

Innovation

Q4.1 SQL

Question C

This question asks to show the Name and first ten characters of the GovernmentFrom as Government of the countries where German is the official language and to sort them alphabetically by name. One of the main functions in SQL is the SELECT function. This is used to return the rows from one or more tables (dev.mysql, 2022). I have used alongside the SUBSTRING function which is a means of accessing a part of a string (dev.mysql, 2022). This function allowed me to select the first ten characters from the GovernmentFrom column. Next, I use an INNER JOIN to join the country table with the countryLanguage table but only where the isOfficial column is true (dev.mysql, 2022). Once the unofficial languages have been determined. I can now access which of the official languages are German and return the name of the country and the substring of the GovernmentFrom column and ORDER BY name ASC to order alphabetically (dev.mysql, 2022).

Question D

This SQL query required the use of the CASE function. This allows the user to access certain information based on more than one condition (dev.mysql, 2022). In this question I was asked to show the PersonID, Personname, Age and another column entitled Visa from the person table. I created a case to return the different visa results based on the person's age. The first case was if when the person's age was less than 20 then they should be in Visa-1. The second was if they were more than 20 but less than 29 inclusive then they would be in Visa-2 and so on. This CASE function allowed the different visas to be assigned based on the person's age and then I used the ORDER BY function again to first order by the Visa column and then the Personname column.

Question G

The next task was to show the name of the city, the arrival date, and the name of the country that city is in for all cities visited by Sara. In order to complete this query, I had to complete three

INNER JOINS as the required information was in four separate tables. First, I selected all the columns I wanted to return. Next, I joined the city table to the country table to get the name of the country the city was in. I then combined that new table with the hasvisitedcity table to discover which cities were visited. And finally, I joined it to the person table to stipulate that I wanted to search for Sara's ID. Again, I used the ORDER BY function to return the city name alphabetically.

Question H

This question asked to show to continent, the name, and the population of the country with the biggest population in each continent. I used an INNER JOIN to only select the max population from each country. I also used the GROUP BY function. This allowed me to group the max population from each country by continent (dev.mysql, 2022). I joined the two tables on the population and used the WHERE clause which selects the information based on a certain condition (w3schools, 2022). In this case, I was asked to only include countries where the population was greater than zero. Finally, I include the ORDER BY function to sort from largest to smallest population and then alphabetically by continent.

Question K

For this question I was asked to show the country code and the percentage of people who can speak a non-official language of all countries in the Caribbean. In order to complete this I needed to use the SUM function which allowed me to calculate the sum of population from each country (dev.mysql, 2022). Next, I used an INNER JOIN and combined the tables on the country code. However, I only do this where the official language is false, and the region is the Caribbean. This was done by using the WHERE clause in order to achieve these conditions. I grouped the results and ordered them by country code to allow for them to be alphabetized.

Question L

This final SQL question asked to show the name, population and personname of all the cities visited by people where the city population is greater than the maximum population of Polynesia. In order to complete this task, I was required to complete three joins. Two of the joins were

INNER JOINS and one was a LEFT JOIN. A left join will only return rows where there is a matching record on the right (w3schools, 2022). This is in contrast to an INNER JOIN which will only return matching rows in both tables. However, a left join was required to drop any country codes which contained cities that had not been visited. Furthermore, I needed to complete a WHERE condition but only by the maximum population of countries from Polynesia. The last step was to order the results alphabetically by city name.

Q4.2 Neo4j

Question C

For this first Neo4j query I was asked to return the names of hobbies and the number of people who have that hobby. The Neo4j command to select information from a database is the MATCH function (neo4j, 2022). This command allowed me to select a person node who has an outgoing relationship with the hobby node. Next, I used the DISTINCT function to only return the unique hobby names (neo4j, 2022). I also used the COUNT function which allowed me to retrieve the count of the person nodes for each hobby (neo4j, 2022). I have used the Neo4j ORDER BY command to order the results in increasing people order (neo4j, 2022).

