

**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? | It’s an unique identifier for a particular record in a database table. It ensures that each row in the table is uniquely identifiable and prevents duplicate entries. |
| How does this differ from a secondary key? | A secondary key is an additional field used for searching or indexing but is not the main identifier. |
| How are primary and foreign keys related? | A Primary Key uniquely identifies a record in a table, while a Foreign Key is a field in another table that refers to the Primary Key, creating a relationship between the two tables. The Foreign Key ensures data consistency and enforces integrity by linking records across tables. |
| Provide a real-world example of a one-to-one relationship | A real-world example of a one-to-one relationship is a Person and National Insurance Number (NIN) system: Each person has only one NIN and each NIN belongs to only one person. |
| Provide a real-world example of a one-to-many relationship | A real-world example of a one-to-many relationship is a Teacher and Students system: One teacher can have many students.  Each student is assigned to only one teacher. |
| Provide a real-world example of a many-to-many relationship | A real-world example of a many-to-many relationship is a Students and Courses system, where students can enrol in multiple courses, and each course can have multiple students. |

# 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 organizes data into tables with rows and columns, following a strict structure known as a schema. These tables are linked through relationships, typically using primary and foreign keys. Data is accessed using SQL (Structured Query Language).  A non-relational database offers more flexibility in how data is stored. Instead of tables, data can be stored in formats like documents, key-value pairs, or graphs. These databases do not require a predefined schema, allowing for easier scalability and quick adaptation to changing data. Non-relational databases often use NoSQL (Not Only SQL) for querying, and they are ideal for applications with unstructured or semi-structured data, like social media platforms or IoT systems.  So, relational databases are well-suited for structured data with predefined relationships, while non-relational databases are better for unstructured, flexible, and scalable data storage. |
| What type of data would benefit off the non-relational model?  Why? | Non-relational databases are great for data that is unstructured or changes often, like social media posts, images, or sensor data. They allow for flexible data storage without needing a fixed structure, making it easier to adapt as data evolves. These databases are also designed to scale easily, handling large volumes of data, which is ideal for applications like real-time analytics, e-commerce, or IoT systems. |

# 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 a regular join, but the table is joined with itself  (example tested on programiz)  SELECT A.customer\_id AS Customer\_id1, B.customer\_id AS customer\_id2, A.Country  FROM Customers A, Customers B  WHERE A.customer\_id <> B.customer\_id  AND A.Country = B.Country  ORDER BY A.Country; |
| Right join | Returns all records from the right table, and the matched records from the left table.  (example tested on w3schools)  SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  FROM Orders  RIGHT JOIN Employees  ON Orders.EmployeeID = Employees.EmployeeID  ORDER BY Orders.OrderID; |
| Full join | Returns all records when there is a match in either left or right table  (example extracted from w3schools)  SELECT Customers.CustomerName, Orders.OrderID  FROM Customers  FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID  ORDER BY Customers.CustomerName; |
| Inner join | Returns records that have matching values in both tables  SELECT ProductID, ProductName, CategoryName  FROM Products  INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID; |
| Cross join | The CROSS JOIN keyword returns all records from both tables (table1 and table2).  SELECT Customers.CustomerName, Orders.OrderID FROM Customers CROSS JOIN Orders; |
| Left join | Returns all records from the left table, and the matched records from the right table  SELECT Customers.CustomerName, Orders.OrderID FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName; |

