Name	Adwait S Purao
UID no.	2021300101
Experiment No.	7

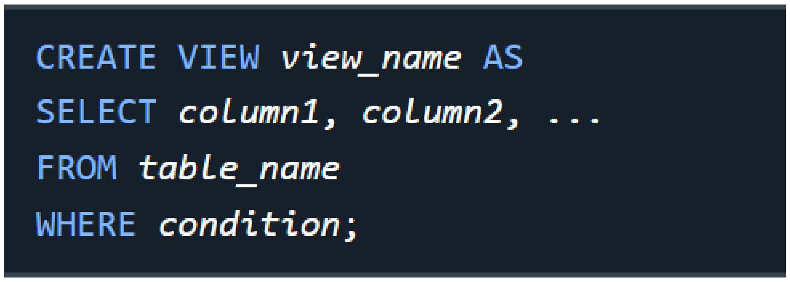
AIM:	To create and query views in MySQL
Program 1	
PROBLEM STATEMENT :	To create various views and perform queries on them in MySQL
Theory :	**Views** View is a data object which does not contain any data. Contents of the view are the resultant of a base table. They are operated just like base table but they don't contain any data of their own. The difference between a view and a table is that views are definitions built on top of other tables (or views). If data is changed in the underlying table, the same change is reflected in the view. A view can be built on top of a single or multiple tables. **Why views?** Views can be effective copies of base tables. Views can have column names and expressions. You can use any clauses in views. Views can be used in INSERT/UPDATE/DELETE. Views can contain expressions in the select list. Views can be views of views. **Restrictions on View definition** The SELECT statement cannot contain a subquery in the FROM clause. The SELECT statement cannot refer to system or user variables. Within a stored program, the definition cannot refer to program parameters or local variables. The SELECT statement cannot refer to prepared statement parameters. Any table or view referred to in the definition must exist. The definition cannot refer to a TEMPORARY table, and you cannot create a TEMPORARY view. Any tables named in the view definition must exist at definition time.

You cannot associate a trigger with a view.
Aliases for column names in the SELECT statement are checked against the maximum column length of 64 characters (not the maximum alias length of 256 characters).

SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.
A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

CREATE VIEW Syntax



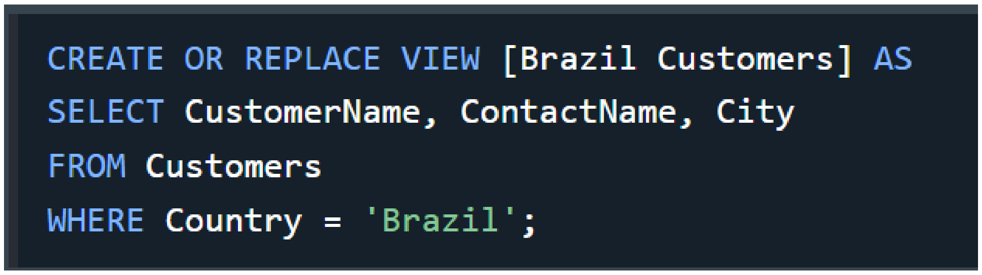
```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Note: A view always shows up-to-date data! The database engine recreates the view, every time a user queries it.

SQL Updating a View

A view can be updated with the CREATE OR REPLACE VIEW statement.

SQL CREATE OR REPLACE VIEW Syntax



```
CREATE OR REPLACE VIEW [Brazil Customers] AS
SELECT CustomerName, ContactName, City
FROM Customers
WHERE Country = 'Brazil';
```

SQL Dropping a View

A view is deleted with the DROP VIEW statement.

