

Data Technician

Name: Nathan Mullings

Course Date: 2025/03/03

Table of contents

Day 1: Task 1	3
Day 1: Task 2	
Day 3: Task 1	
Day 4: Task 1: SQL Practical	
Day 4: Task 2: Written (Optional)	

Day 1: Task 1

Please research and complete the below questions relating to key concepts of databases.

What is a primary key?	Is a unique identifier in a table eg. Customer id
How does this differ from a secondary key?	A secondary key (also known as a foreign key) links to a primary key from another table, but its self is not a primary key.
How are primary and foreign keys related?	A foreign key links and references a primary key from another table.
Provide a real-world example of a one-to-one relationship	Passport ID & person's full name. each person can have only one passport, and each passport belongs to only one person.
Provide a real-world example of a one-to-many relationship	customer name & order id A customer can place multiple orders. Each order belongs to only one customer.
Provide a real-world example of a many-to-many relationship	Students & Courses A student can enrol in multiple courses. A 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?

Relational database is structured data, with predefined schemas and relationships between tables.

Non-relational databases is un-structured data that comes in various formats (e.g., key-value pairs, documents, graphs) that don't require a fixed schema.

What type of data would benefit off the non-relational model?

Social media data would benefit from the non-relational model because it often involves unstructured or semi-structured data (e.g., posts, comments, images, and user interactions) that can vary in format and structure. Non-relational databases allow flexibility in storing and scaling this diverse, evolving data without requiring a predefined schema, making them more suitable for handling the dynamic nature of social media content.

Why?

In short, social media data often includes complex and varied information that fits well with the flexibility of non-relational databases.

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 type of join in SQL where a table is joined with itself. This is useful when comparing rows within the same table. A self-join can be performed using INNER JOIN or LEFT JOIN, treating one instance of the table as the "left" and another as the "right." SELECT country.name AS country_name, AVG(city.population) AS avg_population FROM country JOIN city ON country.code = city.countrycode GROUP BY country.name ORDER BY avg_population DESC;
Right join	A RIGHT JOIN (or RIGHT OUTER JOIN) in SQL returns all records from the right table and the matching records from the left table. If there is no match, NULL values are returned for columns from the left table. SELECT columns FROM left_table RIGHT JOIN right_table ON left_table.common_column = right_table.common_column;
Full join	A FULL JOIN (or FULL OUTER JOIN) in SQL returns all records from both the left and right tables. If there is a match, it includes the corresponding data from both tables. If there is no match, NULL values appear for missing data from either table. SELECT columns FROM left_table FULL JOIN right_table ON left_table.common_column = right_table.common_column;
Inner join	An INNER JOIN in SQL returns only the matching records from both tables based on a specified condition. Rows from both tables are included only if there is a match in the common column. SELECT columns FROM table1 INNER JOIN table2 ON table1.common_column = table2.common_column;

Cross join	A CROSS JOIN in SQL returns the Cartesian product of two tables, meaning it combines every row from the first table with every row from the second table. This type of join does not require a condition to match rows between the tables. As a result, the number of rows in the result will be the product of the number of rows in each table. SELECT columns FROM table1 CROSS JOIN table2; Caution: CROSS JOIN can result in large datasets if the tables have many
	rows, as it produces every possible combination.
	A LEFT JOIN (or LEFT OUTER JOIN) in SQL returns all records from the
	left table and the matching records from the right table. If there is no match, NULL values are returned for columns from the right table.
Left join	SELECT columns FROM left_table
	LEFT JOIN right_table ON left_table.common_column = right_table.common_column;

Day 4: Task 1: SQL Practical

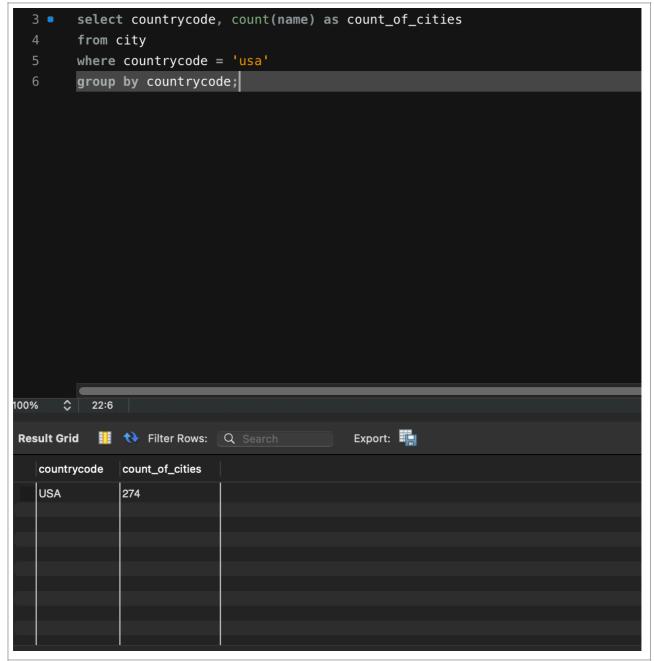
In your groups, work together to answer the below questions. It may be of benefit if one of you shares your screen with the group and as a team answer / take screen shots from there.

