Welcome to the CoGrammar Databases and SQL Lecture

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

CoGrammar

Coding Interview Workshop Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- 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 Academic Sessions. You can submit these questions here: **Questions**

Coding Interview Workshop 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

CoGrammar

Skills Bootcamp 8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks
Guided Learning Hours (GLH): Minimum of 15 hours
Task Completion: First four tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**
Software Engineering - **13 tasks**
Web Development - **13 tasks**

Due Date: 28 April 2024

CoGrammar

Skills Bootcamp Progression Overview

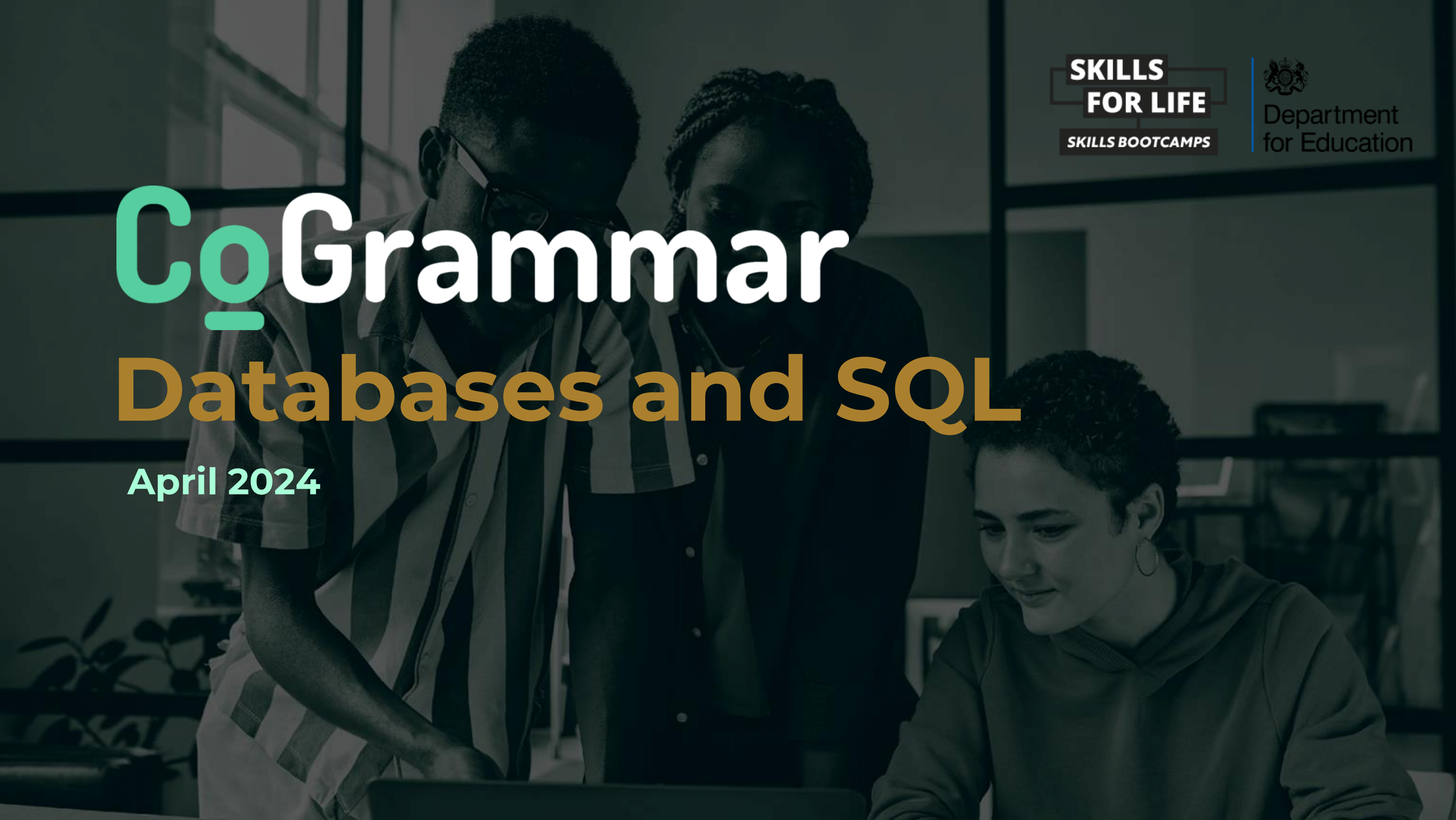
✅ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✅ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity

CoGrammar



SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

CoGrammar

Databases and SQL

April 2024

Portfolio Assignment Reviews

Submit you solutions here!

CoGrammar

Learning Objectives

- ❖ Describe **basic database concepts** and the role of SQL in **data management.**
- ❖ Execute **CRUD operations** using SQL in a relational database system.
- ❖ Construct **SQL queries** to **filter**, **aggregate**, and **manipulate** datasets effectively.
- ❖ Implement **database interactions** within **Python and JavaScript applications**, showcasing practical connection and data manipulation.

