Homework#5

ACS575 Database Systems, Spring 2024

Deadline: April 26

- Submit your homework as a single compressed file, labeled with YourLastName_YourFirstName_ACS575_HW5.zip.
- Your submission should be organized into four parts: Part I and Part II. Please ensure each section is clearly labeled within your file
- Within each section, clearly number your responses to correspond with the question numbers, for example, Part I Q1, Part II Q3(1), and so forth.

Part I. Data Warehouse and Analytical Queries

This part is focused on applying data warehousing concepts and performing analytical queries for OLAP. Oracle DMBS is recommended for these exercises. You may also use other RDMBS supporting analytical query functions.

0. Begin by downloading the provided "**country_dw.zip**" FILE. Inside this zip file, you will find "country_dw_schema.sql" which contains the definitions for your table schema.
After reviewing these schema definitions, execute the script file in your database account to set up the tables accordingly.
Additionally, the zip file contains data files: "region_data.sql" , "population_data.sql", "area_data.sql" and "country_fact_data.sql" . Run these files to populate your tables with data.

Regarding the database structure:

- The GDP attribute is used as the measure in the fact table. To retrieve this data, you will query the COUNTRY_FACT table.

  We consider three dimensions:

- The first dimension is REGION. The REGION table contains columns for 'cid' (country ID), 'country_name', 'region_name', and 3 different continent names, as categorized by the continent model found at the Wikipedia link, http://en.wikipedia.org/wiki/Continent#Separation_of_continents . Query the REGION table to access this data.
- The second dimension is POPULATION. As the population attribute holds continuous values, these have been discretized into quartiles for easier analysis. The top 25 percent of population values are tagged as '1' under the population ID ('pid') in the POPULATION table. The 'pid' for the subsequent top 25 percent is marked as '2', and so on. Query the POPULATION table for these details.
- The last dimension is AREA. Similar to population, the area attribute has been discretized using quartiles. The largest 25 percent values of the area are labeled as '1' under the area ID ('aid') in the AREA table. The 'aid' for the next quarter is '2', and so on. Query the AREA table to access this information.

1. Create a STAR schema diagram of the country data warehouse. Indicate the primary key of each relation and if any, foreign keys.

2. Illustrate the hierarchy structure within the REGION dimension.

For each of the following questions, except for Q7, construct and execute the SQL query as per the instructions provided, including the specific column headings. Additionally, ensure to submit the execution results.

3. Construct an SQL query to determine the average GDP per continent (labeled as continent5). Round the GPD value to no decimal places.

Continent5 Name	Average GDP

4. To explore the GDP within 'Euraisa', write an SQL query to **drill down** into the continent6 regions of 'Euraisa'. Round the GPD value to no decimal places.

Continent6 Name	Average GDP

5. Develop an SQL query to assign a rank to each country in 'America' (where continent6 equals 'America') based on its GDP, in descending order.

RankNo	RegionName	CountryName	GDP
1	…	…	…
2	…	…	…
…	…	…	…

6. Develop an SQL query to first partition the countries within 'America' by region, and then, within each region, assign a rank to each country based on its GDP, in descending order.

RankNo	RegionName	CountryName	GDP
…	…	…	…
…	…	…	…
…	…	…	…

7. Explain on the distinctions between the SQL query statements for Q5 and Q6, focusing on the differences in their approach to ranking countries by GDP within America.

8. Construct an SQL query to find the country with the highest GDP in each region of America, ensuring each identified country has the highest rank (Rank 1) in its region. Adjust the SQL statement from Q6 for this purpose.

RankNo	RegionName	CountryName	GDP
1	Central America and the Caribbean	Guatemala	33000000000
1	North America	United States	6738400000000
1	South America	Brazil	886300000000

9. Create an SQL query to find the country with the lowest GDP in each region of America by adapting the approach from Q8.

RegionName	CountryName	GDP
Central America and the Caribbean	Navassa Island	0
North America	Saint Pierre and Miquelon	66000000
South America	Falkland Islands (Islas Malvinas)	0

10. Use the CUBE function to write a single SQL statement that counts the number of countries by REGION (i.e., continent4 region) and POPULATION dimensions. Ensure your query result matches the provided example, including column headings. A CASE WHEN clause may be necessary for displaying "SUB TOTAL" value as a derived value.

Region	Population Range	Number of Counties
Afro-Eurasia	10647511 - 2826683	55
Afro-Eurasia	2826683 - 72751	35
Afro-Eurasia	< 72751	20
Afro-Eurasia	> 10647511	52
Afro-Eurasia	SUB TOTAL	162
America	10647511 - 2826683	10
America	2826683 - 72751	17
America	< 72751	12
America	> 10647511	12
America	SUB TOTAL	51
Antarctica	< 72751	5
Antarctica	SUB TOTAL	5
Oceania	10647511 - 2826683	2
Oceania	2826683 - 72751	10
Oceania	< 72751	21
Oceania	> 10647511	1
Oceania	SUB TOTAL	34
Others	2826683 - 72751	4
Others	< 72751	9
Others	SUB TOTAL	13
SUB TOTAL	10647511 - 2826683	67
SUB TOTAL	2826683 - 72751	66
SUB TOTAL	< 72751	67
SUB TOTAL	> 10647511	65
SUB TOTAL	SUB TOTAL	265

11. To refine the results from Q10 by excluding subtotals by country and the overall total, write an SQL query employing the ROLLUP function to achieve the desired output.

12. Implement an SQL statement to display the result of Q11 in a cross-tabulation format. Your query should replicate the provided example result. **Hint**: Consider utilizing the PIVOT function for this task.

Number of countries by country and popoluation	'Afro-Eurasia'	'America'	'Antarctica'	'Others'
10647511 - 2826683	55	10	(null)	(null)
2826683 - 72751	35	17	(null)	4
< 72751	20	12	5	9
> 10647511	52	12	(null)	(null)
SUB TOTAL	162	51	5	13

Part II. NoSQL Systems

1. Describe the key characteristics that differentiate NoSQL databases from traditional relational databases.

2. Explain the four main types of NoSQL databases: Key-Value Stores, Document Stores, Column-Family Stores, and Graph Databases.

3. Working with Document Stores - MongoDB

This section involves interaction with a database using MongoDB, a document store, and its query language.

You may utilize the MongDB shell for these exercises. You may use a local MongoDB instance, MongoDB Atlas which is a free tier cluster, or MongoDB Playground for interactive learning platform.
Refer to MongoDB Manual, Getting Started with the mongo Shell , MongoDB Operator

For the tasks listed below, provide a MongoDB command scripts along with their execution results.

You may utilize the MongDB shell for these exercises.

(1) Initialize a collection named ‘games’.

(2) Insert 6 games into the collection. Each document below include four attributes: name, genre, rating and achievements.

"name": "Spy Hunter", "genre": "Racing", "rating": 76, "achievements": ["Speed Demon"]
"name": "Mario Kart 64", "genre": "Racing", "rating": 96, "achievements": ["Game Master", "Speed Demon"]
"name": "Tetris", genre: "Puzzle", "rating": 83, "achievements": ["Puzzle Solver"]
"name": "Mega Man 5", "genre": "Platformer", "rating": 81, "achievements": ["Robot Master"]
"name": "Star Fox", "genre": "Action", "rating": 71
"name": "The Legend of Zelda: Ocarina of Time", "genre": "Action"
"name": "Banjo-Kazooie", "genre": " Platformer", "rating": 92, "achievements": ["Game Master", "Speed Demon"]

To reset the collection due to any input errors, utilize the remove() function on your collection.

(3) Retrieve all the games in the collection.

(4) Display only the name and genre of all games, excluding their _id.

(5) Locate a specific game by its name, such as “Mario Kart 64”, without employing limit(). Instead, use the findOne function.

(6) Find the top 3 highest-rated games.

(7) List all unique genres available in the collection.

(8) Find games with ratings above 90.

(9) Find all games except those in the ‘Racing’ genre.

(10) Identify games that are missing the ‘rating’ attribute.

(11) Count the number of games per genre

(12) Calculate the average rating for each genre.

(13) Locate all the games that possesses both the “Game Maser” and the “Speed Demon” achievements.

(14) Employee the update() function to modify a game named with “Star Fox” by adding two achievements with the following properties:

"name": "Game Master", "points": 100
"name": "Speed Demon", "points": 135

(15) Add a common achievement “Fan Favorite” to all games with a rating of 90 or higher.

(16) Delete a game named with “Banjo-Kazooie”.

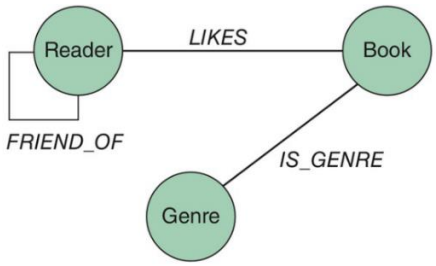
4. Working with Graph Database - Neo4j and Cypher query language

This section involves interaction with a book club database using Neo4j, a graph database, and its query language, Cypher.

You may use a local Neoj4 Community Edition instance or Neo4j Sandbox. You may also utilize the online playground that hosts the book club database.

For Neo4J and Cypher, refer to Neo4j Installation, Neo4j Getting Started , and Cypher Query Language.

(0) The figure below shows the schema diagram of the book club database.



The provided "book_club_db_creation.txt" contains necessary CREATE statements for the database setup. Create the database by executing the CREATE statements. For online execution, no database setup is required as it's already pre-configured.

For the tasks listed below, provide a Cypher query along with its execution result.

(1) Retrieve the names of members who enjoy thriller books.

(2) Determine which book has been liked by the youngest reader.

(3) Determine the genre most beloved by the club members.

(4) Identify the most popular book based on the number of likes.

(5) Find the common liked books between 'Wilfried Lemahieu' and his friends.

(6) Calculate the average age of readers who like 'Detective' genre books.

(7) Find friends of 'Bart Baesens' who liked thriller books, ensuring no name is repeated in your results.

(8) List the books that are liked by friends of 'Wilfried Lemahieu' but not by Wilfried himself.

(9) Determine which reader has the broadest taste in books, meaning they like the widest variety of genres.

(10) Query for books which haven't caught anyone's interest yet. You may get no result for this query.