sSAMRITA VISHWA VIDYAPEETHAM –CHENNAI CAMPUS
AMRITA SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
22AIE303 - DATABASE MANAGEMENT SYSTEM

ACTIVITY SHEET -02- SQL QUERIES ON JOINS

Name: THARUN KAARTHIK G K
Roll no.: CH.EN.U4AIE22062
Date: 19/07/2024

Consider the following tables

Table: Salesman

Salesman_id	Name	City	Commisision
5001	James Hoog	Newyork	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

Table : Customer

Customer_id	Customer_name	City	Grade	Salesman_id
3002	Nick Rimando	Newyork	100	5001
3005	Graham Lusi	California	200	5002
3001	Brad Guran	London		
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	Newyork	200	5001
3009	Geoff Camero	Berlin	100	
3008	Julian Green	London	300	5002
3003	Jory Altidor	Moscow	200	5007

Table:Orders

Ord_no	Purch_amt	Ord_date	Customer_id	Salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Table: company_mast

Com_id	Com_Name
11	Samsung
12	iBall
13	Epsion
14	Zebronics
15	Asus
16	Frontech

Table: item_mast

Pro_ID	Pro_Name	Pro_Price	Pro_Com
101	Mother Board	3200	15
102	Keyboard	450	16
103	ZIP Drive	250	14
104	Speaker	550	16
105	Monitor	5000	11
106	DVD Drive	900	12
107	CD Drive	800	12
108	Printer	2600	13
109	Refill Catridge	350	13
110	Mouse	250	12

Table: emp_department

DPT_CODE	DPT_NAME	DPT_ALLOTMENT
57	IT	65000
63	Finance	15000
47	HR	240000
27	RD	55000
89	QC	75000

Table: emp_details

EMP_IDNO	EMP_FNAME	EMP_LNAME	EMP_DEPT
127323	Michale	Robbin	57
526689	Carlos	Snares	63
843795	Enric	Dosio	57
328717	Jhon	Snares	63
444527	Joseph	Dosni	47
659831	Zanifer	Emily	47
847674	Kuleswar	Sitaraman	57
748681	Henrey	Gabriel	47
555935	Alex	Manuel	57
539569	George	Mardy	27
733843	Mario	Saule	63
631548	Alan	Snappy	27
839139	Maria	Foster	57

Write the below SQL Queries

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

Query:

```
select s.Salesman_name, c.Customer_name,s.city
from Salesman s
join Customer c
on s.Salesman_id = c.Salesman_id
where s.city = c.city;
```

Output:

Salesman_name	Customer_name	city
James Hoog	Nick Rimando	Newyork
Mc Lyon	Fabian Johns	Paris
James Hoog	Brad Davis	Newyork

2.Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000

Query:

```
select s.Order_no,s.Purch_amt,c.Customer_name,c.city
from Salesman_order s
join Customer c
on s.Customer_id = c.Customer_id
where s.Purch_amt < 2000 and s.Purch_amt >500;
```

Output:

Order_no	Purch_amt	Customer_name	city
70007	948.5	Graham Lusi	California
70010	1983.43	Fabian Johns	Paris

3.Write a SQL statement to know which salesman are working for which customer

Query:

```
select s.Salesman_name,c.Customer_name
from Salesman s
join Customer c
on s.Salesman_id = c.Salesman_id
where s.Salesman_id = c.Salesman_id;
```

Output:

Salesman_name	Customer_name
James Hoog	Nick Rimando
Paul Adam	Jory Altidor
Mc Lyon	Fabian Johns
Nail Knite	Graham Lusi
James Hoog	Brad Davis
Nail Knite	Julian Green

4.Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.

Query:

```
select c.Customer_name
from Salesman s
join Customer c
on s.Salesman_id = c.Salesman_id
where s.commision > 0.12;
```

Output:

Customer_name
Nick Rimando
Jory Altidor
Fabian Johns
Graham Lusi
Brad Davis
Julian Green

5.Write a SQL statement to 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% .

