

**Data Technician**

|  |
| --- |
|  |

|  |
| --- |
| Name: |
| Course Date: |
|  |

**Table of contents**

[Day 1: Task 1 3](#_Toc106466110)

[Day 1: Task 2 3](#_Toc1673105856)

[Day 3: Task 1 4](#_Toc1682646377)

[Day 4: Task 1: Written 6](#_Toc135095808)

[Day 4: Task 2: SQL Practical 9](#_Toc1228298484)

[Course Notes 19](#_Toc631049525)

[Additional Information 19](#_Toc1612679949)

# Day 1: Task 1

Please research and complete the below questions relating to key concepts of databases.

|  |  |
| --- | --- |
| What is a primary key? | A primary keyis a field (or a combination of fields) in a database table that uniquely identifies each record (row) in that table. |
| How does this differ from a secondary key? | A secondary key (or alternate key) is another field that could also uniquely identify a record but is not chosen as the primary key. A table can have many secondary keys, and they are often used for searching or indexing |
| How are primary and foreign keys related? | A primary key is a unique identifier for records in a table, while a foreign key is a field in another table that points back to that primary key. Together, they create a link between the two tables and ensure that the data stays consistent. For example, in a customer and orders system, the CustomerID is the primary key in the Customers table, and the same CustomerID appears as a foreign key in the Orders table to show which customer placed each order. |
| Provide a real-world example of a one-to-one relationship | A good real-world example of a one-to-one relationship is between a **person and their passport**. Each person can only have one passport, and each passport belongs to only one person. For instance, a Person table may store personal details, while a Passport table stores passport information, linked by the person’s ID. This ensures that every person is tied to exactly one passport, and vice versa. |
| Provide a real-world example of a one-to-many relationship | A classic real-world example of a one-to-many relationship is between a **customer and their orders**. One customer can place many orders, but each order belongs to only one customer. For example, a customers table may store customer details, while an Orders table records all purchases, with each order linked back to a single customer. |
| Provide a real-world example of a many-to-many relationship | A real-world example of a many-to-many relationship is between **students and courses**. A single student can enroll in multiple courses, and each course can have many students enrolled. For instance, Alice might take both Math and History, while the Math course could include Alice, Ben, and Chloe. This shows how the relationship flows in both directions. |

# Day 1: Task 2

Please research and complete the below questions relating to key concepts of databases.

|  |  |
| --- | --- |
| What is the difference between a relational and non-relational database? | A relational database stores data in tables with rows and columns, using primary and foreign keys to define relationships between them. It follows a fixed schema and is queried with SQL.  A non-relational database stores data in flexible formats like documents, key-value pairs, or graphs, without requiring a strict schema, making it better for unstructured or rapidly changing data, |
| What type of data would benefit off the non-relational model?  Why? | Non-relational databases are best for unstructured or semi-structured datasuch as social media posts, sensor data, product catalogues, or multimedia files.  They work well because they allow flexible data models, can scale easily across large volumes of data, and don’t require a fixed schema. This makes them ideal for situations where the data format changes often or grows very quickly. |

# Day 3: Task 1

Please research the below ‘JOIN’ types, explain what they are and provide an example of the types of data it would be used on.

|  |  |
| --- | --- |
| Self-join | 🔹 **What it is:**  A self-join is when a table is joined to itself. It’s useful when you want to compare rows within the same table.  🔹 **Example use case:**  In an **employees** table, where each employee has a ManagerID referencing another employee in the same table. |
| Right join | **2. Right Join** 🔹 **What it is:**  Returns **all rows from the right table** and the matched rows from the left table. If no match, NULLs appear for the left table’s columns.  🔹 **Example use case:**  When you want all records from a reference table even if no match exists in the main table.  🔹 **Example data:**   * **Orders table:** OrderID | CustomerID * **Customers table:** CustomerID | CustomerName |
| Full join | 🔹 **What it is:**  Returns **all rows from both tables**, matching where possible, and filling in NULLs where no match exists.  🔹 **Example use case:**  When comparing two datasets to see all records and where they overlap.  🔹 **Example data:**   * **OldStudents table:** StudentID | Name * **NewStudents table:** StudentID | Name |
| Inner join | 🔹 **What it is:**  Returns only rows that have a **matching value in both tables**.  🔹 **Example use case:**  Most common join — used when you only want related data.  🔹 **Example data:**   * **Orders:** OrderID | CustomerID * **Customers:** CustomerID | CustomerName |
| Cross join | 🔹 **What it is:**  Returns the **Cartesian product** — every row from the first table combined with every row from the second table.  🔹 **Example use case:**  Useful when you want all combinations, e.g. menu items with all drink options.  🔹 **Example data:**   * **Meals:** MealID | MealName * **Drinks:** DrinkID | DrinkName |
| Left join | 🔹 **What it is:**  Returns **all rows from the left table** and the matching rows from the right table. If no match, NULLs appear for the right table’s columns.  🔹 **Example use case:**  When you want everything from the main table, even if no match exists in the secondary table.  🔹 **Example data:**   * **Customers:** CustomerID | CustomerName * **Orders:** OrderID | CustomerID |