SQL DROP VIEW Syntax

```
DROP VIEW view_name;
```

SQL Inserting into a View

We can insert a row in a View in a same way as we do in a table. We can use the INSERT INTO statement of SQL to insert a row in a View

SQL Inserting into a View Syntax

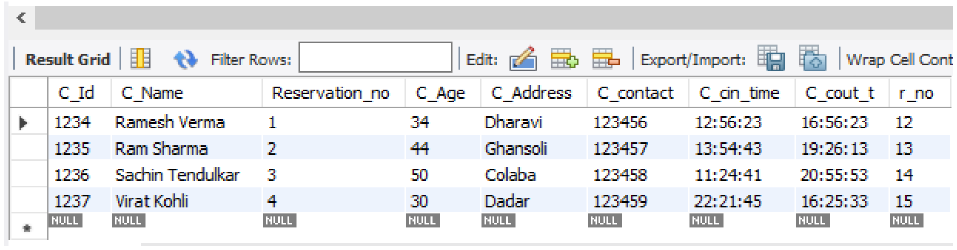
```
INSERT INTO view_name(column1, column2 , column3,..)
VALUES(value1, value2, value3..);

view_name: Name of the View
```

Queries

Query 1: Creation of a view
1)
Statement: A view of customer table is created

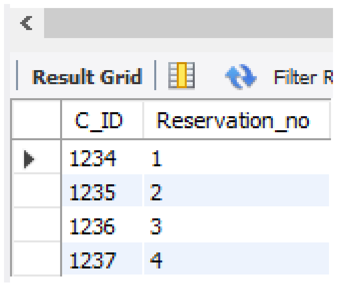
Code:
CREATE VIEW cust_view AS
SELECT C_ID , Reservation_no
FROM customer ;

Original Table:



C_Id	C_Name	Reservation_no	C_Age	C_Address	C_contact	C_cin_time	C_cout_t	r_no
1234	Ramesh Verma	1	34	Dharavi	123456	12:56:23	16:56:23	12
1235	Ram Sharma	2	44	Ghansoli	123457	13:54:43	19:26:13	13
1236	Sachin Tendulkar	3	50	Colaba	123458	11:24:41	20:55:53	14
1237	Virat Kohli	4	30	Dadar	123459	22:21:45	16:25:33	15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output:



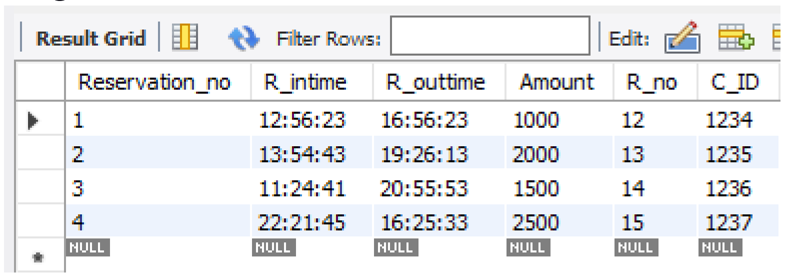
Result Grid | Filter R

	C_ID	Reservation_no
▶	1234	1
	1235	2
	1236	3
	1237	4

2)
Statement: A view of Reservation table is created

Code:
CREATE VIEW reser_view AS
SELECT C_ID , Reservation_no , R_no
FROM customer ;

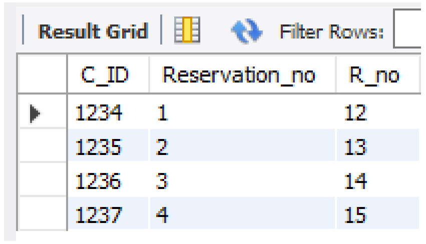
Original Table:



Result Grid | Filter Rows: | Edit:

	Reservation_no	R_intime	R_outtime	Amount	R_no	C_ID
▶	1	12:56:23	16:56:23	1000	12	1234
	2	13:54:43	19:26:13	2000	13	1235
