Congratulations! You passed!

Grade received 100% To pass 80% or higher

Go to next item

Test your knowledge on SQL queries

Total points 4

1. Which of the following SQL functions can data analysts use to clean string variables? Select all that apply. **1 / 1 point**

- [x] SUBSTR

Correct
Data analysts can use the SUBSTR and TRIM functions to clean string variables.

- [ ] COUNTIF
- [x] TRIM

Correct
Data analysts can use the SUBSTR and TRIM functions to clean string variables.

- [ ] LENGTH

2. You are working with a database table that contains data about playlists for different types of digital media. The table includes columns for *playlist_id* and *name*. You want to remove duplicate entries for playlist names and sort the results by playlist ID. **1 / 1 point**

You write the SQL query below. Add a DISTINCT clause that will remove duplicate entries from the *name* column.

NOTE: The three dots (...) indicate where to add the clause.

```
1  SELECT DISTINCT name AS  playlist_name
2  FROM
3  playlist
4  ORDER BY
5  playlist_id
```

Run

Reset

```
+----------------------------+
| playlist_name              |
+----------------------------+
| Music                      |
| Movies                     |
| TV Shows                   |
| Audiobooks                 |
| 90's Music                 |
| Music Videos               |
| Brazilian Music            |
| Classical                  |
| Classical 101 - Deep Cuts  |
| Classical 101 - Next Steps |
| Classical 101 - The Basics |
| Grunge                     |
| Heavy Metal Classic        |
| On-The-Go 1                |
+----------------------------+
```

What playlist name appears in row 6 of your query result?

- TV Shows
- Movies
- Music Videos
- Audiobooks

Correct

The clause `DISTINCT name` will remove duplicate entries from the name column. The complete query is `SELECT DISTINCT name FROM playlist ORDER BY playlist_id`. The DISTINCT clause removes duplicate entries from your query result. The playlist name Music Videos appears in row 6 of your query result.

3. You are working with a database table that contains data about music albums. The table includes columns for *album_id, title,* and *artist_id*. You want to check for album titles that are less than 4 characters long. **1 / 1 point**

You write the SQL query below. Add a LENGTH function that will return any album titles that are less than 4 characters long.

```
1  SELECT
2  *
3  FROM
4  album
5  WHERE
6      LENGTH(title) < 4
```

Run

Reset

```
+----------+-------+-----------+
| album_id | title | artist_id |
+----------+-------+-----------+
|      131 | IV    |        22 |
|      181 | Ten   |       118 |
|      182 | Vs.   |       118 |
|      236 | Pop   |       150 |
|      239 | War   |       150 |
+----------+-------+-----------+
```

What album ID number appears in row 3 of your query result?

- 131
- 239
- 182
- 236

Correct

The function `LENGTH(title) < 4` will return any album names that are less than 4 characters long. The complete query is `SELECT * FROM album WHERE LENGTH(title) < 4`. The LENGTH function counts the number of characters a string contains. The album ID number 182 appears in row 3 of your query result.

4. You are working with a database table that contains customer data. The table includes columns about customer location such as *city, state,* and *country.* You want to retrieve the first 3 letters of each country name. You decide to use the SUBSTR function to retrieve the first 3 letters of each country name, and use the AS command to store the result in a new column called *new_country.*

1 / 1 point

You write the SQL query below. Add a statement to your SQL query that will retrieve the first 3 letters of each country name and store the result in a new column as *new_country.*

NOTE: The three dots (...) indicate where to add the statement.

```
1  SELECT
2  customer_id,
3  SUBSTR(country,1 , 3) AS new_country
4  -- Extract a substring from a string (start at position 5, extract 3 characters):
5  --SELECT SUBSTR("SQL Tutorial", 5, 3) AS ExtractString;
6  FROM
7  customer
8  ORDER BY
9  country
```

Run

Reset

```
+-------------+-------------+
| customer_id | new_country |
+-------------+-------------+
|          56 | Arg         |
|          55 | Aus         |
|           7 | Aus         |
|           8 | Bel         |
|           1 | Bra         |
|          10 | Bra         |
|          11 | Bra         |
|          12 | Bra         |
|          13 | Bra         |
|           3 | Can         |
|          14 | Can         |
|          15 | Can         |
|          29 | Can         |
|          30 | Can         |
|          31 | Can         |
|          32 | Can         |
|          33 | Can         |
|          57 | Chi         |
|           5 | Cze         |
|           6 | Cze         |
|           9 | Den         |
|          44 | Fin         |
|          39 | Fra         |
|          40 | Fra         |
|          41 | Fra         |
+-------------+-------------+
(Output limit exceeded, 25 of 59 total rows shown)
```

What customer ID number appears in row 2 of your query result?

- ○ 28
- ◉ 55
- ○ 3
- ○ 47

Correct

The statement `SUBSTR(country, 1, 3) AS new_country` will retrieve the first 3 letters of each state name and store the result in a new column as *new_country.* The complete query is `SELECT customer_id, SUBSTR(country, 1, 3) AS` ... The SUBSTR function extracts a substring from a string. This

`new_country FROM customer ORDER BY country`. The SUBSTR function extracts a substring from a string. This function instructs the database to return 3 characters of each country, starting with the first character. The customer ID number 55 appears in row 2 of your query result.