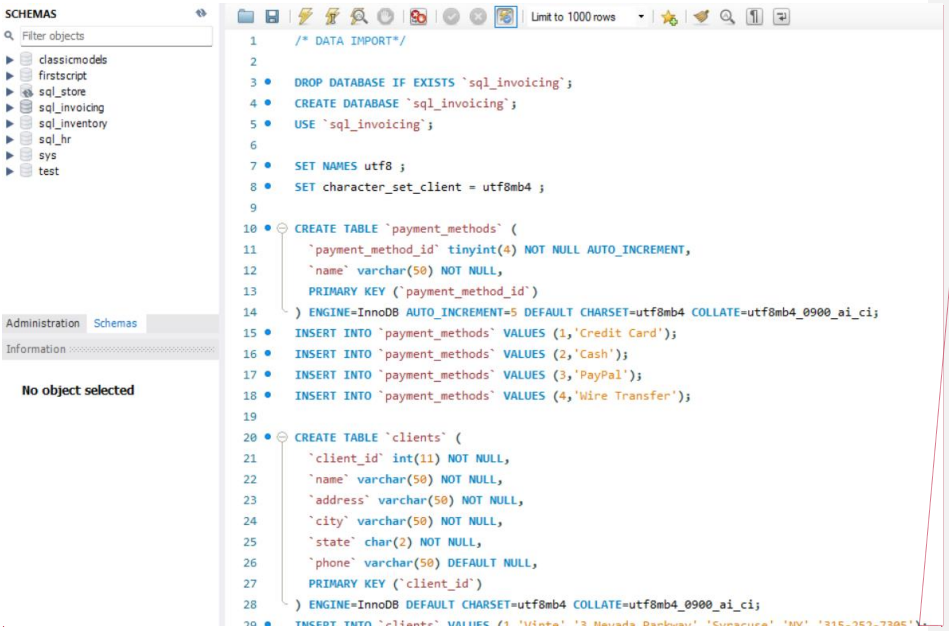
[DOCUMENT TITLE]

[Document subtitle]

Table of Contents:

MySQL Part 1:



Commented [MS1]: Data Import:
I downloaded the database and ran the script in MySQL, creating the databases.

```sql
/* QUERY 1 */

USE sql_store;

SELECT *
FROM customers;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Commented [MS2]: Query 1:
This input shows all the columns in the customers table in the 'sql_store' database.

```sql
/* QUERY 1 */

USE sql_store;

SELECT *
FROM customers
-- where customer_id=1
ORDER BY first_name;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Commented [MS3]: Query 1 continued:
As the task instructed, I input the following under Query 1. This orders the first names of customers in ascending order automatically.

```sql
/* QUERY 2 */

SELECT last_name, first_name, points, points + 10
FROM customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

last_name	first_name	points	points + 10
MacCaffrey	Babara	2273	2283
Brushfield	Ines	947	957
Boagey	Freddi	2967	2977
Roseburgh	Ambur	457	467
Betchley	Clemmie	3675	3685
Twiddell	Elka	3073	3083
Dowson	Ilene	1672	1682
Naseby	Thacher	205	215
Rumgay	Romola	1486	1496
Mynett	Levy	796	806

Commented [MS4]: Query 2:
This shows the columns for first and last names, along with their current points. Another column has been created to add on 10 additional points to the customers.

```sql
/* QUERY 2 */

SELECT last_name, first_name, points, points + 10
FROM customers;

-- Task 1:
SELECT last_name, first_name, points, (points * 10) + 100
FROM customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

last_name	first_name	points	(points * 10) + 100
MacCaffrey	Babara	2273	22830
Brushfield	Ines	947	9570
Boagey	Freddi	2967	29770
Roseburgh	Ambur	457	4670
Betchley	Clemmie	3675	36850
Twiddell	Elka	3073	30830
Dowson	Ilene	1672	16820
Naseby	Thacher	205	2150
Rumgay	Romola	1486	14960
Mynett	Levy	796	8060

Commented [MS5]: Query 2, Task 1:
The result of the instructions show the new column name as the command.

Limit to 1000 rows

```sql
/* QUERY 2 */

SELECT last_name, first_name, points, points + 10
FROM customers;

-- Task 1:
SELECT last_name, first_name, points, (points * 10) + 100
FROM customers;

-- Task 1.2:
SELECT last_name, first_name, points, (points + 10) * 100 AS discount_factor
FROM customers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

last_name	first_name	points	discount_factor
MacCaffrey	Babara	2273	228300
Brushfield	Ines	947	95700
Boagey	Freddi	2967	297700
Roseburgh	Ambur	457	46700
Betchley	Clemmie	3675	368500
Twiddell	Elka	3073	308300
Dowson	Ilene	1672	168200
Naseby	Thacher	205	21500
Rumgay	Romola	1486	149600
Mynett	Levy	796	80600

Commented [MS6]: Query 2, Task 1.2:
Now I have given the column a relevant name by giving the code a table alias 'discount_factor' using 'AS'.
The points calculation has also been altered as per the task instructions.

