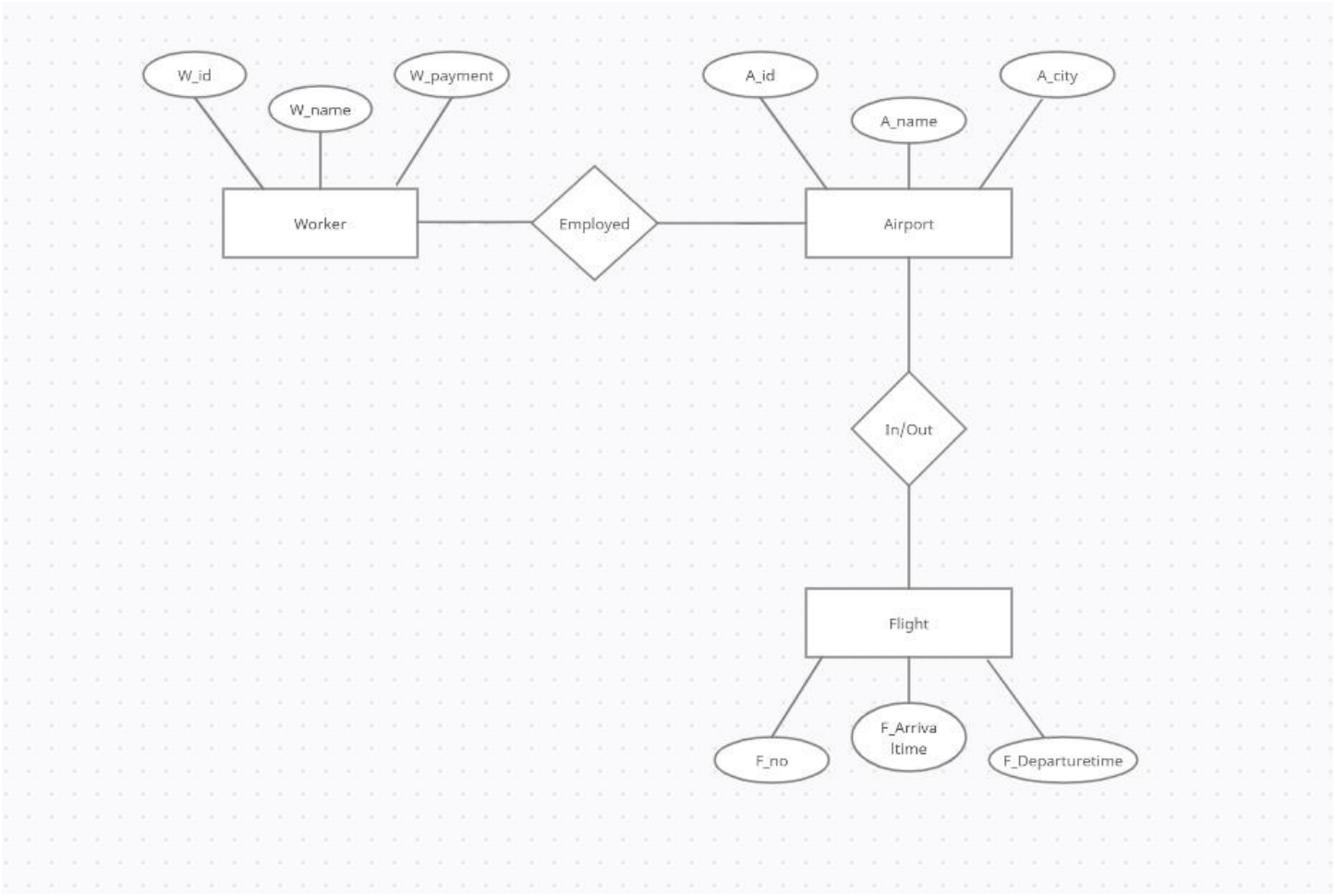
Department of Computer Engineering Class: S.E. (Computer) (Div- A) (Sem-IV)	
Subject: Database Management System Lab (CSL402)	
Sr. No.	**Title of Experiments**
1	Identify the case study and write statement of problem in detail. Design an Entity-Relationship (ER)- Extended Entity-Relationship (EER) Model for a case study.
2	Mapping ER-EER diagram in experiment-1 to Relational schema model.

Experiment No-1

Topic: Airport Management System Design an Entity-Relationship (ER)

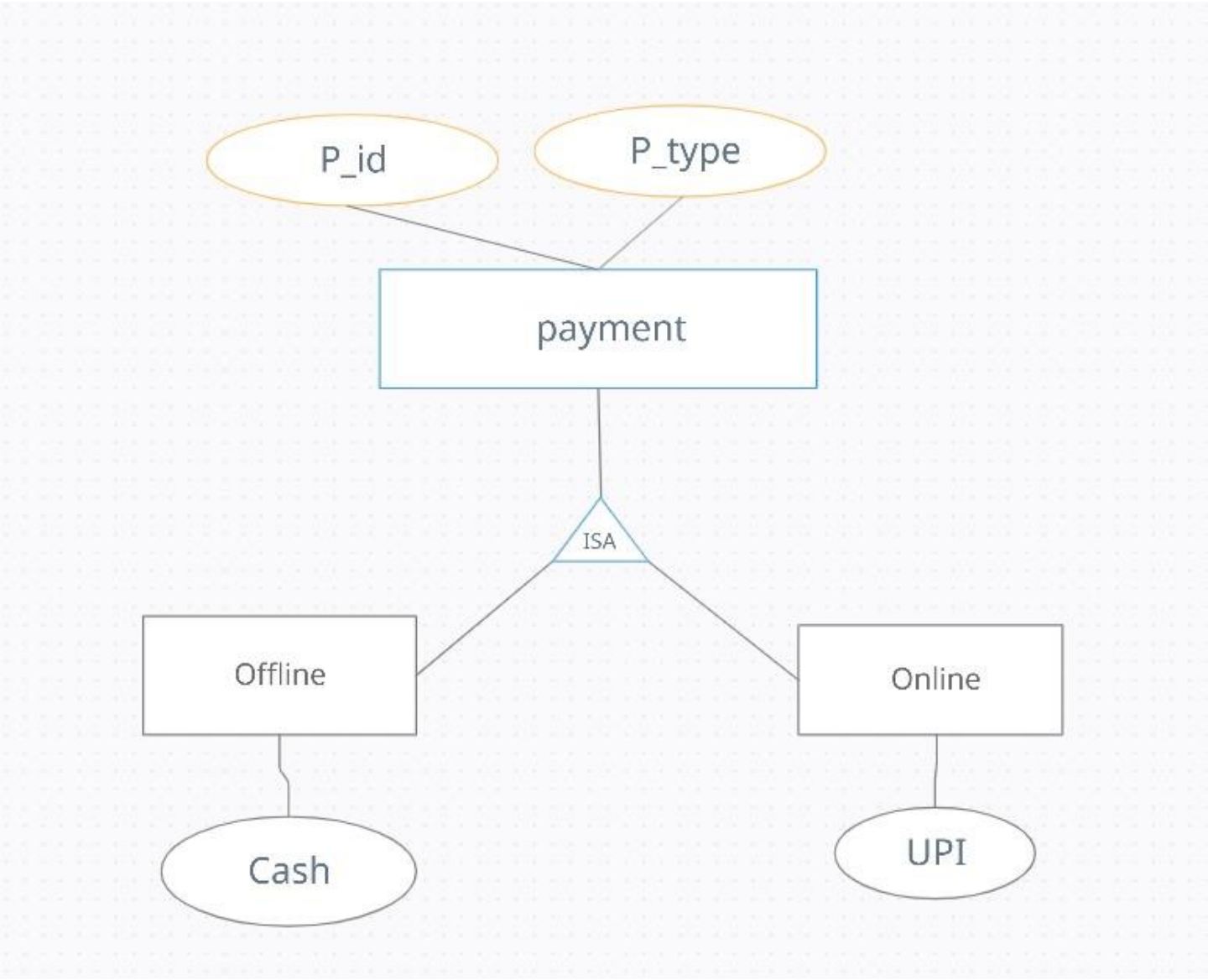
Problem Statement: Airport Management system deals with the management of the airport, airlines, passengers and employees working for an airport.

