SQL Queries and Sample Dataset

(QUESTIONS ASKED FROM ME IN ANGEL ONE)

1️⃣Calculate the Average Time Taken by Users to Make Their First Investment

```
SELECT AVG(TIMESTAMPDIFF(SECOND, signup_time, first_investment_time)) AS
average_time_to_invest
FROM users
WHERE first_investment_time IS NOT NULL;
```

2️⃣Identify Users Who Encounter Errors on 3 Consecutive Tries

```
SELECT user_id
FROM errors
GROUP BY user_id
HAVING COUNT(*) >= 3 AND MAX(timestamp) - MIN(timestamp) <= INTERVAL 1 HOUR;
```

3️⃣Calculate the Churn Rate of Users After 30 Days of Signup

```
SELECT (COUNT(*) / (SELECT COUNT(*) FROM users WHERE signup_time <= NOW() -
INTERVAL 30 DAY)) * 100 AS churn_rate
FROM users
WHERE last_login_time < signup_time + INTERVAL 30 DAY;
```

4️⃣Find the Top 3 Most Popular Investment Options

```
SELECT investment_option, COUNT(*) AS investment_count
FROM investments
GROUP BY investment_option
```

```
ORDER BY investment_count DESC
LIMIT 3;
```

5️⃣ Identify Users Whose First 5 Investments Exceed $10,000 in Total

```
SELECT user_id
FROM investments
GROUP BY user_id
HAVING SUM(amount) > 10000 AND COUNT(*) <= 5;
```

6️⃣ Find the Percentage of Users Who Logged in at Least 5 Times in the First Week After Signup

```
SELECT (COUNT(DISTINCT user_id) / (SELECT COUNT(*) FROM users)) * 100 AS
percentage_active_users
FROM logins
WHERE timestamp >= signup_time AND timestamp < signup_time + INTERVAL 7 DAY
GROUP BY user_id
HAVING COUNT(*) >= 5;
```

7️⃣ Detect Users Who Have Invested at Least Once Every Month in the Past Year

```
SELECT user_id
FROM investments
WHERE timestamp >= NOW() - INTERVAL 1 YEAR
GROUP BY user_id
HAVING COUNT(DISTINCT DATE_FORMAT(timestamp, '%Y-%m')) = 12;
```

8️⃣Query to Find Employees with 10% Salary Increase

```
SELECT e1.employee_id,
       e1.year AS current_year,
       e1.salary AS current_salary,
       e2.salary AS previous_salary
FROM employee_salaries e1
JOIN employee_salaries e2
   ON e1.employee_id = e2.employee_id
   AND e1.year = e2.year + 1
WHERE e1.salary >= e2.salary * 1.10;
```

DATASET WAS NOT GIVEN IN INTERVIEW MANY PEOPLE WANTED TO HAVE A SAMPLE DATA SO THAT THEY CAN CHECK THEIR OUTCOME SO HERE IT IS :-

```
-- Create users table
CREATE TABLE users (
   user_id INT PRIMARY KEY,
   signup_time DATETIME,
   last_login_time DATETIME
);

-- Create investments table
CREATE TABLE investments (
   investment_id INT PRIMARY KEY,
   user_id INT,
   amount DECIMAL(10, 2),
   timestamp DATETIME,
   investment_option VARCHAR(50),
   FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- Create logins table
CREATE TABLE logins (
   login_id INT PRIMARY KEY,
   user_id INT,
   timestamp DATETIME,
   FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

```sql
-- Create errors table
CREATE TABLE errors (
    error_id INT PRIMARY KEY,
    user_id INT,
    timestamp DATETIME,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- Create employee_salaries table
CREATE TABLE employee_salaries (
    employee_id INT,
    year INT,
    salary DECIMAL(10, 2),
    PRIMARY KEY (employee_id, year)
);

-- Sample data for Question 1
INSERT INTO users (user_id, signup_time, last_login_time) VALUES
(1, '2023-01-01 10:00:00', '2023-01-02 10:00:00'),
(2, '2023-01-05 11:00:00', '2023-01-10 12:00:00');

INSERT INTO investments (investment_id, user_id, amount, timestamp, investment_option)
VALUES
(1, 1, 5000.00, '2023-01-02 10:15:00', 'Stocks'),
(2, 2, 6000.00, '2023-01-06 12:00:00', 'Bonds');

-- Sample data for Question 2
INSERT INTO users (user_id, signup_time, last_login_time) VALUES
(3, '2023-02-01 09:00:00', '2023-02-02 10:00:00');

INSERT INTO errors (error_id, user_id, timestamp) VALUES
(1, 3, '2023-02-01 10:10:00'),
(2, 3, '2023-02-01 10:12:00'),
(3, 3, '2023-02-01 10:14:00');

-- Sample data for Question 3
INSERT INTO users (user_id, signup_time, last_login_time) VALUES
(4, '2023-03-01 08:00:00', '2023-03-15 09:00:00');

-- Sample data for Question 4
INSERT INTO investments (investment_id, user_id, amount, timestamp, investment_option)
VALUES
(3, 1, 4000.00, '2023-01-03 10:15:00', 'Stocks'),
(4, 1, 2000.00, '2023-01-04 10:15:00', 'Bonds'),
(5, 2, 3000.00, '2023-01-05 10:15:00', 'Real Estate');

-- Sample data for Question 5
INSERT INTO investments (investment_id, user_id, amount, timestamp, investment_option)
VALUES
(6, 1, 7000.00, '2023-01-05 12:00:00', 'Stocks'),
(7, 2, 8000.00, '2023-01-07 10:00:00', 'Bonds');

-- Sample data for Question 6
INSERT INTO logins (login_id, user_id, timestamp) VALUES
(1, 1, '2023-01-01 10:05:00'),
(2, 1, '2023-01-02 10:15:00'),
(3, 1, '2023-01-03 10:20:00'),
(4, 1, '2023-01-04 10:25:00'),
(5, 1, '2023-01-05 10:30:00');

-- Sample data for Question 7
INSERT INTO investments (investment_id, user_id, amount, timestamp, investment_option)
VALUES
(8, 2, 3000.00, '2023-01-06 10:00:00', 'Stocks'),
(9, 2, 4000.00, '2023-02-06 10:00:00', 'Bonds'),
(10, 2, 5000.00, '2023-03-06 10:00:00', 'Real Estate');

-- Sample data for Question 8
INSERT INTO employee_salaries (employee_id, year, salary) VALUES
(1, 2021, 50000.00),
(1, 2022, 55000.00),
(1, 2023, 66000.00),  -- 20% increase
(2, 2021, 60000.00),
(2, 2022, 66000.00),
(2, 2023, 72000.00);  -- 9.09% increase
```