CoGrammar

Working with SQL

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Data Science Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Data Science Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Lecture Objectives

- **Explore the syntax for the language used for manipulating and managing databases. Structured Query Language (SQL)**

Introduction to SQL

- ★ Structured Query Language (SQL) is a database language that is composed of commands (statements) that enable users to create databases or table structures.
- ★ It is also able to perform various types of data manipulation and data administration as well as query the database to extract useful information.
- ★ SQL is supported by all relational Database Management Systems (DBMS) software.

Introduction to SQL

- ★ Structured Query Language (SQL) is a database language that is composed of commands (statements) that enable users to create databases or table structures.
- ★ It is also able to perform various types of data manipulation and data administration as well as query the database to extract useful information.
- ★ SQL is supported by all relational Database Management Systems (DBMS) software.

Introduction to SQL

- ★ SQL is portable, meaning that a user does not need to relearn when moving from one DBMS to another, because all DBMS will use SQL in almost the same way.
- ★ SQL is easy to learn since its vocabulary is relatively simple. Its basic command set has a vocabulary of fewer than 100 words. It is also a non-procedural language, which means that the user specifies what must be done and not how it has to be done.

Creating Tables

★ **To create new tables in SQL, we use the CREATE TABLE statement. Pass all the columns we want in the table, as well as their data types as arguments to the function.**

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
   ....
);
```

Creating Tables

- Note the optional constraint arguments. They are used to specify rules for data in a table. Constraints that are commonly used in SQL include :
  - NOT NULL - ensures that column cannot have a null value.
  - UNIQUE - ensures that all values in a column are different.
  - DEFAULT - Sets a default value for a column when no value is specified.
  - INDEX - Creates an index which is used to create and retrieve data from the database very quickly.

Inserting Rows

- The table we have just created is empty and needs to be populated with rows or records. We can add entries to a table using the INSERT INTO command.
- There are two ways to write the INSERT INTO command :

Inserting Rows

1. Do not specify the column names where we intend to insert the data. This can be done if we are adding values for all the columns of the table. However, we should ensure that the order of the values are in the same order as the columns in the table. The syntax is as follows :

```
INSERT INTO table_name
      VALUES (value1, value2, value3, ...);
```

Inserting Rows

2. The other way to insert data into our table is to specify both the column names and the values to be inserted. The syntax will be as follows :

```
INSERT INTO table_name (column1, column2, column3, ...)
      VALUES (value1, value2, value3, ...);
```

Tea Time

Let’s take a small break before moving on to the next topic.

Retrieving Data from a Table

★ **The SELECT statement is used to fetch data from a database. The data returned is stored in a result table, known as the result-set. The syntax of a SELECT statement is as follows:**

```
SELECT column1, column2, ...
FROM table_name;
```

★ **If you want to select all columns in a table :**

```
SELECT * FROM table_name;
```

ORDER BY

- We can use the ORDER BY command to sort the results returned in ascending order or descending order. The ORDER BY command sorts the records in ascending order by default. We need to user the DESC keyword to sort the results in descending order.

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

WHERE

- The WHERE clause allows us to filter data depending on a specific condition. The syntax of the WHERE clause is a follows:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

- Note that SQL requires single quotes around text values; however, we do not need to enclose numeric fields in quotes. We can also utilize logical operators to make WHERE conditions as specific as you like.

Aggregate Functions

- SQL has many functions that do all sorts of helpful things. Some of the more commonly used functions are :
  - COUNT() - returns the number of rows.
  - SUM() - returns the total sum of a numeric column.
  - AVG() - returns the average of a set of values.
  - MIN() / MAX() - returns the minimum or maximum value from a column.

Updating Data

- ★ The UPDATE statement is used to modify the existing rows in a table.
- ★ To use the UPDATE statement we :
  - Choose the table where the row we want to change is located.
  - Set the new values for the wanted column(s).
  - Select which of the rows you want to update using the WHERE statement. If we omit this, all rows will change.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Deleting Rows

- Deleting a row is a simple process. All we need to do is select the right table and row we would like to remove. The DELETE statement is used to delete existing rows in a table.

```
DELETE FROM table_name
WHERE condition;
```

- You can also delete all rows in a table.

```
DELETE * FROM table_name;
```

Deleting Tables

- The DROP TABLE statement is used to remove every trace of a table in a database.

```
DROP TABLE table_name;
```

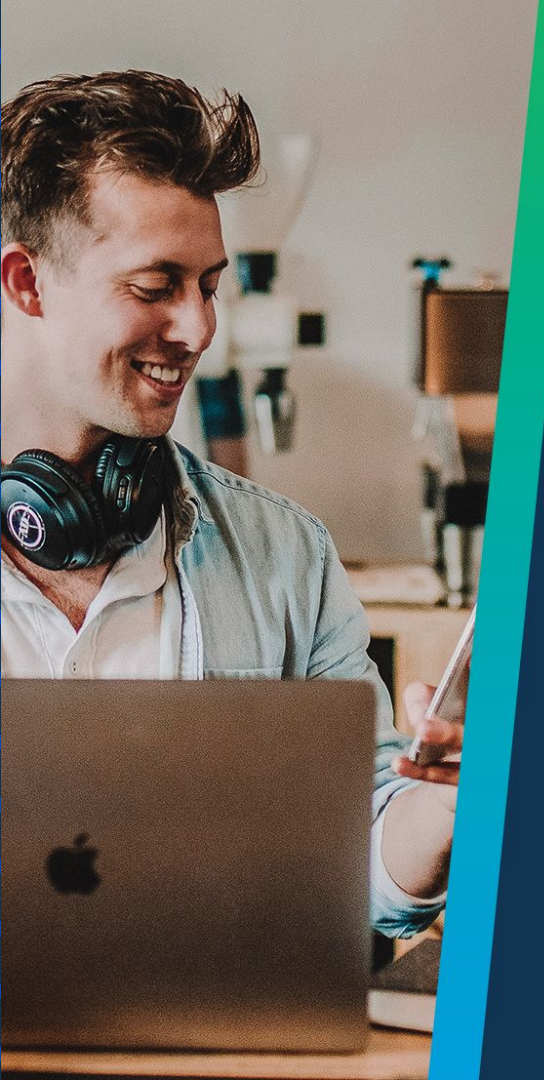
- If we want to delete the data in the table, but not the table itself, we can use the TRUNCATE TABLE statement.

```
TRUNCATE TABLE table_name;
```

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.



CoGrammar

Thank you for joining us

1. Take regular breaks
2. Stay hydrated
3. Avoid prolonged screen time
4. Practice good posture
5. Get regular exercise

"With great power comes great responsibility"