DBMS LAB ASSIGNMENT - 5

NAME: K.Sai Ganesh

REG NO: 19BCS055

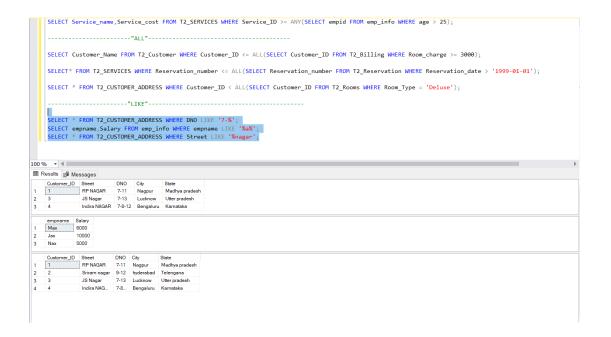
Q1) Illustrate logical ANY, ALL and LIKE operator- the queries should be relevant to your respective databases 3 queries for each operator. One query explaining the difference between ANY and ALL

QUERIES FOR "ANY"

BUSE 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 Reservation);					
	SELECT * FROM IZ_Kooms WHERE number_ot_beds < ANY(Select Number_ot_guests FROM IZ_Keservation);					
	SELECT Service name, Service cost FROM T2_SERVICES_WHERE Service_ID >= ANY(SELECT_empid_FROM_emp_info_WHERE age > 25);					
	Direct Service_Indus, Service_Cost From 12_Service_S where Service_ED /- Arr(Service empire From emp_1110 where age / 25))					
ľ						
100 % 💌						
⊞ Results						
	Customer_ID	Customer_Name	e Phone_numbe	r City :	State Zipcod	e Email_ID
1	1	Lofflin	8688543748	Nagpur	MP 53420	loff@gmail.com
2	2	Ram	8688543744	hyderabad	TN 53420	Ram@gmail.com
<u> </u>	1-					
	Room_number					
1	1	Deluxe	block-2	1	2	
2	2	Economic	block-1	1	1 4	
3	3	Deluxe	block-2	1	4	
1						
1						
	Service_name	Service_cost				
1	Transport	8000				
2	Room	4000				

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

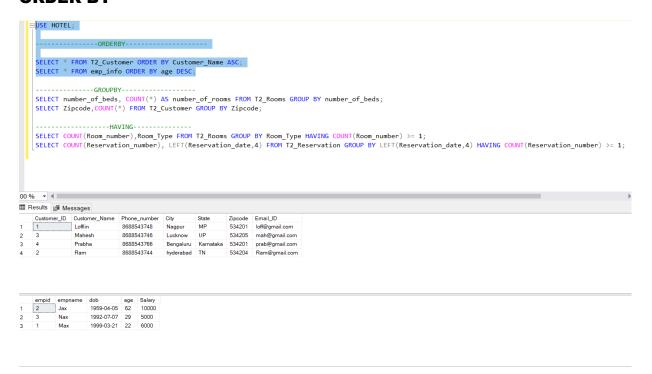
MAX() – returns the maximum value in set

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

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

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

ORDER BY



GROUP BY

HAVING CLAUSE

Q4) Use Aggregate function with group by and having

AVG():

```
EUECT AVG(number_of_beds) FROM T2_Rosms 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-% - (Nocoumn name) 1 3
2 1
```

COUNT():

MIN():

MAX():

```
BUSE 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;
```

SUM():

```
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 % 

Results Messages

(No column name)

1 4000
```

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

QUERY:

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

EXCEPT():

