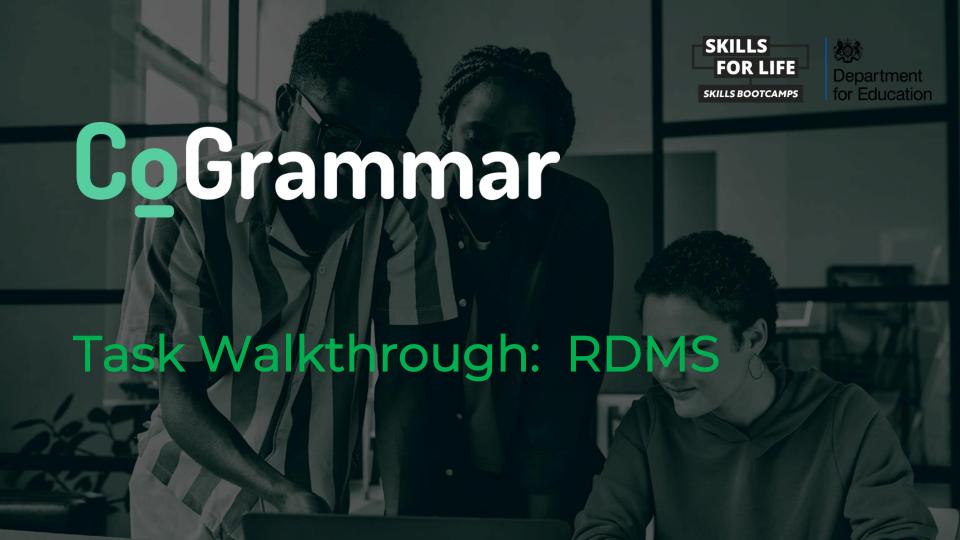
# Welcome to this CoGrammar Task Walkthroughs: Task 21

The session will start shortly...

Questions? Drop them in the chat.







#### 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: <u>Questions</u>

#### 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: <u>Feedback on Lectures</u>

#### 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



Rafig Manan

Scan to report a safeguarding concern



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



Ronald Munodawafa





## **Learning Outcomes**

- Describe the concept of a database.
- Learn basic database terminology and concepts (tables, columns, rows, keys, etc.).
- Explain the importance of database normalisation and apply it to simple database designs.
- Transfer learnings to complete the RDMS tasks.





## 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:
  - o Organisation: Keeps data organised and easy to find.
  - Efficiency: Saves time and effort compared to manual data management.
  - o 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
- Columns
- Rows
- Tables Data Types View

- Relationships
- Keys 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.



## Database Fundamentals





#### **Tables**

**Single type** of data within a **table**, such as a person's name or age, and is **organised vertically** within the table structure. (e.g., "Customer Name," "Product Price").

	vehide_id	customer_name	customer_contact	sale_amount	sale_location
١	1	Kwame Mensah	+225 01 23 45 67	25000.00	Abidjan
	2	Laurent Dubois	+33 6 12 34 56 78	30000.00	Paris
	3	Anna Müller	+49 151 12345678	28000.00	Berlin
	4	Sofia Mabunda	+258 82 123 45671	22000.00	Maputo
	5	Raj Patel	+91 98765 43210	35000.00	Mumbai



#### **Columns**

Structured collection of data organised into rows and columns, where each row represents a unique record and each column represents a different attribute or piece of information. (e.g., "Customers," "Products").

	vehicle_id	customer_name	customer_contact	sale_amount	sale_location
•	1	Kwame Mensan	+225 01 23 45 67	25000.00	Abidjan
	2	Laurent Dubois	+33 6 12 34 56 78	30000.00	Paris
	3	Anna Müller	+49 151 12345678	28000.00	Berlin
	4	Sofia Mabunda	+258 82 123 45671	22000.00	Maputo
	5	Raj Patel	+91 98765 43210	35000.00	Mumbai



#### **Rows**

Also known as a record, represents a single instance of data within a table, containing specific values corresponding to each column of the table. (e.g., a customer's name and address).

	vehide id	customer_name	customer_contact	sale amount	sale_location
•	1	Kwame Mensah	+225 01 23 45 67	25000.00	Abidjan
	2	Laurent Dubois	+33 6 12 34 56 78	30000.00	Paris
	3	Anna Müller	+49 151 12345678	28000.00	Berlin
		Sofia Mabanda	1250 02 125 15071	22000.00	Mapato
	5	Raj Patel	+91 98765 43210	35000.00	Mumbai



