

# COURSE TITLE: DATABASE SYSTEMS LABCOURSE CODE: BCSE302P

# EXPERIMENT NO: 6 SQL JOINS

NAME: PRIYANSHU SONI REGISTER NO: 21BRS1629

**SLOT**: L21+L22

**CLASS NUMBER:** CH2022232501096

**DATE:** 15-APRIL, 2023

FACULTY NAME: L. JANI ANBARASI

Create three tables: salesman, customer and orders.

• SALESMAN( )

#### CODE:

CREATE TABLE SALESMAN\_21BRS1629(SALESMAN\_ID INT PRIMARY KEY, NAME VARCHAR(15), CITY VARCHAR(15), COMMISSION INT);

mysql> SELECT * FROM SALESMAN_21BRS1629;							
SALESMAN_ID	NAME	CITY	COMMISSION				
1	Α	DELHI	5				
2	В	CHENNAI	10				
3	С	CHANDIGARH	15				
4	D	DELHI	20				
5	Ε	DELHI	25				
6	F	AHEMDABAD	30				
7	G	PUNE	35				
8	Н	BENGALURU	69				
9	I	NASIK	45				
10	J	INDORE	55				
++							
10 rows in set (0.00 sec)							

• CUSTOMER(CUSTOMER\_ID, CUST\_NAME, CITY, GRADE, SALESMAN\_ID) **CODE**:

CREATE TABLE CUSTOMER\_21BRS1629(CUSTOMER\_ID INT PRIMARY KEY, CUST\_NAME VARCHAR(15), CITY VARCHAR(15), GRADE INT, SALESMAN\_ID INT,FOREIGN KEY (SALESMAN\_ID) REFERENCES SALESMAN\_21BRS1629(SALESMAN\_ID)):

(SALESMAN_ID) REFERENCES SALESMAN_21BRS1629(SALESMAN_ID));						
mysql> SELECT * FROM CUSTOMER_21BRS1629;						
CUSTOMER_ID	CUST_NAME		GRADE	SALESMAN_ID		
11	Rahul	Delhi	100	1		
12	Priya	Mumbai	900	1		
13	Amit	Kolkata	800	2		
14	Anjali	Chennai	700	NULL		
15	Rohit	Bengaluru	500	3		
16	Sneha	Hyderabad	600	3		
17	Vikram	Pune	550	4		
18	Neha	Ahmedabad	650	NULL		
19	Rajesh	Jaipur	400	5		
20	Smita	Lucknow	510	6		
21	Alok	Patna	110	NULL		
22	Shreya	Surat	200	NULL		
23	Vivek	Kanpur	300	7		
24	Sakshi	Nagpur	450	8		
25	Ravi	Indore	560	9		
26	Anita	Thane	150	9		
27	Aryan	Bhopal	270	10		
28	Sarika	Visakhapatnam	370	10		
18 rows in set (0.00 sec)						

• ORDERS(ORDER\_NO, ORDER\_AMT, ORDER\_DATE, CUST\_ID, SALESMAN\_ID) **CODE**:

CREATE TABLE ORDER\_21BRS1629(ORDER\_NO INT PRIMARY KEY, ORDER\_AMT DECIMAL(10,2), ORDER\_DATE DATE, CUST\_ID INT, SALESMAN\_ID INT, FOREIGN KEY (SALESMAN\_ID) REFERENCES SALESMAN\_21BRS1629 (SALESMAN\_ID) ,FOREIGN KEY(CUST\_ID) REFERENCES CUSTOMER 21BRS1629(CUSTOMER ID) );

