



LABORATORY MANUAL

B.Tech. Semester- IV

DATABASE MANAGEMENT SYSTEMS LAB

Subject code: LC-CSE-209G

Prepared by:

Dr Shipra Arora

Checked by:

Dr Ritu Pahwa

Approved by:

Name : Prof. (Dr.) Isha Malhotra

Sign.:

Sign.:

Sign.:

**DEPARTMENT OF CSE(AI&ML)
DRONACHARYA COLLEGE OF ENGINEERING
KHENTAWAS, FARRUKH NAGAR, GURUGRAM (HARYANA)**

Table of Contents

1. Vision and Mission of the Institute
2. Vision and Mission of the Department
3. Programme Educational Objectives (PEOs)
4. Programme Outcomes (POs)
5. Programme Specific Outcomes (PSOs)
6. University Syllabus
7. Course Outcomes (COs)
8. CO- PO and CO-PSO mapping
9. Course Overview
10. List of Experiments
11. DOs and DON'Ts
12. General Safety Precautions
13. Guidelines for students for report preparation
14. Lab assessment criteria
15. Lab Experiments

Vision and Mission of the Institute

Vision:

“To impart Quality Education, to give an enviable growth to seekers of learning, to groom them as World Class Engineers and managers competent to match the expending expectations of the Corporate World has been ever enlarging vision extending to new horizons of Dronacharya College of Engineering”

Mission:

- M1:** To prepare students for full and ethical participation in a diverse society and encourage lifelong learning by following the principle of ‘Shiksha evam Sahayata’ i.e., Education & Help.
- M2:** To impart high-quality education, knowledge and technology through rigorous academic programs, cutting-edge research, & Industry collaborations, with a focus on producing engineers& managers who are socially responsible, globally aware, & equipped to address complex challenges.
- M3:** Educate students in the best practices of the field as well as integrate the latest research into the academics.
- M4:** Provide quality learning experiences through effective classroom practices, innovative teaching practices and opportunities for meaningful interactions between students and faculty.
- M5:** To devise and implement programmes of education in technology that are relevant to the changing needs of society, in terms of breadth of diversity and depth of specialization.

Vision and Mission of the Department

Vision:

To cultivate skills and make proficient engineers cum trainers in the domain of Artificial Intelligence & Machine Learning for exceptional contributions to the society.

Mission:

M1: To impart intense training and learning to generate knowledge through the state-of-the-art concepts and technologies in Artificial Intelligence and Machine Learning.

M2: To establish centres of excellence by collaborating with the leading industries to exhilarate innovative research and development in AIML and its allied technology.

M3: To inculcate regenerative self-learning abilities, team spirit, and professional ethics among the students for noble cause.

Programme Educational Objectives (PEOs)

PEO1- ANALYTICAL SKILLS:

Using a solid foundation in mathematical, scientific, engineering, and current computing principles, formulate, analyse, and resolve engineering issues in real-world domain.

PEO2- TECHNICAL SKILLS:

Apply artificial intelligence theory and concepts to analyse the requirements, realise technical specifications, and design engineering solutions.

PEO3- SOFT SKILLS:

Through inter-disciplinary projects and a variety of professional activities, demonstrate technical proficiency, AI competency, and foster collaborative learning and a sense of teamwork.

PEO4- PROFESSIONAL ETHICS:

Excel as socially responsible engineers or entrepreneurs with high moral and ethical standards, competence, and soft skills that will enable them to contribute to societal demands and achieve sustainable advancement in emerging computer technologies.

Database Management System Lab (LC-CSE-209G)

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 of mathematics, 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 modeling 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 teamwork:** 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 SPECIFIC OUTCOMES (PSOs)

- PSO1: Fundamentals and critical knowledge of the Computer System:**
Apply the knowledge gained pertaining to build, asses, and analyze the software and hardware aspects of the program to solve real world business problems.
- PSO2: Comprehensive and applicative knowledge of Software Development:**
Ability to evaluate and apply knowledge of data engineering, methodologies, and able to plan, develop, test, analyze, and manage required aspects in heterogenous platforms individually or in team work.
- PSO3: Applications in Computing Domain:**
Ability to acquire computational knowledge and project development abilities using novel tools and methodologies to tackle challenges in the fields related to Deep Learning, Machine learning, Artificial Intelligence.
- PSO4: Applications in Innovations and Research:**
Capacity to direct a team or firm that develops products and to use the knowledge learned to recognise actual research issues

University Syllabus

Contents:

- i. Creation of a database and writing SQL queries to retrieve information from the database.
- ii. Performing Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.
- iii. Creation of Views, Synonyms, Sequence, Indexes, Save point.
- iv. Creating an Employee database to set various constraints.
- v. Creating relationship between the databases.
- vi. Study of PL/SQL block.
- vii. Write a PL/SQL block to satisfy some conditions by accepting input from the user.
- viii. Write a PL/SQL block that handles all types of exceptions.
- ix. Creation of Procedures.
- x. Creation of database triggers and functions
- xi. Mini project (Application Development using Oracle/ MySQL)
 - a. Inventory Control System
 - b. Material Requirement Processing.
 - c. Hospital Management System.
 - d. Railway Reservation System.
 - e. Personal Information System.
 - f. Web Based User Identification System.
 - g. Time Table Management System.
 - h. Hotel Management

Course Outcomes (COs)

Upon successful completion of the course, the students will be able to:

1. Get practical knowledge on designing and creating relational database systems
2. Implement basic SQL DDL Queries
3. Implement basic SQL DML Queries
4. Understand various advanced queries execution such as relational constraints, joins, set operations, aggregate functions, trigger, view and embedded SQL.
5. To design and implement database applications on their own.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	1		1		1	2	2	1	2
CO2	1	1	1						1	1		
CO3	2	2	2						2	2		
CO4	3	3	3	3	3	3			3	3	3	3
CO5	3	3	3	3	3	3		3	3	3	3	3

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	3	1
CO2	1	1	1	1
CO3	1	1	2	2
CO4	2	2	3	3
CO5	3	3	3	3

*3-HIGH
*2-MEDIUM
*1-LOW

Course Overview

Database management has evolved from a specialized computer application to a central component of a modern computing environment and as a result knowledge about database system has become an essential part of computer science. The course serves as a visual guide to the material presented during our lectures. The aim of this course is to provide an introduction to Database management system, with an emphasis on foundational material.

DBMS is a computer application software that provides a way to manage data. The requirement of modern days is to have an automated system that manages, modifies, and updates data perfectly. This is accomplished by a DBMS in robust, accurate, and non-redundant way. Consequently, Oracle, Sybase, Microsoft SQL Server, etc. have emerged as prominent commercial systems while MySQL, PostgreSQL etc. lead as open source software. This Course permits students to apply the conceptual design model to construct the real-world problems. Course provides familiarity of Database Concepts so that students can analyze the various different constraints to populate the database and examine different DBMS concepts to deduce the most suitable pattern of documentation. DBMS lab with mini project targets at practicing and accomplishing this aim by using MySQL as well as gain proficiency to design database and its hierarchical structure for given real world application.

Database Management System Lab (LC-CSE-209G)

List of Experiments mapped with COs

Sr. No.	Program	COs
1	Creation of database and SQL Queries to retrieve information from database	CO1,CO2
2	Performing insertion, deletion, modification, altering and updating operations on the records based on conditions.	CO1,CO3
3	To Manipulate the Operations on the table	CO1,CO2,CO3
4	To Implement the structure of the table	CO1,CO2
5	To Implement the restrictions on the table.	CO2,CO3,CO4
6	To implement the concept of Joins.	CO2,CO3,CO4
7	To implement the concept of grouping of Data.	CO2,CO3,CO4
8	To implement the concept of Sub-Queries.	CO2,CO3,CO4
9	To implement the concept of Indexes and views.	CO2,CO3,CO4
10	To implement the basics of PL/SQL.	CO2,CO3,CO4
	Mini Project(Application Development using Oracle/MySQL) Inventory Control System Material Requirement Processing Hospital Management System Railway Reservation System Personal information System Web Based User Identification System Time Table Management System Hotel Management	CO1,CO2,CO3,CO4 ,CO5

DOs and DON'Ts

DOs

1. Login-on with your username and password.
2. Log off the Computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON'Ts

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.
4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible, isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.
4. Call security and emergency department immediately:

Emergency : 201 (Reception)

Security: 231 (Gate No.1)

Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows: -

- 1) All files must contain a title page followed by an index page. *The files will not be signed by the faculty without an entry in the index page.*
- 2) Student's Name, roll number and date of conduction of experiment must be written on all pages.
- 3) For each experiment, the record must contain the following
 - (i) Aim/Objective of the experiment
 - (ii) Pre-experiment work (as given by the faculty)
 - (iii) Lab assignment questions and their solutions
 - (iv) Test Cases (if applicable to the course)
 - (v) Results/ output

Note:

1. Students must bring their lab record along with them whenever they come for the lab.
2. Students must ensure that their lab record is regularly evaluated.

Database Management System Lab (LC-CSE-209G)

Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	Exemplary (4)	Competent (3)	Needs Improvement (2)	Poor (1)
AC1: Pre-Lab written work (this may be assessed through viva)	Complete procedure with underlined concept is properly written	Underlined concept is written but procedure is incomplete	Not able to write concept and procedure	Underlined concept is not clearly understood
AC2: Program Writing/ Modeling	Unable to understand the reason for errors/ bugs even after they are explicitly pointed out	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied	Assigned problem is properly analyzed & correct solution designed	Assigned problem is properly analyzed
AC3: Identification & Removal of errors/ bugs	Able to identify errors/ bugs and remove them	Able to identify errors/ bugs and remove them with little bit of guidance	Is dependent totally on someone for identification of errors/ bugs and their removal	Unable to understand the reason for errors/ bugs even after they are explicitly pointed out
AC4: Execution & Demonstration	All variants of input /output are tested, Solution is well demonstrated and implemented concept is clearly explained	All variants of input /output are not tested, However, solution is well demonstrated and implemented concept is clearly explained	Only few variants of input /output are tested, Solution is well demonstrated but implemented concept is not clearly explained	Solution is not well demonstrated and implemented concept is not clearly explained
AC5: Lab Record Assessment	All assigned problems are well recorded with objective, design constructs and solution along with Performance analysis using all variants of input and output	More than 70 % of the assigned problems are well recorded with objective, design constructs and solution along with Performance analysis is done with all variants of input and output	Less than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	

LAB EXPERIMENTS

Database Management System Lab (LC-CSE-209G)

Practical #1

Objective: *Create tables and specify the Questionnaires in SQL.*

Theory & Concepts:

Introduction about SQL-

SQL (Structured Query Language) is a nonprocedural language, you specify what you want, not how to get it. A block structured format of English key words is used in this Query language. It has the following components.

DDL (Data Definition Language)-

The SQL DDL provides command for defining relation schemas, deleting relations and modifying relation schema.

DML (DATA Manipulation Language)-

It includes commands to insert tuples into, delete tuples from and modify tuples in the database.

