

NAME: SURYAWANSHI RANDHEER AJIT

REG. NO.: 19BCS104

SUBJECT: DBMS LAB ASSIGNMENT – 5

DATABASE : HOTEL (group 2)

Q1) Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL

QUERIES FOR “ANY”

The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows the 'HOTEL' database structure, including tables like T2_Customer, T2_Rooms, T2_Reservations, and T2_SERVICES. The 'SQLQuery1.sql' window contains three queries using the ANY operator:

```
USE HOTEL;

SELECT * FROM T2_Customer WHERE Customer_ID <= ANY(SELECT Customer_ID FROM T2_Rooms WHERE Customer_ID < 3);

SELECT * FROM T2_Rooms WHERE number_of_beds < ANY(SELECT Number_of_guests FROM T2_Reservations);

SELECT Service_name, Service_cost FROM T2_SERVICES WHERE Service_ID >= ANY(SELECT EMPID FROM T2_EMP_INFO WHERE age > 25);
```

The 'Results' pane shows the output of the first two queries. The first query returns two rows of customer data, and the second query returns three rows of room data.

Customer_ID	Customer_Name	Phone_number	City	State	Zipcode	Email_ID
1	Lofflin	8688543748	Nagpur	MP	534201	loff@gmail.com
2	Ram	8688543744	hyderabad	TN	534204	Ram@gmail.com

Room_number	Room_Type	Room_location	number_of_beds	Customer_ID
1	Deluxe	block-2	1	2
2	Economic	block-1	3	1
3	Deluxe	block-2	1	4

Service_name	Service_cost
Transport	8000
Room	4000

The status bar at the bottom indicates 'Query executed successfully.' and 'localhost (15.0 RTM) | DESKTOP-NLAHG8Q\RANDHE... | HOTEL | 00:00:00 | 7 rows'.

QUERIES FOR “AI!”

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The top menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar contains various icons for file operations, query execution, and formatting. The Object Explorer on the left shows the database structure for 'HOTEL', including tables like T2_Customer, T2_SERVICES, T2_CUSTOMER_ADDRESS, T2_EMP_INFO, T2_Reservation, and T2_Rooms. The main query window shows a complex SQL query involving multiple tables and subqueries. The Results pane at the bottom displays the output of the query, which includes a list of customer names and a detailed table of reservation data.

SQLQuery2.sql - localhost.HOTEL (DESKTOP-NLAHG8Q\RANDHEER (55))* - Microsoft SQL Server Manage... Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

Object Explorer

Connect

Databases

System Databases

Database Snapshots

HOTEL

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.T2_Billing

dbo.T2_Customer

dbo.T2_CUSTOMER_ADDR

dbo.T2_CUSTOMERBACKUI

dbo.T2_EMP_INFO

dbo.T2_Reservation

dbo.T2_Rooms

Columns

Room_number (PK, int)

Room_Type (varchar(10))

Room_location (varchar(100))

SQLQuery2.sql - I...8Q\RANDHEER (55))* SQLQuery1.sql - I...8Q\RANDHEER (53))*

```
SELECT CUSTOMER_NAME FROM T2_Customer WHERE Customer_ID <= ALL (SELECT Customer_ID FROM T2_Billing WHERE Room_charge <= ALL (SELECT Reservation_number FROM T2_Reservation WHERE Reservation_number <= ALL (SELECT Customer_ID FROM T2_Rooms WHERE Room_Type='Delux')));
```

75 %

Results Messages

CUSTOMER_NAME
1 Lofflin
2 Ram

Service_ID	Service_name	Service_cost	Reservation_number
1	Food	3000	1

Customer_ID	Street	DNO	Zipcode	City	State
1	RP NAGAR	7-11	102142	Nagpur	Madhya pradesh
2	Sriram nagar	9-12	102132	hyderabad	Telengana
3	JS Nagar	7-13	106132	Lucknow	Utter pradesh
4	Indira NAG...	7-8...	132132	Bengaluru	Karnataka

Query executed successfully. localhost (15.0 RTM) DESKTOP-NLAHG8Q\RANDHE... HOTEL 00:00:00 7 rows

Ready Ln 4 Col 117 Ch 117 INS

QUERIES FOR “Like”

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The top menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar contains icons for opening, saving, and executing queries. The Object Explorer on the left shows the database structure for 'HOTEL', including tables like T2_CUSTOMER_ADDRESS, T2_EMP_INFO, and T2_CUSTOMERBACKUP. The main query window shows three SQL queries using the LIKE operator:

```
use HOTEL;  
SELECT * FROM T2_CUSTOMER_ADDRESS WHERE DNO LIKE '7-%';  
SELECT EMPNAME,AGE FROM T2_EMP_INFO WHERE EMPNAME LIKE '%a%';  
SELECT * FROM T2_CUSTOMER_ADDRESS WHERE STREET LIKE '%NAGAR';
```

The Results pane shows the output of the queries. The first query returns three rows of customer data. The second query returns three rows of employee data. The third query returns one row of customer data.

