ITM 209

SQL Project Answer Sheet

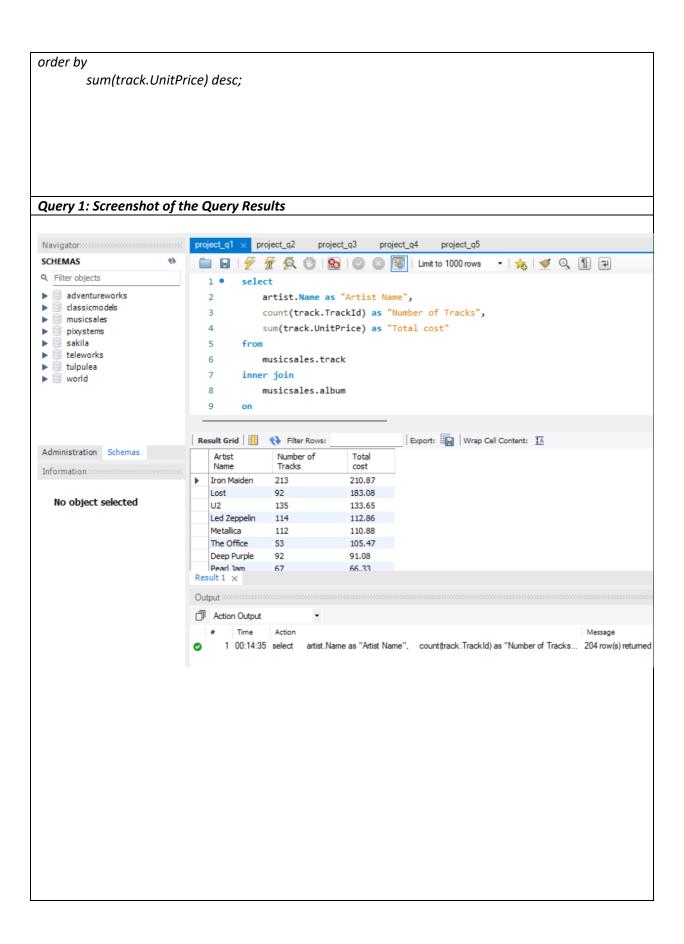
This answer sheet is for you to enter your results for SQL Project. Please rename this file as: firstName.lastName-SQLProject.docx.

In the appropriate text boxes below, you should include the following for each of the queries in order to receive full points. Note that there may be more spaces for queries than the project you are working on as this answer sheet is used for multiple projects. Each project may have differing numbers of queries so only fill out as many as specified in the project instructions:

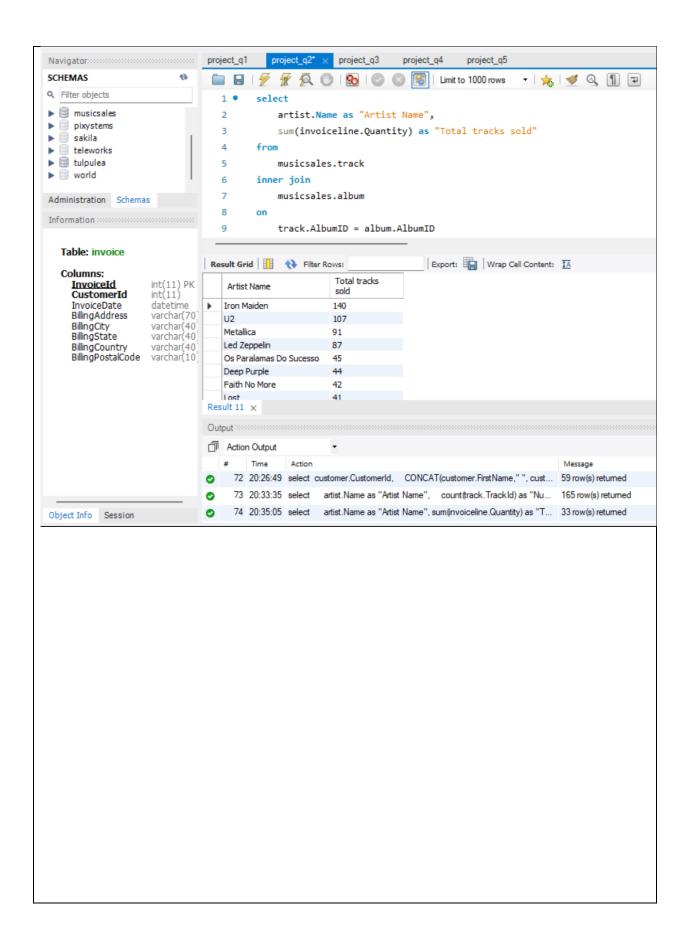
- 1. Query statement used. Query logic should be copied and pasted from your MySQL view in text form. If you are having trouble copying and pasting, look for a pop-up within VDI asking you to enable this feature.
- **2. Screenshots of query results.** Your query results *must* include the following within each screenshot:
 - Query Logic/ Statements
 - Result Grid
 - If more rows are returned than possible to fit in one screenshot, show at least the first five rows.
 - Action Output (at the bottom)
 - Action output must correspond to the correct query.
 - Schema including your MSU netID (on left under Navigator)