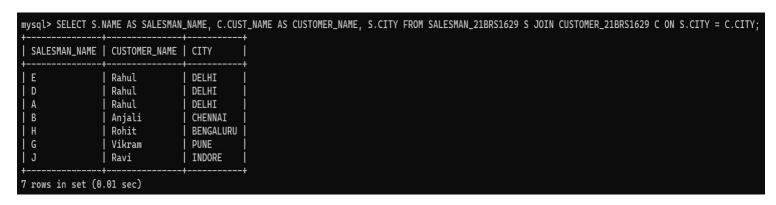
mysql> SELECT * FROM ORDER_21BRS1629;							
ORDER_NO	ORDER_AMT	ORDER_DATE	CUST_ID	SALESMAN_ID			
111	2500.00	2010-06-23	11	1			
112	1500.00	2010-06-23	11	1			
113	1600.00	2010-06-23	12	1			
114	1500.00	2010-06-23	12	1			
115	2500.00	2010-06-23	13	2			
116	1800.00	2010-06-23	13	2			
117	1900.00	2010-06-23	14	NULL			
118	2000.00	2010-06-23	14	NULL			
119	1100.00	2010-06-23	15	3			
120	1200.00	2010-06-23	15	3			
121	1300.00	2011-06-23	16	3			
122	1400.00	2011-06-23	16	3			
123	5000.00	2011-06-23	17	4			
124	5000.00	2011-06-23	17	NULL			
125	1500.00	2011-06-23	18	5			
126	1500.00	2011-06-23	19	NULL			
127	5000.00	2011-06-23	20	NULL			
128	500.00	2011-06-23	21	7			
129	50.00	2011-06-23	22	8			
130	550.00	2011-06-23	24	9			
131	600.00	2011-06-23	25	9			
132	990.00	2011-06-23	26	10			
133	900.00	2011-06-23	27	10			
+	t	<b></b>		+			
23 rows in set (0.00 sec)							

1. Prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

#### CODE:

SELECT S.NAME AS SALESMAN\_NAME, C.CUST\_NAME AS CUSTOMER\_NAME, S.CITY FROM SALESMAN\_21BRS1629 S JOIN CUSTOMER\_21BRS1629 C ON S.CITY = C.CITY;

# **OUTPUT**:



2. Make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.

#### CODE:

SELECT O.ORDER\_NO, O.ORDER\_AMT, C.CUST\_NAME, C.CITY FROM ORDER\_21BRS1629 O JOIN CUSTOMER\_21BRS1629 C ON O.CUST\_ID = C.CUSTOMER\_ID WHERE O.ORDER\_AMT BETWEEN 500 AND 2000;



3. Know which salesman is working for which customer.

#### CODE:

SELECT S.NAME, C.CUST\_NAME FROM SALESMAN\_21BRS1629 S JOIN CUSTOMER\_21BRS1629 C ON S.SALESMAN\_ID = C.SALESMAN\_ID;

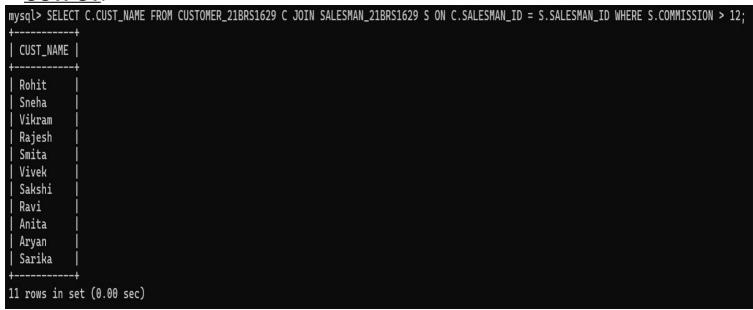
#### **OUTPUT:**

```
mysql> SELECT S.NAME, C.CUST_NAME FROM SALESMAN_21BRS1629 S JOIN CUSTOMER_21BRS1629 C ON S.SALESMAN_ID = C.SALESMAN_ID;
| NAME | CUST_NAME |
         Rahul
         Priya
 В
         Amit
        Rohit
         Sneha
 D
         Vikram
 E
F
G
         Rajesh
         Smita
         Vivek
 H
         Sakshi
         Ravi
         Anita
 J
         Aryan
 J
         Sarika
14 rows in set (0.00 sec)
```

4. Find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.

# CODE:

SELECT C.CUST\_NAME FROM CUSTOMER\_21BRS1629 C JOIN SALESMAN 21BRS1629 S ON C.SALESMAN ID = S.SALESMAN ID WHERE S.COMMISSION > 12;

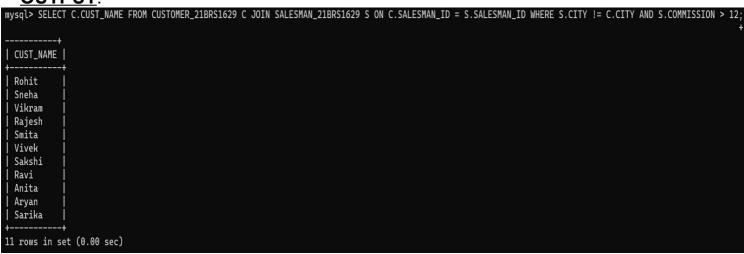


5. Find the list of customers who appointed a salesman for their jobs who does not live in the same city where their customer lives, and gets a commission is above 12%.

