

## SQL WORKSHOP DOCUMENT – STUDENT VERSION

AUGUST 31 TUESDAY, 2021



Reference : Chinook db

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

```
select * FROM tracks;
```

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

```
select 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 TrackId, name 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

```
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

```
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

```
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 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 ASC, 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%%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 "2009-01-01" 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, Country FROM customers WHERE Country IN ("Belgium", "Norway");
```

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

```
SELECT * FROM tracks WHERE Composer LIKE "%Zappa%";
```

15. HOW MANY TRACKS AND INVOICES ARE THERE IN THE DIGITAL MUSIC STORE, DISPLAY SEPERATELY

```
1.SELECT * FROM tracks 2. SELECT * 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

```
1.SELECT name , min(Milliseconds) from tracks 2.SELECT name , max(Milliseconds) from tracks
```

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

```
SELECT TrackId, Milliseconds FROM tracks WHERE Milliseconds < ( SELECT avg(Milliseconds) from tracks) GROUP BY TrackId;
```

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

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

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

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

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

```
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**

```
SELECT tracks.AlbumId, albums.Title, min(Milliseconds) FROM tracks INNER JOIN albums ON tracks.AlbumId = albums.AlbumId GROUP BY tracks.AlbumId ORDER BY Milliseconds 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**

```
SELECT tracks.AlbumId, 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 Milliseconds DESC;
```

**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" ));
```