Customer_ID	Street	DNO	Zipcode	City	State
1	RP NAGAR	7-11	102142	Nagpur	Madhya pradesh
2	JS Nagar	7-13	106132	Lucknow	Utter pradesh
3	Indira NAGAR	7-8-12	132132	Bengaluru	Karnataka

EMPNAME	AGE
Max	22
Jax	62
Nax	29

Customer_ID	Street	DNO	Zipcode	City	State
1	RP NAGAR	7-11	102142	Nagpur	Madhya pradesh

The status bar at the bottom indicates that the query was executed successfully, returning 10 rows in 00:00:00 seconds.

QUERY FOR DIFFERENCE BETWEEN ANY AND ALL

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left shows the 'HOTEL' database structure, including tables like 'T2_Customer' and 'T2_Billing'. The main query window shows the following SQL code:

```
use HOTEL;
SELECT CUSTOMER_NAME FROM T2_Customer WHERE Customer_ID <=ALL (SELECT Customer_ID FROM T2_Billing WHERE Room_char
SELECT CUSTOMER_NAME FROM T2_Customer WHERE Customer_ID <=ANY (SELECT Customer_ID FROM T2_Billing WHERE Room_char
```

The Results pane shows the output of the query, displaying two tables of customer names:

CUSTOMER_NAME
1 Lofflin
2 Ram

CUSTOMER_NAME
1 Lofflin
2 Ram
3 Mahesh
4 Prabha

The status bar at the bottom indicates that the query was executed successfully, returning 6 rows in 00:00:00 seconds.

Q2) One query for each Aggregate function.

The aggregate functions are MIN(), MAX(), COUNT(), AVG(), SUM()

AVG() – return the average of the set

MIN() – returns the minimum value in a set

MAX() – returns the maximum value in set

SUM() – returns the sum of all distinct values of a set

COUNT() – returns the number of items in a set

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'HOTEL' database selected. The right pane shows a query window with the following SQL code:

```
use HOTEL;
SELECT AVG(AGE) AS AVERAGE_AGE FROM T2_EMP_INFO;
SELECT MAX(NUMBER_OF_BEDS) AS MAX_NUMBER_OF_BEDS FROM T2_Rooms;
SELECT MIN(SERVICE_COST) AS MIN_SERVICE_CHARGE FROM T2_SERVICES;
SELECT COUNT(CUSTOMER_ID) FROM T2_Customer WHERE CUSTOMER_NAME LIKE '%a%';
SELECT SUM(ROOM_CHARGE) AS TOTAL_CHARGE FROM T2_Billing;
```

Below the query window, the 'Results' pane shows the output of the query. The results are as follows:

	AVERAGE_AGE
1	37

	MAX_NUMBER_OF_BEDS
1	3

	MIN_SERVICE_CHARGE
1	3000

	(No column name)
1	3

	TOTAL_CHARGE
1	12000

The status bar at the bottom indicates 'Query executed successfully. localhost (15.0 RTM) DESKTOP-NLAHG8Q\RANDHEER... HOTEL 00:00:00 5 rows'.

Q3) Illustrate the usage of order by, group by and having clause (2 queries for each case)

ORDER BY

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'HOTEL' database selected. The right pane shows two SQL queries in separate windows. The first window, 'SQLQuery1.sql', contains a query that selects all columns from 'T2_Customer' and 'T2_EMP_INFO' tables, ordered by 'Customer_Name' and 'AGE' respectively. The second window, 'SQLQuery2.sql', contains a query that selects all columns from 'T2_Customer' and 'T2_EMP_INFO' tables, ordered by 'Customer_Name' and 'AGE' respectively. The results of these queries are displayed in the 'Results' pane at the bottom. The first result set shows customer information, and the second result set shows employee information.

```
use HOTEL;
SELECT * FROM T2_Customer ORDER BY Customer_Name ASC;
SELECT * FROM T2_EMP_INFO ORDER BY AGE DESC;
```