#### CODE:

SELECT C.CUST\_NAME FROM CUSTOMER\_21BRS1629 C JOIN SALESMAN\_21BRS1629 S ON C.SALESMAN\_ID = S.SALESMAN\_ID WHERE S.CITY != C.CITY AND S.COMMISSION > 12;

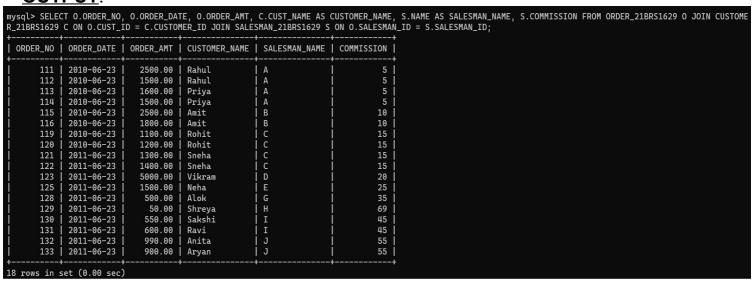
#### **OUTPUT:**



6. Find the details of an order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.

# CODE:

SELECT O.ORDER\_NO, O.ORDER\_DATE, O.ORDER\_AMT, C.CUST\_NAME AS CUSTOMER\_NAME, S.NAME AS SALESMAN\_NAME, S.COMMISSION FROM ORDER\_21BRS1629 O JOIN CUSTOMER\_21BRS1629 C ON O.CUST\_ID = C.CUSTOMER\_ID JOIN SALESMAN\_21BRS1629 S ON O.SALESMAN ID = S.SALESMAN ID;



7. Make a join on the tables: salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come.

#### CODE:

SELECT S.SALESMAN\_ID, S.NAME AS SALESMAN\_NAME, S.CITY AS SALESMAN\_CITY, S.COMMISSION, C.CUSTOMER\_ID, C.CUST\_NAME, C.CITY AS CUSTOMER\_CITY, C.GRADE, O.ORDER\_NO, O.ORDER\_AMT, O.ORDER\_DATE FROM SALESMAN\_21BRS1629 S JOIN CUSTOMER\_21BRS1629 C ON S.SALESMAN\_ID = C.SALESMAN\_ID JOIN ORDER\_21BRS1629 O ON C.CUSTOMER\_ID = O.CUST\_ID;

