

```
mysql> delete from emp018 where empno='e2';
Query OK, 1 row affected (0.01 sec)

mysql> select * from emp018;
```

empno	empname	job	city	salary	deptno
e1	Charu	HR	Agra	50000	d1
e3	Ankita	Marketing	Delhi	60000	d3
e4	jai	Finance	Bangalore	50000	d4
e5	Anil	Sales	Bangalore	10000	d5
e6	Shivani	Sales	Bangalore	70000	d6
NULL	NULL	NULL	NULL	NULL	NULL

```
6 rows in set (0.00 sec)
```

## EXPERIMENT 2

**Aim:** To explore 'select' statement using various clauses like where, order by, between, like, group by, having and logical/relational operators. To implement various DML statements.

### LOGICAL OPERATORS:

**Query 1:** Display the employees whose salary is more than 30,000 and live in Agra.

```
mysql> select empname from emp018 where salary>30000 and city='Agra';
+-----+
| empname |
+-----+
| Charu   |
+-----+
1 row in set (0.00 sec)
```

**Query 2:** Display the names of those employees who live in Agra or their salaries are more than 70,000.

```
mysql> select empname from emp018 where salary>70000 or city='Agra';
+-----+
| empname |
+-----+
| Charu   |
+-----+
1 row in set (0.00 sec)
```

**Query 3:** Display the details of employees who do not live in Jhansi.

```
mysql> select empname from emp018 where city!='Jhansi';
+-----+
| empname |
+-----+
| Charu   |
| Ankita  |
| jai     |
| Anil    |
| Shivani |
+-----+
5 rows in set (0.00 sec)
```

**Query 4:** Display the employee no., employee name and city of those employees whose employee no., are between e1 and e2.

```
mysql> select empno, empname, city from emp018 where empno between 'e1' and 'e2';
+-----+-----+-----+
| empno | empname | city |
+-----+-----+-----+
| e1    | Charu   | Agra |
+-----+-----+-----+
1 row in set (0.00 sec)
```

**Query 5:** Display those employees whose salaries are not more than 60,000.

```
mysql> select empname from emp018 where salary<60000;
+-----+
| empname |
+-----+
| Charu   |
| jai     |
| Anil    |
+-----+
3 rows in set (0.00 sec)
```

**Query 6:** Display the names of those employees whose salaries are more than 70,000.

```
mysql> select empname from emp018 where salary>70000;
Empty set (0.00 sec)
```

**Query 7:** Display the names of those employees who are working for department d4.

```
mysql> select empname from emp018 where deptno='d4';
+-----+
| empname |
+-----+
| jai     |
+-----+
1 row in set (0.00 sec)
```



**Query 8:** Display the employee no., employee name and salary from table employee where salary=50,000.

```
mysql> select empno,empname,salary from emp018 where salary=50000;
+-----+-----+-----+
| empno | empname | salary |
+-----+-----+-----+
| e1    | Charu   | 50000  |
| e4    | jai     | 50000  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Query 9:** Display the records of those whose salary is 50,000.

```
mysql> select * from emp018 where salary=50000;
+-----+-----+-----+-----+-----+-----+
| empno | empname | job      | city      | salary | deptno |
+-----+-----+-----+-----+-----+-----+
| e1    | Charu   | HR       | Agra      | 50000  | d1     |
| e4    | jai     | Finance  | Bangalore | 50000  | d4     |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Like command:**

**Query 1:** Display empno, empname of those employees whose name start from 'C'.

```
mysql> select empno,empname from emp018 where empname like 'C%';
+-----+-----+
| empno | empname |
+-----+-----+
| e1    | Charu   |
+-----+-----+
1 row in set (0.01 sec)
```

**Query 2:** Display empno,empname of those employee whose name contain substring 'hi'.

```
mysql> select * from emp018 where empname like '%hi%';
+-----+-----+-----+-----+-----+-----+
| empno | empname | job      | city      | salary | deptno |
+-----+-----+-----+-----+-----+-----+
| e6    | Shivani | Sales    | Bangalore | 70000  | d6     |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

**Query 3:** Display empno,empname of those employee whose name 3rd alphabet is 'i'.