View definition-

The SQL DDL includes commands for defining views.

Transaction Control- SQL includes for specifying the beginning and ending of transactions.

Embedded SQL and Dynamic SQL-

Embedded and Dynamic SQL define how SQL statements can be embedded with in general purpose programming languages, such as C, C++, JAVA, COBOL, Pascal and Fortran.

Integrity-

The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must specify. Updates that violate integrity constraints are allowed.

Authorization-

The SQL DDL includes commands for specifying access rights to relations and views.

Data Definition Language-

The SQL DDL allows specification of not only a set of relations but also information about each relation, including-

- Schema for each relation
- The domain of values associated with each attribute.
- The integrity constraints.
- The set of indices to be maintained for each relation.
- The security and authorization information for each relation.

Database Management System Lab (LC-CSE-209G)

- The physical storage structure of each relation on disk.

Domain types in SQL-

The SQL standard supports a variety of built in domain types, including-

- Char (n) - A fixed length character length string with user specified length.
- Varchar (n) - A variable character length string with user specified maximumlength n.
- Int- An integer.
- Small integer- A small integer.
- Numeric (p, d)-A Fixed point number with user defined precision.
- Real, double precision- Floating point and double precision floating point numbers with machine dependent precision.
- Float (n) - A floating point number, with precision of at least n digits.
- Date- A calendar date containing a (four digit) year, month and day of the month.
- Time- The time of day, in hours, minutes and seconds Eg. Time '09:30:00'.
- Number- Number is used to store numbers (fixed or floating point).

DDL statement for creating a table-

Syntax-

Create table tablename
(columnname datatype(size), columnname datatype(size));

Creating a table from a table-

Syntax-

```
CREATE TABLE TABLENAME  
[(columnname, columnname, .....)]  
AS SELECT columnname, columnname..... FROM tablename;
```

Database Management System Lab (LC-CSE-209G)

OUTPUT:-

```
mysql>
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.05 sec)

mysql> create database sujal_24282;
Query OK, 1 row affected (0.02 sec)

mysql> use sujal_24282;
Database changed
mysql> create table students(rollno int primary key, s_name varchar(20), gender varchar(1), dob date, address1 varchar(
0), address2 varchar(20), city varchar(20), percentage int);
Query OK, 0 rows affected (0.05 sec)

mysql> desc students;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| rollno | int | NO | PRI | NULL | |
| s_name | varchar(20) | YES | | NULL | |
| gender | varchar(1) | YES | | NULL | |
| dob | date | YES | | NULL | |
| address1 | varchar(20) | YES | | NULL | |
| address2 | varchar(20) | YES | | NULL | |
| city | varchar(20) | YES | | NULL | |
| percentage | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.02 sec)

mysql>
```

Database Management System Lab (LC-CSE-209G)

Practical #2

Performing insertion, deletion, modification, altering and updating operations on the records based on conditions.

Insertion of data into tables-

Syntax-

```
INSERT INTO tablename  
[(columnname, columnname,  
.....)]Values(expression,  
expression);
```

Inserting data into a table from another table:

Syntax-

```
INSERT INTO tablename  
SELECT columnname, columnname,  
.....FROM tablename;
```

Insertion of selected data into a table from another table:

Syntax-

```
INSERT INTO tablename  
SELECT columnname, columnname.....  
FROM tablename  
WHERE columnname= expression;
```

Retrieving of data from the tables-

Syntax-

```
SELECT * FROM tablename;
```

The retrieving of specific columns from a table-

Syntax-

```
SELECT columnname, columnname,  
....FROM tablename;
```

Elimination of duplicates from the select statement-

Syntax-

```
SELECT DISTINCT columnname,  
columnnameFROM tablename;
```

Database Management System Lab (LC-CSE-209G)

Selecting a data set from table data-

Syntax-

SELECT columnname,
columnnameFROM tablename
WHERE searchcondition;

OUTPUT:-

```
mysql> create table employee(Empno int primary key, Empname varchar(20), Job varchar(30), Mgr int, Sal int);
Query OK, 0 rows affected (0.03 sec)

mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Empno | int           | NO   | PRI | NULL    |       |
| Empname | varchar(20)   | YES  |     | NULL    |       |
| Job     | varchar(30)   | YES  |     | NULL    |       |
| Mgr     | int           | YES  |     | NULL    |       |
| Sal     | int           | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> insert into employee values(1232132, "Sujal", "Developer", 5242218, 100000);
Query OK, 1 row affected (0.02 sec)

mysql> insert into employee values(1232526, "Piyush", "Junior Developer", 2772318, 150000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(1256836, "Satyam", "Junior Developer", 2777638, 140000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(3726836, "Jitesh", "Senior Developer", 3763738, 200000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(2988336, "Divyansh", "Team Leader", 8783823, 250000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| Empno | Empname | Job          | Mgr | Sal |
+-----+-----+-----+-----+-----+
| 1232132 | Sujal   | Developer    | 5242218 | 100000 |
| 1232526 | Piyush  | Junior Developer | 2772318 | 150000 |
| 1256836 | Satyam  | Junior Developer | 2777638 | 140000 |
| 2988336 | Divyansh | Team Leader   | 8783823 | 250000 |
| 3726836 | Jitesh  | Senior Developer | 3763738 | 200000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Database Management System Lab (LC-CSE-209G)

```
mysql> update employee set Job= Junior Developer where Empno = 1232132;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| Empno | Empname | Job          | Mgr   | Sal   |
+-----+-----+-----+-----+-----+
| 1232132 | Sujal   | Junior Developer | 5242218 | 100000 |
| 1232526 | Piyush  | Junior Developer | 2772318 | 150000 |
| 1256836 | Satyam  | Junior Developer | 2777638 | 140000 |
| 2988336 | Divyansh | Team Leader    | 8783823 | 250000 |
| 3726836 | Jitesh  | Senior Developer | 3763738 | 200000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> alter table employee rename column Mgr to SSN;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| Empno | Empname | Job          | SSN   | Sal   |
+-----+-----+-----+-----+-----+
| 1232132 | Sujal   | Junior Developer | 5242218 | 100000 |
| 1232526 | Piyush  | Junior Developer | 2772318 | 150000 |
| 1256836 | Satyam  | Junior Developer | 2777638 | 140000 |
| 2988336 | Divyansh | Team Leader    | 8783823 | 250000 |
| 3726836 | Jitesh  | Senior Developer | 3763738 | 200000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Q1. Create the following tables:

i) client master

<u>columnname</u>	<u>datatype</u>	<u>size</u>
<u>client_no</u>	varchar2	6
<u>name</u>	varchar2	20
<u>address1</u>	varchar2	30
<u>address2</u>	varchar2	30
<u>city</u>	varchar2	15
<u>state</u>	varchar2	15
<u>pincode</u>	number	6
<u>bal_due</u>	number	10,2

ii) Product master

<u>Columnname</u>	<u>datatype</u>	<u>size</u>
<u>Product_no</u>	varchar2	
<u>Description</u>	varchar2	
<u>Profit_percent</u>	number	
<u>Unit_measure</u>	varchar2	
<u>Qty_on_hand</u>	number	
<u>Reoder lvnumber</u>		
<u>Sell_price</u>	number	
<u>Cost_price</u>	number	

Database Management System Lab (LC-CSE-209G)

Q2- Insert the following data into their respective tables:

Client No	Name	City	Pincode	State	Balance Due
0001	Ivan	Bombay	400054	Maharashtra	15000
0002	Vandana	Madras	780001	Tamilnadu	0
0003	Pramada	Bombay	400057	Maharashtra	5000
0004	Basu	Bombay	400056	Maharashtra	0
0005	Ravi	Delhi	100001		2000
0006	Rukmini	Bombay	400050	Maharashtra	0

Data for Product Master:

Product No.	Description	Profit %	Unit Measure	Qty_on_hand	Reorder_lvl	Selling Price	Cost Price
P00001	1.44floppies	5	piece	100	20	525	500
P03453	Monitors	6	piece	10	3	12000	11200
P06734	Mouse	5	piece	20	5	1050	500
P07865	1.22 floppies	5	piece	100	20	525	500
P07868	Keyboards	2	piece	10	3	3150	3050
P07885	CD Drive	2.5	piece	10	3	5250	5100
P07965	540 HDD	4	piece	10	3	8400	8000
P07975	1.44 Drive	5	piece	10	3	1050	1000
P08865	1.22 Drive	5	piece	2	3	1050	1000

Q3:- On the basis of above two tables answer the following Questionries:

- i) Find out the names of all the clients.
- ii) Retrieve the list of names and cities of all the clients.
- iii) List the various products available from the product_master table.
- iv) List all the clients who are located in Bombay.
- v) Display the information for client no 0001 and 0002.
- vi) Find the products with description as '1.44 drive' and '1.22 Drive'.
- vii) Find all the products whose sell price is greater then 5000.
- viii) Find the list of all clients who stay in in city 'Bombay' or city 'Delhi' or'Madras'.
- ix) Find the product whose selling price is greater than 2000 and less than orequal to 5000.
- x) List the name, city and state of clients not in the state of 'Maharashtra'.

Database Management System Lab (LC-CSE-209G)

Practical #3

Objective:- To Manipulate the Operations on the table.

DML (Data Manipulation Language) Data manipulation is

- The retrieval of information stored in the database.
- The insertion of new information into the database.
- The deletion of information from the database.
- The modification of information stored by the appropriate data model.

There are basically two types.

- (i) **Procedural DML**:- require a user to specify what data are needed and how to get those data.
- (ii) **Non Procedural DML** : require a user to specify what data are needed without specifying how to get those data.

Updating the content of a table:

In creation situation we may wish to change a value in table without changing all values in the tuple . For this purpose the update statement can be used.

Update table name

Set columnname = expression, columnname

=expression.....Where columnname = expression;

Deletion Operation:-

A delete query is expressed in much the same way as Query. We can delete whole tuple (rows) we can delete values on only particular attributes.

Deletion of all rows

Syntax:

Delete from tablename :

Deletion of specified number of rows

Syntax:

Delete from table

name Where search

condition ;

Computation in expression lists used to select data

+ Addition

* multiplication

/ Division

- Subtraction

** exponentiation

() Enclosed operation

Database Management System Lab (LC-CSE-209G)

Renaming columns used with Expression Lists: - The default output column names can be renamed by the user if required

Syntax:

```
Select column name
           t_columnname,
           Columnname
           t_columnname,
From table name;
```

Logical Operators:

The logical operators that can be used in SQL sentences are

AND	all of must be included
OR	any of may be included
NOT	none of could be included

Range Searching: Between operation is used for range searching.

Pattern Searching:

The most commonly used operation on string is pattern matching using the operation 'like' we describe patterns by using two special characters.

- Percent (%) ; the % character matches any substring we consider the following examples.
- 'Perry %' matches any string beginning with perry
- '% idge %' matches any string containing ' idge' as substring.
- ' - - - ' matches any string exactly three characters.
- ' - - - %' matches any string of at least of three characters.