CoGrammar

Learning Objectives

❖ Design **simple relational database schemas**, emphasizing the importance of **normalization** and efficient **data organization**.

CoGrammar

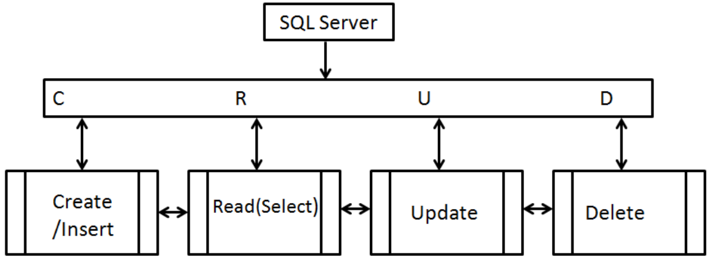
Which of the following is NOT a basic CRUD operation in database management?

A. Create
B. Retrieve
C. Update
D. Discard

CoGrammar

Correct Answer: D

- ❖ **CRUD Operations** are the 4 basic operations which act as the foundation of any computer programming language.
- ❖ CRUD stands for: Create, Retrieve, Update and **Delete**.
- ❖ It is important to understand **what** CRUD operations are and **how** they are implemented in SQL Statements.



CoGrammar

Which SQL keyword is used to retrieve data from a database?

A. SELECT
B. UPDATE
C. DELETE
D. INSERT

Correct Answer: A

- All four options are valid SQL statements.
- It is important to know the difference between the different SQL statements and what they accomplish.
- **SELECT** is a statement which retrieves data from one or more tables.
- **UPDATE** updates data in a table, **DELETE** deletes data from a table and **INSERT** inserts data into a table.

CoGrammar

Which of the following is NOT a valid SQL data type?

A. INT
B. STRING
C. DATE
D. BOOLEAN

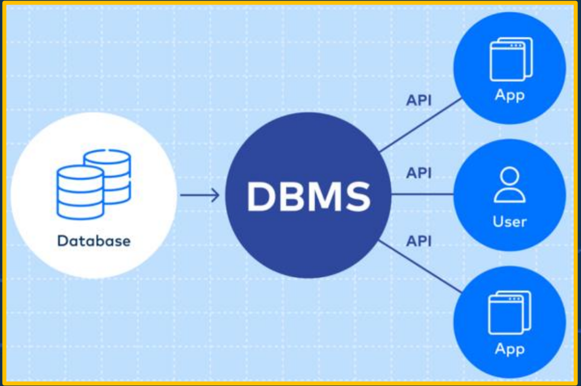
Correct Answer: B

- Strings of data are stored in SQL in different ways and there are a number of different String Data Types in SQL but STRING **is not** a type in SQL.
- Strings can be of the **CHAR(n)**, **VARCHAR(n)** or **TEXT** type, both of which can store strings.

Databases

An organized collection of structured information, or data.

- ❖ A database is usually controlled by a database engine, commonly known as a **Database Management System (DBMS).**
- ❖ DBMSs serve as a **tool** between a user and their data, **organising** and **cataloging** the data for **quick and easy retrieval.**
- ❖ The data and the DBMS, and the applications associated with them are referred to as a **database system**, usually shortened to **database.**



CoGrammar

Relational Databases

Any database system that allows data to be associated and grouped by common attributes.

- Relational databases are comprised of a number of tables (**relations**), within each are:
  - Rows also known as records or tuples
  - Columns also known as attributes or fields
- Each record is identified with a **unique key**, known as the **primary key**.
- Records from one table can be references in other tables using their key, in this case they are called **foreign keys**.
- Each table/relation represents one **“entity type”**.

CoGrammar

RDMS Example

Consider the following scenario:

You are the captain of a spaceship and you would like to store all of the data pertaining to the ship in one place so you can easily access and update the data. Some examples of the data you need to store is: information on each of the crewmates, information about each of the areas of the ship and the shift schedules which places each crew member at different areas of the ship at different times.

RDMS Example

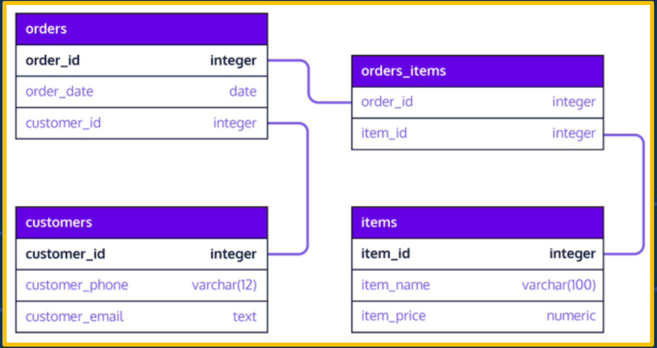
crewID	firstName	lastName	roomNum	shiftTime	shiftArea
0	Zahra	Mohamed	7	19:00	Hull
1	Moumita	Aich	12	19:00	Engine
2	Anri	Lombard	4	19:00	Cafetaria
3	Julien	Nyambal	3	19:00	Stock
0	Zahra	Mohamed	7	20:00	Engine
1	Moumita	Aich	12	20:00	Stock

CoGrammar

Database Schema

Defines how data is organised within a relational database.