	Customer_ID	Customer_Name	Phone_number	City	State	Zipcode	Email_ID
1	1	Lofflin	8688543748	Nagpur	MP	534201	loff@gmail.com
2	3	Mahesh	8688543746	Lucknow	UP	534205	mah@gmail.com
3	4	Prabha	8688543766	Bengaluru	Karnataka	534201	prab@gmail.com
4	2	Ram	8688543744	hyderabad	TN	534204	Ram@gmail.com

	EMPID	EMPNAME	DOB	AGE
1	2	Jax	1959-04-05	62
2	3	Nax	1992-07-07	29
3	1	Max	1999-03-21	22

Query executed successfully. localhost (15.0 RTM) DESKTOP-NLAHG8Q\RANDHE... HOTEL 00:00:00 7 rows

GROUP BY

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'HOTEL' database selected. The right pane shows a SQL query window with the following query:

```
use HOTEL;  
SELECT NUMBER_OF_BEDS, COUNT(*) AS NUMBER_OF_ROOMS FROM T2_Rooms GROUP BY number_of_beds;  
SELECT ZIPCODE, COUNT(*) FROM T2_Customer GROUP BY Zipcode;
```

The query results are displayed in two tables. The first table shows the results for the first query, and the second table shows the results for the second query.

	NUMBER_OF_BEDS	NUMBER_OF_ROOMS
1	1	2
2	3	1

	ZIPCODE	(No column name)
1	534201	2
2	534204	1
3	534205	1

The status bar at the bottom indicates that the query was executed successfully, returning 5 rows.

HAVING CLAUSE

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'HOTEL', including tables like 'T2_Rooms' and 'T2_Reservation'. The main query window shows the following SQL code:

```
use HOTEL;
SELECT COUNT(Room_number), Room_Type FROM T2_Rooms GROUP BY Room_Type HAVING COUNT(Room_number) >=1;
SELECT COUNT(RESERVATION_NUMBER), LEFT(RESERVATION_DATE,4) FROM T2_Reservation GROUP BY LEFT(Reservation_date,4)
```

The query results are displayed in two tables. The first table shows the count of rooms for each room type, and the second table shows the count of reservations for each reservation date.

	(No column name)	Room_Type
1	2	Deluxe
2	1	Economic

	(No column name)	(No column name)
1	3	1999

The status bar at the bottom indicates that the query was executed successfully, returning 3 rows.

Q4) Use Aggregate function with group by and having

AVG():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left shows the 'HOTEL' database structure, including tables like 'T2_Rooms'. The central query window contains the following SQL code:

```
use HOTEL;
SELECT AVG(NUMBER_OF_BEDS) FROM T2_Rooms GROUP BY Room_location HAVING Room_location LIKE 'BLOCK%';
```

The Results pane at the bottom shows the output of the query, which consists of two rows:

	(No column name)
1	3
2	1

The status bar at the bottom indicates that the query was executed successfully on the 'localhost (15.0 RTM)' instance, returning 2 rows in 00:00:00 seconds.

COUNT():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'HOTEL' database selected. The central pane shows a SQL query in the 'SQLQuery2.sql' window:

```
use HOTEL;
SELECT COUNT(Customer_ID) FROM T2_Reservation GROUP BY Check_in_date HAVING Check_in_date >= '1992-02-03' ;
```

The right pane shows the 'Results' tab with the following data:

(No column name)	
1	2
2	1

The status bar at the bottom indicates: 'Query executed successfully. | localhost (15.0 RTM) | DESKTOP-NLAHG8Q\RANDHE... | HOTEL | 00:00:00 | 2 rows'.

MIN():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'HOTEL' database selected. The right pane shows a query window with the following SQL query:

```
use HOTEL;
SELECT MIN(AGE) FROM T2_EMP_INFO GROUP BY AGE HAVING AGE>25;
```

The query results are displayed in the 'Results' tab, showing two rows of data:

	(No column name)
1	29
2	62

The status bar at the bottom indicates that the query was executed successfully, returning 2 rows in 00:00:00 seconds.