.



3. Experiment Writeup:

- In above ER-Diagram there are three entities i.e., Worker, Airport and Flight.
- Every entity has some attributes which shows the property of entities.
- Customer entity have attributes as W_id which primary key, W_name, and W_payment.
- Similarly, food entity have F_id, F_Arrivaltime, F_Departuretime and description attributes.

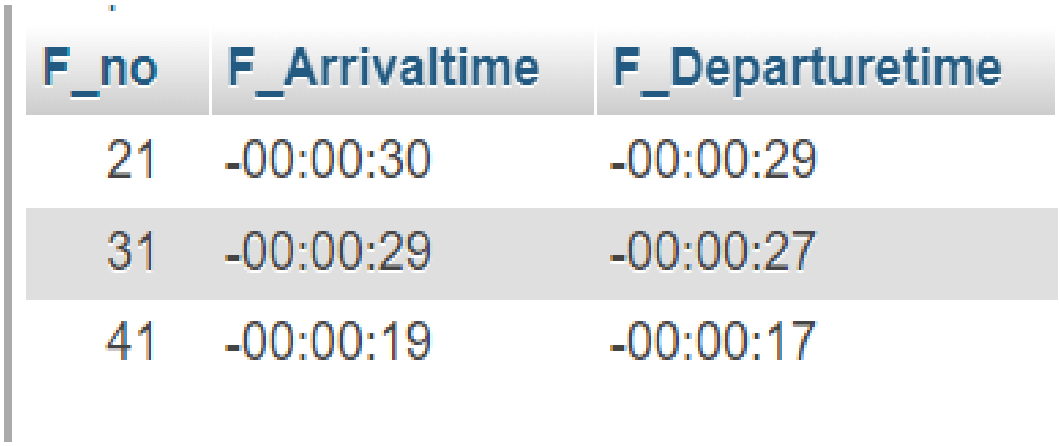

P_id
P_type
payment
ISA
Offline
Online
Cash
UPI

Experiment No-2

Topic: Mapping ER-EER diagram in Airport Management System to Relational schema model.

Problem Statement: Airport Management system deals with the management of the airport, airlines, passengers and employees working for an airport. .

	A_id	A_name	A_city
Edit Copy Delete	1	Manchester Airport	Manchester
Edit Copy Delete	2	Chatrapati Shivaji Maharaj International Airport	Mumbai
Edit Copy Delete	3	Frankfurt International Airport	Frankfurt



F_no	F_Arrivaltime	F_Departuretime
21	-00:00:30	-00:00:29
31	-00:00:29	-00:00:27
41	-00:00:19	-00:00:17

W_id	W_name	W_payment
12	Ramesh	10000
2	SUResh	100000
2	Sahil	800000

3. Experiment Writeup:

- For creating a table
  syntax: create table <table_name>;
- For Inserting data into table
  syntax: insert into <table_name> (columns name)
  values(data);
- For display table
  syntax: select * from <table_name>;

Department of Computer Engineering

Class: S.E. (Computer) (Div- A) (Sem-IV) Academic Year: 2021-22 (EVEN)

Subject: Database Management System Lab

Course Code: CSL402

Experiment No.: 3

Experiment Title: Create a database using Data definition Language (DDL) and apply integrity constraints for the specified system.

Name of Student: Atharva Vichare

Student Roll No.: SEA170

CASE STUDY:- Airport Management System

* Experiment no - 3

Title-
Create a database using Data definition language, apply integrity constraints for the specified system.

Theory
DDL commands
DDL means data definition language. It is used to create and modify the structure of database objects - SQL

1] Create database -
This statement is used to create a new database in MYSQL.
Syntax -
Create Database DatabaseName;

2] Drop Database -
It is used to delete a database from MySQL
Syntax -
Drop Database DatabaseName;

3] Use Database.
It is used to make database as current database

3] Use Database :
Syntax-
Use DatabaseName:

4] Create Table
It is used to create a table in the current database.
Syntax-
Create table table_name(
column1 datatype,
column2 datatype,
column3 datatype
);

5] Drop Table
It is used to delete a table from the current database.
Syntax-
ALTER Drop Table table_name:

6] Alter Table
It is used to del. delete a table from current database
Syntax-
ALTER TABLE table_name,
ADD column_name datatype,

7] Show Table-
It is used to list the table for which you have access to privileges
Syntax-
Show tablename;

8] Describe Table
It is used to describe the structure of the table.
Syntax-
Desc Table_name;

9] Insert into
It is used to insert a record in a table
Syntax
Insert into Table_name (col1, col2, col3)
values (val1, val2, val3)

4. Describe table:-
It is used to describe the structure of the table.

Syntax:-
Desc DTABLES;

Example:-
Desc ~~Tab~~ student;

Name:- Om Yogendra Singh
Roll No:- SEA154
DBMS

* Experiment No:- 3

Theory:-

DDL Commands:-
The DDL commands are as follows:-
1. Create
2. Alter
3. Drop

1. Create :- This command is used to create a new table in SQL. The user has to give information like table name, column names, and their datatypes.