Query 1: SQL Statement Used (TEXT ONLY)

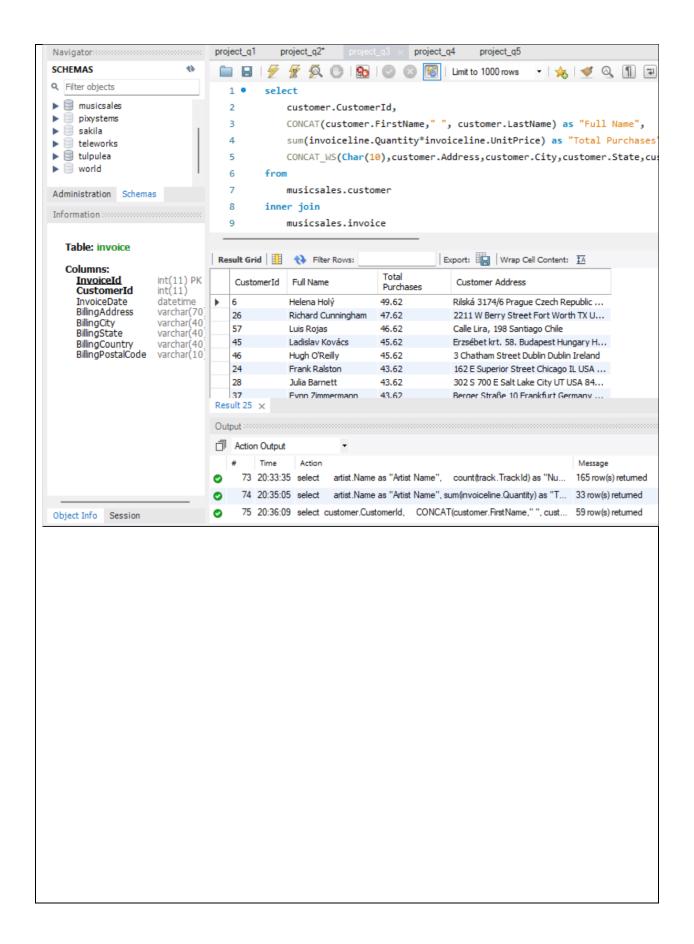
```
select
  artist.Name as "Artist Name",
  count(track.TrackId) as "Number of Tracks",
  sum(track.UnitPrice) as "Total cost"
from
        musicsales.track
inner join
        musicsales.album
on
        track.AlbumID = album.AlbumID
inner join
        musicsales.artist
on
        album.ArtistId = artist.ArtistId
group by
        artist.ArtistId
```



