SQL Banking Assignment

1.Fetch the transaction id, date and amount of all debit transactions

```
Select transaction_id, transaction_date, transaction_amount
from Transactions
where credit_debit_flag = 'D';
```

transaction_id integer	transaction_date date	transaction_amount double precision
3	2024-07-29	50
6	2024-07-28	100
10	2024-07-25	200

2. Fetch male employees who earn more than 5000 salaries.

```
select emp_name from employees
where salary > 5000 and gender= 'M';
```

emp_name character varying (100)
James Bond
David Smith

3. Fetch employees whose name starts with J or whose salary is greater than or equal to 70000

```
select emp_name
from employees
where salary > 70000 or Upper(emp_name) like 'J%';
```

emp_name character varying (100)
James Bond

4. Fetch accounts with balance in between 1000 to 3000

```
select * from accounts
where balance between 1000 and 3000;
```

account_no [PK] bigint	balance integer	account_status character varying (10)	date_of_opening date
1100444104	1100	Active	2022-10-15
1100444105	2200	Active	2022-12-10

SQL Banking Assignment

5. Find customers who did not provide a phone no

```
select * from customers
where phone is Null;
```

customer_id [PK] character varying (10)	first_name character varying (40)	last_name character varying (40)	phone bigint	address character varying (200)	dob date
C5	Steven	Smith	[null]	Chennai	1994-12-20
C6	Jason	Holder	[null]	Chennai	1995-02-01

6. Find all the different products purchased by the customers

```
select ca.customer_id, concat(cs.first_name, ' ' ,cs.last_name) as customer_name, pr.prod_name
from customer_accounts ca
join products pr
on ca.prod_id= pr.prod_id
join customers cs
on ca.customer_id=cs.customer_id
group by 1,2,3
order by 1;
```

customer_id character varying (10)	customer_name text	prod_name character varying (100)
C1	Satya Sharma	Home Loan
C1	Satya Sharma	Personal Loan
C1	Satya Sharma	Savings Account
C2	Jaswinder Singh	Current Account
C3	Satya Sharma	Current Account

7. Sort all the active accounts with product name, customer_id & customer_name based on highest balance and based on the earliest opening date

```
select a.account_no, a.balance, a.date_of_opening, pr.prod_name,
ca.customer_id,concat(cs.first_name, ' ' ,cs.last_name) as customer_name from accounts a
join customer_accounts ca
on a.account_no = ca.account_no
join customers cs
on ca.customer_id= cs.customer_id
join products pr
on ca.prod_id = pr.prod_id
where lower(account_status) = 'active'
order by balance desc, date_of_opening, customer_id asc;
```

account_no bigint	balance integer	date_of_opening date	prod_name character varying (100)	customer_id character varying (10)	customer_name text
1100444106	3300	2022-11-05	Current Account	C3	Satya Sharma
1100444105	2200	2022-12-10	Current Account	C2	Jaswinder Singh
1100444104	1100	2022-10-15	Personal Loan	C1	Satya Sharma
1100444102	900	2020-01-10	Home Loan	C1	Satya Sharma
1100444103	500	2021-11-21	Personal Loan	C1	Satya Sharma
1100444101	100	2020-01-01	Savings Account	C1	Satya Sharma

SQL Banking Assignment

8. Fetch the oldest 5 transactions.

```
select * from transactions
order by transaction_date asc
limit 5;
```

transaction_id integer	transaction_date date	transaction_amount double precision	credit_debit_flag character varying (1)	account_no bigint
10	2024-07-25	200	D	1100444105
9	2024-07-25	100	C	1100444105
8	2024-07-26	200	C	1100444104
7	2024-07-27	100	C	1100444103
6	2024-07-28	100	D	1100444102

9. Find customers who are either from Bangalore/Chennai and their phone number is available OR those who were born before 1990

```
select first_name, last_name, phone, address, dob from customers
where address in ('Bangalore', 'Chennai') and phone is not null or extract(year from dob)<1990
```

first_name character varying (40)	last_name character varying (40)	phone bigint	address character varying (200)	dob date
Satya	Sharma	9900889911	Bangalore	1990-03-01
Jaswinder	Singh	9900889922	Mumbai	1980-03-24

10. Categorise accounts based on their balance. [Below 1k is Low balance, between 1k to 2k is average balance, above 2k is high balance]

```
select account_no, balance,
case
when balance < 1000 then 'Low'
when balance between 1000 and 2000 then 'Average'
when balance > 2000 then 'High'
end as Category
from accounts
order by balance desc;
```

account_no [PK] bigint	balance integer	category text
1100444106	3300	High
1100444105	2200	High
1100444104	1100	Average
1100444102	900	Low
1100444103	500	Low
1100444101	100	Low