```sql
/* TASK 2 */

USE sql_store;

SELECT name, unit_price AS original_price, (unit_price * 1.1) AS new_price
FROM products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	original_price	new_price
Foam Dinner Plate	1.21	1.331
Pork - Bacon,back Peameal	4.65	5.115
Lettuce - Romaine, Heart	3.35	3.685
Brocolinni - Gaylan, Chinese	4.53	4.983
Sauce - Ranch Dressing	1.63	1.793
Petit Baguette	2.39	2.629
Sweet Pea Sprouts	3.29	3.619
Island Oasis - Raspberry	0.74	0.814
Longan	2.26	2.486
Broom - Push	1.09	1.199

Commented [MS7]: Task 2:
This query shows the all the products in our database with their original price and their new price - the new price being an increase of 10%.

```sql
/* Task 3 */

USE sql_store;

SELECT first_name, last_name, birth_date
FROM customers
WHERE birth_date > '1990-01-01';
```

Result Grid | Filter Rows: | Export:

first_name	last_name	birth_date
Elka	Twiddell	1991-09-04
Thacher	Naseby	1993-07-17
Romola	Rumgay	1992-05-23

Commented [MS8]: Task 3:
This query finds and shows the customers who were born after the date 1990-01-01. 3 customers were returned.

```sql
/* TASK 4 */

USE sql_inventory;

SELECT name, quantity_in_stock
FROM products
ORDER BY quantity_in_stock DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Expo

	name	quantity_in_stock
▶	Sweet Pea Sprouts	98

Commented [MS9]: Task 4:
Now using the 'sql_inventory' database, I used this query to find the name of the product with the highest amount of stock.
(To improve readability and clarity, I returned 1 row after ordering the 'quantity_in_stock' in descending order to ensure the highest value shows).

```sql
/* TASK 5 */

USE sql_inventory;

SELECT name, unit_price
FROM products
ORDER BY unit_price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	name	unit_price
▶	Pork - Bacon,back Peameal	4.65

Commented [MS10]: Task 5:
Again, using the 'sql_inventory' database, I found the most expensive product.
(Same method was applied as Task 4, I limited it to return 1 row and ensured the highest price shows by ordering the 'unit_price' in descending order).