```
mysql> select * from emp018 where empname like '__i%';
+-----+-----+-----+-----+-----+-----+
| empno | empname | job      | city      | salary | deptno |
+-----+-----+-----+-----+-----+-----+
| e4    | jai     | Finance  | Bangalore | 50000  | d4     |
| e5    | Anil    | Sales    | Bangalore | 10000  | d5     |
| e6    | Shivani | Sales    | Bangalore | 70000  | d6     |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

**In command:**

**Query 1:** Display empno, city & name of those employees who are either 'HR' or 'Director'.

```
mysql> select empno,empname,city from emp018
-> where job in ('HR','Director');
+-----+-----+-----+
| empno | empname | city |
+-----+-----+-----+
| e1    | Charu   | Agra |
+-----+-----+-----+
1 row in set (0.00 sec)
```

**Query 2:** Display empno, city & empname of those who do not live in agra.

```
mysql> select empno,city,empname from emp018 where city not in ('Agra');
+-----+-----+-----+
| empno | city      | empname |
+-----+-----+-----+
| e3    | Delhi     | Ankita  |
| e4    | Bangalore | jai     |
| e5    | Bangalore | Anil    |
| e6    | Bangalore | Shivani |
+-----+-----+-----+
4 rows in set (0.01 sec)
```

## JSS Academy of Technical Education – NOIDA

Department of Computer Science and Engineering

### Order by command

**Query:** Display the records of table employee in descending order of their salary

```
mysql> select * from emp018
-> order by salary desc;
```

empno	empname	job	city	salary	deptno
e6	Shivani	Sales	Bangalore	70000	d6
e3	Ankita	Marketting	Delhi	60000	d3
e1	Charu	HR	Agra	50000	d1
e4	jai	Finance	Bangalore	50000	d4
e5	Anil	Sales	Bangalore	10000	d5
NULL	NULL	NULL	NULL	NULL	NULL

6 rows in set (0.00 sec)

---

### Lower function

**Query:** Display the name of all employees in lower case.

```
mysql> select lower(empname) from emp018;
+-----+
| lower(empname) |
+-----+
| charu          |
| ankita         |
| jai            |
| anil           |
| shivani        |
| NULL           |
+-----+
6 rows in set (0.01 sec)
```

### Upper function

**Query:** Display the name of all employees in upper case.

```
mysql> select upper(empname) from emp018;
+-----+
| upper(empname) |
+-----+
| CHARU          |
| ANKITA         |
| JAI            |
| ANIL           |
| SHIVANI        |
| NULL           |
+-----+
6 rows in set (0.01 sec)
```

### Substring function

**Query:** Display first four characters of the designation that the employees hold

```
mysql> select substr(job,1,4) from emp018;
+-----+
| substr(job,1,4) |
+-----+
| HR              |
| Mark            |
| Fina            |
| Sale            |
| Sale            |
| NULL            |
+-----+
6 rows in set (0.00 sec)
```

### ASCII function

**Query:** Print the ascii value of first character of column city.

```
mysql> select ascii(city) from emp018;
+-----+
| ascii(city) |
+-----+
|           65 |
|           68 |
|           66 |
|           66 |
|           66 |
|          NULL |
+-----+
6 rows in set (0.00 sec)
```

### Ltrim command

**Query:** Remove all the white spaces from the left of the string.

```
mysql> select ltrim(' Meenakshi')from dual;
+-----+
| ltrim(' Meenakshi') |
+-----+
| Meenakshi           |
+-----+
1 row in set (0.01 sec)
```

