Just IT

B2W group

Apprenticeships | Training | Recruitment

Data Technician

Name: Youhem Rouainia

Course Date: 13/01/2024

Table of contents

Day 1: Task 1

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

What is a primary key?	The primary key is used uniquely to identify an entity.
How does this differ from a secondary key?	The secondary key (foreign) references an entity in another table.so it is a primary key in another table.
How are primary and foreign keys related?	The primary key is a unique key in that table, but it is at the same time a foreign key in another table as it is connected.
Provide a real-world example of a one-to-one relationship	The one-to-one relationship is like husband-wife. For example one customer can have multiple sales ,but each sale is placed by only one customer.
Provide a real-world example of a one-to-many relationship	The one-to-many relationship is like mother ---< kids for example we can have one customer who has a lot of subscriptions.
Provide a real-world example of a many-to-many relationship	The many-to-many relationship is quite complicated relationship which cannot be directly linked as it causes problem and slowing down the cardinalities. For example, we have students>----< teachers so this relationship can be broken down by creating another entity which allows the other two table to be linked through it like a classroom: students---<classroom>----teachers.

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 the relational database the data stored in database objects called tables which is a collection of related data entries, and it contains columns and rows. For example, we have MySQL, Oracle and Microsoft Access and it is optimised for analytics as we call it a star Shema. However, the non-relational database the data is stored in key value pair like images, videos, social media which is a big data having a big volume, a fast velocity and diversity. The NoSQL has more floors and buildings which can be vertically and horizontally, it is designed to handle large datasets that don't fit well into the relational database. For example, we have MongoDB and Redis .
What type of data would benefit off the non-relational model? **Why?**	The suitable data for the non-relational database is the unstructured /semi structured which has a changing requirements like social media, videos. As the relational database has a large volume which can scale horizontally to handle large amounts of data, and it can as well digest and organize various type of information side by side. It stores documents that are highly detailed and contain a range of different type of information in different formats. Example of non-relational database includes MongoDB, Redis Cassandra and HBase.

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 when the table is joined with itself. This allows us to compare rows within the same table. For example, a self-join can help us retrieve employee- manager relationship, where each employee in a table has a reference to their manager's ID in order to extract the list of employees along with the names of their managers. Another example of self-join using the W3 website matching products that got the same price. SELECT A.ProductName AS ProductName1, B.ProductName AS ProductName2, A.Price FROM Products A, Products B WHERE A.ProductID <> B.ProductID AND A.Price = B.Price ORDER BY A.Price;
Right join	The right join is used to combine data from two tables based on a related column. It returns all records from the right table along with the matching records from the left table. In case of no matching the return value is NULL. For example, we consider two tables customer table containing details of the customer placing an order and the order table contains the details of the customer so as the right join gives the matching rows and the rows that are present in the right table but not in the left table.
Full join	

	The full join is used to combine the records from two or more tables. It retrieves all rows from both tables involved in the joined when there is a match in left table or right table. For example, we consider two tables order table containing details of the order and datils of an employee and an employee table contains details of the employee, so the full join gives the employee's name with the order ID.
Inner join	The inner join allows us to combine two or more tables based on a related column, returning only the records that satisfy the join condition. For example, we consider two tables a teacher table which contains data about the teacher while the course table holds information about the course that these teachers teach. The common column between These tables is ID from the teacher table and teacherID from the course table.so now if we write a query to retrieve the courseID, teacherID, teacher Name and their salary by joining the teacher table and the course table using the inner join. Another example using the W3 website when selecting the two tables order details and products in order to select all orders details with the products name. SELECT OrderDetails.OrderDetailID, Products.ProductName FROM OrderDetails INNER JOIN Products ON OrderDetails.OrderDetailID = Products.ProductID;
Cross join	The cross join returns the cartesian product of two or more tables. This means it matches each row from the left table with every row from the right table, resulting in a combination of all possible pairs of records. For example, we consider having two tables: the customer table and the orders table by using the cross-join command to match data of the customer and orders table. The output will return all similar records from both tables and if there are rows in customer

	or orders that do not match any entries table those rows will also be listed.
Left join	The left join allows to retrieve all records from the left table and only the matching records from the right table. For example, we consider having two tables employee table containing the employee details and the department table containing the details of the department. The output of these left join gives the matching rows and rows that are present in the left table but not in the right table, so we will have employeeID , name , department name and location.

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**:*
    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 500-word essay here

1. Understanding the Business Requirements

The client would like to store 3 entities: inventory, sales and customer information. Therefore, we need to ensure all fields for these entities are stored accurately and efficiently. For example, the client might want to maintain records for the price and stock levels of various inventory items. They may want to keep track of the quantity of items sold, and the revenue generated from sales. They may also want to store customer information in the database, for example customer names, customer IDs, contact information, and loyalty points that the customers have earned through the loyalty program.

