Task 3

SQL for Data Analysis

## What is the difference between WHERE and HAVING?

- WHERE is used to filter rows before any grouping is done.  
- HAVING is used to filter groups after the GROUP BY clause.  
  
Example:  
```sql  
-- Using WHERE  
SELECT category, COUNT(\*)   
FROM products   
WHERE price > 100   
GROUP BY category;  
  
-- Using HAVING  
SELECT category, COUNT(\*)   
FROM products   
GROUP BY category  
HAVING COUNT(\*) > 10;  
```

## What are the different types of joins?

- INNER JOIN – Returns records with matching values in both tables.  
- LEFT JOIN – Returns all records from the left table, and matched records from the right table.  
- RIGHT JOIN – Returns all records from the right table, and matched records from the left table.  
- FULL OUTER JOIN – Returns all records when there is a match in either table (not supported directly in MySQL).  
- CROSS JOIN – Returns the Cartesian product of both tables.

## How do you calculate average revenue per user in SQL?

Assuming you have an orders table with user\_id and order\_total:  
  
```sql  
SELECT   
 AVG(user\_revenue) AS avg\_revenue\_per\_user  
FROM (  
 SELECT user\_id, SUM(order\_total) AS user\_revenue  
 FROM orders  
 GROUP BY user\_id  
) AS revenue\_per\_user;  
```

## What are subqueries?

Subqueries are SQL queries nested inside another query. They can be used in WHERE, FROM, or SELECT clauses to help filter or compute intermediate values.  
  
Example:  
```sql  
SELECT name   
FROM customers   
WHERE id IN (  
 SELECT customer\_id   
 FROM orders   
 WHERE total > 1000  
);  
```

## How do you optimize a SQL query?

- Use indexes on columns in WHERE, JOIN, and ORDER BY.  
- Avoid using SELECT \*. Select only the required columns.  
- Use the EXPLAIN command to analyze query performance.  
- Limit the data being processed using filters or LIMIT.  
- Prefer JOINs over deeply nested subqueries when possible.

## What is a view in SQL?

A view is a virtual table created from a SQL query. It simplifies complex queries and helps with data abstraction and security.  
  
Example:  
```sql  
CREATE VIEW customer\_orders AS  
SELECT c.name, o.order\_date, o.total  
FROM customers c  
JOIN orders o ON c.id = o.customer\_id;  
```

## How would you handle null values in SQL?

- Use IS NULL or IS NOT NULL to filter nulls.  
- Use COALESCE() or IFNULL() to replace nulls with default values.  
- Remember: aggregate functions like SUM, AVG, and COUNT often ignore nulls.  
  
Example:  
```sql  
SELECT name, COALESCE(phone, 'N/A') AS phone\_number   
FROM customers;  
```