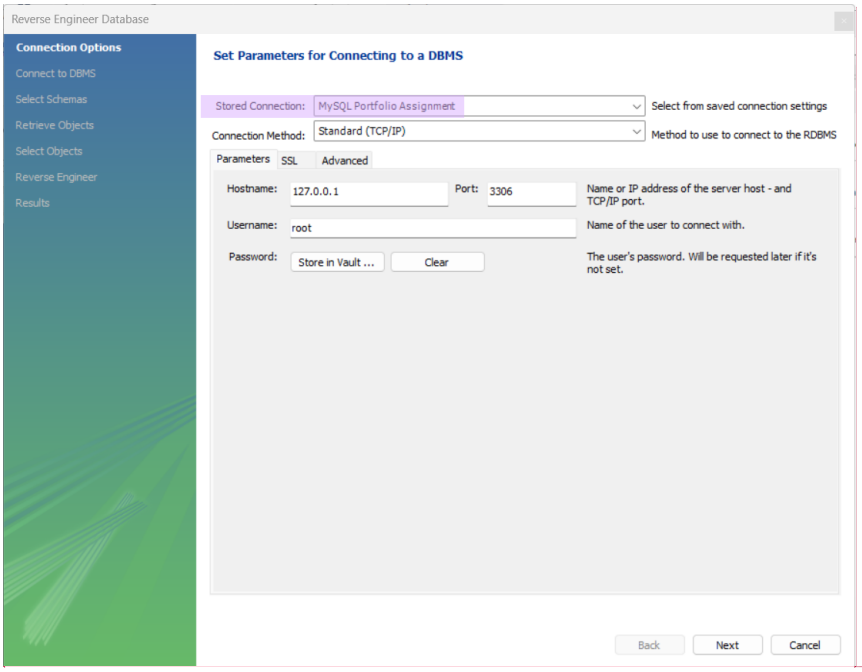
```sql
/* TASK 6 */

USE sql_store;

SELECT first_name, last_name, address, birth_date
FROM customers
ORDER BY birth_date ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conter

first_name	last_name	address	birth_date
Ilene	Dowson	50 Lillian Crossing	1964-08-30

Commented [MS11]: Task 6:
I queried to use the database 'sql_store' then queried to find the information of the oldest customer.
As per instruction, the columns first and last name, address and birth_date are shown.

EER Diagram:



Commented [MS12]: Reverse Engineer Database.
I went to the 'Database' menu and selected ' Reverse Engineer' in the dropdown.
The highlighted area in the image shows the database connection I chose.

Connect to DBMS and Fetch Information

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

- Connect to DBMS
- Retrieve Schema List from Database
- Check Common Server Configuration Issues

Execution Completed Successfully

Fetch finished.

Commented [MS13]: This is one of many windows that confirm the actions are complete.

Select Schemas to Reverse Engineer

Select the schemas you want to include:

- [ ] classicmodels
- [ ] firstscript
  - [ ] sql_hr
  - [ ] sql_inventory
  - [ ] sql_invoicing
