

**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 uniquely identifies each record in a database table. |
| How does this differ from a secondary key? | A secondary key provides an alternative way to locate records. However, it may not be unique or used as the primary key. It offers an additional method for searching of filtering data. |
| How are primary and foreign keys related? | A primary key becomes a foreign key in another table. It is used to establish a relationship between tables. |
| Provide a real-world example of a one-to-one relationship | A person and their passport.  Each person has only one passport, and each passport is issued to only one person. |
| Provide a real-world example of a one-to-many relationship | A mother and her children.  One mother can have multiple children, but each child has only one biological mother. |
| Provide a real-world example of a many-to-many relationship | Students and courses.  A single student can be enrolled in multiple courses.  A single course can have multiple students enrolled in it. |

# 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? | The key difference between relational and non-relational databases lies in how they store, organise, and manage data.  Relational Databases (SQL Databases): Store data in tables, which consist of rows and columns.  Non-relational Databases (NoSQL Databases): They do not use the traditional tabular structure. Instead, they store data in key-value pairs, similar to DynamoDB. Many databases serve different purposes; for example, MongoDB stores data in flexible, semi-structured formats like JSON. |
| What type of data would benefit off the non-relational model?  Why? | Non-relational databases are particularly well-suited for certain types of data and use cases.  **Types of data:**  **Unstructured and Semi-structured Data:**  Examples Include User-Generated content, such as social media posts, comments, reviews, messages, and articles.  **IoT sensor data:**  Examples**:** Reading from devices, time-series data.  **Multimedia:**  Example: Images, video, audio files.  **Why**:  **Flexible Schema:** Non-relational databases don’t need a predefined schema. This means I can store data as it arrives, even if different records have different fields or structures**.**  I can store data in its natural format. E.g. JSON documents in a document database.  Large volumes of data (Big Data)  Examples:  Web analytics data: Billions of page views, clicks, and user interactions.  Real-time data streams: Continuous influx of data from various sources.  **Why: Horizontal Scalability:** NoSQL databases are designed to scale horizontally by adding more servers to a cluster. It allows them to handle massive amounts of data and high traffic loads by distributing the data and workload across many machines. On the other hand, relational databases scale vertically. |

# 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.

In relational databases, JOIN clauses are used to combine rows from two or more tables based on a related column between them. This is essential for retrieving meaningful information when data is normalised (spread across multiple tables to reduce redundancy).

A group of circles with text

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| Self-join | A Self-Join is not a distinct join type, such as INNER or LEFT. Instead, it is a JOIN where a table joins to itself. I can use table aliases to treat the same table as two separate tables. This technique enables us to compare rows within the same table. |
| Right join | The Right Join returns all rows from the right table, along with the matching rows from the left table. If a row in the right table has no match, the columns from the left table will contain NULL values. |
| Full join | The FULL JOIN returns all rows from both the left and right tables. If there's no match, NULL values are filled in for the columns from the table that lacks a match. It's the combination of LEFT JOIN and RIGHT JOIN. |
| Inner join | The INNER JOIN returns only the rows that have matching values in both tables based on the join condition. If a row in one table doesn't have a matching row in the other table, it is excluded from the results. |
| Cross join | The CROSS JOIN produces the Cartesian product of two tables. This means it pairs each row from the first table with every row from the second table. There is no ON clause required for a CROSS JOIN.  **Example Use Case & Data Types:** Less common for general data retrieval, but useful for generating combinations or permutations. For instance, if you want to produce all possible pairings of employees with departments (without considering actual assignments), or all combinations of product sizes and colours. There are no specific data type requirements for the "join" itself, as it doesn't depend on a matching column, but the data in the tables can be of any type. |
| Left join | The LEFT JOIN returns all rows from the left table, and the matching rows from the right table. If there's no match for a row in the left table, the columns from the right table will have NULL values. |