	3	11:24:41	20:55:53	1500	14	1236
	4	22:21:45	16:25:33	2500	15	1237
*	NULL	NULL	NULL	NULL	NULL	NULL

Output:



Result Grid | Filter Rows:

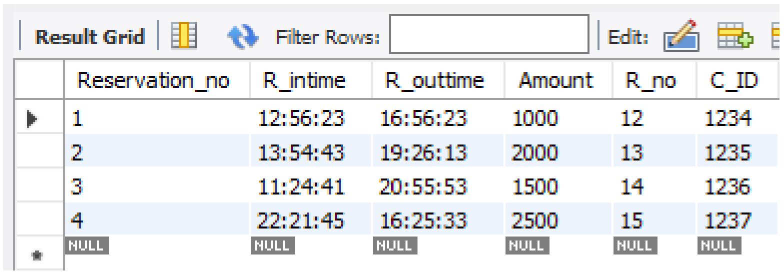
	C_ID	Reservation_no	R_no
▶	1234	1	12
	1235	2	13
	1236	3	14
	1237	4	15

Query 2: Creating a view with natural join

Statement: A new view is created as a natural join of two views

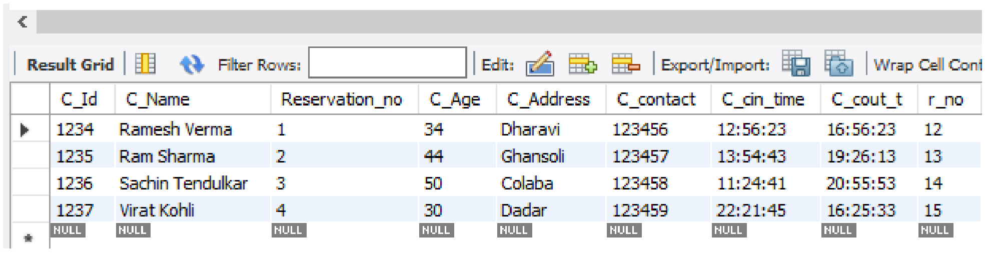
Code:
create view cust1_view
as select c_id , reservation_no,R_no

	from reservation natural join customer; **Original Table:** **Reservation table**



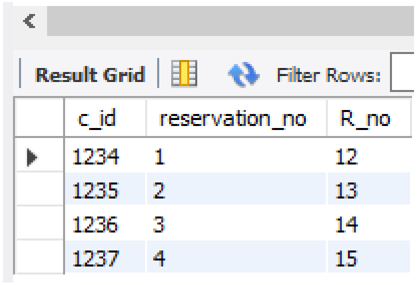
Reservation_no	R_intime	R_outtime	Amount	R_no	C_ID
1	12:56:23	16:56:23	1000	12	1234
2	13:54:43	19:26:13	2000	13	1235
3	11:24:41	20:55:53	1500	14	1236
4	22:21:45	16:25:33	2500	15	1237
NULL	NULL	NULL	NULL	NULL	NULL

Customer table



C_Id	C_Name	Reservation_no	C_Age	C_Address	C_contact	C_cin_time	C_cout_t	r_no
1234	Ramesh Verma	1	34	Dharavi	123456	12:56:23	16:56:23	12
1235	Ram Sharma	2	44	Ghansoli	123457	13:54:43	19:26:13	13
1236	Sachin Tendulkar	3	50	Colaba	123458	11:24:41	20:55:53	14
1237	Virat Kohli	4	30	Dadar	123459	22:21:45	16:25:33	15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output:



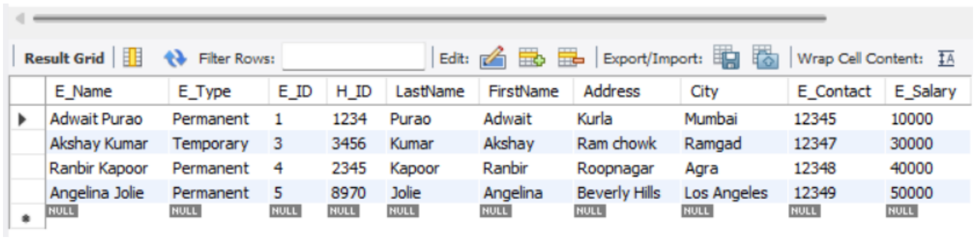
c_id	reservation_no	R_no
1234	1	12
1235	2	13
1236	3	14
1237	4	15

Query 3: Cross join on views employee and hotel

Statement:A new view employ_view is created by the cross join of employee and hotel table

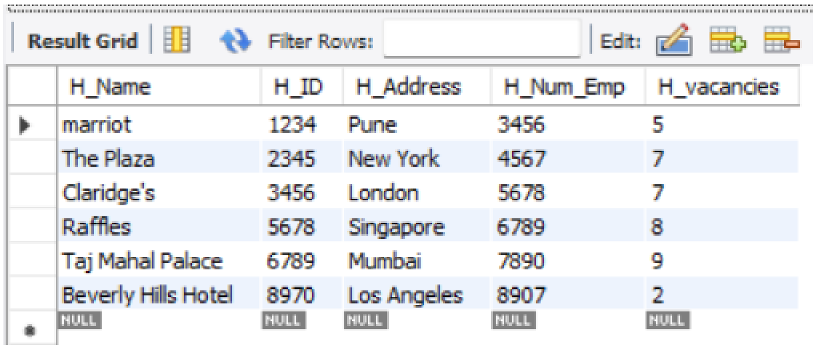
Code:
create view employ_view
as select e_id,e_name,e_type
from employee cross join hotel_info;

Original view:
Employee Table


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

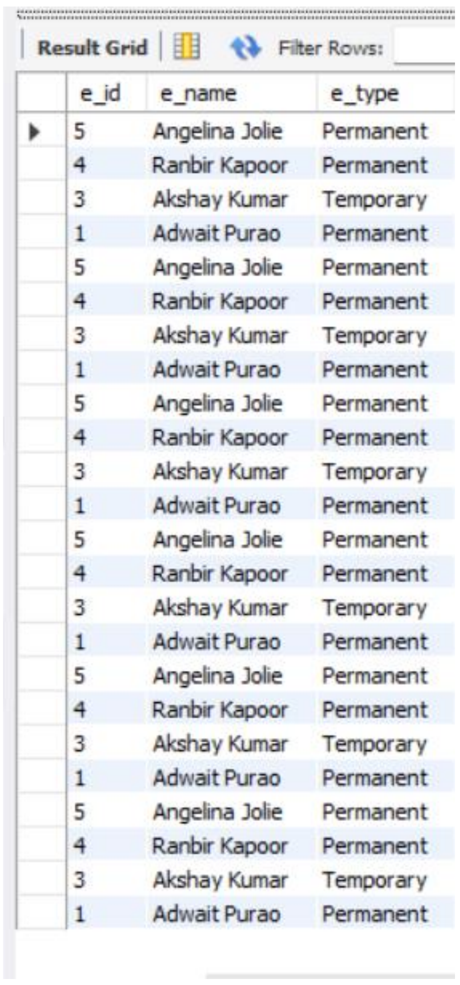
E_Name	E_Type	E_ID	H_ID	LastName	FirstName	Address	City	E_Contact	E_Salary
Adwait Purao	Permanent	1	1234	Purao	Adwait	Kurla	Mumbai	12345	10000
Akshay Kumar	Temporary	3	3456	Kumar	Akshay	Ram chowk	Ramgad	12347	30000
Ranbir Kapoor	Permanent	4	2345	Kapoor	Ranbir	Roopnagar	Agra	12348	40000
Angelina Jolie	Permanent	5	8970	Jolie	Angelina	Beverly Hills	Los Angeles	12349	50000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Hotel table


Result Grid | Filter Rows: | Edit:

H_Name	H_ID	H_Address	H_Num_Emp	H_vacancies
marriot	1234	Pune	3456	5
The Plaza	2345	New York	4567	7
Claridge's	3456	London	5678	7
Raffles	5678	Singapore	6789	8
Taj Mahal Palace	6789	Mumbai	7890	9
Beverly Hills Hotel	8970	Los Angeles	8907	2
NULL	NULL	NULL	NULL	NULL

Output:


Result Grid | Filter Rows:

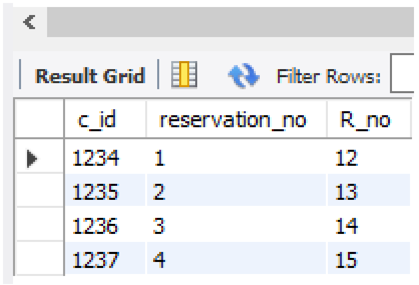
e_id	e_name	e_type
5	Angelina Jolie	Permanent
4	Ranbir Kapoor	Permanent
3	Akshay Kumar	Temporary
1	Adwait Purao	Permanent
5	Angelina Jolie	Permanent
4	Ranbir Kapoor	Permanent
3	Akshay Kumar	Temporary
1	Adwait Purao	Permanent
5	Angelina Jolie	Permanent
4	Ranbir Kapoor	Permanent
3	Akshay Kumar	Temporary
1	Adwait Purao	Permanent
5	Angelina Jolie	Permanent
4	Ranbir Kapoor	Permanent
3	Akshay Kumar	Temporary
1	Adwait Purao	Permanent
5	Angelina Jolie	Permanent
4	Ranbir Kapoor	Permanent