- [x] sql_store
  - [ ] test

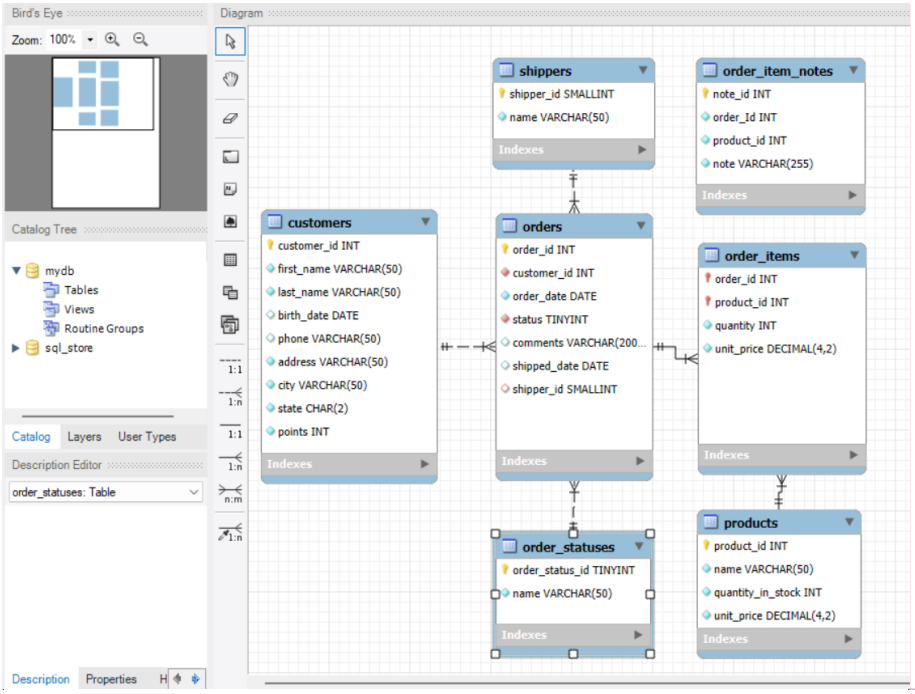
Commented [MS14]: I have selected 'sql_store' schema as seen on the left.

Select Objects to Reverse Engineer

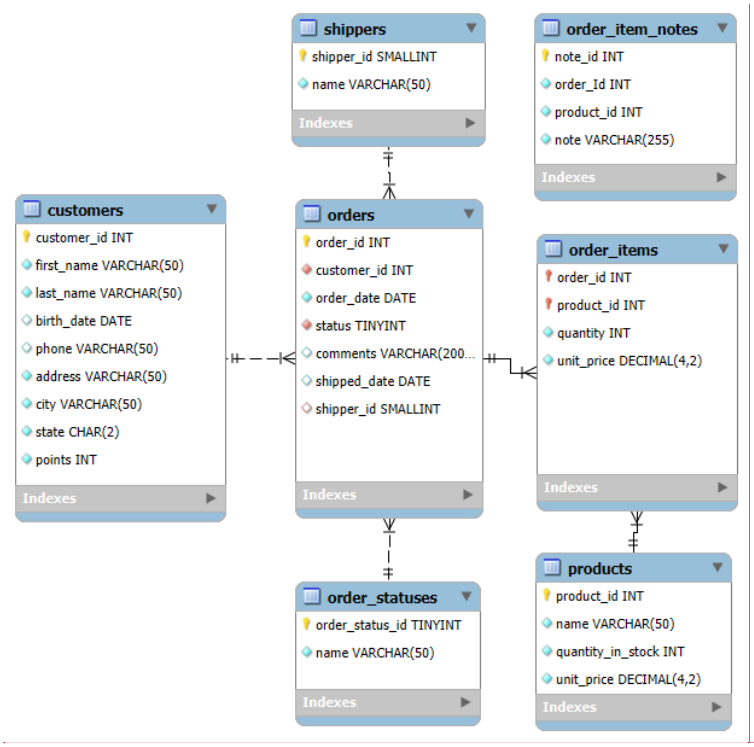
- [x] Import MySQL Table Objects

7 Total Objects, 7 Selected

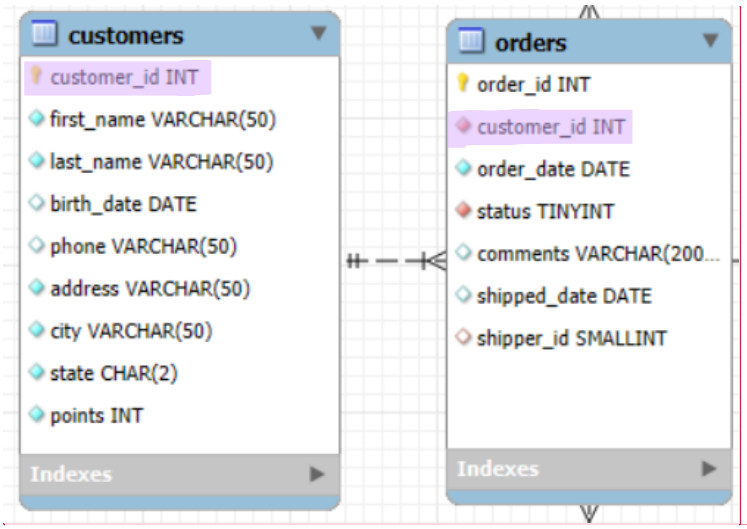
Show Filter



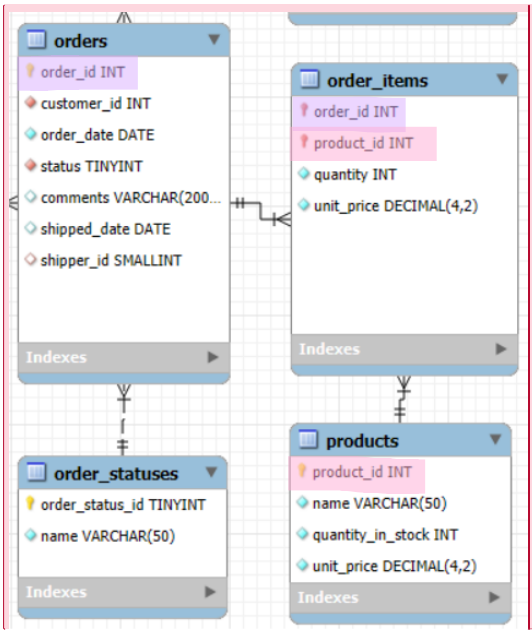
Commented [MS15]: EER Diagram:
This is the rearranged EER Diagram (Enhanced Entity-Relationship) and its view in MySQL Workbench.

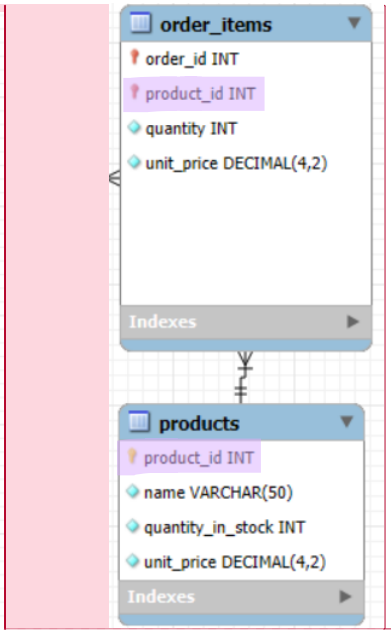


Commented [MS16]: I have exported the model as a PNG and attached it here.



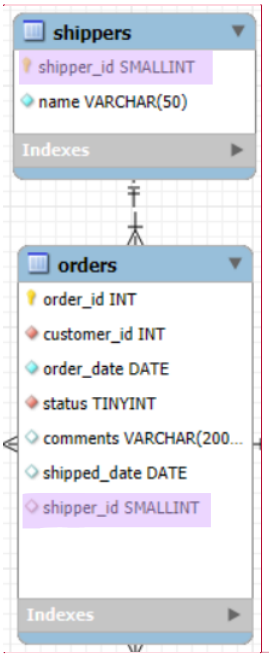
Commented [MS17]: Customers Table:
- Stores customer details, such as customer_id, first_name, last_name, birth_date, etc.
- Primary key: customer_id
- Relationships: Linked to the orders table by customer_id as a foreign key.

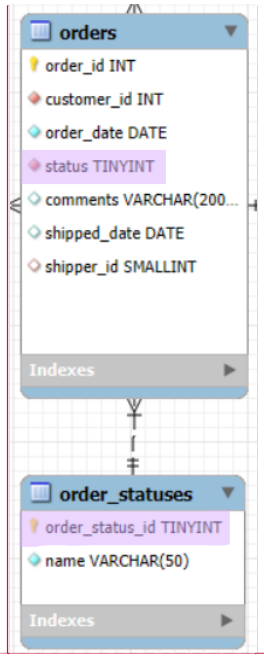




Commented [MS18]: Order_Items Table:
- Contains item details, such as order_id, product_id, quantity, and unit_price.
- Primary Key: order_id and product_id
- Relationships (Foreign Keys): order_id links to the orders table.
product_id links to the products table.

