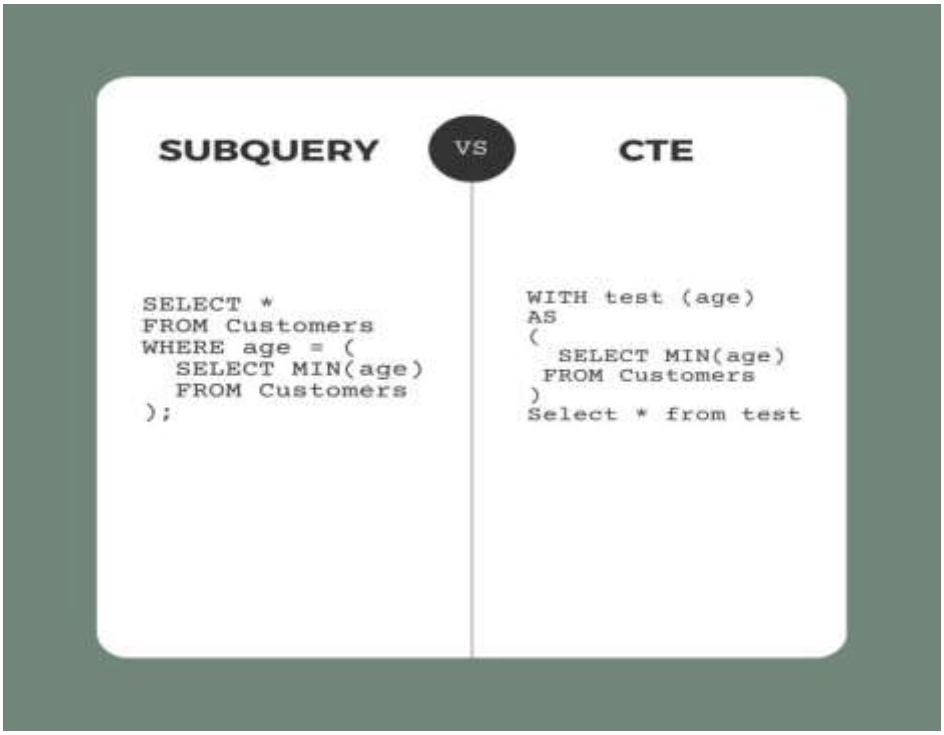
Session 5 – Manual

Advanced Query Techniques

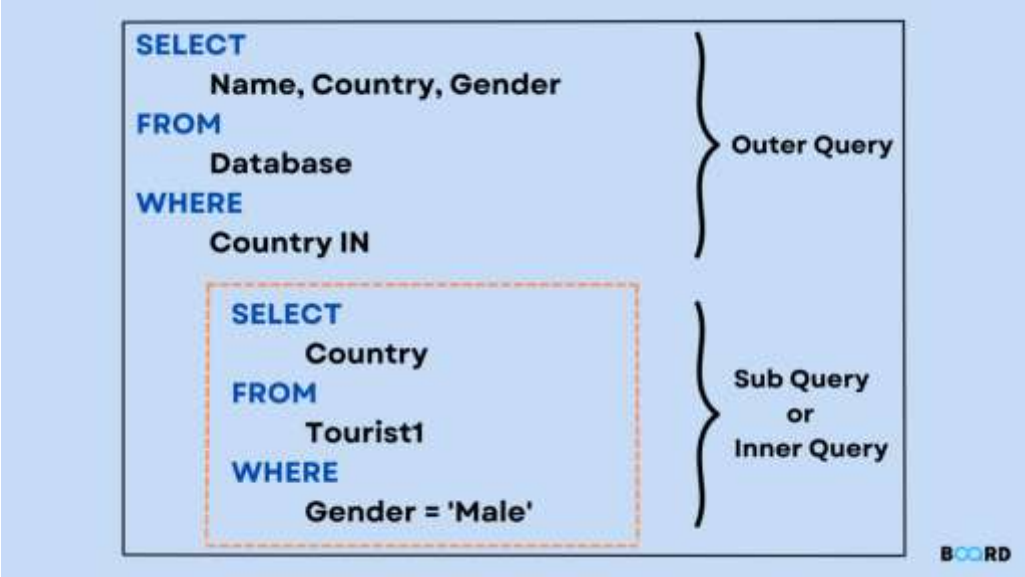
Objective:

Session: Advanced Query Techniques

- Enhance skills in applying different types of joins through practical examples and optimization techniques.
- Learn to construct and utilize subqueries in different parts of SQL queries (SELECT, WHERE, FROM).
- Understand and implement derived tables for more readable and efficient queries.
- Gain proficiency in creating and using Common Table Expressions (CTEs) to organize and simplify complex queries, including recursive CTEs.



