

SQL qUERY ASSIGNMENT

A look into the power of querying relational databases using T-SQL



03 May 2024

JUST it

[Company address]

This project showcases my ability to write SQL queries using Transact SQL to pull up data from the relational database AdventureWorks2022. I used Azure Data Studios in a Skillable Labs virtual machine to write these SQL queries.

**Tasks**

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( Name) AS [Number of Cities in USA]

FROM dbo.city

WHERE CountryCode LIKE 'USA';

I wanted to use AS to name the field. I used FROM to retrieve data from dbo.city table and add a condition for CountryCode column only bring “USA” string using WHERE clause and LIKE operator for pattern matching. I can also use = to determine an exact match.

A screenshot of a computer

Description automatically generated

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 prioritizing healthcare resources and interventions.

SELECT TOP 1 Name AS [Country], LifeExpectancy as [Life Expectancy]

FROM dbo.country

ORDER by [Life Expectancy] DESC;

Looking for the highest life expectancy required me to use dbo.country table and the TOP clause paired with 1 to bring the first row. I referred the Name column as “Country” and LifeExpectancy column as “Life Expectancy”. I then ordered the “Life Expectancy” in descending order using ORDER clause and the DESC keyword to ensure the top row displayed the highest value.

A screenshot of a computer

Description automatically generated

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.

SELECT Name

FROM dbo.city

WHERE Name LIKE '%New%';

The aim here was to bring up all names that had “New” in it, so I used the WHERE clause and LIKE operator. I then included % on both ends of the string to indicate that it should contain the characters in that order. LIKE is useful here as it can partially match or include wildcard searches.

A screenshot of a computer

Description automatically generated

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.

SELECT TOP 10 \*

FROM dbo.city

ORDER BY Population DESC;

To only retrieve the first 10 cities, I had to use TOP 10 and I included \* to retrieve all the fields for each record that belonged to the cities. I ordered the Population column in descending order to display all the highest values.

A screenshot of a computer

Description automatically generated

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.

SELECT Name AS [City], Population

FROM dbo.city

WHERE Population > 2000000

ORDER BY Population;

I needed to only retrieve names of cities and respective population values exceeding 2 million. So I began by including both Name, which I relabelled as “City” and Population column in SELECT clause. I used > to indicate greater than with 2000000 in digit form in the WHERE clause. I also ordered the list using Population so it was easier to see changes in the values.

A screenshot of a computer

Description automatically generated

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.

SELECT Name AS [City]

FROM dbo.city

WHERE Name LIKE 'Be%'

ORDER BY [City];

For the blogger, I wanted to display just a list of cities and called the Name column as “City”. Using WHERE, I created a condition that the entries will have “Be” at the start of the string, by placing the characters in front of %. This wildcard ensures all strings that are retrieved have “Be” at the start of the string. I ordered the list by “City” column.

A screenshot of a computer

Description automatically generated

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.

SELECT Name AS [City], Population

FROM dbo.city

WHERE Population BETWEEN 500000 AND 1000000

ORDER BY Population DESC;

Since the committee wanted only city names and population values I relabelled Name column as “Citty” and included Population in my SELECT clause. I used BETWEEN keyword to indicate that I wanted the values to 500000 and 1000000 which were included in digit form. I also ordered the list by Population and descending order to better display the values.

A screenshot of a computer

Description automatically generated

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.

SELECT Name AS [City]

FROM dbo.city

ORDER BY [City];

The teacher only wanted a list of city names I only included Name and relabelled it as “City” from the city table. I ordered the list by “City” in ascending order by not including any keywords. As the default is ascending order the list printed was in alphabetical order as the teacher requested.

A screenshot of a computer

Description automatically generated

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.

SELECT TOP 1 Name AS [City], Population

FROM dbo.city

ORDER BY Population DESC;

To show the most populated cities I needed to order the Population column in descending order using DESC as it is all values, so the highest value appears at the top. As the real estate investment firm wanted the most populated, I then used TOP 1 with Name column labelled as “City” to only pull the first row. If they wanted more rows I would use TOP 5 or TOP 10 to showcase top 5 or top 10 respectively.

