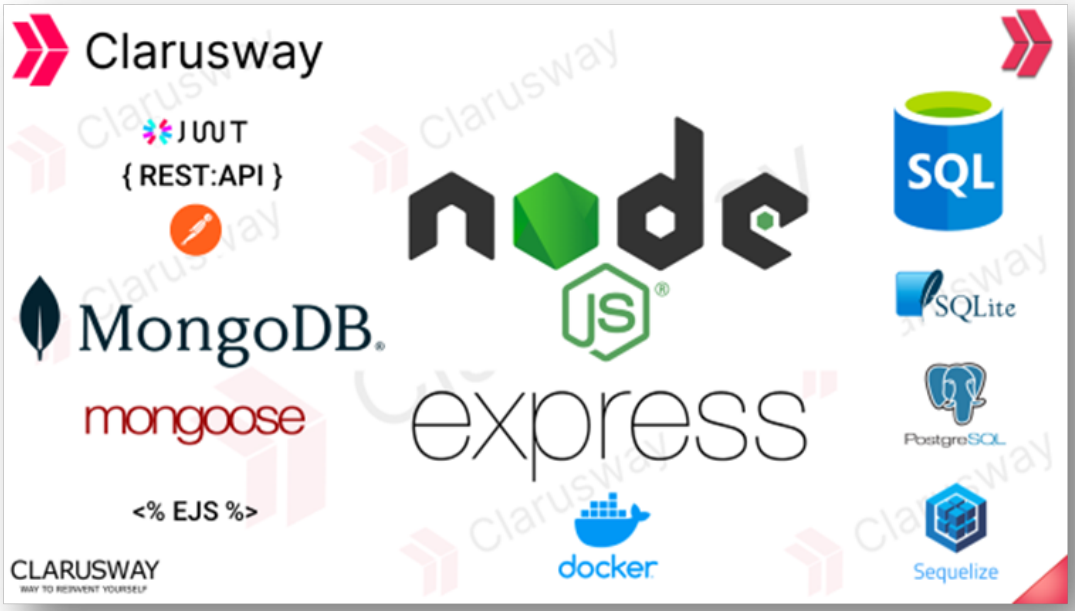
Clarusway



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 track_name, composer
FROM tracks;
```

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

```
SELECT *
FROM tracks;
```

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

```
SELECT DISTINCT track_name, composer
FROM tracks;
```

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

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

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

```
SELECT track_name, trackid
FROM tracks
WHERE composer = 'Jorge Be(n)';
```

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

```
SELECT *
FROM invoices
WHERE total > 25.0;
```

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

```
SELECT * FROM invoices
WHERE total < 15.0
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

```
SELECT * FROM invoices
WHERE total > 10.0
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 billing_country <> '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 ASC, total DESC;
```

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

```
SELECT track_name FROM tracks
WHERE track_name LIKE 'B%S';
```

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

```

```

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

```
SELECT c.first_name, c.last_name FROM customers c
JOIN invoices i ON c.customerid = i.customerid
WHERE i.billing_country IN ('Norway', 'Belgium');
```

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

```
SELECT track_name FROM tracks
WHERE composer = 'Zappa';
```

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

```
 STORE, DISPLAY SEPERATELY
```

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

```
SELECT COUNT(DISTINCT composer) AS composer_count
FROM tracks
WHERE media_type = 'Digital Media';
```

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

```
SELECT AlbumId, COUNT(TrackId) AS num_tracks FROM tracks
GROUP BY AlbumId
ORDER BY num_tracks DESC;
```

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

```

```

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

```
SELECT track_name, milliseconds
FROM tracks
WHERE milliseconds < (SELECT AVG(milliseconds) FROM tracks);
```

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

```
SELECT composer, COUNT(*) AS track_count FROM tracks
GROUP BY composer;
```

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

```
SELECT track_name, genre_name FROM tracks
JOIN genres ON tracks.genreid = genres.genreid;
```

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

```
SELECT albums.albumid, albums.title AS album_title, artists.name AS artist_name FROM albums
JOIN artists ON albums.artistid = artists.artistid
WHERE artists.name = 'YourArtistName';
```

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

```

```

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

```

```

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'.

```

```

☺ **Thanks for Attending** ✍

Clarusway