#### **Data Types**

Specifies the kind of data a column can hold, such as text (VARCHAR, TEXT), numbers (INT, DECIMAL), dates (DATE, TIMESTAMP), or binary data (BIT), ensuring consistency and facilitating efficient storage and retrieval of information.

	vehide_id	customer_name	customer_contact	sale_amount	sale_location
١	1	Kwame Mensah	+225 01 23 45 67	25000.00	Abidjan
	2	Laurent Dubois	+33 6 12 34 56 78	30000.00	Paris
	3	Anna Müller	+49 151 12345678	28000.00	Berlin
	4	Sofia Mabunda	+258 82 123 45671	22000.00	Maputo
	5	Raj Patel	+91 98765 43210	35000.00	Mumbai



## Keys, Relationships

#### **Primary Keys**

A unique identifier for each record within that table, ensuring that each row is uniquely identifiable. It serves as a reference point for other tables and is used to enforce data integrity and facilitate efficient data retrieval.

	vehide_id	customer_name	customer_contact	sale_amount	sale_location
١	1	Kwame Mensah	+225 01 23 45 67	25000.00	Abidjan
	2	Laurent Dubois	+33 6 12 34 56 78	30000.00	Paris
	3	Anna Müller	+49 151 12345678	28000.00	Berlin
	4	Sofia Mabunda	+258 82 123 45671	22000.00	Maputo
	5	Raj Patel	+91 98765 43210	35000.00	Mumbai



## Keys, Relationships

#### Relationships

Foreign keys in a database are columns that establish a relationship between tables by referencing the primary key of another table.

		Vehicle	∍Sales				Sales	Staff
	sale_id	vehicle_id	customer_name	salesman_id		salesman_id	sale_name	sale_location
•	1	1	Kwame Mensah	1	▶	1	John Smith	New York
	2	2	Laurent Dubois	2	1	2	Mary Jones	Los Angeles
	3	3	Anna Müller	3		3	David Lee	Chicago
	4	4	Sofia Mabunda	4		4	Elizabeth Brown	Miami
	5	5	Raj Patel	5		5	Villiam Miller	San Francisco



## Database Normalisation





### Normalisation

- Imagine storing a customer's address multiple times in a database instead of maintaining it in a separate table, that will lead to duplicate information.
- Data normalisation involves organising data in a database to minimise redundancy and dependency.
- Redundancy: Repeated data across tables, which can lead to:
  - Errors: Updating one value in multiple places can lead to inconsistencies.
  - Wasted Storage: Duplicate data takes up unnecessary space.
  - Inefficiency: Queries become slower with redundant data to search through.



## Normalisation Forms: Step-by-Step Approach

Normalisation Form	Description
1NF (First Normal Form)	Eliminates repeating groups of data within a table. Each record (row) should be unique and identifiable by a primary key.
2NF (Second Normal Form)	Meets 1NF requirements and eliminates partial dependencies. All non-key attributes must depend on the entire primary key, not just a part of it.
3NF (Third Normal Form)	Meets 2NF requirements and eliminates transitive dependencies. No non-key attribute should depend on another non-key attribute.



#### Unnormalised

EMPLOYEE	JOB	STATE_CODE	HOME_STATE
E001, Alice, J01	Chef	26	Michigan
E001, Alice, J02	Waiter	26	Michigan
E002, Bob, J02	Waiter	56	Wyoming



#### 1NF

EMPLOYEE_ID	NAME	JOB_CODE	JOB	STATE_CODE	HOME_STATE
E001	Alice	J01	Chef	26	Michigan
E001	Alice	J02	Waiter	26	Michigan
E002	Bob	J02	Waiter	56	Wyoming



#### 2 NF

#### roles table

EMPLOYEE_I D	JOB_COD E
E001	J01
E001	J02
E002	J02
E002	J03
E003	J01

#### employees table

EMPL OYEE _ID	NAME	STAT E_CO DE	HOME _STA TE
E001	Alice	26	Michig an
E002	Bob	56	Wyomi ng
E003	Alice	56	Wyomi na

#### jobs table

JOB_CODE	JOB
J01	Chef
J02	Waiter
J03	Bartender



3 NF

#### employees table

EMPLOYEE_ID	NAME	STATE_COD E
E001	Alice	26
E002	Bob	56
E003	Alice	56

#### states table

STATE_CODE	HOME_STATE		
26	Michigan		
56	Wyoming		







#### Auto-graded task

Answer the following questions:

- What is normalisation?
- 2. When is a table in 1NF?
- 3. When is a table in 2NF?
- 4. When is a table in 3NF?
- 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

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



## Questions and Answers





Thank you for attending





