Name: Esteban Salazar Correa

Date: 2024-07-08

Code: 200582339

SECTION A

1. C
2. B
3. D
4. B
5. A

SECTION B

1.Local storage refers to store data on the same device that is running the database while cloud storage refers to store data on a remote way, internet accessible storage that is managed by cloud service providers as Amazon or google for example.

2. SQL is a programming language that is used to manage and manipulate relational databases, through this essential tool we can stablish a communication between an app and a database allowing us data manipulation and control and, in this way, maintain data-driven applications and systems.

3. The ALTER TABLE statement is used to modify the structure of an existing table.

```
ALTER TABLE employees
ADD COLUMN testing_date DATE;
```

4. The GROUP BY statement allow you to group rows that have a common value into summary rows

```
SELECT product_category, SUM(sales) AS total_sales
FROM sales_data
GROUP BY product_category;
```

5. CASE statement is used to perform conditional logic within a query, it evaluates a set of conditions and return a value as result of these conditions.

```
SELECT LASTNAME, salary,
   CASE
      WHEN salary < 50000 THEN 'Low Salary'
      ELSE 'High Salary'
   END AS salary_category
FROM
   employee;
```

6. CHAR(): It stores a fixed-length, not less and not more characters than the previous fixed-length.

VARCHAR(): Stores a variable-length string of characters.

DATE(): It is used to extract the date portion from a date/time value following a date format.

SECTION C

1.

```
CREATE DATABASE School;


USE School;


CREATE TABLE students(
   student_id INT primary key,
   student_name VARCHAR(50),
   student_age INT,
   registered_date DATE
);
```

2.

INSERT INTO students (student_id, student_name, student_age, registered_date) VALUES (1, 'Alice', 20, '2024-01-15');

INSERT INTO students (student_id, student_name, student_age, registered_date) VALUES (2, 'Bob', 22, '2024-02-20');

UPDATE students

SET student_age = 23

WHERE student_id = 2;

DELETE FROM students

WHERE student_id = 1;

3.

SELECT * FROM students;

SELECT * FROM students WHERE registered_date > '2024-01-31';

SELECT * FROM students WHERE student_age > 21;

4.

SELECT student_name

FROM students

ORDER BY student_name DESC;

SELECT student_name, student_age

FROM students

ORDER BY student_age DESC

LIMIT 1;

5.

```
SELECT AVG(student_age) AS average_student_age
FROM students;

SELECT COUNT(*) AS num_students_2024
FROM students
WHERE registered_date > 2024-00-00;

SELECT student_age, COUNT(*) AS num_students
FROM students
GROUP BY student_age;
```

6.

```
CREATE DATABASE Sales;

USE Sales;

CREATE TABLE cookie_sales(
   id INT auto_increment primary key,
   first_name VARCHAR(20) NOT NULL,
   sale_amount DECIMAL(5,2) NOT NULL,
   sale_date DATE NOT NULL
);

INSERT INTO cookie_sales (first_name, sale_amount, sale_date) VALUES ('Esteban', 85, '2024-07-08');
INSERT INTO cookie_sales (first_name, sale_amount, sale_date) VALUES ('Julian', 8, '2024-03-02');
INSERT INTO cookie_sales (first_name, sale_amount, sale_date) VALUES ('Pedro', 115, '2023-01-22');
INSERT INTO cookie_sales (first_name, sale_amount, sale_date) VALUES ('Johan', 50, '2022-09-18');
INSERT INTO cookie_sales (first_name, sale_amount, sale_date) VALUES ('Esteban', 205, '2024-05-08');
```

```
SELECT *
FROM cookie_sales
WHERE sale_amount > 10;

SELECT first_name, SUM(sale_amount) AS total_sales
FROM cookie_sales
GROUP BY first_name;
```

7.

```
SELECT first_name, SUM(sale_amount) AS total_sales
FROM cookie_sales
GROUP BY first_name
ORDER BY total_sales DESC;

SELECT first_name, ROUND(AVG(sale_amount), 2) AS average_sale_amount
FROM cookie_sales
GROUP BY first_name
ORDER BY average_sale_amount DESC;

SELECT first_name, MAX(sale_amount) AS max_sale_amount
FROM cookie_sales
GROUP BY first_name
ORDER BY max_sale_amount DESC
LIMIT 1;
```