# 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 data that would need to be stored for this particular database would consist of a list of different categories such as: Inventory, sales transactions, suppliers, employees, customers. The data for inventory is essential because it must be updated regularly to ensure that items are available for sale or need to be restocked. This allows the shop to manage stock efficiently by controlling shortages, monitoring stock levels and disposing of any unwanted goods. The sales transaction data would be essential in tracking what has been sold, when it was sold and it can also mention who it was sold to. The sales data should record every sale made with the payment method and amount. There would also need information about employees about who is processing which sales. Customer information would be very important as it would show who our regular customers are and can include contact information for marketing purposes.*    *The data base users would essentially include managers, cashiers and the shop owners. The managers would analyse sales data and monitor stock levels. Their goal would be to improve business performance ensuring a smooth operation. The cashier’s job would be to process sales and ensure a healthy level of customer service. Shop owners tend to not involve themselves however if involved they would ensure business performance is strong and make strategic decisions for business growth.*   1. ***Designing the Database Schema****:*   *A diagram of a small corner shop  AI-generated content may be incorrect.*  *The database is structured with separate tables for products, customers, and sales, allowing for quick access and updates. For this shop, the Products table holds details like Product ID, Name, Price, and Stock Level. The Customers table stores Customer ID, Name, Email, and Loyalty Points. The Sales table links to both, recording each sale with Sale ID, Product ID, Customer ID, Date, and Quantity. These relationships help track inventory, update stock after sales, and keep customers engaged with the loyalty program. With this structure, the shop can quickly check stock levels, track customer purchases, and automatically update loyalty points, making daily operations more efficient.*   1. ***Implementing the Database****:*   ***Table 1 Products***    *CREATE TABLE Products (Productid INT PRIMARY KEY AUTO\_INCREMENT, Product\_NAME VARCHAR(30), Category VARCHAR(30), Price DECIMAL(15,2)  NOT NULL, Stock\_Level INT NOT NULL);*  ***Table 2 Customers***    *CREATE TABLE Customers (Customerid INT PRIMARY KEY AUTO\_INCREMENT, Name VARCHAR(30), Email VARCHAR(30), Bonus\_Points INT NOT NULL);*  ***Table 3 Sales***    *CREATE TABLE Sales (Salesid INT PRIMARY KEY, Productid INT NOT NULL, Customerid INT NOT NULL,Date DATE, Quantity INT NOT NULL, FOREIGN KEY (Productid) REFERENCES Products (Productid), FOREIGN KEY (customerid) REFERENCES Customers (Customerid));*   1. ***Populating the Database****:*   *INSERT INTO Products (Product\_Name,Category, Price, Stock\_Level) Values ('Milk','Dairy',1.50,10),('Eggs','Poultry',2.50,8),('Bread','Bakery',1.30,6);*  *INSERT INTO Customers (Name,Email, Bonus\_Points) Values ('Jane Doe','JaneDoe@example.com',50),('Calvin Klein','CalvinKlein@example.com',75),('John Smith','JohnSmith@example.com',70),('Trent Donald', 'Trentdonald@example.com', 60);*   1. ***Maintaining the Database****:*   *To maintain the database accurate, secure and efficient implement :*  *1. data validation*  *Examples: constraints on stock levels and unique customer emails*  *2. regular backups*  *Examples: using automated schedules. Exporting the data every 24hrs.*  *3. Ensure security*  *with user roles, encrypted sensitive data, and strong authentication. For example, only manager can update prices and only IT Team can delete or aggregated things.*  *4. Optimise performance with indexing on frequently queried columns (e.g., CustomerID in Sales).* |

# 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.

|  |
| --- |
| SELECT COUNT(\*) FROM city WHERE CountryCode = 'USA'; |

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.

|  |
| --- |
| SELECT Name, LifeExpectancy  FROM country  WHERE LifeExpectancy IS NOT NULL  ORDER BY LifeExpectancy DESC  LIMIT 1; |

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.

|  |
| --- |
| SELECT name  FROM city  WHERE name LIKE '%New%'; |

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.

|  |
| --- |
| SELECT name, population  FROM city  ORDER BY population DESC  LIMIT 10; |

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.

|  |
| --- |
| SELECT name FROM world.city  Where population>2000000; |

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.

|  |
| --- |
| SELECT name  FROM city  Where Name LIKE 'Be%'; |

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.

|  |
| --- |
| SELECT Name, Population  FROM city  WHERE Population BETWEEN 500000 AND 1000000  ORDER BY Population DESC; |

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.

|  |
| --- |
| SELECT Name, Population  FROM city  ORDER BY Name ASC; |

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.

|  |
| --- |
| SELECT Name, Population  FROM city  ORDER BY Population DESC  LIMIT 1; |

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.

|  |
| --- |
| SELECT name, COUNT(\*) AS name\_frequency  FROM city  GROUP BY name  ORDER BY name ASC; |

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.

|  |
| --- |
| SELECT name, population  FROM city  ORDER BY population ASC  LIMIT 1; |

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.

|  |
| --- |
| SELECT Name, Population  FROM country  ORDER BY Population DESC  LIMIT 1; |

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.

|  |
| --- |
| SELECT city.Name  FROM city  JOIN country ON city.ID = country.Capital  WHERE country.Name = 'Spain'; |

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.

|  |
| --- |
| SELECT Name, LifeExpectancy  FROM country  WHERE LifeExpectancy = (SELECT MAX(LifeExpectancy) FROM country); |

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.

|  |
| --- |
| SELECT  city.Name AS City,  city.CountryCode,  country.Name AS Country,  city.District,  city.Population  FROM city  JOIN country ON city.CountryCode = country.Code  WHERE country.Continent = 'Europe'  ORDER BY city.Population DESC; |

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.

|  |
| --- |
| SELECT countrycode , avg(population) from city group by countrycode ; |

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.

|  |
| --- |
| SELECT  country.Name AS Country,  city.Name AS Capital,  city.Population AS Population\_Capital  FROM country  JOIN city ON country.Capital = city.ID  ORDER BY city.Population DESC; |

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.

|  |
| --- |
| SELECT  Name AS Country,  Population,  SurfaceArea,  (Population / SurfaceArea) AS PopulationDensity  FROM country  ORDER BY PopulationDensity ASC; |

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.

|  |
| --- |
| (Used GNP for the scenario)  SELECT  city.Name AS City,  city.CountryCode,  country.Name AS Country,  city.Population,  (country.GNP / country.Population) AS GNP\_per\_Capita  FROM city  JOIN country ON city.CountryCode = country.Code  WHERE (country.GNP / country.Population) >  (SELECT AVG(country.GNP / country.Population) FROM country)  ORDER BY GNP\_per\_Capita DESC; |

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.

|  |
| --- |
| SELECT name, population  FROM city  ORDER BY population DESC  LIMIT 10 OFFSET 30; |

|  |
| --- |
| **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.**