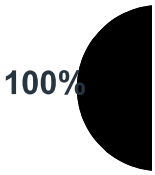
# Results

Seif Kungulio




**100**
Out of 100 points

**34:18**
Time for this attempt

## Your Answers:

1 5 / 5 points

According to the PostgreSQL document, which of the following is true about the CREATE DATABASE command?

- ◯ The CREATE DATABASE command can only be invoked by the superuser role
- ◯ The CREATE DATABASE command can only be utilized from the pgAdmin RDMS
- ✓ ◉ The CREATE DATABASE command creates a copy of the template1 database

### Feedback

**Based on your answer**

That is correct. The CREATE DATABASE command creates a copy of the template1 database.

2 5 / 5 points

Choose the query below that outputs the names of employee sorted alphabetically.

- ◯ SELECT * FROM employee GROUP BY lastname
- ◯ SELECT * FROM employee ORDER BY lastname DESC;
- ✓ ◉ SELECT * FROM employee ORDER BY lastname;

3 5 / 5 points

Which of the following SQL keywords allows you to ensure that only those records that are non-duplicative are returned in a query:

- ◯ DISCREET
- ◯ IDEAL

○ UNIQUE

✓ ● DISTINCT

4 5 / 5 points

Which of the following SQL commands will globally change the price of undershirts to &10.95?

✓ ●
```
UPDATE product
        SET price = 10.95
        WHERE product = ‘undershirts’;
```

○
```
SELECT product,
        price
        FROM employee
        WHERE product = ‘undershirts’ AND price = 10.95;
```

○
```
UPDATE employee
        SET price = 1095
        WHERE id = 212;
```

5 5 / 5 points

Which of the following SQL keyword is used to import data into PostgreSQL?

✓ ● COPY

○ INJECT

○ IMPORT

6 5 / 5 points

Given the table below, select the query that will find the sum of all salaries for the ‘east’ region.

employee

| Id | lastname | firstname | salary | region | supervisor |
|---|---|---|---|---|---|
| 1 | Jones | Alice | 47520.00 | East | McClanahan |
| 2 | Barber | John | 99100.00 | North | Jones |
| 3 | McGraw | Lana | 110570.00 | North | Jones |

| Id | lastname | firstname | salary | region | supervisor |
|---|---|---|---|---|---|
| 4 | Justice | Bernice | 82010.00 | South | Andrews |
| 5 | Pedigrew | Alexander | 71700.00 | West | Allen |

✓ SELECT sum(salary) AS TotSalary FROM employee WHERE region = ‘east’;

SELECT sum(salary) AS TotSalary FROM employee WHERE region = east;

SELECT max(salary) AS TotSalary FROM employee WHERE region = ‘east’;

SELECT max(salary) AS TotSalary FROM employee WHERE region = east;

---

**7** 5 / 5 points

Given the tables below, what value(s) will be returned from the following query? The / separates the column title and values returned.

SELECT count(id) AS totalforsup FROM employee WHERE supervisor = ‘Jones’;

employee

| Id | lastname | firstname | salary | region | supervisor |
|---|---|---|---|---|---|
| 1 | Jones | Alice | 47520.00 | East | McClanahan |
| 2 | Barber | John | 99100.00 | North | Jones |
| 3 | McGraw | Lana | 110570.00 | North | Jones |
| 4 | Justice | Bernice | 82010.00 | South | Andrews |
| 5 | Pedigrew | Alexander | 71700.00 | West | Alleny |

sumregion / 3

totalforsup / 1

Error

✓ totalforsup / 2

---

**8** 5 / 5 points

Which of the following queries will properly create the table country with an appropriate id, country name, and country region code. The table should have an appropriate primary key.

```
ALTER TABLE country

ALTER COLUMN COUNTRY_ID TYPE INTEGER;
```

```
CREATE TABLE country (COUNTRY_ID varchar(3)

, COUNTRY_NAME varchar(45)

, REGION_ID decimal(10,0)
```

```
);
```

✓ ● 
```
CREAT TABLE country (COUNTRY_ID INTEGER PRIMARY KEY
, COUNTRY_NAME varchar(45) ,
, REGION_ID decimal(10,0)
);
```

9 5 / 5 points

Which of the following reserved words would be used to find the top sales in a given region without using an aggregator?

○ FIRST

✓ ● LIMIT

○ TOP

10 5 / 5 points

Which type of SQL statement allows you to CREATE, ALTER, or DROP records in a table?

○ Transaction Control Language (TCL)

○ Data Control Language (DCL)

✓ ● Data Definition Language (DDL)

○ Data Manipulation Language (DML)

11 5 / 5 points

The ANSI/ISO SQL standard defines how to properly write SQL programming code.

○ True

✓ ● False

12 5 / 5 points

Select the SQL query that will correctly insert a row of data in the following table

product

| | |
|---|---|
| Id | Integer |
| prod_name | Varchar(30) |
| price | numeric |

| | |
|---|---|
| Id | Integer |
| intro_year | DATE |

```
INSERT INTO product

VALUES (1,AdvantaX, 32.99, 2021-02-01);
```

✓
```
INSERT INTO product

VALUES (1,'AdvantaX', 32.99, '2021-02-01');
```

```
 UPDATE product

SET VALUES = (1, 'AdvantaX', 32.99, '2021-02-01');
```

13 5 / 5 points

Which of the following is a limitation of using aggregates in queries?

- [ ] Aggregates take longer to write
- ✓ [x] Aggregates cannot be nested.
- [ ] Aggregates require the use of a GROUP BY clause
- ✓ [x] All non-aggregate columns must be included in the GROUP BY clause

14 5 / 5 points

Which of the following is the correct way to assign a foreign key to a table? Assume there is no foreign key column in the table.

```
ALTER TABLE airplane

ADD CONSTRAINT FOREIGN KEY (lender_id);
```

```
ALTER TABLE airplane

ADD column lender_id SERIAL;


ALTER TABLE airplane

ADD CONSTRAINT lender_id

FOREIGN KEY (lender_id);
```

```
ALTER TABLE airplane
ADD COLUMN lender_id integer;
```

✓

```
ALTER TABLE airplane
ADD CONSTRAINT fk_lender
FOREIGN KEY (lender_id)
REFERENCES lender(id);
```

```
UPDATE TABLE airplane
SET FOREIGN KEY TO (lender_id);
```

---

**15** 5 / 5 points

Which of the following SQL statements will drop the column supervisor from the employee table?

```
UPDATE TABLE employee
DROP COLUMN employee;
```

✓

```
ALTER TABLE employee
DROP COLUMN supervisor;
```

```
ALTER TABLE employee
DELETE supervisor
```

```
UPDATE TABLE employee
DELETE column supervisor;
```

---

**16** 5 / 5 points

Which of the following SQL keywords allows you to ensure that only those records that are non-duplicative are returned in a query:

IDEAL

UNIQUE

✓ DISTINCT

DISCREET

17 5 / 5 points

Which of the following shorthand SQL keywords would be the best option for replacing an entire record when using the same id?

- UPDATE
- SELECT/INSERT
- INSERT
- ✓ REPLACE

18 5 / 5 points

Setting table columns to only accept certain data types of varying lengths is an example of what type of SQL data quality policy?

- Record Lock
- Transaction Policy
- ✓ Constraint

19 5 / 5 points

What is the default sort order when using the ORDER BY sorting clause?

- Alphabetical
- ✓ Ascending
- Descending

20 5 / 5 points

Using appropriate SQL statements, create the below table in your public schema and introduce the values of the cus_tier column given the following logic:

- If the customer is with the company less than two years and their order value is less than 1500, then their tier is 'bronze'
- If the customer is with the company two years or more, but less than 5 years, and their order values is equal to 1500 or more but less than 10000, then their tier is 'silver'
- If the customer is with the company 5 years or more but less than 10 years and their order value is more than 10000 but less than 20000, then their tier is 'gold'
- If the customer is with the company 10 years or more and their order value is 20000 or more than their tier is 'platinum'

customer

| Id | lastname | firstname | total sales | longevity | order_value | cus_tier |
|---|---|---|---|---|---|---|
| 1 | Jones | Alice | 47520.00 | 10 | 30000 | |
| 2 | Barber | John | 99100.00 | 1 | 500 | |
| 3 | McGraw | Lana | 110570.00 | 8 | 13210 | |
| 4 | Justice | Bernice | 82010.00 | 2 | 5000 | |
| 5 | Pedigrew | Alexander | 71700.00 | 2 | 5010 | |

What is the appropriate statement and output of the cus_tier column?

```
UPDATE public.customer
SET cus_tier = CASE
WHEN longevity < 2 AND order_value < 1500 THEN 'bronze'
WHEN longevity >= 2 AND longevity < 5 AND order_value >= 1500 AND order_value < 10000 THEN 'silver'
WHEN longevity >= 5 AND longevity < 10 AND order_value >= 10000 AND order_value <20000 THEN 'gold'
WHEN longevity >=10 AND order_value > 10000 THEN 'platinum'
END;
```

Output: PLATINUM, BRONZE, SILVER, GOLD, GOLD

✓

```
UPDATE public.customer
SET cus_tier = CASE
WHEN longevity < 2 AND order_value < 1500 THEN 'bronze'
WHEN longevity >= 2 AND longevity < 5 AND order_value >= 1500 AND order_value < 10000 THEN 'silver'
WHEN longevity >= 5 AND longevity < 10 AND order_value >= 10000 AND order_value <20000 THEN 'gold'
WHEN longevity >=10 AND order_value > 10000 THEN 'platinum'
END;
```

Output: PLATINUM, BRONZE, GOLD, SILVER, SILVER

```
FOR (longevity, order_value) IN public.customer:
SET cus_tier = CASE
WHEN longevity < 2 AND order_value < 1500 THEN 'bronze'
WHEN longevity >= 2 AND longevity < 5 AND order_value >= 1500 AND order_value < 10000 THEN 'silver'
WHEN longevity >= 5 AND longevity < 10 AND order_value >= 10000 AND order_value <20000 THEN 'gold'
WHEN longevity >=10 AND order_value > 10000 THEN 'platinum'
END;
```

OUTPUT: GOLD, PLATINUM, SILVER, BRONZE, BRONZE

```
SELECT id, lastname, firstname, total_sales, longevity, order_value,
CASE cust_tier
WHEN longevity < 2 AND order_value < 1500 THEN 'bronze'
```

```
WHEN longevity >= 2 AND longevity < 5 AND order_value >= 1500 AND order_value < 10000 THEN 'silver'

WHEN longevity >= 5 AND longevity < 10 AND order_value >= 10000 AND order_value <20000 THEN 'gold'

WHEN longevity >=10 AND order_value > 10000 THEN 'platinum'

END cus_tier

FROM public.customer;
```

Output: GOLD, BRONZE, PLATINUM, SILVER, SILVER

---