Quizzes for SQL for Data Analysis – Level 1

Quiz #1 - Database Terminology

#	Question	Answer
1	A relational database is an organized collection of data like a logical container, and the data is stored as a collection of records. A record is like one line of text combined from multiple attributes related to some entity.	Yes/No
2	A database table looks like a two-dimensional array made up of rows and columns. Each row is related to a specific attribute and must have a pre-defined data type. Each column is a collection of attributes related to some logical entity.	Yes/No Each column is related to a specific attribute and must have a pre-defined data type. Each row is a collection of attributes related to some logical entity.
3	A primary key is used to uniquely identify a single row in a table, and it must be based on a single column.	Yes/No It can be by using a single column or a group of columns that their combined values uniquely identify every row in a table.
4	A foreign key is used to navigate from one table to another table. A foreign key is a column or group of columns in a table that references a primary key in another table in the database.	Yes/No
5	A relational model has the following building blocks:	1. Entity, Attribute, Relationship 2. Entity, Columns, Rows 3. Schema, Attribute, Tables 4. Schema, Attribute, Entity
6	A database schema is a logical structure, constraints, and associations of a group of related tables inside a database. A schema is also a namespace of a collection of tables.	Yes/No
7	Metadata is data about the database structure. It is data about the database objects that will be used to store actual data.	Yes/No
8	An index is a special support table that keeps rows in a specific order. We will have a corresponding row in the index table for every row in the data table. The order of the rows in the index table will be based on the required index.	Yes/No
9	The role of indexes is to facilitate the retrieval of a subset of a table's rows and columns without the need to inspect every row in the table.	Yes/No
10	A major advantage to partitioning is that when we query data, the required data may be raised on a couple of partitions instead of scanning the entire table.	Yes/No

Quiz #2 - Creating Databases, Schemas and Tables

#	Question	Answer
1	What is the proper flow to create a database instance with a schema and tables?	1. Schemas, Database, Tables 2. Database, Tables, Schemas 3. Database, Schemas, Tables 4. Schemas, Tables, Database
2	When creating a new database table, we must specify the list of attributes as columns and the data type of each attribute.	Yes/No
3	The following SQL code will create a table called "netflix" and a schema called "titles". Inside the table, we will have an attribute called "title_type" as a varchar data type that can be a NULL value. CREATE TABLE netflix.titles (title_id varchar(10), title_type varchar(5) NOT NULL);	Yes/No The following SQL code will create a table called titles and a schema called "netflix". Inside the table, we will have an attribute called title_type as a varchar data type that can not be a NULL value.
4	The following SQL code will create a table called "titles" that will accept rows only if the "release_year" column has a value greater or equal to 1999. CREATE TABLE netflix.titles (release_year integer CHECK (release_year>=1999));	Yes/No
5	The following SQL code will create a table called "credits" inside the "netflix" schema where the "title_id" is a foreign key to the "titles" table. CREATE TABLE netflix.credits (person_id integer, title_id varchar(10) REFERENCES netflix.titles(title_id), name varchar(300), character varchar(300), role varchar(10))	Yes/No
6	When using the INSERT INTO statement to insert new rows into a table, is it recommended to provide the values in	Yes/No

	the correct order without mentioning the column's names.	The recommended option is to explicitly list the columns before the values. In that case, we don't need to know the order of the columns in the table.
7	The COPY command available in PostgreSQL can be used to insert a large amount of data from a CSV file.	Yes/No
8	Using SQL, we can perform the following update and delete options:	• Update rows using the UPDATE statement • Rename columns or add new columns using the ALTER TABLE statement • Remove rows, columns, or tables using the DELETE statement • All answers are right
9		
10		

Quiz #3 - Retrieving Data with Queries

#	Question	Answer
1	The SELECT section as part of a query determines which columns to include in the query's result set. The FROM section is used to identify the tables from which to retrieve data.	Yes/No
2	In the following query, the SELECT and FROM are keywords, and the list of columns and tables are called identifiers. Keywords and identifiers are case insensitive. On the other hand, a quoted identifier is case-sensitive. SELECT title_id, title FROM netflix.titles;	Yes/No
3	Before applying a complex query, we can limit the number of records received from a query to a specific number by using the WHERE conditions.	Yes/No LIMIT number of records
4	Comments in SQL code are useful or the following main reasons:	• Mentioning WHAT and WHY we were doing when creating a complex query • Mute a specific line or a block of code for troubleshooting • All answers are right
5	The following SQL code will filter all columns from all rows where the release year is between 1979 and 1989. SELECT * from netflix.titles WHERE release_year BETWEEN 1980 AND 1990	Yes/No The upper and lower limits are inclusive.
6	The following SQL code will filter rows where the "seasons" column has some value. SELECT title, title_type, release_year, seasons FROM netflix.titles WHERE seasons IS NOT NULL	Yes/No
7	The following SQL code will filter all rows with "SHOW" title type that their release_year is 2000 or all rows with "SHOW" title type that their runtime is more than 100mins. SELECT title, title_type, runtime, release_year FROM netflix.titles WHERE	Yes/No Missing parentheses. It should be: title_type = 'SHOW' AND (runtime>100 OR release_year = 2000)

	title_type = 'SHOW' AND release_year = 2000 OR runtime>100	
8	The following SQL code will filter all rows where the "release_year" is not equal to specific years - 2000, 2004, 2010. SELECT * FROM netflix.titles WHERE release_year NOT IN (2000, 2004, 2010)	Yes/No
9	An alias is used to re-label a column name or table name or assign a label to those columns generated by expressions or built-in function calls.	Yes/No
10	A search pattern is useful for searching a portion of the information from a string-type column. When building a search pattern, we can combine text and wildcard characters. A wildcard is a special character used to match parts of a value. There are two wildcard characters: underscore and percent.	Yes/No
11	The following SQL code will filter all rows with movies to TV shows with a title that starts with the word "Top". SELECT title, title_type FROM netflix.titles WHERE title LIKE '%Top%'	Yes/No All movies or tv shows with a title that includes the "Top" word somewhere inside the title.
12	The following SQL code will the distinct list age_certification for movies or TV shows, excluding NULL values. SELECT DISTINCT age_certification FROM netflix.titles WHERE age_certification IS NOT NULL	Yes/No
13	The following SQL code will count the number of distinct age_certification rows. SELECT DISTINCT COUNT(age_certification) AS unique_agecerti FROM netflix.titles; WHERE age_certification IS NOT NULL	Yes/No The distinct goes inside the aggregate function: COUNT(DISTINCT age_certification).

14	The following SQL code will select all titles sorted by the "release_year" and the "runtime" column. This means that it will first order the rows by "release_year", but if some rows have the same "release_year", it will order them by "runtime". SELECT title, release_year, runtime FROM netflix.titles ORDER BY release_year, runtime	Yes/No
15	In the following query, we are grouping the rows based on "tite_type" that their "release_year" is more than 2000 and calculating the average "imdb_score" for each "title_type" group. SELECT title_type, AVG(imdb_score) FROM netflix.titles WHERE release_year >= 2000 GROUP BY title_type	Yes/No
16	HAVING is working on individual rows from the source tables and WHERE is working on the group-level aggregated rows.	Yes/No WHERE is working on individual rows from the source tables and HAVING is working on the group-level aggregated rows.
17	The order of query execution looks like the following pipeline. Each section receives the result of the previous section and produces an output that the next section takes as input. FROM >> WHERE >> GROUP BY >> HAVING >> SELECT >> ORDER BY	Yes/No