Syntax:-

```
Create Table
table_name ( column1
datatype,
column2 datatype,
column3 datatype,
....
);
```

Example:-
```
Create table student (
Std_Id int not null,
stu_name varchar(20),
address varchar (30),
primary key (Std_Id)
);
```

2. Alter :- This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

Syntax :-

```
Alter table table_name
add column_name datatype;
```

Example :-

```
Alter table student
add fees int;
```

3. Drop :- This command is used to remove an existing table along with its structure from the Database.

Syntax :-

```
Drop table Tablename;
```

Example :-

```
Drop table department;
```

* Integrity Constraints:-

1. Primary Key:-
Primary Key uniquely identifies each record in a table. It must have unique values and cannot contain nulls.

2. Unique:-
Unique constraint enforces a column or set of column to have unique values. If a column has a unique constraint, it means that particular column cannot have duplicate values in a table.

3. Not Null:-
Not Null constraint makes sure that a column does not hold NULL value. When we don't provide value for a particular column while inserting a record into a table, it takes NULL value by default.

4. Check:-
The constraint is used for specifying range of values for a particular column of a table. When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

5. Default:-
The default constraint provides a default value to a column when there is no value provided while inserting a record into a table.

Results:-

1. Show databases:-

```
Enter password: ************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1
mysql> create databases SEA170;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'databases SEA170' at line 1
mysql> create database SEA170;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sea170             |
| sys                |
| world              |
+--------------------+
7 rows in set (0.01 sec)

mysql> Use SEA170;
Database changed
```

2. Show tables:-

```
mysql> show tables;
+------------------+
| Tables_in_sea170 |
+------------------+
| airport          |
| flight           |
| worker           |
+------------------+
3 rows in set (0.00 sec)
```

3. Describe Table:-

```
mysql> create table worker(
    -> w_id int not null,
    -> w_name varchar(20),
    -> w_payment int not null,
    -> primary key(w_id)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> desc worker;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| w_id      | int         | NO   | PRI | NULL    |       |
| w_name    | varchar(20) | YES  |     | NULL    |       |
| w_payment | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

\

```
mysql> create table airport(
    -> a_id int not null,
    -> a_name varchar(20),
    -> a_city varchar(20),
    -> primary key(a_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc airport
    -> desc airport;
ERROR 1064 (42000): You have an error in your SQL syntax; check the ma
mysql> ;
ERROR:
No query specified

mysql> desc airport;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| a_id   | int         | NO   | PRI | NULL    |       |
| a_name | varchar(20) | YES  |     | NULL    |       |
| a_city | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
) at line 3
mysql> create table flight(
    -> f_id int not null,
    -> f_ArrivalTime int not null,
    -> f_DepartureTime int not null,
    -> primary key(f_id)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc flight;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| f_id            | int  | NO   | PRI | NULL    |       |
| f_ArrivalTime   | int  | NO   |     | NULL    |       |
| f_DepartureTime | int  | NO   |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

DDL Commands:-

1.Create:-

```
mysql> create table worker(
    -> w_id int not null,
    -> w_name varchar(20),
    -> w_payment int not null,
    -> primary key(w_id)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> desc worker;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| w_id      | int         | NO   | PRI | NULL    |       |
| w_name    | varchar(20) | YES  |     | NULL    |       |
| w_payment | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
mysql> create table airport(
    -> a_id int not null,
    -> a_name varchar(20),
    -> a_city varchar(20),
    -> primary key(a_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc airport
    -> desc airport;
ERROR 1064 (42000): You have an error in your SQL syntax; check the ma
mysql> ;
ERROR:
No query specified

mysql> desc airport;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| a_id   | int         | NO   | PRI | NULL    |       |
| a_name | varchar(20) | YES  |     | NULL    |       |
| a_city | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
) at line 3
mysql> create table flight(
    -> f_id int not null,
    -> f_ArrivalTime int not null,
    -> f_DepartureTime int not null,
    -> primary key(f_id)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc flight;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| f_id            | int  | NO   | PRI | NULL    |       |
| f_ArrivalTime   | int  | NO   |     | NULL    |       |
| f_DepartureTime | int  | NO   |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

2. Alter:-

```
mysql> alter table airport
    -> add state varchar(20);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc airport;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| a_id   | int         | NO   | PRI | NULL    |       |
| a_name | varchar(20) | YES  |     | NULL    |       |
| a_city | varchar(20) | YES  |     | NULL    |       |
| state  | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
mysql> alter table worker
    -> add address varchar(20);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc worker;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| w_id      | int         | NO   | PRI | NULL    |       |
| w_name    | varchar(20) | YES  |     | NULL    |       |
| w_payment | int         | NO   |     | NULL    |       |
| address   | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
mysql> alter table flight
    -> add f_name varchar(20);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc flight;
+----------------+------------+------+-----+---------+-------+
| Field          | Type       | Null | Key | Default | Extra |
+----------------+------------+------+-----+---------+-------+
| f_id           | int        | NO   | PRI | NULL    |       |
| f_ArrivalTime  | int        | NO   |     | NULL    |       |
| f_DepartureTime| int        | NO   |     | NULL    |       |
| f_name         | varchar(20)| YES  |     | NULL    |       |
+----------------+------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3. Drop:-

```
mysql> drop table worker;
Query OK, 0 rows affected (0.07 sec)

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

mysql> drop table flight;
Query OK, 0 rows affected (0.05 sec)
```

INTEGRITY CONTRAINTS:-

1. Primary Key:-

```
mysql> create table worker(
    -> w_id int not null,
    -> w_name varchar(20),
    -> w_payment int not null,
    -> primary key(w_id)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> desc worker;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| w_id      | int         | NO   | PRI | NULL    |       |
| w_name    | varchar(20) | YES  |     | NULL    |       |
| w_payment | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

2. Unique:-

```
mysql> create table airport1(
    -> a_id int unique,
    -> a_name varchar(20) default 'Berlin',
    -> check(a_id>=101)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc airport1;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| a_id   | int         | YES  | UNI | NULL    |       |
| a_name | varchar(20) | YES  |     | Berlin  |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

3. Not Null:-

```
mysql> create table worker(
    -> w_id int not null,
    -> w_name varchar(20),
    -> w_payment int not null,
    -> primary key(w_id)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> desc worker;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| w_id      | int         | NO   | PRI | NULL    |       |
| w_name    | varchar(20) | YES  |     | NULL    |       |
| w_payment | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

4. Check:-

```
mysql> create table airport1(
    -> a_id int unique,
    -> a_name varchar(20) default 'Berlin',
    -> check(a_id>=101)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc airport1;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| a_id   | int         | YES  | UNI | NULL    |       |
| a_name | varchar(20) | YES  |     | Berlin  |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

