Database Management: Understanding and Applying Joins and Subqueries

Introduction

This document aims to provide a foundational understanding of working with multiple tables in databases, focusing on database relationships. It's designed to serve as a preparatory guide for students who are about to dive into these essential topics in database management systems.

Section 1: Working with Multiple Tables

Understanding Database Relationships

1. One-to-Many Relationships

- Definition: This occurs when a record in one table can be associated with multiple records in another table.
- Example: A single customer having multiple orders; here, the customer table has a one to-many relationship with the orders table.

2. Many-to-Many Relationships

- Definition: This exists when multiple records in a table are associated with multiple records in another table.
- Example: Students and courses; a student can enroll in many courses, and a course can have many students.

3. One-to-One Relationships

- Definition: This relationship exists when a single record in one table is associated with a single record in another table.
- Example: Person and Passport; each person has at most one passport, and each passport is assigned to at most one person. In this scenario, the person's record in the 'Person' table will have a unique identifier (like a Social Security Number) that matches a unique identifier in the 'Passport' table (like a Passport Number).

4. Many-to-One Relationships

- Definition: This relationship exists when multiple records in one table are associated with

a single record in another table.

- Example: Employees and Department; multiple employees can work in the same department, but each employee belongs to only one department. In this case, many employee records in the 'Employees' table will be linked to a single department record in the 'Department' table through a department identifier.

Using JOINs to Combine Data from Multiple Tables

1. INNER JOIN

- Usage: Retrieves records that have matching values in both tables.
- Syntax Example: SELECT * FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;
- Application: Used when you only want to retrieve the rows where there is at least one match in both tables.

2. LEFT JOIN (or LEFT OUTER JOIN)

- Usage: Returns all records from the left table (table1), and the matched records from the right table (table2).
- Syntax Example: SELECT * FROM table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;
- Application: Useful when you want all records from the left table, regardless of whether they have matches in the right table.

3. RIGHT JOIN (or RIGHT OUTER JOIN)

- Usage: Returns all records from the right table, and the matched records from the left table.
- Syntax Example: SELECT * FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;
- Application: Opposite of LEFT JOIN; use when you need all records from the right table.

4. FULL JOIN (or FULL OUTER JOIN)

- Usage: Selects records that have matching records in either the left or right table. ● Syntax Example: SELECT * FROM table1 FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;

- Application: Use when you want to combine all records from both tables, with NULL in place where there is no match.

Section 2: Subqueries

Understanding Subqueries and Their Applications

1. Definition

- A subquery is a query nested inside another query, such as SELECT, INSERT, UPDATE, or DELETE. Also known as inner queries or nested queries.

2. Applications

- Used for complex queries, like when you want to compare values in a column with a set of values returned by another query.

Types of Subqueries

1. Single-Row Subqueries

- Returns only one row.
- Example: SELECT * FROM table1 WHERE column1 = (SELECT column2 FROM table2 WHERE condition);

2. Multiple-Row Subqueries

- Returns more than one row.
- Uses operators like IN, ANY, ALL.
- Example: SELECT * FROM table1 WHERE column1 IN (SELECT column2 FROM table2 WHERE condition);

3. Multiple-Column Subqueries

- Returns more than one column.
- Often used in the SELECT clause.
- Example: SELECT * FROM table1 WHERE (column1, column2) IN (SELECT column3, column4 FROM table2 WHERE condition);

Correlated Subqueries

- A subquery that references one or more columns from the outer SQL query.
- Evaluated once for each row processed by the outer query.
- Example: SELECT * FROM table1 t1 WHERE EXISTS (SELECT 1 FROM table2 t2 WHERE t1.id = t2.id);