

**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 key** is a column (or a combination of columns) in a table that uniquely identifies each row. It ensures no duplicates and no null values. Every table in a relational database should have a primary key.  Example: *StudentID* in a Students table. |
| How does this differ from a secondary key? | A **secondary key** (sometimes called an *alternate key*) is any attribute that can also be used to identify records, but it is not the main (primary) identifier.   * **Primary key** = the main unique identifier. * **Secondary key** = another candidate that could have been chosen as the primary key.   Example: In a Students table, *StudentID* might be the primary key, while *Email* could be a secondary key since it also uniquely identifies a student. |
| How are primary and foreign keys related? | * A **primary key** uniquely identifies records in one table. * A **foreign key** is a field in another table that refers to the primary key.  This link creates relationships between tables, enforcing **referential integrity**.   Example: *CustomerID* is the primary key in the Customers table, and it 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 | **Person -> Passport**  Each person has exactly one passport, and each passport belongs to exactly one person.  **Country -> Capital City**  each country has one capital, and each capital belongs to one country.  **Person -> Birth Certificate**  each person has exactly one birth certificate, and each certificate belongs to only one person. |
| Provide a real-world example of a one-to-many relationship | **Teacher -> Students**  One teacher can teach many students, but each student is assigned to one main teacher.  **Author -> Books**  one author can write many books, but each book is written by one main author.  **Bank -> Branches**  one bank has many branches, but each branch belongs to one bank. |
| Provide a real-world example of a many-to-many relationship | **Students <-> Courses**  A student can enroll in many courses, and each course can have many students.  This is usually implemented with a *junction/bridge table* (e.g., Enrollments).  **Movies ↔ Actors**  one movie has many actors, and one actor can be in many movies.  **Products ↔ Orders**  one product can appear in many orders, and one order can contain many product |

# 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? | **Relational Database (RDBMS):**   * Organises data into **tables (rows and columns)** with a fixed schema. * Enforces **relationships** between tables using primary and foreign keys. * Supports **ACID transactions** (Atomicity, Consistency, Isolation, Durability) for reliability. * Query language is **SQL**, powerful for joins, aggregations, and analytics. * Best for structured, consistent, transactional data.   **Examples:** MySQL, PostgreSQL, SQL Server, Oracle.  **Non-relational Database (NoSQL):**   * Stores data in more **flexible formats**: documents (JSON), key-value pairs, wide-column stores, or graphs. * Schema is **dynamic** records can have different fields. * Designed for **horizontal scalability** (data sharded across many servers). * Prioritises speed, flexibility, and scale over strict ACID guarantees (though some support them). * Best for semi-structured, high-volume, or rapidly changing data.   **Examples:** MongoDB (document), Cassandra (wide-column), Redis (key-value), Neo4j (graph). |
| What type of data would benefit off the non-relational model?  Why? | * **Semi-structured or unstructured data**  Example: JSON documents, logs, user profiles, IoT sensor data. * **Rapidly evolving datasets**  Example: Applications where fields change often (e.g., adding new features to a mobile app). * **High-volume, high-velocity data**  Example: Clickstream data, social media feeds, event logs, or time-series metrics. * **Highly connected data**  Example: Social networks, fraud detection, recommendation engines (ideal for graph databases). * **Global-scale workloads**  Example: E-commerce platforms, gaming leaderboards, content delivery networks needing fast distributed access.  **Why?**  1. **Flexibility** – Non-relational databases allow new fields without redesigning schemas, making them ideal for fast-changing business needs. 2. **Scalability** – They scale horizontally across many servers, handling millions of users or events in real time. 3. **Performance** – By denormalising data, queries can return results faster (no complex joins). 4. **Data variety** – They support diverse formats (JSON, key-value, graph), which map naturally to how modern applications generate and consume data. |