- Considered to be the **"blueprint"** of a database, which **describes** how the data may **relate to other tables** or other data models.
- It defines any **logical constraints** such as table names, fields, data types and the relationships between these entities.
- Commonly use **visual representations** to illustrate database architecture.
- The process of designing data schemas is known as **data modelling**.



CoGrammar

Normalisation

Database organisation technique that includes creating smaller tables and establishing links between those tables according to rules.

- ❖ Aims to protect the data and make the database more **flexible** by **eliminating redundancy** and **inconsistent dependencies**.
- ❖ Redundant data **wastes disk space** and creates **maintenance problems**.
- ❖ Inconsistent dependencies refers to **how related data is stored** in our database tables.
- ❖ Inconsistent dependencies can make data **difficult to access**.
- ❖ There are a few **rules** for database normalisation, each of which is called a **normal form**.

CoGrammar

Normal Forms

- It is not always necessary comply to the highest normal form, **third normal form** is the usually highest level necessary.
- **First Normal Form**
  - Eliminate repeating groups in individual tables.
  - Create a separate table for each set of related data.
  - Identify each set of related data with a primary key.
- **Second Normal Form**
  - Create separate tables for sets of values that apply to multiple records
  - Relate these tables with a foreign key
- **Third Normal Form**
  - Eliminate fields that don't depend on the key

CoGrammar

Normalisation Example

Let's attempt to normalise the spaceship database together!

1. Separate related and repeated data
2. Identify primary keys for all records
3. Relate the tables using foreign keys

CoGrammar

Normalisation Example

Crew Table

crewID 🔑	firstName	lastName	roomNum
0	Zahra	Mohamed	7
1	Moumita	Aich	12
2	Anri	Lombard	4
3	Julien	Nyambal	3

Area Table

areaID 🔑	areaName
0	Hull
1	Engine
2	Cafetaria
3	Stock

Shift Table

areaID 🔑	shiftTime 🔑	crewID
0	19:00	0
1	19:00	1
0	20:00	2
1	20:00	0

CoGrammar

Let’s Breathe!

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

CoGrammar

SQL

Structures Query Language is a programming language used for communicating with relational databases.

- ❖ **SQL** is used by nearly all relational databases to **query**, **manipulate**, and **define data.**
- ❖ It was first developed at **IBM** in the **1970s** and is still widely used today although new programming languages are beginning to appear.
- ❖ SQL is divided into the following categories:
  - ➢ **Data Definition Language:** Defines the data objects within the database based on the defined data model.
  - ➢ **Data Manipulation Language:** Used to insert, update, or delete.
  - ➢ **Data Query Language:** Used to query or select data

CoGrammar

SQL: RDBMSs

- ❖ We use SQL to interact with Relational Database Management Systems and it **facilitates several operations**.
- ❖ Different DBMSs use different **dialects** of SQL.
- ❖ There are many different RDBMSs which we could work with. We will talk about two in this lecture:
  - ➢ **SQLite:** Server-less database which is self-contained, best used for small, standalone apps and basic development and testing.
  - ➢ **MySQL:** Requires a server to run and requires a client-server architecture to interact over a network. This is best used for web-based applications, larger apps with multiple user access.

CoGrammar

MySQL: Data Types

❖ MySQL supports three main groups of data types

Date Type	**Description**	**Allowed Range of Values**
	Numeric Data Types	
INT	Normal sized integer	–2147483648 to 2147483647, or 0 to 4294967295 if UNSIGNED
BIT	0 or 1	0 or 1
BOOL or BOOLEAN	Synonyms for TINYINT(1)	Zero is considered as false, nonzero values are true.