The users of the database will include business owners, shop floor employees, and the IT department. Business owners will need access in order to check records and modify entries. Shop floor employees, for example shelf stockers or cashiers, will need basic access to the database. Their actions would affect stock level, revenue, and loyalty points. For example, a cashier scanning product at the checkout would affect all three of the previously mentioned fields. The IT department would need full database access in order to maintain the database by cleaning data and ensuring that there are no leaks.

2. Designing the Database Schema:

Our database will contain three tables.

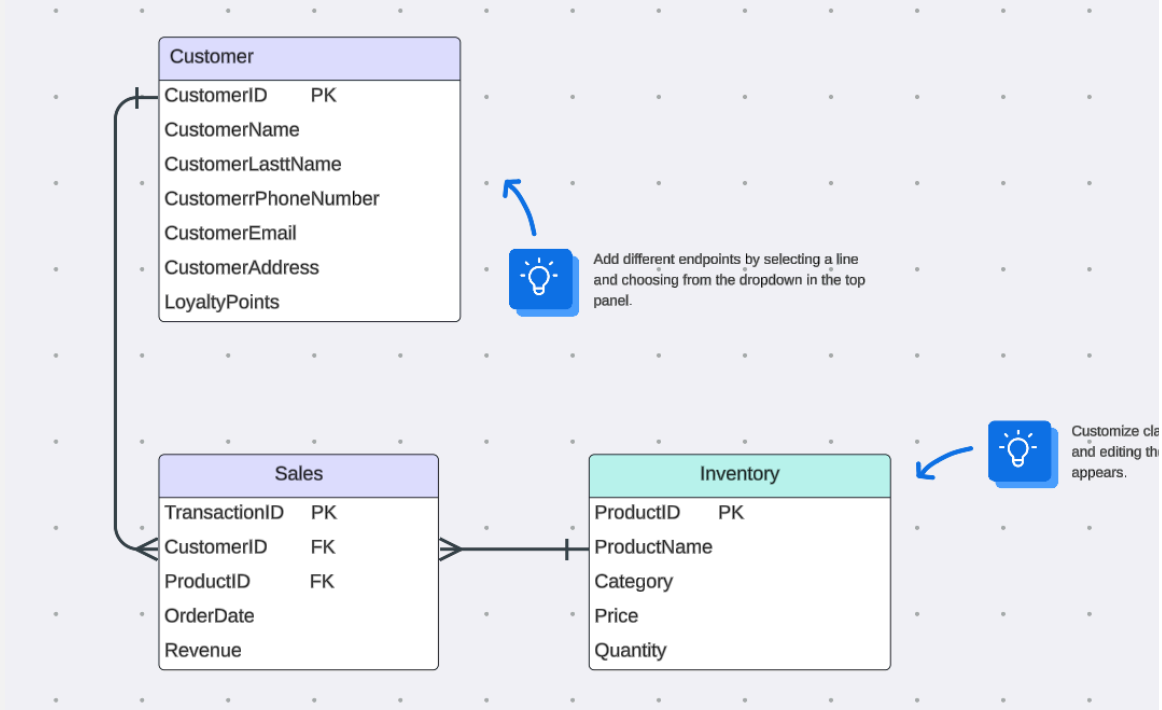
The first table is the Inventory table. This entity represents the items that the shop sells. The key attributes could include ProductID as a primary key (PK), ProductName, Category,Price and Quantity. For example:

ProductID	ProductName	Category	Price	Quantity
1	Milk	Dairy	£1.50	15
2	Bread	Bakery	£1.00	20

The second table is the Customer table. This entity represents the company's customers. The key attributes could include CustomerID as primary key (PK), CustomerName, CustomerLastName, CustomerEmail, CustomerPhoneNumber and CustomerAddress and LoyaltyPoints.

The last table is the Sales table. This entity represents the orders placed by customers. Key attributes could include TransactionID (PK), CustomerID (FK), referencing customer ProductID (FK), OrderDate and Revenue.

In designing these tables, we set the cardinalities between each entity. We can create relationships between them. For example, one customer can have multiple sales, but each sale is placed by only one customer. A sale can contain multiple inventories, and one inventory can be part of sales.



3. Implementing the Database:

We would use MySQL to create this database.
To create the database, we could use the following statement:

CREATE DATABASE databasename;

To create tables, we would use the CREATE TABLE statement. Below is the statement we would use to create the Inventory table.