Session: Advanced Query Techniques



General example of subquery

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
WHERE age = (
  SELECT MIN(age)
  FROM Customers
);
```

customer_id	first_name	last_name	age	country
2	Robert	Luna	22	USA
3	David	Robinson	22	UK

Working with Subqueries

Key Points:

- **Purpose**: To use subqueries for breaking down complex queries into manageable parts.

Types of Subqueries:

1. **Single-Row Subquery**:

```
SELECT
  column_name
FROM
  table_name
WHERE
  column_name = (SELECT MAX(column_name) FROM table_name);
```

2. **Multi-Row Subquery**:

```
SELECT
  column_name
FROM
  table_name
WHERE
  column_name IN (SELECT column_name FROM another_table);
```

3. **Subqueries in SELECT**:

```
SELECT
  column_name,
  (SELECT MAX(column_name) FROM another_table) as max_value
FROM
  table_name;
```

4. **Subqueries in FROM**:

```
SELECT
  sub.column_name
FROM
  (SELECT column_name FROM table_name) sub;
```

Working with Derived Tables

Key Points:

- **Purpose**: To use derived tables for more readable and efficient queries.

Example:

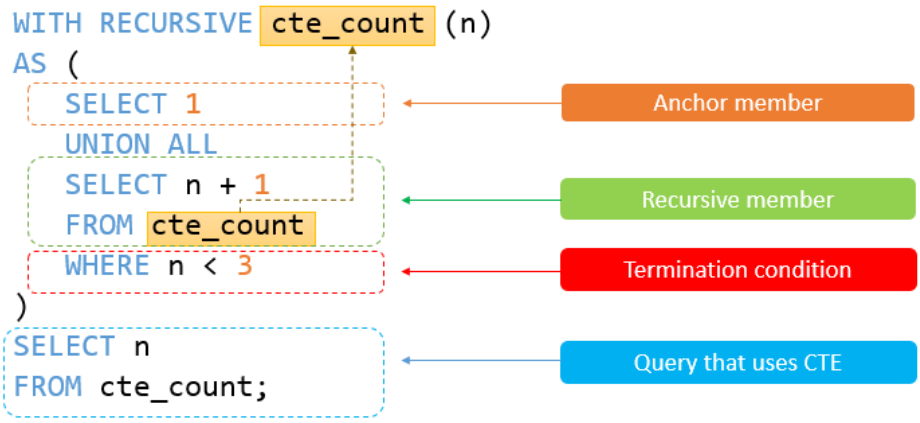
```
SELECT
  a.column_name, b.column_name
FROM
  (SELECT column_name FROM table_name) a
JOIN
  another_table b ON a.common_field = b.common_field;
```

Benefits:

- Temporary result set for complex queries.
- Simplifies query structure.

Common Table Expressions (CTEs)

General Example of CTE



Key Points:

- **Purpose**: To organize and simplify complex queries using CTEs.

Syntax:

```
WITH cte_name AS (
  SELECT column_name FROM table_name
)
SELECT
  column_name
FROM
  cte_name;
```

Examples:

- **Simple CTE**:

```
WITH Sales_CTE AS (
  SELECT
    employee_id, SUM(sales) as total_sales
  FROM
    sales
  GROUP BY
    employee_id
)
SELECT
  employee_id, total_sales
FROM
  Sales_CTE;
```

- **Recursive CTE**:

```
WITH Recursive_CTE (column_name, level) AS (
  SELECT
    column_name, 1
  FROM
    table_name
  UNION ALL
  SELECT
    t.column_name, r.level + 1
  FROM
    table_name t
  JOIN
    Recursive_CTE r ON t.parent_id = r.id
)
SELECT
  column_name, level
FROM
  Recursive_CTE;
```

By following this detailed manual, you will develop a comprehensive understanding of advanced query techniques, and how to effectively use these concepts to write powerful SQL queries.