MySQL: Data Types

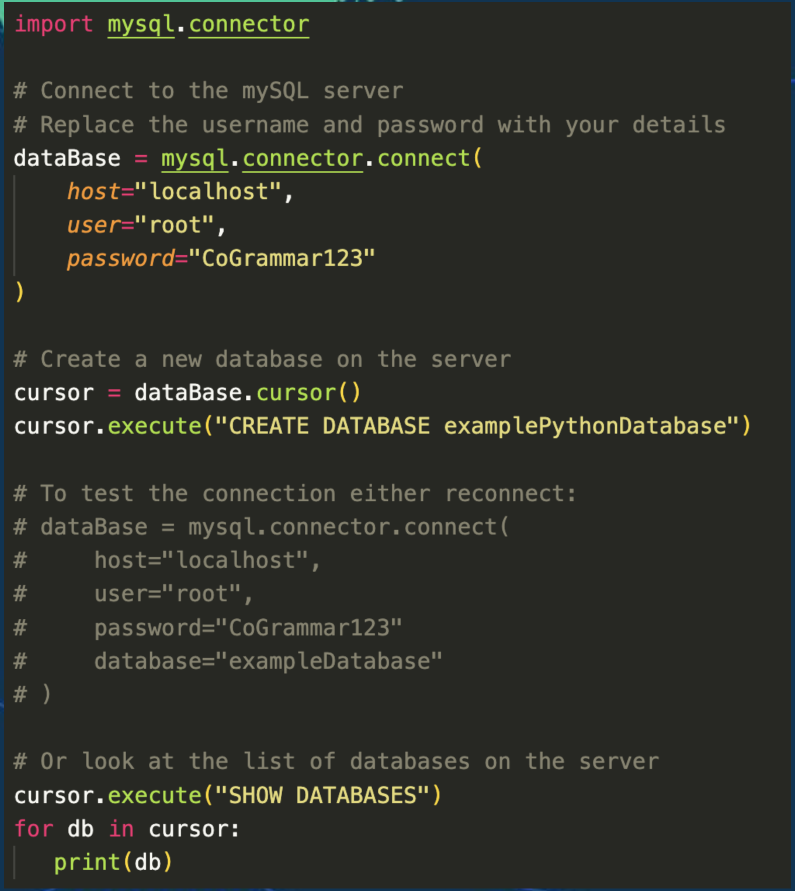
Date Type	Description	Allowed Range of Values
Date and Time Data Types		
DATE	Date	1 Jan 1000 to 31 Dec 9999
TIME	Time	– 838:59:59 to 838:59:59
String Data Types		
VARCHAR(n)	Variable-length string of up to n characters.	0 – 65535 characters
TEXT	Normal-sized text field	0 – 65535 characters

CoGrammar

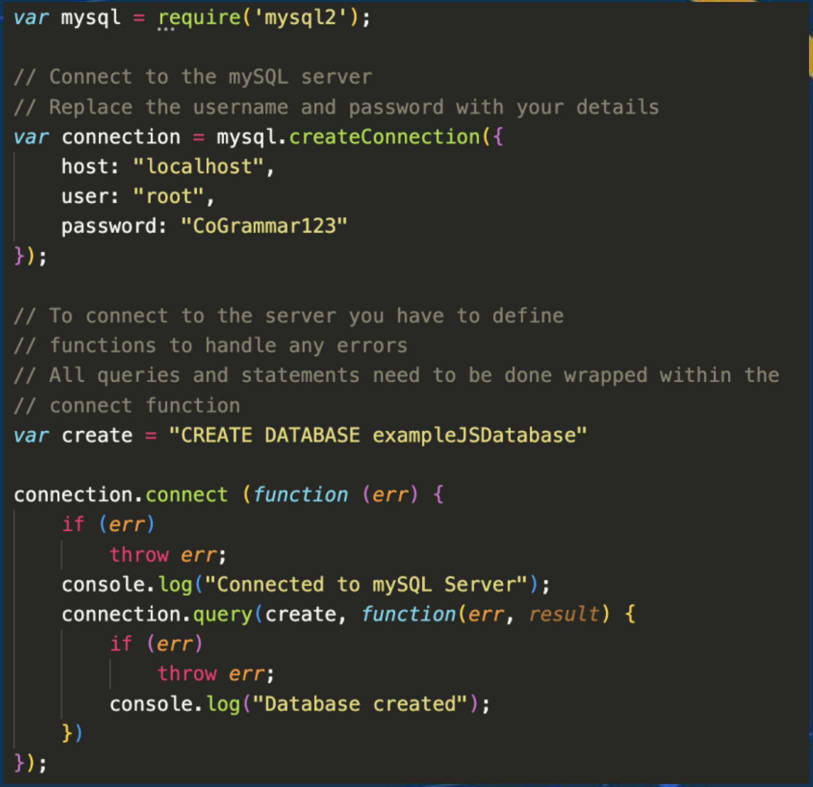
mySQL in Python and JS

1. Install mySQL on your device using the installer. Take note of your username and password.
2. Install the mySQL Driver to access MySQL through Python or JavaScript. Run the following command on the terminal:

```
# Python
pip install mysql-connector-python

# JavaScript
npm install mysql2
```



```python
import mysql.connector

# Connect to the mySQL server
# Replace the username and password with your details
dataBase = mysql.connector.connect(
    host="localhost",
    user="root",
    password="CoGrammar123"
)

# Create a new database on the server
cursor = dataBase.cursor()
cursor.execute("CREATE DATABASE examplePythonDatabase")

# To test the connection either reconnect:
# dataBase = mysql.connector.connect(
#     host="localhost",
#     user="root",
#     password="CoGrammar123"
#     database="exampleDatabase"
# )

# Or look at the list of databases on the server
cursor.execute("SHOW DATABASES")
for db in cursor:
  print(db)
```



```javascript
var mysql = require('mysql2');

// Connect to the mySQL server
// Replace the username and password with your details
var connection = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "CoGrammar123"
});

// To connect to the server you have to define
// functions to handle any errors
// All queries and statements need to be done wrapped within the
// connect function
var create = "CREATE DATABASE exampleJSDatabase"

connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(create, function(err, result) {
        if (err)
            throw err;
        console.log("Database created");
    })
});
```

CoGrammar

CRUD Operations

Create, Read, Update and Delete

- ❖ These are the 4 basic operations which act as the **foundation** of any computer programming language.
- ❖ We need to understand CRUD in SQL to interact with databases.