# Day 4: Task 1: Written

In your groups, discuss and complete the below activity. You can either nominate one writer or split the elements between you. Everyone however must have the completed work below:

*Imagine you have been hired by a small retail business that wants to streamline its operations by creating a new database system. This database will be used to manage inventory, sales, and customer information. The business is a small corner shop that sells a range of groceries and domestic products. It might help to picture your local convenience store and think of what they sell. They also have a loyalty program, which you will need to consider when deciding what tables to create.*

*Write a 500-word essay explaining the steps you would take to set up and create this database. Your essay should cover the following points:*

1. ***Understanding the Business Requirements****:*
   1. *What kind of data will the database need to store?*
   2. *Who will be the users of the database, and what will they need to accomplish?*
2. ***Designing the Database Schema****:*
   1. *How would you structure the database tables to efficiently store inventory, sales, and customer information?*
   2. *What relationships between tables are necessary (e.g., how sales relate to inventory and customers)?*
3. ***Implementing the Database****:*
   1. *What SQL commands would you use to create the database and its tables?*
   2. *Provide examples of SQL statements for creating tables and defining relationships between them.*
4. ***Populating the Database****:*
   1. *How would you input initial data into the database? Give examples of SQL INSERT statements.*
5. ***Maintaining the Database****:*
   1. *What measures would you take to ensure the database remains accurate and up to date?*
   2. *How would you handle backups and data security?*