3	Akshay Kumar	Temporary
1	Adwait Purao	Permanent
5	Angelina Jolie	Permanent
4	Ranbir Kapoor	Permanent
3	Akshay Kumar	Temporary
1	Adwait Purao	Permanent

Query 4: Dropping a view

Statement:
A view is completely deleted

Code:
drop view cust1_view;

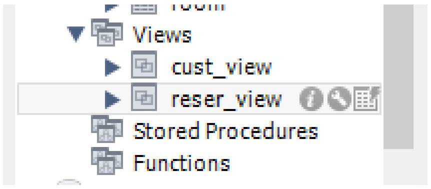
Original view:


Result Grid | Filter Rows:

c_id	reservation_no	R_no
1234	1	12
1235	2	13
1236	3	14
1237	4	15

	Output:

Output:

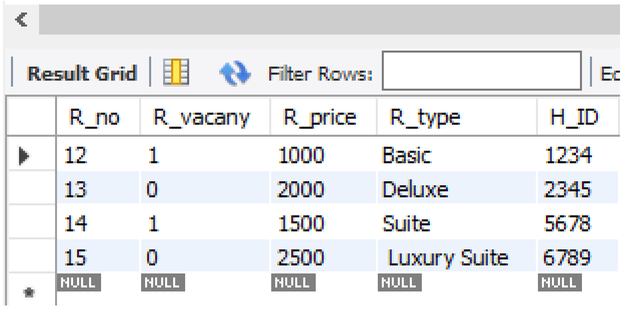


Query 5: Order by in views

Statement:
Ordering the values in the view as per the price

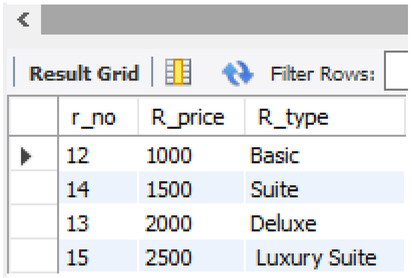
Code:
CREATE VIEW room_view AS
SELECT r_no,R_price,R_type
FROM room
ORDER BY R_price;

Original Table:



R_no	R_vacany	R_price	R_type	H_ID
12	1	1000	Basic	1234
13	0	2000	Deluxe	2345
14	1	1500	Suite	5678
15	0	2500	Luxury Suite	6789
NULL	NULL	NULL	NULL	NULL

Output:



r_no	R_price	R_type
12	1000	Basic
14	1500	Suite
13	2000	Deluxe
15	2500	Luxury Suite

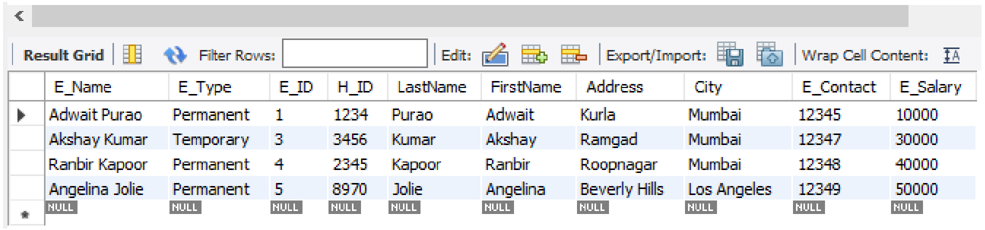
Query 6: Updating a view

Statement: A new view is created with the following parameters

Code:

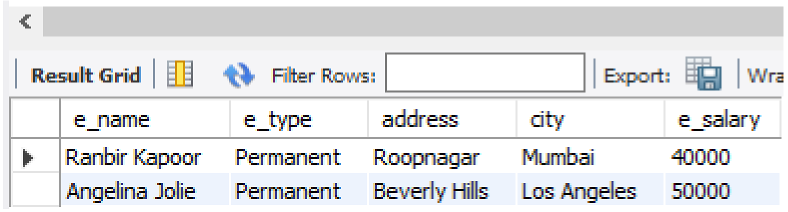
```
CREATE OR REPLACE VIEW emp_view AS
SELECT e_name,e_type, address, city ,e_salary
FROM employee
WHERE E_Salary>35000;
```

Original Table:(Employee Table)



E_Name	E_Type	E_ID	H_ID	LastName	FirstName	Address	City	E_Contact	E_Salary
Adwait Purao	Permanent	1	1234	Purao	Adwait	Kurla	Mumbai	12345	10000
Akshay Kumar	Temporary	3	3456	Kumar	Akshay	Ramgad	Mumbai	12347	30000