# 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 | **Self-Join** **What it is:**  A table joins with itself. Useful for comparing rows within the same table.  **Use case example:**   * Employees table where each employee has a manager\_id that refers back to another row in the same table.  **Example query:**   SELECT e.employee\_id, e.name AS employee, m.name AS manager FROM Employees e JOIN Employees m ON e.manager\_id = m.employee\_id; |
| Right join | **Right Join** **What it is:**  Returns all rows from the **right** table, and matching rows from the left. If there’s no match, left side shows NULL.  **Use case example:**   * Show all orders (right table) whether or not they are linked to a customer (left table).  **Example query:**   SELECT c.first\_name, o.order\_id, o.amount FROM Customers c RIGHT JOIN Orders o ON c.customer\_id = o.customer\_id; |
| Full join | **Full Join (Full Outer Join)** **What it is:**  Returns **all rows from both tables**. Where matches exist, they’re combined; where not, unmatched side shows NULL.  **Use case example:**   * Compare Customers and Orders to find which customers placed orders, which didn’t, and if there are any orphan orders.  **Example query:**   SELECT c.first\_name, o.order\_id, o.amount FROM Customers c FULL OUTER JOIN Orders o ON c.customer\_id = o.customer\_id; |
| Inner join | **Inner Join** **What it is:**  Returns only the rows that match in both tables.  **Use case example:**   * Show only customers who have placed at least one order.  **Example query:**   SELECT c.first\_name, o.order\_id, o.amount FROM Customers c INNER JOIN Orders o ON c.customer\_id = o.customer\_id; |
| Cross join | **Cross Join** **What it is:**  Cartesian product: every row from one table joins with every row from another table.  **Use case example:**   * Generate all possible combinations of products and promotions.  **Example query:**   SELECT p.product\_name, pr.promo\_name FROM Products p CROSS JOIN Promotions pr; |
| Left join | **Left Join** **What it is:**  Returns all rows from the **left** table, and matches from the right. If no match, right side shows NULL.  **Use case example:**   * Show all customers, even if they haven’t placed an order.  **Example query:**   SELECT c.first\_name, o.order\_id, o.amount FROM Customers c LEFT JOIN Orders o ON c.customer\_id = o.customer\_id; |

# 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 | **Designing a Database for a Small Retail Business** A small corner shop, like many local convenience stores, needs a reliable way to manage its operations. The shop sells groceries and household items, processes daily sales, and runs a loyalty program to encourage repeat customers. Currently, much of this may be recorded manually or across separate systems, which makes it difficult to maintain accurate stock, track customer activity, or generate reports. A well-designed database can solve these challenges by integrating inventory, sales, and customer data into a single structured system. The following sections outline the process of setting up such a database, step by step. **Understanding the Business Requirements** The first step is to identify exactly what data the database must store and who will use it. The database needs to capture **products** (name, barcode, price, stock levels), **customers** (contact details, loyalty ID, points balance), **sales transactions** (time, payment method, totals, line items), and optionally **suppliers** (to support re-ordering). The users include shop staff, who require quick access at checkout to process sales, and managers, who need reliable reports on best-selling items, low stock, and customer loyalty trends. **Designing the Database Schema** For a shop of this size, a **normalized relational schema** is appropriate. It avoids duplication and supports fast lookups.   * **Products**: Primary key product\_id. Secondary key barcode to guarantee each product is unique and scannable. * **Customers**: Primary key customer\_id. Secondary keys loyalty\_number or email for unique identification. * **Sales**: Primary key sale\_id. Foreign key to customer\_id (nullable for walk-in customers). * **SaleItems**: Primary key sale\_item\_id. Foreign keys to sale\_id and product\_id. * **LoyaltyTx**: Primary key loyalty\_tx\_id. Foreign keys to customer\_id and optionally sale\_id. * **Suppliers**: Primary key supplier\_id. Secondary key supplier\_name to avoid duplicates.   Relationships include: Customers 1–M Sales, Sales 1–M SaleItems, and Products 1–M SaleItems. This structure supports loyalty tracking, stock management, and reporting. **Implementing the Database** The implementation uses SQL Data Definition Language (DDL). For example:  CREATE TABLE Products (  product\_id INT PRIMARY KEY AUTO\_INCREMENT,  barcode VARCHAR(32) UNIQUE,  name VARCHAR(100),  category VARCHAR(50),  unit\_price DECIMAL(10,2),  stock\_quantity INT DEFAULT 0 );  Similar commands define Customers, Sales, SaleItems, and LoyaltyTx, each with primary keys and foreign keys to maintain referential integrity. **Populating the Database** Initial stock and customer records are inserted with SQL INSERT statements. For example:  INSERT INTO Customers (loyalty\_number, first\_name, last\_name, email) VALUES ('L-0001', 'Alice', 'Smith', '[alice@example.com](mailto:alice@example.com)');  Transactions automatically add rows to **Sales** and **SaleItems**, with loyalty points recorded in **LoyaltyTx** **Maintaining the Database** To keep the database accurate, constraints (NOT NULL, UNIQUE, CHECK) ensure valid data entry. Indexes on keys like barcode and loyalty\_number speed up searches. Backups should be taken nightly with periodic incremental copies, and role-based security ensures staff only access what they need. Routine audits, such as checking stock against reorder levels, prevent discrepancies. **Conclusion** By carefully understanding the business, designing a schema with clear primary and secondary keys, implementing the structure in SQL, and planning for maintenance, the corner shop gains a robust system for managing inventory, customers, and sales. This improves efficiency, reduces errors, and enables data-driven decisions that support both daily operations and long-term growth. |

# 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.

|  |
| --- |
|  |

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.**