

**Data Technician**

|  |
| --- |
|  |

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

**Table of contents**

[Day 1: Task 1 3](#_Toc193099469)

[Day 1: Task 2 4](#_Toc193099470)

[Day 3: Task 1 4](#_Toc193099471)

[Day 4: Task 1: SQL Practical 6](#_Toc193099472)

[Day 4: Task 2: Written (Optional) 14](#_Toc193099473)

[**Course Notes** 18](#_Toc193099474)

[**Additional Information** 19](#_Toc193099475)

# 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 identifier for a record in a database table. It ensures that each row in the table can be uniquely identified, preventing duplicate records. The primary key must contain unique values for each row, and it cannot have a NULL value. |
| How does this differ from a secondary key? | Compared to a primary key, the secondary key (also known as a non-primary key) it does not need to be unique; it may allow NULL values (depending on the database design), and it is normally used for indexing or improving query performance (e.g., searching for records by a non-unique attribute like name). |
| How are primary and foreign keys related? | A foreign key is a column or a set of columns in a table that creates a link between the data in two tables. It refers to the primary key in another table, establishing a relationship between the two tables.  This creates a parent-child relationship where the primary key is in the parent table, and the foreign key is in the child table. |
| Provide a real-world example of a one-to-one relationship | The relationship between a person and their passport is a one-to-one relationship. |
| Provide a real-world example of a one-to-many relationship | The relationship between an employee and the department they work for is a one-to-many relationship. |
| Provide a real-world example of a many-to-many relationship | The relationship between the students and the courses they are enrolled to is a many-to-many relationship. |

# 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 main difference between relational and non-relational databases lies in their usage, flexibility, scalability, availability, and the type of data they handle (structured or unstructured). In relational databases, data is stored in tables with rows and columns, where each row represents a record and each column represents an attribute. These tables are often linked to each other through unique identifiers (keys). In contrast, non-relational databases, also known as NoSQL databases, store data in a much more flexible manner. They use formats such as key-value pairs, documents, graphs, or wide-column stores, and do not require a fixed schema. This flexibility allows non-relational databases to accommodate data that doesn't fit neatly into a table or that may change over time without needing to adjust the schema.  Another key difference is scalability. Relational databases are typically designed to scale vertically (by adding more power to a single server), while non-relational databases are built for horizontal scaling (by distributing data across multiple servers).  Relational databases are ideal for structured data, such as names, dates, and quantities, which can be easily organized in a table. In contrast, non-relational databases are better suited for unstructured data, like text documents, images, audio, video, and other media types. |
| What type of data would benefit off the non-relational model?  Why? | Non-relational databases are well-suited for storing unstructured, semi-structured, big data, and dynamic data that comes in various formats. These types of data are best handled by non-relational databases because they don’t follow a fixed schema but instead require a flexible structure. This flexibility is important as the data may grow and evolve over time, especially as more data is added. |

# 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. This is useful for querying hierarchical data or comparing rows within the same table. For example, in an Employees table where each employee has a ManagerID referring to another employee, a self-join can be used to list employees along with their managers. |
| Right join | A right join returns all records from the right table and the matched records from the left table. If there is no match, the result is NULL on the side of the left table. If we want to list all departments, including those without any employees assigned, a right join between Employees and Departments on DepartmentID would serve this purpose. |
| Full join | A full join returns all records when there is a match in either the left or right table. Records without a match in one table will have NULLs for the columns of that table. List all students and courses, including students who aren't enrolled in any courses and courses that have no students. |
| Inner join | An inner join retrieves records that have matching values in both tables involved in the join. If there is no match, the record is excluded from the result set. It could be used to retrieve a list of employees (from table Employees) along with their department name (from table Departments), excluding employees who don't belong to a department. The inner join is between these tables on the DepartmentID/EmployeeID/etc field. |
| Cross join | A cross join returns the Cartesian product of the two tables, meaning it combines each row from the first table with all rows from the second table. This can result in a large number of rows in the result set. For example, if you have a Colors table with 3 colors and a Sizes table with 4 sizes, a cross join will produce 12 combinations, representing all possible color-size pairs. |
| Left join | A left join returns all records from the left table and the matched records from the right table. If there is no match, the result is NULL on the side of the right table. Using the same Employees and Departments tables, a left join would list all employees, including those not assigned to any department, with NULL values for department details where applicable. Those without orders will show NULL for order details. |

Day 4: Task 1: 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(city.name) as Number\_of\_cities, city.countrycode AS Name FROM city WHERE city.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 country.Name, country.LifeExpectancy  FROM country  ORDER BY country.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 city.name AS particular\_city\_search, city.countrycode AS Location  FROM city  WHERE city.name  LIKE '%New%'  ORDER BY city.countrycode ASC |

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 city.name, city.population  FROM city  ORDER BY city.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 city.name, city.population  FROM city  WHERE city.population > 2000000  ORDER BY city.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 city.name AS BE\_city\_search, city.countrycode AS Location  FROM city  WHERE city.name LIKE 'Be%'  ORDER BY city.countrycode ASC |

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 city.name, city.population, city.countrycode AS Location  FROM city  WHERE city.population BETWEEN 500000 AND 1000000  ORDER BY city.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 city.name  FROM city  ORDER BY city.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 city.Name, city.Population  FROM city  ORDER BY city.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 city.Name, count(city.name) AS Occurences, city.countrycode AS Country  FROM city  GROUP BY city.name  ORDER BY city.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 city.Name, city.Population  FROM city  ORDER BY city.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 country.Name, country.Population  FROM country  ORDER BY country.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 country.Name, city.name AS Capital\_City  FROM country  JOIN city ON country.Capital = city.ID  WHERE 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 AS City, country.Name as Country  FROM country  JOIN city ON country.code = city.countrycode  WHERE country.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 country.Name as Country, AVG(city.population) AS average\_population  FROM country  JOIN city ON country.code = city.countrycode  GROUP BY country.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.Region, city.name AS Capital\_City, city.Population  FROM country  JOIN city ON country.Capital = city.ID  ORDER BY country.Region |

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 country.Name, country.Population, (country.Population / country.surfacearea) AS Density  FROM country  JOIN city ON country.Capital = city.ID  ORDER BY city.Population 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.

|  |
| --- |
| SELECT country.Name, country.Population, (COALESCE(country.Population, 0) / NULLIF(country.surfacearea, 0)) AS Density  FROM country  WHERE country.Population > 0  ORDER BY country.Population ASC |

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 city.Name, city.Population  FROM city  ORDER BY city.population DESC  LIMIT 10 OFFSET 30 |

# Day 4: Task 2: Written (Optional)

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 |  |

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