

**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 table with a column or combination of columns that contain values that uniquely identify each row in a table. Enforces the entity integrity of the table. |
| How does this differ from a secondary key? | The primary key contains the unique values, and the secondary (foreign) key contains values that reference the primary key. A secondary key can be replicated whereas the primary key is unique. |
| How are primary and foreign keys related? | The primary key is the field that identifies each row. The foreign key links each of the unique references to another table. It works as a bridge based on primary keys across tables allowing for data to be related across them. |
| Provide a real-world example of a one-to-one relationship | Country and capital city. Can only be one capital city to each country making them unique. A person and their fingerprints. A person and they passport. |
| Provide a real-world example of a one-to-many relationship | A customer placing many orders. A salesperson deals across a timeframe. This is a one-way relationship and cannot be interchangeable. |
| Provide a real-world example of a many-to-many relationship | When multiple records in one table are linked to different records in another table. The same as the customer placing more than one order but there are orders placed by multiple customers. There can be multiples for either side of the values. This uses a junction table as an intermediary. |

# 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? | They differ in how they organise and store data. Relational databases use tabular structure with predefined relationships that are accessed using SQL. NoSQL uses flexible data models; key-value pairs, documents, graphs. These are suitable for whichever data structure they are handling. NoSQL prioritises performance and availability of just consistency. Relational databases are used for transactional systems and NoSQL is more for large-scale, real-time processing. |
| What type of data would benefit off the non-relational model?  Why? | Unstructured or semi-structured data would be best for the non-relational model because of the large data volumes which has a higher frequency of change and have diverse formats. This can be social media content, sensor data and multimedia files. |

# 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 | Join the table to itself for comparing rows within the same table. |
| Right join | Returns all records from the right table and matched records from the left table. |
| Full join | Returns all records when there is a match in either of the tables. This will return all rows as well as NULLS |
| Inner join | Returns records with matching values in both tables e.g. customer\_id from 1-”” even with different column names. |
| Cross join | Combine each row of the first table with each row of the second table. |
| Left join | Returns all records from the left table and matched records from the right table. |

# 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 | For the convenience store database, we would want to use a structured data as it conforms to a pre-defined schema or data model. We would use Textual, Numerical, Boolean and Date and Time data.   The users of the database would be business owners, till operators, any member of the business that is involved in marketing. We would also need access as data engineers so we can make updates or changes to the database.   The business owner would want to track their sales and inventory, keep up to date with revenue streams/profit margins. To be able to track KPI’s based on Footfall vs. Sales, they would also need to be able to identify products that are outliers to see what products under-performing or over-performing are. Finally, they would need to understand the data quickly and easily with visualisations.   A relational database with well-defined tables and relationships between tables. We would need Primary and foreign keys. Each table should have a primary key and we can use foreign keys to form relationships.   The tables would be structured with the following columns.   **Customer Table**  Customer\_ID, First Name, Last Name, Email, Phone Number, Address, City, County, Post Code, Loyalty Points.    Order\_ID, Customer\_ID, Order Date, Shipping Address, Total Amount, Order Status, Loyalty Points.  **Order Items Table.** Order\_Item\_ID, Order\_ID, Product\_ID, Quantity, Price Per Unit, Subtotal.  Customer Table has a relationship with the Orders table via the Customer\_ID and is a one-to-many relationship.   To create the database in SQL we would use the command **CREATE DATABASE** followed by the name we wish to add to the database e.g Shop\_Database. To create tables we would use the following syntax.  \*\*CREATE TABLE\*\* Customers ( Customer\_ID (INT) First Name (Varchar 50) Last Name (Varchar 50) Email (Varchar 100) Phone Number (Char) Address (Varchar 100) City (Varchar 50) County (Varchar 50) Post Code (Varchar 50) Loyalty Points (Numerical));  Creating relationships in SQL involves defining how tables are connected to each other, typically using **primary keys** and **foreign keys**. Here's a concise guide:   * A **primary key** uniquely identifies each record in a table.   Copy the codeCREATE TABLE Customers (  CustomerID INT PRIMARY KEY,  Name VARCHAR(100),  Email VARCHAR(100) );   * A **foreign key** establishes a relationship between two tables by referencing the primary key of another table.   Copy the codeCREATE TABLE Orders (  OrderID INT PRIMARY KEY,  OrderDate DATE,  CustomerID INT,  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) );   * You can enforce referential integrity using constraints like ON DELETE or ON UPDATE.   Copy the codeCREATE TABLE Orders (  OrderID INT PRIMARY KEY,  OrderDate DATE,  CustomerID INT,  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  ON DELETE CASCADE  ON UPDATE CASCADE );   * If tables already exist, you can add relationships using ALTER TABLE.   Copy the codeALTER TABLE Orders ADD CONSTRAINT FK\_CustomerOrder FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);   * Use JOIN to retrieve related data from multiple tables.   Copy the codeSELECT Customers.Name, Orders.OrderDate FROM Customers JOIN Orders ON Customers.CustomerID = Orders.CustomerID;  **To populate the database**  **Orders Table** INSERT INTO table\_name (column1, column2, column3, ...)  VALUES (value1, value2, value3, ...);  A. **Automate updates** or checks when data is inserted or modified.  Example: Automatically update a last\_updated timestamp.  B. **Regular Data Audits**  Periodically review data for inconsistencies, duplicates, or outdated entries.  C. **Normalization**  Structure the database to reduce redundancy and improve consistency.  **Handling Backups and Data Security**  **Automated Backups:**  Schedule full and incremental backups (e.g., daily full, hourly incremental).  Store backups in multiple locations (on-premises + cloud).  **Backup Testing:**  Regularly test backup restoration to ensure data can be recovered.  Automate test restores in a staging environment.  **Encryption:**  Encrypt data at rest (e.g., using TDE or disk-level encryption).  Encrypt data in transit using TLS/SSL.  Access Control and Authentication:  Use strong authentication (e.g., MFA) and least-privilege principles.  Monitor and log access to sensitive data. |

