

**Data Technician**

|  |
| --- |
|  |

|  |
| --- |
| Name: Michael Thirlaway |
| Course Date: 14/07/2025 |
|  |

**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 is a unique identifier for records in a table. It ensures that no two rows are the same. An example of a primary key would be a CustomerID, because it would be a unique number only for that customer specifically. |
| How does this differ from a secondary key? | A secondary key is also a unique column, but differs from the primary because it is not used as the primary identifier for that customer. Something like the customer’s email would be used as a secondary key, and this is often for identifying in the real world, for instance, an advisor might ask for a customers email or address to verify them, but internally, the computer prefers numbers like the CustomerID. |
| How are primary and foreign keys related? | A foreign key serves as a reference to the primary key in a different table, thus creating a link between the two. For example, in an orders table, the CustomerID might be one of the rows, which would make it the foreign key that would link back to the primary key in the customer table, which means we know which customer made the order. |
| Provide a real-world example of a one-to-one relationship | A one-to-one relationship would be like a customer and their passport. Each customer can only have one passport, and a passport cannot be shared by multiple customers. |
| Provide a real-world example of a one-to-many relationship | A one to many relationship could be a teacher and their students. One teacher has multiple students, but the students as a group only have one teacher. |
| Provide a real-world example of a many-to-many relationship | Students and courses/modules would constitute a many to many relationship. This is because each student can enrol in multiple courses, and the courses themselves will 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 stores data in tables, arranged into rows and columns. The tables are then linked using keys. This is an optimal setup when the data is structured and follows a strict format. Examples of these are like SQL databases.  A non-relational database (NoSQL) stores data in a more flexible way such as documents, graphs, wide columns, or key value pairs. These are more suitable when the structure is liable to change or dealing with bigger or more complex datasets. |
| What type of data would benefit off the non-relational model?  Why? | Things like social media posts, product catalogues, or user profiles are suitable for a non-relational model. Anything that doesn’t follow a strict format.  This is because non-relational databases let you store and access data without needing a strict schema, so it’s easier to scale and handle lots of different types of data quickly. For example, this is optimal for apps that grow fast or change a lot. |

# 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 used to join a table with itself, comparing rows in the same table. An example would be in an employees table, finding all employees who report to the same manager. |
| Right join | A right join shows all records from the right table, and the matching records from the left table. NULLS will appear if there are no matches for the left table columns. An example would be showing all customers even if they haven’t made an order. |
| Full join | A full join returns all records when there is a match in either left or right table. An example would be combining two customer lists from different regions to see all customers and their shared or missing information. |
| Inner join | An inner join returns only the rows with matching values in both tables. For example, finding orders that have a valid customer id in the customers table. |
| Cross join | A cross join returns the Cartesian product of two tables (meaning every row from the first table is paired with the second, used when you want a combination of two sets). An example would be, if you have a table of colours and a table of sizes, say for t-shirts, a cross join will give all colour-size combinations for the product variants. |
| Left join | A left join returns all rows from the left table, and matching rows from the right table. NULLS will appear for results with no match. An example would be getting all employees and their assigned projects, even if some don’t have a project. |

# 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 | The convenience store database will use a structured database with textual, numerical, Boolean, and dates.  The users of the database include till operators, the IT team, business owners, and the data engineers so they can update or make changes to the data as required.  Business owners/directors would use the database to track sales and inventory, revenue streams and profit, as well as KPIs to track how well their loyalty programme keeps customers returning. It would allow them to see the most profitable products vs any outliers that don’t sell as well. And with visual graphs, the Directors could decipher the data quickly.  We would use a relational database with defined tables and clear relationships between the tables, using primary and foreign keys to link them. Each table will have a primary key, using foreign keys in the subsequent tables.  The tables will be structured as follows:  **Customer Table**  CustomerID, FullName, Email, PhoneNumber, LoyaltyPointTotal  **Sales Table**  SaleID, CustomerID, ItemID, SaleQuantity, PointMovement, SaleValue  **Inventory Table**  ItemID, Name, Price, LoyaltyPointValue, Stock  **Staff Table**  StaffID, FullName, AccessLevel  The Customer Table has a relationship with the Sales table via the CustomerID, and the Inventory Table is linked to the sales table via the ItemID.  To create the database in SQL, we’d use the command CREATE DATABASE, followed by the name we want to add to the database, such as Shop\_Database. To create tables, the following syntax would be used:  -- Customer Table  CREATE TABLE Customer (  CustomerID INT PRIMARY KEY,  FullName VARCHAR(100),  Email VARCHAR(100),  PhoneNumber VARCHAR(20),  LoyaltyPointTotal INT  );  -- Inventory Table  CREATE TABLE Inventory (  ItemID INT PRIMARY KEY,  Name VARCHAR(100),  Price DECIMAL(10,2),  LoyaltyPointValue INT,  Stock INT  );  -- Staff Table  CREATE TABLE Staff (  StaffID INT PRIMARY KEY,  FullName VARCHAR(100),  AccessLevel VARCHAR(50)  );  -- Sales Table  CREATE TABLE Sales (  SaleID INT PRIMARY KEY,  CustomerID INT,  ItemID INT,  SaleQuantity INT,  PointMovement INT,  SaleValue DECIMAL(10,2),  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),  FOREIGN KEY (ItemID) REFERENCES Inventory(ItemID)  );  The foreign keys establish the relationships between the two tables by referencing the primary keys of another table.  To input data into the tables, you would populate them by using INSERT functions. For example:  INSERT INTO Customer (CustomerID, FullName, Email, PhoneNumber, LoyaltyPointTotal)  VALUES (1, 'Alice Johnson', 'alice@example.com', '07700111222', 120);   * This would create a record for Alice Johnson, showing her email, phone number, and her loyalty points.   The same would be done for the other tables with all their relevant information.  Maintaining the database  To maintain the database remains accurate and up to date, we would use a series of measures, including:   * Input validation. Using front-end and back-end checks to ensure only valid data is entered, such as the correct email format and phone number length. * Constraints. Enforcing accuracy with primary key, foreign key, NOT NULL, and CHECK constraints. * Regular data audits. Scheduled reviews by the IT team to find and correct errors, duplicates, or outdated records. * Triggers or stored procedures. This would automate updates where possible, such as updating loyalty points with each sale. * Staff training. This ensures the users entering the data understand how to use the system properly, reducing the risk of errors.   To handle backups and data security, additional measures are necessary, such as:   * Regular backups. Automated full backups, like on a weekly basis, and storing them securely, either on site or on a cloud. * Access control. Different users would have different roles and permissions to restrict access depending on the user’s job. For instance, only an administrator could edit staff records. * Encryption. This encrypts sensitive data like personal information, both at rest (using TDE or disc level encryption) or in transit using TLS/SSL. * Audit logs. This tracks changes to important tables, such as the sales and customer tables, and could identify suspicious activity. * Recovery plan. Having a tested recovery process in case of failure, attack, or data loss to recover lost 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)**
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 shortest 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.**