Class 16: 3/24/2025

Final Exam: Wednesday, May 14, 5:30 - 7:30 PM.

```
SELECT *
FROM Album a
LEFT JOIN Artist ar
    ON Artist.ArtistId = Album.ArtistId
WHERE upper(a.Title) like '%R%O%C%K%'

SELECT SUBSTRing(name,1,1) as first_character,
    count(*) as ctr
FROM Artist a
GROUP BY first_character
order by ctr DESC

SELECT * from Customer c

SELECT c.Country , count(*), count(state),
    count(state)/count(*)
FROM Customer c
--WHERE state is  null
GROUP BY c.Country
ORDER BY 4 DESC
```

```
SELECT *, e.HireDate - e.BirthDate,
       (JULIANDAY(e.HireDate)-
JULIANDAY(e.BirthDate ))/365.24
FROM Employee e limit 5
```

Even More SQL

Find all the tracks that are longer than the average track length

```
SELECT AVG(Milliseconds) FROM Track
```

```
SELECT Name, Milliseconds
FROM Track
WHERE Milliseconds >
```

```
SELECT Name, Milliseconds
FROM Track
WHERE Milliseconds > (SELECT AVG(Milliseconds) FROM Track);
```

Even More SQL

List all the customers that have not made a purchase yet

```
SELECT DISTINCT CustomerId FROM Invoice
```

```
SELECT * FROM Customer
WHERE CustomerId
```

```
SELECT * FROM Customer
WHERE CustomerId NOT IN
                        (SELECT DISTINCT CustomerId FROM Invoice);
```

```
SELECT * FROM Customer c
LEFT JOIN Invoice i ON c.customerID=i.customerid
WHERE i.customerid is null
```

Even More SQL

Print all albums with the total number of tracks on each album

```
SELECT * FROM Album LIMIT 5
```

```
SELECT * FROM Track LIMIT 5
```

```
SELECT Album.Title, COUNT(Track.TrackId) AS TrackCount
FROM Album
LEFT JOIN Track ON Album.AlbumId = Track.AlbumId
GROUP BY Album.Title
ORDER BY TrackCount DESC;
```

Even More SQL

List all employees along with the name of their manager (SELF - JOIN)

```
SELECT * FROM Employee LIMIT 5
```

```
SELECT e1.FirstName || ' ' || e1.LastName AS Employee,
       e2.FirstName || ' ' || e2.LastName AS Manager
FROM Employee e1
LEFT JOIN Employee e2 ON e1.ReportsTo = e2.EmployeeId;
```

Even More SQL

Find the artist(s) with the most albums

```
SELECT ArtistId, COUNT(AlbumId) AS AlbumCount
FROM Album
GROUP BY ArtistId
```

```
SELECT Artist.Name, AlbumCounts.AlbumCount
FROM Artist
JOIN (
    SELECT ArtistId, COUNT(AlbumId) AS AlbumCount
    FROM Album
    GROUP BY ArtistId
) AS AlbumCounts ON Artist.ArtistId = AlbumCounts.ArtistId
ORDER BY AlbumCounts.AlbumCount DESC
LIMIT 1;
```