### Rtrim command

**Query:** Remove all the white spaces from the right of the string.

```
mysql> select rtrim('Meenakshi ')from dual;
+-----+
| rtrim('Meenakshi ') |
+-----+
| Meenakshi           |
+-----+
1 row in set (0.00 sec)
```

### Concatenate command

**Query:** join the attribute of empname & city of all employees.

```
mysql> select concat(empname,city) from emp018;
+-----+
| concat(empname,city) |
+-----+
| CharuAgra            |
| AnkitaDelhi          |
| jaiBangalore         |
| AnilBangalore        |
| ShivaniBangalore     |
| NULL                 |
+-----+
6 rows in set (0.01 sec)
```

### Replace command

**Query9:** replace the attribute empname which contain the string.

```
mysql> select replace(empname, 'an','am')from emp018;
+-----+
| replace(empname, 'an','am') |
+-----+
| Charu                        |
| Ankita                      |
| jai                         |
| Anil                       |
| Shivami                    |
| NULL                       |
+-----+
6 rows in set (0.00 sec)
```

### **NUMERIC FUNCTIONS:**

#### **Absolute function**

**Query:** display the value of -8.5 in dual

```
mysql> select abs(-8.5) from dual;
+-----+
| abs(-8.5) |
+-----+
|         8.5 |
+-----+
1 row in set (0.00 sec)
```

#### **Title: Round function**

**Query:** display the rounded value of 23.45 to two decimal places

```
mysql> select round(23.45) from dual;
+-----+
| round(23.45) |
+-----+
|          23 |
+-----+
1 row in set (0.00 sec)
```

#### **Power command**

**Query:** display the value of pow (2,3) in dual.

```
mysql> select power(2,3) from dual;
+-----+
| power(2,3) |
+-----+
|          8 |
+-----+
1 row in set (0.01 sec)
```

#### **Square root function**

**Query:**display square root of 65536 from dual

```
mysql> select sqrt(65536)from dual;
+-----+
| sqrt(65536) |
+-----+
|          256 |
+-----+
1 row in set (0.00 sec)
```

### Power exponent

**Query:** display the value of exponent raised to 3.3 from dual

```
mysql> select exp(3.3)from dual;
+-----+
| exp(3.3) |
+-----+
| 27.112638920657883 |
+-----+
1 row in set (0.01 sec)
```

### Greatest function

**Query:** display largest number from dual.

```
mysql> select greatest(2,3,100,200,300,400)from dual;
+-----+
| greatest(2,3,100,200,300,400) |
+-----+
| 400 |
+-----+
1 row in set (0.01 sec)
```

### Least function

**Query:** display the least number among the dual.

```
mysql> select least(-1,2,0,4,5)from dual;
+-----+
| least(-1,2,0,4,5) |
+-----+
| -1 |
+-----+
1 row in set (0.00 sec)
```

### Mod command

**Query:** display the remainder of 15/7.

```
mysql> select mod(15,7)from dual;
+-----+
| mod(15,7) |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```



### Truncate command

**Query:** display the value of 1284.6189 truncate to one decimal place

```
mysql> select truncate(1284.6189,1) from dual;
+-----+
| truncate(1284.6189,1) |
+-----+
|           1284.6      |
+-----+
1 row in set (0.00 sec)
```

### Sin function.

**Query:** display the sine value of 60.

```
mysql> select sin(60)from dual;
+-----+
| sin(60)          |
+-----+
| -0.3048106211022167 |
+-----+
1 row in set (0.00 sec)
```

### cos function.

**Query:** display the cos value of 32.

```
mysql> select cos(32)from dual;
+-----+
| cos(32)          |
+-----+
| 0.8342233605065102 |
+-----+
1 row in set (0.00 sec)
```

### Title: tan function.

**Query:** display the tan value of 92.

