Advanced SQL queries examples:

Notes:

- In all examples any phrase in all caps surrounded by square brackets [] are placeholders
- All examples are just single uses of the methods and there are many more ways to use join's, group by's and sub queries. These are just examples of what could be achieved with SQL and I recommend you to use this document as a starting point to find an advanced query that is suitable for your project.
- All red text is explanations and black text is headings or SQL queries

---

JOIN:

A JOIN is a SELECT statement that connects 2 or more tables together to combine the rows in a database.

More information on JOINS: SQL Joins

JOIN example:

Format:

```
SELECT [COLUMNS].[TABLE NAME]
FROM [TABLE 1 NAME]
LEFT JOIN [TABLE 2 NAME]
ON [TABLE 1 NAME].[CONNECTING COLUMN] = [TABLE 2 NAME].[CONNECTING COLUMN];
```

Real example:

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID = Orders.CustomerID;
```

This example of a left join connects the second table directly onto the first table meaning it will print out the first tables columns normally and match the appropriate values from the second table on the end.

Subquery:

A subquery is a query inside another query that can find new information based on the first queries result.

More information on subqueries: SQL IN Operator *note scroll down to IN(SELECT)

Subquery example:

Format:

```
SELECT [COLUMNS]
FROM [TABLE 1 NAME]
WHERE [COLUMN] IN (SELECT [COLUMN] FROM [TABLE 2 NAME] WHERE [CONDITION]);
```

Real example:

```
SELECT *
FROM Customers
WHERE CustomerID IN (SELECT CustomerID FROM Orders);
```

This example of a subquery first finds all customers who have made an order in the order table then proceeds to use the list of customerIDs to print out all the customers information if they are in the list of customerIDs that have made an order.

Subqueries can be used to create a list of values that meet a specific requirement in the same or another table and an example of a subquery that uses the same table is:

Real example:

```
SELECT JobTitle
FROM Jobs
WHERE Salary > (SELECT AVG(Salary) FROM Jobs);
```

This query first uses a subquery to find the average salary then compares each row to see if the average salary is greater and outputs all rows greater than the average salary.

GROUP BY:

A GROUP BY clause allows the output to be split into certain groups depending on what it is split by. In addition a HAVING clause underneath that adds a requirement a group must meet to be output. Finally an ORDER BY can be used to change the order the groups are displayed

More information on GROUP BY: SQL GROUP BY
More information on HAVING: SQL HAVING

GROUP BY example:

Format:

```
SELECT [COLUMNS]
FROM [TABLE NAME]
GROUP BY [COLUMN];
```

Real example:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

This example of a GROUP BY uses puts all of the customers in groups by country and finds how many customers by country.

In order to have the countries in alphabetical order an ORDER BY clause can be used as follows:

Real example:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY Country;
```

Finally in order to add a condition groups need to meet in order to be displayed a having can be used. An example of this is the following example which requires the groups by countries to have at least 10 customers:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 10
ORDER BY Country;
```