Ranbir Kapoor	Permanent	4	2345	Kapoor	Ranbir	Roopnagar	Mumbai	12348	40000
Angelina Jolie	Permanent	5	8970	Jolie	Angelina	Beverly Hills	Los Angeles	12349	50000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output:



e_name	e_type	address	city	e_salary
Ranbir Kapoor	Permanent	Roopnagar	Mumbai	40000
Angelina Jolie	Permanent	Beverly Hills	Los Angeles	50000

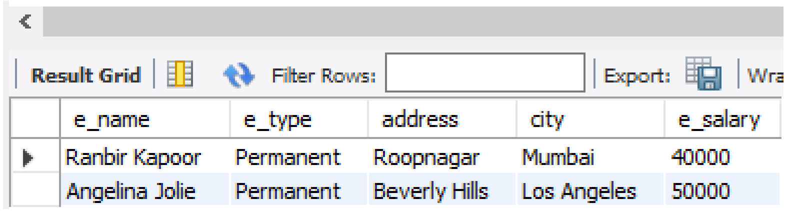
Query 7: Deleting from a view

Statement: Deleting entry of Ranbir Kapoor from emp_view table

Code:

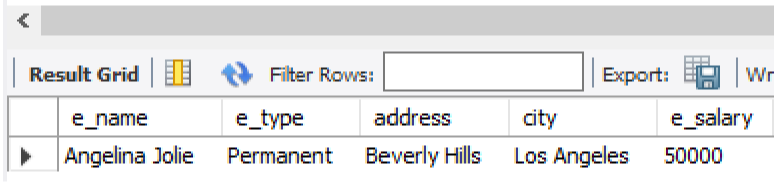
```
DELETE FROM emp_view
WHERE E_NAME="Ranbir Kapoor";
```

Original Table:(Emp_view table)



e_name	e_type	address	city	e_salary
Ranbir Kapoor	Permanent	Roopnagar	Mumbai	40000
Angelina Jolie	Permanent	Beverly Hills	Los Angeles	50000

Output:



Result Grid | Filter Rows: | Export: | Wr

	e_name	e_type	address	city	e_salary
▶	Angelina Jolie	Permanent	Beverly Hills	Los Angeles	50000

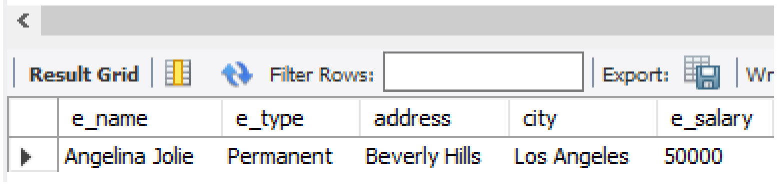
Query 8: Updating a view

Statement:
Updating the salary from 50000 to 100000

Code:

```
UPDATE emp_view
  SET e_salary=100000
  WHERE e_name = 'Angelina Jolie';
```

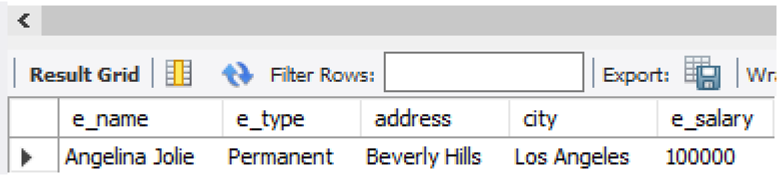
Original Table:



Result Grid | Filter Rows: | Export: | Wr

	e_name	e_type	address	city	e_salary
▶	Angelina Jolie	Permanent	Beverly Hills	Los Angeles	50000

Output:



Result Grid | Filter Rows: | Export: | Wr

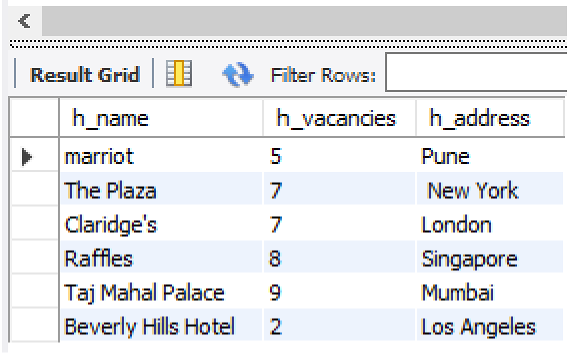
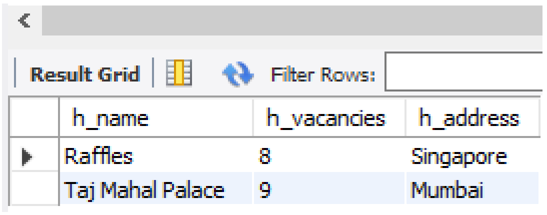
