

**Data Technician**

|  |
| --- |
|  |

|  |
| --- |
| Teacher’s Name: Dr. Ala Mostafa  Candidate Name: Abul K M Raquibuddin |
| Course Date: 21/04/2025 |
|  |

**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, which uniquely identifies every row in the table. For example, Student\_ID is the primary key for the Student entity. |
| How does this differ from a secondary key? | Secondary key or foreign key, which refer other attribute from the other table. For example, Course\_ID at the student table refereeing the course details from the course table. |
| 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. A foreign key column in a table points to a column with unique values in another table (often the primary key column) to create a way of cross-referencing the two tables. |
| Provide a real-world example of a one-to-one relationship | Relationship between students and Students contact information where one student has unique phone number. |
| Provide a real-world example of a one-to-many relationship | Relationship between customer and orders where one customer can place many orders. |
| Provide a real-world example of a many-to-many relationship | Relationship between authors and books where many authors can release many books, and many books can be released by many authors. However, it is good practice to avoid many to many entity relationship by putting another entity at the middle. |

# 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? | In relational database two or more tables build relationship in respect to their correspondence key such as primary key, foreign key, or composite key to access related data. For example, entity relationship diagram that use in SQL. More example of relational database systems includes Microsoft SQL server, Oracle, and Microsoft Access.    On the other hand, non-relational database or NoSQL, there are no tables, rows, primary keys or foreign keys, it uses a storage data in a more natural and flexible way such as single data structure, for example store data in JASON like normal documents, and graphs. It can handle large dataset that do not fit well into traditional relational database. Characteristics of NoSQL: |
| What type of data would benefit off the non-relational model?  Why? | Unstructured, semi structured, high volume and real time analytical data benefits non-relational model, well suited for storing and managing unstructured or semi-structured data such as documents, and large datasets. It does not need rigid schema, so the structure can change over time. Also, they can handle large volume of data and scale horizontally  Characteristics of NoSQL: |

# 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 | It is a type of SQL query where table where the table is joined with itself. It's used when we want to compare rows within the same table.  For example, it can be used when primary key of the table pointing to the foreign key to refers another entity.  Syntax:  SELECT column\_name(s) FROM table1 T1, table1 T2 WHERE condition; |
| Right join | It is a type of SQL query where rows from the right table compare with the rows from the left table, if matched then return columns to the left otherwise return null value to the left columns  Syntax:  SELECT column\_name(s) FROM table1 RIGHT JOIN table2 ON table1.column\_name = table2.column\_name; |
| Full join | It is a type of SQL query where the table rows will be compare with the either left or right and if found matched value it will return the value accordingly, otherwise will return null value.  Syntax:  SELECT column\_name(s)  FROM table1  FULL JOIN table2  ON table1.column\_name = table2.column\_name; |
| Inner join | It is a type of SQL query where will compere between two tables to for the rows with matching values in both tables, if they found will return the matched values otherwise will return null value. In conclusion, it selects records that have matching values in both tables  Syntax:  SELECT column\_name(s) FROM table1 INNER JOIN table2 ON table1.column\_name = table2.column\_name; |
| Cross join | The cross joint returns all records from both tables. This means every row from the first table is combined with every row from the second table.  Syntax:  SELECT column\_name(s) FROM table1 CROSS JOIN table2; |
| Left join | In this query returns all rows from the left table, and the matching rows from the right table if there is no matching it will return null value.  Syntax:  SELECT column\_name(s)  FROM table1  LEFT JOIN table2  ON table1.column\_name = table2.column\_name; |