Query:

```
select c.Customer_name
from Salesman s
join Customer c
on s.Salesman_id = c.Salesman_id
where s.commision > 0.12 and not s.city = c.city;
```

Output:

Customer_name
Jory Altidor
Graham Lusi
Julian Green

6.Write a SQL statement to 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.

Query:

```
SELECT o.Order_no, o.Order_date, o.Purch_amt, c.Customer_name, s.Salesman_Name, s.Commision
FROM Orders o
JOIN Customer c ON o.Customer_id = c.Customer_id
JOIN Salesman s ON o.Salesman id = s.Salesman id;
```

Output:

Order_no	Order_date	Purch_amt	Customer_name	Salesman_Name	Commision
70001	2012-10-05	150.5	Graham Lusi	Nail Knite	0.13
70002	2012-10-05	65.26	Nick Rimando	James Hoog	0.15
70003	2012-10-10	2480.3	Geoff Camero	Lauson Hen	0.12
70004	2012-08-17	110.5	Geoff Camero	Lauson Hen	0.12
70005	2012-07-27	2400.6	Brad Davis	James Hoog	0.15
70007	2012-09-10	948.5	Graham Lusi	Nail Knite	0.13
70008	2012-09-10	5760	Nick Rimando	James Hoog	0.15
70009	2012-09-10	270.65	Brad Guran	Pit Alex	0.11
70010	2012-10-10	1983.43	Fabian Johns	Mc Lyon	0.14
70011	2012-08-17	75.29	Jory Altidor	Paul Adam	0.13
70012	2012-06-27	250.45	Julian Green	Nail Knite	0.13
70013	2012-04-25	3045.6	Nick Rimando	James Hoog	0.15

7.Write a SQL statement to 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.

Query:

```
SELECT s.Salesman_id, s.Salesman_Name, s.city, s.Commision, c.Customer_id,
       c.Customer_name, c.grade, o.Order_no, o.Purch_amt, o.Order_date
FROM Orders o
JOIN Customer c ON o.Customer_id = c.Customer_id
JOIN Salesman s ON o.Salesman_id = s.Salesman_id;
```

Output:

Salesman_id	Salesman_Name	city	Commision	Customer_id	Customer_name	grade	Order_no	Purch_amt	Order_date
5002	Nail Knite	Paris	0.13	3005	Graham Lusi	200	70001	150.5	2012-10-05
5001	James Hoog	Newyork	0.15	3002	Nick Rimando	100	70002	65.26	2012-10-05
5003	Lauson Hen	NULL	0.12	3009	Geoff Camero	100	70003	2480.3	2012-10-10
5003	Lauson Hen	NULL	0.12	3009	Geoff Camero	100	70004	110.5	2012-08-17
5001	James Hoog	Newyork	0.15	3007	Brad Davis	200	70005	2400.6	2012-07-27
5002	Nail Knite	Paris	0.13	3005	Graham Lusi	200	70007	948.5	2012-09-10
5001	James Hoog	Newyork	0.15	3002	Nick Rimando	100	70008	5760	2012-09-10
5005	Pit Alex	London	0.11	3001	Brad Guran	NULL	70009	270.65	2012-09-10
5006	Mc Lyon	Paris	0.14	3004	Fabian Johns	300	70010	1983.43	2012-10-10
5007	Paul Adam	Rome	0.13	3003	Jory Altidor	200	70011	75.29	2012-08-17
5002	Nail Knite	Paris	0.13	3008	Julian Green	300	70012	250.45	2012-06-27
5001	James Hoog	Newyork	0.15	3002	Nick Rimando	100	70013	3045.6	2012-04-25

8.Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.

Query:

```
select Customer_name from Customer
order by Customer_name;
```

Output:

Customer_name
Brad Davis
Brad Guran
Fabian Johns
Geoff Camero
Graham Lusi
Jory Altidor
Julian Green
Nick Rimando

9.Write a SQL statement to 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.