Even More SQL

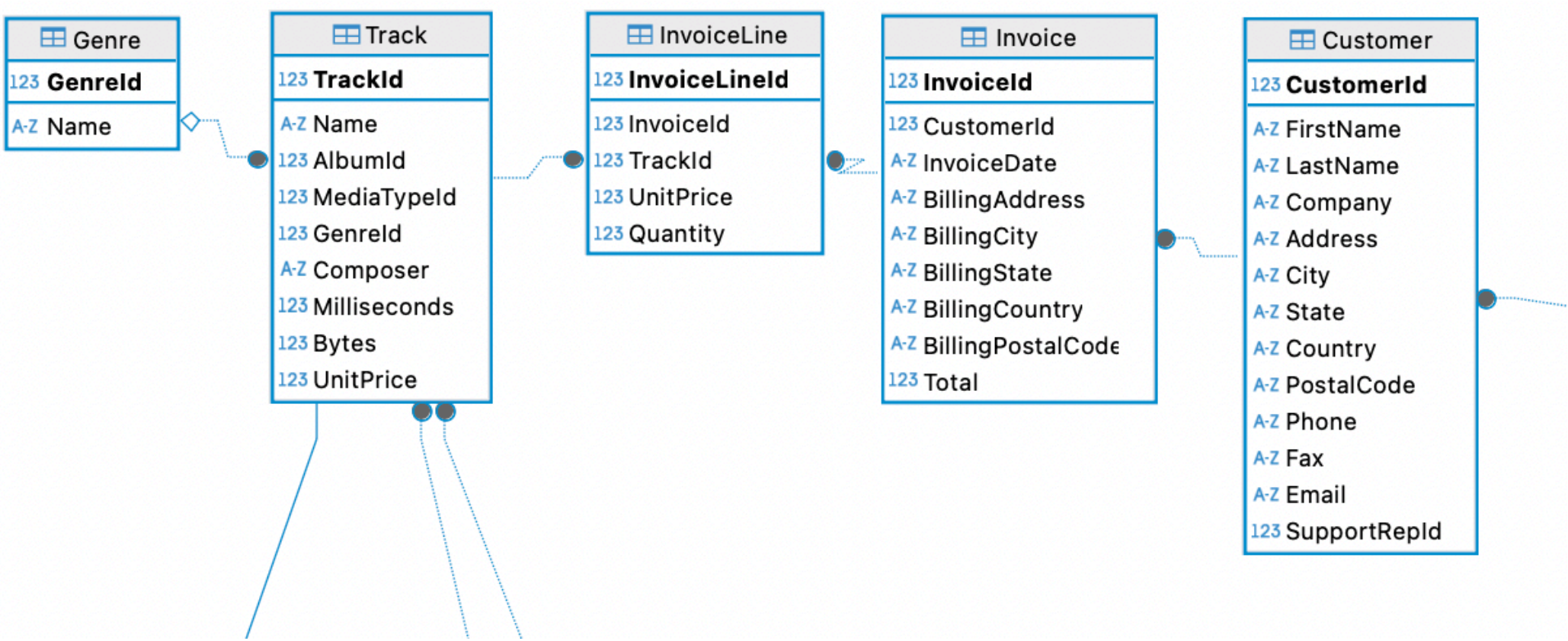
List customers and their total spending using a subquery

```
SELECT * FROM Customer LIMIT 5
SELECT * FROM Invoice LIMIT 5
```

```
SELECT Customer.FirstName, Customer.LastName, TotalSpent.Total
FROM Customer
JOIN (
    SELECT CustomerId, SUM(Total) AS Total
    FROM Invoice
    GROUP BY CustomerId
) AS TotalSpent ON Customer.CustomerId = TotalSpent.CustomerId
ORDER BY TotalSpent.Total DESC;
```

Even More SQL

By customer, find total number of genres



Even More SQL

By customer, find total number of genres

```
SELECT * FROM customer LIMIT 5
SELECT * FROM Invoice LIMIT 5
SELECT * FROM InvoiceLine LIMIT 5
SELECT * FROM Track LIMIT 5
SELECT * FROM Genre LIMIT 5
```

```
SELECT Customer.FirstName, Customer.LastName, COUNT(DISTINCT
Genre.GenreId) AS GenreCount
FROM Customer
JOIN Invoice ON Customer.CustomerId = Invoice.CustomerId
JOIN InvoiceLine ON Invoice.InvoiceId = InvoiceLine.InvoiceId
JOIN Track ON InvoiceLine.TrackId = Track.TrackId
JOIN Genre ON Track.GenreId = Genre.GenreId
GROUP BY Customer.CustomerId
ORDER BY GenreCount DESC;
```

Even More SQL

Find all Genres and Purchased tracks and available tracks and percent of the available purchased

```
SELECT * FROM Genre LIMIT 5
SELECT * FROM Track LIMIT 5
SELECT * FROM InvoiceLine LIMIT 5
```

```
SELECT Genre.Name AS Genre, COUNT(InvoiceLine.TrackId) AS
PurchasedTracks ,
           count(Track.TrackId ) as TotalTracks,
           1.000*COUNT(InvoiceLine.TrackId)/count(Track.TrackId )
as per_purchased
FROM Genre
FULL OUTER JOIN Track ON Genre.GenreId = Track.GenreId
FULL OUTER JOIN InvoiceLine ON Track.TrackId = InvoiceLine.TrackId
GROUP BY Genre.Name
ORDER BY 4 desc;
```