CREATE TABLE Inventory (
ProductName varchar(255) NOT NULL,
ProductID varchar(20) NOT NULL,
Category varchar(20),
Price TINYINT,
Quantity SMALLINT,
PRIMARY KEY (ProductID));

The Quantity and Price data types are set to integer and all others set to strings of differing lengths. We have used the NOT NULL command for ProductName and ProductID. The primary key for this table was set as ProductID, which can be used for defining relationships with other tables. Similar statements would be needed to create the other tables.

4. Populating the Database:

We could input the data manually into the MySQL database that we have created. Alternatively, we could use an insert statement to populate the database, for example:

```
INSERT INTO Inventory (ProductName, ProductID, ProductType)
VALUES ('Milk', ‘1’, 'Dairy');
```

We could also use an INSERT INTO SELECT statement in order to copy relevant data from one table into another:

```
INSERT INTO Inventory (CustomerId)
SELECT CustomerId
FROM Customers;
```

This would result in the customer ID information being copied from the Inventory table into the Customers table.

5. Maintaining the Database:

The database will require regular updates to maintain accuracy. For example, if 5 of 15 bottles of milk are sold, the record must reflect the remaining 10. Routine data quality audits would be conducted to identify and address inconsistencies, duplicates, or outdated records, maintaining the integrity of the database.
To secure the database, daily incremental and weekly full backups will be implemented. Access will be restricted to trusted staff using an access list and role-based control, ensuring GDPR compliance and reducing data breach risks. Activity logs will be used to monitor changes and detect any potential security breaches. Sensitive data will be encrypted to prevent unauthorised access. Parameterised queries will be employed to safeguard against SQL injection attacks. While customers currently have no way to input data directly into the database, this practice remains essential to uphold.

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:

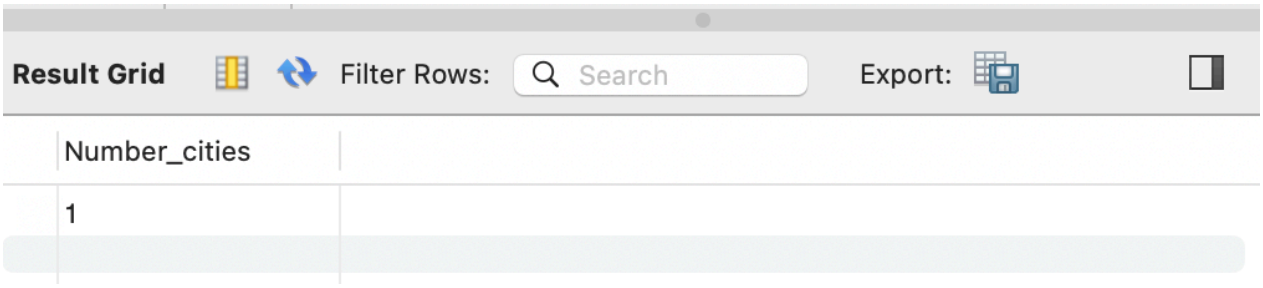
1. **Download world_db(1)**
2. **Follow each step to create your database**

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.

```
SELECT COUNT(Region) Number_cities
FROM world.country
WHERE Code LIKE 'USA%';
```

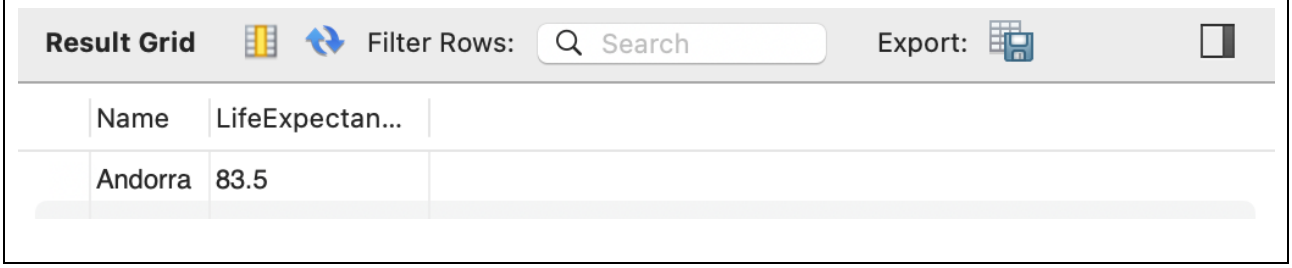
Counting the cities in USA by using the aggregation function count, to determine the total number of cities.



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.

```
SELECT Name , LifeExpectancy From world.country
WHERE LifeExpectancy=(SELECT  MAX(LIfeExpectancy)
FROM world.country);
```

In this SQL statement I have compared the life expectancy with the maximum number of life expectancy to identify the country with highest life expectancy .