# 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:**    1. **What kind of data will the database need to store?**   According to scenario there are 3 tables: inventory, sales and customers. However, we have includes two more tables sale item and loyalty Transactions to enhance accessibility. All are the primary and foreign keys are auto incremental, does not require to user input. Also, dates are accepted from the system and loyalty points are related to the spending and the category of products. So, the business needs to store information on inventory table (names, categories, prices, and Stock quantities), sales item table (quantity), and customers table (first names, last names, email, and phone). This data will support daily operations and strategic decisions.  **b.** **Who will be the users of the database, and what will they need to accomplish?**  Typical users include cashiers, store managers, and database administrators. Cashiers will record sales and update inventory, managers will monitor stock and analyse sales, and admins will maintain data integrity and security.  **2. Designing the Database Schema:**  **a. How would you structure the database tables to efficiently store inventory, sales, and customer information?**  **Customer Table**   |  |  |  | | --- | --- | --- | | Field Name | Data Type | Description | | customer\_id | INT (PK) | Unique customer ID | | first\_name | VARCHAR(50) | Customer's first name | | last\_name | VARCHAR(50) | Customer's last name | | email | VARCHAR(100) | Email address | | phone | VARCHAR(20) | Contact number | | loyalty\_points | INT | Loyalty points |   **Inventory Table**   |  |  |  | | --- | --- | --- | | Field Name | Data Type | Description | | product\_id | INT (PK) | Unique product ID | | name | VARCHAR | Product name | | category | TEST | Item category | | price | DECIMAL | Unit price | | stock\_quantity | INT | Stock quantity |   **Sale Item Table**   |  |  |  | | --- | --- | --- | | Field Name | Data Type | Description | | sale\_item\_id | INT (PK) | Unique ID of sales item | | sale\_id | INT (FK) | Links to Sales.sale\_id | | Product\_id | INT (FK) | Links to Inventory.product\_id | | quantity | INT | Quantity of product sales | | Subtotal | DECIMAL | Total amount sales |   **Sales Table**   |  |  |  | | --- | --- | --- | | Field Name | Data Type | Description | | sale\_id | INT (PK) | Unique sales ID | | Customer\_id | INT (FK) | Links to Customer.customer\_id | | Sale\_date | INT (FK) | Links to Inventory.product\_id | | Total\_amount | INT | Quantity of product sales |   **Loyalty Transaction Table**   |  |  |  | | --- | --- | --- | | Field Name | Data Type | Description | | loyalty\_id | INT (PK) | Unique sales ID | | customer\_id | INT (FK) | Links to Customer.customer\_id | | points\_earned | INT (FK) | Links to Inventory.product\_id | | Points\_redeemed | INT | Quantity of product sales | | transaction\_date | DATETIME | Date of the transaction |   **b. What relationships between tables are necessary (e.g., how sales relate to inventory and customers)?**    ***3.* Implementing the Database:**  **a. What SQL commands would you use to create the database and its tables?**  Syntax:  CREATE DATABASE databaseName;  Example:  CREATE DATABASE IF NOT EXISTS retailDB;  **b. Provide examples of SQL statements for creating tables and defining relationships between them.**  **Customer table:**  CREATE TABLE customers(  customer\_id INT NOT NULL,  first\_name VARCHAR(50) NOT NULL,  last\_name VARCHAR(50) NOT NULL,  address VARCHAR(100) NOT NULL,  city VARCHAR(50) NOT NULL,  country VARCHAR(50) NOT NULL,  email VARCHAR(50) NULL,  phone VARCHAR(12) NULL,  loyalty\_points INT NOT NULL,  PRIMARY KEY (customer\_id),  FOREIGN KEY (loyalty\_points)  REFERENCES retaildb.Loyalty\_Transactions(loyalty\_id)  );  **Inventory Table:**  CREATE TABLE Inventory(  product\_id INT NOT NULL,  product\_name VARCHAR(50) NOT NULL,  category VARCHAR(50) NULL,  price DECIMAL NOT NULL,  stock\_quantity INT NOT NULL,  PRIMARY KEY (product\_id)  );  **Loyalty Transactions Table:**  CREATE Loyalty\_Transactions(  Loyalty\_id INT NOT NULL,  Customer\_id INT NOT NULL,  Points\_earned INT,  transation\_date DATE,  PRIMARY KEY (Loyalty\_id),  FOREIGN KEY (customer\_id) REFERENCES Customers (customer\_id)  );  **Sales Item Table:**  CREATE TABLE sales\_item(  sale\_id INT NOT NULL,  product\_id INT NULL,  sales\_item VARCHAR(50) NULL,  quantity INT NULL,  subtotal DECIMAL NULL,  PRIMARY KEY (sale\_id),  FOREIGN KEY (product\_id)  REFERENCES retaildb.inventory (product\_id)  );  **Sales Table:**  CREATE TABLE Sales(  sales\_id INT NOT NULL,  customer\_id INT NOT NULL,  sale\_date DATE,  total\_amount DECIMAL,  PRIMARY KEY (sales\_id),  FOREIGN KEY (customer\_id) REFERENCES Customers (customer\_id)  );  **4. Populating the Database:**  **a. How would you input initial data into the database? Give examples of SQL INSERT statements.**  **Customer table:**  INSERT INTO retaildb.customers(  first\_name,  last\_name,  address,  city,  country,  email,  phone,  loyalty\_points  )  VALUES ('1', 'Amin', 'Uddin', '7 Keppel Road', 'London', 'UK', 'abcd@yahoo.com', '+447768543', '100');    **Inventory table**:  INSERT INTO inventory(product\_name, category, price, stock\_quantity)  VALUES ('Lux', 'Soap', '1.00', '200');    **Sale item table:**  INSERT INTO sale\_item(sale\_item\_id, sale\_id, product\_id, quantity, subtotal)  VALUES ('1', '1', '1', '20', '500'); |
|  |  |

