JOINS

Question1:

Calculate the number of customer has a primary account type as "SAVINGS"

Output:

No_of_Saving_acc_Customers
7

Query:

SELECT COUNT(Customer_id) as No_of_Saving_acc_Customers FROM bank_account_details where Account_type = 'SAVINGS';

Question 2:

1) Calculate the number of account owned by the each customer

Output:

Customer_id	No_of_accounts
123001	2
123002	2
123003	1
123004	3
123005	1
123006	3
123007	4
123008	0

Query:

SELECT a.Customer_id , count(b.Account_type) as No_of_accounts from bank_customer a left join bank_account_details b on

a.customer_id = b.Customer_id group by a.Customer_id ;

2) calculate the number of account owned by the each customer having at least two account

Output

Customer_id	No_of_accounts
123001	2
123002	2
123004	3
123006	3
123007	4

Query

SELECT a.Customer_id , count(b.Account_type) as No_of_accounts from bank_customer a left join bank_account_details b on

a.customer_id = b.Customer_id group by a.Customer_id having No_of_accounts >= 2 ;

Question 3

List the customer name whose primary account type is "Credit Card"

Output:

customer_id	customer_name	Account_type
123006	Noah	Credit Card
123007	Charlie	Credit Card

Query:

select a.customer_id , b.customer_name , a.Account_type from bank_account_details a

inner join bank_customer b on a.Customer_id = b.customer_id where Account_type = "Credit Card";

List the customer name whose primary account type is "Credit Card" and order the result in descending manner

Output:

customer_id	customer_name	Account_type
123006	Noah	Credit Card
123007	Charlie	Credit Card

Query:

```
select a.customer_id , b.customer_name , a.Account_type from bank_account_details a

inner join bank_customer b on a.Customer_id = b.customer_id where Account_type = "Credit Card" order by b.customer_name DESC;
```

Question 4

Query to report the details of customer(customer id, customer_name, account number) having linking account number.

Output

customer_id	customer_name	Account_Number	Linking_Account_Number
123002	George	5000-1700-5001	4000-1956-2001
123004	Jack	5000-1700-6091	4000-1956-2900
123004	Jack	5000-1700-7791	4000-1956-2900
123001	Oliver	5000-1700-3456	4000-1956-3456
123007	Charlie	4000-1956-9977	5000-1700-9800

Query

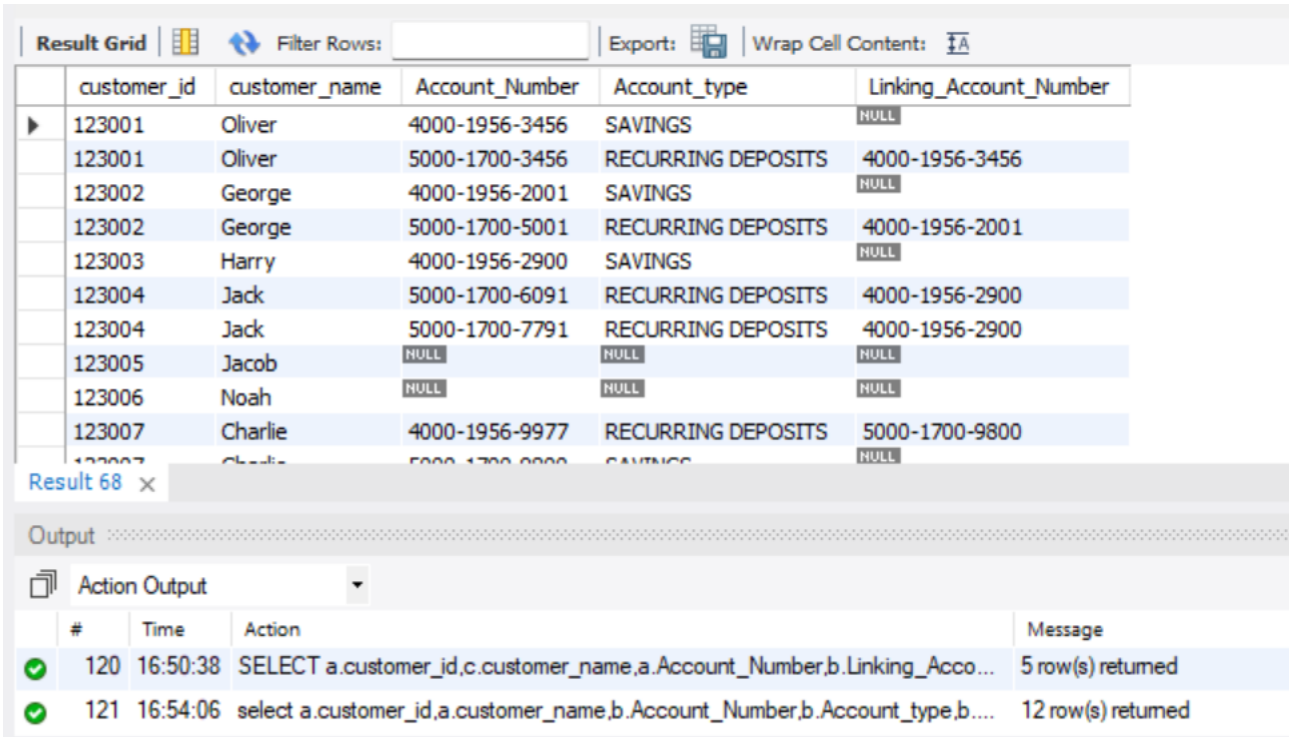
```
SELECT
a.customer_id,c.customer_name,a.Account_Number,b.Linking_Account_Number from  bank_account_details a inner join bank_account_relationship_details b on

a.Account_Number = b.Account_Number inner join bank_customer c on c.customer_id = a.Customer_id

  where b.Linking_Account_Number is not null;
```