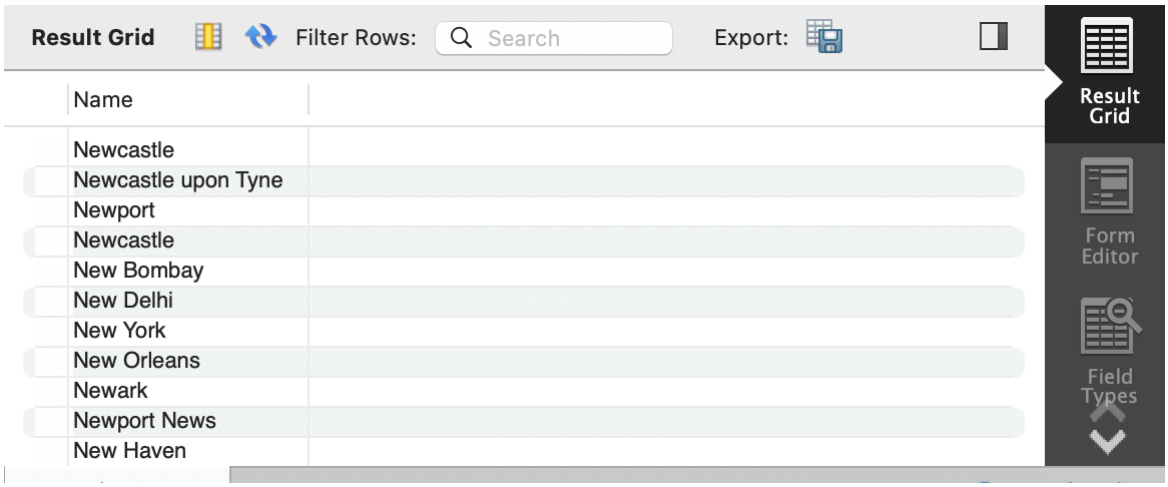


Name	LifeExpectan...
Andorra	83.5

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.

```
SELECT Name
FROM world.city
WHERE Name LIKE'NEW%'
;

I have selected all cities starting with the letters “NEW’.
```

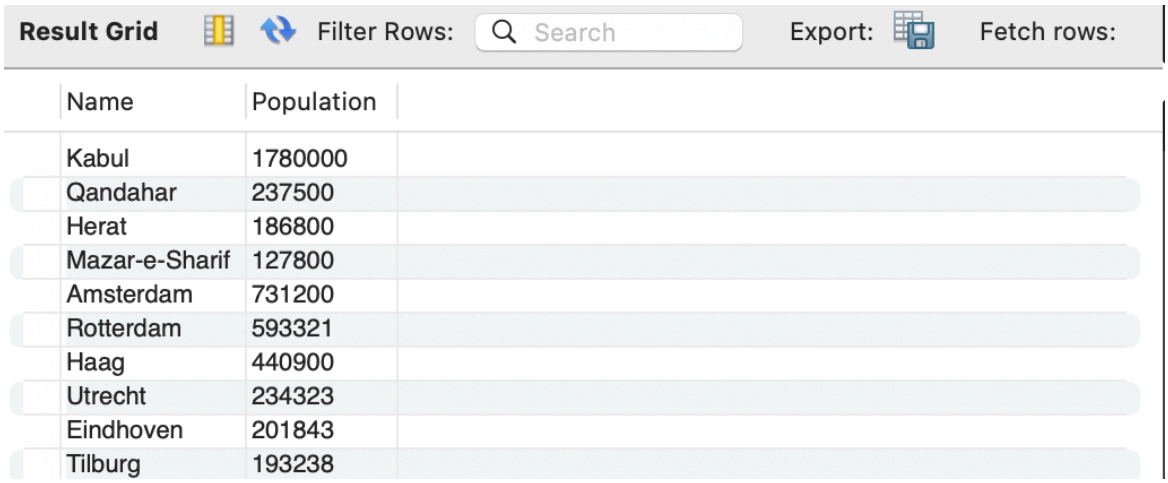


Name
Newcastle
Newcastle upon Tyne
Newport
Newcastle
New Bombay
New Delhi
New York
New Orleans
Newark
Newport News
New Haven

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.

```
SELECT Name, Population
FROM world.city;

LISTING THE FIRST 10 CITIES BY THE HIGHEST POPULATION.
```



Name	Population
Kabul	1780000
Qandahar	237500
Herat	186800
Mazar-e-Sharif	127800
Amsterdam	731200
Rotterdam	593321
Haag	440900
Utrecht	234323
Eindhoven	201843
Tilburg	193238

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.

```
SELECT Name, Population
FROM world.city
WHERE Population> 2000000;
```

By filtering the number of population, we List the countries exceeding 2 million population.