Setting up the database:

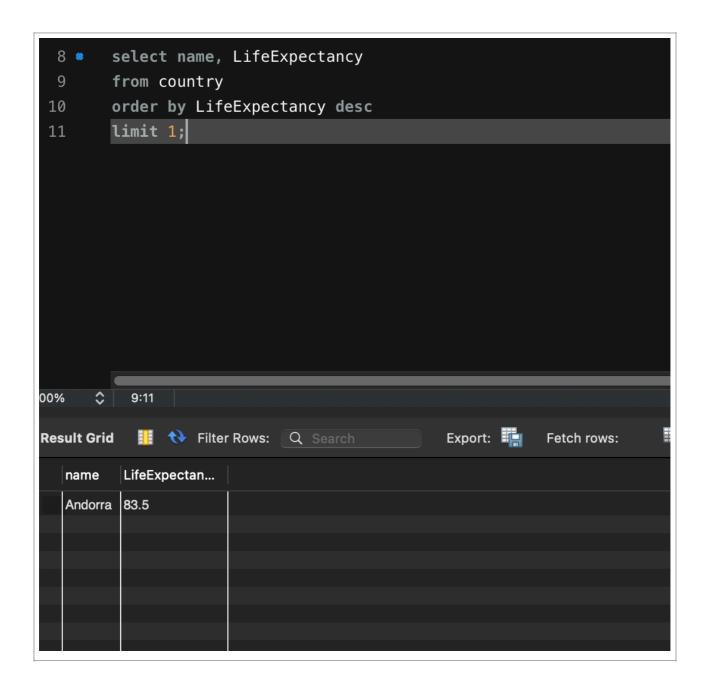
- 1. Download world_db(1) here
- 2. Follow each step to create your database here

For each question I would like to see both the syntax used and the output.

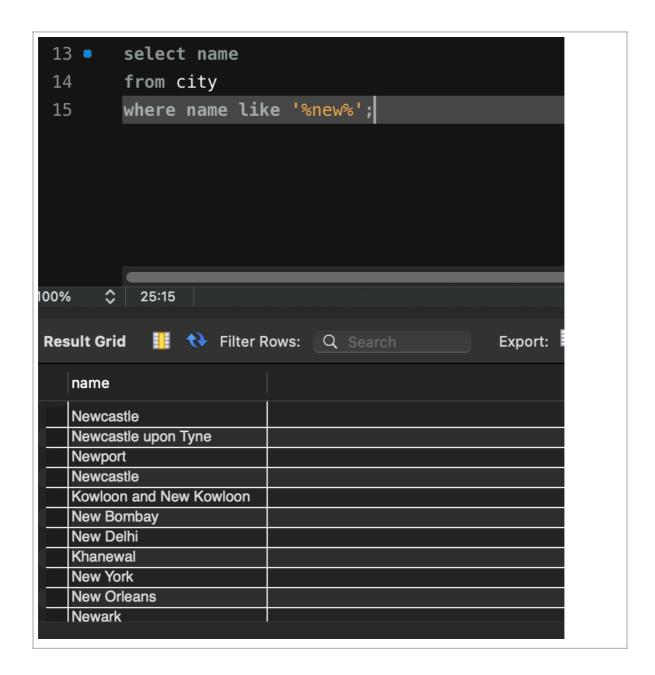
1. **Count Cities in USA:** *Scenario:* You've been tasked with conducting a demographic analysis of cities in the United States. Your first step is to determine the total number of cities within the country to provide a baseline for further analysis.



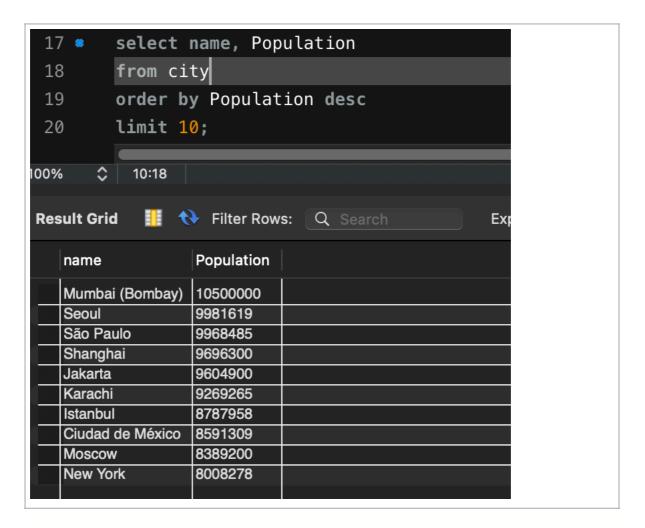
2. **Country with Highest Life Expectancy:** Scenario: As part of a global health initiative, you've been assigned to identify the country with the highest life expectancy. This information will be crucial for prioritising healthcare resources and interventions.



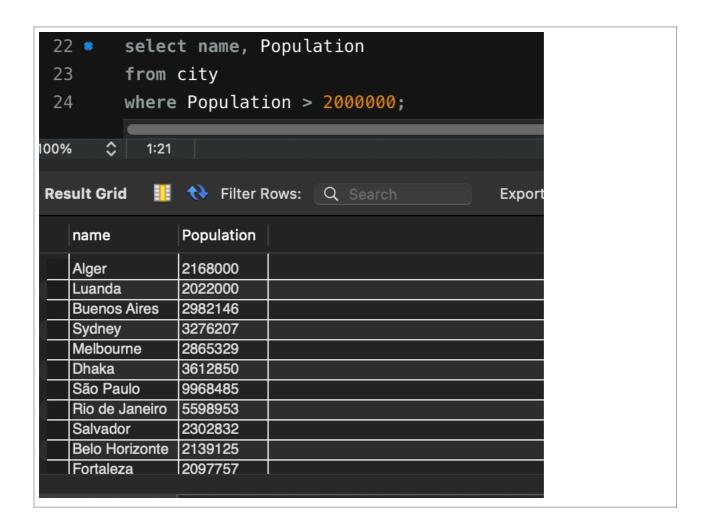
3. "New Year Promotion: Featuring Cities with 'New: Scenario: In anticipation of the upcoming New Year, your travel agency is gearing up for a special promotion featuring cities with names including the word 'New'. You're tasked with swiftly compiling a list of all cities from around the world. This curated selection will be essential in creating promotional materials and enticing travellers with exciting destinations to kick off the New Year in style.



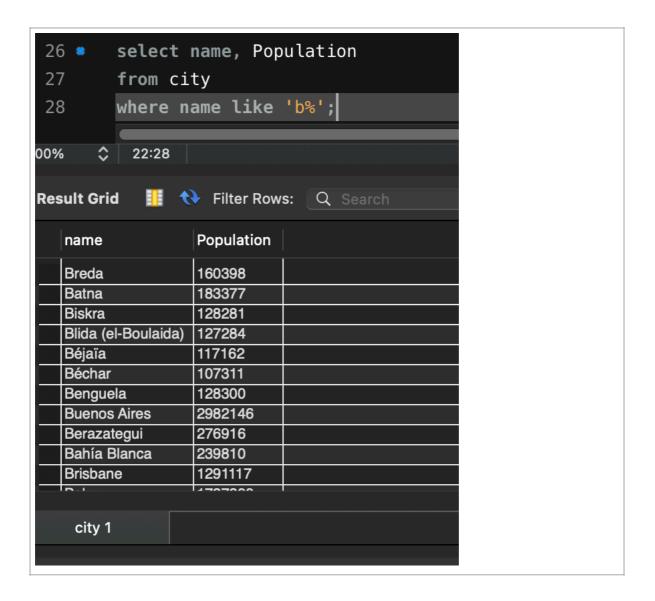
4. **Display Columns with Limit (First 10 Rows):** *Scenario:* You're tasked with providing a brief overview of the most populous cities in the world. To keep the report concise, you're instructed to list only the first 10 cities by population from the database.



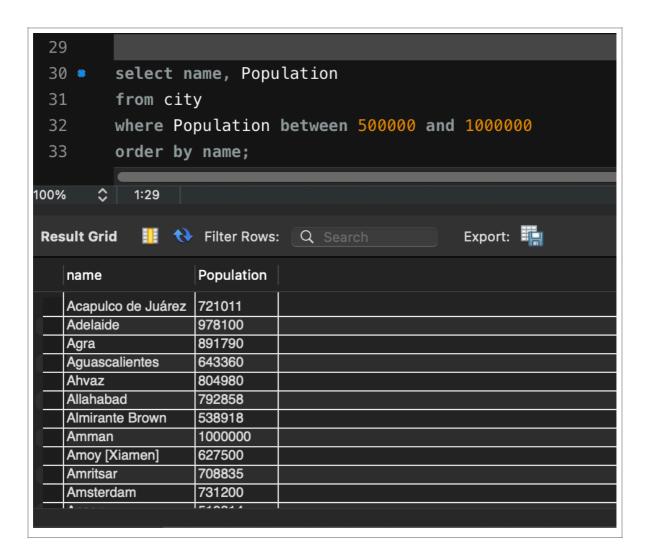
5. **Cities with Population Larger than 2,000,000:** *Scenario:* A real estate developer is interested in cities with substantial population sizes for potential investment opportunities. You're tasked with identifying cities from the database with populations exceeding 2 million to focus their research efforts.