Query to report the details of all customer having linking account number or not.

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	customer_name	Account_Number	Account_type	Linking_Account_Number
123001	Oliver	4000-1956-3456	SAVINGS	NULL
123001	Oliver	5000-1700-3456	RECURRING DEPOSITS	4000-1956-3456
123002	George	4000-1956-2001	SAVINGS	NULL
123002	George	5000-1700-5001	RECURRING DEPOSITS	4000-1956-2001
123003	Harry	4000-1956-2900	SAVINGS	NULL
123004	Jack	5000-1700-6091	RECURRING DEPOSITS	4000-1956-2900
123004	Jack	5000-1700-7791	RECURRING DEPOSITS	4000-1956-2900
123005	Jacob	NULL	NULL	NULL
123006	Noah	NULL	NULL	NULL
123007	Charlie	4000-1956-9977	RECURRING DEPOSITS	5000-1700-9800

Result 68

Output

Action Output

#	Time	Action	Message
120	16:50:38	SELECT a.customer_id,c.customer_name,a.Account_Number,b.Linking_Acco...	5 row(s) returned
121	16:54:06	select a.customer_id,a.customer_name,b.Account_Number,b.Account_type,b....	12 row(s) returned

Query

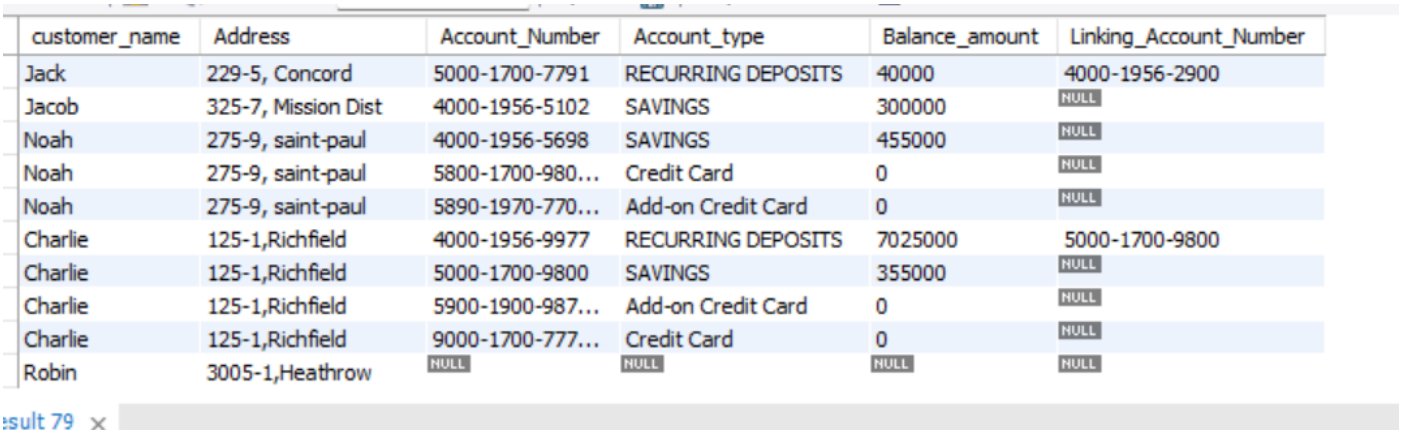
```
select
a.customer_id,a.customer_name,b.Account_Number,b.Account_type,
b.Linking_Account_Number from bank_customer a

left join bank_account_relationship_details b on a.customer_id =
b.Customer_id;
```

