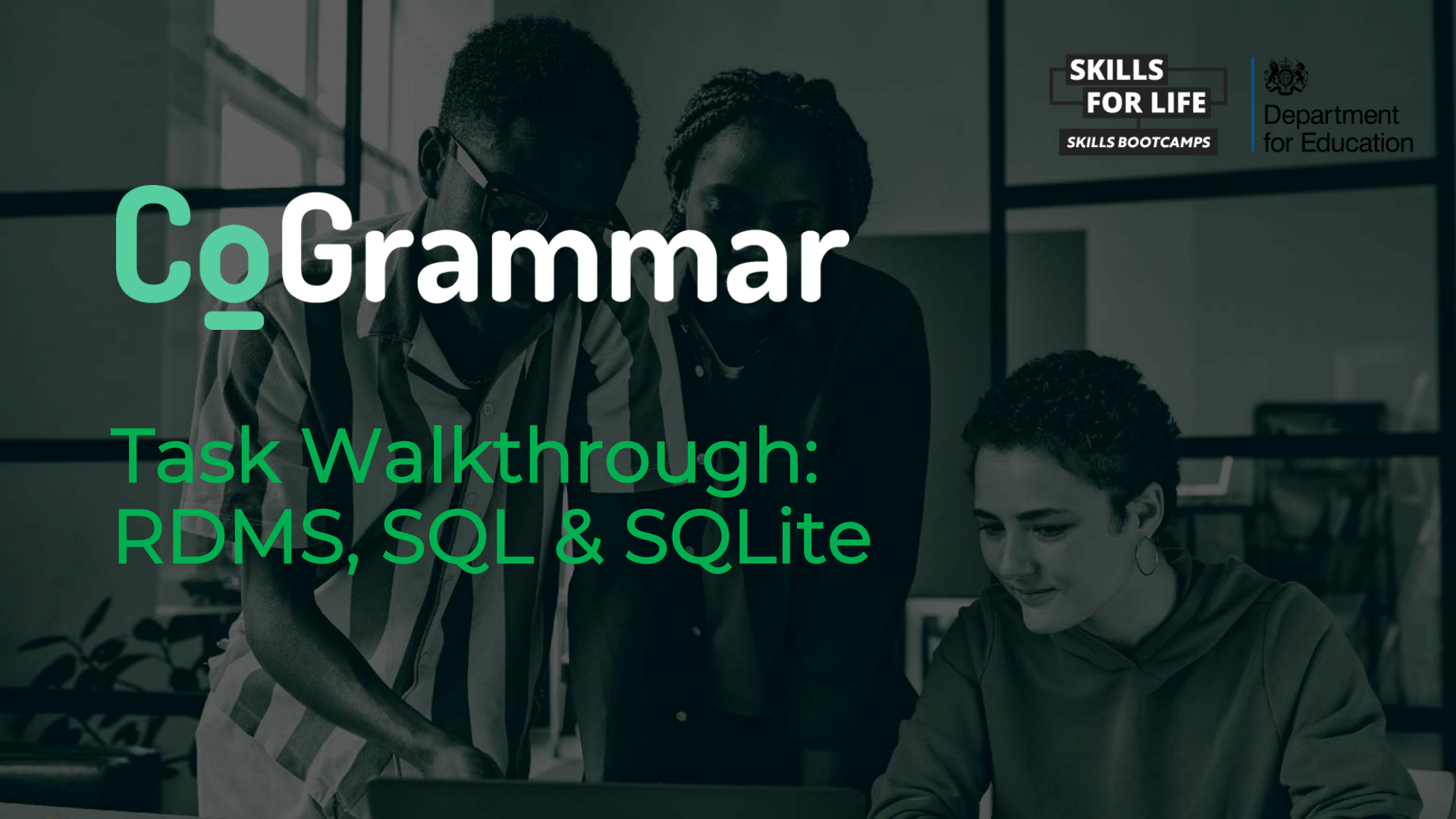
Welcome to this CoGrammar

Task Walkthroughs: Task 21, 22 and 23

The session will start shortly...

Questions? Drop them in the chat.



Task Walkthrough: RDMS, SQL & SQLite

Software Engineering Session 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** throughout this session, should you wish to ask any follow-up questions.
- 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

Software Engineering Session 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

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Learning Outcomes

- Describe the concept of a database.
- Learn basic database terminology and concepts (tables, columns, rows, keys, etc.).
- Explain the basic concepts of SQL.
- Implement Python scripts to interact with SQL databases.
- Transfer learnings to complete the RDMS, SQL and SQLite tasks.

Task Walkthrough: Relational Databases

CoGrammar

What is a Database?

- A database is a structured collection of data organised for easy access, retrieval, and management.
- Picture a well-organised library, but with information you can search and access in seconds.

Advantages of Databases

- Databases offer many benefits:
  - Organisation: Keeps data organised and easy to find.
  - Efficiency: Saves time and effort compared to manual data management.
  - Accuracy: Reduces errors and inconsistencies in data.
  - Sharing: Allows multiple users to access and share data securely.

Database Toolbox

- Databases are like powerful digital toolboxes for storing and managing information.
- Let's explore the essential components that make them work:

- Schema
- Tables
- Columns
- Rows
- Data Types
- Keys
- Relationships
- View
- Index
- Join
- CRUD

Speaking the Database Language

- Clear and consistent naming conventions are crucial for databases.
- Use descriptive and easy-to-understand names:
  - customer_name is better than cust_nm
  - order_date is clearer than ord_dt
- Consistency is key: choose a convention (e.g., lowercase_with_underscores) and stick to it.

