2 Written Homework

1. What is the difference between a view and a table in sql?

   Tables are files that actually hold data in them. They consist of columns (fields) and rows(records).

   Views are a virtual construct based on the tables. They use the information on one or more tables to generate a table that does not actually store data. Instead, the view only holds a reference to the requested data on the original table(s). This not only has the benefit of reducing redundancy when we want to have a table that shows only limited data or joint data from the original tables; it also means that the view is always "updated" because it just pulls data from the real tables.

2. How does a GROUP BY statement in SQL work?

   GROUP BY takes the values of a column (or set of values of multiple columns) that are repeated, and places all of them together on a single row in the resulting view.

   Note that 1. GROUP BY must have as arguments at least all of SELECT's arguments that are not an aggregate; 2. Using GROUP BY tends to make sense used alongside aggregate functions, i.e. the aggregate is based on the columns used for grouping.

   Source: https://www.geeksforgeeks.org/sql-group-by/

3. What is HTTP and how do HTTP requests work?

   It stands for Hypertext Transfer Protocol. This protocol is based on the exchange of messages between a client and a server. The client starts by sending a message (request) with a particular structure to the server. In its simplest form, this request contains information on the action the client wants the server to do and on what resources such action should be performed. There could be additional details in the message, including authentication information (e.g. authentication token), client's properties, images, among other things the server might need to perform the aforementioned action.

   If the server received the request, it will then proceed to reply with another message (response) to the client. This response also has a structure and can contain a wider variety of information as well.

   Source: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

4. What are the characteristics of a RESTful API?

   - It is a client-server architecture: This means that the client infrastructure is separated from the server infrastructure. They both interact through an API. This decoupling allows both sides, client and server, to have certain freedom to change without affecting the other.

- Stateless: The server does not hold any information on the state of the application on the client side and vice versa. Each request sent to the server contains all the necessary information about the client for the server to perform an action.
- Cacheable: Allows the client cache the response to a request, as long as the server gives permission for it. If the client is allowed to cache the data, then any equivalent requests produced by the client in the future can use that response data, instead of having to send a request to the server again.
- Uniform Interface: By placing certain architectural constraints, it is possible to simplify the interactions between the client and the server. There are four interface constraints established by REST: Identification of resources, manipulation of resources through representations, self-descriptive messages, hypermedia as the engine of application state.
- Layered System: Components are placed in specific layers or subsystems. These layers are hierarchical. Components in one layer can only interact with layers adjacent to their own.
- Code on Demand: Additional functionality can be added to the client side without requiring an update or restructuring to it. The server can provide smaller programs (applets, scripts, etc.) to the client which the latter can run locally to perform actions that are not permanently part of its base code.

Source: https://restfulapi.net

5. Given the two tables 'Birth Info' and 'Astrological Info', what is the result of the following queries in table form.

Table 1: Birth_Info

name	**birth_month**
Don	September
Meghna	June
Aly	January
Ara	September
Kirit	May

Table 2: Astrological_Info

month	**astrological_sign**
January	Aquarius
September	Virgo
June	Gemini

a. SELECT name, birth_month, astrological_sign FROM birth_info LEFT JOIN astrological_info ON birth_info.birth_month = astrological_info.month

name	**birth_month**	**astrological_sign**
Aly	January	Aquarius
Don	September	Virgo

Ara	September	Virgo
Meghna	June	Gemini
Kirit	May	Null

b. SELECT name, birth_month, astrological_sign FROM birth_info OUTER JOIN astrological_info ON birth_info.birth_month = astrological_info.month

Note: OUTER JOIN not supported in MySQL. For testing, I implemented the OUTER JOIN using a UNION of LEFT JOIN and RIGHT JOIN.

name	**birth_month**	**astrological_sign**
Aly	January	Aquarius
Don	September	Virgo
Ara	September	Virgo
Meghna	June	Gemini
Kirit	May	Null

c. SELECT name, birth_month, astrological_sign FROM birth_info INNER JOIN astrological_info ON birth_info.birth_month = astrological_info.month

name	**birth_month**	**astrological_sign**
Don	September	Virgo
Meghna	June	Gemini
Aly	January	Aquarius
Ara	September	Virgo

d. SELECT name, birth_month, astrological_sign FROM birth_info RIGHT JOIN astrological_info ON birth_info.birth_month = astrological_info.month

name	**birth_month**	**astrological_sign**
Don	September	Virgo
Meghna	June	Gemini
Aly	January	Aquarius
Ara	September	Virgo

6. What do these HTTP response codes mean:

Code	**Meaning**
200	OK
201	Created
400	Bad Request

401	Unauthorized
300	Multiple Choice
403	Forbidden
404	Not Found
500	Internal Sever Error

Sources: https://www.restapitutorial.com/httpstatuscodes.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status