

**Data Technician**

|  |
| --- |
|  |

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

**Table of contents**

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

[Day 1: Task 2 4](#_Toc1673105856)

[Day 3: Task 1 6](#_Toc1682646377)

[Day 4: Task 1: Written 9](#_Toc135095808)

[Day 4: Task 2: SQL Practical 13](#_Toc1228298484)

[Course Notes 24](#_Toc631049525)

[Additional Information 25](#_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 unique entity identifier in a database table. It ensures that each row in the table is distinct and can be uniquely identified. It is important because:   * Ensures there are no duplicates or missing records. * Helps databases quickly locate and access records. * Acts as a reference point for establishing relationships between tables in a relational database. |
| How does this differ from a secondary key? | A primary key is the main identifier for records in a table, while a secondary key is used to retrieve data based on non-primary columns.  Secondary keys are columns or sets of columns that can also uniquely identify records in a table. A table can have multiple secondary keys. |
| How are primary and foreign keys related? | Primary keys serve as unique identifiers for each row in a database table. Foreign keys link data in one table to the data in another table. |
| Provide a real-world example of a one-to-one relationship | A one-to-one relationship ensures that each record in both tables corresponds to exactly one record in the other table. For example, in an organisation, each employee is assigned a unique employee ID card. An employee can have only one ID card. The relationship between employee and their unique employee ID card would be a one-to-one relationship. |
| Provide a real-world example of a one-to-many relationship | In a retail store, each customer can place multiple orders, but each order is placed by only one customer. This forms a one-to-many relationship between Customers and Orders. |
| Provide a real-world example of a many-to-many relationship | In a school, students can join multiple clubs, and each club can have multiple students as members. This creates a many-to-many relationship between Students and Clubs. |

# 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 and a non-relational database differ primarily in how they store and organise data.  For example, in relational Databases:   * Data is stored in tables with rows representing individual records and columns representing attributes. * They follow a predefines schema – which means the structure of the data is fixed and must be defined before inserting data. * It has strong data integrity – as data is enforced through constraints for example, primary keys, foreign keys and unique constraints to maintain relationships. * Scalability – It scales vertically – which means you only upgrade the hardware of a single server. * Query language – Uses structured query language (SQL) for defining and manipulating data. * Examples – MySQL, Oracle and SQL server.   In non-relational Databases:   * Structure - Data is stored in various forms, such as key-value pairs, documents, graphs, or wide-column stores, depending on the type of NoSQL database. * Schema - It has a flexible schema, allowing for changes in the data structure without requiring a predefined schema. * Data Integrity - Generally, more relaxed in terms of data consistency and integrity compared to relational databases, but can be tuned based on use cases * Scalability - Designed to scale horizontally (across multiple servers with partitions), which makes them more suitable for handling large-scale data. * Query Language - Does not use SQL. Instead, each NoSQL database has its own methods for data retrieval and manipulation. * Examples - MongoDB, Cassandra and Couchbase.   To summarise:  Relational databases are well-suited for applications where structured, consistent data and complex queries are needed.  Non-relational databases are more flexible and can handle unstructured or semi-structured data, often used for big data. |
| What type of data would benefit off the non-relational model?  Why? | Non-relational databases are ideal for data types that are unstructured, semi-structured, hierarchical, or rapidly evolving, as well as for high-volume, real-time data that requires scalability and flexible data models.  Unstructured Data - non-relational databases can store unstructured data without needing to define a rigid schema. For example, you can store a variety of data formats and a flexible way such as JSON documents that can evolve over time without requiring changes to a predefined schema making it easier to accommodate varying or unpredictable data types.  Semi-Structured Data – non relational databases like document stores are designed to handle semi structured data they allow for documents with varying structures which means that not all documents need to follow the same structure or schema. Disable X ability allows for easier handling of data that doesn't fit neatly into tables.  Real time data-non relational databases such as time series database or key value stores can handle high speed rights and retrievals of real time 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 | A self-join is essentially an inner join or outer join applied to the same table. The table is referenced twice, with different aliases to distinguish between the two instances.  Commonly used for hierarchical relationships (e.g., employees and managers). |
| Right join | Alright join you include all the data from the right table in the results even if there's no match in the left table. It includes only the matching rows from the left table. The non-matching rows from the left table are filled with null. |
| Full join | Full outer join includes all rows from both tables. Rows with no matches in the other table have null values for the columns of that table.  It is Useful for combining all data while identifying mismatches. |
| Inner join | Inner join retrieves rows that have matching values in both tables. If there is no match, those rows will be excluded from the results. |
| Cross join | Cross join is the combination every row from the first table with every row from the second table. No condition is required for the join for example, no ON clause. The result includes every possible combination of rows between the two tables.  It is commonly used for scenarios like creating combinations, testing or when explicitly required. |
| Left join | Left join retrieves all rows from the left table. It includes matching rows from the right table. The non-matching rows from the right table will be filled with null. |

# 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 | 1. **Understanding the Business Requirements**   The client would like to store 3 entities: inventory, sales and customer information. Therefore, we need to ensure all fields for these entities are stored accurately and efficiently. For example, the client might want to maintain records for the price and stock levels of various inventory items. They may want to keep track of the quantity of items sold, and the revenue generated from sales. They may also want to store customer information in the database, for example customer names, customer IDs, contact information, and loyalty points that the customers have earned through the loyalty program.  The users of the database will include business owners, shop floor employees, and the IT department. Business owners will need access in order to check records and modify entries. Shop floor employees, for example shelf stockers or cashiers, will need basic access to the database. Their actions would affect stock level, revenue, and loyalty points. For example, a cashier scanning product at the checkout would affect all three of the previously mentioned fields. The IT department would need full database access in order to maintain the database by cleaning data and ensuring that there are no leaks.  **2. Designing the Database Schema:**  Our database will contain three tables.  The first table is the Inventory table. This entity represents the items that the shop sells. The key attributes could include ProductID as a primary key (PK), ProductName, Category, Price and Quantity. For example:    The second table is the Customer table. This entity represents the company’s customers. The key attributes could include CustomerID as primary key (PK), CustomerName, CustomerLastName, CustomerEmail, CustomerPhoneNumber and CustomerAddress and LoyaltyPoints.  The last table is the Sales table. This entity represents the orders placed by customers. Key attributes could include TransactionID (PK), CustomerID (FK), referencing customer ProductID (FK), OrderDate and Revenue.    In designing these tables, we set the cardinalities between each entity. We can create relationships between them. For example, one customer can have multiple sales, but each sale is placed by only one customer. A sale can contain multiple inventories, and one inventory can be part of sales.    **3. Implementing the Database:**  We would use MySQL to create this database.  To create the database we could use the following statement:  CREATE DATABASE databasename;  To create tables we would use the CREATE TABLE statement. Below is the statement we would use to create the Inventory table.  CREATE TABLE Inventory (  ProductName varchar(255) NOT NULL,  ProductID varchar(20) NOT NULL,  Category varchar(20),  Price TINYINT,  Quantity SMALLINT,  PRIMARY KEY (ProductID) );  The Quantity and Price data types are set to integer and all others set to strings of differing lengths. We have used the NOT NULL command for ProductName and ProductID. The primary key for this table was set as ProductID, which can be used for defining relationships with other tables. Similar statements would be needed to create the other tables.  **4. Populating the Database:**  We could input the data manually into the MySQL database that we have created. Alternatively, we could use an insert statement to populate the database, for example:  INSERT INTO Inventory (ProductName, ProductID, ProductType)  VALUES ('Milk', ‘1’, 'Dairy');  We could also use an INSERT INTO SELECT statement in order to copy relevant data from one table into another:  INSERT INTO Inventory (CustomerId)  SELECT CustomerId  FROM Customers;  This would result in the customer ID information being copied from the Inventory table into the Customers table.  **5. Maintaining the Database:**  The database will require regular updates to maintain accuracy. For example, if 5 of 15 bottles of milk are sold, the record must reflect the remaining 10. Routine data quality audits would be conducted to identify and address inconsistencies, duplicates, or outdated records, maintaining the integrity of the database.  To secure the database, daily incremental and weekly full backups will be implemented. Access will be restricted to trusted staff using an access list and role-based control, ensuring GDPR compliance and reducing data breach risks. Activity logs will be used to monitor changes and detect any potential security breaches. Sensitive data will be encrypted to prevent unauthorised access. Parameterised queries will be employed to safeguard against SQL injection attacks. While customers currently have no way to input data directly into the database, this practice remains essential to uphold. |

# 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 Highest 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.**