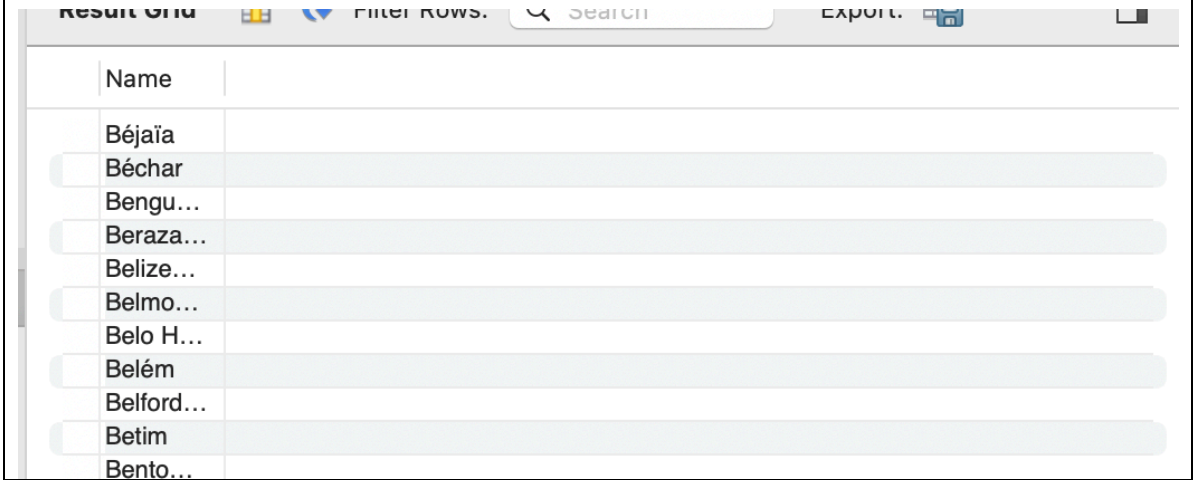
Result Grid | Filter Rows: Search | Export:

Name	Population
Alger	2168000
Luanda	2022000
Buenos Aires	2982146
Sydney	3276207
Melbourne	2865329
Dhaka	3612850
São Paulo	9968485
Rio de Janeiro	5598953
Salvador	2302832
Belo Horizonte	2139125
Fortaleza	2097757

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.

```
SELECT Name
FROM world.city
WHERE Name LIKE 'Be%';
```

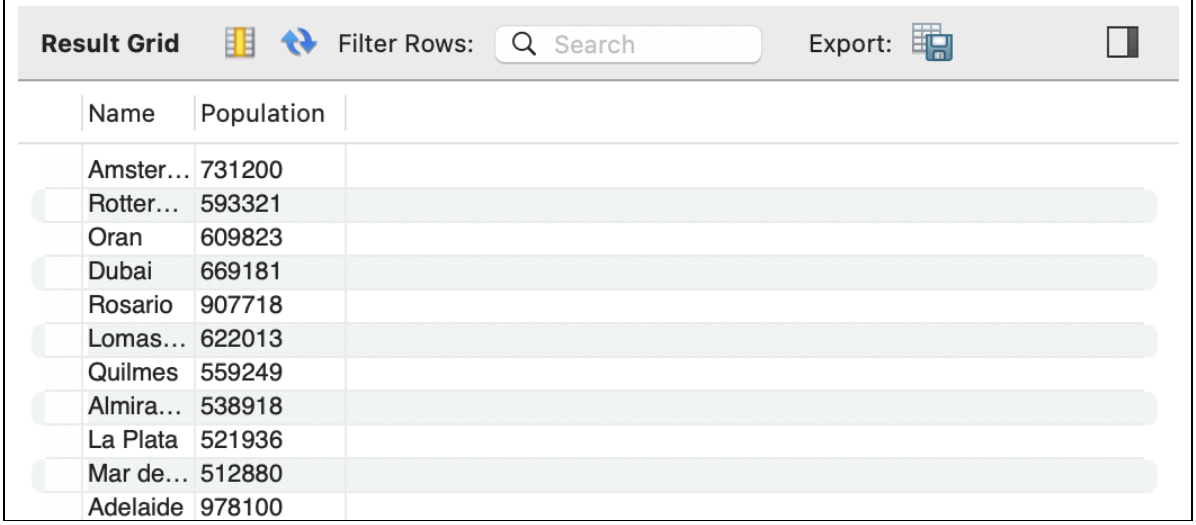
Listing the countries starting with the letter'Be', using the filter.



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.

```
SELECT Name, Population
FROM world.city
WHERE Population BETWEEN 500000 AND 1000000;
```

Filtering the number of population between 500,000 and 1,000,000 in order to list the name of the countries.

