Fonte Labs Podcast – Episode 2
SQL SELECT Queries

Jeremy Fonte
09/04/2013

- I'm Jeremy Fonte and this is the Fonte Labs podcast
- Today I'll be talking about SQL SELECT queries, the querying method for relational databases.
- Relational databases have been the dominant form of data storage for many, many years – and while there are now new, popular databases in the form of "NoSQL" products, traditional databases such as SQL Server, Oracle Database and MySQL are in widespread use.
- The basis of these relational databases is tables connected to one another through relations, that is, columns in one table that correspond to columns in another table. These tables can then be joined on the common column, allowing data operations to be performed across the two or more tables.
- SQL is the language that governs everything from creating tables to inserting values to retrieving, deleting, and updating data in the DB. In this podcast we'll be covering SELECT queries using SQL, which are used to retrieve data from a database.
- The simplest SELECT statement is "SELECT * FROM MyTable". This selects all columns from the specified table, in this case, "MyTable". The star or asterisk is essentially a wildcard that selects every column from the specified table or tables.
- To select specific columns from a database you use a query of the form

  SELECT ColumnA FROM MyTable
  or
  SELECT ColumnA, ColumnB FROM MyTable

- Selecting columns from multiple tables is typically performed using a JOIN statement. This SQL keyword allows you to connect two tables that share one or more common columns – by joining the tables, you can access any or all of the columns in either table in a single SQL SELECT statement.
- There are four main joins:
  - An INNER JOIN only display rows that match between the two tables.
  - A LEFT OUTER JOIN or LEFT JOIN retains all rows from the table on the left side of the JOIN statement, leaving out any rows on the right side that do not exist on the left side.
  - A RIGHT OUTER JOIN or RIGHT JOIN retains all rows from the table on the right side of the JOIN statement, leaving out any rows on the left side that do not exist on the right side
  - A FULL OUTER JOIN or OUTER JOIN retains all rows from either side that do not exist on the other side.

    - Some databases, notably MS Access, do not have a built-in OUTER JOIN, and rather, a tedious workaround must be employed to achieve an OUTER JOIN.
- An example of an INNER JOIN is:

  ```
  SELECT Name, Salary, PhoneNumber
  FROM Employees
  INNER JOIN Payroll ON Employees.ID = Payroll.EmployeeID
  ```

- An important point to notice is that if any column, table, or other database object has spaces in its name, you must surround the name with square brackets, like so:

  ```
  SELECT [First Column] FROM [Sales Orders]
  ```

- Sometimes you'll want to specify a condition pertaining to which rows should be returned; this is accomplished with the WHERE clause. For example:

  ```
  SELECT [Race Times] FROM [Athletes] WHERE [Gender] = "Male"
  ```

- Another critical feature of SELECT queries is aggregate queries. These queries can do anything from summing values to counting values and more. The aggregate queries generally group by some value.
- Say you want to sum the Salaries of employees by department, to see which departments are spending the most on Payroll. This is a typical query which would result:

  ```
  SELECT SUM(Salary), Department
  FROM Payroll
  GROUP BY Department
  ```

  This query sums the salaries of employees on payroll, grouping by department. This means each department will be listed along with the sum of the salaries of employees in that department.
- When you need to have a WHERE clause in an aggregate function, you must use the HAVING clause instead. It performs the same basic function as WHERE.

  ```
  SELECT SUM(Salary), Department
  FROM Payroll
  GROUP BY Department
  HAVING Salary >= 100000
  ```

This will retrieve the sum of salaries more than $100,000 a year, by department.

- That wraps up this episode of the Fonte Labs podcast.
- Thank you and goodbye.