1. **Create:** To add or insert data into the SQL tables.
   a. **CREATE** TableName (ColumnName1 Datatype, …);

   b. **INSERT INTO** TableName (ColumnName1, …) **VALUES** (Value 1, … Value N), … , (Value 1, … Value N);

2. **Read:** To retrieve or fetch data from the SQL tables.
   a. **SELECT** * **FROM** TableName **WHERE** Condition;

CoGrammar

CRUD Operations

3. **Update:** To make updates to the records in the SQL tables.

    a. **UPDATE** TableName **SET** ColumnName = Value **WHERE** Condition;

4. **Delete:** To remove or delete records from the SQL tables.

    a. **DELETE FROM** TableName **WHERE** Condition;

CREATE

- **SYNTAX:**

CREATE TABLE tableName (column1 datatype constraint, column2 datatype constraint, ... , columnN datatype constraint)

- **Constraints (these are not compulsory):**
  - NOT NULL - column cannot have a NULL value
  - UNIQUE - all values in a column are different
  - PRIMARY KEY - combines NOT NULL and UNIQUE
  - FOREIGN KEY - prevents actions that would destroy table links
  - CHECK -
  - DEFAULT - sets default value if none is specified
  - AUTO_INCREMENT - field is automatically given a new value

CoGrammar

```python
# Create the Table
createTable = """CREATE TABLE crew (crewID INT AUTO_INCREMENT PRIMARY KEY,
firstName VARCHAR(255), lastName VARCHAR(255), roomNumber INT)"""
cursor.execute(createTable)
```

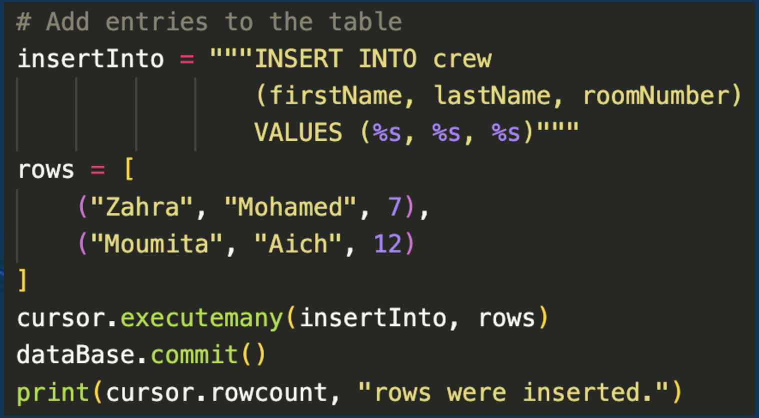
```javascript
// Create the Table
let createTable = `CREATE TABLE crew (crewID INT AUTO_INCREMENT PRIMARY KEY,
    firstName VARCHAR(255), lastName VARCHAR(255), roomNumber INT)`
connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(createTable, function(err, result) {
        if (err)
            throw err;
        console.log("Table created");
    })
});
```