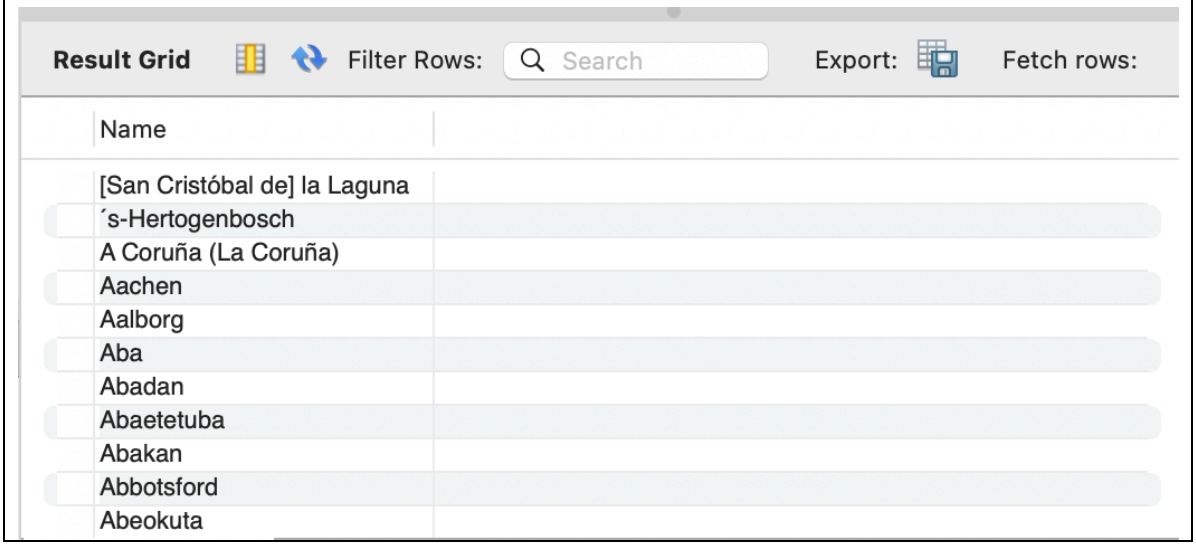


Name	Population
Amster…	731200
Rotter…	593321
Oran	609823
Dubai	669181
Rosario	907718
Lomas…	622013
Quilmes	559249
Almira…	538918
La Plata	521936
Mar de…	512880
Adelaide	978100

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.

```
SELECT Name FROM world.city
ORDER BY Name;

By selectiong the name of the cities we get by default the list of cities in ascending
order.
```



Name
[San Cristóbal de] la Laguna
´s-Hertogenbosch
A Coruña (La Coruña)
Aachen
Aalborg
Aba
Abadan
Abaetetuba
Abakan
Abbotsford
Abeokuta

9. **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.

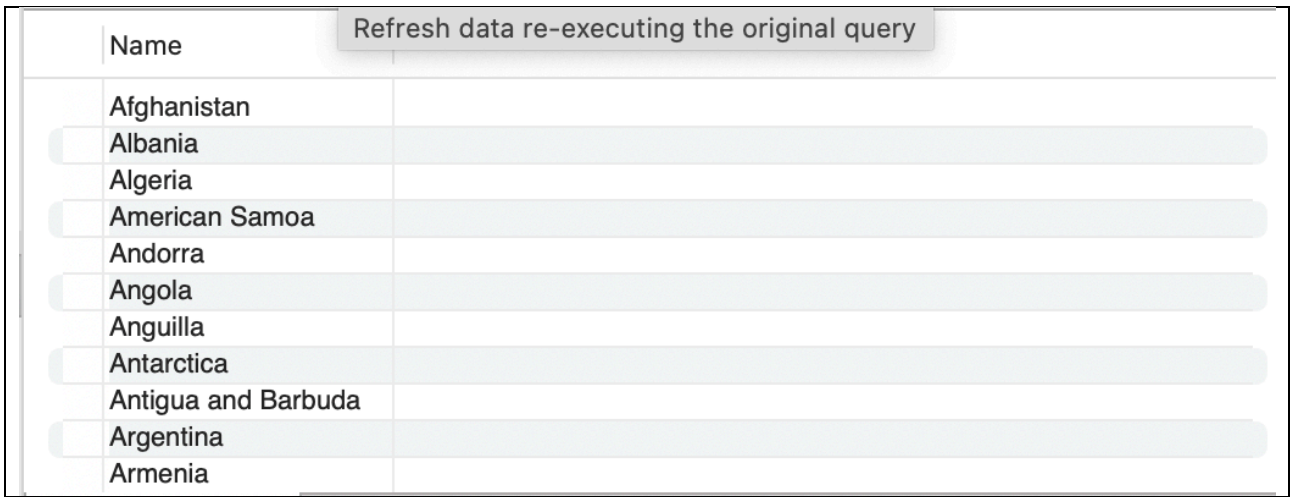
```
SELECT NAME
FROM world.country
WHERE(SELECT MAX(LifeExpectancy) AND MAX(Population)
FROM world.country)
;
```

In this SQL statement I compared the maximum number of life expectancy with the maximum number of populations to list the country name which match those criteria as the same time.