# Day 4: Task 1: Written

In your groups, discuss and complete the activity below. 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 store inventory, sales, and customer information efficiently?*
   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 | **Setting Up a Retail Database System**  An efficient database system is vital for any modern business, even a small corner shop. As a newly appointed consultant for such a retail venture, I aim to design and implement a reliable database to streamline inventory, sales, and customer management, including a key loyalty programme. This essay details the systematic steps I would take to achieve this, from understanding core business needs to ensuring long-term data integrity.  **1. Understanding the Business Requirements**  The initial and most critical phase involves deeply understanding the business's operational needs. For a local convenience store, the database must primarily  store product information (e.g., item name, unit price, current stock levels, supplier)  sales transaction details (e.g., date, time, total amount, items purchased), and  customer data (e.g., name, contact information, loyalty program membership). The loyalty program specifically requires tracking points earned and redeemed.  The primary users of this database are the shop owner (who needs high-level sales reports, profit analysis, and stock overview), cashiers (for quick sales entry, price lookups, and basic stock checks), and possibly inventory staff (for receiving new stock and updating quantities). Their main tasks include processing sales efficiently, managing stock accurately, enrolling and tracking loyalty members, and producing various operational and financial reports. This clarity ensures the database is designed to support real-world workflows.  2**. Designing the Database Schema**  With requirements understood, the next step is to design the database schema, which defines the tables' structure and relationships. I would propose the following normalised tables to minimise data redundancy and ensure integrity:  A diagram of a retail inventory schema  AI-generated content may be incorrect.  **3. Implementing the Database**  Using SQL, I would then create the database and its tables.  CREATE DATABASE RetailDB;  USE RetailDB;  CREATE TABLE Suppliers (  SupplierID INT PRIMARY KEY AUTO\_INCREMENT,  SupplierName VARCHAR(50) NOT NULL,  ContactPerson VARCHAR(50),  Email VARCHAR(50) UNIQUE  );  CREATE TABLE Products (  ProductID INT PRIMARY KEY AUTO\_INCREMENT,  ProductName VARCHAR(100) NOT NULL,  Category VARCHAR(50)  StockLevel INT NOT NULL DEFAULT 0,  UnitPrice DECIMAL(10, 2) NOT NULL,  SupplierID INT,  FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)  );  CREATE TABLE Customers (  CustomerID INT PRIMARY KEY AUTO\_INCREMENT,  FullName VARCHAR(50) NOT NULL,  Email VARCHAR(100) UNIQUE,  PhoneNumber VARCHAR(20),  Membership BOOLEAN  LoyaltyPoints INT DEFAULT 0  );  CREATE TABLE Sales (  Sale\_ID INT PRIMARY KEY AUTO\_INCREMENT,  Product\_ID INT  Customer\_ID INT  Item VARCHAR(30)  Quantity INT  Total\_Amount DECIMAL(10, 2) NOT NULL,  Date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  );  CREATE TABLE SubCategory (  Category INT PRIMARY KEY AUTO\_INCREMENT,  Item INT NOT NULL,  UnitPrice DECIMAL(10, 2) NOT NULL,  Quantity INT NOT NULL,    );  These statements define the table structures, ensuring that PRIMARY KEY Constraints uniquely identify each record and FOREIGN KEY Constraints enforce referential integrity, preventing orphaned data and maintaining relationships.  4**. Populating the Database**  Initial data would be input using INSERT statements. It includes existing inventory, registered loyalty customers, and initial supplier details.  INSERT INTO Suppliers (SupplierName, ContactPerson, ContactEmail) VALUES  ('Dairy Delights Ltd', 'John Milk', 'john@dairy.com'),  ('Clean Home Supplies', 'Sarah Sparkle', 'sarah@cleanhome.com');  INSERT INTO Products (ProductName, UnitPrice, StockLevel, Supplier\_ID) VALUES  ('Milk (1L)', 1.20, 50, 1),  ('Bread (Loaf)', 1.00, 30, 1),  ('Washing Up Liquid', 2.50, 20, 2),  ('Chocolate Bar', 0.80, 100, 1);  These examples demonstrate how new data is added, reflecting the initial state of the business.  **5. Maintaining the Database**  Ongoing maintenance is crucial for maintaining the database's accuracy, security, and optimal performance.  **Accuracy and Updates**: Data accuracy would be ensured through input validation (e.g., using CHECK constraints for Current Stock to prevent negative values) and regular data audits. Staff training on correct data entry procedures is crucial. UPDATE statements would be used after sales or new deliveries for stock updates.  **Backups:** Automated daily backups of the entire database would be configured to a secure, off-site location (e.g., cloud storage). Weekly full and incremental daily backups would provide granular recovery options in case of data loss due to hardware failure, accidental deletion, or cyber-attack.  **Data Security:** Access control would be implemented by creating specific user roles (e.g., 'cashier', 'owner') with minimal necessary permissions (e.g., cashiers can INSERT into Sales and Sale Items, SELECT from Products, but not DELETE or DROP tables). Strong password policies would be enforced. While a third-party payment processor would ideally handle payment card data, sensitive customer information like loyalty points would be protected with appropriate encryption if stored directly.  Following these structured steps, the small retail business can transition from manual operations to an efficient, reliable, and scalable database system, enabling better decision-making and improved customer service. |

# Day 4: Task 2: SQL Practical

In your groups, work together to answer the questions below. 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.

|  |
| --- |
|  |

19.**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.

|  |
| --- |
|  |

**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:

|  |
| --- |
|  |

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.**