Query:

```
select Customer_name,grade from Customer
where grade < 300 or grade is null
order by Customer_name;
```

Output:

Customer_name	grade
Brad Davis	200
Brad Guran	NULL
Geoff Camero	100
Graham Lusi	200
Jory Altidor	200
Nick Rimando	100

10.Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.

Query:

```
select c.Customer_name, c.city, o.Order_no, o.Order_date, o.Purch_amt
from Customer c
join Orders o
on c.Customer_id = o.Customer_id
order by Order_date;
```

Output:

Customer_name	city	Order_no	Order_date	Purch_amt
Nick Rimando	Newyork	70013	2012-04-25	3045.6
Julian Green	London	70012	2012-06-27	250.45
Brad Davis	Newyork	70005	2012-07-27	2400.6
Geoff Camero	Berlin	70004	2012-08-17	110.5
Jory Altidor	Moncow	70011	2012-08-17	75.29
Graham Lusi	California	70007	2012-09-10	948.5
Nick Rimando	Newyork	70008	2012-09-10	5760
Brad Guran	London	70009	2012-09-10	270.65
Graham Lusi	California	70001	2012-10-05	150.5
Nick Rimando	Newyork	70002	2012-10-05	65.26
Geoff Camero	Berlin	70003	2012-10-10	2480.3
Fabian Johns	Paris	70010	2012-10-10	1983.43

11.Write a SQL statement to make a report with customer name, city, order number, order date, order amount, salesman name and commission to find that either any of the existing customers have placed no order or placed one or more orders by their salesman or by own.

Query:

```
SELECT c.Customer_name, s.city, o.Order_no, o.Order_date, o.Purch_amt, s.Salesman_Name,  s.Commision
FROM Orders o
JOIN Customer c ON o.Customer_id = c.Customer_id
JOIN Salesman s ON o.Salesman_id = s.Salesman_id;
```

Output:

Customer_name	city	Order_no	Order_date	Purch_amt	Salesman_Name	Commision
Graham Lusi	Paris	70001	2012-10-05	150.5	Nail Knite	0.13
Nick Rimando	Newyork	70002	2012-10-05	65.26	James Hoog	0.15
Geoff Camero	NULL	70003	2012-10-10	2480.3	Lauson Hen	0.12
Geoff Camero	NULL	70004	2012-08-17	110.5	Lauson Hen	0.12
Brad Davis	Newyork	70005	2012-07-27	2400.6	James Hoog	0.15
Graham Lusi	Paris	70007	2012-09-10	948.5	Nail Knite	0.13
Nick Rimando	Newyork	70008	2012-09-10	5760	James Hoog	0.15
Brad Guran	London	70009	2012-09-10	270.65	Pit Alex	0.11
Fabian Johns	Paris	70010	2012-10-10	1983.43	Mc Lyon	0.14
Jory Altidor	Rome	70011	2012-08-17	75.29	Paul Adam	0.13
Julian Green	Paris	70012	2012-06-27	250.45	Nail Knite	0.13
Nick Rimando	Newyork	70013	2012-04-25	3045.6	James Hoog	0.15

12.Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customer or not yet join under any of the customers.

Query:

```
select Salesman_name from Salesman
order by Salesman_name;
```

Output:

Salesman_name
James Hoog
Lauson Hen
Mc Lyon
Nail Knite
Paul Adam
Pit Alex

13.Write a SQL statement to make a list for the salesmen who works 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.

Query:

```
Select distinct(s.Salesman_name) as Salesman
from Salesman s left join Customer c
on s.Salesman_id=c.Salesman_id
left join Orders o on c.Customer_id=o.Customer_id;
```

Output:

Salesman
James Hoog
Nail Knite
Lauson Hen
Pit Alex
Mc Lyon
Paul Adam

14.Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

Query:

```
Select distinct(s.Salesman_name) as Salesman
from Salesman s left join Customer c
on s.Salesman_id=c.Salesman_id
left join Orders o on c.Customer_id = o.Customer_id
where o.Purch_amt >= 2000 and Grade is not NULL;
```

Output:

Salesman
James Hoog

15.Write a SQL statement to make a report with customer name, city, order no, order date, purchase amount for those customers from the existing list who placed one or more orders or which order(s) have been placed by the customer who is not on the list.

Query:

```
Select c.Customer_name as Customer_Name, c.City, o.Order_no, o.Order_date, o.Purch_amt
from Customer c
right join Orders o
on c.Customer_id= o.Customer_id;
```

Output:

Customer_Name	City	Order_no	Order_date	Purch_amt
Graham Lusi	California	70001	2012-10-05	150.5
Nick Rimando	Newyork	70002	2012-10-05	65.26
Geoff Camero	Berlin	70003	2012-10-10	2480.3
Geoff Camero	Berlin	70004	2012-08-17	110.5
Brad Davis	Newyork	70005	2012-07-27	2400.6
Graham Lusi	California	70007	2012-09-10	948.5
Nick Rimando	Newyork	70008	2012-09-10	5760
Brad Guran	London	70009	2012-09-10	270.65
Fabian Johns	Paris	70010	2012-10-10	1983.43
Jory Altidor	Moncow	70011	2012-08-17	75.29
Julian Green	London	70012	2012-06-27	250.45
Nick Rimando	Newyork	70013	2012-04-25	3045.6

16. Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who is neither in the list nor have a grade.

Query:

```
Select c.Customer_name as Customer_Name, c.City, o.Order_no, o.Order_date, o.Purch_amt
from Customer c left join Orders o on c.Customer_id = o.Customer_id where c.Grade is not NULL
union
Select c.Customer_name as Customer_Name, c.City, o.Order_no, o.Order_date, o.Purch_amt
from Orders o right join Customer c on c.Customer_id = o.Customer_id where c.Grade is not NULL;
```

Output:

Customer_Name	City	Order_no	Order_date	Purch_amt
Nick Rimando	Newyork	70013	2012-04-25	3045.6
Nick Rimando	Newyork	70008	2012-09-10	5760
Nick Rimando	Newyork	70002	2012-10-05	65.26
Jory Altidor	Moncow	70011	2012-08-17	75.29
Fabian Johns	Paris	70010	2012-10-10	1983.43
Graham Lusi	California	70007	2012-09-10	948.5
Graham Lusi	California	70001	2012-10-05	150.5
Brad Davis	Newyork	70005	2012-07-27	2400.6
Julian Green	London	70012	2012-06-27	250.45
Geoff Camero	Berlin	70004	2012-08-17	110.5
Geoff Camero	Berlin	70003	2012-10-10	2480.3

17.Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa.

Query:

```
Select s.Salesman_name as Salesman, c.Customer_name as Customer_Name
from Salesman s cross join Customer c;
```

Output:

Salesman	Customer_Name
Paul Adam	Brad Guran
Mc Lyon	Brad Guran
Pit Alex	Brad Guran
Lauson Hen	Brad Guran
Nail Knite	Brad Guran
James Hoog	Brad Guran
Paul Adam	Nick Rimando
Mc Lyon	Nick Rimando
Pit Alex	Nick Rimando
Lauson Hen	Nick Rimando
Nail Knite	Nick Rimando
James Hoog	Nick Rimando
Paul Adam	Jory Altidor
Mc Lyon	Jory Altidor

18.Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa for that customer who belongs to a city.

Query:

```
Select s.Salesman_name as Salesman, c.Customer_name as Customer_Name
from Salesman s cross join Customer c
where s.City is not NULL;
```

Output:

Salesman	Customer_Name
Paul Adam	Brad Guran
Mc Lyon	Brad Guran
Pit Alex	Brad Guran
Nail Knite	Brad Guran
James Hoog	Brad Guran
Paul Adam	Nick Rimando
Mc Lyon	Nick Rimando
Pit Alex	Nick Rimando
Nail Knite	Nick Rimando
James Hoog	Nick Rimando
Paul Adam	Jory Altidor
Mc Lyon	Jory Altidor
Pit Alex	Jory Altidor
Nail Knite	Jory Altidor

19.Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa for those salesmen who belongs to a city and the customers who must have a grade.