*Your essay should include specific examples of SQL commands and explain why each step is necessary for creating a functional and efficient database for the retail business.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Please write your 500-word essay here | ***Understanding the Business Requirements:***  *The database will store core business data, including product and stock information, sales transactions, and customer details to support loyalty schemes. Together, this enables effective stock management, performance monitoring, evidence-based decision making and loyalty rewards.*  *Users will include:*  *Sales staff to record transactions and update stock levels.*  *Shop managers to analyses sales trends and customer loyalty data.*  *The owner to make sure they are happy with the running of the shop.*  *IT staff to ensure that the database has no errors, the right people do have access and update access when staff change.*  *Collectively, users will rely on the database to keep stock up to date and relevant, reward customer loyalty, make data driven decisions and ensure data security.*  ***Designing the Database Schema:***  **Products Table**  Columns: Product ID (PK), Product Name, Category, Price, Stock Level  **Customers Table**  Columns: Customer ID (PK), Name, Email, Loyalty Points  **Sales Table**  Columns: Sale ID (PK), Product ID (FK), Customer ID (FK), Date, Quantity  **Relationships Between Tables**  - Products → Sales: One-to-many (a product can appear in many sales).  - Customers → Sales: One-to-many (a customer can make many purchases).  These relationships ensure that sales are properly linked to both products and customers.  ER Diagram   |  |  |  |  | | --- | --- | --- | --- | | *TABLE* | *COLUMN* | *DATA TYPE* | *NOTES* | | *ProductID* | *ProductID* | *INT* | Surrogate key; numeric IDs are efficient for indexing and joins. | |  | *ProductName* | *VARCHAR(100)* | *Text for Product Name* | |  | *Category* | *VARCHAR(50)* | *Text Describing Product Type* | |  | *Price* | *DEcimal(6,2)* | *For Precise* | |  | *Stocklevel* | *INT* | *Integer* | | *Customers* | *CustomerID* | *INT* | *Surrogate key for uniqueness; easier than using name/email as this can be changed overtime in some cases* | |  | *Name* | *VARCHAR(100)* | *Customer name* | |  | *Email* | *VARCHAR(100)* | *TEXT* | |  | *LoyalityPoints* | *INT* | *Numeric Value for rewards* | | *Sales* | *SalesID* | *INT* | *Surragatekey* | |  | *ProductID* | *INT* | *Foreign key for transaction* | |  | *CustomerID* | *INT* | *Foriegn Key linking customers table* | |  | *Date* | *DATE* | *Transaction date* | |  | *Quantity* | *INT* | *No .of items sold* |  Database Schema Type The design uses a Star Schema:   * 1. Central fact table: Sales (records transactions).   2. Dimension tables: Products, Customers (describe entities).   3. Advantage: Optimized for querying sales, customer analysis, and reporting.  Key Selection: Surrogate vs Natural Key Surrogate Key: Used for ProductID, CustomerID, SaleID.  Numeric IDs (INT) are compact, fast for indexing, and stable over time.  Avoids problems with natural keys  ***Implementing the Database Schema***  *To implement the database, we can use tools such as Microsoft Access or MySQL. The process begins by creating a new database and opening a table where all information will be stored. Next, column headers are added to represent the details required, such as Product Name, Price, and Stock Level. These columns ensure the data is organized and easy to manage. Once the structure is ready, rows of sample data can be added, for example, Milk priced at £1.20 with 50 units in stock or Bread priced at £1.00 with 30 units available.*  *For example:*  *CREATE DATABASE RetailStore;*  *USE RetailStore;*  *CREATE TABLE Products (*  *ProductID INT PRIMARY KEY AUTO\_INCREMENT,*  *ProductName VARCHAR(50),* *Category VARCHAR(50)*  *Price DECIMAL(5,2),*  *StockLevel INT*  *);*  *CREATE TABLE Customers (* *CustomerID INT PRIMARY KEY,* *Name VARCHAR(50),* *Email VARCHAR(100),* *LoyaltyPoints INT* *);*  *CREATE TABLE Sales (* *SaleID INT PRIMARY KEY,* *ProductID INT,* *CustomerID INT,* *Date DATE,* *Quantity INT* *);*  *I would then link tables with Foreign Key commands, so sales connect to customers and products.*   1. ***4. Populating the Database***   To populate this database, the data can be entered manually or imported from excel files.  In SQL, the INSERT operation can be used to insert values into the fields.    INSERT INTO Products (ProductName, Category, Price, StockLevel)  VALUES (“Milk”, “Dairy”, 1.50, 10)  (“Bread”, “Bakery” 1.00, 20);    INSERT INTO Customers (CustomerID, Name, Email, LoyaltyPoints)  VALUES (12345678, “Jane Doe”, [janedoe@example.com](mailto:janedoe@example.com), 50);  INSERT INTO Sales (SalesID, ProductID, CustomerID, Date, Quantity) VALUES (12345, 123456, 1234567, 25/09/2025, 10)   1. To import data from excel, first you would need to export the excel data as .CSV file. In the SQL Workbench, a new schema would need to be created. The data can then be imported as a table by locating the file path, or simply by dragging and dropping the spreadsheet.   ***5. Maintaining the Database***  To ensure accuracy, the database should be updated regularly (e.g. shop staff reducing stock after each sale) and monitored for errors (e.g. as duplicates or missing entries). Large weekly backups should be created to prevent large data loss and differential backups should be made daily to prevent smaller data loss. Data security can be maintained by IT staff restricting access: staff can record sales, while managers can update product details or prices and when a staff member leaves the business, they no longer have access. These measures keep the database reliable, current, and secure.  UPDATE Products SET StockLevel = StockLevel - 2 WHERE ProductName = 'Milk';    Heres the link to th shared document if this is hard to view  [Day 4 TASK 1.docx](https://b2wcompletetraining057-my.sharepoint.com/:w:/g/personal/saharshjakkula_bootcamp_justit_co_uk/EcgB3VnAZ91ImdIi6q4C674BSz7vWxals-ZEIm925214mA?e=8eAYEs) |

# Day 4: Task 2: SQL Practical

In your groups, work together to answer the below questions. It may be of benefit if one of you shares your screen with the group and as a team answer / take screen shots from there.

**Setting up the database:**

1. **Download world\_db(1)**
2. **Follow each step to create your database**

**For each question I would like to see both the syntax used and the output.**

1. **Count Cities in USA:** *Scenario:* You've been tasked with conducting a demographic analysis of cities in the United States. Your first step is to determine the total number of cities within the country to provide a baseline for further analysis.

|  |
| --- |
|  |

1. **Country with Highest Life Expectancy:** *Scenario:* As part of a global health initiative, you've been assigned to identify the country with the highest life expectancy. This information will be crucial for prioritising healthcare resources and interventions.

|  |
| --- |
|  |

1. **"New Year Promotion: Featuring Cities with 'New :** *Scenario:* In anticipation of the upcoming New Year, your travel agency is gearing up for a special promotion featuring cities with names including the word 'New'. You're tasked with swiftly compiling a list of all cities from around the world. This curated selection will be essential in creating promotional materials and enticing travellers with exciting destinations to kick off the New Year in style.

|  |
| --- |
|  |

1. **Display Columns with Limit (First 10 Rows):** *Scenario:* You're tasked with providing a brief overview of the most populous cities in the world. To keep the report concise, you're instructed to list only the first 10 cities by population from the database.

|  |
| --- |
|  |

1. **Cities with Population Larger than 2,000,000:** *Scenario:* A real estate developer is interested in cities with substantial population sizes for potential investment opportunities. You're tasked with identifying cities from the database with populations exceeding 2 million to focus their research efforts.

|  |
| --- |
|  |

1. **Cities Beginning with 'Be' Prefix:** *Scenario:* A travel blogger is planning a series of articles featuring cities with unique names. You're tasked with compiling a list of cities from the database that start with the prefix 'Be' to assist in the blogger's content creation process.

|  |
| --- |
|  |

1. **Cities with Population Between 500,000-1,000,000:** *Scenario:* An urban planning committee needs to identify mid-sized cities suitable for infrastructure development projects. You're tasked with identifying cities with populations ranging between 500,000 and 1 million to inform their decision-making process.

|  |
| --- |
|  |

1. **Display Cities Sorted by Name in Ascending Order:** *Scenario:* A geography teacher is preparing a lesson on alphabetical order using city names. You're tasked with providing a sorted list of cities from the database in ascending order by name to support the lesson plan.

|  |
| --- |
|  |

1. **Most Populated City:** *Scenario:* A real estate investment firm is interested in cities with significant population densities for potential development projects. You're tasked with identifying the most populated city from the database to guide their investment decisions and strategic planning.

|  |
| --- |
|  |

1. **City Name Frequency Analysis: Supporting Geography Education** *Scenario*: In a geography class, students are learning about the distribution of city names around the world. The teacher, in preparation for a lesson on city name frequencies, wants to provide students with a list of unique city names sorted alphabetically, along with their respective counts of occurrences in the database. You're tasked with this sorted list to support the geography teacher.

|  |
| --- |
|  |

1. **City with the Lowest Population:** *Scenario:* A census bureau is conducting an analysis of urban population distribution. You're tasked with identifying the city with the lowest population from the database to provide a comprehensive overview of demographic trends.

|  |
| --- |
|  |

1. **Country with Largest Population:** *Scenario:* A global economic research institute requires data on countries with the largest populations for a comprehensive analysis. You're tasked with identifying the country with the highest population from the database to provide valuable insights into demographic trends.

|  |
| --- |
|  |

1. **Capital of Spain:** *Scenario:* A travel agency is organising tours across Europe and needs accurate information on capital cities. You're tasked with identifying the capital of Spain from the database to ensure itinerary accuracy and provide travellers with essential destination information.

|  |
| --- |
|  |

1. **Country with Shortest Life Expectancy:** *Scenario:* A healthcare foundation is conducting research on global health indicators. You're tasked with identifying the country with the highest life expectancy from the database to inform their efforts in improving healthcare systems and policies.

|  |
| --- |
| This question mentions both shortest and highest? |

1. **Cities in Europe:** *Scenario:* A European cultural exchange program is seeking to connect students with cities across the continent. You're tasked with compiling a list of cities located in Europe from the database to facilitate program planning and student engagement.

|  |
| --- |
|  |

1. **Average Population by Country:** *Scenario:* A demographic research team is conducting a comparative analysis of population distributions across countries. You're tasked with calculating the average population for each country from the database to provide valuable insights into global population trends.

|  |
| --- |
|  |

1. **Capital Cities Population Comparison:** *Scenario:* A statistical analysis firm is examining population distributions between capital cities worldwide. You're tasked with comparing the populations of capital cities from different countries to identify trends and patterns in urban demographics.

|  |
| --- |
|  |

1. **Countries with Low Population Density:** *Scenario:* An agricultural research institute is studying countries with low population densities for potential agricultural development projects. You're tasked with identifying countries with sparse populations from the database to support the institute's research efforts.

|  |
| --- |
|  |

1. **Cities with High GDP per Capita:** *Scenario:* An economic consulting firm is analysing cities with high GDP per capita for investment opportunities. You're tasked with identifying cities with above-average GDP per capita from the database to assist the firm in identifying potential investment destinations.

|  |
| --- |
|  |

1. **Display Columns with Limit (Rows 31-40):** *Scenario:* A market research firm requires detailed information on cities beyond the top rankings for a comprehensive analysis. You're tasked with providing data on cities ranked between 31st and 40th by population to ensure a thorough understanding of urban demographics.

|  |
| --- |
|  |

|  |
| --- |
| **Course Notes** |

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:

|  |
| --- |
| **Additional Information** |

We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

**END OF WORKBOOK**

**Please check through your work thoroughly before submitting and update the table of contents if required.**

**Please send your completed work booklet to your trainer.**