A screenshot of a computer

Description automatically generated

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's l

SELECT Name AS [City], COUNT (Name) AS [Frequency]

FROM dbo.city

GROUP BY Name

ORDER BY [City];

The teacher wanted cities sorted alphabetically so I labelled the Name column as “City” and used COUNT keyword to work out the frequency of the city names. In this case I didn’t need to use DISTINCT to count multiple cities with the same name as it returned the same number of rows.

A screenshot of a computer

Description automatically generated

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.

SELECT TOP 1 Name AS [City], Population

FROM dbo.city

ORDER BY Population;

A screenshot of a computer

Description automatically generated

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.

SELECT TOP 1 Name AS [Country], Population

FROM dbo.country

ORDER BY Population DESC;

A screenshot of a computer

Description automatically generated

1. **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 travellers with essential destination information.

SELECT co.Name AS [Country], ci.Name AS [Capital City], co.Region, co.LocalName, co.HeadofState, ci.Population, co.GovernmentForm

FROM dbo.city AS ci

INNER JOIN dbo.country AS co

ON ci.ID = co.Capital

WHERE co.Name = 'Spain';

A screenshot of a computer

Description automatically generated

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.

SELECT TOP 1 Name As [Country with Highest Life Expectancy], LifeExpectancy as [Life Expectancy]

FROM dbo.country

ORDER BY [Life Expectancy] DESC;

A screenshot of a computer

Description automatically generated

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.

SELECT ci.Name as [City], co.Name as [Country], co.Region

FROM dbo.city AS ci

INNER JOIN dbo.country AS co

ON ci.Countrycode = co.Code

WHERE co.Region LIKE '%Europe%';

A screenshot of a computer

Description automatically generated

This code connects the country code field in City table to Code field in Country table.

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.

SELECT co.Name as [Country], AVG(ci.Population) AS[Average Population]

FROM dbo.city AS ci

INNER JOIN dbo.country AS co

ON ci.CountryCode = co.Code

GROUP BY co.Name, co.Code

ORDER BY [Average Population] DESC;

A screenshot of a computer

Description automatically generated

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.

SELECT ci.Name as [Capital City], co.Name AS [Country], ci.Population AS [Population]

FROM dbo.city AS ci

INNER JOIN dbo.country AS co

ON ci.ID = co.Capital

ORDER BY [Population] DESC;

A screenshot of a computer

Description automatically generated

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.

SELECT Name AS [Country], CAST(ROUND(SUM(Population/SurfaceArea),2) as DECIMAL (10,2)) AS [Population Density]

FROM dbo.country

GROUP BY Name

ORDER BY [Population Density]

A screenshot of a computer

Description automatically generated

BONUS TASKS: Challenge yourself: These are optional tasks. Feel free to skip.

1. **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.

There is GNP instead of GDP. To work out GNP per Capita I would have to divide GNP by country’s population.

SELECT ci.Name AS [City], CAST(ROUND(SUM(co.GNP \* 1000000 / co.Population), 2) AS DECIMAL(10, 2)) AS [GNP per CAPITA]

FROM dbo.city AS ci

INNER JOIN dbo.country AS co

ON ci.CountryCode = co.Code

WHERE co.Population != 0

GROUP BY ci.Name

HAVING CAST(ROUND(SUM(co.GNP \* 1000000 / co.Population), 2) AS DECIMAL(10, 2)) >

    (SELECT AVG(GNPCalculated)

    FROM (SELECT ROUND(SUM(co.GNP \* 1000000 / co.Population), 2) AS GNPCalculated

          FROM dbo.country AS co

          WHERE co.Population != 0

          GROUP BY co.Code) AS [Average])

ORDER BY [GNP per CAPITA];

Or

SELECT ci.Name AS [City], CAST(ROUND(SUM(co.GNP\*1000000 / co.Population,2) AS DECIMAL(10, 2)) AS [GNP per Capita]

FROM dbo.city AS ci

INNER JOIN dbo.country AS co

ON ci.CountryCode = co.Code

WHERE co.GNP IS NOT NULL

AND co.Population IS NOT NULL

AND co.Population > 0

AND co.GNP/co.Population > (SELECT AVG( FROM dbo.city AS ci

A screenshot of a computer

Description automatically generated

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.

SELECT Name, Population

FROM dbo.city

ORDER BY Population DESC

OFFSET 30 ROWS

FETCH NEXT 10 ROWS ONLY

A screenshot of a computer

Description automatically generated