EXISTS():

```
-----FXCEPT-----
   SELECT Customer_ID FROM T2_Customer
    SELECT Customer_ID FROM T2_Reservation;
          ---EXISTS
   WHERE NOT EXISTS
    (SELECT Customer_ID FROM T2_Reservation);
      -----UNION--
   SELECT City FROM T2_CUSTOMER_ADDRESS
   UNION
SELECT City FROM T2 Customer;
    ----INTERSECTION
   SELECT Room_charge FROM T2_Billing
   SELECT Service_cost FROM T2_SERVICES;
100 % -
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)
     ORDER BY Customer_ID FROM IZ_BIIIIng
ORDER BY Customer_ID ASC;
-----NOT EXISTS-----
SELECT * FROM T2_Customer
WHERE NOT EXISTS
     (SELECT Customer_ID FROM T2_Reservation);
    SELECT City FROM T2_CUSTOMER_ADDRESS
     UNION
     SELECT City FROM T2_Customer;
   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():

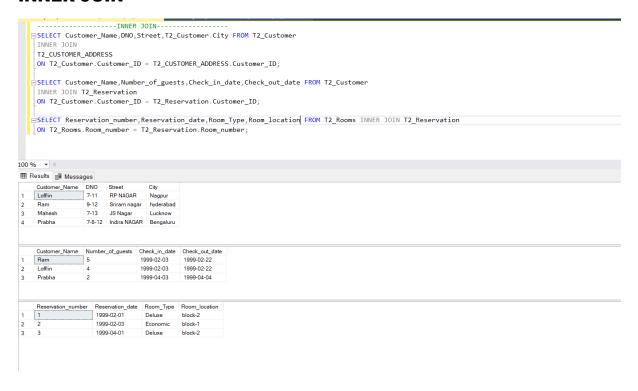
```
-----EXCEPT-----
  □SELECT Customer_ID FROM T2_Customer
    EXCEPT
    SELECT Customer_ID FROM T2_Reservation;
   SELECT Customer_ID FROM T2_Rooms
   WHERE EXISTS
(SELECT Customer_ID FROM T2_Billing)
    ORDER BY Customer_ID ASC;
   SELECT * FROM T2_Customer
    WHERE NOT EXISTS
    (SELECT Customer_ID FROM T2_Reservation);
    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;
.00 % 🔻 🔻
City
1 Bengaluru
   hyderabad
2 hyderaba
3 Lucknow
4 Nagpur
```

INTERSECT:

```
-----EXCEPT-----
    □SELECT Customer_ID FROM T2_Customer
     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;
    -----NOT EXISTS-----
SELECT * FROM T2_Customer
     WHERE NOT EXISTS
     (SELECT Customer_ID FROM T2_Reservation);
    SELECT City FROM T2_CUSTOMER_ADDRESS
     SELECT City FROM T2_Customer;
     SELECT CITY FROM 12_cdscomer,
------INTERSECTION------
SELECT Room_charge FROM 12_Billing
    SELECT Service_cost FROM T2_SERVICES;
100 % 🔻 🔻
Room_charge
1 3000
```

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

INNER JOIN



LEFT OUTER JOIN

```
----LEFT OUTER JOIN-----
    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 emp_info LEFT OUTER JOIN
    T2 SERVICES ON
    T2_SERVICES.Service_ID = emp_info.empid;
   -----RIGHT OUTER JOIN-----
RIGHT OUTER JOTN T2 Customer
Customer_ID Customer_Name Phone_number
                                                      Zipcode Email_ID
                                                                          Room_number Room_Type Room_location number_of_beds Customer_ID
                                                            loff@gmail.com 2
                                     Nagpur
          Lofflin
                          8688543748
                                              MP
                                                      534201
                                                                                     Economic
                                                                                              block-1
                          8688543744
                                      hyderabad TN
                                                      534204
                                                             Ram@gmail.com 1
                                                                                                          NULL NULL
                                      Lucknow UP
                                                      534205 mah@gmail.com NULL
              Mahesh
                          8688543746
                                                                                    NULL
                                                                                              NULL
                          8688543766
                                                      534201
                                                             prab@gmail.com 3
                                                  Customer_ID Street
                        DNO
                              City
                                      State
           RP NAGAR
                             Nagpur Madhya pra
hyderabad Telengana
                        7-11
                                      Madhya pradesh 2
                                                                 1999-02-03
                                                                            1999-02-22
                                                                                                      1999-02-03
                                                                          1999-02-22
              Sriram nagar 9-12
                                                                  1999-02-03
                                                                                                      1999-02-01
              JS Nagar
                        7-13
                              Lucknow
                                      Utter pradesh
                                                  NULL
                                                                 NULL
                                                                            NULL
                                                                                        NULL
                                                                                                     NULL
                                                                                                                  NULL NULL
              Indira NAGAR 7-8-12 Bengaluru Karnataka
                                                                          1999-04-04
   empid empname dob
                                    Service_ID Service nais Servic _cost
1 Food 3000
                                                                 Reservation_number
                  1999-03-21 22
                               6000
                  1959-04-05 62
                               10000 2
   2
          Jax
                                              Transport
                                                        4000
                  1992-07-07 29
                               5000 3
                                    NULL
                                                                  NULL
          Armaan
                  2001-12-11 20 6000
                                             NULL
                  1976-05-30 45 10000 NULL
```

RIGHT OUTER JOIN