CoGrammar

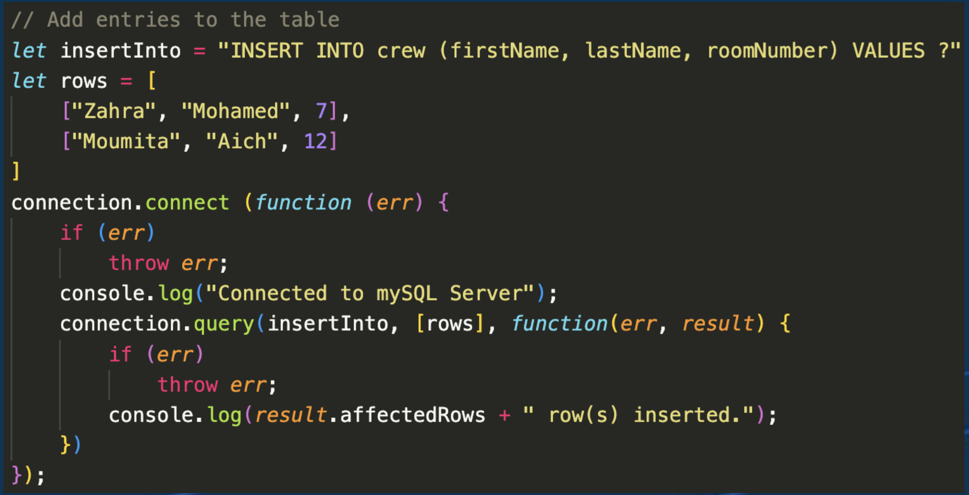
INSERT INTO

❖ **SYNTAX:**

INSERT INTO tableName (column1, column2, … , columnN) **VALUES** [
(value1, value2, …, valueN), … , (value1, value2, …, valueN)]



```python
# Add entries to the table
insertInto = """INSERT INTO crew
                (firstName, lastName, roomNumber)
                VALUES (%s, %s, %s)"""
rows = [
    ("Zahra", "Mohamed", 7),
    ("Moumita", "Aich", 12)
]
cursor.executemany(insertInto, rows)
dataBase.commit()
print(cursor.rowcount, "rows were inserted.")
```



```javascript
// Add entries to the table
let insertInto = "INSERT INTO crew (firstName, lastName, roomNumber) VALUES ?"
let rows = [
    ["Zahra", "Mohamed", 7],
    ["Moumita", "Aich", 12]
]
connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(insertInto, [rows], function(err, result) {
        if (err)
            throw err;
        console.log(result.affectedRows + " row(s) inserted.");
    })
});
```

CoGrammar

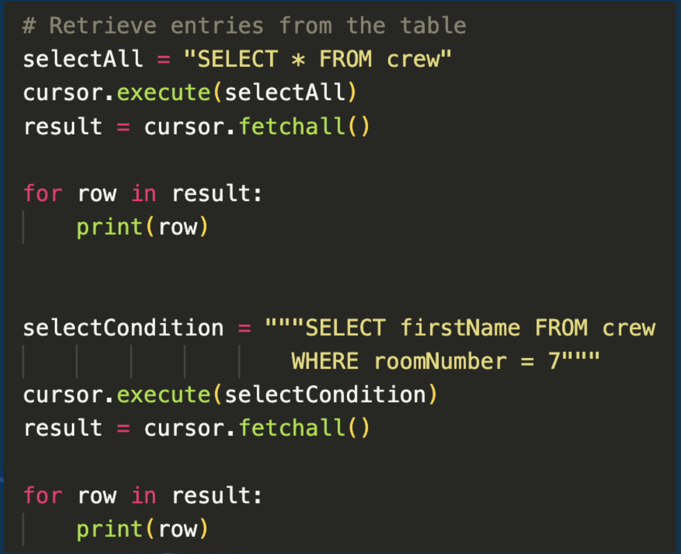
SELECT

- **SYNTAX:**

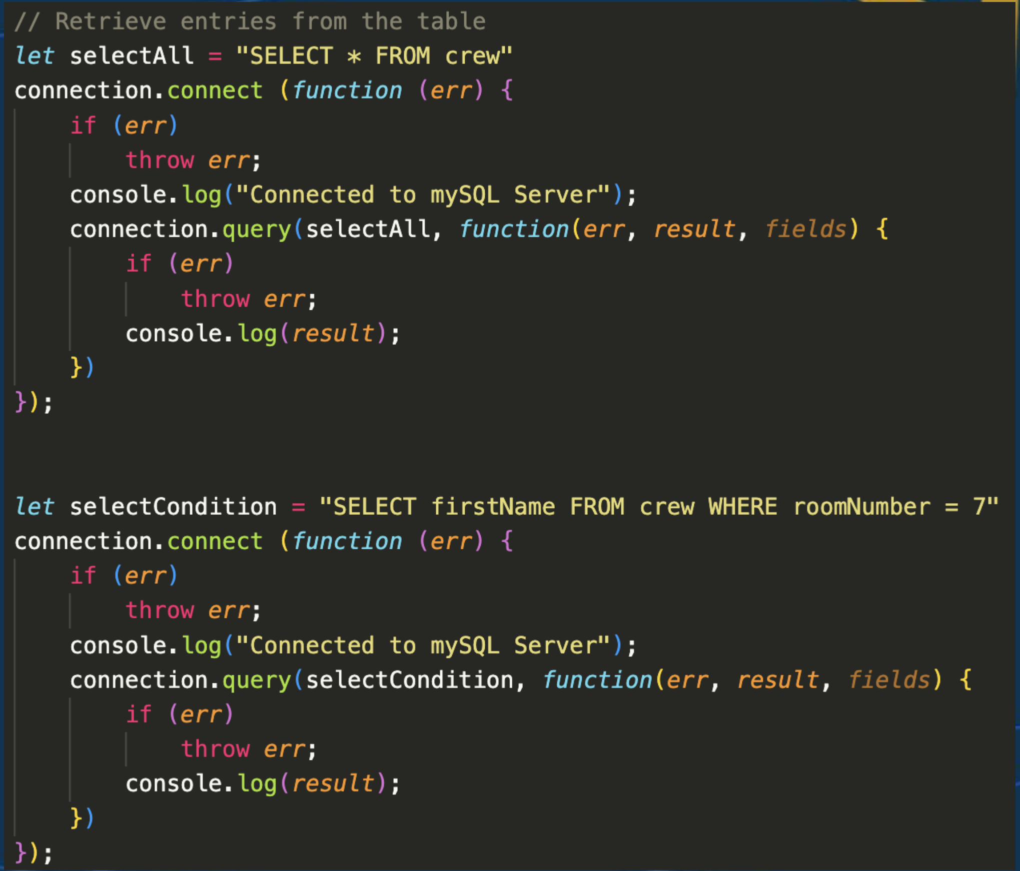
  SELECT column1, column2, … , columnN) **FROM** tableName **WHERE** Condition (**GROUP BY** column **HAVING** filter **ORDER BY** column)