```
mysql> select tan(92)from dual;
+-----+
| tan(92)          |
+-----+
| 1.2442700581287092 |
+-----+
1 row in set (0.00 sec)
```

### **Aggregate Functions:**

#### **Average function**

```
mysql> select avg(salary) salary1 from emp018;
+-----+
| salary1 |
+-----+
| 48000.0000 |
+-----+
1 row in set (0.01 sec)
```

**Query:** Calculate the average salary of the employees from the table employee and name the calculated average as salary1.

#### **MAX Function**

**Query:** Calculate the maximum salary of the employees from the table employee and name the calculated maximum salary as maxsalary.

```
mysql> select max(salary) maxsalary from emp018;
+-----+
| maxsalary |
+-----+
|      70000 |
+-----+
1 row in set (0.00 sec)
```

#### **MIN Function**

**Query:** Calculate the minimum salary of the employees from the table employee and name the calculated maximum salary as minsalary.

```
mysql> select min(salary) minsalary from emp018;
+-----+
| minsalary |
+-----+
|      10000 |
+-----+
1 row in set (0.00 sec)
```

### Count Function

**Query:** Calculate the total number of records in the table employee.

```
mysql> select count(empno) from emp018;
+-----+
| count(empno) |
+-----+
|           5 |
+-----+
1 row in set (0.00 sec)
```

### SUM Function

**Query:** Calculate the total salary of all the employees in the table employee.

```
mysql> select sum(salary) from emp018;
+-----+
| sum(salary) |
+-----+
|      240000 |
+-----+
1 row in set (0.00 sec)
```

### EXPERIMENT NO. 3

**Aim** : To implement nested and correlated queries and to create various views and indexes and to implement Cursors and triggers using PL/SQL.

**Query** : Get the details of the employees working on 'p1'.

```
mysql> select empno,empname from emp018,proj018 where emp018.empno=proj018.eno and
-> pno='p1';
+-----+-----+
| empno | empname |
+-----+-----+
| e1    | Charu   |
| e5    | Anil    |
+-----+-----+
2 rows in set (0.01 sec)
```

Query : Get the details of the employees working on p1 and p2.

```
mysql> select empno,empname from emp018,proj018 where emp018.empno=proj018.eno and
-> (pno='p1' or pno='p2');
+-----+-----+
| empno | empname |
+-----+-----+
| e1    | Charu   |
| e3    | Ankita  |
| e5    | Anil    |
+-----+-----+
3 rows in set (0.00 sec)
```

Query : Perform LEFT OUTER JOIN on table project and assignedto.

```
mysql> select * from proj018
-> left join
-> asto018
-> using(pno);
+-----+-----+-----+-----+
| pno | pname | eno | eno |
+-----+-----+-----+-----+
| p1  | DBMS  | e1  | e1  |
| p1  | DBMS  | e1  | e5  |
| p1  | DBMS  | e1  | e4  |
| p1  | DBMS  | e1  | e3  |
| p1  | DBMS  | e5  | e1  |
| p1  | DBMS  | e5  | e5  |
| p1  | DBMS  | e5  | e4  |
| p1  | DBMS  | e5  | e3  |
| p2  | NW    | e3  | e3  |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

## JSS Academy of Technical Education – NOIDA

Department of Computer Science and Engineering

Query : Perform RIGHT OUTER JOIN on table project and assignedto.

```
mysql> select * from projo18
-> right join
-> asto018
-> using(pno);
```

pno	eno	pname	eno
p2	e3	NW	e3
p4	e6	NULL	NULL
p6	e1	NULL	NULL
p8	e4	NULL	NULL
p1	e3	DBMS	e5
p1	e3	DBMS	e1
p1	e4	DBMS	e5
p1	e4	DBMS	e1
p1	e5	DBMS	e5
p1	e5	DBMS	e1
p1	e1	DBMS	e5
p1	e1	DBMS	e1

```
12 rows in set (0.00 sec)
```