Question D

The purpose of this question was to return the number of people who have a salary. For this I have used the WHERE function alongside the EXISTS function. The exists function will only return information where the clause exists (neo4j, 2022). This is what allowed me to match the people nodes who have a salary. However, I only want to return the number of people who have a salary. This required the COUNT function which returned the count of people where a salary existed.

Question G

This question asked that I return the names and ages of people who have the same hobby as Barbara Smith's maternal grandmother. The first step was to discover who Barbara Smith's maternal grandmother was. I did this by using the relationship "MOTHER_OF" twice and I called

the node gm (neo4j, 2022). Next, I used that gm node to find all the people who had the relationship with her hobby. Finally, I returned their names and ages of those people and ordered them alphabetically by name and then from youngest to eldest.

Question H

The goal of this question was to return the name of people and the person they married or if unmarried it should show null. Once again, I used the relationship “MARRIED_TO” to retrieve the spouse’s name of each person node. I allowed the relationship to go either direction as not all married person nodes have an outgoing relationship. This allowed me to retrieve all nodes names and spouse names/null.

Question K

The next question asked to return the number of people under 20 who have the hobby Reading. First step was to use the MATCH function with the relationship to the hobby reading. Next, I included a WHERE clause to only return nodes whose age was less than 20. Finally, I used the AS function to return the information under the name I wish to call the column (neo4j, 2022).

Question L

The final Neo4j question asked that I return a list of salaries of people who earn less than 50,000. In order to do this, I used the MATCH and WHERE functions to return the nodes with a salary less than fifty thousand. Next, I used the SET function (neo4j, 2022) which allowed me to set each salary to a rounded figure using the ROUND function (neo4j, 2022). Finally, I return the values using the COLLECT function which will return a list containing a set of values (neo4j, 2022). Furthermore, within that list I have used the TOINTEGER function which will convert the type of expression into an integer value (neo4j, 2022) which return a list of salaries in the desired format.

Q4.4 Python: app.py/appDB.py

Q4.4.1

This choice shows the user a list of employee names in alphabetical order and the names of the departments in which they work, in groups of two. For this I have used a SQL query which can be found in appDB.py. Part of this command uses an INNER JOIN. This is a SQL command which returns all rows from selected tables if there is a match between the columns that the join is on (w3schools, 2022). In this scenario, there are department IDs in the dept table that are not referenced in the employee table. Therefore, those rows will not be joined as there is no match. This SQL query also uses the commands LIMIT and OFFSET. The LIMIT command is used to specify the number of rows returned by the query (sqltutorial, 2022). The OFFSET command is optional, and its purpose is to skip a specified number of rows before returning the query (sqltutorial, 2022). ORDER BY is then used to order the results alphabetically (w3schools, 2022). In the choice_one function I have started with an offset of zero to allow for the first two rows to be returned. Then unless the key “q” is pressed it will add two to the offset and run the SQL query again. It will continue to do this until the key is selected and then the program returns to the main menu.

Q4.4.2

The purpose of choice two is to show the minimum salary, average salary and maximum salary of the employee ID entered by the user. This requires the use of SQL aggregate functions: MIN, AVG and MAX which will take the salary and provide the minimum, average and maximum values of that salary (dev.mysql., 2022). I have also used the FORMAT function which rounds the number to the decimal place specified in a string format (dev.mysql, 2022). In this program I have chosen to format to two decimal places. I have also used a placeholder for the parameter in the SQL query. This placeholder allows for the value supplied by the user to then be passed into the query when run (pynative, 2022). Similarly, in order to check if the employee ID entered is valid, I wrote another query to return employee ID that are like the user’s input using a placeholder again. Thus, this choice takes the three desired salary values and INNER JOINS them to the employee ID

where the employee ID is the same as the user's input. As a result, if the employee ID exists the three values are returned, otherwise nothing is returned. The built-in python function `exit` is used to stop the program running (delftstack, 2022).

Q4.4.3

Option three asks the user to input a month and then returns all employees born in that month. This question required the `datetime` package to be imported into the `app.py` program. This package is useful when working with dates and times (docs.python, 2022). I used this in the `mtNum` function to convert the month entered to numeric format. It is important to note that I have used the python `casefold` function when asking for the user input. This is a build-in function which ignores the case of letters when matching (programiz, 2022). Therefore, if the abbreviation for February was entered as "fEb", it would be accepted and returned as two. The program will continue to ask the user to enter a month if neither 1-12 nor one of the 12-month abbreviations is entered. Next, the SQL query uses the `MONTH` command. This command returns the month of a date (dev.mysql, 2022). Consequently, this SQL query gets the required details from the database if the month of the dob is like the one entered by the user.

Q4.4.4

This question asked the user for employee information and adds that employee to the database. This requires the use of placeholders in the SQL command along with the `INSERT` command which will insert the information into the database (dev.mysql, 2022). However, there can be issues when inserting into a database. If invalid data or format was used it can cause handling errors (thepythonguru, 2022). On top of this, a primary key cannot be duplicated (w3schools, 2022). Furthermore, a foreign key will prevent invalid data being entered as it is linked to a parent table (w3schools, 2022). In this case, the `eid` is a primary key. If an `eid` already used was entered, it will throw a Data Error. If an invalid dob was entered or entered in the wrong format, it will throw an Operational Error. Finally, if a did was entered which did not exist in the parent table as it is a foreign key, then it will throw an Integrity Error. To facilitate all these errors, I have used `try` and `except` statements (w3schools, 2022). As a result, the program will try to add a new employee. If there is an error, it will read the error and print out the appropriate response to that error.

Q4.4.5

Choice five asks the user to enter an employee ID and returns the name and budget of all departments managed by that employee. This question requires the use of both SQL and Neo4j queries. Firstly, I need to connect to the Neo4j server. Then with the global variable driver I use a `read_transaction` function which will read the selected query from the database (neo4j, 2022). In this case, I am using relationships between nodes to discover if the employee ID entered manages a department and what that department ID is. Then I am passing that department ID into a SQL query to acquire the budget for that department. Finally, I use the python format function to return the budget as formatted string with commas (geeksforgeeks, 2022).

Q4.4.6

This question asks the user to enter an employee ID and a department ID and updates the Neo4j database to show that the employee ID entered now manages that department. This choice was broken down into a number of functions. It starts off with the `add_mgr_to_dept` function where the user is asked to input the two values. Then I use SQL queries to validate that the eid and did exist in the SQL database. If they are invalid, a note is printed out and the user is asked for another eid and did. If they are valid the next step is to check if the department is already managed by another employee. I do this by first off creating a constraint if one does not already exist in Neo4j. I have done this to make sure that each node is unique, otherwise, the database could create duplicate node for E01 for example which would be an incorrect reflection of the data (neo4j, 2022). Next, I use a `read_transaction` from Neo4j to check if there are any departments managed by employees and return the name of those departments. Afterwards I retrieve the employee ID of the departments already managed. If the eid variable is not empty a note is returned, and the program is stopped because that department is already managed. Otherwise, the next function is called, and the manager is added to that department. This is done using a write transaction to update Neo4j (neo4j, 2022).

Q4.4.7

This question shows details of all the departments from SQL database. I created an empty list which I append this information to. If the length of the list is zero, then it will read the information from the database and add it to the empty list. This is done by using the python append method (w3schools, 2022). Subsequently, if the user chooses option seven again and the list is not empty then it will read the details from the list instead of reading it from the database again.

Bibliography

(2022, April 25). Retrieved from w3schools: https://www.w3schools.com/sql/sql_join_inner.asp

(2022, April 25). Retrieved from sqltutorial: <https://www.sqltutorial.org/sql-limit/>

(2022, April 26). Retrieved from w3schools: https://www.w3schools.com/sql/sql_orderby.asp

(2022, April 26). Retrieved from dev.mysql.:
<https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html>

(2022, April 26). Retrieved from pynative: <https://pynative.com/python-mysql-execute-parameterized-query-using-prepared-statement/>

(2022, April 26). Retrieved from dev.mysql: https://dev.mysql.com/doc/refman/8.0/en/string-functions.html#function_format

(2022, April 26). Retrieved from delftstack: [https://www.delftstack.com/howto/python/python-exit/#:~:text=the%20site%20module.-,Exit%20Programs%20With%20the%20exit\(\)%20Function%20in%20Python,a%20System%20Exit%20exception%20when%20executed.](https://www.delftstack.com/howto/python/python-exit/#:~:text=the%20site%20module.-,Exit%20Programs%20With%20the%20exit()%20Function%20in%20Python,a%20System%20Exit%20exception%20when%20executed.)

(2022, April 26). Retrieved from docs.python: <https://docs.python.org/3/library/datetime.html>

(2022, April 26). Retrieved from programiz: <https://www.programiz.com/python-programming/methods/string/casefold>

(2022, April 26). Retrieved from dev.mysql: https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html#function_month

(2022, April 26). Retrieved from dev.mysql: https://dev.mysql.com/doc/refman/8.0/en/string-functions.html#function_insert

(2022, April 26). Retrieved from w3schools:
https://www.w3schools.com/sql/sql_primarykey.ASP

(2022, April 26). Retrieved from thepythonguru: <https://thepythonguru.com/handling-errors/>

(2022, April 26). Retrieved from w3schools:
https://www.w3schools.com/sql/sql_foreignkey.asp

(2022, April 26). Retrieved from w3schools:
https://www.w3schools.com/python/python_try_except.asp

(2022, April 26). Retrieved from neo4j: <https://neo4j.com/docs/api/python-driver/current/api.html>

(2022, April 26). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/python-string-format-method/>

(2022, April 26). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/constraints/examples/>

(2022, April 26). Retrieved from w3schools: https://www.w3schools.com/python/ref_list_append.asp

(2022, April 28). Retrieved from dev.mysql: <https://dev.mysql.com/doc/refman/8.0/en/order-by-optimization.html>

(2022, April 27). Retrieved from dev.mysql: <https://dev.mysql.com/doc/refman/8.0/en/join.html>

(2022, April 27). Retrieved from dev.mysql: https://dev.mysql.com/doc/refman/8.0/en/string-functions.html#function_substring

(2022, April 27). Retrieved from dev.mysql: <https://dev.mysql.com/doc/refman/8.0/en/select.html>

(2022, April 28). Retrieved from dev.mysql: https://dev.mysql.com/doc/refman/8.0/en/flow-control-functions.html#operator_case

(2022, April 28). Retrieved from dev.mysql: <https://dev.mysql.com/doc/refman/5.7/en/group-by-modifiers.html>

(2022, April 29). Retrieved from w3schools: https://www.w3schools.com/sql/sql_where.asp

(2022, April 28). Retrieved from dev.mysql: https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_sum

(2022, April 28). Retrieved from w3schools: https://www.w3schools.com/sql/sql_join_left.asp

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/clauses/match/>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/functions/aggregating/#functions-count>

(2022, April 29). Retrieved from neo4j: <https://community.neo4j.com/t/enumerating-distinct-property-values/11673>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/clauses/order-by/>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/functions/predicate/#functions-exists>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/graphql-manual/current/type-definitions/relationships/>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/clauses/>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/clauses/set/>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/functions/mathematical-numeric/#functions-round>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/functions/aggregating/#functions-collect>

(2022, April 29). Retrieved from neo4j: <https://neo4j.com/docs/cypher-manual/current/functions/scalar/#functions-tointeger>