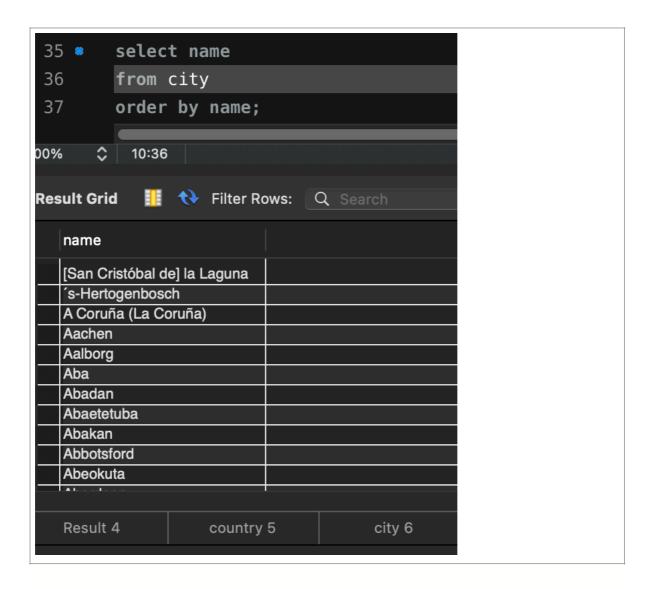
6. **Cities Beginning with 'Be' Prefix:** *Scenario:* A travel blogger is planning a series of articles featuring cities with unique names. You're tasked with compiling a list of cities from the database that start with the prefix 'Be' to assist in the blogger's content creation process.



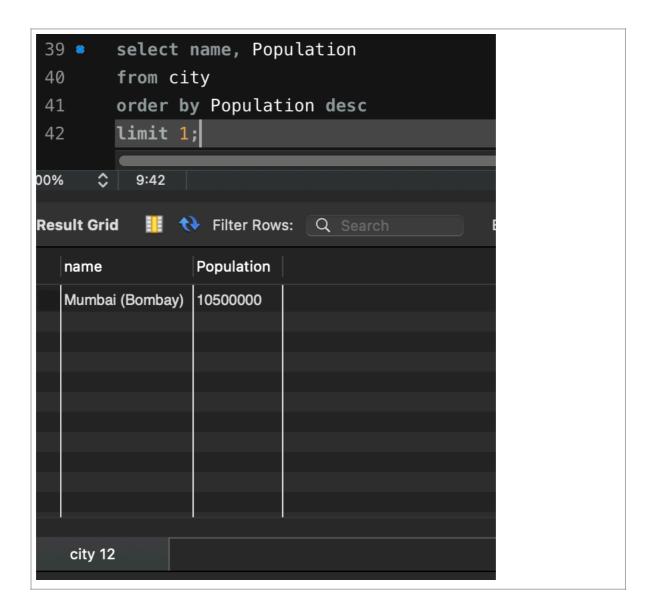
7. **Cities with Population Between 500,000-1,000,000:** *Scenario:* An urban planning committee needs to identify mid-sized cities suitable for infrastructure development projects. You're tasked with identifying cities with populations ranging between 500,000 and 1 million to inform their decision-making process.



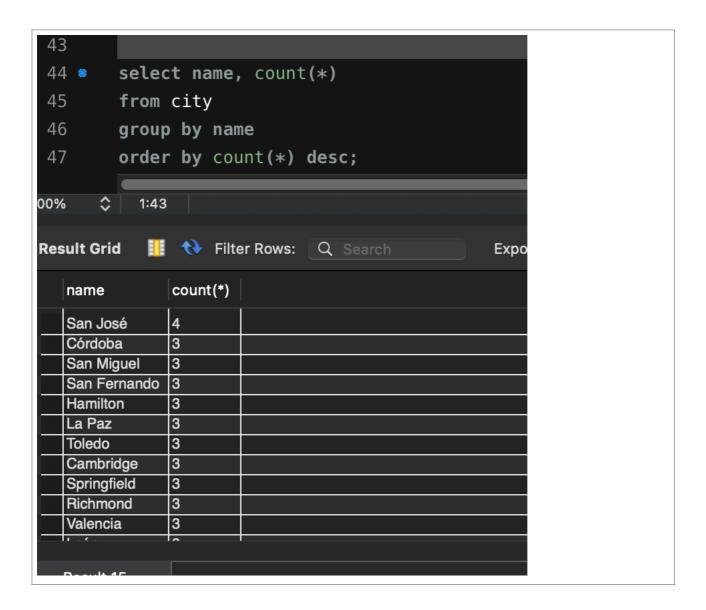
8. **Display Cities Sorted by Name in Ascending Order:** *Scenario:* A geography teacher is preparing a lesson on alphabetical order using city names. You're tasked with providing a sorted list of cities from the database in ascending order by name to support the lesson plan.