## JSS Academy of Technical Education – NOIDA

### Department of Computer Science and Engineering

Query: Get the details of the employees working on 'p1'

```
mysql> select empno,empname from emp018 where empno in(select eno from asto018 where
-> pno='p1');
+-----+-----+
| empno | empname |
+-----+-----+
| e1    | Charu   |
| e3    | Ankita  |
| e4    | jai     |
| e5    | Anil    |
+-----+-----+
4 rows in set (0.02 sec)
```

Query : Get the details of the employees working on 'p1' or 'p2'.

```
mysql> select empno,empname from emp018 where exists(select * from asto018 where
-> (pno='p1' or pno='p2')and asto018.eno=emp018.empno);
+-----+-----+
| empno | empname |
+-----+-----+
| e1    | Charu   |
| e3    | Ankita  |
| e4    | jai     |
| e5    | Anil    |
+-----+-----+
4 rows in set (0.00 sec)
```

Query : Get the employee no. of the employees who do not work on 'p1'.

```
mysql> select distinct empno from emp018 where empno not in(select eno from
-> asto018 where pno='p1');
+-----+
| empno |
+-----+
| e6    |
+-----+
1 row in set (0.01 sec)
```

Query : Get the details of the employees who work on all the projects.

```
mysql> select empno from emp018 where not exists(select * from projo18 where not
-> exists(select * from asto018 where asto018.eno=emp018.empno and
-> asto018.pno=projo18.pno));
+-----+
| empno |
+-----+
| e3    |
+-----+
1 row in set (0.00 sec)
```

# JSS Academy of Technical Education – NOIDA

## Department of Computer Science and Engineering

### Views:

Query:- Create a view to display details of employee not working on any project.

```
mysql> create view emppro as select * from emp018 where empno NOT IN(select empno from  
-> asto018);  
Query OK, 0 rows affected (0.02 sec)
```

Query:- Create a view to display employee name and project name not working on any project.

```
mysql> create view myproj as select empname,pno from emp018,asto018 where  
-> emp018.empno=asto018.eno AND pno IN('p11','p23');  
Query OK, 0 rows affected (0.01 sec)
```

Query:-Drop the view created in the table myproj.

```
mysql> drop view myproj;  
Query OK, 0 rows affected (0.01 sec)
```

### Union:

Query:-Find the employee who are working on project p11 or p21.

```
mysql> select eno from asto018 where pno='p1' UNION select eno from asto018 where  
-> pno='p2';  
+-----+  
| eno |  
+-----+  
| e3 |  
| e4 |  
| e5 |  
| e1 |  
+-----+  
4 rows in set (0.00 sec)
```

### Implementation of Restrictions

Query:-Create a table Employee with all fields as mandatory (not null) and make empno as the primary key.

```
mysql> create table em018 (empno varchar(20) PRIMARY KEY,empname varchar(20) not  
-> null,salary integer(6) not null);  
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

Query:-Create a table Employee such that empno. are unique across the entire column and rest all fields are not null.

```
mysql> create table emp18 (empno varchar(20) UNIQUE,empname varchar(20) not  
-> null,salary integer(6) not null);  
Query OK, 0 rows affected, 1 warning (0.07 sec)
```

Query: Add a constraint to the salary attribute in employee such that minimum salary is 1000.

```
mysql> alter table emp18  
-> add constraint checker  
-> check(salary>1000);  
Query OK, 0 rows affected (0.11 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Query:-Add unique key to employee name in the existing table.

```
mysql> alter table emp18 add constraint uk unique(empname);  
Query OK, 0 rows affected (0.09 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Query:-Drop unique key constraint from employee table.

```
mysql> alter table emp18 drop constraint uk;  
Query OK, 0 rows affected (0.07 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Query:-Drop primary key constraint from employee table.

```
mysql> alter table emp18 drop constraint pk;  
ERROR 3940 (HY000): Constraint 'pk' does not exist.
```



# **JSS Academy of Technical Education – NOIDA**

**Department of Computer Science and Engineering**