**Sub-queries**

A sub-query is a SELECT statement that is nested within another SELECT statement.

It can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself.

**Example of sub-query**

A screenshot of a computer screen

AI-generated content may be incorrect.

**Nb:** If we needed to return the details of the least expensive books of each type of the sub query would return more than one row.

A screenshot of a table

AI-generated content may be incorrect.

A Multiple row sub-query returns more than one row. We cannot use the single row operators with multiple rows. Therefore, we need special operators like **IN**.

**Sub-queries – IN**

When IN is used all the values in the outer query are matched against a list of values returned by the inner query.

Example

A screenshot of a book

AI-generated content may be incorrect.

**Nb**: Sub queries can be used when we need to gather information from one table before using this information as part of a condition in a query that operates on another table.

**Example**: Using the following tables, we can use a sub-query to determine the details of students on advanced courses.

A screenshot of a table

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Nb:** Using the following tables, we can also use a similar sub-query to determine the details of students who are NOT on advanced courses.

A screenshot of a computer

AI-generated content may be incorrect.



**JOINS**

A join is used when a SQL query requires data from more than one table in the database. Rows in one table may be joined to rows in another table according to common values existing in corresponding columns.

**Types of SQL Joins**

1. **INNER JOIN** – Returns only matching rows from both tables.
2. **LEFT JOIN (LEFT OUTER JOIN)** – Returns all rows from the left table and matching rows from the right table. If there’s no match, NULL is returned.
3. **RIGHT JOIN (RIGHT OUTER JOIN)** – Returns all rows from the right table and matching rows from the left table. If no match is found, NULL is returned.
4. **FULL JOIN (FULL OUTER JOIN)** – Returns all rows when there is a match in either table. If no match is found, NULL is returned from the non-matching side.

A group of circles with text

AI-generated content may be incorrect.

**Examples**:

A close-up of a diagram

AI-generated content may be incorrect.

A text on a white background

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A table of text with a student

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Nb**: We can order this output using the **ORDER BY** clause

A screenshot of a computer

AI-generated content may be incorrect.

Generally, we would join tables to determine which ‘rows’ from the ‘joined’ table satisfied certain specifications.

**Example**: looking for students on ‘advanced’ courses

A screenshot of a computer

AI-generated content may be incorrect.

**Nb**: We can join as many tables as we need to satisfy a query.

A table of information with text

AI-generated content may be incorrect.

A table with text on it

AI-generated content may be incorrect.