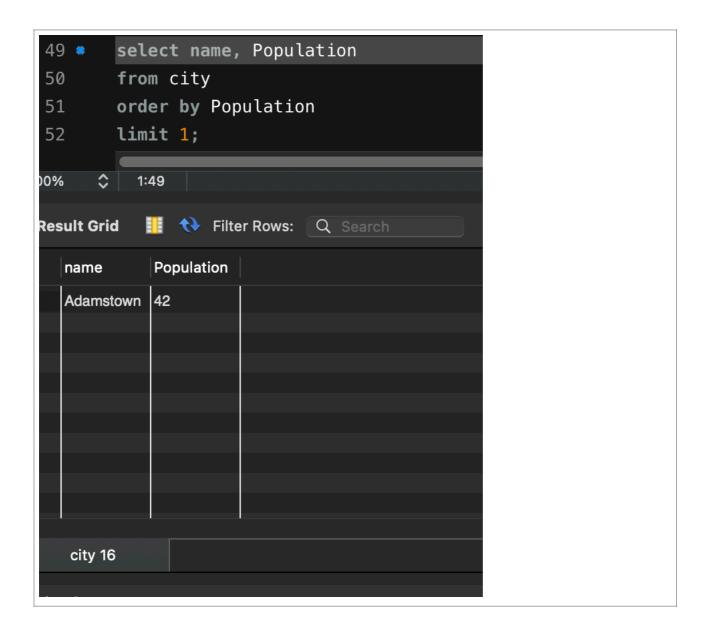
 Most Populated City: Scenario: A real estate investment firm is interested in cities with significant population densities for potential development projects. You're tasked with identifying the most populated city from the database to guide their investment decisions and strategic planning.



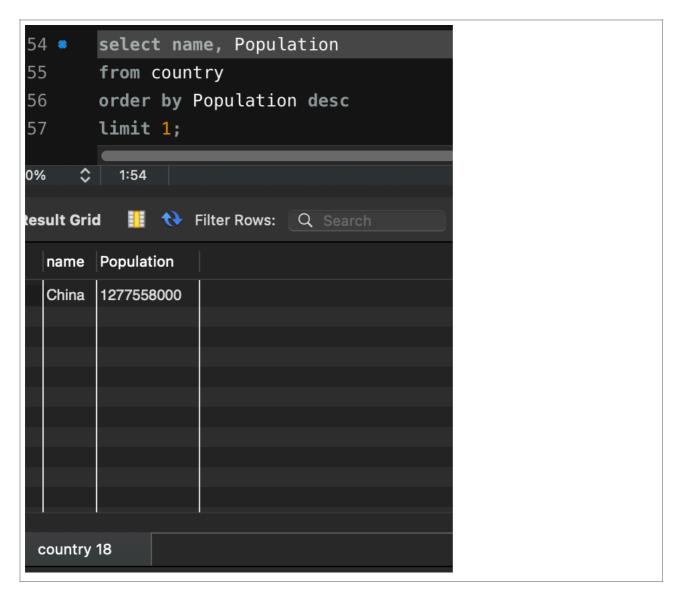
10. **City Name Frequency Analysis: Supporting Geography Education** *Scenario*: In a geography class, students are learning about the distribution of city names around the world. The teacher, in preparation for a lesson on city name frequencies, wants to provide students with a list of unique city names sorted alphabetically, along with their respective counts of occurrences in the database. You're tasked with this sorted list to support the geography teacher.



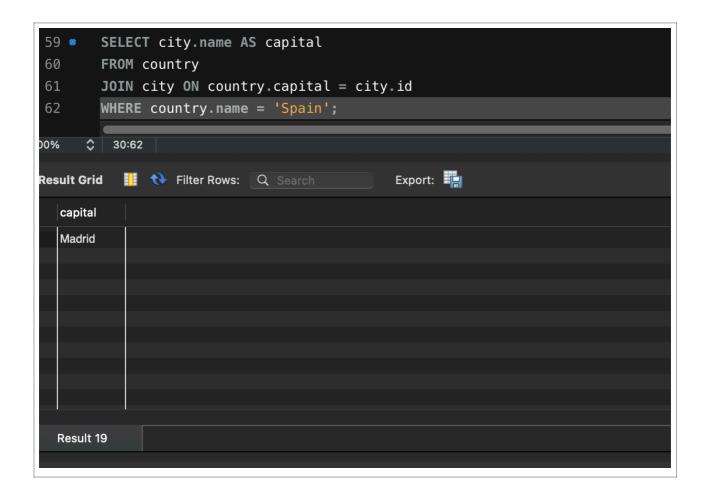
11. City with the Lowest Population: Scenario: A census bureau is conducting an analysis of urban population distribution. You're tasked with identifying the city with the lowest population from the database to provide a comprehensive overview of demographic trends.



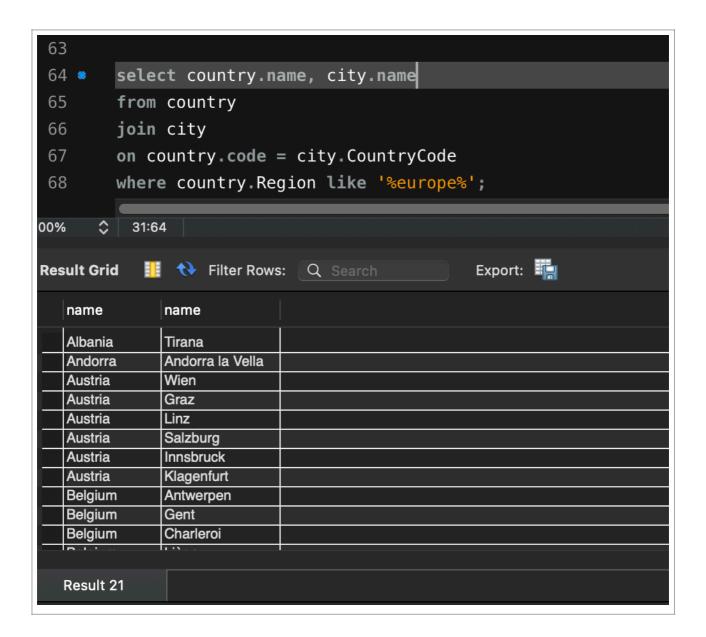
12. **Country with Largest Population:** Scenario: A global economic research institute requires data on countries with the largest populations for a comprehensive analysis. You're tasked with identifying the country with the highest population from the database to provide valuable insights into demographic trends.



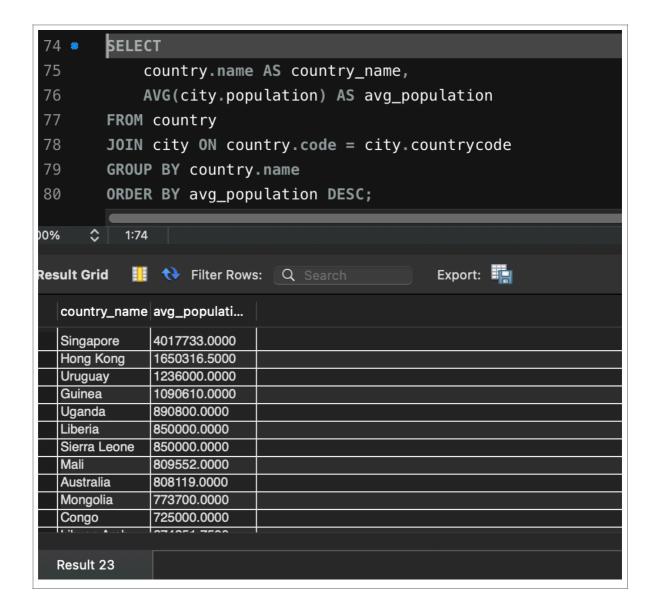
13. **Capital of Spain:** Scenario: A travel agency is organising tours across Europe and needs accurate information on capital cities. You're tasked with identifying the capital of Spain from the database to ensure itinerary accuracy and provide travellers with essential destination information.



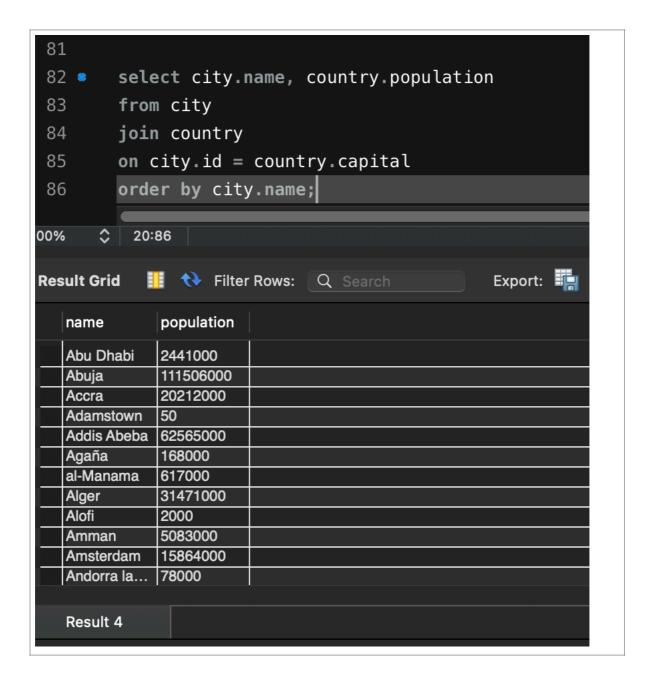
14. **Cities in Europe:** *Scenario:* A European cultural exchange program is seeking to connect students with cities across the continent. You're tasked with compiling a list of cities located in Europe from the database to facilitate program planning and student engagement.