- ***** can be used when selecting all columns.
- **Group By:** Groups rows with the same values, which allows us to perform data aggregation with aggregate functions like COUNT(), MAX(), MIN() SUM() AVG()
- **Having:** Allows us to set conditions when we use aggregate functions
- **Order By:** Allows us to sort by specific columns in ASC or DESC order

CoGrammar



```python
# Retrieve entries from the table
selectAll = "SELECT * FROM crew"
cursor.execute(selectAll)
result = cursor.fetchall()

for row in result:
    print(row)


selectCondition = """SELECT firstName FROM crew
                   WHERE roomNumber = 7"""
cursor.execute(selectCondition)
result = cursor.fetchall()

for row in result:
    print(row)
```



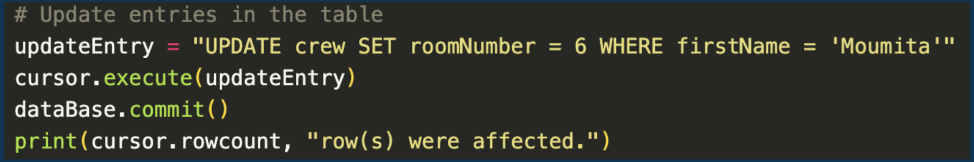
```javascript
// Retrieve entries from the table
let selectAll = "SELECT * FROM crew"
connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(selectAll, function(err, result, fields) {
        if (err)
            throw err;
        console.log(result);
    })
});


let selectCondition = "SELECT firstName FROM crew WHERE roomNumber = 7"
connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(selectCondition, function(err, result, fields) {
        if (err)
            throw err;
        console.log(result);
    })
});
```

CoGrammar

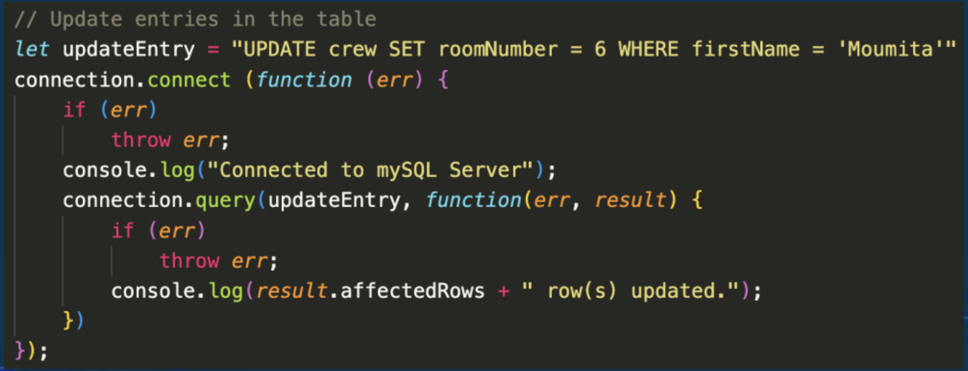
UPDATE

- **SYNTAX:**

UPDATE tableName **SET** column1 = value1, ... columnN = valueN **WHERE** Condition



```
# Update entries in the table
updateEntry = "UPDATE crew SET roomNumber = 6 WHERE firstName = 'Moumita'"
cursor.execute(updateEntry)
dataBase.commit()
print(cursor.rowcount, "row(s) were affected.")
```



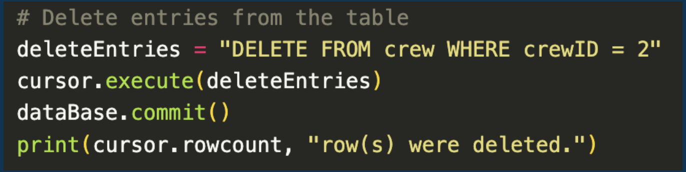
```
// Update entries in the table
let updateEntry = "UPDATE crew SET roomNumber = 6 WHERE firstName = 'Moumita'"
connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(updateEntry, function(err, result) {
        if (err)
            throw err;
        console.log(result.affectedRows + " row(s) updated.");
    })
});
```