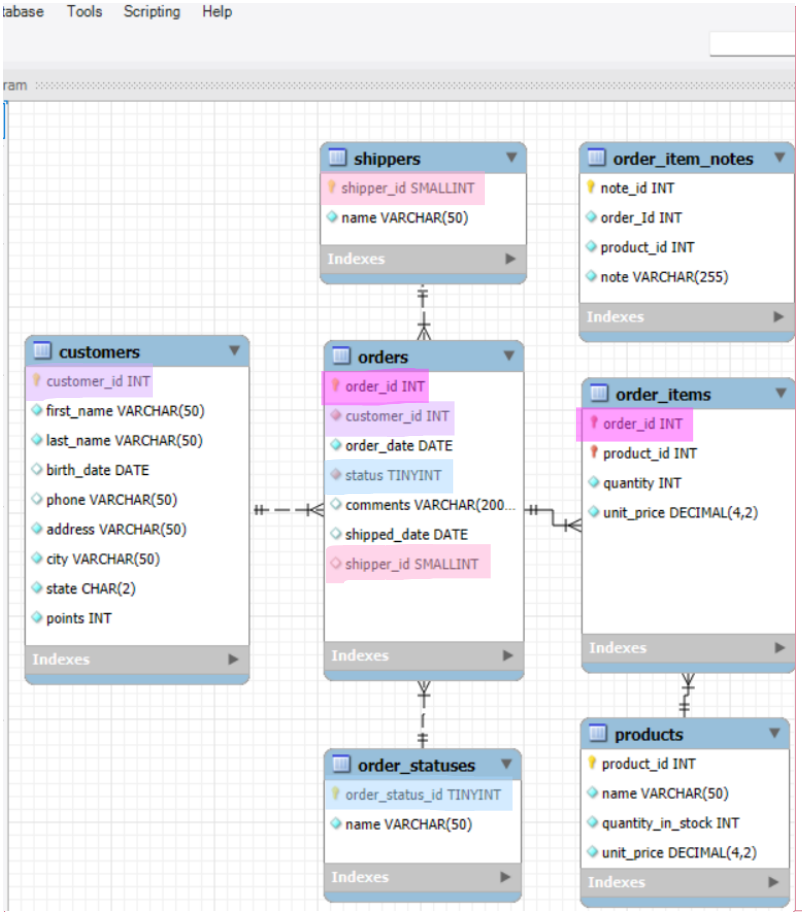
Commented [MS19]: Products Table:
- Contains product details e.g. product_id, name, quantity_in_stock, and unit_price.
- Primary Key: product_id
- Relationships: Linked to order_items through product_id as a foreign key.





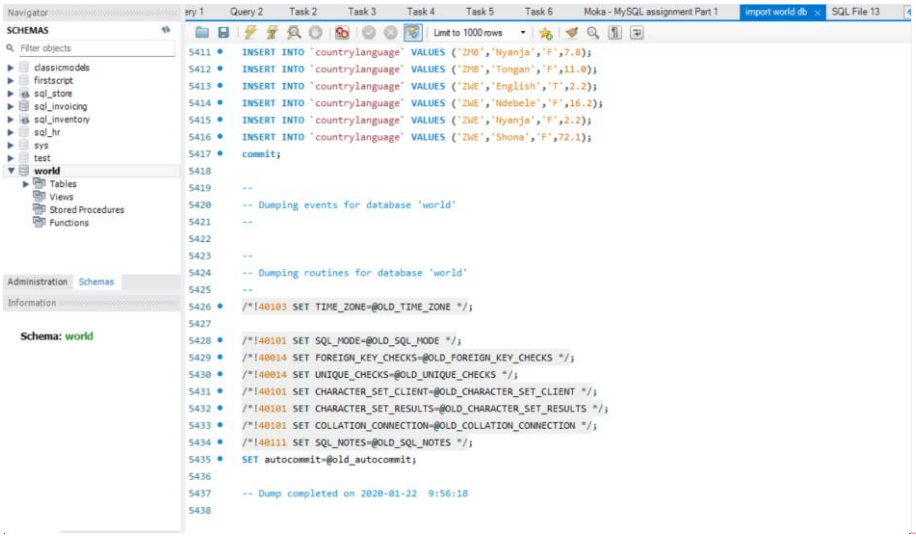
Commented [MS20]: Shippers Table:
- Contains shipper information, shipper_id and name.
- Primary Key: shipper_id
- Relationships: Linked to orders by shipper_id as a foreign key.

Commented [MS21]: Order_Statuses Table:
- Stores the status of orders, with columns order_status_id and name.
- Primary Key: order_status_id
- Relationships: Linked to orders by status as a foreign key.

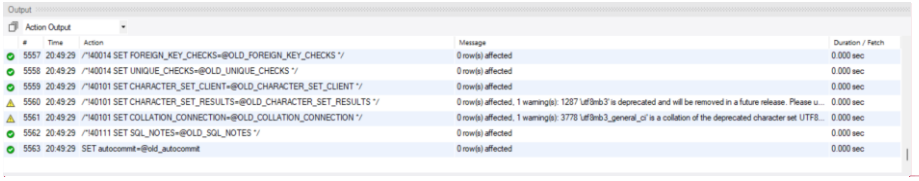


Commented [MS22]: Orders Table:
- Contains order information, including order_id, customer_id, order_date, status, and shipper_id.
- Primary Key: order_id
- Relationships:
customer_id links to the customers table.
shipper_id links to the shippers table.
order_statuses by status to define the order's status.

My SQL Part 2:



Commented [MS23]: Data Import:
I downloaded the database and ran the script in MySQL, creating the databases.



Commented [MS24]: Action 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.

```
#Task 1

SELECT COUNT(*) AS total_cities_in_USA
FROM city
WHERE CountryCode = 'USA';
```

Result Grid | Filter Rows: | Export:

total_cities_in_USA
274

Commented [MS25]: *Result:*
The total number of cities within the Unites States (USA) is 274.

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 prioritizing healthcare resources and interventions.

```
#Task 2

SELECT Name AS country, LifeExpectancy
FROM country
ORDER BY LifeExpectancy DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export:

country	LifeExpectancy
Andorra	83.5

Commented [MS26]: *Result:*
The country with the highest life expectancy is Andorra.

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 travelers with exciting destinations to kick off the New Year in style.

```
#Task 3

SELECT Name AS city, CountryCode
FROM city
WHERE Name LIKE '%New%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city	CountryCode
Newcastle	AUS
Newcastle upon Tyne	GBR
Newport	GBR
Newcastle	ZAF
Kowloon and New Kowloon	HKG
New Bombay	IND
New Delhi	IND
Khanewal	PAK

city 17

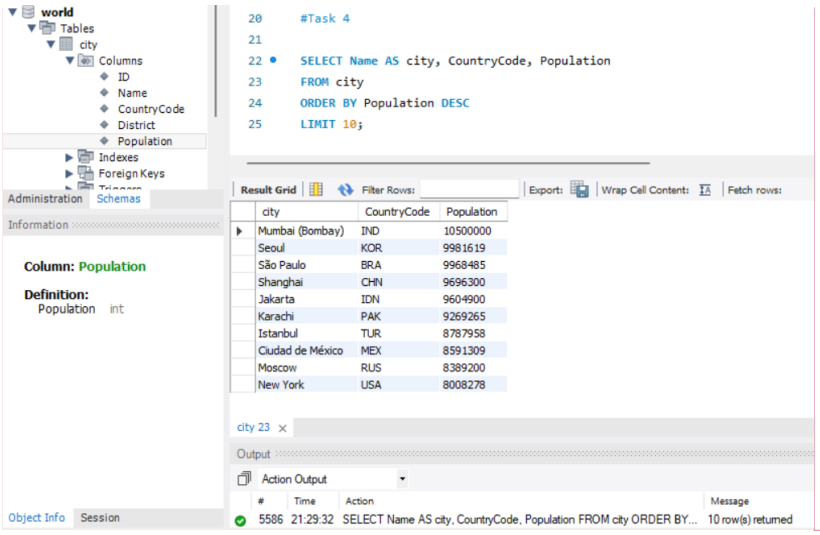
Output

Action Output

#	Time	Action	Message
5580	21:23:43	SELECT Name AS city, CountryCode FROM city WHERE Name LIK...	14 row(s) returned

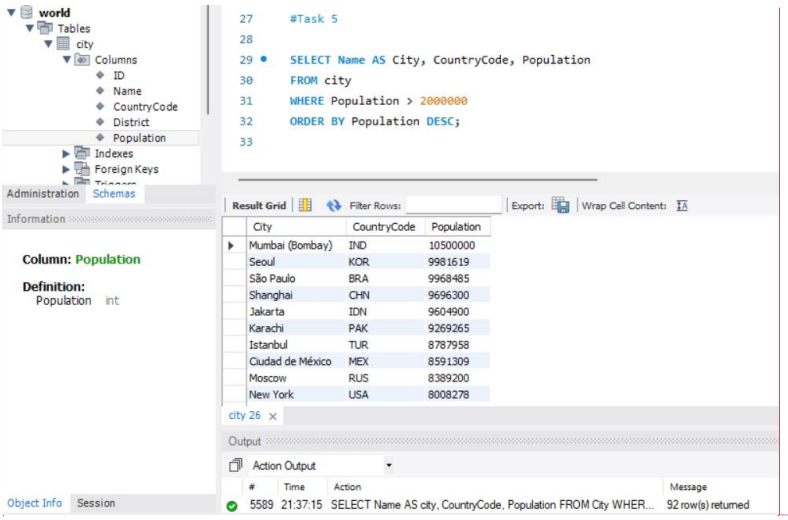
Commented [MS27]: *Result:*
This is a query that found a list of ciries with names that include the word 'New'.
14 cities were selected.

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.



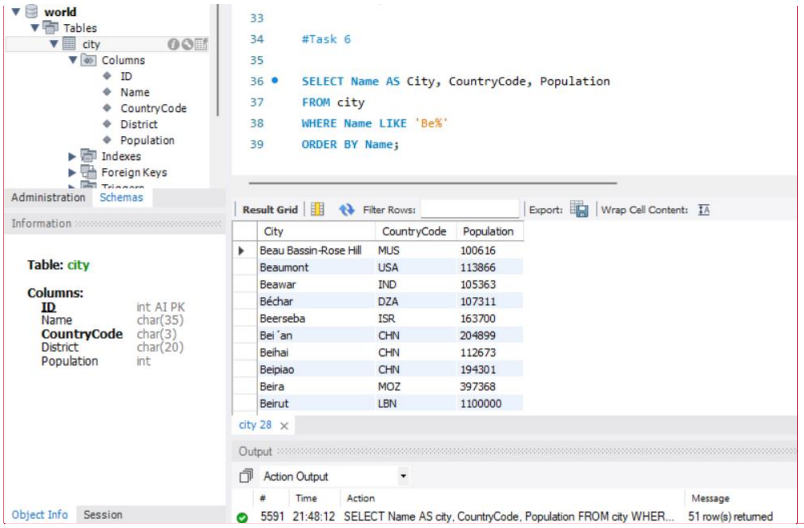
Commented [MS28]: *Result:*
Here is a brief overview of the top 10 most populated cities in the world.
(P.S: I have included the Navigator sidebar in this screenshot to show that I have been dragging and dropping columns into my code)

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.



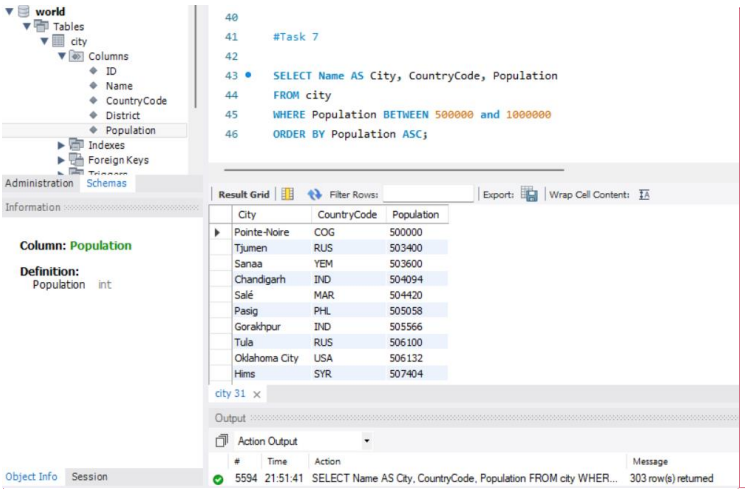
Commented [MS29]: *Result:*
92 cities exceed over 2 million in population size.

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 starts with the prefix 'Be' to assist in the blogger's content creation process.



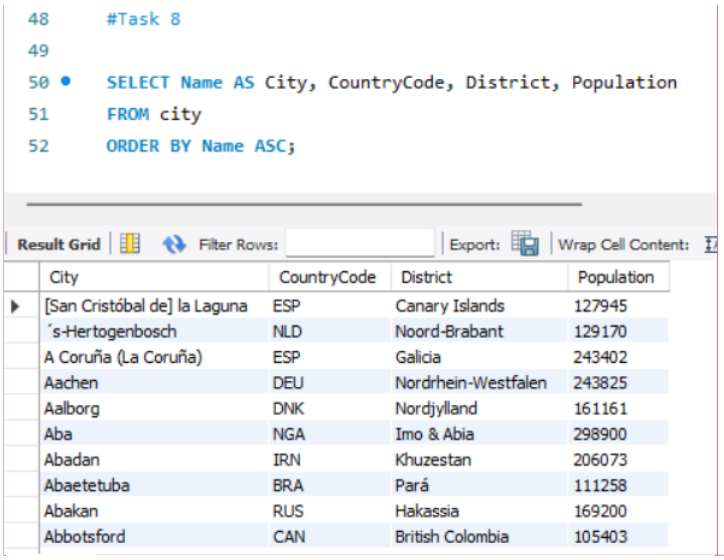
Commented [MS30]: *Result:*
A number of 51 cities starts with the prefix 'Be'.

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.



Commented [MS31]: *Result:*
This query returns a list of cities which population range from 500,000 and 1 million.

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.



Commented [MS32]: *Result:*
This is a list of city information ordered in ascending order of city names for the geography lesson.

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.

```
#Task 9

SELECT Name AS City, Population
FROM city
ORDER BY Population DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Expo

City	Population
Mumbai (Bombay)	10500000

Commented [MS33]: *Result:*
The most populated city is Mumbai (Bombay) with a population of 10,500,000.

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's lesson.

```
#Task 10

SELECT name AS City, COUNT(*) AS Frequency
FROM city
GROUP BY name
ORDER BY name ASC;
```

Result Grid | Filter Rows: | Export: | Wra

City	Frequency
[San Cristóbal de] la Laguna	1
´s-Hertogenbosch	1
A Coruña (La Coruña)	1
Aachen	1
Aalborg	1
Aba	1
Abadan	1
Abaetetuba	1
Abakan	1
Abbotsford	1
Abeokuta	1
Aberdeen	1

Commented [MS34]: *Result:*
This is a list of unique names of cities in alphabetical order.

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.

```
#Task 11

SELECT Name AS City, Population
FROM city
ORDER BY Population ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Exp

City	Population
Adamstown	42

Commented [MS35]: Result:
The city with the lowest population is Adamstown with a population of 42.

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.

