Jonathan Owen
May 3, 2022
IT FDN 130 A Sp 22
Assignment 03

The SELECT Statement and its Clauses

Introduction

Assignment three was an introduction to extracting data from a database through the use of `SELECT` statements. This paper wiLL briefly describe the `SELECT` statement and some clauses that may be used within it.

The SELECT Statement

The `SELECT` statement is the primary method used to extract data from a database. The `SELECT` statement is made up of multiple parts called clauses which allow the user to refine the data that is returned. The most basic `SELECT` statement will return all the rows and columns of a single table. As `SELECT` statements become more complex, values from multiple tables may be returned and filtered to match specific criteria. Additionally, new columns may be created from calculations from column data.

The SELECT Clause

The `SELECT` clause is the foundation of any `SELECT` statement. The `SELECT` clause is used to select the columns being returned from a query. The returned columns can either be made up of values directly from the tables referenced in the query or calculated values. Calculated values may use SQL functions, such as `AVG` or `COUNT`. Calculations may also be derived using column values. When using calculated values, care must be taken to ensure the correct data type is returned. By default, the column's assigned name will be used at a header value. This may be changed by setting an alias through the use of the `AS` keyword.

The FROM Clause

The `FROM` clause of a `SELECT` statement defines from which tables data will be retrieved. A simple `FROM` clause will contain a single table. A more complicated query may require data from multiple tables. If this is the case, a join must be defined to link the additional table through a foreign and primary key combination using the `JOIN` keyword.

The WHERE Clause

In order to return only data of interest, a `WHERE` clause must be added to a select statement. A `WHERE` clause will refine the returned data by applying filters so that only values which match specified parameters are returned. Within a `WHERE` clause, keywords such as `BETWEEN`, `LIKE`, and `IN` may be used to match values within a range or data set. Additionally, logical operators may be used (i.e., =, <, >=,). When filtering a string value, wildcards are available to allow for substrings to be matched. To filter on multiple parameters within a `WHERE` clause, logical `AND` or `OR` keywords are used. If the order of operation is important among the filtering parameters, parentheses are used to group the parameters. If a `GROUP BY` statement is used within a `SELECT` statement, the `WHERE` clause is applied first.

The ORDER BY Clause

Before the return of data from a select statement, data may be sorted in ascending or descending order on the desired column or columns with the `ORDER BY` clause. By default, returned data will be in ascending order once a column is defined in an `ORDER BY` clause. In order to sort data in descending order, the `DESC` keyword must be applied after the column name. Similarly, if explicitly defining ascending order, the `ASC` keyword is used. When sorting on multiple columns, the column names are separated by commas. The sorting will be applied in the order the columns are listed.

Summary

The use of clauses within a select statement is a powerful tool for extracting data from a database. As a database becomes larger, more complex `SELECT` statements may be constructed or be needed to extract only relevant information to a user's question. With the use of aliases and the `ORDER BY` clause, data may be presented in a logical format that is easy to read.