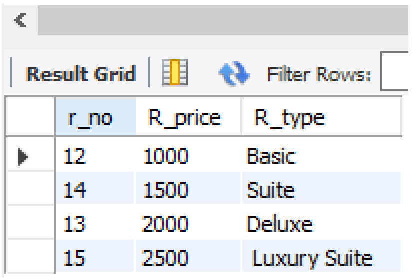
	e_name	e_type	address	city	e_salary
▶	Angelina Jolie	Permanent	Beverly Hills	Los Angeles	100000

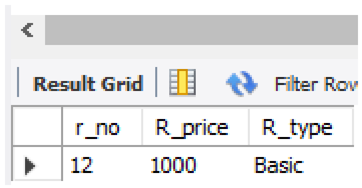
Query 9: Querying hotel_view

Statement:
Selecting hotels having vacancies more than 5

Code:
Creation hotel_view

```
CREATE VIEW hotel_view AS
SELECT h_name,h_vacancies, h_address
FROM hotel_info;
```

	Query: **select * from hotel_view where h_vacancies>7;** **Original Table:(hotel_view)**  Result Grid Filter Rows: | h_name | h_vacancies | h_address | | marriot | 5 | Pune | | The Plaza | 7 | New York | | Claridge's | 7 | London | | Raffles | 8 | Singapore | | Taj Mahal Palace | 9 | Mumbai | | Beverly Hills Hotel | 2 | Los Angeles | **Output:**  Result Grid Filter Rows: | h_name | h_vacancies | h_address | | Raffles | 8 | Singapore | | Taj Mahal Palace | 9 | Mumbai | **Query 10: (Querying room_view)** **Statement: Selecting room type as basic** **Code:** **select * from room_view where r_type="Basic";** **Original Table:**  Result Grid Filter Rows: | r_no | R_price | R_type | | 12 | 1000 | Basic | | 14 | 1500 | Suite | | 13 | 2000 | Deluxe | | 15 | 2500 | Luxury Suite | **Output:**

	 Result Grid Filter Row r_no R_price R_type 12 1000 Basic

Conclusion
In this experiment we learnt that significance of views , we learnt that views can help the user if the same group of tables are accessed continuously . We learnt and implemented various functions on views like CREATE VIEW , REPLACE VIEW , DROP VIEW , INSERT INTO VIEW .