NAME
Aruba
Afghanistan
Angola
Anguilla
Albania
Andorra
Netherlands Antilles
United Arab Emirates
Argentina
Armenia
American Samoa

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.

```
SELECT DISTINCT Name  FROM world.country
ORDER BY Name;
```



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.

```
SELECT NAME FROM world.city
WHERE (SELECT MIN(Population)
FROM world.city);
```

Using the minimum aggregation function to get the name of cities with the lowest population.

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.

```
SELECT NAME FROM world.city
WHERE (SELECT MAX(Population)
FROM world.city);
```

Listing all countries with the maximum population using the aggregation function maximum.

NAME
Kabul
Qanda…
Herat
Mazar-…
Amster…
Rotter…
Haag
Utrecht
Eindho…
Tilburg
Gronin…

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.

```
SELECT city.Name AS CityName, country.Name AS CountryName
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE country.Name = 'Spain'
ORDER BY city.Name ASC;
```

In this SQL statement I joined the two tables city and country to match the code of country and from there we must use the filter to get all cities in the country Spain.

CityName	CountryName
[San Cristóbal de] la Laguna	Spain
A Coruña (La Coruña)	Spain
Albacete	Spain
Alcalá de Henares	Spain
Alcorcón	Spain
Algeciras	Spain
Alicante [Alacant]	Spain
Almería	Spain
Badajoz	Spain
Badalona	Spain
Barakaldo	Spain

14. **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.

```
SELECT Name , LifeExpectancy From world.country
WHERE LifeExpectancy=(SELECT  MAX(LIfeExpectancy)
FROM world.country)
;
```

Result Grid Filter Rows: Search Export:

Name	LifeExpectan...
Andorra	83.5

15. **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 CityName,
       city.Population,
       country.Name AS CountryName,
       country.Continent
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE country.Continent = 'Europe'
```

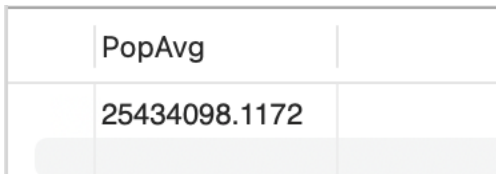
ORDER BY city.Name ASC;

Joining the two tables city and country by matching the same code of the country ,filtering the continent that matches the name Europe we get the name of cities and their populations with the continent name,

CityName	Population	CountryName	Continent
A Coruña (La Coruña)	243402	Spain	Europe
Aachen	243825	Germany	Europe
Aalborg	161161	Denmark	Europe
Abakan	169200	Russian Federation	Europe
Aberdeen	213070	United Kingdom	Europe
Aix-en-Provence	134222	France	Europe
Albacete	147527	Spain	Europe
Alcalá de Henares	164463	Spain	Europe
Alcorcón	142048	Spain	Europe
Alessandria	90289	Italy	Europe

16. **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 AVG(Population) PopAvg
FROM world.country;



PopAvg
25434098.1172

I have calculated the average population (see the screenshot above), however I did calculate the population for each country.(see the below).

SELECT DISTINCT(country.name) , country.population
FROM world.country
JOIN city ON country.code=city.countrycode;

name	population
Aruba	103000
Afghanistan	22720000
Angola	12878000
Anguilla	8000
Albania	3401200
Andorra	78000
Netherlands Antilles	217000
United Arab Emirates	2441000
Argentina	37032000
Armenia	3520000
American Samoa	68000

17. **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 city.Name AS CityName, country.Name AS CountryName ,
            city.Population
FROM city
JOIN country ON city.CountryCode = country.Code
ORDER BY city.Population DESC;

In the SQL statement above I have joined the country table with city using the code
field and by ordering number of population from the highest to the lowest, in order to
get the city name, country name and their population.
```

CityName	CountryName	Population
Mumbai (Bombay)	India	10500000
Seoul	South Korea	9981619
São Paulo	Brazil	9968485
Shanghai	China	9696300
Jakarta	Indonesia	9604900
Karachi	Pakistan	9269265
Istanbul	Turkey	8787958
Ciudad de México	Mexico	8591309
Moscow	Russian Federation	8389200
New York	United States	8008278
Tokyo	Japan	7980230

18. **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.