MAX():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'HOTEL' database selected. The right pane shows a query window with the following SQL query:

```
use HOTEL;
SELECT MAX(ROOM_CHARGE) FROM T2_Billing GROUP BY LEFT(Payment_date,7) HAVING LEFT(Payment_date,7) LIKE '2021-%';
```

The query results are displayed in a table with 3 rows:

	(No column name)
1	5000
2	6000
3	3000

The status bar at the bottom indicates: 'Query executed successfully. | localhost (15.0 RTM) | DESKTOP-NLAHG8Q\RANDHE... | HOTEL | 00:00:00 | 3 rows'.

SUM():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The 'Object Explorer' on the left shows the 'HOTEL' database structure, including tables like 'dbo.T2_Customer' and 'dbo.T2_CUSTOMER_ADDR'. The 'Query Editor' window shows a SQL query: `use HOTEL; SELECT SUM(SERVICE_COST) FROM T2_SERVICES GROUP BY Service_cost HAVING SERVICE_COST BETWEEN 4000 AND 6000;`. The 'Results' pane shows a single row with the value 4000. The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

SQLQuery2.sql - localhost.HOTEL (DESKTOP-NLAHG8Q\RANDHEER (55))* - Microsoft SQL Server Manage... Quick Launch (Ctrl+Q)

File Edit View Query Project Tools Window Help

Object Explorer

- Databases
 - System Databases
 - Database Snapshots
 - HOTEL
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.T2_Billing
 - dbo.T2_Customer
 - dbo.T2_CUSTOMER_ADDR
 - dbo.T2_CUSTOMERBACKUP
 - dbo.T2_EMP_INFO
 - Columns
 - EMPID (PK, smallint, n...
 - EMPNAME (varchar(m...
 - DOB (date, null)
 - AGE (Computed, int, n...
 - Keys

SQLQuery2.sql - l...8Q\RANDHEER (55))* SQLQuery1.sql - l...8Q\RANDHEER (53))*

```
use HOTEL;
SELECT SUM(SERVICE_COST) FROM T2_SERVICES GROUP BY Service_cost HAVING SERVICE_COST BETWEEN 4000 AND 6000;
```

75 %

Results Messages

(No column name)
4000

Query executed successfully. localhost (15.0 RTM) DESKTOP-NLAHG8Q\RANDHE... HOTEL 00:00:00 1 rows

Ready Ln 2 Col 106 Ch 106 INS

Q5) Write at least 3 nested queries using order by, group by and having clause.

QUERY:

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left shows the 'HOTEL' database structure, including tables like 'T2_Customer', 'T2_Reservations', and 'T2_SERVICES'. The main query window shows a nested SQL query:

```
USE HOTEL;
SELECT CUSTOMER_NAME, COUNT(*) FROM T2_Customer
WHERE Customer_ID = ANY(
    SELECT Customer_ID FROM T2_Reservations
    WHERE Reservation_number = ANY(
        SELECT Reservation_number FROM T2_SERVICES
        WHERE Service_cost >= 4000
    )
)
GROUP BY Customer_Name HAVING CUSTOMER_NAME LIKE '%a%'
ORDER BY CUSTOMER_NAME DESC;
```

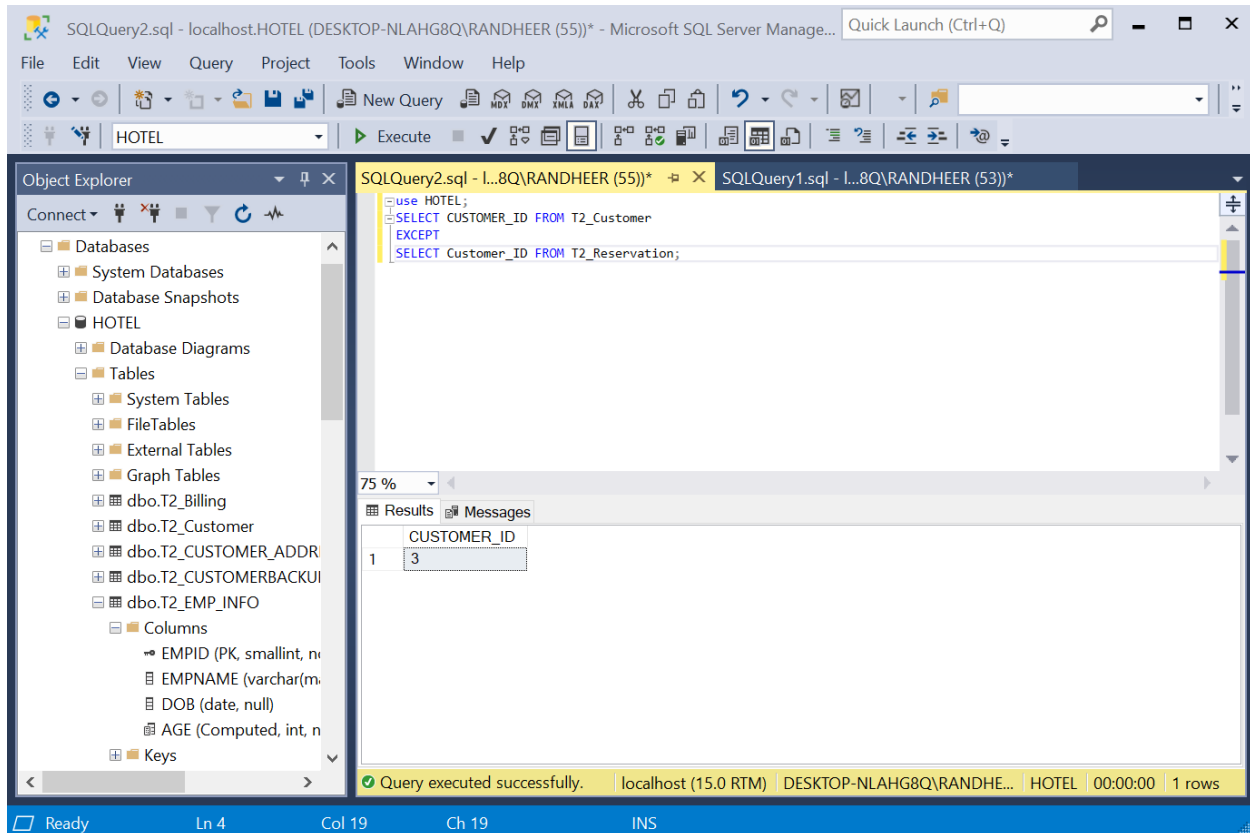
The query results are displayed in a table with the following data:

	CUSTOMER_NAME	(No column name)
1	Prabha	1

The status bar at the bottom indicates 'Query executed successfully.' and shows the execution time as 00:00:00 for 1 row.

Q6) Illustrate the Usage of Except, Exists, Not Exists, Union, Intersection

EXCEPT():



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure for 'HOTEL', including tables like 'dbo.T2_Customer' and 'dbo.T2_Reservation'. The main query window shows the following SQL code:

```
use HOTEL;
SELECT CUSTOMER_ID FROM T2_Customer
EXCEPT
SELECT Customer_ID FROM T2_Reservation;
```

The Results pane shows the output of the query, which is a single row with the value 3 in the 'CUSTOMER_ID' column.

CUSTOMER_ID
3

The status bar at the bottom indicates that the query was executed successfully, returning 1 row.

EXISTS():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'HOTEL' database selected. The right pane shows a SQL query window with the following query:

```
use HOTEL;
SELECT CUSTOMER_ID FROM T2_Rooms
WHERE EXISTS
(SELECT Customer_ID FROM T2_Billing)
ORDER BY CUSTOMER_ID ASC;
```

The query results are displayed in the 'Results' tab, showing three rows of data:

	CUSTOMER_ID
1	1
2	2
3	4

The status bar at the bottom indicates that the query was executed successfully, returning 3 rows.

NOT EXISTS():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'HOTEL' database selected. The right pane shows a SQL query window with the following code:

```
use HOTEL;
SELECT * FROM T2_Customer
WHERE NOT EXISTS
(SELECT Customer_ID FROM T2_Reservation);
```

Below the query window, the 'Results' tab is active, showing a table with the following columns: Customer_ID, Customer_Name, Phone_number, City, State, Zipcode, Email_ID. The table is currently empty, indicating 0 rows returned.

The status bar at the bottom indicates: Query executed successfully. localhost (15.0 RTM) DESKTOP-NLAHG8Q\RANDHEER... HOTEL 00:00:00 0 rows

UNION():

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the 'HOTEL' database selected. The right pane shows a SQL query window with the following code:

```
use HOTEL;  
SELECT CITY FROM T2_CUSTOMER_ADDRESS  
UNION  
SELECT CITY FROM T2_CUSTOMER;
```

The query results are displayed in the bottom right pane, showing a table with one column, 'CITY', and four rows of data:

	CITY
1	Bengaluru
2	hyderabad
3	Lucknow
4	Nagpur

The status bar at the bottom indicates that the query was executed successfully, returning 4 rows.

INTERSECT:

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'HOTEL' database selected. The right pane shows a query window with the following SQL code:

```
use HOTEL;  
SELECT ROOM_CHARGE FROM T2_Billing  
INTERSECT  
SELECT SERVICE_COST FROM T2_SERVICES;
```

The query results are displayed in the 'Results' pane, showing a single row with the value 3000 for the column ROOM_CHARGE.

	ROOM_CHARGE
1	3000

The status bar at the bottom indicates that the query was executed successfully, returning 1 row.

Q7) INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN- 3 queries for each instance

INNER JOIN

The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'HOTEL'. The central query editor contains the following SQL code:

```
USE HOTEL;
SELECT CUSTOMER_NAME, DNO, STREET, T2_Customer.City FROM T2_Customer
T2_CUSTOMER_ADDRESS
ON T2_Customer.Customer_ID = T2_CUSTOMER_ADDRESS.Customer_ID;

SELECT CUSTOMER_NAME, NUMBER_OF_GUESTS, CHECK_IN_DATE, CHECK_OUT_DATE FROM T2_Customer
T2_Reservation
ON T2_Customer.Customer_ID = T2_Reservation.Customer_ID;

SELECT RESERVATION_NUMBER, RESERVATION_DATE, ROOM_TYPE, ROOM_LOCATION FROM T2_Rooms
T2_Reservation
ON T2_Rooms.Room_number = T2_Reservation.Room_number;
```

The Results pane shows the output of the query, which is an INNER JOIN of the three tables. The results are displayed in three separate tables:

CUSTOMER_NAME	NUMBER_OF_GUESTS	CHECK_IN_DATE	CHECK_OUT_DATE
1 Loffin	7-11	RP NAGAR	Nagpur
2 Ram	9-12	Sriram nagar	hyderabad
3 Mahesh	7-13	JS Nagar	Lucknow
4 Prabha	7-8-12	Indira NAGAR	Bengaluru

RESERVATION_NUMBER	RESERVATION_DATE	ROOM_TYPE	ROOM_LOCATION
1 1	1999-02-01	Deluxe	block-2
2 2	1999-02-03	Economic	block-1
3 3	1999-04-01	Deluxe	block-2

The status bar at the bottom indicates that the query was executed successfully on the 'HOTEL' database, returning 3 rows in 00:00:00 seconds.

LEFT OUTER JOIN

SQLQuery2.sql - localhost:1433\HOTEL (DESKTOP-NLAHG8Q\RANDHEER (55)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

- Databases
 - System Databases
 - Database Snapshots
 - HOTEL
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.T2_Billing
 - dbo.T2_Customer
 - dbo.T2_CUSTOMER_ADDR
 - dbo.T2_CUSTOMERBACKUI
 - dbo.T2_EMP_INFO
 - Columns
 - EMPID (PK, smallint, n)
 - EMPNAME (varchar(m
 - DOB (date, null)
 - AGE (Computed, int, n
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics

SQLQuery2.sql - L:\8Q\RANDHEER (55))

```
USE HOTEL;
SELECT * FROM T2_Customer
LEFT OUTER JOIN T2_Rooms
ON T2_Customer.Customer_ID = T2_Rooms.Customer_ID;

SELECT * FROM T2_Customer_ADDRESS
LEFT OUTER JOIN T2_Reservation
ON T2_Reservation.Customer_ID = T2_Customer_ADDRESS.Customer_ID;

SELECT * FROM T2_EMP_INFO
LEFT OUTER JOIN T2_SERVICES
ON T2_SERVICES.SERVICE_ID = T2_EMP_INFO.EMPID;
```

Results

	Customer_ID	Customer_Name	Phone_number	City	State	Zipcode	Email_ID	Room_number	Room_Type	Room_location	number_of_beds	Customer_ID
1	1	Loffin	8888543748	Nagpur	MP	534201	loff@gmail.com	2	Economic	block-1	3	1
2	2	Ram	8888543744	hyderabad	TN	534204	Ram@gmail.com	1	Deluxe	block-2	1	2
3	3	Mahesh	8888543746	Lucknow	UP	534205	mah@gmail.com	NULL	NULL	NULL	NULL	NULL
4	4	Prabha	8888543766	Bengaluru	Karnataka	534201	prab@gmail.com	3	Deluxe	block-2	1	4

	Customer_ID	Street	DNO	Zipcode	City	State	Reservation_number	Check_in_date	Check_out_date	Number_of_guests	Reservation_date	Customer_ID
1	1	RP NAGAR	7-11	102142	Nagpur	Madhya pradesh	2	1999-02-03	1999-02-22	4	1999-02-03	1
2	2	Sriram nagar	9-12	102132	hyderabad	Telangana	1	1999-02-03	1999-02-22	5	1999-02-01	2
3	3	JS NAGAR	7-13	106132	Lucknow	Uttar pradesh	NULL	NULL	NULL	NULL	NULL	NULL
4	4	Indira NAG...	7-8...	132132	Bengaluru	Karnataka	3	1999-04-03	1999-04-04	2	1999-04-01	4

Query executed successfully. localhost (15.0 RTM) | DESKTOP-NLAHG8Q\RANDHE... HOTEL 00:00:00 11 rows

RIGHT OUTER JOIN

The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows two SQL queries using Right Outer Joins. The first query joins T2_Customer and T2_Rooms, and the second joins T2_SERVICES and T2_EMP_INFO. The Results pane shows the output of the first query, displaying columns for Room details and Customer information.

SQL Query 1:

```
SELECT * FROM T2_Customer
LEFT OUTER JOIN T2_CUSTOMER_ADDRESS
ON T2_Customer.Customer_ID = T2_CUSTOMER_ADDRESS.Customer_ID;
```

SQL Query 2:

```
SELECT * FROM T2_SERVICES
LEFT OUTER JOIN T2_EMP_INFO
ON T2_SERVICES.SERVICE_ID = T2_EMP_INFO.EMPID;
```

Results (Query 1):

Room_number	Room_Type	Room_location	number_of_beds	Customer_ID	Customer_ID	Customer_Name	Phone_number	City	State	Zipcode	Email_ID
1	Deluxe	block-2	1	2	2	Ram	8688543744	hyderabad	TN	534204	Ram@gmail.com
2	Economic	block-1	3	1	1	Lofflin	8688543748	Nagpur	MP	534201	loff@gmail.com
3	Deluxe	block-2	1	4	4	Prabha	8688543766	Bengaluru	Karnataka	534201	prab@gmail.com

Results (Query 2):

Reservation_number	Check_in_date	Check_out_date	Number_of_guests	Reservation_date	Customer_ID	Room_number	Customer_ID	Street	DNO	Zipcode	City	Sta
1	1999-02-03	1999-02-22	5	1999-02-01	2	1	2	Sriram nagar	9-12	102132	hyderabad	Tel
2	1999-02-03	1999-02-22	4	1999-02-03	1	2	1	RP NAGAR	7-11	102142	Nagpur	Ma
3	1999-04-03	1999-04-04	2	1999-04-01	4	3	4	Indira NAG...	7-8...	132132	Bengaluru	Kar

Results (Query 3):

Service_ID	Service_name	Service_cost	Reservation_number	EMPID	EMPNAME	DOB	AGE
1	Food	3000	1	1	Max	1999-03-21	22
2	Transport	8000	3	2	Jax	1959-04-05	62
3	Room	4000	2	3	Nax	1992-07-07	29

Query executed successfully. localhost (15.0 RTM) DESKTOP-NLAHG8Q\RANDHE... HOTEL 00:00:00 9 rows

Q8) Use all the above condition in JOIN as well.

QUERY:

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'HOTEL' database selected. The right pane shows a SQL query in 'SQLQuery2.sql' with the following code:

```
use HOTEL;
SELECT COUNT(*), ROOM_LOCATION FROM T2_Rooms
JOIN T2_Reservation
ON T2_Rooms.Customer_ID = T2_Reservation.Customer_ID
JOIN T2_Customer
ON T2_Rooms.Customer_ID = T2_Customer.Customer_ID
GROUP BY ROOM_LOCATION
HAVING ROOM_LOCATION LIKE 'BLOCK%'
ORDER BY ROOM_LOCATION DESC;
```

Below the query editor, the 'Results' tab shows the output of the query. The results are displayed in a table with two columns: '(No column name)' and 'ROOM_LOCATION'. The table contains two rows of data:

	(No column name)	ROOM_LOCATION
1	2	block-2
2	1	block-1

The status bar at the bottom indicates that the query was executed successfully, returning 2 rows.