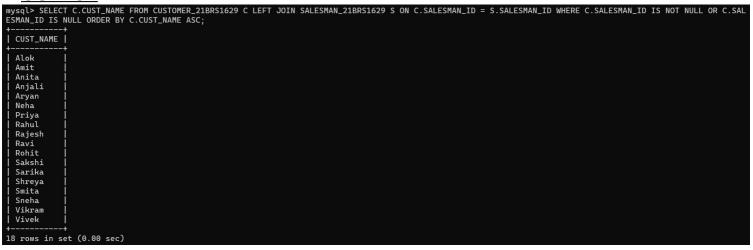
# **OUTPUT:**

	<del>!</del>	<u> </u>				<del>!</del>	<del> </del>	! <u>!</u>		<del>!</del>
SMAN_ID	SALESMAN_NAME	SALESMAN_CITY	COMMISSION	CUSTOMER_ID	CUST_NAME	CUSTOMER_CITY	GRADE	ORDER_NO	ORDER_AMT	ORDER_DATE
1	А	DELHI	5	11	Rahul	Delhi	100	111	2500.00	2010-06-23
1	A	DELHI	5	11	Rahul	Delhi	100	112	1500.00	2010-06-23
1	A	DELHI	5	12	Priya	Mumbai	900	113	1600.00	2010-06-23
1	A	DELHI	5	12	Priya	Mumbai	900	114	1500.00	2010-06-23
2	В	CHENNAI	10	13	Amit	Kolkata	800	115	2500.00	2010-06-23
2	B	CHENNAI	10	13	Amit	Kolkata	800	116	1800.00	2010-06-23
3	l C	CHANDIGARH	15	15	Rohit	Bengaluru	500	119	1100.00	2010-06-23
3	C	CHANDIGARH	15	15	Rohit	Bengaluru	500	120	1200.00	2010-06-23
3	C	CHANDIGARH	15	16	Sneha	Hyderabad	600	121	1300.00	2011-06-23
3	C	CHANDIGARH	15	16	Sneha	Hyderabad	600	122	1400.00	2011-06-23
4	D	DELHI	20	17	Vikram	Pune	550	123	5000.00	2011-06-23
4	D	DELHI	20	17	Vikram	Pune	550	124	5000.00	2011-06-23
5	E	DELHI	25	19	Rajesh	Jaipur	400	126	1500.00	2011-06-23
6	F	AHEMDABAD	30	20	Smita	Lucknow	510	127	5000.00	2011-06-23
8	H	BENGALURU	69	24	Sakshi	Nagpur	450	130	550.00	2011-06-23
9	I	NASIK	45	25	Ravi	Indore	560	131	600.00	2011-06-23
9	I	NASIK	45	26	Anita	Thane	150	132	990.00	2011-06-23
10	J	INDORE	55	27	Aryan	Bhopal	270	133	900.00	2011-06-23

8. Make a list in ascending order for the customer who works either through a salesman or by own.

# CODE:

SELECT C.CUST\_NAME FROM CUSTOMER\_21BRS1629 C LEFT JOIN SALESMAN\_21BRS1629 S ON C.SALESMAN\_ID = S.SALESMAN\_ID WHERE C.SALESMAN\_ID IS NOT NULL OR C.SALESMAN\_ID IS NULL ORDER BY C.CUST NAME ASC;



9. Make a list in ascending order for the customer who holds a grade less than 300 and works either through a salesman or by own.

# CODE:

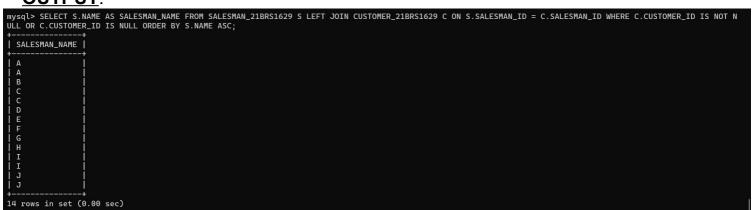
SELECT C.CUST\_NAME FROM CUSTOMER\_21BRS1629 C LEFT JOIN SALESMAN\_21BRS1629 S ON C.SALESMAN\_ID = S.SALESMAN\_ID WHERE C.GRADE<300 AND (C.SALESMAN\_ID IS NOT NULL OR C.SALESMAN\_ID IS NULL) ORDER BY C.CUST\_NAME ASC;

#### **OUTPUT:**

10. Make a list in ascending order for the salesmen who work either for one or more customer or not yet join under any of the customers.

# CODE:

SELECT S.NAME AS SALESMAN\_NAME FROM SALESMAN\_21BRS1629 S LEFT JOIN CUSTOMER\_21BRS1629 C ON S.SALESMAN\_ID = C.SALESMAN\_ID WHERE C.CUSTOMER\_ID IS NOT NULL OR C.CUSTOMER ID IS NULL ORDER BY S.NAME ASC

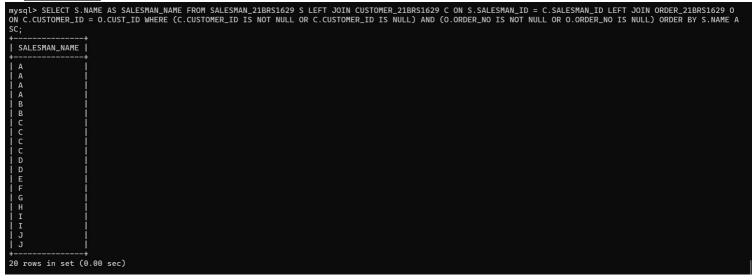


11. Make a list for the salesmen who work either for one or more customer or not yet join under any of the customers who placed either one or more orders or no order to their supplier.

#### CODE:

SELECT S.NAME AS SALESMAN\_NAME FROM SALESMAN\_21BRS1629 S LEFT JOIN CUSTOMER\_21BRS1629 C ON S.SALESMAN\_ID = C.SALESMAN\_ID LEFT JOIN ORDER\_21BRS1629 O ON C.CUSTOMER\_ID = O.CUST\_ID WHERE (C.CUSTOMER\_ID IS NOT NULL OR C.CUSTOMER\_ID IS NULL) AND (O.ORDER NO IS NOT NULL OR O.ORDER NO IS NULL) ORDER BY S.NAME ASC;

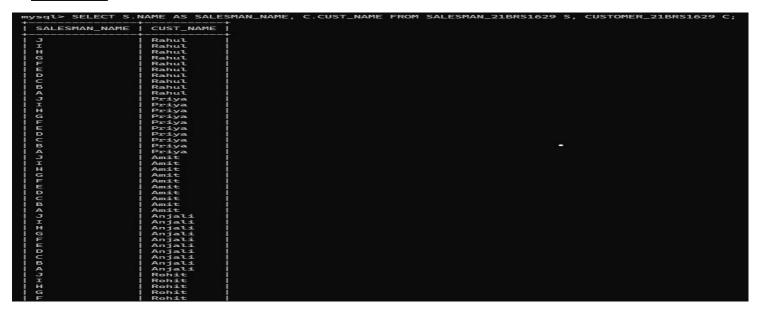
#### **OUTPUT**:



12. Make a Cartesian product between salesman and customer i.e. each salesman will appear for all customers and vice versa.

# CODE:

SELECT S.NAME AS SALESMAN\_NAME, C.CUST\_NAME FROM SALESMAN\_21BRS1629 S, CUSTOMER 21BRS1629 C;



1.	! Allth	(	Vivek
E	Amit	B	Vivek
D	Amit	A	Vivek
C	Amit	J	Sakshi
B	Amit	I	Sakshi
A	Amit	H	Sakshi
J	Anjali	G	Sakshi
I	Anjali	F	Sakshi
H	Anjali	E	Sakshi
j G	Anjali	D	Sakshi
İF	Anjali	C	Sakshi
İΕ	Anjali	B	Sakshi
j D	Anjali	A	Sakshi
i c	Anjali	J	Ravi
В	Anjali	ļ I	Ravi
İÃ	Anjali	H	Ravi
j	Rohit	G	Ravi
Ī	Rohit	<u>F</u>	Ravi
İĤ	Rohit	E	Ravi
Ğ	Rohit	D	Ravi
İF	Rohit	l c	Ravi
Ė	Rohit	B	Ravi
D	Rohit	A	Ravi
C	Rohit	J	Anita
B		I	Anita
	Rohit	H	Anita
A	Rohit	G   F	Anita   Anita
j	Sneha	F	Anita     Anita
ļī	Sneha	D	Anita
H	Sneha	C	Anita
G	Sneha	B	Anita
<u> </u>	Sneha	A	Anita
Į E	Sneha	j	Aryan
D	Sneha	Ĭ	Aryan
ļ c	Sneha	i Ĥ	Aryan
B	Sneha	Ġ	Aryan
ļ A	Sneha	F	Aryan
J	Vikram	E	Aryan
ļ I	Vikram	j D	Aryan
H	Vikram	j c	Aryan
G	Vikram	B	Aryan
ļ F	Vikram	A	Aryan
E	Vikram	J	Sarika
D	Vikram	I	Sarika
C	Vikram	H	Sarika
B	Vikram	G	Sarika
A	Vikram	ļ F	Sarika
J	Neha	E	Sarika
I	Neha	D	Sarika
H	Neha	l c	Sarika
G	Neha	B	Sarika
F	Neha	A	Sarika
E	l Neha	<del> </del>	tt
j D	Neha	180 rows in set	(0.00 sec)
U	Neria .		