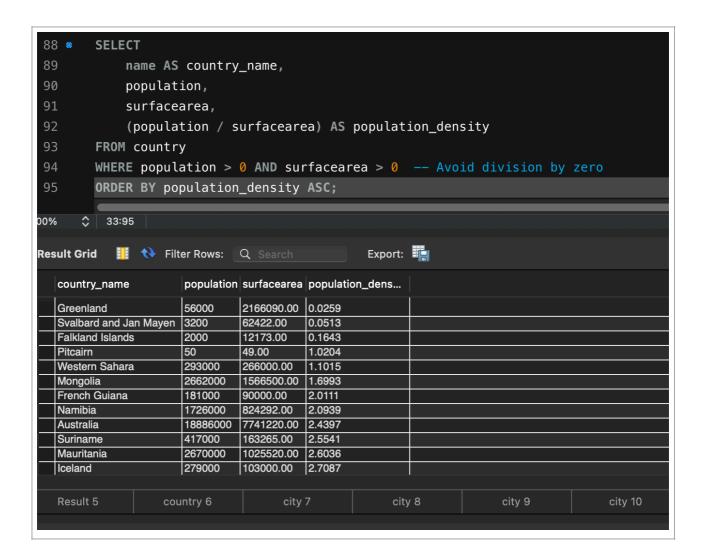
15. Average Population by Country: Scenario: A demographic research team is conducting a comparative analysis of population distributions across countries. You're tasked with calculating the average population for each country from the database to provide valuable insights into global population trends.



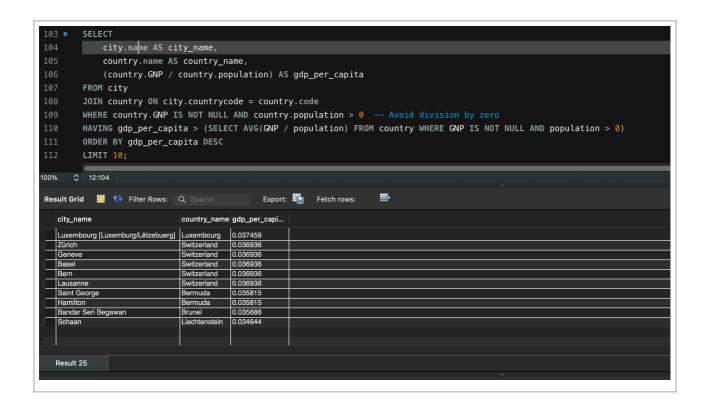
16. Capital Cities Population Comparison: Scenario: A statistical analysis firm is examining population distributions between capital cities worldwide. You're tasked with comparing the populations of capital cities from different countries to identify trends and patterns in urban demographics.



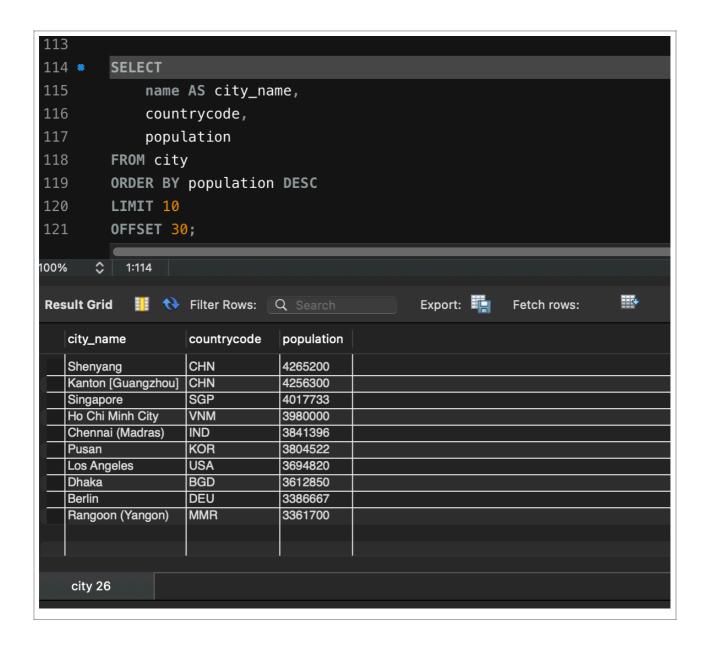
17. **Countries with Low Population Density:** Scenario: An agricultural research institute is studying countries with low population densities for potential agricultural development projects. You're tasked with identifying countries with sparse populations from the database to support the institute's research efforts.



18. 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.



19. **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.



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:

- a. What kind of data will the database need to store?
- b. Who will be the users of the database, and what will they need to accomplish?

2. Designing the Database Schema:

- a. How would you structure the database tables to efficiently store inventory, sales, and customer information?
- 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?
- b. Provide examples of SQL statements for creating tables and defining relationships between them.

4. Populating the Database:

a. How would you input initial data into the database? Give examples of SQL INSERT statements.

5. Maintaining the Database:

- a. What measures would you take to ensure the database remains accurate and up to date?
- b. 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 500word essay here

Designing and Implementing a Database for a Small Retail Business

Understanding the Business Requirements

To create an efficient database for a small retail business, it is crucial to understand the data that needs to be stored and the users who will interact with it. The database should include information on:

• **Inventory**: Product names, categories, stock levels, supplier details, and pricing.



- Sales: Transaction records, date and time of sales, items sold, total amounts, and payment methods.
- **Customers**: Personal details (name, phone number, email), purchase history, and loyalty program details.
- **Employees**: Staff managing sales and inventory, along with user roles and permissions. The primary users of the database will be store employees, managers, and possibly suppliers. Employees will need to process sales update inventory and register customers in the loyalty

Employees will need to process sales, update inventory, and register customers in the loyalty program. Managers will require access to reports on sales trends, stock levels, and customer activity.

Designing the Database Schema

A well-structured schema ensures data integrity and efficient retrieval. The key tables and their relationships include:

- **Products (Inventory Management)**: Stores product details.
- **Customers**: Tracks customer information.
- Sales: Records each transaction, linking to products and customers.
- Sales_Items: Manages the many-to-many relationship between sales and products.
- Loyalty_Program: Tracks points and rewards for customers.

Table Relationships

- A sale belongs to one customer, but a customer can have multiple sales (one-to-many).
- A **sale** consists of multiple **products**, and a product can be in multiple sales (many-to-many, requiring a junction table Sales_Items).
- The **loyalty program** links to the **customers** table.

Implementing the Database

Using SQL, we create the tables with appropriate relationships:

```
CREATE DATABASE RetailDB;
USE RetailDB;
CREATE TABLE Products (
    product id INT PRIMARY KEY AUTO INCREMENT,
    name VARCHAR(100) NOT NULL,
    category VARCHAR(50),
    price DECIMAL(10,2) NOT NULL,
    stock quantity INT NOT NULL,
    supplier VARCHAR(100)
);
CREATE TABLE Customers (
    customer id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    phone VARCHAR(15),
    email VARCHAR(100),
    loyalty points INT DEFAULT 0
```

```
);
CREATE TABLE Sales (
    sale id INT PRIMARY KEY AUTO INCREMENT,
    customer id INT,
    sale date DATETIME DEFAULT CURRENT TIMESTAMP,
    total amount DECIMAL(10,2),
    FOREIGN KEY (customer id) REFERENCES
Customers(customer id)
);
CREATE TABLE Sales Items (
    sale id INT,
    product id INT,
    quantity INT,
    subtotal DECIMAL(10,2),
    PRIMARY KEY (sale id, product id),
    FOREIGN KEY (sale id) REFERENCES Sales(sale id),
    FOREIGN KEY (product id) REFERENCES Products(product id)
);
Populating the Database
Initial data can be inserted using SQL:
INSERT INTO Products (name, category, price, stock quantity,
supplier) VALUES
('Milk', 'Dairy', 1.50, 100, 'Local Dairy Co.'),
('Bread', 'Bakery', 2.00, 50, 'Bakery Supplies Ltd.');
INSERT INTO Customers (name, phone, email) VALUES
('John Doe', '1234567890', 'john@example.com');
INSERT INTO Sales (customer id, total amount) VALUES
(1, 15.00);
INSERT INTO Sales Items (sale id, product id, quantity,
subtotal) VALUES
(1, 1, 2, 3.00),
(1, 2, 1, 2.00);
Maintaining the Database
```

To ensure data accuracy and security:

• **Data Validation**: Implement constraints (e.g., NOT NULL, CHECK) to prevent incorrect data entries.

- Regular Backups: Schedule daily or weekly backups using MySQL's mysqldump command: mysqldump -u root -p RetailDB > backup.sql
- Data Security: Assign roles and permissions using SQL's GRANT statement: GRANT SELECT, INSERT, UPDATE ON RetailDB.* TO 'store user'@'localhost';

By following these steps, the small retail business can effectively manage its operations, ensuring accurate inventory tracking, streamlined sales processing, and customer relationship management. This database will help improve efficiency, reduce errors, and support future growth.



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.