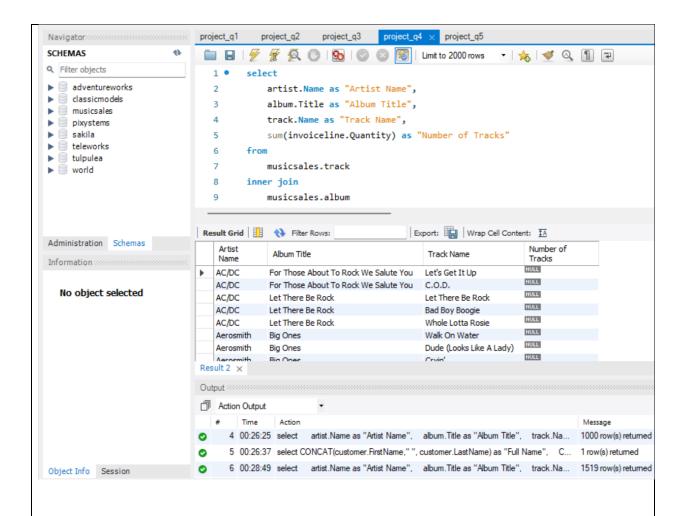
```
Query 2: SQL Statement Used (TEXT ONLY)
select
 artist.Name as "Artist Name",
       sum(invoiceline.Quantity) as "Total tracks sold"
from
       musicsales.track
inner join
       musicsales.album
on
       track.AlbumID = album.AlbumID
inner join
       musicsales.artist
on
       album.ArtistId = artist.ArtistId
inner join
       musicsales.invoiceline
on
       track.TrackId = invoiceline.TrackId
group by
       artist.ArtistId
having
       sum(invoiceline.Quantity) >= 20
order by
       sum(invoiceline.Quantity) desc;
Query 2: Screenshot of the Query Results
```



Query 3: SQL Statement Used (TEXT ONLY) select customer.CustomerId, CONCAT(customer.FirstName," ", customer.LastName) as "Full Name", sum(invoiceline.Quantity*invoiceline.UnitPrice) as "Total Purchases", CONCAT_WS(Char(10),customer.Address,customer.City,customer.State,customer.Country,customer.P ostalCode) as "Customer Address" from musicsales.customer inner join musicsales.invoice on customer.CustomerId = invoice.CustomerId inner join musicsales.invoiceline on invoice.InvoiceId = invoiceline.InvoiceId group by customer.CustomerId having sum(invoiceline.Quantity*invoiceline.UnitPrice) >= 35 order by sum(invoiceline.Quantity*invoiceline.UnitPrice) desc; **Query 3: Screenshot of the Query Results**



```
Query 4: SQL Statement Used (TEXT ONLY)
select
  artist.Name as "Artist Name",
  album.Title as "Album Title",
  track.Name as "Track Name",
  sum(invoiceline.Quantity) as "Number of Tracks"
from
        musicsales.track
inner join
        musicsales.album
on
        track.AlbumID = album.AlbumID
inner join
        musicsales.artist
on
        album.ArtistId = artist.ArtistId
left join
        musicsales.invoiceline
on
        track.TrackId = invoiceline.TrackId
group by
        track.TrackId
having
        sum(invoiceline.Quantity) is null;
Query 4: Screenshot of the Query Results
```



Query 5: SQL Statement Used (TEXT ONLY)

select

CONCAT(customer.FirstName," ", customer.LastName) as "Full Name",

CONCAT_WS(Char(10),invoice.BillingAddress,invoice.BillingCity,invoice.BillingState,invoice.BillingCountry,invoice.BillingPostalCode) as "Full Invoice Address",

CONCAT_WS(Char(10),customer.Address,customer.City,customer.State,customer.Country,cust omer.PostalCode) as "Address In FIIe" from

musicsales.customer

inner join

	musicsales.invoice
on	
	customer.CustomerId = invoice.CustomerId
where	
	customer.PostalCode <> invoice.BillingPostalCode;
Query 5: Screenshot of the Query Results	
· /	• • •

