Congratulations! You passed!

Grade received 100% To pass 80% or higher

Go to next item

Weekly challenge 3

Latest Submission Grade 100%

1. A data analyst is analyzing medical data for a health insurance company. The dataset contains billions of rows of data. Which of the following tools will handle the data most efficiently? 1 / 1 point

- ○ A spreadsheet
- ◉ SQL
- ○ A presentation
- ○ A word processor

Correct
SQL will handle the data most efficiently. SQL can handle huge amounts of data.

2. In which of the following situations would a data analyst use spreadsheets instead of SQL? Select all that apply. 1 / 1 point

- ☑ When working with a small dataset

Correct
An analyst would choose to use spreadsheets instead of SQL when visually inspecting data or working with a small dataset.

- ☐ When using a language to interact with multiple database programs
- ☐ When working with a dataset with more than 1,000,000 rows
- ☑ When visually inspecting data

Correct
An analyst would choose to use spreadsheets instead of SQL when visually inspecting data or working with a small dataset.

3. A data analyst runs a SQL query to extract some data from a database for further analysis. How can the analyst save the data? Select all that apply. 1 / 1 point

- [ ] Run a SQL query to automatically save the data.
- [x] Download the data as a spreadsheet.

Correct
The analyst can save the data by downloading the data as a spreadsheet or creating a new table for the data.

- [ ] Use the UPDATE query to save the data.
- [x] Create a new table for the data.

Correct
The analyst can save the data by downloading the data as a spreadsheet or creating a new table for the data.

4. You are working with a database table that contains invoice data. The table includes columns for *invoice_id* and *billing_state*. You want to remove duplicate entries for billing state and sort the results by invoice ID. **1 / 1 point**

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

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

```
1  SELECT DISTINCT billing_state
2  FROM
3  invoice
4  ORDER BY
5  invoice_id
```

Run

Reset

```
+---------------+
| billing_state |
+---------------+
|          None |
|            AB |
|            MA |
|        Dublin |
|            CA |
|            WA |
|            NV |
|            WI |
|            NS |
|           NSW |
|            SP |
|            NT |
|            VV |
|            RJ |
|            DF |
|            BC |
|            AZ |
|            ON |
|            MB |
|            RM |
|            TX |
|            UT |
|            FL |
|            IL |
|            QC |
+---------------+
(Output limit exceeded, 25 of 26 total rows shown)
```

What billing state appears in row 17 of your query result?

- ○ WI

- (●) AZ
- ( ) NV
- ( ) CA

Correct
The clause `DISTINCT billing_state` will remove duplicate entries from the *billing_state* column. The complete query is `SELECT DISTINCT billing_state FROM invoice ORDER BY invoice_id`. The DISTINCT clause removes duplicate entries from your query result. The billing state AZ appears in row 17 of your query result.

5. You are working with a database table that contains customer data. The table includes columns about customer location such as *city, state, country,* and *postal_code*. You want to check for city names that are greater than 9 characters long. **1 / 1 point**

You write the SQL query below. Add a LENGTH function that will return any city names that are greater than 9 characters long.

```
1  SELECT
2  *
3  FROM
4  customer
5  WHERE
6      LENGTH(city) > 9
```

Run

Reset

customer_id	first_name	last_name	company	address	city
1	Luís	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Car
9	Kara	Nielsen	None	Sønder Boulevard 51	Copenhagen
12	Roberto	Almeida	Riotur	Praça Pio X, 119	Rio de Janeiro
16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View
20	Dan	Miller	None	541 Del Medio Avenue	Mountain View
26	Richard	Cunningham	None	2211 W Berry Street	Fort Worth
28	Julia	Barnett	None	302 S 700 E	Salt Lake City
33	Ellie	Sullivan	None	5112 48 Street	Yellowknife
54	Steve	Murray	None	110 Raeburn Pl	Edinburgh
56	Diego	Gutiérrez	None	307 Macacha Güemes	Buenos Aires

What is the first name of the customer that appears in row 7 of your query result?

- (●) Julia
- ( ) Diego
- ( ) Roberto
- ( ) Kara

Correct
The function `LENGTH(city) > 9` will return any city names that are greater than 9 characters long. The complete query is `SELECT * FROM customer WHERE LENGTH(city) > 9`. The LENGTH function counts the number of characters a string contains. Julia is the first name of the customer that appears in row 7 of your query result.

6. Fill in the blank: refers to the process of converting data from one type to another. **1 / 1 point**

1 / 1 point

- Cleaning
- Typecasting
- Querying
- Formatting

Correct
Typecasting refers to the process of converting data from one type to another.

7. Fill in the blank: In SQL databases, the _____ function can be used to convert data from one datatype to another. **1 / 1 point**

- SUBSTR
- LENGTH
- CAST
- TRIM

Correct
The CAST function can be used to convert data from one datatype to another.

8. What SQL function lets you add strings together to create new text strings that can be used as unique keys? **1 / 1 point**

- CAST
- COALESCE
- LENGTH
- CONCAT

Correct
The CONCAT function lets you add strings together to create new text strings that can be used as unique keys.

9. You are working with a database table that contains invoice data. The table includes columns about billing location such as *billing_city, billing_state,* and *billing_country.* You want to retrieve the first 4 letters of each city name. You decide to use the SUBSTR function to retrieve the first 4 letters of each city name, and use the AS command to store the result in a new column called *new_city.* **1 / 1 point**

You write the SQL query below. Add a statement to your SQL query that will retrieve the first 4 letters of each city name and store the result in a new column as *new_city.*

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

```
1  SELECT
2  invoice_id,
3  SUBSTR(billing_city, 1, 4) AS new_city
4  FROM
5  invoice
6  ORDER BY
7  billing_city
```

Run

Reset

```
+------------+----------+
| invoice_id | new_city |
+------------+----------+
|         32 | Amst     |
|        161 | Amst     |
|        184 | Amst     |
|        206 | Amst     |
|        258 | Amst     |
|        379 | Amst     |
|        390 | Amst     |
|         23 | Bang     |
|         45 | Bang     |
|         97 | Bang     |
|        218 | Bang     |
|        229 | Bang     |
|        284 | Bang     |
|          7 | Berl     |
|         29 | Berl     |
|         30 | Berl     |
|         40 | Berl     |
|         52 | Berl     |
|         95 | Berl     |
|        104 | Berl     |
|        224 | Berl     |
|        225 | Berl     |
|        236 | Berl     |
|        247 | Berl     |
|        269 | Berl     |
+------------+----------+
(Output limit exceeded, 25 of 412 total rows shown)
```

What invoice ID number appears in row 7 of your query result?

- ○ 23
- ○ 97
- ○ 206
- ◉ 390

Correct

The statement `SUBSTR(billing_city, 1, 4) AS new_city` will retrieve the first 4 letters of each city name and store the result in a new column as *new_city*. The complete query is `SELECT invoice_id, SUBSTR(billing_city, 1, 4) AS new_city FROM invoice ORDER BY billing_city`. The SUBSTR function extracts a substring from a string. This function instructs the database to return 4 characters of each billing city, starting with the first character. The invoice ID number 390 appears in row 7 of your query result.