Oracle functions:

Functions are used to manipulate data items and return result. Functions follow the format of function _name (argument1, argument2 ..) .An arrangement is user defined variable or constant. The structure of function is such that it accepts zero or more arguments.

Examples:

Avg return average value of n

Syntax:

```
Avg ([distinct/all]n)
Min                    return minimum value of expr.
```

Syntax:

```
MIN((distinct/all )expr)
Count                  Returns the no of rows where expr is not null
```

Database Management System Lab (LC-CSE-209G)

Syntax:

Count ([distinct/all]expr)

Count (*) Returns the no rows in the table, including duplicates and those with nulls.
Max Return max value of expr

Syntax:

Max ([distinct/all]expr)

Sum Returns sum of values of n

Syntax:

Sum ([distinct/all]n)

Sorting of data in table

Syntax:

Select
columnname,
columnnameFrom
table
Order by columnname;

OUTPUT:-

```
mysql> update employee set Job= Junior Developer* where Empno = 1232132;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| Empno | Empname | Job          | Mgr    | Sal    |
+-----+-----+-----+-----+-----+
| 1232132 | Sujal   | Junior Developer | 5242218 | 100000 |
| 1232526 | Piyush  | Junior Developer | 2772318 | 150000 |
| 1256836 | Satyam  | Junior Developer | 2777638 | 140000 |
| 2988336 | Divyansh | Team Leader    | 8783823 | 250000 |
| 3726836 | Jitesh  | Senior Developer | 3763738 | 200000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> alter table employee rename column Mgr to SSN;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| Empno | Empname | Job          | SSN    | Sal    |
+-----+-----+-----+-----+-----+
| 1232132 | Sujal   | Junior Developer | 5242218 | 100000 |
| 1232526 | Piyush  | Junior Developer | 2772318 | 150000 |
| 1256836 | Satyam  | Junior Developer | 2777638 | 140000 |
| 2988336 | Divyansh | Team Leader    | 8783823 | 250000 |
| 3726836 | Jitesh  | Senior Developer | 3763738 | 200000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> delete from employee where Empno = 1232132;
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| Empno | Empname | Job          | SSN    | Sal    |
+-----+-----+-----+-----+-----+
| 1232526 | Piyush  | Junior Developer | 2772318 | 150000 |
| 1256836 | Satyam  | Junior Developer | 2777638 | 140000 |
| 2988336 | Divyansh | Team Leader    | 8783823 | 250000 |
| 3726836 | Jitesh  | Senior Developer | 3763738 | 200000 |
+-----+-----+-----+-----+-----+
```

Database Management System Lab (LC-CSE-209G)

```
mysql> select * from emp;
+-----+-----+-----+-----+
| E_id   | E_name | Age  | Salary |
+-----+-----+-----+-----+
| 29863632 | Sujal | 21  | 250000 |
| 37363632 | Sanchit | 22 | 212000 |
| 28223632 | Piyush | 20  | 210000 |
| 22892612 | Abhay  | 19  | 120000 |
| 19282612 | Kunal  | 18  | 178000 |
| 23282612 | Paras  | 22  | 131000 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select count(E_name) from emp;
+-----+
| count(E_name) |
+-----+
| 6              |
+-----+
1 row in set (0.01 sec)

mysql> select max(Age) from emp;
+-----+
| max(Age) |
+-----+
| 22       |
+-----+
1 row in set (0.00 sec)

mysql> select min(Age) from emp;
+-----+
| min(Age) |
+-----+
| 18       |
+-----+
1 row in set (0.00 sec)

mysql> select sum(Salary) from emp;
+-----+
| sum(Salary) |
+-----+
| 1101000    |
+-----+
```

Question.1 Using the table client master and product master answer the following Questionnaires.

- i. Change the selling price of '1.44 floppy drive to Rs.1150.00
- ii. Delete the record with client 0001 from the client master table.
- iii. Change the city of client_no'0005' to Bombay.
- iv. Change the bal_due of client_no '0001, to 1000.
- v. Find the products whose selling price is more than 1500 and also find the newselling price as original selling price *15.
- vi. Find out the clients who stay in a city whose second letter is a.
- vii. Find out the name of all clients having 'a' as the second letter in their names.
- viii. List the products in sorted order of their description.
- ix. Count the total number of orders
- x. Calculate the average price of all the products.
- xi. Calculate the minimum price of products.
- xii. Determine the maximum and minimum prices . Rename the tittle as 'max_price' and min_price respectively.
- xiii. Count the number of products having price greater than or equal to 1500.

Database Management System Lab (LC-CSE-209G)

```
mysql> select avg(Salary) from emp;
+-----+
| avg(Salary) |
+-----+
| 183500.0000 |
+-----+
1 row in set (0.00 sec)

mysql> create view A as select age from emp where age>20;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from A;
+----+
| age |
+----+
| 21  |
| 22  |
| 22  |
+----+
3 rows in set (0.01 sec)

mysql> select Salary from emp group by Salary;
+-----+
| Salary |
+-----+
| 250000 |
| 212000 |
| 210000 |
| 120000 |
| 178000 |
| 131000 |
+-----+
6 rows in set (0.01 sec)

mysql>
mysql> select E_name, Salary from emp order by Salary;
+-----+-----+
| E_name | Salary |
+-----+-----+
| Abhay  | 120000 |
| Panas  | 131000 |
| Kunal  | 178000 |
+-----+-----+
```

```
mysql> select * from emp where salary>=150000 order by E_name;
+-----+-----+-----+-----+
| E_id   | E_name | Age | Salary |
+-----+-----+-----+-----+
| 29863632 | Paras | 21 | 250000 |
| 37363632 | Sanchit | 22 | 212000 |
| 28223632 | Abhay | 20 | 210000 |
| 19282612 | Kunal | 18 | 178000 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from emp where E_name LIKE '%Sat%';
+-----+-----+-----+-----+
| E_id   | E_name | Age | Salary |
+-----+-----+-----+-----+
| 28223632 | Abhay | 20 | 210000 |
| 19282612 | Kunal | 18 | 178000 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from emp where E_name LIKE '%s%';
+-----+-----+-----+-----+
| E_id   | E_name | Age | Salary |
+-----+-----+-----+-----+
| 29863632 | Paras | 21 | 250000 |
| 37363632 | Sanchit | 22 | 212000 |
| 28223632 | Abhay | 20 | 210000 |
| 19282612 | Kunal | 18 | 178000 |
| 23282612 | Sujal | 22 | 131000 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from emp where E_name LIKE 's%';
+-----+-----+-----+-----+
| E_id   | E_name | Age | Salary |
+-----+-----+-----+-----+
| 28223632 | Abhay | 20 | 210000 |
| 19282612 | Kunal | 18 | 178000 |
| 23282612 | Sujal | 22 | 131000 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Practical .4

Database Management System Lab (LC-CSE-209G)

Objective:- To Implement the structure of the table

Modifying the Structure of Tables- Alter table command is used to changing the structure of a table. Using the alter table clause you cannot perform the following tasks:

- (i) change the name of table
- (ii) add a column
- (iii) drop a column
- (iv) decrease the size of a table if table data exists.

The following tasks you can perform through alter table command.

(i) **Adding new columns:**

Syntax

ALTER TABLE tablename
ADD (newcolumnname newdatatype (size));

(ii) **Modifying existing table**

Syntax:

ALTER TABLE tablename
MODIFY (newcolumnname newdatatype (size));

NOTE: Oracle not allow constraints defined using the alter table, if the data in the table,violates such constraints.

Removing/Deleting Tables- Following command is used for removing or deleting atable.

Syntax:

DROP TABLE tablename:

Defining Integrity constraints in the ALTER TABLE command-

You can also define integrity constraints using the constraint clause in the ALTER TABLE command. The following examples show the definitions of several integrityconstraints.

Add PRIMARY KEY-

(1) Syntax:

ALTER TABLE tablename
ADD PRIMARY KEY (columnname);

(2) **Add FOREIGN KEY-**

Syntax:

Database Management System Lab (LC-CSE-209G)

ALTER TABLE tablename

ADD CONSTRAINT constraintname

FOREIGN KEY(columnname) REFERENCES tablename; Dropping integrity constraints in the ALTER TABLE command:

You can drop an integrity constraint if the rule that it enforces is no longer true or if the constraint is no longer needed. Drop the constraint using the ALTER TABLE command with the DROP clause. The following examples illustrate the dropping of integrity constraints.

(1) **DROP the PRIMARY KEY**-Syntax:

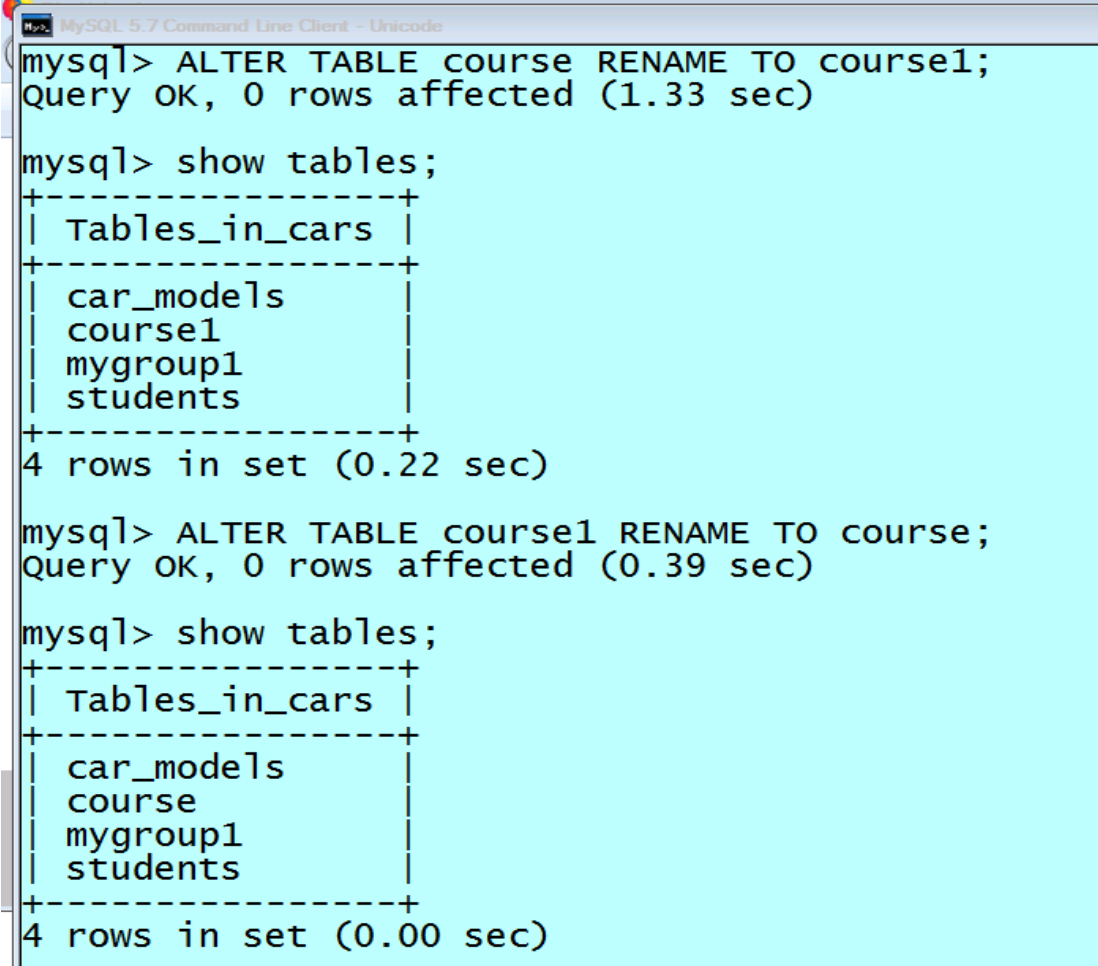
ALTER TABLE tablename DROP PRIMARY KEY

(2) **DROP FOREIGN KEY**-Syntax:

ALTER TABLE tablename

DROP CONSTRAINT constraintname;

OUTPUT:-



```
MySQL 5.7 Command Line Client - Unicode
mysql> ALTER TABLE course RENAME TO course1;
Query OK, 0 rows affected (1.33 sec)

mysql> show tables;
+-----+
| Tables_in_cars |
+-----+
| car_models     |
| course1        |
| mygroup1       |
| students       |
+-----+
4 rows in set (0.22 sec)

mysql> ALTER TABLE course1 RENAME TO course;
Query OK, 0 rows affected (0.39 sec)

mysql> show tables;
+-----+
| Tables_in_cars |
+-----+
| car_models     |
| course         |
| mygroup1       |
| students       |
+-----+
4 rows in set (0.00 sec)
```

Database Management System Lab (LC-CSE-209G)

```
mysql> alter table mahasiswa add (gender char(1));
Query OK, 2 rows affected (0.84 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim_mah   | int(11)       | NO   | PRI | NULL    |       |
| nama_mah  | varchar(25)   | YES  |     | NULL    |       |
| alamat_mah| text          | YES  |     | NULL    |       |
| id_fak    | int(11)       | NO   | MUL | NULL    |       |
| no_telp   | varchar(12)   | YES  |     | NULL    |       |
| gender    | char(1)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.10 sec)
```

Question 1. Create the following tables:

Challan Header

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key
s_order_no	varchar2	6	Foreign key references s_order_no of sales_order table
challan_date	date		not null
billed_yn	char	1	values ('Y','N'). Default 'N'

Table Name : Challan_Details

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key/Foreign key references Product_no of product_master
Qty_disp	number	4,2	not null

Q2. Insert the following values into the challan header and challan_details tables:

(i)	Challan No	S_order No	Challan Date	Billed
	CH9001	019001	12-DEC-95	Y
	CH865	046865	12-NOV-95	Y
	CH3965	010008	12-OCT-95	Y

Data for challan_details table

Challan No	Product No	Qty Disp
CH9001	P00001	4
CH9001	P07965	1
CH9001	P07885	1
CH6865	P07868	3
CH6865	P03453	4
CH6865	P00001	10
CH3965	P00001	5
CH3965	P07975	2

Answer the following Questionries

Database Management System Lab (LC-CSE-209G)

- Q1. Make the primary key to client_no in client_master.
- Q2. Add a new column phone_no in the client_master table.
- Q3. Add the not null constraint in the product_master table with the columns description, profit percent, sell price and cost price.
- Q4. Change the size of client_no field in the client_master table.
- Q5. Select product_no, description where profit percent is between 20 and 30 both inclusive.

Practical #5

Objective:- To Implement the restrictions on the table.

Data constraints: Besides the cell name, cell length and cell data type there are other parameters i.e. other data constraints that can be passed to the DBA at check creation time. The constraints can either be placed at column level or at the table level.

i. Column Level Constraints: If the constraints are defined along with the column definition, it is called a column level constraint.

ii. Table Level Constraints: If the data constraint attached to a specify cell in a table reference the contents of another cell in the table then the user will have to use table level constraints.

Null Value Concepts:- while creating tables if a row lacks a data value for particular column that value is said to be null. Column of any data types may contain null values unless the column was defined as not null when the table was created

Syntax:

Create table tablename

(columnname data type (size) not null)

Primary Key: primary key is one or more columns in a table used to uniquely identify each row in the table. Primary key values must not be null and must be unique across the column. A multicolumn primary key is called composite primary key.

Syntax: [primary key as a column constraint](#)

Create table tablename

(columnname datatype (size) primary key,....)

[Primary key as a table constraint](#)

Create table tablename

(columnname datatype (size), columnname datatype (size)...Primary key (columnname,columnname));

Default value concept: At the line of cell creation a default value can be assigned to it. When the user is loading a record with values and leaves this cell empty, the DBA will automatically load this cell with the default value specified. The data type of the default value should match the data type of the column

Syntax:

Create table tablename

(columnname datatype (size) default value,....);

Foreign Key Concept : Foreign key represents relationship between tables. A

Database Management System Lab (LC-CSE-209G)

foreign key is column whose values are derived from the primary key of the same of some other table . the existence of foreign key implies that the table with foreign key is related to the primary key table from which the foreign key is derived .A foreign key must have corresponding primary key value in the primarykey table to have meaning.

Foreign key as a column constraint

Syntax :

Create table table name
(columnname datatype (size) references another table name);

Foreign key as a table constraint:

Syntax :

Create table name
(columnname datatype
(size)...primary key
(columnname);
foreign key (columnname)references table name);

Check Integrity Constraints: Use the check constraints when you need to enforce integrity rules that can be evaluated based on a logical expression following are a few examples of appropriate check constraints.

- A check constraints name column of the client_master so that the name is entered in upper case.
- A check constraint on the client_no column of the client_master so thatno client_no value starts with 'c'

Syntax:

Create table tablename
(columnname datatype (size) CONSTRAINT
constraintname)Check (expression));

Database Management System Lab (LC-CSE-209G)

OUTPUT:-

Command Prompt - mysql -uroot -p

```
mysql> create table 052_mk
-> (
-> kode_mk char(10) not null,
-> nama_mk varchar(50) not null,
-> sks char(2) not null,
-> semester char(2) not null,
-> kode_dsn char(5) not null,
-> primary key(kode_mk),
-> constraint fk_mk_dsn
-> foreign key(kode_dsn) references 052_dosen(kode_dsn)
-> )engine = InnoDB;
Query OK, 0 rows affected (1.27 sec)
```

```
mysql> desc 052_mk;
```

Field	Type	Null	Key	Default	Extra
kode_mk	char(10)	NO	PRI	NULL	
nama_mk	varchar(50)	NO		NULL	
sks	char(2)	NO		NULL	
semester	char(2)	NO		NULL	
kode_dsn	char(5)	NO	MUL	NULL	

```
5 rows in set (0.06 sec)
```

Question.1 Create the following tables:

Columnname	Datatype	Size	Attributes
Salesman_no	varchar2	6	Primary key/first letter must start with 's'
Sal_name	varchar2	20	Not null
Address	varchar2		Not null
City	varchar2	20	
State	varchar2	20	
Pincode	Number	6	
Sal_amt	Number	8,2	Not null, cannot be 0
Tgt_to_get	Number	6,2	Not null, cannot be 0
Ytd_sales	Number	6,2	Not null, cannot be 0
Remarks	Varchar2	30	

Database Management System Lab (LC-CSE-209G)

i. Sales_order

Columnname	Datatype	Size	Attributes
S_order_no	vchar2	6	Primary/first letter must be 0
S_order_date	Date	6	Primary key reference clientno of client_master table
Client_no	Varchar2	25	
Dely_add	Varchar2	6	
Salesman_no	Varchar2	6	Foreign key references salesman_no of salesman_master table
Dely_type	Char	1	Delivery part(p)/full(f),default f
Billed_yn	Char	1	
Dely_date	Date		Can not be less than s_order_date
Order_status	Varchar2	10	Values ('in process','fulfilled','back order','canceled

I. Sales_order_details

Column	Datatype	Size	Attributes
S_order_no	Varchar2	6	Primary key/foreign key references s_order_no of sales_order
Product_no	Varchar2	6	Primary key/foreign key references product_no of product_master
Qty_order	Number	8	
Qty_disp	Number	8	
Product_rate	Number	10,2	

Insert the following data into their respective tables using insert statement:

Data for sales_man master table

Salesman_no	Salesman name	Address	City	Pin code	State	Salamt	Tgt_to_get	Ytd Sales	Remarks
500001	Kiran	A/14 worli	Bom bay	400002	Mah	3000	100	50	Good
500002	Manish	65,nariman	Bom bay	400001	Mah	3000	200	100	Good
500003	Ravi	P-7 Bandra	Bom bay	400032	Mah	3000	200	100	Good
500004	Ashish	A/5 Juhu	Bom bay	400044	Mah	3500	200	150	Good

Database Management System Lab (LC-CSE-209G)

(ii)

Data for salesorder table:

S_orderno	S_orderdate	Client no	Dely type	Bill yn	Salesman no	Delay date	Orderst atus
019001	12-jan-96	0001	F	N	50001	20-jan-96	Ip
019002	25-jan-96	0002	P	N	50002	27-jan-96	C
016865	18-feb-96	0003	F	Y	500003	20-feb-96	F
019003	03-apr-96	0001	F	Y	500001	07-apr-96	F
046866	20-may-96	0004	P	N	500002	22-may-96	C
010008	24-may-96	0005	F	N	500004	26-may-96	Ip

(i)

Data for sales_order_details table:

S_order no	Product no	Qty ordered	Qty disp	Product_rate
019001	P00001	4	4	525
019001	P07965	2	1	8400
019001	P07885	2	1	5250
019002	P00001	10	0	525
046865	P07868	3	3	3150
046865	P07885	10	10	5250
019003	P00001	4	4	1050
019003	P03453	2	2	1050
046866	P06734	1	1	12000
046866	P07965	1	0	8400
010008	P07975	1	0	1050
010008	P00001	10	5	525

Practical #6

Objective:- To implement the concept of Joins

Joint Multiple Table (Equi Join): Sometimes we require to treat more than one table although manipulate data from all the tables as though the tables were not separate object but one single entity. To achieve this we have to join tables. Tables are joined on column that have same data type and data with in tables.

The tables that have to be joined are specified in the FROM clause and the joining attributes in the WHERE clause.

Algorithm for JOIN in SQL:

1. Cartesian product of tables (specified in the FROM clause)
2. Selection of rows that match (predicate in the WHERE clause)
3. Project column specified in the SELECT clause.

1. Cartesian product:-

Consider two table student and

course
Select B.*,P.*

FROM student B, course P;

2. INNER JOIN:

Cartesian product followed by

selection
Select B.*,P.*

FROM student B, Course P WHERE

B.course # P.course # ;

3. LEFT OUTER JOIN:

LEFT OUTER JOIN = Cartesian product + selection but include rows from the left table which are unmatched with nulls in the values of attributes belonging to the second table

Exam:

Select B.*,P.*

FROM student B left join course P ON

B.course # P.course #;

4. RIGHT OUTER JOIN:

RIGHT OUTER JOIN = Cartesian product + selection but include rows from right table which are unmatched

Exam:

Select B.*,P.*

From student B RIGHT JOIN course P

B.course# = P course # ;

Database Management System Lab (LC-CSE-209G)

5. FULL OUTER JOIN

Exam

Select B.*,P.*

From student B FULL JOIN course P On

B.course # = P course # ;

OUTPUT:-