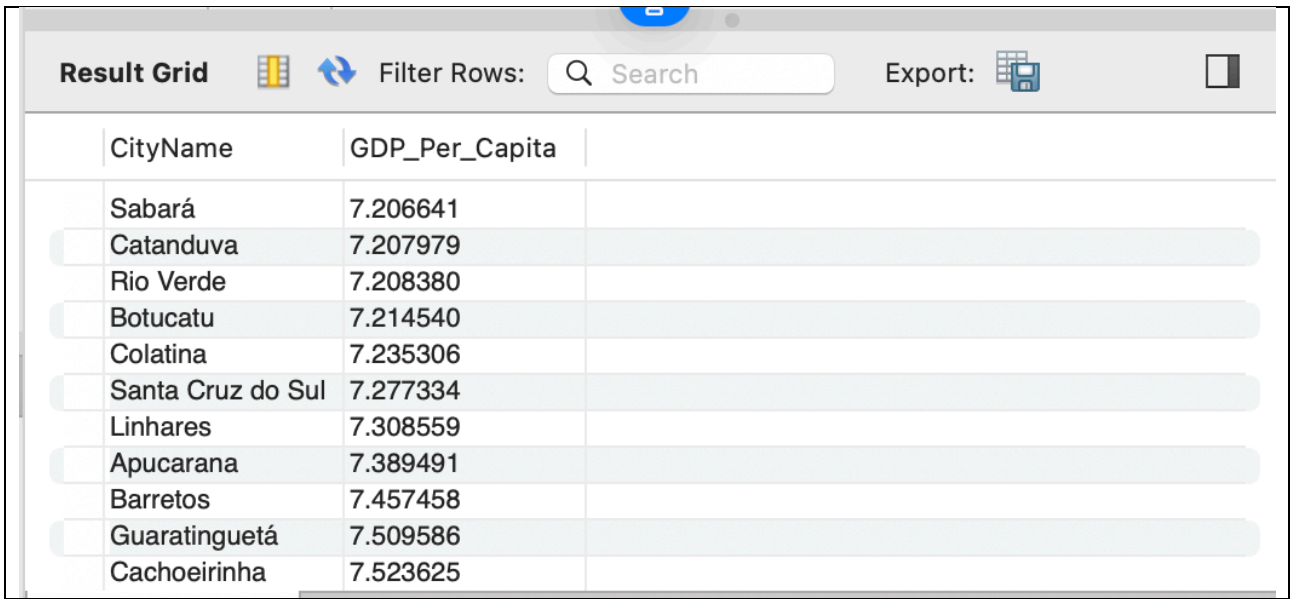
```
SELECT  country.Name AS CountryName ,
          city.Population
FROM city
JOIN country ON city.CountryCode = country.Code
ORDER BY city.Population ASC;
```

CountryName	Population
Pitcairn	42
Cocos (Keeling) Islands	167
Tokelau	300
Holy See (Vatican City State)	455
Cocos (Keeling) Islands	503
Nauru	559
Anguilla	595
Niue	682
Christmas Island	700
Norfolk Island	800
Anguilla	961

19. **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 CityName,
      (country.GNP / city.Population) AS GDP_Per_Capita
FROM city
JOIN country ON city.CountryCode = country.Code
WHERE (country.GNP / city.Population) > (
   SELECT AVG(country.GNP / city.Population)
   FROM city
   JOIN country ON city.CountryCode = country.Code
   WHERE city.Population > 0
)
AND city.Population > 0;
```

Joining the two tables country and city twice, calculating the GDP per capita by dividing the GNP per population and then comparing the GDP by the average of GDP to get the cities above the average GDP.



Result Grid | Filter Rows: Search | Export:

CityName	GDP_Per_Capita
Sabará	7.206641
Catanduva	7.207979
Rio Verde	7.208380
Botucatu	7.214540
Colatina	7.235306
Santa Cruz do Sul	7.277334
Linhares	7.308559
Apucarana	7.389491
Barretos	7.457458
Guaratinguetá	7.509586
Cachoeirinha	7.523625

20. **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 AS CityName,
       city.Population,
       country.Name AS CountryName
FROM city
JOIN country ON city.CountryCode = country.Code
ORDER BY city.Population DESC
LIMIT 10 OFFSET 30;
```

Result Grid | Filter Rows: Search | Export: | Fetch rows:

CityName	Population	CountryName
Shenyang	4265200	China
Kanton [Guangzhou]	4256300	China
Singapore	4017733	Singapore
Ho Chi Minh City	3980000	Vietnam
Chennai (Madras)	3841396	India
Pusan	3804522	South Korea
Los Angeles	3694820	United States
Dhaka	3612850	Bangladesh
Berlin	3386667	Germany
Rangoon (Yangon)	3361700	Myanmar

Result Grid | Form Editor | Field Types

Result 2 | Read Only

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.