Question 5

Query to report the customer_name, Address,Account_Number, Account_type,Balance_amount,Linking_Account_Number from the above tables.

Output:



customer_name	Address	Account_Number	Account_type	Balance_amount	Linking_Account_Number
Jack	229-5, Concord	5000-1700-7791	RECURRING DEPOSITS	40000	4000-1956-2900
Jacob	325-7, Mission Dist	4000-1956-5102	SAVINGS	300000	NULL
Noah	275-9, saint-paul	4000-1956-5698	SAVINGS	455000	NULL
Noah	275-9, saint-paul	5800-1700-980...	Credit Card	0	NULL
Noah	275-9, saint-paul	5890-1970-770...	Add-on Credit Card	0	NULL
Charlie	125-1,Richfield	4000-1956-9977	RECURRING DEPOSITS	7025000	5000-1700-9800
Charlie	125-1,Richfield	5000-1700-9800	SAVINGS	355000	NULL
Charlie	125-1,Richfield	5900-1900-987...	Add-on Credit Card	0	NULL
Charlie	125-1,Richfield	9000-1700-777...	Credit Card	0	NULL
Robin	3005-1,Heathrow	NULL	NULL	NULL	NULL

esult 79 ×

Query:

```
Select a.customer_name, a.Address,b.Account_Number,
b.Account_type,b.Balance_amount,c.Linking_Account_Number from
bank_customer a left join bank_account_details b

on a.customer_id = b.Customer_id left join
bank_account_relationship_details c on b.Account_Number =
c.Account_Number;
```

Question 6

Query to report all the Bank Customer and their respective account number,Account type,Balance amount,Account status and Relationship type.

Output

customer_id	customer_name	Account_Number	Account_type	Balance_amount	Account_status	Relationship_type
123004	Jack	5000-1700-7791	RECURRING DEPOSITS	40000	ACTIVE	S
123005	Jacob	4000-1956-5102	SAVINGS	300000	ACTIVE	P
123006	Noah	4000-1956-5698	SAVINGS	455000	ACTIVE	P
123006	Noah	5800-1700-980...	Credit Card	0	ACTIVE	P
123006	Noah	5890-1970-770...	Add-on Credit Card	0	ACTIVE	S
123007	Charlie	4000-1956-9977	RECURRING DEPOSITS	7025000	ACTIVE	S
123007	Charlie	5000-1700-9800	SAVINGS	355000	ACTIVE	P
123007	Charlie	5900-1900-987...	Add-on Credit Card	0	ACTIVE	S
123007	Charlie	9000-1700-777...	Credit Card	0	INACTIVE	P
123008	Robin	NULL	NULL	NULL	NULL	NULL

Query

```
Select a.customer_id,a.customer_name, b.Account_Number,
b.Account_type,b.Balance_amount,b.Account_status,b.Relationship_type from bank_customer a left join bank_account_details b

on a.customer_id = b.Customer_id ;
```

Question 7

create a table called employees with following below data

query to get the all the employee id, employee name and their manager name and manager id

Output:

EmpId	EmpName	ManagerId
1	Mark	3
2	Natasha	4
3	Chris	2
4	Robert	NULL
5	Steve	2

EmpId	EmpName	ManagerId	ManagerName
1	Mark	3	Chris
2	Natasha	4	Robert
3	Chris	2	Natasha
4	Robert	NULL	NULL
5	Steve	2	Natasha

Query

```
create table employee (EmpId int , EmpName varchar(20),ManagerId int);

insert into employee
values(1,"Mark",3),(2,"Natasha",4),(3,"Chris",2),(4,"Robert",null),(5,"Steve",2);


select a.EmpId,a.EmpName,a.ManagerId , b.EmpName as ManagerName from employee a Left join employee b ON a.ManagerId = b.EmpId

 order by a.EmpId ;
```

Question 8

Write a difference between the inner join, left join, right join, self join and cross join

Answer

Inner Join:

Retrieves rows where there's a match in both tables based on the specified condition.

Syntax:

```
SELECT * FROM table1

INNER JOIN table2 ON table1.column = table2.column;
```

This query will return rows where the values in table1.column match the values in table2.column.

Left Join:

Retrieves all rows from the left table and matching rows from the right table. If no match is found in the right table, NULL values are used for the right table's columns.

Syntax:

SELECT * FROM table1

LEFT JOIN table2 ON table1.column = table2.column;

This query will return all rows from table1, along with matching rows from table2.

Right Join:

Retrieves all rows from the right table and matching rows from the left table. If no match is found in the left table, NULL values are used for the left table's columns.

Syntax:

SELECT * FROM table1

RIGHT JOIN table2 ON table1.column = table2.column;

This query will return all rows from table2, along with matching rows from table1.

Self Join:

Joins a table with itself, typically using aliases to distinguish between the two instances of the table.

Syntax:

SELECT * FROM table1 AS t1 INNER

JOIN table1 AS t2 ON t1.column = t2.column;

This query will join table1 with itself using the aliases t1 and t2.

Cross Join:

Produces a Cartesian product of all rows from both tables, resulting in a potentially large result set.

Syntax:

SELECT * FROM table1

CROSS JOIN table2;

This query will return all possible combinations of rows between table1 and table2.

Question 9

Explain full outer join and how to achieve in MySQL

Answer

A **Full Outer join** returns all rows from both the left and the right tables, along with the matching rows between them. If there's no match in either the left or right table, the result will include NULL

values for columns from the table without a match. In MySQL, there's no direct FULL OUTER JOIN syntax, but you can achieve the same result using a combination of a LEFT JOIN and a RIGHT JOIN, along with a UNION to combine the results

Syntax

```
-- Retrieve records from table1 and matching records from table2
SELECT *
FROM table1
LEFT JOIN table2 ON table1.id = table2.id
```

UNION

```
-- Retrieve records from table2 that don't have a match in table1
SELECT *
FROM table1
RIGHT JOIN table2 ON table1.id = table2.id
WHERE table1.id IS NULL;
```

Question 10

Query to report the customer_name, Address,Account_Number, Account_type,Balance_amount,Linking_Account_Number from the above tables whose customer name is Oliver,Harry.

Output

customer_name	Address	Account_Number	Account_type	Balance_amount	Linking_Account_Number
Oliver	225-5, Emeryville	4000-1956-3456	SAVINGS	200000	NULL
Oliver	225-5, Emeryville	5000-1700-3456	RECURRING DEPOSITS	9400000	4000-1956-3456
Harry	2909-5,walnut creek	4000-1956-2900	SAVINGS	750000	NULL

Query:

```
Select a.customer_name, a.Address,b.Account_Number,
b.Account_type,b.Balance_amount,c.Linking_Account_Number from
bank_customer a inner join bank_account_details b

on a.customer_id = b.Customer_id inner join
bank_account_relationship_details c on c.Account_Number =
b.Account_Number where a.customer_name in ("Oliver","Harry");
```

Question 11

Query to report the average balance amount of the each customer in their account.

Output

customer_id	customer_name	Average_Bal
123001	Oliver	4800000.0000
123002	George	3950000.0000
123003	Harry	750000.0000
123004	Jack	2731666.6667
123005	Jacob	300000.0000
123006	Noah	151666.6667
123007	Charlie	1845000.0000
123008	Robin	NULL

Query:

```
select a.customer_id ,  a.customer_name , avg(b.Balance_amount) as
Average_Bal from bank_customer a left join bank_account_details b

 on a.customer_id = b.Customer_id

 group by a.customer_id;
```