```
mysql> select * from student;
+-----+-----+-----+-----+-----+
| Rollno | Name  | Address | Age  | Phone |
+-----+-----+-----+-----+-----+
| 24256 | Sujal | ABC1    | 21  | 7293799291 |
| 24266 | Sahil | ABC2    | 21  | 9262810921 |
| 24268 | Abhay | ABC3    | 20  | 7013810921 |
| 24272 | Paras | ABC6    | 19  | 9923259261 |
| 24273 | Piyush | ABC5    | 20  | 8950259261 |
| 24287 | kunal | ABC4    | 19  | 8293230911 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from studentcourse;
+-----+-----+
| Rollno | course_id |
+-----+-----+
| 24225 | 9          |
| 24254 | 3          |
| 24256 | 2          |
| 24272 | 4          |
| 24273 | 1          |
| 24275 | 1          |
| 24287 | 5          |
+-----+-----+
7 rows in set (0.00 sec)

mysql> Select studentcourse.course_id, student.name, student.age
-> from student
-> INNER JOIN studentcourse
-> ON student.rollno=studentcourse.rollno;
+-----+-----+-----+
| course_id | name  | age  |
+-----+-----+-----+
| 2          | Sujal | 21   |
| 4          | Paras | 19   |
| 1          | Piyush | 20  |
| 5          | kunal | 19   |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> Select student.name, studentcourse.course_id
```

Answer the following Queries:

1. Find out the product which has been sold to 'Ivan Sayross.'
2. Find out the product and their quantities that will have do delivered.
3. Find the product_no and description of moving products.
4. Find out the names of clients who have purchased 'CD DRIVE'
5. List the product_no and s_order_no of customers haaving qty ordered less than 5from the order details table for the product "1.44 floppies".
6. Find the products and their quantities for the orders placed by 'Vandan Saitwal 'and "Ivan Bayross".
7. Find the products and their quantities for the orders placed by client_no "C00001" and "C00002"
8. Find the order No., Client No and salesman No. where a client has been receivedby more than one salesman.

Database Management System Lab (LC-CSE-209G)

9. Display the s_order_date in the format “dd-mm-yy” e.g. “12- feb-96”
10. Find the date , 15 days after date.

```
mysql> Select student.name, studentcourse.course_id
-> FROM student
-> LEFT JOIN studentcourse
-> ON studentcourse.rollno=student.rollno;
+-----+-----+
| name   | course_id |
+-----+-----+
| Sujal  | 2         |
| Sahil  | NULL      |
| Abhay  | NULL      |
| Paras  | 4         |
| Piyush | 1         |
| kunal  | 5         |
+-----+-----+
6 rows in set (0.01 sec)

mysql> Select student.name, studentcourse.course_id
-> FROM student
-> RIGHT JOIN studentcourse
-> ON studentcourse.rollno=student.rollno;
+-----+-----+
| name   | course_id |
+-----+-----+
| NULL   | 9         |
| NULL   | 3         |
| Sujal  | 2         |
| Paras  | 4         |
| Piyush | 1         |
| NULL   | 1         |
| kunal  | 5         |
+-----+-----+
7 rows in set (0.00 sec)

mysql> Select student.name, studentcourse.course_id
-> FROM student
-> LEFT JOIN studentcourse
-> ON studentcourse.rollno=student.rollno
-> UNION
-> Select student.name, studentcourse.course_id
-> FROM student
-> RIGHT JOIN studentcourse
-> ON studentcourse.rollno=student.rollno;
```

```
mysql> Select student.name, studentcourse.course_id
-> FROM student
-> LEFT JOIN studentcourse
-> ON studentcourse.rollno=student.rollno
-> UNION
-> Select student.name, studentcourse.course_id
-> FROM student
-> RIGHT JOIN studentcourse
-> ON studentcourse.rollno=student.rollno;
+-----+-----+
| name   | course_id |
+-----+-----+
| Sujal  | 2         |
| Sahil  | NULL      |
| Abhay  | NULL      |
| Paras  | 4         |
| Piyush | 1         |
| kunal  | 5         |
| NULL   | 9         |
| NULL   | 3         |
| NULL   | 1         |
+-----+-----+
9 rows in set (0.01 sec)
```


Practical # 7

Objective:- To implement the concept of grouping of Data.

Grouping Data From Tables:

There are circumstances where we would like to apply the aggregate function not only to a single set of tuples, but also to a group of sets of tuples, we specify this wish in SQL using the group by clause. The attribute or attributes given in the group by clause are used to form group. Tuples with the same value on all attributes in the group by clause are placed in one group.

Syntax:

```
SELECT columnname, columnname FROM
tablename
GROUP BY columnname;
```

At times it is useful to state a condition that applies to groups rather than to tuples. For example we might be interested in only those branches where the average account balance is more than 1200. This condition does not apply to a single tuple, rather it applies to each group constructed by the GROUP BY clause. To express such Query, we use the having clause of SQL. SQL applies predicates in the having clause.

Syntax:

```
SELECT columnname, columnname
FROM tablename
GROUP BY columnname; HAVING searchcondition;
```

Answer the following Queries:

Q1.- Print the description and total quantity sold for each product. Q2.- Find the value of each product sold.

Q3.- Calculate the average quantity sold for each client that has a maximum order value of 15000.

Q4.- Find out the products which has been sold to Ivan. Q5.- Find the names of clients who have 'CD Drive'.

Q6.- Find the products and their quantities for the orders placed by 'Vandana' and 'Ivan'.

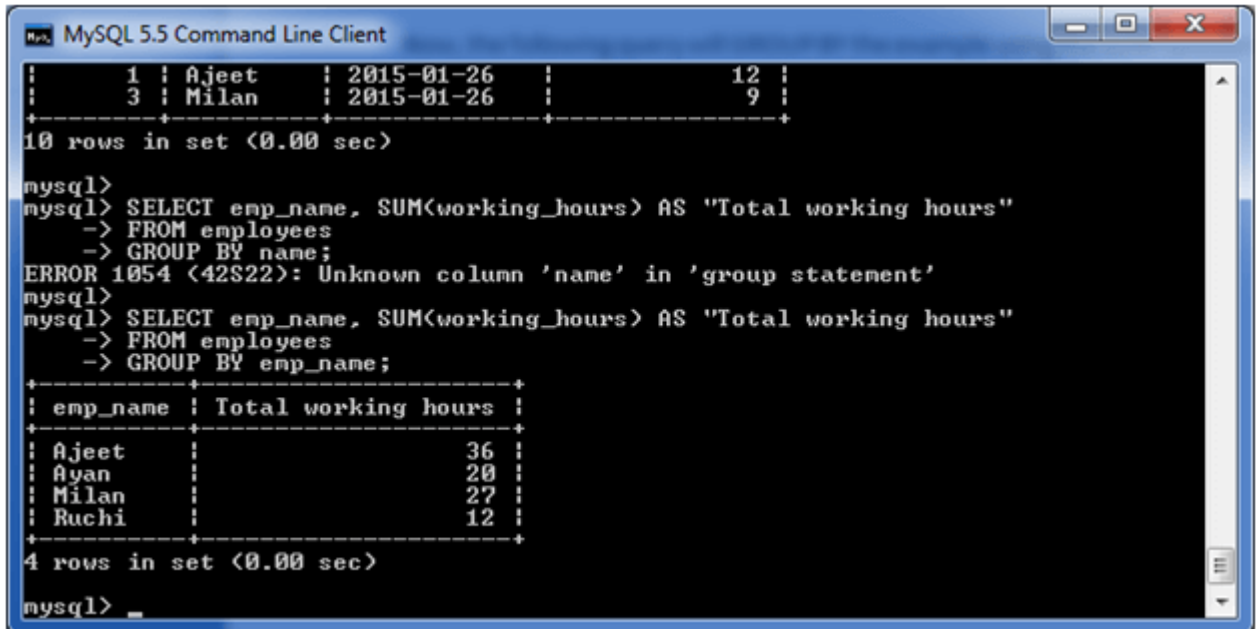
Q7.- Select product_no, total qty_ordered for each product.

Q8.- Select product_no, product description and qty ordered for each product. Q9.- Display the order number and day on which clients placed their order.

Q10.- Display the month and Date when the order must be delivered.

Database Management System Lab (LC-CSE-209G)

OUTPUT:-



```
mysql>
+----+-----+-----+-----+
| 1 | Ajeet | 2015-01-26 | 12 |
| 3 | Milan | 2015-01-26 | 9  |
+----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
mysql> SELECT emp_name, SUM(working_hours) AS "Total working hours"
-> FROM employees
-> GROUP BY name;
ERROR 1054 (42S22): Unknown column 'name' in 'group statement'
mysql>
mysql> SELECT emp_name, SUM(working_hours) AS "Total working hours"
-> FROM employees
-> GROUP BY emp_name;
+----+-----+-----+-----+
| emp_name | Total working hours |
+----+-----+-----+-----+
| Ajeet    | 36                   |
| Ayan     | 20                   |
| Milan    | 27                   |
| Ruchi    | 12                   |
+----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

Practical #8

Objective:- To implement the concept of SubQueries.

SubQueries:- A subQuery is a form of an SQL statement that appears inside another SQL statement. It also termed as nested Query. The statement containing a subQuery called a parent statement. The rows returned by the subQuery are used by the following statement.

It can be used by the following commands:

1. To insert records in the target table.
2. To create tables and insert records in this table.
3. To update records in the target table.
4. To create view.
5. To provide values for the condition in the WHERE , HAVING IN ,SELECT,UPDATE, and DELETE statements.

Exam:-

Creating clientmaster table from oldclient_master, table

```
Create table client_master  
AS SELECT * FROM oldclient_master;
```

Using the Union, Intersect and Minus Clause:

Union Clause:

The user can put together multiple Queries and combine their output using the union clause . The union clause merges the output of two or more Queries into a single set of rows and column. The final output of union clause will be

Output: = Records only in Query one + records only in Query two + A single set of records with is common in the both Queries.

Syntax:

```
SELECT  
columnname,  
columnnameFROM  
tablename 1  
UNION  
SELECT columnname,  
columnnameFrom  
tablename2;
```

Database Management System Lab (LC-CSE-209G)

Intersect Clause: The user can put together multiple Queries and their output using the intersect clause. The final output of the intersect clause will be :

Output = A single set of records which are common

in both Queries Syntax:

```
SELECT columnname, columnname FROM tablename 1
INTERSECT
SELECT columnname, columnname FROM tablename
2;
```

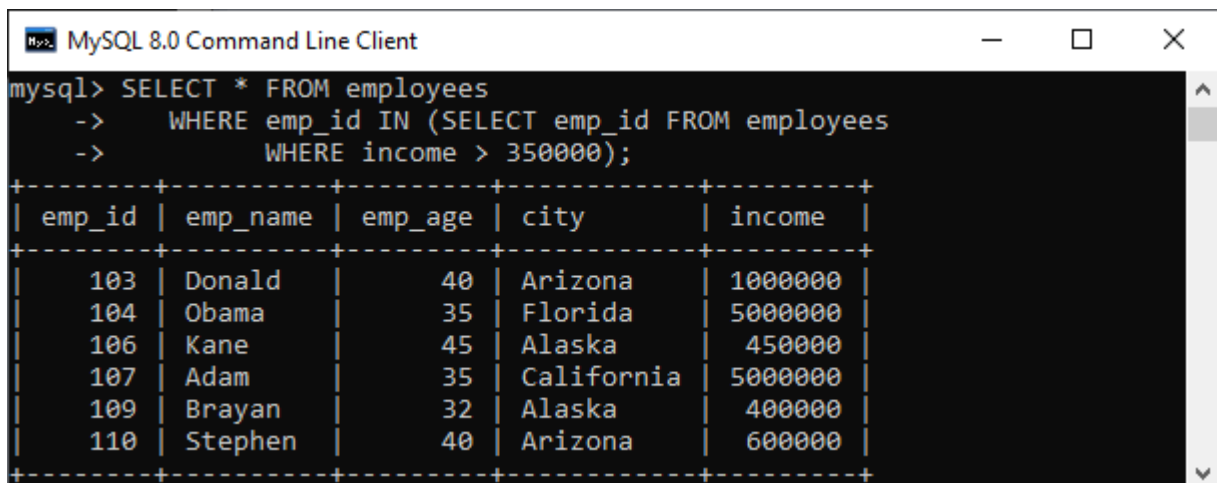
MINUS CLAUSE:- The user can put together multiple Queries and combine their output = records only in

Query one

Syntax:

```
SELECT columnname, columnname FROM tablename ;
MINUS
SELECT columnname, columnname FROM tablename ;
```

OUTPUT:-



The screenshot shows a MySQL 8.0 Command Line Client window. The user has entered the following query:

```
mysql> SELECT * FROM employees
-> WHERE emp_id IN (SELECT emp_id FROM employees
-> WHERE income > 350000);
```

The output is a table with 5 columns: emp_id, emp_name, emp_age, city, and income. The data is as follows:

emp_id	emp_name	emp_age	city	income
103	Donald	40	Arizona	1000000
104	Obama	35	Florida	5000000
106	Kane	45	Alaska	450000
107	Adam	35	California	5000000
109	Brayan	32	Alaska	400000
110	Stephen	40	Arizona	600000

Database Management System Lab (LC-CSE-209G)

Answer the following Queries:

Question.

1. Find the product_no and description of non- moving products.
2. Find the customer name, address, city and pincode for the client who has placed order no "019001"
3. Find the client names who have placed order before the month of may 96.
4. Find out if product "1.44 Drive" is ordered by only client and print the client_name to whom it was sold.
5. find the names of client who have placed orders worth Rs.10000 or more.
6. Select the orders placed by 'Rahul Desai'
7. Select the names of persons who are in Mr. Pradeep's department and who have also worked on an inventory control system.
8. Select all the clients and the salesman in the city of Bombay.
9. Select salesman name in "Bombay" who has atleast one client located at "Bombay"
10. Select the product_no, description, qty_on-hand, cost_price of non_moving items in the product_master table.

Practical # 9

Objective:- To implement the concept of Indexes and views.

Indexes- An index is an ordered list of content of a column or group of columns in a table. An index created on the single column of the table is called simple index. When multiple table columns are included in the index it is called composite index.

Creating an Index for a table:-

Syntax (Simple)

```
CREATE
INDEX
index_name ON
tablename(column
name);
```

Composite Index:-

```
CREATE INDEX index_name
ON tablename(columnname,columnname);
```

Creating an Unique Index:-

```
CREATE UNIQUE INDEX
indexfilename ON
tablename(columnname);
```

Dropping Indexes:-

An index can be dropped by using DROP INDEX

SYNTAX:-

```
DROP INDEX indexfilename;
```

Views:-

Logical data is how we want to see the current data in our database. Physical data is how this data is actually placed in our database.

Views are masks placed upon tables. This allows the programmer to develop a method via which we can display predetermined data to users according to our desire.

Views may be created for the following reasons:

1. The DBA stores the views as a definition only. Hence there is no duplication of data.
2. Simplifies Queries.
3. Can be Queried as a base table itself.
4. Provides data security.
5. Avoids data redundancy.

Database Management System Lab (LC-CSE-209G)

Creation of Views:-

Syntax:-

```
CREATE VIEW viewname AS SELECT  
columnname,columnnameFROM tablename  
WHERE columnname=expression_list;
```

Renaming the columns of a view:-

Syntax:-

```
CREATE VIEW viewname AS  
SELECT newcolumnname....  
FROM tablename  
WHERE columnname=expression_list;
```

Selecting a data set from a view-

Syntax:-

```
SELECT columnname, columnname  
FROM viewname  
WHERE search condition;
```

Destroying a view-

Syntax:-

```
DROP VIEW viewname;
```

OUTPUT:-

Database Management System Lab (LC-CSE-209G)

```
c:\wamp\bin\mysql\mysql5.5.16\bin\mysql.exe
mysql> create INDEX index_1 On product (product_name);
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe product;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| product_id    | int(2)        | YES  |     | NULL    |      |
| product_name  | varchar(20)   | YES  | MUL | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.03 sec)

mysql>
```

```
mysql> select * from studentdetails;
+-----+-----+-----+
| S_id | Name   | Address |
+-----+-----+-----+
| 24256 | Sujal | ABC1    |
| 24266 | kunal | ABC5    |
| 24268 | paras | ABC4    |
| 24272 | sanchit | ABC2   |
| 24273 | abhay | ABC3    |
| 24287 | piyush | ABC6    |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from studentmarks;
+-----+-----+-----+-----+
| Id   | Name   | Marks | Age |
+-----+-----+-----+-----+
| 24256 | Sujal | 82    | 20 |
| 24266 | kunal | 89    | 21 |
| 24268 | paras | 91    | 20 |
| 24272 | sanchit | 88   | 19 |
| 24273 | abhay | 92    | 18 |
| 24287 | piyush | 90    | 19 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> CREATE VIEW detailview AS
-> SELECT Name, Address
-> FROM studentdetails
-> WHERE S_id>24000;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from detailview;
+-----+-----+
| Name   | Address |
+-----+-----+
| Sujal  | ABC1    |
| kunal  | ABC5    |
| paras  | ABC4    |
| sanchit | ABC2   |
| abhay  | ABC3    |
| piyush | ABC6    |
+-----+-----+
```

Answer the following Questions

Q1. Create an index on the table client_master, field client_no.

Database Management System Lab (LC-CSE-209G)

- Q2. Create an index on the sales_order, field s_order_no.
- Q3. Create an composite index on the sales_order_details table for the columns s_order_no and product_no.
- Q4. Create an composite index ch_index on challan_header table for the columns challan no and s_order_no.
- Q5. Create an uniQuestion index on the table salesman_master, field salesman_no.
- Q6. Drop index ch_index on table challan_header.
- Q7. Create view on salesman_master whose sal_amt is less than 3500.
- Q8. Create a view client_view on client_master and rename the columns as name, add1, add2, city, pcode, state respectively.
- Q9. Select the client names from client_view who lives in city 'Bombay'.
- Q10. Drop the view client_view.

Practical # 10

Objective:- To implement the basics of PL/SQL.

Introduction – PL/SQL bridges the gap between database technology and procedural programming languages. It can be thought of as a development tool that extends the facilities of Oracles SQL database language. Via PL/SQL you can insert, delete, update and retrieve table data as well as writing loops or branching to another block of code.

PL/SQL Block structure-

DECLARE

Declarations of memory variables used later BEGIN

SQL executable statements for manipulating table data. EXCEPTIONS

SQL and/or PL.SQL code to handle errors.END;

Displaying user Messages on the screen – Any programming tool requires a method through which messages can be displayed to the user.

dbms_output is a package that includes a number of procedure and functions that accumulate information in a buffer so that it can be retrieved later. These functions canalso be used to display message to the user.

put_line: put a piece of information in the buffer followed by a end of line marker. It canalso be used to display message to the user.

Setting the server output on:

SET SERVER OUTPUT ON:

Example: Write the following code in the PL/SQL block to display message to

Database Management System Lab (LC-CSE-209G)

```
userDBMS_OUTPUT.PUT_LINE('Display user message');
```

Conditional control in PL/SQL-

Syntax:

```
IF <condition> THEN  
  <Action> ELSEIF<condition>  
  <Action>  
ELSE  
  <Action>ENDIF;
```

The WHILE LOOP:

Syntax:

```
WHILE <condition>  
LOOP  
  <Action>  
ENDLOOP;
```

The FOR LOOP statement:

Syntax:

```
FOR variable IN [REVERSE] start—end  
LOOP  
  <Action>  
ENDLOOP;
```

The GOTO statement: The goto statement allows you to change the flow of control within a PL/SQL Block.

Database Management System Lab (LC-CSE-209G)

1 a) Write a PL/SQL block to find the maximum number from given three numbers.

```
declare
a number;
b number;
c number;
begin
a:=&a;
b:=&b;
c:=&c;
if (a>b and a>c) then
    dbms_output.put_line('a is maximum ' || a);
elsif (b>a and b>c) then
    dbms_output.put_line('b is maximum ' || b);
else
    dbms_output.put_line('c is maximum ' || c);
end if;
end;
```

OUTPUT:-

19 21 8
b is maximum 21

Q1. WAP in PL/SQL for addition of two numbers.

Q2. WAP in PL/SQL for addition of 1 to 100 numbers.

Q3. WAP in PL/SQL to check the given number is even or odd.

Q4. WAP in PL/SQL to inverse a number, eg. Number 5639 when inverted must be display output 9365.

Q5. WAP in PL/SQL for changing the price of product 'P00001' to 4000 if the price is less than 4000 in product_master table. The change is recorded in the old_price_table along with product_no and the date on which the price was changed last.

MINI PROJECT(Sample)

ROADWAY TRAVELS

“Roadway Travels” is in business since 1977 with several buses connecting different places in India. Its main office is located in Hyderabad.

The company wants to computerize its operations in the following areas:

Reservations

Ticketing

Cancellations

Reservations :

Reservations are directly handled by booking office. Reservations can be made 60 days in advance in either cash or credit. In case the ticket is not available, a wait listed ticket is issued to the customer. This ticket is confirmed against the cancellation.

Cancellation and modification:

Cancellations are also directly handled at the booking office. Cancellation charges will be charged.

Wait listed tickets that do not get confirmed are fully refunded.

Database Management System Lab (LC-CSE-209G)

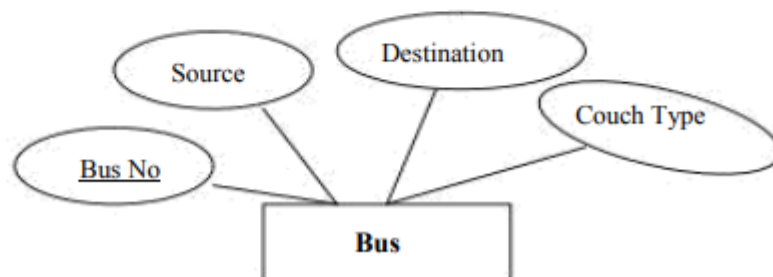
AIM: Analyze the problem and come with the entities in it. Identify what Data has to be persisted in the databases.

The Following are the entities:

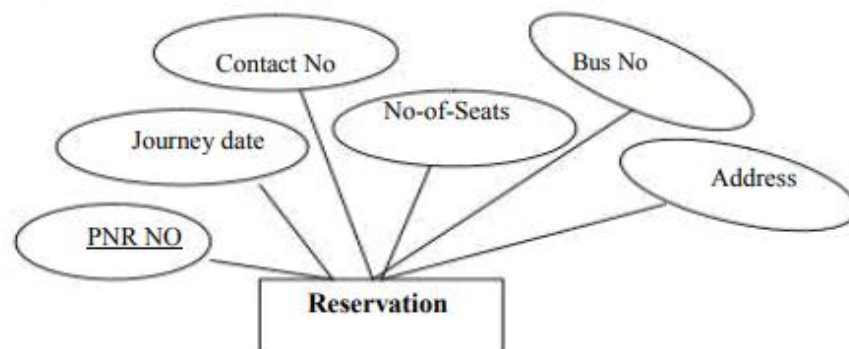
1. Bus
2. Reservation
3. Ticket
4. Passenger
5. Cancellation

The attributes in the Entities:

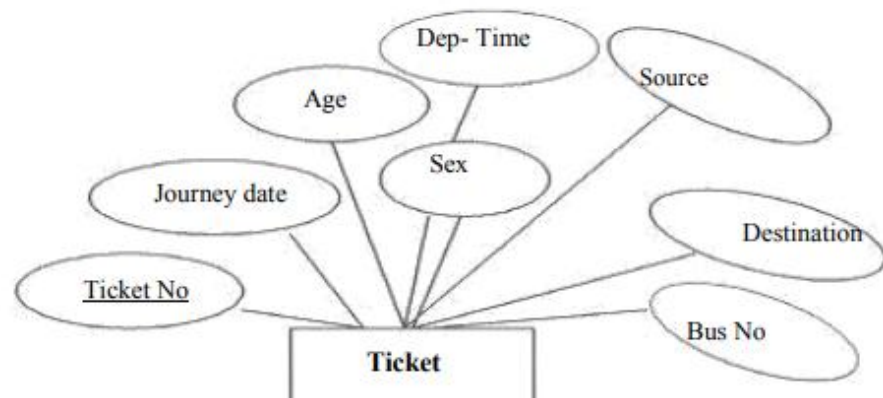
Bus:(Entity)



Reservation (Entity)

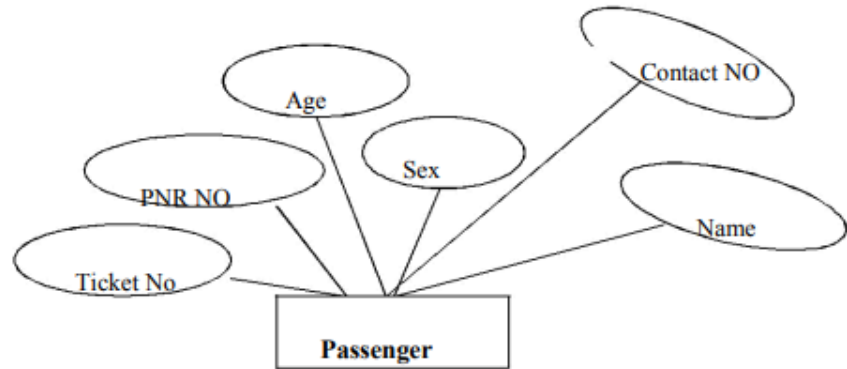


Ticket :(Entity)

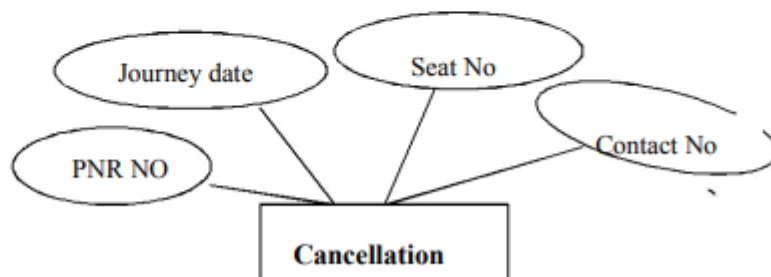


Database Management System Lab (LC-CSE-209G)

Passenger:

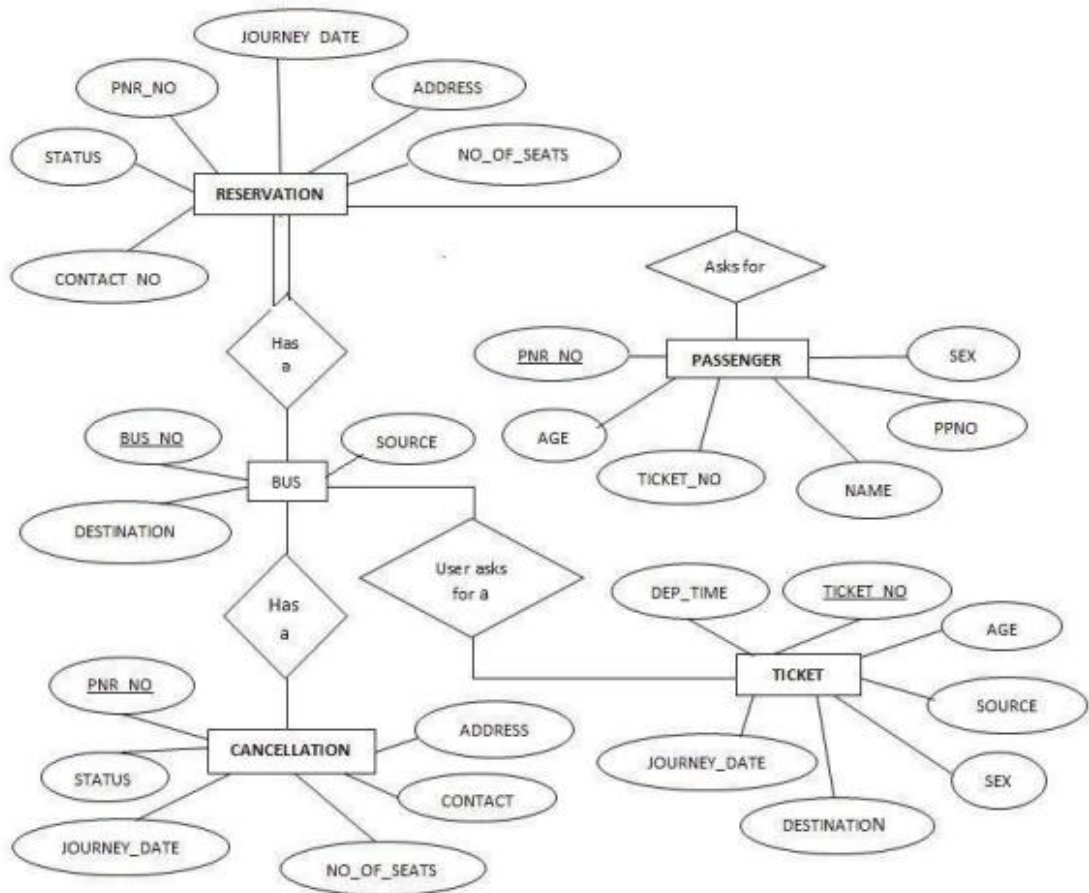


Cancellation (Entity)



Database Management System Lab (LC-CSE-209G)

Concept design with E-R Model:



The following are tabular representation of the above entities and relationships

BUS:

<u>COLOUMN NAME</u>	<u>DATA TYPE</u>	<u>CONSTRAINT</u>
Bus No	varchar2(10)	Primary Key
Source	varchar2(20)	
Destination	varchar2(20)	
Couch Type	varchar2(20)	

Database Management System Lab (LC-CSE-209G)

Reservation:

<u>COLOUMN NAME</u>	<u>DATA TYPE</u>	<u>CONSTRAINT</u>
PNRNo	number(9)	Primary Key
Journey date	Date	
No-of-seats	integer(8)	
Address	varchar2(50)	
Contact No	Number(9)	Should be equal to 10 numbers and not allow other than numeric
BusNo	varchar2(10)	Foreign key
Seat no	Number	

Database Management System Lab (LC-CSE-209G)

Ticket:

<u>COLOUMN NAME</u>	<u>DATA TYPE</u>	<u>CONSTRAINT</u>
Ticket_No	number(9)	Primary Key
Journey date	Date	
Age	int(4)	
Sex	Char(10)	
Source	varchar2(10)	
Destination	varchar2(10)	
Dep-time	varchar2(10)	
Bus No	Number2(10)	

Passenger:

<u>COLOUMN NAME</u>	<u>DATA TYPE</u>	<u>CONSTRAINT</u>
PNR No	Number(9)	Primary Key
Ticket No	Number(9)	Foreign key
Name	varchar2(15)	
Age	integer(4)	
Sex	char(10)	(Male/Female)
Contact no	Number(9)	Should be equal to 10 numbers and not allow other than numeric

Cancellation:

<u>COLOUMN NAME</u>	<u>DATA TYPE</u>	<u>CONSTRAINT</u>
PNR No	Number(9)	Foriegn-key
Journey-date	Date	
Seat no	Integer(9)	
Contact_No	Number(9)	Should be equal to 10 numbers and not allow other than numeric

Database Management System Lab (LC-CSE-209G)

Creating of Tables on ROAD WAY TRAVELS:

Table is a primary object of database, used to store data in form of rows and columns. It is created using following command:

```
Create Table <table_name> (column1 datatype(size), column2 datatype(size),column(n) datatype(size));
```

Example:

```
SQL> create table Bus(Bus_No varchar(5), source varchar(20), destination varchar(20),CouchType varchar2(10),fair number);
```

Table Created.

create table for the object-relation feature we will discuss it afterwards.

Desc command

Describe command is external command of Oracle. The describe command is used to view the structure of a table as follows.

Desc <table name>

```
SQL> desc bus;
```

Name	Null?	Type
BUS_NO	NOT NULL	INTEGER2(5)
SOURCE		VARCHAR2(20)
DESTINATION		VARCHAR2(20)
COUCH TYPE		VARCHAR2(10)
FAIR	NUMBER	

Reservation Table:

```
SQL> create table Reservation(PNR_NO Numeric(9), No_of_seats Number(8), Address varchar(50), Contact_No Numeric(9), Status char(3));
```

Table created.

```
SQL> desc Reservation
```

Name	Null?	Type
PNR_NO		NUMBER(9)
NO_OF_SEATS		NUMBER(8)
ADDRESS		VARCHAR2(50)
CONTACT_NO		NUMBER(9)
STATUS		CHAR(3)

Database Management System Lab (LC-CSE-209G)

Cancellation Table:

```
SQL> create table Cancellation(PNR_NO Numeric(9), No_of_seats Number(8), Address  
varchar(50), Contact_No Numeric(9), Status char(3));
```

Table created.

```
SQL> desc Cancellation
```

Name	Null?	Type
PNR_NO		NUMBER(9)
NO_OF_SEATS		NUMBER(8)
ADDRESS		VARCHAR2(50)
CONTACT_NO		NUMBER(9)
STATUS		CHAR(3)

Ticket Table:

```
SQL> create table Ticket(Ticket_No Numeric(9) primary key, age number(4), sex char(4)  
Not null, source varchar(2), destination varchar(20), dep_time varchar(4));
```

Table created.

```
SQL> desc Ticket
```

Name	Null?	Type
TICKET_NO	NOT NULL	NUMBER(9)
AGE		NUMBER(4)
SEX	NOT NULL	CHAR(4)
SOURCE		VARCHAR2(2)
DESTINATION		VARCHAR2(20)
DEP_TIME		VARCHAR2(4)

AIM: Applying Constraints on Road Way Travels Tables.

Constraints

Domain Integrity constraints

Entity Integrity constraints

Referential Integrity constraint

Oracle allows programmers to define constraints

Column Level

Table Level

Example

```
SQL> create table Ticket ( Ticket_No Numeric(9) , age number(4), sex char(4) Not null,  
source varchar(2), destination varchar(20), dep_time varchar(4));
```

Table created.

Database Management System Lab (LC-CSE-209G)

Check Constraint

```
SQL> create table Reservation(PNR_NO Numeric(9), No_of_seats Number(8), Address
varchar(50), Contact_No Numeric(10) constraint ck check(length(contact_no)=10), Status
char(3));
```

Table created.

Check constraint with alter command

```
SQL> alter table Ticket add constraint check_age check(age>18);
```

Table altered.

Unique Constraint

Example:

```
SQL> create table Ticket(Ticket_No Numeric(9) unique, age number(4), sex char(4) 1, source
varchar(2), destination varchar(20), dep_time varchar(4));
```

Referential Integrity Constraint

References constraint defined at column level

Example:

```
SQL> create table Passenger(PNR_NO Numeric(9) references reservation , Ticket_NO
Numeric(9) references ticket, Name varchar(20), Age Number(4), Sex char(10), PPNO
varchar(15));
```

Table created.

Database Management System Lab (LC-CSE-209G)

a) Insert command

Insert into <table name> values(a list of data values);

Insert into <table name>(column list) values(a list of data);

```
SQL> insert into emp_master (empno,ename,salary) values (1122,'Smith',8000); 1
```

row created.

Adding values in a table using Variable method.

```
SQL> insert into Passenger values(&PNR_NO,&TICKET_NO, '&Name', &Age, '&Sex', '&PPNO');
```

Enter value for pnr_no: 1

Enter value for ticket_no: 1

Enter value for name: SACHIN

Enter value for age: 12

Enter value for sex: m

Enter value for ppno: sd1234

```
old 1: insert into Passenger values(&PNR_NO,&TICKET_NO, '&Name', &Age, '&Sex', '&PPNO')
```

```
new 1: insert into Passenger values(1,1,'SACHIN',12,'m','sd1234')
```

1 row created.

Note:- Execute at least 20 queries

VIVA QUESTIONS

1. What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

2. What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

3. What is a Database system?

The database and DBMS software together is called as Database system.

4. Advantages of DBMS?

- Redundancy is controlled.
- Unauthorized access is restricted.
- Providing multiple user interfaces.
- Enforcing integrity constraints.
- Providing backup and recovery.

5. Disadvantage in File Processing System?

- Data redundancy & inconsistency.
- Difficult in accessing data.
- Data isolation.
- Data integrity.
- Concurrent access is not possible.
- Security Problems.

6. Describe the three levels of data abstraction?

Three levels of abstraction:

Physical level: The lowest level of abstraction describes how data are stored.

Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.

View level: The highest level of abstraction describes only part of entire database.

7. Define the "integrity rules"

There are two Integrity rules.

Entity Integrity: States that Primary key cannot have NULL value

Referential Integrity: States that Foreign Key can be either a NULL value or should be Primary Key value of other relation.

8. What is extension and intension?

Extension: It is the number of tuples present in a table at any instance. This is time dependent.

Intension: It is a constant value that gives the name, structure of table and the constraints laid on it.

9. What is Data Independence?

Data independence means that "The application is independent of the storage structure and access strategy of data". In other words, the ability to modify the schema definition in one level should not affect the schema definition in the next higher level.

Two types of Data Independence:

Physical Data Independence: Modification in physical level should not affect the logical level.

Logical Data Independence: Modification in logical level should affect the view level.

Database Management System Lab (LC-CSE-209G)

10. What is a view? How it is related to data independence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that directly represents the view instead a definition of view is stored in data dictionary. Growth and restructuring of base tables is not reflected in views. Thus the

View can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

11. What is Data Model?

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

12. What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

13. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

14. What is an Entity?

It is a 'thing' in the real world with an independent existence.

15. What is an Entity type?

It is a collection (set) of entities that have same attributes.

16. What is an Entity set?

It is a collection of all entities of particular entity type in the database.

17. What is an Extension of entity type?

The collections of entities of a particular entity type are grouped together into an entity set.

18. What is Weak Entity set?

An entity set may not have sufficient attributes to form a primary key, and its primary key comprises of its partial key and primary key of its parent entity, then it is said to be Weak Entity set.

19. What is an attribute?

It is a particular property, which describes the entity.

20. What is a Relation?

A relation is defined as a set of tuples.

21. What is degree of a Relation?

It is the number of attribute of its relation schema.

22. What is Relationship?

It is an association among two or more entities.

23. What is Relationship set?

The collection (or set) of similar relationships.

24. What is Relationship type?

Relationship type defines a set of associations or a relationship set among a given set of entity types.

25. What is degree of Relationship type?

It is number of entity type participating

Database Management System Lab (LC-CSE-209G)

1. What is DDL (Data Definition Language)?

A data base schema is specifies by a set of definitions expressed by a special language called DDL.

2. What is VDL (View Definition Language)?

It specifies user views and their mappings to the conceptual schema.

3. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organized by appropriate data model.

4. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

5. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

6. What is DDL Interpreter?

It interprets DDL statements and records them in tables containing metadata.

7. 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.

8. What do you mean by Correlated sub query?

A correlated sub query can be easily identified if it contains any references to the parent sub query columns in its WHERE clause. Columns from the sub query cannot be referenced anywhere else in the parent query.

9. Are the resulting relations of PRODUCT and JOIN operation the same?

No.

PRODUCT: Concatenation of every row in one relation with every row in another.

JOIN: Concatenation of rows from one relation and related rows from another.

10. What is database Trigger?

A database trigger is a PL/SQL block that can defined to automatically execute for insert, update, and delete statements against a table. The trigger can be defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. For any one table, there are twelve events for which you can define database triggers. A database trigger can call database procedures that are also written in PL/SQL.

11. What are stored-procedures? What are the advantages of using them?

Stored procedures are database objects that perform a user defined operation. A stored procedure can have a set of compound SQL statements. A stored procedure executes the SQL commands and returns the result to the client. Stored procedures are used to reduce network traffic.

12. Define super key and give example to illustrate the super key?

Set of one or more attributes taken collectively, allowing to identify uniquely an entity in the entity set. Eg1. {SSN} and {SSN, Cust_name} of customer table are super keys. Eg2. {Branch name} and {Branch name, Branch city} of Branch table re super keys.

13. Define candidate key and give example to illustrate the candidate key?

Database Management System Lab (LC-CSE-209G)

Super keys with no proper subset are called the candidate keys. Otherwise it is called minimal super key. Candidate key is nothing but the primary key used in SQL. Eg1. {SSN} is the candidate key for the super keys {SSN} and {SSN, Cust_name} of customer table. Eg2. {Branch name} is the candidate key for the super keys {Branch name} and {Branch name, Branch city} of Branch table.

14. What is Primary key?

A key chosen to act as the means by which to identify tuples in a relation.

15. What is foreign key?

A foreign key of relation R is a set of its attributes intended to be used (by each tuple in R) for identifying/referring to a tuples in some relation S. (R is called the referencing relation and S the referenced relation.) For this to make sense, the set of attributes of R forming the foreign key should "correspond to" some superkey of S. Indeed, by definition we require this superkey to be the primary key of S.

14. What is a Cursor?

A cursor is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the active set.

Database Management System Lab (LC-CSE-209G)

1. What is normalization?

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- Minimizing redundancy
- Minimizing insertion, deletion and update anomalies.

2. What is Functional Dependency?

A Functional dependency is denoted by $X \rightarrow Y$ between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation state r of R . The constraint is for any two tuples t_1 and t_2 in r if $t_1[X] = t_2[X]$ then they have $t_1[Y] = t_2[Y]$.

3. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

4. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency $X \rightarrow Y$ is fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

5. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

6. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD $X \rightarrow A$ either of the following is true

X is a Super-key of R .

A is a prime attribute of R .

In other words, if every non prime attribute is non-transitively dependent on primary key.

7. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD $X \rightarrow A$, X must be a candidate key.

8. What is 4NF?

A relation schema R is said to be in 4NF if for every multivalued dependency $X \twoheadrightarrow Y$ that holds over R , one of following is true X is subset or equal to (or) $XY = R$. X is a super key.

Database Management System Lab (LC-CSE-209G)

9. What is 5NF?

A Relation schema R is said to be 5NF if for every join dependency $\{R_1, R_2 \dots R_n\}$ that holds R, one the following is true

i) $R_i = R$ for some i.

ii) The join dependency is implied by the set of FD, over R in which the left side is key of R.

10. What is dependency preservation?

Dependency Preservation Property enables us to enforce a constraint on the original relation from corresponding instances in the smaller relations.

11. What is Lossless join property?

Lossless join property enables us to find any instance of the original relation from corresponding instances in the smaller relations

12. What are Multivalued dependencies?

A multivalued dependency (MVD) $X \twoheadrightarrow Y$ specified on R, where X, and Y are both subsets of R and $Z = R - (X \cup Y)$ specifies the following restrictions on $r(R)$

$t_3[X] = t_4[X] = t_1[X] = t_2[X]$

$t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$

$t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$