# 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.

|  |
| --- |
| Syntax:  SELECT column1,column2  FROM table\_name  WHERE condition; |

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.

|  |
| --- |
| Syntax:  SELECT column1, column2  FROM table\_name  WHERE expr\_operator (SELECT select\_list  FROM table); |
|  |

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.

|  |
| --- |
| Syntax:  SELECT column1, column2  FROM table\_name  WHERE column LIKE pattern; |

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.

|  |
| --- |
| Syntax:  SELECTcolumn1, column2  FROM table\_name  ORDER BY column1, column2, ASC|DESC  LIMIT number; |

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.

|  |
| --- |
| Syntax:  SELECTcolumn1,column2  FROM table\_name  WHERE condition  ORDER BY column1, column2, ASC|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.

|  |
| --- |
| Syntax:  SELECT column1, column2…  FROM table\_name  WHERE column LIKE pattern; |

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.

|  |
| --- |
| Syntax:  SELECT column1, column2,…  FROM table\_name  WHERE condition  ORDER BY column1, column2, ... ASC|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.

|  |
| --- |
| Syntax:  SELECT column1, column2, ...  FROM table\_name  ORDER BY column1, column2, ...ASC|DESC; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name(s)  FROM table\_name  ORDER BY column1, column2, ...ASC|DESC,  LIMIT number; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name, COUNT(column\_name) AS alias\_name  FROM table\_name  GROUP BY column\_name(s)  ORDER BY column\_name(s) 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.

|  |
| --- |
| Syntax:  SELECT column\_name(s)  FROM table\_name,  ORDER BY column1, column2, ... ASC  LIMIT number; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name(s)  FROM table\_name,  ORDER BY column1, column2, ... DESC  LIMIT number; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name AS alias\_name, column\_name A Salias\_name  FROM table1  LEFT JOIN table2  ON table1.column\_name = table2.column\_name  WHERE condition; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name(s)  FROM table\_name  WHERE condition  ORDER BY column1, column2, ... ASC  LIMIT number; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name AS alias\_name, column\_name AS alias\_name  FROM table1  LEFT JOIN table2  ON table1.column\_name = table2.column\_name  WHERE condition; |

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.

|  |
| --- |
| Syntax:  SELECT column\_name(s)  FROM table1  LEFT JOIN (column\_name1, column\_name2, column\_name3  FROM table1  GROUP BY column\_name) AS alias\_name  ON table1.column\_name = table2.column\_name  WHERE condition; |
|  |

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.

|  |
| --- |
| Syntax:  SELECT column\_name(s)  FROM table1  LEFT JOIN table2  ON table1.column\_name = table2.column\_name  ORDER BY column1 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.

|  |
| --- |
| Syntax:  SELECT column\_name AS alias\_name  FROM table\_name  ORDER BY column1 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.

|  |
| --- |
| Syntax:  SELECT city.Name, A.GDP, A.AVG\_GDP  FROM (SELECT Name, capital, (GNP/Population) AS GDP,  (SELECT AVG(GNP/Population) FROM country) AS AVG\_GDP  FROM country  WHERE (GNP/Population) > (SELECT AVG(GNP/Population) FROM country)) AS A  LEFT JOIN city  ON A.Capital = city.ID  ORDER BY A.GDP 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.

|  |
| --- |
| Syntax:  SELECTcolumn\_name(s)  FROM table\_name  ORDER BY Column DESC  LIMIT number; |

|  |
| --- |
| **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:

I sincerely appreciate your dedication and the effort you put into teaching us every day. Your clear explanations make complex topics easier to understand, and your encouragement inspires me to do my best. I am thankful for the time and attention you give to facilitating our learning and development. I am grateful to be in your class.

This week, we learned many important topics on Databases and Structured Query Language (SQL) includes. Some of the key areas covered are outlined below:

* Database definition and design
* Table, Entity, Primary key, Foreign key, Composite key, and data modelling
* Relational Database Management System, and Non-Relational Database
* Microsoft Access:
  + Data types, Tables, ERD, Query, and Reporting
* SQL and MySQL:
  + SELECT, WHERE, ORDER BY, GROUP BY, Aliases
  + Wildcard characters such as ‘%’, ‘-‘
  + Logical operators such as AND, OR, NOT, BETWEEN, IN, LIKE and mathematical calculations such as COUNT(), AVG(), SUM(), MIN(), MAX()
  + CREATE DATABASE, CREATE TABLE, DROP TABLE, ALTER TABLE, DROP TABLE, INSERT INTO, UPDATE, DELETE
  + Self Join, LEFT JOIN, RIGHT JOIN, INNER JOIN
  + Worked on real databases
  + Linking MySQL with Power BI

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