Query:

```
Select s.Salesman_name as Salesman, c.Customer_name as Customer_Name
from Salesman s cross join Customer c
where s.City is not NULL and c.Grade is not NULL;
```

Output:

Salesman	Customer_Name
Paul Adam	Nick Rimando
Mc Lyon	Nick Rimando
Pit Alex	Nick Rimando
Nail Knite	Nick Rimando
James Hoog	Nick Rimando
Paul Adam	Jory Altidor
Mc Lyon	Jory Altidor
Pit Alex	Jory Altidor
Nail Knite	Jory Altidor
James Hoog	Jory Altidor
Paul Adam	Fabian Johns
Mc Lyon	Fabian Johns
Pit Alex	Fabian Johns

20.Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa for those salesmen who must belong a city which is not the same as his customer and the customers should have an own grade.

Query:

```
Select s.Salesman_name as Salesman, c.Customer_name as Customer_Name
from Salesman s cross join Customer c
where s.City is not NULL and s.City != c.City and c.Grade is not NULL;
```

Output:

Salesman	Customer_Name
Paul Adam	Nick Rimando
Mc Lyon	Nick Rimando
Pit Alex	Nick Rimando
Nail Knite	Nick Rimando
Paul Adam	Jory Altidor
Mc Lyon	Jory Altidor
Pit Alex	Jory Altidor
Nail Knite	Jory Altidor
James Hoog	Jory Altidor
Paul Adam	Fabian Johns
Pit Alex	Fabian Johns
James Hoog	Fabian Johns
Paul Adam	Graham Lusi

21.Write a SQL query to display all the data from the item_mast, including all the data for each item's producer company.

Query:

```
select i.Pro_id, i.Pro_name, i.Pro_Price, c.Com_id, c.Com_name from item_mast i
join company_mast c
on i.Pro_Com = c.Com_id;
```

Output:

Pro_id	Pro_name	Pro_Price	Com_id	Com_name
105	Monitor	5000	11	Samsung
110	Mouse	250	12	iBall
107	CD Drive	800	12	iBall
106	DVD Drive	900	12	iBall
109	Refill Catridge	350	13	Epsion
108	Printer	2600	13	Epsion
103	ZIP Drive	250	14	Zebronics
101	Mother Board	3200	15	Asus
104	Speaker	550	16	Frontech
102	Keyboard	450	16	Frontech

22.Write a SQL query to display the item name, price, and company name of all the products.

Query:

```
select i.Pro_name as Item_name, i.Pro_Price as Price, c.Com_name as Company_name
from item_mast i
join company_mast c
on i.Pro_Com = c.Com_id;
```

Output:

Item_name	Price	Company_name
Monitor	5000	Samsung
Mouse	250	iBall
CD Drive	800	iBall
DVD Drive	900	iBall
Refill Catridge	350	Epsion
Printer	2600	Epsion
ZIP Drive	250	Zebronics
Mother Board	3200	Asus
Speaker	550	Frontech
Keyboard	450	Frontech

23.Write a SQL query to display the average price of items of each company, showing the name of the company.

Query:

```
select avg(i.Pro_Price) as Average_Price, c.Com_name as Company_name
from item_mast i
join company_mast c
on i.Pro_Com = c.Com_id
group by c.Com_Name;
```

Output:

Average_Price	Company_name
3200.0000	Asus
500.0000	Frontech
250.0000	Zebronics
5000.0000	Samsung
650.0000	iBall
1475.0000	Epsion

24.Write a SQL query to display the names of the company whose products have an average price larger than or equal to Rs. 350.

Query:

```
select avg(i.Pro_Price) as Average_Price, c.Com_name as Company_name
from item_mast i
join company_mast c
on i.Pro_Com = c.Com_id
group by c.Com_Name
having Average_Price > 350 or Average_Price = 350;
```

Output:

Average_Price	Company_name
3200.0000	Asus
500.0000	Frontech
5000.0000	Samsung
650.0000	iBall
1475.0000	Epsion

25.Write a SQL query to display the name of each company along with the ID and price for their most expensive product.

Query:

```
select c.Com_name as Company_name, c.Com_id as Company_id, max(i.Pro_Price) as Expensive_Product
from item_mast i
join company_mast c
on i.Pro_Com = c.Com_id
group by c.Com_Name, c.Com_id;
```

Output:

Company_name	Company_id	Expensive_Product
Asus	15	3200
Frontech	16	550
Zebronics	14	250
Samsung	11	5000
iBall	12	900
Epsion	13	2600

26.Write a query in SQL to display all the data of employees including their department.

Query:

```
select e.*,d.DPT_NAME from emp_details e
join emp_department d
on e.EMP_DEPT = d.DPT_CODE;
```

Output:

EMP_IDNO	EMP_FNAME	EMP_LNAME	EMP_DEPT	DPT_NAME
127323	Michale	Robbin	57	IT
526689	Carlos	Snares	63	Finance
843795	Enric	Dosio	57	IT
328717	Jhon	Snares	63	Finance
444527	Joseph	Dosni	47	HR
659831	Zanifer	Emily	47	HR
847674	Kuleswar	Sitaraman	57	IT
748681	Henrey	Gabriel	47	HR
555935	Alex	Manuel	57	IT
539569	George	Mardy	27	RD
733843	Mario	Saule	63	Finance
631548	Alan	Snappy	27	RD
839139	Maria	Foster	57	IT

27.Write a query in SQL to display the first name and last name of each employee, along with the name and sanction amount for their department.

Query:

```
select e.EMP_FNAME, e.EMP_LNAME, d.DPT_NAME, d.DPT_ALLOTMENT
from emp_details e
join emp_department d
on e.EMP_DEPT = d.DPT_CODE;
```

Output:

EMP_FNAME	EMP_LNAME	DPT_NAME	DPT_ALLOTMENT
Michale	Robbin	IT	65000
Carlos	Snares	Finance	15000
Enric	Dosio	IT	65000
Jhon	Snares	Finance	15000
Joseph	Dosni	HR	240000
Zanifer	Emily	HR	240000
Kuleswar	Sitaraman	IT	65000
Henrey	Gabriel	HR	240000
Alex	Manuel	IT	65000
George	Mardy	RD	55000
Mario	Saule	Finance	15000
Alan	Snappy	RD	55000
Maria	Foster	IT	65000

28.Write a query in SQL to find the first name and last name of employees working for departments with a budget more than Rs. 50000.

Query:

```
select e.EMP_FNAME, e.EMP_LNAME, d.DPT_NAME, d.DPT_ALLOTMENT
from emp_details e
join emp_department d
on e.EMP_DEPT = d.DPT_CODE
where d.DPT_ALLOTMENT > 50000;
```

Output:

EMP_FNAME	EMP_LNAME	DPT_NAME	DPT_ALLOTMENT
Michale	Robbin	IT	65000
Enric	Dosio	IT	65000
Joseph	Dosni	HR	240000
Zanifer	Emily	HR	240000
Kuleswar	Sitaraman	IT	65000
Henrey	Gabriel	HR	240000
Alex	Manuel	IT	65000
George	Mardy	RD	55000
Alan	Snappy	RD	55000
Maria	Foster	IT	65000

29.Write a query in SQL to find the names of departments where more than two employees are working.

Query:

```
select d.DPT_NAME from emp_details e
join emp_department d
on e.EMP_DEPT = d.DPT_CODE
group by d.DPT_NAME
having count(EMP_FNAME) > 2 ;
```

Output:

DPT_NAME
IT
Finance
HR

Date:

Marks (10):

Signature of Course Faculty: