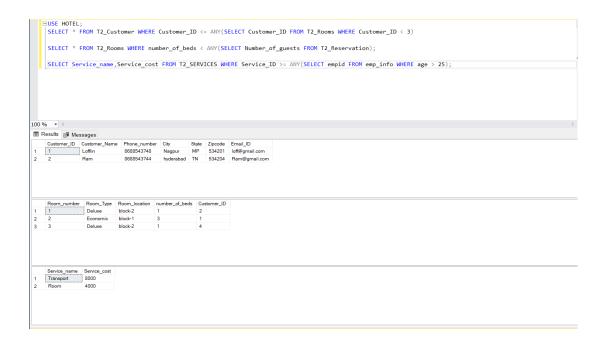
### **DBMS LAB ASSIGNMENT - 5**

**NAME: Bipul Gautam** 

**REG NO: 19BCS023** 

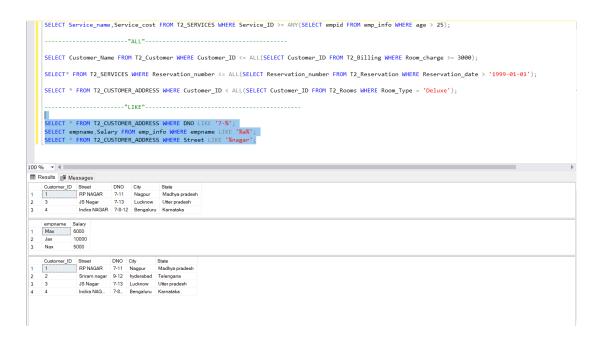
Q1) Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL

## **QUERIES FOR "ANY"**



### **QUERIES FOR "AII"**

### **QUERIES FOR "Like"**



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

```
SELECT AVG(age) AS average_age FROM emp_info;

SELECT MAX(number_of_beds) AS Max_numofbeds 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_charges FROM T2_Billing;

100 % 4

Max_numofbeds
1 3

Max_numofbeds
1 3

(No column name)
1 3

total_charges
1 10000
```

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

#### **ORDER BY**

```
SELECT * FROM T2_Customer ORDER BY Customer Name ASC;
SELECT * FROM emp_info ORDER BY age DESC

SELECT number_of_beds, COUNT(*) AS number_of_rooms FROM T2_Rooms GROUP BY number_of_beds;
SELECT CUBUT(Room_number), Room_Type FROM T2_Rooms GROUP BY ROOM_Type HAVING COUNT(Room_number) >= 1;
SELECT COUNT(Room_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_number) >= 1;

SELECT COUNT(Reservation_number), LEFT(Reservation_date, 4) FROM T2_Reservation GROUP BY LEFT(Reservation_date, 4) HAVING COUNT(Reservation_date, 4) HAVING COUNT(Reservation_date, 4) HAVING COUNT(Reservation_date, 4
```

#### **GROUP BY**

### **HAVING CLAUSE**

Q4) Use Aggregate function with group by and having

## AVG():

```
ELECT AVX(number of_beds) FROM T2_Rooms GROUP BY Room_location HAVING Room_location LINE 'block%';

SELECT COUNT(Customer_ID) FROM T2_Reservation GROUP BY Check_in_date MAVING Check_in_date >= '1992-02-03';

SELECT MIN(Salary) FROM emp_info GROUP BY age HAVING age > 25;

SELECT MAX(Room_charge) FROM T2_Billing GROUP BY LEFT(Payment_date,7) HAVING LEFT(Payment_date,7) LINE '2021-%';

SELECT SUN(Service_cost) FROM T2_SERVICES GROUP BY Service_cost HAVING Service_cost BETWEEN 4000 AND 6000;

TO SELECT SUN(Service_cost) FROM T2_SERVICES GROUP BY Service_cost HAVING Service_cost BETWEEN 4000 AND 6000;
```

## COUNT():

```
EUSE HOTEL;

SELECT AVG(number_of_beds) FROM T2_Rooms GROUP BY Room_location HAVING Room_location LIKE 'block%';

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

SELECT MIN(Salary) FROM emp_info GROUP BY age HAVING age > 25;

SELECT MAX(Room_charge) FROM T2_Billing GROUP BY LEFT(Payment_date,7) HAVING LEFT(Payment_date,7) LIKE '2021-%';

SELECT SUM(Service_cost) FROM T2_SERVICES GROUP BY Service_cost HAVING Service_cost BETWEEN 4000 AND 6000;

100 % - *

BROWNERS AND SERVICES GROUP BY SERVICES GROUP BY SERVICE COST BETWEEN 4000 AND 6000;
```

# MIN():

# MAX():

# SUM():

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

### **QUERY:**

```
SELECT Customer_Name, COUNT(*) FROM T2_customer

WHERE Customer_ID = AINY(

SELECT Customer_ID from T2_Reservation

WHERE Reservation_number = AINY(

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 Results of Messages

Customer_Mame (No column name)

1 Probles

1
```

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

# EXCEPT():

## EXISTS():

```
-----EXCEPT-----
   SELECT Customer_ID FROM T2_Customer
    EXCEPT
    SELECT Customer_ID FROM T2_Reservation;
    SELECT Customer_ID FROM T2_Rooms
    WHERE EXISTS
(SELECT Customer_ID FROM T2_Billing)
    ORDER BY Customer_ID ASC;
    SELECT * FROM T2_Customer
    WHERE NOT EXISTS
    (SELECT Customer_ID FROM T2_Reservation);
     ------UNION---
   SELECT City FROM T2_CUSTOMER_ADDRESS
    SELECT City FROM T2_Customer;
     -----INTERSECTION--
   SELECT Room_charge FROM T2_Billing
   INTERSECT

SELECT Service_cost FROM T2_SERVICES;
100 % ▼ 4
Customer_ID
1 1
2 2
3 4
```

## NOT EXISTS():

```
-----EXCEPT-----
  □SELECT Customer_ID FROM T2_Customer
EXCEPT
    SELECT Customer_ID FROM T2_Reservation;
    -----EXISTS-----
   SELECT Customer_ID FROM T2_Rooms
    WHERE EXISTS
    (SELECT Customer_ID FROM T2_Billing)
    SELECT City FROM T2_CUSTOMER_ADDRESS
    UNION
    SELECT City FROM T2_Customer;
     -----INTERSECTION--
   SELECT Room_charge FROM T2_Billing
   INTERSECT
  SELECT Service_cost FROM T2_SERVICES;
100 % 🔻 🔻
   Customer_ID Customer_Name Phone_number City State Zipcode Email_ID
```

# UNION():

```
-----EXCEPT-----
  ⊟SELECT Customer_ID FROM T2_Customer
    SELECT Customer_ID FROM T2_Reservation;
    -----EXISTS----
   SELECT Customer_ID FROM T2_Rooms
    WHERE EXISTS
(SELECT Customer_ID FROM T2_Billing)
   ------NOT EXISTS------
    ------UNION-------SELECT City FROM T2_CUSTOMER_ADDRESS
    (SELECT Customer_ID FROM T2_Reservation);
    SELECT City FROM T2_Customer;
      -----INTERSECTION--
   SELECT Room_charge FROM T2_Billing
   SELECT Service_cost FROM T2_SERVICES;
.00 % 🔻 🔻
1 Bengaluru
2 hyderabad
3 Lucknow
4 Nagpur
```

### **INTERSECT:**

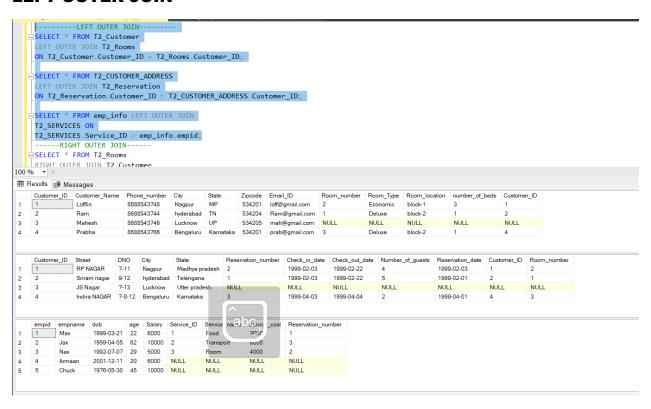
```
-----EXCEPT-----
   SELECT Customer_ID FROM T2_Customer
    EXCEPT
    SELECT Customer_ID FROM T2_Reservation;
   SELECT Customer_ID FROM T2_Rooms
    WHERE EXISTS
(GELECT Customer_ID FROM T2_Billing)
ORDER BY Customer_ID ASC;
-----NOT EXISTS------
    SELECT * FROM T2_Customer
    WHERE NOT EXISTS
    (SELECT Customer_ID FROM T2_Reservation);
      -----UNION--
   SELECT City FROM T2_CUSTOMER_ADDRESS
     SELECT City FROM T2_Customer;
     SELECT Room_charge FROM T2_Billing
     SELECT Service_cost FROM T2_SERVICES;
100 % 🔻 🔻
```

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

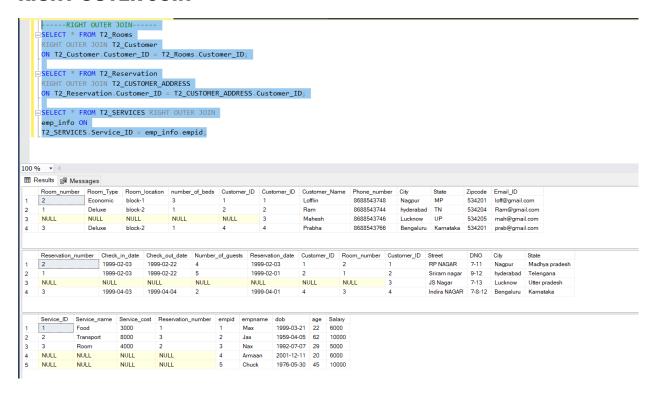
#### **INNER JOIN**

```
-----INNER JOIN-----
   ☐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
    INNER JOIN T2 Reservation
    ON T2_Customer.Customer_ID = T2_Reservation.Customer_ID;
   SELECT Reservation_number,Reservation_date,Room_Type,Room_location FROM T2_Rooms INNER JOIN T2_Reservation
  ON T2_Rooms.Room_number = T2_Reservation.Room_number;
100 % 🔻 🔻
Results Messages
    Customer_Name DNO
Lofflin 7-11
                        Street
RP NAGAR
                                    Nagpur
     Ram
                  9-12 Sriram nagar hyderabad
                  7-13 JS Nagar Lucknow
7-8-12 Indira NAGAR Bengaluru
     Prabha
     Customer_Name Number_of_guests Check_in_date Check_out_date
    Ram
                                  1999-02-03
                                              1999-02-22
                                 1999-02-03
                                              1999-02-22
     Reservation_number
                  er Reservation_date
1999-02-01
                                  Deluxe
                                             block-2
                                   Economic
                     1999-02-03
                                            block-1
```

### **LEFT OUTER JOIN**



#### **RIGHT OUTER JOIN**



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

#### **QUERY:**