CoGrammar

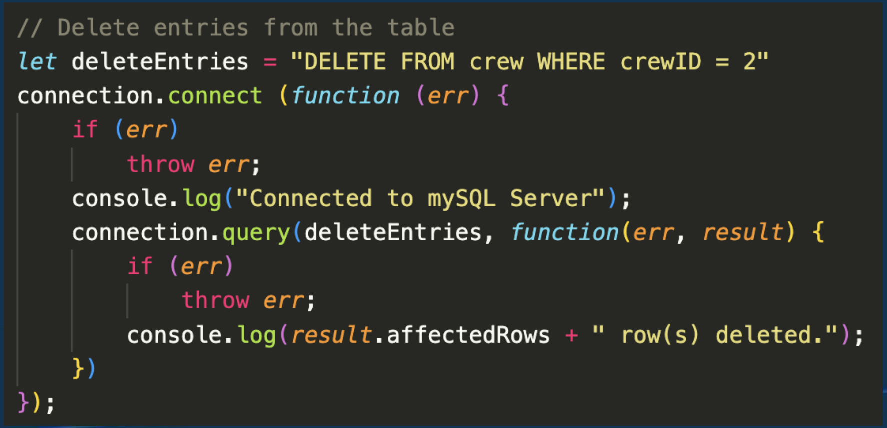
DELETE

❖ **SYNTAX:**

DELETE FROM tableName **WHERE** Condition



```python
# Delete entries from the table
deleteEntries = "DELETE FROM crew WHERE crewID = 2"
cursor.execute(deleteEntries)
dataBase.commit()
print(cursor.rowcount, "row(s) were deleted.")
```



```javascript
// Delete entries from the table
let deleteEntries = "DELETE FROM crew WHERE crewID = 2"
connection.connect (function (err) {
    if (err)
        throw err;
    console.log("Connected to mySQL Server");
    connection.query(deleteEntries, function(err, result) {
        if (err)
            throw err;
        console.log(result.affectedRows + " row(s) deleted.");
    })
});
```

CoGrammar

Which of the following is an advantage of using a relational database?

A. Flexibility in data storage
B. High scalability for large datasets
C. Data consistency and integrity
D. Low cost of implementation

Correct Answer: C

- ❖ Relational Databases have many advantages but the main one is **data consistency and integrity**, this is due to the validations and rules defined for each RDMS.
- ❖ **Non-relational databases** are more commonly known for their **flexibility** specifically when it comes to the way data is stored.
- ❖ Relational databases are only **vertically scalable** but are not flexible enough to be horizontally scalable.
- ❖ Cost refers here to cost in resources (space and time) but this is not an advantage for RDMSs.

CoGrammar

What is the purpose of the GROUP BY clause in SQL?

A. Filter rows based on a specified condition
B. Sort result set in ascending/descending order
C. Group rows that have the same values into summary rows
D. Perform arithmetic operations on numeric columns

CoGrammar

Correct Answer: C

- **Group By:** Groups rows with the same values, which allows us to perform data aggregation with aggregate functions like COUNT(), MAX(), MIN() SUM() AVG()
- We use HAVING or WHERE to filter rows
- We use ORDER BY to sort the result set

Portfolio Assignment: SE

Library Management System

Objective: Develop a library management system using SQL and Python which allows library staff to interact with the system to log loans, returns and new books being added to the system.

CoGrammar

Portfolio Assignment: SE

Requirements:

- ➢ Develop the application in Python, using libraries like mysql for SQL connections.
- ➢ Design a database schema for a library management system, including tables for books, members, and loans.
- ➢ Implement SQL queries to manage books, members, and loans.
- ➢ Write SQL queries to search for books, track loans, and manage library inventory.
- ➢ Implement a command-line or GUI interface for library staff to interact with the system.
- ➢ Include a README file that explains how to use the application, with examples of different scenarios and how to interpret output.

CoGrammar

Portfolio Assignment: DS

Analyzing Customer Purchase Data

Objective: Apply SQL skills to analyse customer purchase data from an e-commerce database, focusing on CRUD operations, data manipulation, and database interactions within Python applications. Use SQL queries to calculate relevant metrics based on the data like the total revenue and average order value.

CoGrammar

Portfolio Assignment: DS

Requirements:

- ➢Use Python's mysql module to connect to the database and execute SQL queries.
- ➢Set up a database with tables for customers, products, and orders.
- ➢Populate the database with sample data.
- ➢Implement SQL queries to create, read, update, and delete records in the database.
- ➢Write Python code to generate reports based on the analysis. This can include graphs and other visualisations as well.
- ➢Write SQL queries to calculate total revenue, average order value, and other relevant metrics.
- ➢Provide a README file detailing the tool's functionality, how to run it, and examples of output given sample datasets.

CoGrammar

Portfolio Assignment: WD

Building an Online Marketplace

Objective: Create an online marketplace that interacts with a database using SQL and JavaScript. Include tables for products, sellers, and orders. Develop an attractive frontend using HTML for buyers to browse through products and place orders.

CoGrammar

CoGrammar

Q & A SECTION

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

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar