Clarusway

Backend Workshop -1-

Workshop

Subject:

- Introduction to Backend Development

Learning Goals

- Having general information about the backend.

Introduction

- As developers, we should also be able to express what we know. This study was prepared to support this purpose.

Prerequisites

- We will use the VSCode you are familiar with.
- At the same time, we need to install Nodejs on our computer.

Lets start

1. What Is Backend Development ?

Answer:

Bakend development covers server-side web application logic and integration and activities, like writing APIs, creating libraries, and working with system components instead of frontend development, which focuses on customer-facing services and programs. Backend developers build code that allows a database and an application to communicate with one another. Backend developers take care and maintain the backend of a website, Including databases, servers, and apps, and they control what you don't see.

2. What is skills a backend developer must have ?

Answer:

- Back-End Programming Language
- Knowledge of Front-End Technology
- Knowledge of Backend Frameworks
- Version Control System
- Knowledge of Databases
- Knowledge of API

  It would be nice to have these

- Server Handling
- Data Structures and Algorithms
- Problem Solving
- Communication Skills

3. Why backend is important ?

Answer:

Backend development in the software development describe the programming and coding behind the user interface. Backend includes all of the functionality that happens on the server side, such as

- database interactions,

- business logic,
- routing.
- building the APIs
- services all of these that make up the backbone of web and mobile applications.

4.What is SQL?

Answer: SQL (Structured Query Language) is a language used to manage and manipulate relational databases. SQL is used to create database queries, insert, update, delete data, and alter database structures.

5. What is JOIN and what are its types?

Answer: JOIN is used to combine rows from two or more tables based on a related column. The main types of JOIN are:

- INNER JOIN: Returns only the matching rows between the tables.
- LEFT (OUTER) JOIN: Returns all rows from the left table and the matching rows from the right table.
- RIGHT (OUTER) JOIN: Returns all rows from the right table and the matching rows from the left table.
- FULL (OUTER) JOIN: Returns matching rows as well as non-matching rows from both tables.

6.What is the difference between WHERE and HAVING clauses in SQL?

Answer:

- WHERE: The WHERE clause is used to filter records before any groupings are made. It applies to individual rows.

- HAVING: The HAVING clause is used to filter records after the groupings are made. It applies to groups of rows created by the GROUP BY clause.

- Using WHERE to filter individual rows SELECT employee_id, name, department FROM employees WHERE salary > 5000;

- Using HAVING to filter groups of rows SELECT department, AVG(salary) AS average_salary FROM employees GROUP BY department HAVING AVG(salary) > 5000;

  In this example, the WHERE clause filters employees with a salary greater than 5000, while the HAVING clause filters departments with an average salary greater than 5000.

7. What is the difference between WHERE and HAVING clauses in SQL?

Answer:

- GROUP BY: Used to group rows that have the same values in specified columns into summary rows.
- HAVING: Used to filter records that work on summarized GROUP BY results.
- SELECT department, COUNT(*) FROM employees GROUP BY department HAVING COUNT(*) > 10;

8. What is the difference between PRIMARY KEY and FOREIGN KEY?

Answer:

- PRIMARY KEY: A field (or combination of fields) that uniquely identifies each record in a table. Each table can have only one primary key, and it cannot contain null values.
- FOREIGN KEY: A field (or combination of fields) in one table that uniquely identifies a row of another table. It is used to establish and enforce a link between the data in the two tables.

9. What is SQL Injection and how can it be prevented?

Answer: SQL Injection is a code injection technique that exploits a security vulnerability in an application's software by manipulating SQL queries. It can be prevented by:

- Using parameterized queries
- Using ORM (Object-Relational Mapping) tools
- Validating and sanitizing inputs
- Using prepared statements

10.What is an INDEX in SQL and why is it used?

Answer: An INDEX is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional space and slower write operations. Indexes are used to quickly locate data without having to search every row in a database table.

Backend
Teamwork
-1-

Teamwork

Subject: SQL

Learning Goals

- To be able to write SQL statements that will perform the desired query.

Introduction

We use the SQL language when performing operations on relational databases. You can perform many different operations on the DB with SQL, but this work only includes querying.

Lets start

Write SQL statements that produce the desired output.

1. WRITE A QUERY THAT RETURNS TRACK NAME AND ITS COMPOSER FROM TRACKS TABLE

```
SELECT Name, Composer FROM tracks;
```

2. WRITE A QUERY THAT RETURNS ALL COLUMNS FROM TRACKS TABLE

```sql
SELECT * FROM tracks;
```

3. WRITE A QUERY THAT RETURNS THE UNIQUE NAME OF COMPOSERS OF EACH TRACK

```sql
SELECT DISTINCT Composer FROM tracks;
```

4. WRITE A QUERY THAT RETURNS UNIQUE ALBUMID, MEDIATYPEID FROM TRACKS TABLE

```sql
SELECT DISTINCT AlbumId, MediaTypeId FROM tracks;
```

5. WRITE A QUERY THAT RETURNS TRACK NAME AND TRACKID OF 'Jorge Ben'

```sql
SELECT Name, TrackId
FROM tracks
WHERE Composer = 'Jorge Ben';
```

6. WRITE A QUERY THAT RETURNS ALL INFO OF THE INVOICES OF WHICH TOTAL AMOUNT IS GREATER THAN $25

```sql
SELECT *
FROM invoices
WHERE Total > 25;
```

7. WRITE A QUERY THAT RETURNS ALL INFO OF THE INVOICES OF WHICH TOTAL AMOUNT IS LESS THAN $15. JUST RETURN 5 ROWS

```sql
SELECT *
FROM invoices
WHERE Total < 15
LIMIT 5;
```

8. WRITE A QUERY THAT RETURNS ALL INFO OF THE INVOICES OF WHICH TOTAL AMOUNT IS GREATER THAN $10. THEN SORT THE TOTAL AMOUNTS IN DESCENDING ORDER, LASTLY DISPLAY TOP 2 ROWS

```sql
SELECT *
FROM invoices
WHERE Total > 10
```

```
ORDER BY Total DESC
LIMIT 2;
```

9. WRITE A QUERY THAT RETURNS ALL INFO OF THE INVOICES OF WHICH BILLING COUNTRY IS NOT CANADA. THEN SORT THE TOTAL AMOUNTS IN ASCENDING ORDER, LASTLY DISPLAY TOP 10 ROWS

```
SELECT *
FROM invoices
WHERE NOT BillingCountry = 'CANADA'
ORDER BY Total ASC
LIMIT 10;
```

10. WRITE A QUERY THAT RETURNS INVOICEID, CUSTOMERID AND TOTAL DOLLAR AMOUNT FOR EACH INVOICE. THEN SORT THEM FIRST BY CUSTOMERID IN ASCENDING, THEN TOTAL DOLLAR AMOUNT IN DESCENDING ORDER.

```
SELECT InvoiceId, CustomerId, Total
FROM invoices
ORDER BY CustomerId, Total DESC;
```

11. WRITE A QUERY THAT RETURNS ALL TRACK NAMES THAT START WITH 'B' AND END WITH 'S'

```
SELECT Name
FROM tracks
WHERE Name LIKE 'B%' AND Name LIKE '%s';
(ALTERNATIVE -- WHERE name LIKE 'B%s');
```

12. WRITE A QUERY THAT RETURNS THE NEWEST DATE AMONG THE INVOICE DATES BETWEEN 2008 AND 2011

```
SELECT InvoiceDate
FROM invoices
WHERE InvoiceDate BETWEEN '2008-01-01' AND '2012-01-01'
ORDER BY InvoiceDate DESC
LIMIT 1;
```

13. WRITE A QUERY THAT RETURNS THE FIRST AND LAST NAME OF THE CUSTOMERS WHO HAVE ORDERS FROM NORWAY AND BELGIUM

```
SELECT FirstName, LastName
FROM customers
WHERE Country IN ('Belgium', 'Norway')
```

14. WRITE A QUERY THAT RETURNS THE TRACK NAMES OF ‘ZAPPA’

```
SELECT Composer, Name
FROM tracks
WHERE Composer LIKE '%Zappa';
```

15. HOW MANY TRACKS AND INVOICES ARE THERE IN THE DIGITAL MUSIC

```
STORE, DISPLAY SEPERATELY
SELECT COUNT(*)
FROM tracks;
SELECT COUNT(*)
FROM invoices;
```

16. HOW MANY COMPOSERS ARE THERE IN THE DIGITAL MUSIC STORE

```
SELECT COUNT(DISTINCT Composer)
FROM tracks;
```

17. HOW MANY TRACKS DOES EACH ALBUM HAVE, DISPLAY ALBUMID AND NUMBER OF TRACKS SORTED FROM HIGHEST TO LOWEST

```
SELECT AlbumId,
COUNT(*) AS number_of_tracks
FROM tracks
GROUP BY AlbumId
ORDER BY number_of_tracks DESC;
```

18. WRITE A QUERY THAT RETURNS TRACK NAME HAVING THE MINIMUM AND MAXIMUM DURATION, DISPLAY SEPERATELY

```
SELECT Name, MIN(Milliseconds) AS Min, MAX(Milliseconds) as Max
FROM tracks;
```

19. WRITE A QUERY THAT RETURNS THE TRACKS HAVING DURATION LESS THAN THE AVERAGE DURATION

```
SELECT *
FROM tracks
WHERE Milliseconds < 393599.212103911
```

```sql
SELECT *
FROM tracks
WHERE Milliseconds < (
SELECT AVG(Milliseconds)
FROM tracks);
```

20. WRITE A QUERY THAT RETURNS THE TOTAL NUMBER OF EACH COMPOSER's TRACK.

```sql
SELECT Composer, COUNT(*)
FROM tracks
GROUP BY Composer;

SELECT Composer, COUNT(Composer)
FROM tracks
GROUP BY Composer;

SELECT Composer, COUNT(Composer)
FROM tracks
WHERE Composer IS NOT NULL
GROUP BY Composer;
```

21. WRITE A QUERY THAT RETURNS THE GENRE OF EACH TRACK.

```sql
SELECT tracks.Name, genres.Name
FROM tracks
JOIN genres
ON tracks.GenreId = genres.GenreId;

SELECT t.Name, g.Name
FROM tracks t
JOIN genres g
ON t.GenreId = g.GenreId;
```

22. WRITE A QUERY THAT RETURNS THE ARTIST's ALBUM INFO.

```sql
SELECT *
FROM artists
LEFT JOIN albums
ON albums.ArtistId = artists.ArtistId
```

23. WRITE A QUERY THAT RETURNS THE MINIMUM DURATION OF THE TRACK IN EACH ALBUM. DISPLAY ALBUMID, ALBUM TITLE AND DURATION OF THE TRACK. THEN SORT THEM FROM HIGHEST TO LOWEST

```sql
SELECT tracks.AlbumId, albums.Title,
MIN(tracks.Milliseconds) AS min_duration
FROM tracks
JOIN albums
ON tracks.AlbumId = albums.AlbumId
GROUP BY tracks.AlbumId, albums.Title
ORDER BY min_duration DESC;
```

24. WRITE A QUERY THAT RETURNS ALBUMS WHOSE TOTAL DURATION IS HIGHER THAN 60 MIN. DISPLAY ALBUM TITLE AND THEIR DURATIONS. THEN SORT THE RESULT FROM HIGHEST TO LOWEST

```sql
SELECT albums.Title, SUM(tracks.Milliseconds) AS total_duration
FROM tracks
JOIN albums ON tracks.AlbumId = albums.AlbumId
GROUP BY tracks.AlbumId
HAVING total_duration > 3600000
ORDER BY total_duration DESC;
```

25. WRITE A QUERY THAT RETURNS TRACKID, TRACK NAME AND ALBUMID INFO OF THE ALBUM WHOSE TITLE ARE ‘Prenda Minha’, 'Heart of the Night' AND 'Out Of Exile'.

```
SELECT Trackid, Name, Albumid
FROM tracks
WHERE albumid IN (
SELECT AlbumId
FROM albums
WHERE Title IN ('Prenda Minha', 'Heart of the Night', 'Out Of Exile'));
```

☺ Thanks for Attending ✍

Clarusway