Basic SELECT & Filtering

Q1.List the names and ages of all clients under 30 years old.

```
SELECT name, age
FROM clients_banking
WHERE age < 30;
```

Q2.Show the top 5 clients with the highest estimated income.

```
SELECT client_id, name, estimated_income
FROM clients_banking
ORDER BY 2 DESC
LIMIT 5;
```

Q3.Retrieve clients who joined the bank before 2010.

```
SELECT
    client_id,
    EXTRACT(YEAR FROM joined_bank) AS date_joined
FROM clients_banking
WHERE EXTRACT(YEAR FROM joined_bank) < 2010;
```

-------OR-------

```
SELECT client_id, name, joined_bank
FROM clients_banking
WHERE joined_bank < DATE '2010-01-01';
```

Joins Qns

Q4.List each clients name along with their gender.

```
SELECT cb.name, g.gender
FROM clients_banking cb
JOIN gender g
ON cb.genderid = g.genderid;
```

Q5.Show each clients name,age,and type of banking relationship.

```
SELECT
    cb.name,
        cb.age,
        br.banking_relationship
FROM clients_banking cb
JOIN banking_relationship br
ON cb.brid = br.brid;
```

Q6.Display clients and their assigned investment advisor names.

```
SELECT
    cb.name,
    ia.investment_advisor
FROM clients_banking cb
JOIN investment_advisor ia
ON cb.iaid = ia.iaid;
```

Aggregates Qns

Q7.What is the average estimated income of all clients?

```
SELECT
        ROUND(AVG(estimated_income)::numeric,0) AS avg_est_income
FROM clients_banking;
```

Q8.How many clients have more than one property?

```
SELECT
    name,
    COUNT(DISTINCT client_id) AS total_count,
    properties_owned
FROM clients_banking
WHERE properties_owned > 1
GROUP BY 1,3;
```

Q9.Find the total savings(Saving Accounts) by gender.

```
SELECT
        g.gender,
        ROUND(SUM(cb.saving_accounts)::numeric,0) AS total_saving
FROM gender g
JOIN clients_banking cb
ON g.genderid = cb.genderid
GROUP BY 1;
```

Group by

Q10.Group clients by banking relationship and show the average bank loan for each group.

```
SELECT
        cb.client_id,
        br.banking_relationship,
        AVG(cb.bank_loans) AS avg_bank_loans
FROM clients_banking cb
JOIN banking_relationship br
ON cb.brid = br.brid
GROUP BY 1,2;
```

Q11.Count how many clients each investment advisor manages.

```
SELECT
        ir.investment_advisor,
        COUNT(cb.client_id) AS client_count
FROM investment_advisor ir
JOIN clients_banking cb
ON ir.iaid = cb.iaid
GROUP BY 1
ORDER BY 2 DESC;
```

Advanced Query

Q12.Find the top 3 locations(Location ID) with the highest total business lending.

```
SELECT
        location_id,
        SUM(business_lending) AS total_business_lending
FROM clients_banking
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3;
```

Q13.List all Female clients who have more than $500,000 is savings.

```
SELECT
        g.gender,
        cb.saving_accounts
FROM gender g
JOIN clients_banking cb
ON g.genderid = cb.genderid
WHERE cb.saving_accounts > 500000;
```

Q14.What's the correlation between Age and Bank loans for clients with Gold loyalty classification?

```
SELECT
    CORR(age, bank_loans) AS age_loan_correlation
FROM clients_banking
WHERE loyalty_classification = 'Gold';
```

Q15.List all client names and their gender.

```
SELECT
        cb.name,
         g.gender
FROM clients_banking cb
JOIN gender g
ON cb.genderid = g.genderid;
```

Q16.Show clients who joined the bank after 2015

```
SELECT name, joined_bank
FROM clients_banking
WHERE joined_bank > '12-31-2015';
```

-------OR-------

```
SELECT
         name,
         EXTRACT(YEAR FROM joined_bank) AS joined_year
FROM clients_banking
WHERE EXTRACT(YEAR FROM joined_bank) > 2015;
```

Q17.Get the total number of clients by gender

```
SELECT
    g.gender,
    COUNT(cb.client_id) AS total_client
FROM gender g
JOIN clients_banking cb
```

```
ON g.genderid = cb.genderid
GROUP BY 1;
```

Q18.List the top 10 clients with the highest total bank deposits

```
SELECT
        name,
        SUM(bank_deposits) AS total_bank_deposits
FROM clients_banking
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
```

Q19.Show the number of clients per banking relationship type

```
SELECT
        br.banking_relationship,
        COUNT(cb.client_id) AS client_count
FROM banking_relationship br
JOIN clients_banking cb
ON br.brid = cb.brid
GROUP BY 1;
```

Q20.Find the average saving account balance grouped by occupation

```
SELECT
        occupation,
        ROUND(AVG(saving_accounts)::numeric,0) AS avg_saving_accounts
FROM clients_banking
GROUP BY 1;
```

Q21.Which investment advisor manages the most clients?

```
SELECT
        ia.investment_advisor,
        COUNT(cb.client_id) AS total_client
FROM investment_advisor ia
JOIN clients_banking cb
ON ia.iaid = cb.iaid
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Q22.List all clients who have more than 2 million in business lending

```
SELECT
        name,
        business_lending
FROM clients_banking
WHERE business_lending > 2000000;
```

Q23.Find the average age of clients for each loyalty classification.

```
SELECT
        loyalty_classification,
        AVG(age) AS avg_age
FROM clients_banking
GROUP BY 1;
```

Q24.Which nationality has the highest average foreign currency account balance?

```
SELECT
        nationality,
        AVG(foreign_currency_account) AS avg_foreign_currency
FROM clients_banking
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Advanced Qns

Q25.Find the client with the highest total across all types of accounts.

```
SELECT
        name,
        (checking_accounts + saving_accounts + foreign_currency_account) AS total_balance
FROM clients_banking
ORDER BY total_balance DESC
LIMIT 1;
```

Q26.Calculate the risk-weighted score for each client.

```
SELECT
        name,
        (risk_weighting * bank_deposits) AS risk_weighted_score
FROM clients_banking;
```

Q27.Which gender has the highest average total deposits (savings + checking + business lending)?

```
SELECT
        g.gender,
        AVG(saving_accounts + checking_accounts + business_lending) AS avg_total_deposits
FROM clients_banking cb
JOIN gender g
ON cb.genderid = g.genderid
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Q28.List the investment advisors and the total properties owned by their clients.

```
SELECT
        ia.investment_advisor,
        SUM(cb.properties_owned) AS total_properties
FROM clients_banking cb
JOIN investment_advisor ia
ON cb.iaid = ia.iaid
GROUP BY 1
```

```
ORDER BY 2 DESC;
```

Q29.Find clients who own no properties and have a risk weighting above 3.

```
SELECT
    name,
        risk_weighting,
        properties_owned
FROM clients_banking
WHERE properties_owned = 0 AND risk_weighting > 3;
```