5. Default:-

```
mysql> create table airport1(
    -> a_id int unique,
    -> a_name varchar(20) default 'Berlin',
    -> check(a_id>=101)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc airport1;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| a_id   | int         | YES  | UNI | NULL    |       |
| a_name | varchar(20) | YES  |     | Berlin  |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

Department of Computer Engineering **Class: S.E. (Computer) (Div- A) (Sem-IV) Academic Year: 2021-22 (EVEN)**
Subject: Database Management System Lab
Course Code: CSL402
Experiment No.: 4
Experiment Title: Apply DML commands for the specified system.
Name of Student: Atharva Vichare
Student Roll No.: SEA170

CASE STUDY:- Airport Management System

* Experiment -4

Title-
Apply DML commands for the specified system.

Theory-
The SQL commands that deals with the manipulation of data present in the database belongs to DML. and this includes most of the SQL ~~sta~~ statements. It is the component of the SQL statement that controls access to data end to the database.

List of DML commands-

1) Update-
The update statement is used to modify the existing records in a table.
Syntax-
Update table_name
set column1 = value1, column2 = value2, ...
where condition;

2) Delete
The delete statement is used to delet

existing records in a table.
Syntax-
Delete from table name where condition.

3] Delete clause
It is used to delete all records from table in current database.
Syntax:
Delete from table_Name;

4] Delete with where clause
It is used to delete for relative records from a table in current database.
Syntax:
Delete from table_name where condition;

5] Select clause
It is used to select a tuple/row.
Syntax select * from table_name

6] Where clause
It is used to select a tuple/row with a condition from a table
Syntax select * from Table_name
where condition

Results:-

1.Where Clause:-

```
mysql> select w_id from worker
    -> where w_payment < 9000;
+------+
| w_id |
+------+
|   21 |
|   22 |
+------+
2 rows in set (0.00 sec)

mysql> select * from airport
    -> where a_id = 2001;
+------+----------------+--------+
| a_id | a_name         | a_city |
+------+----------------+--------+
| 2001 | Mumbai Airport | Mumbai |
+------+----------------+--------+
1 row in set (0.00 sec)
```

2.Select:-

```
mysql> select w_name from worker;
+----------------+
| w_name         |
+----------------+
| Kamlesh Mishra |
| Ramesh Gupta   |
| Ramesh Jaiswal |
+----------------+
3 rows in set (0.00 sec)

mysql> select w_name from worker;
+----------------+
| w_name         |
+----------------+
| Kamlesh Mishra |
| Ramesh Gupta   |
| Ramesh Jaiswal |
+----------------+
3 rows in set (0.00 sec)

mysql> select a_city from airport;
+-----------+
| a_city    |
+-----------+
| Mumbai    |
| Pune      |
| Bangalore |
| Chennai   |
| Kochin    |
+-----------+
5 rows in set (0.00 sec)
```

3. Insert:-

```
mysql> insert into worker values(21,"Kamlesh Mishra",7000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into worker values(22,"Ramesh Gupta",8000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into worker values(29,"Ramesh Jaiswal",9000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from worker;
+------+---------------+-----------+
| w_id | w_name        | w_payment |
+------+---------------+-----------+
|   21 | Kamlesh Mishra |      7000 |
|   22 | Ramesh Gupta   |      8000 |
|   29 | Ramesh Jaiswal |      9000 |
+------+---------------+-----------+
3 rows in set (0.00 sec)
```

4. Update:-

```
mysql> update airport
    -> set a_city = "Thiruvanathapuram"
    -> where a_id = 2009;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from airport;
+------+------------------+-------------------+
| a_id | a_name           | a_city            |
+------+------------------+-------------------+
| 2001 | Mumbai Airport   | Mumbai            |
| 2004 | Pune Airport     | Pune              |
| 2005 | Bangalore Airport | Bangalore        |
| 2006 | Chennai Airport  | Chennai           |
| 2009 | Kochin Airport   | Thiruvanathapuram |
+------+------------------+-------------------+
5 rows in set (0.00 sec)
```

5. Delete command:-

```
mysql> select * from worker;
+------+--------------+-----------+
| w_id | w_name       | w_payment |
+------+--------------+-----------+
|   22 | Ramesh Gupta |      8000 |
|   29 | Ramesh Desai |     89889 |
+------+--------------+-----------+
2 rows in set (0.00 sec)

mysql> delete from flight
    -> where f_id = 2122;
Query OK, 1 row affected (0.01 sec)

mysql> select * from flight;
+------+---------------+-----------------+
| f_id | f_ArrivalTime | f_DepartureTime |
+------+---------------+-----------------+
| 3022 | 20:33         | 21:35           |
| 3042 | 10:33         | 11:35           |
| 3922 | 06:33         | 08:20           |
| 3942 | 12:33         | 01:35           |
+------+---------------+-----------------+
4 rows in set (0.00 sec)
```

Department of Computer Engineering
Class: S.E. (Computer) (Div- A) (Sem-IV)
Academic Year: 2021-22 (EVEN)

Subject: Database Management System Lab

Course Code: CSL402

Experiment No.: 5

Experiment Title:

Perform simple queries, string manipulation operations and aggregate functions.

Name of Student: ATHARVA S. VICHARE

Student Roll No.: SEA170

Experiment no 5

Theory -

Connectives in SQL -
SQL allows the use of logical connectives and or and the not operands of the logical connectives can be expressions involving the comparison operators.

1] AND Operators -
The AND Operators displays a record, if all conditions are separated by 'AND' is True
Syntax -
SELECT column1, column2.
from table_name
WHERE condition1 AND condition2;

2] OR Operators -
The OR Operators displays a record, if all conditions are seperated by 'OR'.
Syntax -
SELECT column1, column2
from table_name
WHERE condition1 OR condition2;

3] NOT Operator
The 'NOT' Operator displays a record if the condition is not true.
Syntax

Syntax-
SELECT column1, column2
FROM Table_name
where NOT condition;

* 1] Between Comparison Operators
The 'BETWEEN' Operator selects value within given range. The 'BETWEEN' operator is inclusive, starting and ending values are included

Syntax-
SELECT column1, ~~column2~~
FROM Table_name
WHERE column1 BETWEEN value1 AND value2;

2] String Operation
SQL specifies string by enclosing them in single quotes.

3] Like Operator
Used in where clause to search specific pattern in column.
i) % - represents 0 or multiple characters
ii) _ - represent single character.

Syntax-

```
SELECT column1, column2
FROM table_name
WHERE column1 like <pattern>;
```

* Aggregate Functions-
Aggregate functions are functions that takes collection of values as input and returns single value as output.

1] Avg-
It is used to find average value
Syntax-

```
SELECT avg (column1)
FROM table_name
WHERE column = '<value>';
```

2] COUNT,
It is used to find number of values
Syntax

```
SELECT COUNT(*)
FROM table_name;
```

3] SUM
It is used to find sum of values
Syntax-

```
SELECT SUM(column_1)
FROM table_name;
```

4] MIN

It is used to find minimum value

Syntax -

```
SELECT MIN(column1)
FROM table_name;
```

5] MAX -

It is used to find maximum value

Syntax -

```
SELECT MAX (column1)
FROM table name;
```

MIN AGGREGATE FUNCTION

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select min(w_payment) from worker;
+----------------+
| min(w_payment) |
+----------------+
|           8000 |
+----------------+
1 row in set (0.00 sec)
```

MAX AGGREGATE FUNCTION

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select max(w_payment) from worker;
+---------------+
| max(w_payment) |
+---------------+
|         89889 |
+---------------+
1 row in set (0.00 sec)
```

LIKE STRING FUNCTION

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select w_name from worker
    -> where w_name like 'P%';
+----------------+
| w_name         |
+----------------+
| Paresh         |
| Paritosh Gowda |
+----------------+
2 rows in set (0.00 sec)

mysql> select w_name from worker
    -> where w_name like 'A%';
+-------------+
| w_name      |
+-------------+
| Abhi Gavkar |
+-------------+
1 row in set (0.00 sec)

mysql> select w_name from worker
    -> where w_name like 'bh%';
Empty set (0.00 sec)

mysql> select w_name from worker
    -> where w_name like '%bh%';
+-------------+
| w_name      |
+-------------+
| Abhi Gavkar |
+-------------+
1 row in set (0.00 sec)
```

NOT LIKE STRING FUNCTION

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select * from worker
    -> where w_name not like '%d_';
+------+-------------+-----------+-----------+
| w_id | w_name      | w_payment | w_address |
+------+-------------+-----------+-----------+
|   22 | Ramesh Gupta |      8000 | Pune      |
|   23 | Paresh      |      8000 | Surat     |
|   25 | Abhi Gavkar |     89000 | Andheri   |
|   29 | Ramesh Desai |     89889 | Dadar     |
+------+-------------+-----------+-----------+
4 rows in set (0.00 sec)
```

DISTINCT

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select distinct(w_payment)
    -> from worker;
+-----------+
| w_payment |
+-----------+
|      8000 |
|      9000 |
|     89000 |
|     89889 |
+-----------+
4 rows in set (0.00 sec)
```

AVG AGGREGATE FUNCTION

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select avg(w_payment) from worker;
+----------------+
| avg(w_payment) |
+----------------+
|     40777.8000 |
+----------------+
1 row in set (0.00 sec)
```

BETWEEN CLAUSE

```
mysql> SELECT * FROM WORKER;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select w_name
    -> from worker
    -> where w_payment between 8000 and 89000;
+----------------+
| w_name         |
+----------------+
| Ramesh Gupta   |
| Paresh         |
| Paritosh Gowda |
| Abhi Gavkar    |
+----------------+
4 rows in set (0.00 sec)
```

AND & OR CONNECTIVE

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select w_id
    -> from worker
    -> where w_payment >=8000 or w_payment <=89000;
+------+
| w_id |
+------+
|   22 |
|   23 |
|   24 |
|   25 |
|   29 |
+------+
5 rows in set (0.00 sec)
```

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select w_id
    -> from worker
    -> where w_payment >=8000 or w_payment <=89000;
+------+
| w_id |
+------+
|   22 |
|   23 |
|   24 |
|   25 |
|   29 |
+------+
5 rows in set (0.00 sec)
```

SUM AGGREGATE FUNCTION

```
mysql> select * from worker;
+------+---------------+-----------+-----------+
| w_id | w_name        | w_payment | w_address |
+------+---------------+-----------+-----------+
|   22 | Ramesh Gupta  |      8000 | Pune      |
|   23 | Paresh        |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar   |     89000 | Andheri   |
|   29 | Ramesh Desai  |     89889 | Dadar     |
+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select sum(w_payment)
    -> from worker;
+----------------+
| sum(w_payment) |
+----------------+
|         203889 |
+----------------+
1 row in set (0.00 sec)
```

Department of Computer Engineering
Class: S.E. (Computer) (Div- A) (Sem-IV)
Academic Year: 2021-22 (EVEN)

Subject: Database Management System Lab

Course Code: CSL402

Experiment No.: 6

Experiment Title: Execute queries using order by, group by and having clauses.

Name of Student: Atharva Vichare

Student Roll No.: SEA170

Experiment 6

Theory-

* Order By

The order by clause causes the tuples in the result of the query to appear in a sorted array

Syntax:

```
SELECT column1, column2
FROM table_name
WHERE condition
ORDER BY column1 asc/desc;
```

By default the order by clause lists value in ascending order.

* Group By

The attribute or attributes given in the group by clause are used to perform groups. Tuples with the same value on all attributes in the group by clause are placed in one group

Syntax-

```
SELECT column1, column2
FROM table_name
WHERE condition
GROUP BY column1;
```

* Having Clause -
The having clause enables us to specify conditions that filter which group result appear in the results i.e. it works on the subset of group.

Syntax
```
SELECT column1, column2
FROM table_name
GROUP BY column1
HAVING condition;
```

Checking Conditions

There WHERE clause place condition on the selected col columns	The HAVING clause place condition on groups created by GROUP BY clause.

* NULL & NOT NULL
SQL uses the special keyword NULL is a predicate to test for a null value.
Syntax -
```
SELECT column1
FROM table_name
WHERE column1 IS NULL;
```

The predicate is NOT NULL succeeds if the value on which it is applied is not null.

Syntax

```
SELECT column1
FROM table_name
WHERE column1 IS NOT NULL;
```

ORDER BY

```
mysql> select * from worker
    -> order by w_payment asc;
+------+----------------+-----------+-----------+
| w_id | w_name         | w_payment | w_address |
+------+----------------+-----------+-----------+
|   22 | Ramesh Gupta   |      8000 | Pune      |
|   23 | Paresh         |      8000 | Surat     |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   25 | Abhi Gavkar    |     89000 | Andheri   |
|   29 | Ramesh Desai   |     89889 | Dadar     |
+------+----------------+-----------+-----------+
5 rows in set (0.00 sec)
```

ORDER BY DESCENDING

```
mysql> select * from worker
    -> order by w_payment desc;
+------+----------------+-----------+-----------+
| w_id | w_name         | w_payment | w_address |
+------+----------------+-----------+-----------+
|   29 | Ramesh Desai   |     89889 | Dadar     |
|   25 | Abhi Gavkar    |     89000 | Andheri   |
|   24 | Paritosh Gowda |      9000 | Bangalore |
|   22 | Ramesh Gupta   |      8000 | Pune      |
|   23 | Paresh         |      8000 | Surat     |
+------+----------------+-----------+-----------+
5 rows in set (0.00 sec)
```

GROUP BY

```
mysql> select w_name, avg(w_payment)
    -> from worker
    -> group by w_name;
+----------------+-----------------+
| w_name         | avg(w_payment)  |
+----------------+-----------------+
| Ramesh Gupta   |       8000.0000 |
| Paresh         |       8000.0000 |
| Paritosh Gowda |       9000.0000 |
| Abhi Gavkar    |      89000.0000 |
| Ramesh Desai   |      89889.0000 |
+----------------+-----------------+
5 rows in set (0.01 sec)
```

NOT NULL , NULL

```
mysql> select * from airport
    -> where a_id is NULL;
Empty set (0.00 sec)

mysql> select * from airport
    -> where a_id is NOT NULL;
+------+------------------+------------------+-------------+
| a_id | a_name           | a_city           | a_state     |
+------+------------------+------------------+-------------+
| 2001 | Mumbai Airport   | Mumbai           | Maharashtra |
| 2004 | Pune Airport     | Pune             | Maharashtra |
| 2005 | Bangalore Airport| Bangalore        | Karnataka   |
| 2006 | Chennai Airport  | Chennai          | Tamil Nadu  |
| 2009 | Kochin Airport   | Thiruvanathapuram| Kerala      |
+------+------------------+------------------+-------------+
5 rows in set (0.00 sec)
```

HAVING CLAUSE

```
mysql> select avg(w_payment)
    -> from worker
    -> group by w_payment
    -> having avg(w_payment)
    -> >9000
    -> order by w_payment desc;
+-----------------+
| avg(w_payment) |
+-----------------+
|     89889.0000 |
|     89000.0000 |
+-----------------+
2 rows in set (0.00 sec)
```

GROUP BY WHERE

```
mysql> select w_payment from worker
    -> where w_payment > 9000 group by w_name;
+-----------+
| w_payment |
+-----------+
|     89000 |
|     89889 |
+-----------+
2 rows in set (0.00 sec)
```

HAVING GROUP BY

```
mysql> select w_name, avg(w_payment) from worker
    -> group by w_name having avg(w_payment) > 9000;
+--------------+----------------+
| w_name       | avg(w_payment) |
+--------------+----------------+
| Abhi Gavkar  |     89000.0000 |
| Ramesh Desai |     89889.0000 |
+--------------+----------------+
2 rows in set (0.00 sec)
```

Department of Computer Engineering
Class: S.E. (Computer) (Div- A) (Sem-IV)
Academic Year: 2021-22 (EVEN)

Subject: Database Management System Lab

Course Code: CSL402

Experiment No.: 7

Experiment Title:

Implement queries using multiple relations, set and join operations.

Name of Student: Atharva Vichare

Student Roll No.: SEA 170

EXPERIMENT NO 7

Experiment 7

- Dot notation- qualifies an SQL identifier with another SQL identifier of which it is a component. These components are seprated with (.) o symbol

Syntax: i) table_name column_name,
ii) view_name colum_name

- Rename operations -
SQL allows remaining relations and attributes using as clause, oldname as new_name

- Set operations
SQL set operations are performed on two sets of SQL queries.

- Union operations
SQL set operations are performed on two sets of SQL queries.
Syntax - SELECT column_name from table1
union
SELECT column_name from table2

- Intersect Operations
  The Intersect operations corresponds to set instructions operation (∩).
  It allows us to find tuples that are in both input relations
  The input relation
  Syntax:-
  SELECT column_name from table1
  intersect
  SELECT column_name from table2

- Intersect Operations
  The intersect operations corresponds to set instruction operation (∩)

- Join operation in SQL
  A join operation is a constraint product which requires the tuples in the two relations match under some condition followed by selection.

- Natural join
  Natural join matches tuples with the same value for all common attribute and retains only one copy of each common column.
  Synton
  SELECT column(s) name
  FROM Table1 natural join Table2

Outer Joins

- Left outer join - preserves tuples only in the relation named from before (to be left of) the left outer join operation
- Right Outer Join - preserves tuples only in the relation named after (to the right of) the right outer join operation

The full outer join preserves tuples in both operation

The outerjoin works in a manner similar to the join operation but preserve those tuples that would be lost in a join by creating tuples in the result containing null values

LEFT AND RIGHT OUTER JOIN

```
mysql> select * from airport natural left outer join worker;
+---------+------+-----------------+-----------------+------------+------+---------------+-----------+-----------+
| AgentId | a_id | a_name          | a_city          | a_state    | w_id | w_name        | w_payment | w_address |
+---------+------+-----------------+-----------------+------------+------+---------------+-----------+-----------+
|       1 | 2001 | Mumbai Airport  | Mumbai          | Maharashtra|   22 | Ramesh Gupta  |      8000 | Pune      |
|       3 | 2004 | Pune Airport    | Pune            | Maharashtra|   24 | Paritosh Gowda|      9000 | Bangalore |
|       3 | 2005 | Bangalore Airport| Bangalore      | Karnataka  |   24 | Paritosh Gowda|      9000 | Bangalore |
|       4 | 2006 | Chennai Airport | Chennai         | Tamil Nadu |   25 | Abhi Gavkar   |     89000 | Andheri   |
|       5 | 2009 | Kochin Airport  | Thiruvanathapuram| Kerala    |   29 | Ramesh Desai  |     89889 | Dadar     |
+---------+------+-----------------+-----------------+------------+------+---------------+-----------+-----------+
5 rows in set (0.00 sec)

mysql> select * from airport natural right outer join worker;
+---------+------+---------------+-----------+-----------+------+-----------------+-----------------+------------+
| AgentId | w_id | w_name        | w_payment | w_address | a_id | a_name          | a_city          | a_state    |
+---------+------+---------------+-----------+-----------+------+-----------------+-----------------+------------+
|       1 |   22 | Ramesh Gupta  |      8000 | Pune      | 2001 | Mumbai Airport  | Mumbai          | Maharashtra|
|       2 |   23 | Paresh        |      8000 | Surat     | NULL | NULL            | NULL            | NULL       |
|       3 |   24 | Paritosh Gowda|      9000 | Bangalore | 2005 | Bangalore Airport| Bangalore      | Karnataka  |
|       3 |   24 | Paritosh Gowda|      9000 | Bangalore | 2004 | Pune Airport    | Pune            | Maharashtra|
|       4 |   25 | Abhi Gavkar   |     89000 | Andheri   | 2006 | Chennai Airport | Chennai         | Tamil Nadu |
|       5 |   29 | Ramesh Desai  |     89889 | Dadar     | 2009 | Kochin Airport  | Thiruvanathapuram| Kerala    |
+---------+------+---------------+-----------+-----------+------+-----------------+-----------------+------------+
6 rows in set (0.00 sec)
```

RENAME OPERATION

```
mysql> select w_name as worker_name, a_name as air_name
    -> from worker natural join airport;
+----------------+-------------------+
| worker_name    | air_name          |
+----------------+-------------------+
| Ramesh Gupta   | Mumbai Airport    |
| Paritosh Gowda | Pune Airport      |
| Paritosh Gowda | Bangalore Airport |
| Abhi Gavkar    | Chennai Airport   |
| Ramesh Desai   | Kochin Airport    |
+----------------+-------------------+
5 rows in set (0.00 sec)
```

UNION ALL

```
mysql> select a_name from airport union all select w_name from worker;
+-------------------+
| a_name            |
+-------------------+
| Mumbai Airport    |
| Pune Airport      |
| Bangalore Airport |
| Chennai Airport   |
| Kochin Airport    |
| Ramesh Gupta      |
| Paresh            |
| Paritosh Gowda    |
| Abhi Gavkar       |
| Ramesh Desai      |
+-------------------+
10 rows in set (0.00 sec)
```

UNION SET

```
mysql> select a_name from airport union select w_name from worker;
+-------------------+
| a_name            |
+-------------------+
| Mumbai Airport    |
| Pune Airport      |
| Bangalore Airport |
| Chennai Airport   |
| Kochin Airport    |
| Ramesh Gupta      |
| Paresh            |
| Paritosh Gowda    |
| Abhi Gavkar       |
| Ramesh Desai      |
+-------------------+
10 rows in set (0.00 sec)
```

Department of Computer Engineering
Class: S.E. (Computer) (Div- A) (Sem-IV)
Academic Year: 2021-22 (EVEN)

Subject: Database Management System Lab

Course Code: CSL402

Experiment No.: 8

Experiment Title:

Implementation of nested queries and queries with exists operator.

Name of Student: ATHARVA VICHARE

Student Roll No.: SEA 170

* Experiment 8

- Nested Query

In nested query, a query is written inside a query. The result of the inner query is used in execution of outer query. It is embedded in 'WHERE clause'.

Syntax-

```
SELECT <column_name>
FROM <table>
WHERE [condition]
  (select <column_name>
   from <table_name>);
```

Example

```
Select * from food worker
where w_payment > (
select avg (w_payment)
from worker);
```

Subqueries can be used with select, insert, update & delete clause along with operators like =, <, >, <=, >=, IN, BETWEEN etc.

2 IN Operators

The IN operator allows you to specify multiple values in where clause.

Syntax :-

```
SELECT <col_name>
FROM <table_name>
WHERE <column_name> IN (val1, val2, );
```

2 NOT-IN Operators

The NOT-IN operator allows you to specify multiple columns values which are not present in the condition of where clause.

Syntax:

```
SELECT <col_name>
FROM <table_name>
WHERE <col_name> NOT IN (val1, val2, ...);
```

Example:

```
SELECT avg(w_payment >
FROM worker
WHERE w_id not in (
SELECT w_id
FROM
```

```
SELECT w_address
FROM worker
WHERE w_id
NOT in (23, 24);
```

* EXISTS operator -

The exist operator is used to test for the existance of any record in sub query. The EXIST operator returns true if the subquery returns one or more records.

Syntax -

```
SELECT <col_name>
FROM <table_name>
WHERE EXISTS (
    select <col_name>
    FROM <table_name>
    WHERE [condition]);
```

* NOT EXISTS ~

Syntax -

```
SELECT <col_name> FROM <table_name>
WHERE NOT EXISTS (SELECT <col_name>
                  FROM <table_name>
                  WHERE [condition]);
```

EXISTS OPERATOR

```
mysql> select w_name from worker where exists(select * from airport where worker.AgentID = airport.AgentId);
+----------------+
| w_name         |
+----------------+
| Ramesh Gupta   |
| Paritosh Gowda |
| Abhi Gavkar    |
| Ramesh Desai   |
+----------------+
4 rows in set (0.00 sec)
```

NOT EXISTS

```
mysql> select w_name from worker where not exists(select * from airport where worker.AgentID = airport.AgentId);
+--------+
| w_name |
+--------+
| Paresh |
+--------+
1 row in set (0.00 sec)
```

NOT IN OPERATOR

```
mysql> select w_address from worker where w_id in (23,24);
+-----------+
| w_address |
+-----------+
| Surat     |
| Bangalore |
+-----------+
2 rows in set (0.00 sec)

mysql> select w_address from worker where w_id not in (23,24);
+-----------+
| w_address |
+-----------+
| Pune      |
| Andheri   |
| Dadar     |
+-----------+
3 rows in set (0.00 sec)
```

NESTED

```
mysql> select * from worker
    -> where w_payment>(
    -> select avg(w_payment)
    -> from worker);
+------+--------------+-----------+-----------+---------+
| w_id | w_name       | w_payment | w_address | AgentId |
+------+--------------+-----------+-----------+---------+
|   25 | Abhi Gavkar  |     89000 | Andheri   |       4 |
|   29 | Ramesh Desai |     89889 | Dadar     |       5 |
+------+--------------+-----------+-----------+---------+
2 rows in set (0.00 sec)
```

Department of Computer Engineering

Class: S.E. (Computer) (Div- A) (Sem-IV) Academic Year: 2021-22 (EVEN)

Subject: Database Management System Lab

Course Code: CSL402

Experiment No.: 9

Experiment Title: Implementation of views and triggers.

Name of Student: Atharva S. Vichare

Student Roll No.: SEA170

CASE STUDY:- Airport Management System

* Experiment 9

* Theory-

View in SQL

1) The relations used in query are actual relations which are part of logical model stored in the database.

2) It is not desirable for all users to see the entire logical model

The create view command is :-

create view view_name
as <query expression>;

Example

create view Worker_Details w_address, tgt#@
as (select w_id, w_name, w_payment from worker)

Triggers-

A trigger is a stored procedure in database that the system executes automatically as an effect of a modification to the database.

1. before insert bigger

create bigger Worker_Record1
before insect
on worker.

for each row
set new.w_payment = n.w_payment + 1000 ;

ii) After insert trigger :-
create trigger Worker_hobby
after insert
on worker
for w_id =id set id = id - 1 where w_payment=
new.w_payment.

Results:-

1.View in SQL:-