# 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)** [**here**](https://justit831-my.sharepoint.com/:u:/g/personal/danpe_justit_co_uk/Ef6vAaaYVi5FhHqKGxqnn60B9g2khoYekEIO3Y7J00UcJQ?e=pv9NNE)
2. **Follow each step to create your database** [**here**](https://justit831-my.sharepoint.com/:b:/g/personal/danpe_justit_co_uk/EdeCKl2Sas1Hl7u9amDy0fIB9jGVCKxSR0u2-lFOvS5rXw?e=xKv1U7)

**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(Name), CountryCode  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  ORDER BY lifeexpectancy DESC; |

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,population  FROM city  WHERE population >= 2000000  ORDER BY population DESC; |

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,population  FROM city  WHERE name LIKE 'Be%';  ORDER BY name DESC |

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  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; |

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 COUNT(name) AS Occurences, name  FROM city  GROUP BY name  ORDER BY Occurences DESC |

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; |

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; |

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 country.name AS Country, city.name AS City  FROM city  LEFT JOIN country  ON city.id = country.capital  WHERE capital IS NOT NULL  AND country.name = 'spain'; |

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, continent  FROM city  RIGHT JOIN country  ON city.countrycode = country.code  WHERE continent = 'europe' |

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 name, AVG(population)  FROM country  GROUP BY name |

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 City,city.population AS Population  FROM city  LEFT JOIN country  ON city.id = country.capital  WHERE capital IS NOT NULL  ORDER BY population |

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.

|  |
| --- |
| SurfaceArea,  ROUND(Population / SurfaceArea, 2) AS PopulationDensity  FROM  country  WHERE  Population IS NOT NULL AND population > 0  ORDER BY  PopulationDensity ASC  LIMIT 20; |

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.

|  |
| --- |
| SELECT  city.Name AS City,  city.CountryCode,  country.Name AS Country,  city.Population AS CityPopulation,  ROUND(country.GNP / country.Population, 2) AS GDPPerCapita  FROM  city  JOIN  country ON city.CountryCode = country.Code  WHERE  country.Population > 0 AND country.GNP IS NOT NULL  AND (country.GNP / country.Population) > (  SELECT AVG(GNP / Population)  FROM country  WHERE Population > 0 AND GNP IS NOT NULL  )  ORDER BY  GDPPerCapita 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 AS City,  CountryCode,  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:

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