```
#Task 12

SELECT Name AS Country, Population
FROM country
ORDER BY Population DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export:

Country	Population
China	1277558000

Commented [MS36]: *Result:*
The country with the highest population is China with a population of 1,277,558,000.

13. **Capital of Spain:** *Scenario:* A travel agency is organizing 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 travelers with essential destination information.

```
#Task 13

SELECT city.CountryCode, country.name AS country, city.name AS city
FROM country
INNER JOIN city
ON city.CountryCode = country.code
WHERE country.code = "ESP"
ORDER BY city.population DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CountryCode	country	city
ESP	Spain	Madrid

Commented [MS37]: *Work process notes:*
- As there is no capital city column in either city or country tables, assuming the capital is the most populated city of Spain, I ordered by population in descending order to find the capital.

(see next comment & image)

```
#Task 13

SELECT city.CountryCode, country.name AS Country, city.name AS CapitalCity
FROM country
JOIN city
ON city.ID = country.capital
WHERE country.name = 'Spain';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CountryCode	Country	CapitalCity
ESP	Spain	Madrid

Commented [MS38]: Work process notes:
- The previous query relied on assuming the capital was the most populated city, however that is not always the case so it cannot be applied if it were another country.
- Therefore, looking at the table again, the Capital column (in country table) matched the ID column (in city table) so it could be joined. I tested this with other countries.

Result:
The capital of Spain is Madrid

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.

Duplicate - Same scenario as Task 2 above.

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.

```
#Task 15

SELECT city.name, continent
FROM city
JOIN country
ON country.code = city.countrycode
WHERE continent = 'Europe'
ORDER BY city.name;

-- join tables experimentation:
SELECT * FROM city, country;
SELECT * FROM city CROSS JOIN country ON country.code = city.CountryCode;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	continent
[San Cristóbal de] la Laguna	Europe
´s-Hertogenbosch	Europe
A Coruña (La Coruña)	Europe
Aachen	Europe
Aalborg	Europe
Abakan	Europe
Aberdeen	Europe
Aix-en-Provence	Europe
Albacete	Europe
Alcalá de Henares	Europe

Commented [MS39]: *Work process notes:*
- I joined the tables together using the country code.

Result:
These are cities located in 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.

```sql
#Task 16

SELECT name, AVG(population) AS average_population
FROM country
GROUP BY name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

name	average_population
Aruba	103000.0000
Afghanistan	22720000.0000
Angola	12878000.0000
Anguilla	8000.0000
Albania	3401200.0000
Andorra	78000.0000
Netherlands Antilles	217000.0000
United Arab Emirates	2441000.0000

Commented [MS40]: *Result:*
I used the AVG aggregate function to find the average population across countries, however it was hard to read because of the decimals.
(Read next comment).

```sql
#Task 16

SELECT name, AVG(population) AS average_population
FROM country
GROUP BY name;

-- (To improve readability: rounded to zero decimal places)
SELECT name AS Country, ROUND(AVG(population), 0) AS Average_Population
FROM country
GROUP BY name
ORDER BY average_population DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Country	Average_Population
China	1277558000
India	1013662000
United States	278357000
Indonesia	212107000
Brazil	170115000
Pakistan	156483000
Russian Federation	146934000
Bangladesh	129155000

Commented [MS41]: *Result:*
After some further learning, I found out that I can use the ROUND function to zero decimal places to improve readability.
(I also fixed the column names and put the numbers in descending order).

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.

```
/* ID = capital */

SELECT ID from city;
-- PRIMARY KEY

SELECT capital FROM country;
-- FOREIGN KEY
```

```
#Task 17

SELECT country.name AS Country, city.name AS CapitalCity, city.population AS CityPopulation
FROM country
JOIN city
ON city.ID = country.capital
ORDER BY city.population DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Country	CapitalCity	CityPopulation
South Korea	Seoul	9981619
Indonesia	Jakarta	9604900
Mexico	Ciudad de México	8591309
Russian Federation	Moscow	8389200
Japan	Tokyo	7980230
China	Peking	7472000
United Kingdom	London	7285000
Egypt	Cairo	6789479
Iran	Teheran	6758845
Peru	Lima	6464693
Thailand	Bangkok	6320174
Colombia	Santafé de Bogotá	6260862
Congo, The Demo...	Kinshasa	5064000

Commented [MS42]: *Result:*
I found the capital cities by joining tables and ordered it by city population in descending order to improve readability.

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.

```sql
#Task 18

SELECT name AS country, population, surfacearea, (population / surfacearea) AS population_density
FROM country
WHERE (population / surfacearea) < 25
ORDER BY population_density DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	population	surfacearea	population_density
Mozambique	19680000	801590.00	24.5512
Kyrgyzstan	4699000	199900.00	23.5068
Laos	5433000	236800.00	22.9434
Bahamas	307000	13878.00	22.1213
Congo, The Democratic ...	51654000	2344858.00	22.0286
Chile	15211000	756626.00	20.1037
Peru	25662000	1285216.00	19.9671
Brazil	170115000	8547403.00	19.9025
Sweden	8861400	449964.00	19.6936
Saint Helena	6000	314.00	19.1083
Uruguay	3337000	175016.00	19.0668
Christmas Island	2500	135.00	18.5185
Equatorial Guinea	453000	28051.00	16.1492
Somalia	10097000	637657.00	15.8345
Vanuatu	190000	12189.00	15.5878
Solomon Islands	444000	28896.00	15.3654

Commented [MS43]: *Result:*
I calculated the population density, and ordered it in descending order.
These are the countries with sparse populations to support the study for potential agricultural development projects.

For an agricultural research institute looking at countries for potential development, a threshold of **25 people per square kilometer** could be ideal. Here's why:

- **Viability for Agriculture**: Areas with fewer than 25 people per square kilometer are typically rural and have more open land, which is essential for agricultural projects.
- **Avoiding Extremely Sparse Options**: Setting the threshold at 25 captures sparsely populated countries that are still accessible and may already have some basic infrastructure, making them more practical for development than extremely remote areas.
- **Targeting Usable Land**: Going much lower (like 10) might yield very isolated or inhospitable regions, while 25 gives a balance between sparsity and potential for growth.

So a threshold of **25** would focus on countries with available space but still within reach for potential agricultural initiatives.

BONUS TASKS:

19. **Cities with High GDP per Capita:** *Scenario:* An economic consulting firm is analyzing 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.

Commented [MS44]: *Work process notes:*
- Firstly, I queried to look at the city and country tables to look at the values stored in them. I found that there's GNP instead of GDP.
- I went through a couple errors when trying to pull data from tables. But then I noticed the city table had a country code column so I used the join clause.

```
-- column info pull:
SELECT *
from country;
SELECT *
from city;

SELECT city.name AS city, ROUND(AVG(country.GNP), 0) AS average_gnp
FROM city
JOIN country ON city.CountryCode = country.code
GROUP BY city.name
ORDER BY average_gnp DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

city	average_gnp
Chicago	8510700
Phoenix	8510700
San Antonio	8510700
San Francisco	8510700
Austin	8510700
Milwaukee	8510700

Commented [MS45]: *Work process notes:*
- I had extra decimal places, so I applied rounding for readability.
- Lastly, I put the avg. GNP column in descending order **However, this did not provide correct insight for the firm, as it wasn't showing only above-average.** (Read next comment).

```sql
-- correction
SELECT city.name as city, country.GNP AS average_GNP
FROM city, country
WHERE GNP > (SELECT AVG(GNP) FROM country)
ORDER BY GNP;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

city	average_GNP
Ciomas	137635.00
Sawhaj	137635.00
Embu	137635.00
Guaymallén	137635.00

Commented [MS46]: *Work process notes:*
- I used the WHERE clause to filter the data so the GNP returned is over the average calculated GNP.
- Then I ordered the column in ascending order.

Result:
The resulting cities are above-average in GNP.

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.

```
#Task 20
SELECT *
FROM city
ORDER BY Population DESC
LIMIT 10 OFFSET 30;
```

Result Grid | Filter Rows: | Edit: | Export/Import

ID	Name	CountryCode	District	Population
1896	Shenyang	CHN	Liaoning	4265200
1897	Kanton [Guangzhou]	CHN	Guangdong	4256300
3208	Singapore	SGP	–	4017733
3769	Ho Chi Minh City	VNM	Ho Chi Minh City	3980000
1027	Chennai (Madras)	IND	Tamil Nadu	3841396
2332	Pusan	KOR	Pusan	3804522
3794	Los Angeles	USA	California	3694820
150	Dhaka	BGD	Dhaka	3612850
3068	Berlin	DEU	Berliini	3386667
2710	Rangoon (Yangon)	MMR	Rangoon [Yangon]	3361700
NULL	NULL	NULL	NULL	NULL

city 24

Output

Action Output

#	Time	Action	Message
25	13:18:29	SELECT * FROM city ORDER BY Population DE...	10 row(s) returned

Commented [MS47]: *Work process notes:*
- I did some research on how to skip the first top 30 ranked and found out about OFFSET clause which allows this.
- I limited the returned rows by 10 as the firm wants from ranked between 31st to 40th.

Result:
This is the details for cities beyond the top rankings of population for cities, in descending order.

Task 3 – Interview Part 1:

1. **What is a Query?**

A command or instruction to retrieve, insert, update, or delete data in a database.

2. **What is the SELECT statement?**

The select statement is used to return data from a database in the output, specifically columns from a table.

3. **What is the WHERE clause?**

It's a clause that filters tables to get data that meets specific conditions.

4. **What is the Primary key?**

A unique identifier for a column that helps identify each row. It makes sure evert row is unique and doesn't contain null values.

5. **What is a Database**

A collection of data that can be managed, updated and queried. The data is stores in tables with rows and columns.

Task 4 – Interview questions Part 2:

1. **List the different types of relationships in SQL and give examples.**

Primary Key – Uniquely identifies each row in a table. It ensures that each record is unique and cannot have null values.

E.g. Students table had a unique StudentID as primary Key – This uniquely identifies each student individually, and all the columns in that table will be linked to the StudentID primary key.

Foreign Key – Creates a relationship between tables by linking a column in one table to a primary key in another table.

E.g. A table for Authors with AuthorID as primary key, listing information about individual authors. A table for Books may reference the book's authors using AuthorID as a foreign key.

2. **What is Normalization?**

Normalization is organizing data in a database to reduce redundancy, ensures data integrity and improves efficiency. It follows a process (stages of normal forms) which structures the database, so it is easier to query and maintain.

3. **Model query to show the population of Germany:**

SELECT population

FROM world

WHERE name = ‘Germany’;

4. **Select the query which gives the name of countries beginning with U.**

SELECT name

FROM world

WHERE name LIKE ‘U%’;

5. **Select the answer which shows the problem with this SQL code – the intended result should be the continent of France:**

SELECT continent

FROM world

WHERE ‘name’ = ‘France’

b) ‘name’ should be name

6. **Select the code which shows the countries that end in A or L.**

SELECT name

FROM world

WHERE name LIKE ‘%a’ OR name LIKE ‘%l’;

7. **Given the table on the left, select the query which produces this table on the right.**

name	region	area	population	gdp
Afghanistan	South Asia	652225	26000000	
Albania	Europe	28728	3200000	6656000000
Algeria	Middle East	2400000	32900000	75012000000
Andorra	Europe	468	64000	
...				

name	population
Bahrain	1234571
Swaziland	1220000
Timor-Leste	1066409

SELECT name, population

FROM world

WHERE population BETWEEN 100000 AND 1250000;