```
mysql> CREATE VIEW Worker_Details as(select w_id, w_name, w_payment, w_address, AgentID from worker);
Query OK, 0 rows affected (0.03 sec)

mysql> SELECT * FROM Worker_Details;
+------+---------------+-----------+-----------+---------+
| w_id | w_name        | w_payment | w_address | AgentID |
+------+---------------+-----------+-----------+---------+
|   22 | Ramesh Gupta  |      8000 | Pune      |       1 |
|   23 | Paresh        |      8000 | Surat     |       2 |
|   24 | Paritosh Gowda |      9000 | Bangalore |       3 |
|   25 | Abhi Gavkar   |     89000 | Andheri   |       4 |
|   29 | Ramesh Desai  |     89889 | Dadar     |       5 |
+------+---------------+-----------+-----------+---------+
5 rows in set (0.00 sec)

mysql> SELECT w_name from Worker_Details where w_id between 22 and 25;
+---------------+
| w_name        |
+---------------+
| Ramesh Gupta  |
| Paresh        |
| Paritosh Gowda |
| Abhi Gavkar   |
+---------------+
4 rows in set (0.00 sec)
```

2.Triggers:-

i) Before insert trigger:-

```
mysql> create trigger Worker_Record1
    -> before insert
    -> on worker
    -> for each row
    -> set new.w_payment = new.w_payment+1000;
Query OK, 0 rows affected (0.07 sec)

mysql> show triggers;
+---------------+--------+--------+-----------------------------------+--------+---------------------+-----------------------------------------------------------+---------------+----------------------+----------------------+-----------------------+
| Trigger       | Event  | Table  | Statement                         | Timing | Created             | sql_mode                                                  | Definer       | character_set_client | collation_connection | Database Collation    |
+---------------+--------+--------+-----------------------------------+--------+---------------------+-----------------------------------------------------------+---------------+----------------------+----------------------+-----------------------+
| Worker_Record1 | INSERT | worker | set new.w_payment = new.w_payment+1000 | BEFORE | 2022-04-22 20:26:32.73 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci    |
+---------------+--------+--------+-----------------------------------+--------+---------------------+-----------------------------------------------------------+---------------+----------------------+----------------------+-----------------------+
1 row in set (0.01 sec)
```

ii) After insert trigger:-

```
mysql> create trigger Worker_hobby
    -> after insert
    -> on worker
    -> for each row
    -> update w_id set id = id-1 where w_payment = new.w_payment;
Query OK, 0 rows affected (0.03 sec)

mysql> show triggers;
+---------------+--------+--------+-----------------------------------------------+--------+---------------------+-----------------------------------------------------------+---------------+----------------------+----------------------+-----------------------+
| Trigger       | Event  | Table  | Statement                                     | Timing | Created             | sql_mode                                                  | Definer       | character_set_client | collation_connection | Database Collation    |
+---------------+--------+--------+-----------------------------------------------+--------+---------------------+-----------------------------------------------------------+---------------+----------------------+----------------------+-----------------------+
| Worker_Record1 | INSERT | worker | set new.w_payment = new.w_payment+1000        | BEFORE | 2022-04-22 20:26:32.73 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci    |
| Worker_hobby  | INSERT | worker | update w_id set id = id-1 where w_payment = new.w_payment | AFTER  | 2022-04-22 20:36:37.63 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci    |
+---------------+--------+--------+-----------------------------------------------+--------+---------------------+-----------------------------------------------------------+---------------+----------------------+----------------------+-----------------------+
2 rows in set (0.01 sec)
```

Department of Computer Engineering **Class: S.E. (Computer) (Div- A) (Sem-IV)** **Academic Year: 2021-22 (EVEN)**
Subject: Database Management System Lab
Course Code: CSL402
Experiment No.: 10
Experiment Title: Implement stored procedure and functions.
Name of Student: Atharva S. Vichare
Student Roll No.: SEA170

Case-Study : Airport Management System

* Experiment no 10

Theory -

Procedure -

A procedure is a collection of pre-compiled SQL statements stored inside the database.

It is a sub-routine or sub-program in the regular computing language. A procedure always contain a name, parameter list and SQL statements.

To create a Procedure.

Example

```
DELIMIERER
CREATE PROCEDURE GETnames()
BEGIN
select w_name from worker;
END //
```

To call stored procedures :-

Example - call Getnames()

* Store function in my SQL

A stored function in MySQL is a set of SQL statements that perform some task operations & return a single

value. It is one of the type of stored program in MySQL, but it has some different that are follow

- The function parameter may contain only the IN parameter, while the procedure can allow IN, OUT, INOUT procedure.
- The store function can return only a single value defined in the function header.
- The stored function may also be called within SQL statement
- It may not procedure a result set.

Example

```
DELIMITER $$
CREATE FUNCTION w_payment_COMP(w_payment INT)
RETURNS VARCHAR(20)
deterministic
begin
DECLARE COST VARCHAR(20);
IF w_payment > 9000 THEN
SET cost = "COST IS MORE";
ELSE
SET COST = "COST IS LESS"
END IF;
RETURN(COST);
END $$
```

* Stored Function Call

Example : Select w_name, w_address
FROM worker;

Results:-

1. Procedure

i) Create procedure command :-
ii) Call a stored procedure :

```
mysql> DELIMITER |
mysql> CREATE PROCEDURE GETnames()
    -> BEGIN
    -> select w_name from worker;
    -> END |
Query OK, 0 rows affected (0.08 sec)

mysql> call GETnames() |
+----------------+
| w_name         |
+----------------+
| Ramesh Gupta   |
| Paresh         |
| Paritosh Gowda |
| Abhi Gavkar    |
| Ramesh Desai   |
+----------------+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.04 sec)
```

2. Functions:-

i)Stored Function in MySQL :-

```
'> DELIMITER $$
'> CREATE FUNCTION w_payment_COMP(w_payment INT)
'> RETURNS VARCHAR(20)
'> deterministic
'> begin
'> DECLARE COST VARCHAR(20);
'> IF w_payment > 9000 THEN
'> SET COST = "COST IS MORE";
'> ELSE
'> SET COST = "COST IS LESS";
'> END IF;
'> RETURN (COST);
'> END$$
```

ii) Stored function call:-

```
mysql> use SEA170;
Database changed
mysql> SELECT w_name, w_address
    -> FROM worker;
+----------------+-----------+
| w_name         | w_address |
+----------------+-----------+
| Ramesh Gupta   | Pune      |
| Paresh         | Surat     |
| Paritosh Gowda | Bangalore |
| Abhi Gavkar    | Andheri   |
| Ramesh Desai   | Dadar     |
+----------------+-----------+
5 rows in set (0.00 sec)
```