Task Walkthrough: Relational Databases

CoGrammar

Task 21

CoGrammar

Auto-graded task

Answer the following questions:

1. What is normalisation?
2. When is a table in 1NF?
3. When is a table in 2NF?
4. When is a table in 3NF?
5. Using the `INVOICE` table given below, draw its dependency diagram and identify all dependencies (including transitive and partial dependencies). You can assume that the table does not contain any repeating groups and that an invoice number references more than one product. Hint: This table uses a composite primary key.

INV_NUM	PROD_NUM	SALE_DATE	PROD_LABEL	VEND_CODE	VEND_NAME	QUANT_SOLD	PROD_PRICE
211347	AA-E3522QW	15-Jan-2018	Rotary sander	211	NeverFail, Inc.	1	$34.46
211347	QD-300932X	15-Jan-2018	0.25-in. Drill bit	211	NeverFail, Inc.	8	$2.73
211347	RU-995748G	15-Jan-2018	Band saw	309	BeGood, Inc.	1	$31.59
211348	AA-E3522QW	15-Jan-2018	Rotary sander	211	NeverFail, Inc.	2	$34.46
211349	GH-778345P	16-Jan-2018	Power drill	157	ToughGo, Inc.	1	$69.32

6. Using the answer to the above question, remove all partial dependencies and draw the new dependency diagrams.
7. Using the answer to the above question, remove all transitive dependencies and draw the new dependency diagrams.

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

CoGrammar

Task Walkthrough: SQL

CoGrammar

What is SQL?

- SQL stands for Structured Query Language
- SQL is a database language that is composed of commands that enable users to:
  - create databases or table structures,
  - perform various types of data manipulation and data administration as well as
  - query the database to extract useful information.

Aspects of SQL?

- Data Definition Language (DDL):
  - Defines databases
  - Defines views
  - Defines access rights
- Data Manipulation Language (DML):
  - INSERT
  - UPDATE
  - DELETE
  - SELECT

SQL: Important Keywords

- **CREATE TABLE :** Creates a new table
- **NOT NULL :** Ensures that a column doesn't contain null values
- **UNIQUE :** Ensures that there are no repetitions
- **PRIMARY KEY :** Defines a primary key
- **FOREIGN KEY :** Defines a foreign key
- **DROP TABLE :** Deletes a table entirely

Task 22

CoGrammar

Auto-graded task

1. Go to the **DB Fiddle Online SQL Editor**. This is where you can write and test your SQL code. Once you are happy with your code, paste it into a text file and save the file in your task folder as **Student.txt**.
2. Write the SQL code to create a table called `Student`. The table structure is summarised in the table below.

   Note that `STU_NUM` must be set up as the primary key.

Attribute Name	Data Type
STU_NUM	CHAR(6)
STU_SNAME	VARCHAR(15)
STU_FNAME	VARCHAR(15)
STU_INITIAL	CHAR(1)
STU_STARTDATE	DATE
COURSE_CODE	CHAR(3)
PROJ_NUM	INT(2)

3. After you have created the table, write the SQL code to enter the following rows of data into the table as below:

STU_NUM	STU_SNAME	STU_FNAME	STU_INITIAL	STU_STARTDATE	COURSE_CODE	PROJ_NUM
01	Snow	Jon	E	2014-04-05	201	6
02	Stark	Arya	C	2017-07-12	305	11
03	Lannister	Jamie	C	2012-09-05	101	2
04	Lannister	Cercei	J	2012-09-05	101	2
05	Greyjoy	Theon	I	2015-12-09	402	14
06	Tyrell	Margaery	Y	2017-07-12	305	10
07	Baratheon	Tommen	R	2019-06-13	201	5

4. Write the SQL code to return all records which have a `COURSE_CODE` of 305.
5. Write the SQL code to change the course code to 304 for the person whose student number is 07.
6. Write the SQL code to delete the row of the person named Jamie Lannister, who started on 5 September 2012, whose course code is 101 and project number is 2. Use logical operators to include all of the information given in this problem.
7. Write the SQL code to change the `PROJ_NUM` to 14 for all those students who started before 1 January 2016 and whose course code is at least 201.
8. Write the SQL code that will delete the `Student` table entirely. Hint: Use `DROP TABLE`.

Task Walkthrough: SQLite

CoGrammar

Database Interaction: SQLite

- SQLite is a lightweight, self-contained SQL database engine that requires minimal setup and configuration.
- It is often used for smaller-scale projects or applications where simplicity and ease of use are prioritised.

SQLite: Key Features

- **Zero Configuration:** SQLite databases are self-contained.
- **Single File:** The entire database is stored in a single file.
- **SQL Support:** SQLite supports standard SQL syntax.

SQLite Syntax

```
import sqlite3

db = sqlite3.connect('data/student_db')
cursor = db.cursor()

cursor.execute('''
    CREATE TABLE student(id INTEGER PRIMARY KEY, name TEXT,
                         grade INTEGER)
''')

db.commit()
```

Basic SQLite Syntax

```
cursor.execute('''INSERT INTO student(name, grade)
                  VALUES(?,?)''', (name1,grade1))
db.commit()

students_ = [(name1,grade1),(name2,grade2),(name3,grade3)]
cursor.executemany(''' INSERT INTO student(name, grade) VALUES(?,?)''',
students_)
db.commit()
```

Task 23

CoGrammar

Auto-graded task

1. Create a Python file called **database_manip.py**.
2. Write the code to do the following tasks:
    - Create a table called `python_programming`.
    - Insert the following new rows into the `python_programming` table:

id	name	grade
55	Carl Davis	61
66	Dennis Fredrickson	88
77	Jane Richards	78
12	Peyton Sawyer	45
2	Lucas Brooke	99

    - Select all records with a `grade` between 60 and 80.
    - Change Carl Davis's `grade` to 65.
    - Delete Dennis Fredrickson's row.
    - Change the `grade` of all students with an `id` greater than 55 to 80.

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar