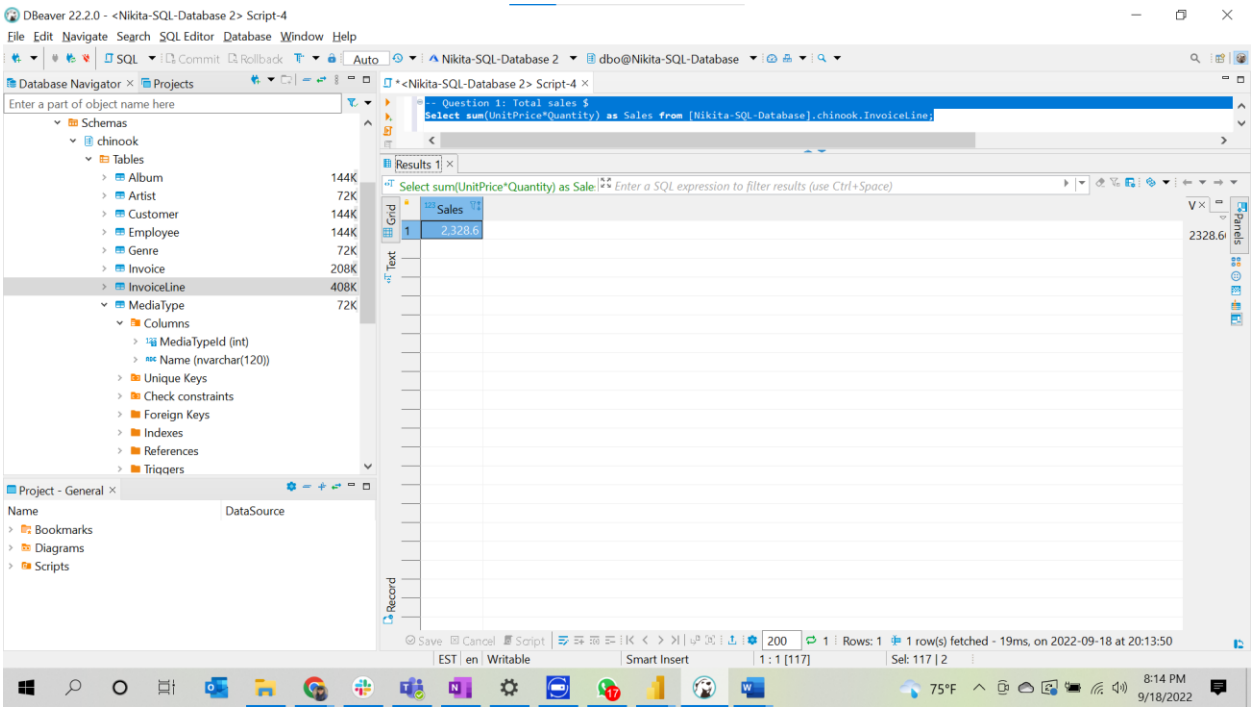
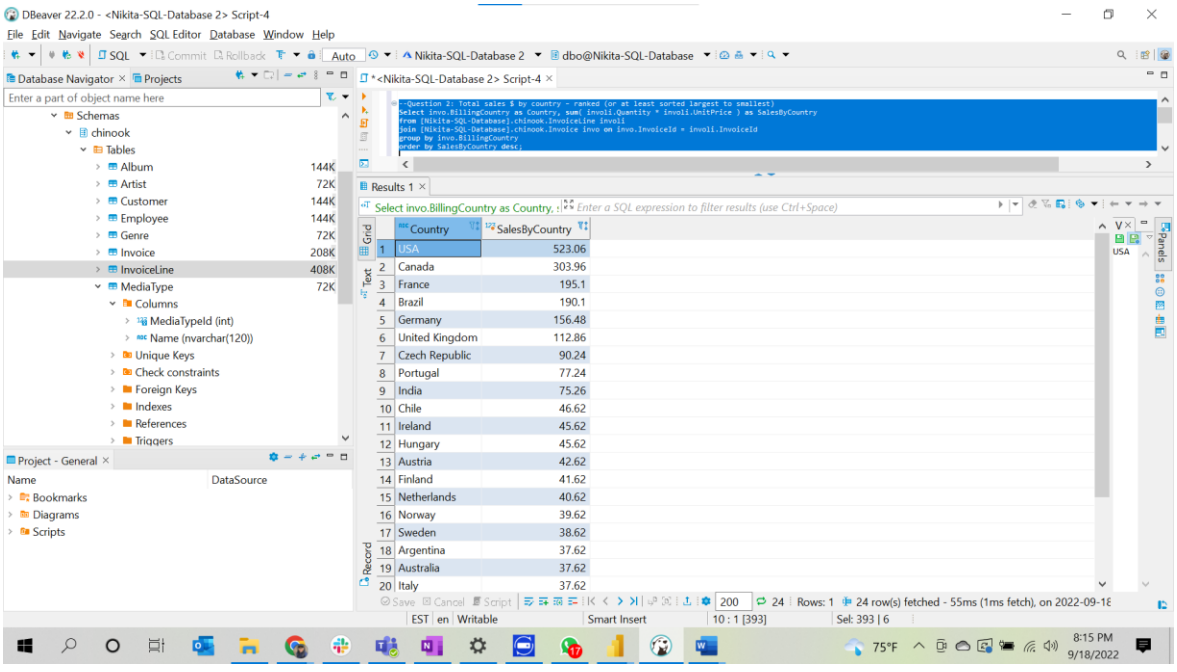
1. AZURE SQL

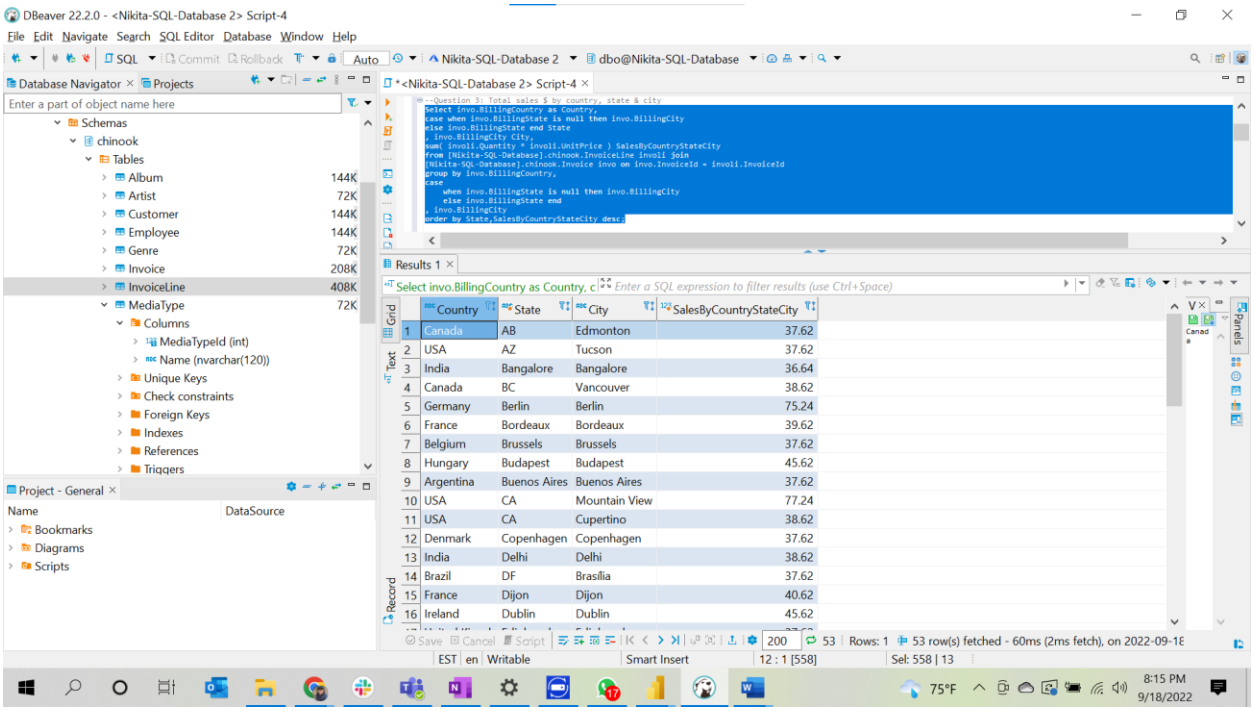
```
-- Question 1: Total sales $
Select sum(UnitPrice*Quantity) as Sales from [Nikita-SQL-Database].chinook.InvoiceLine;
```



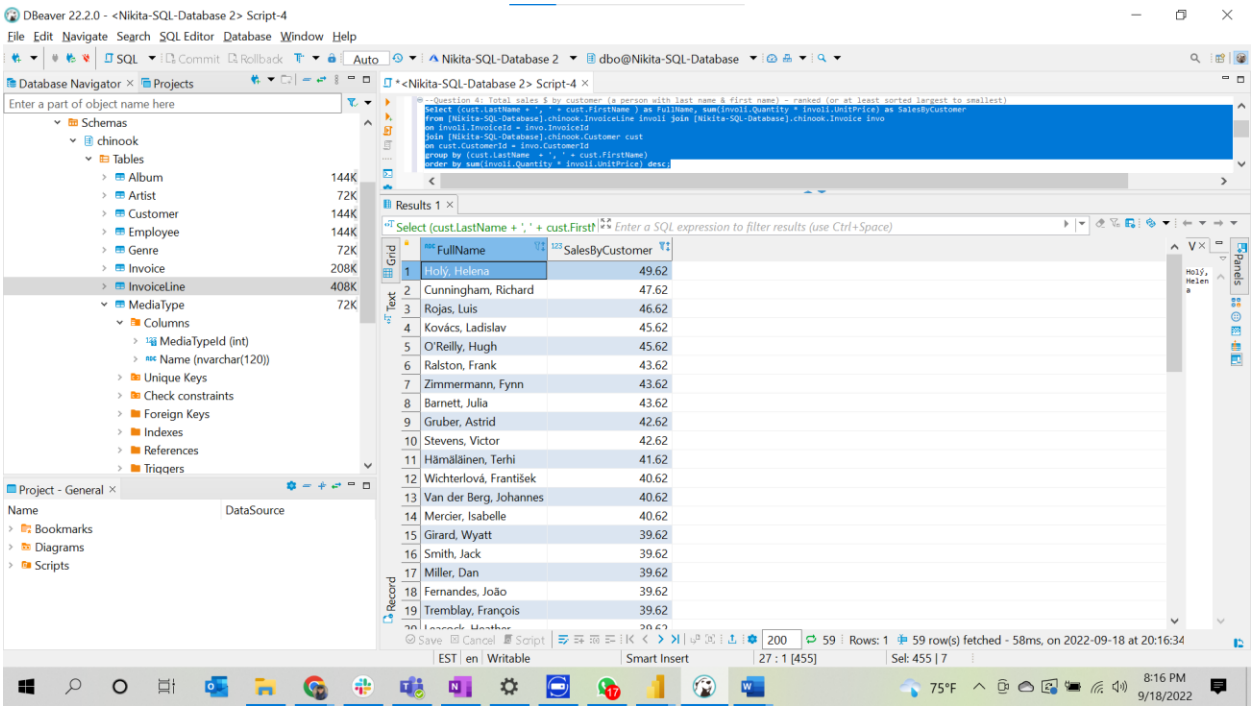

```
--Question 2: Total sales $ by country - ranked (or at least sorted largest to smallest)
Select invo.BillingCountry as Country, sum( involi.Quantity * involi.UnitPrice ) as SalesByCountry
from [Nikita-SQL-Database].chinook.InvoiceLine involi
join [Nikita-SQL-Database].chinook.Invoice invo on invo.InvoiceId = involi.InvoiceId
group by invo.BillingCountry
order by SalesByCountry desc;
```



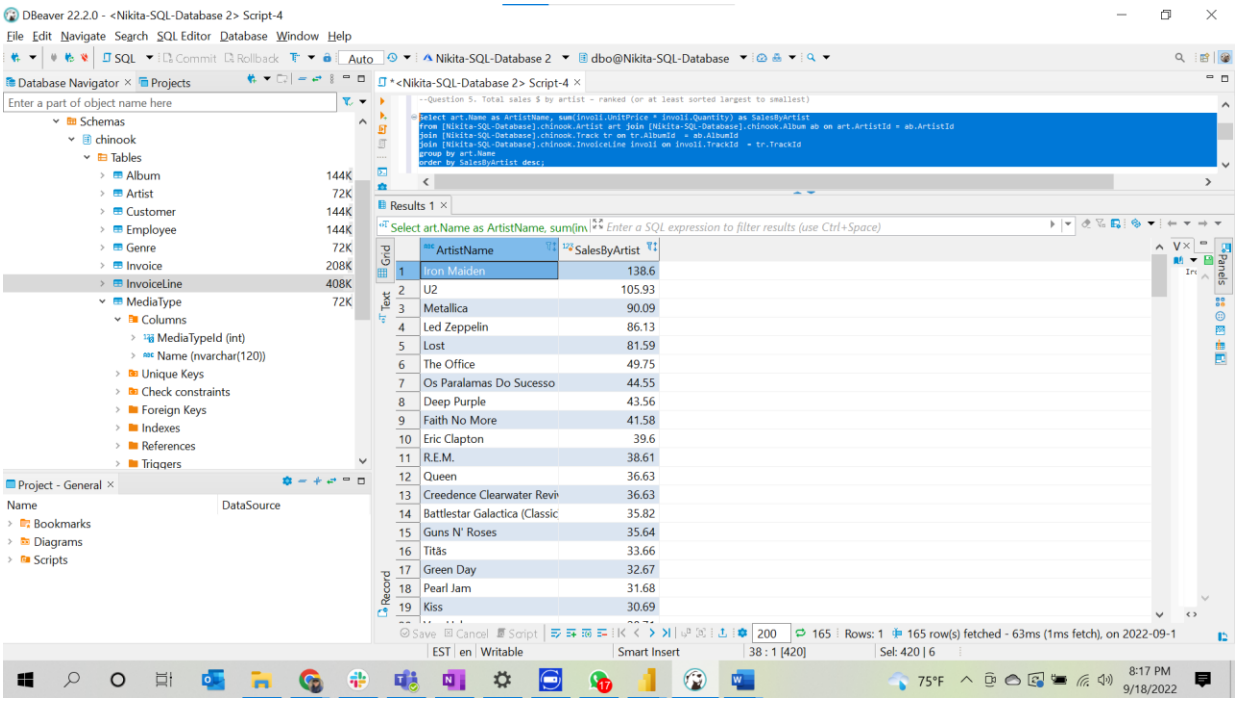
```sql
--Question 3: Total sales $ by country, state & city
Select invo.BillingCountry as Country,
case when invo.BillingState is null then invo.BillingCity
else invo.BillingState end State
, invo.BillingCity City,
sum( involi.Quantity * involi.UnitPrice ) SalesByCountryStateCity
from [Nikita-SQL-Database].chinook.InvoiceLine involi join
[Nikita-SQL-Database].chinook.Invoice invo on invo.InvoiceId = involi.InvoiceId
group by invo.BillingCountry,
case
    when invo.BillingState is null then invo.BillingCity
    else invo.BillingState end
, invo.BillingCity
order by State,SalesByCountryStateCity desc;
```



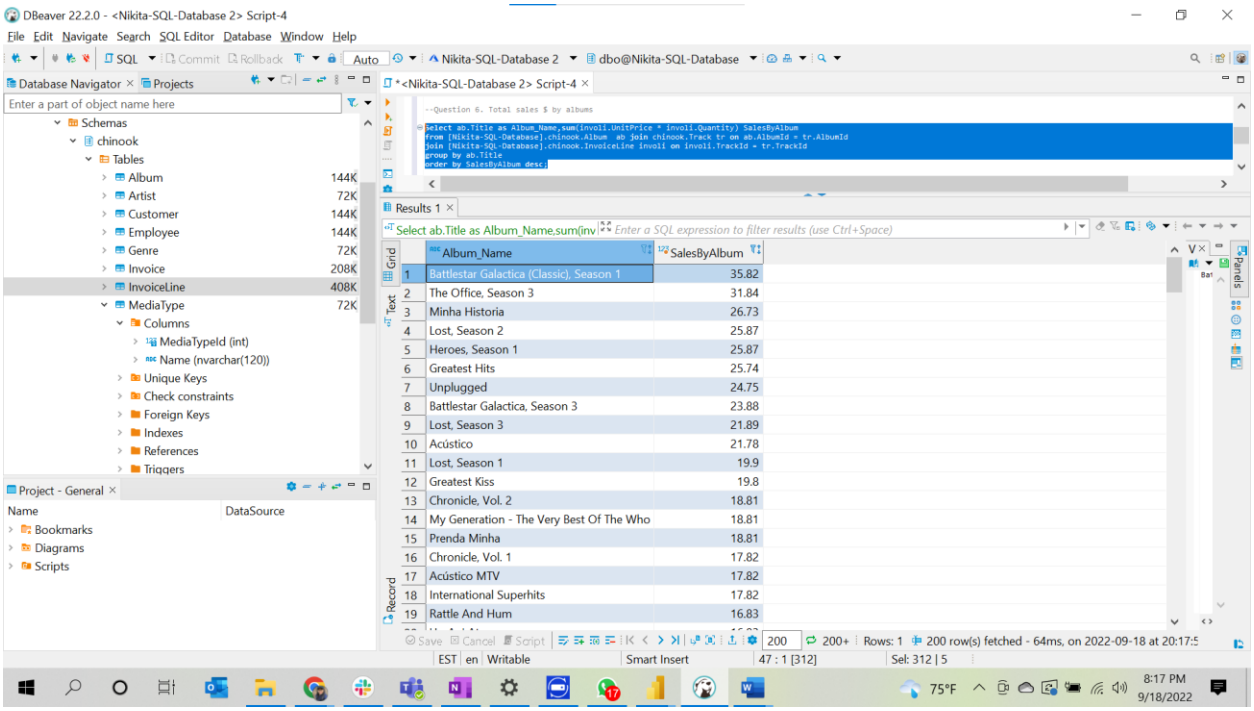

```sql
--Question 4: Total sales $ by customer (a person with last name & first name) – ranked (or at least sorted largest to smallest)
Select (cust.LastName + ', ' + cust.FirstName ) as FullName, sum(involi.Quantity * involi.UnitPrice) as SalesByCustomer
from [Nikita-SQL-Database].chinook.InvoiceLine involi join [Nikita-SQL-Database].chinook.Invoice invo
on involi.InvoiceId = invo.InvoiceId
join [Nikita-SQL-Database].chinook.Customer cust
on cust.CustomerId = invo.CustomerId
group by (cust.LastName  + ', ' + cust.FirstName)
order by sum(involi.Quantity * involi.UnitPrice) desc;
```



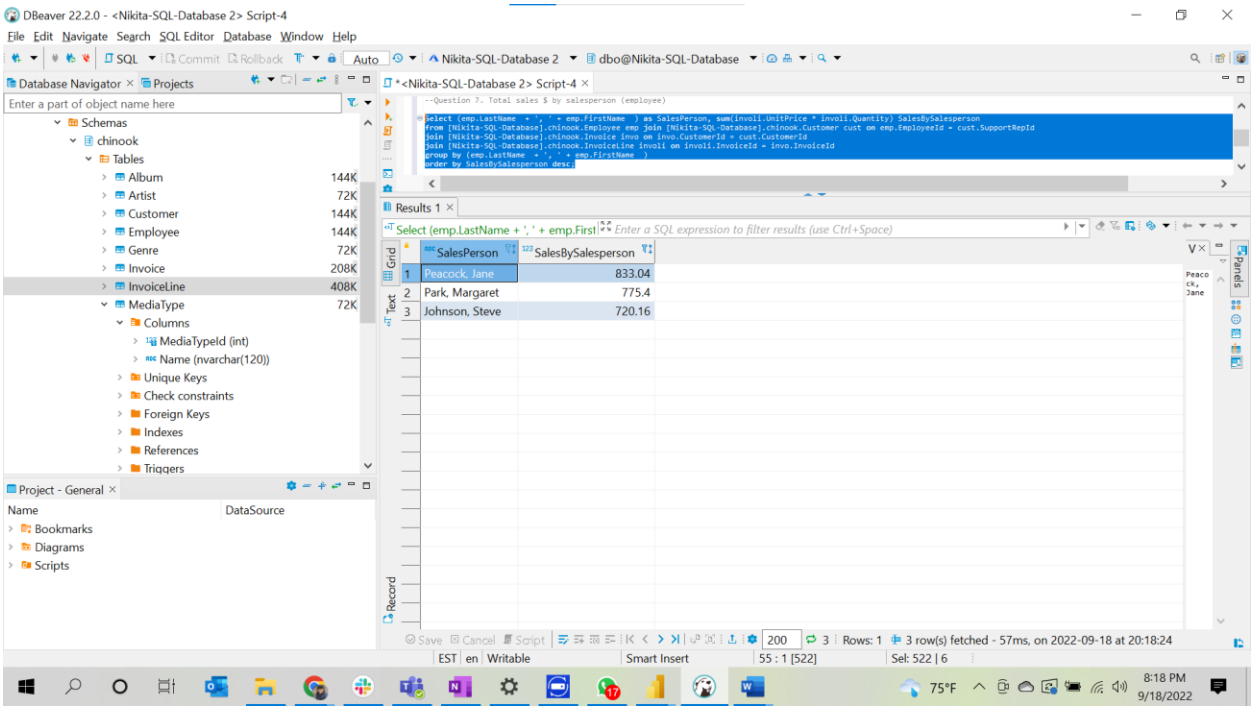

```
--Question 5. Total sales $ by artist - ranked (or at least sorted largest to smallest)

Select art.Name as ArtistName, sum(involi.UnitPrice * involi.Quantity) as SalesByArtist
from [Nikita-SQL-Database].chinook.Artist art join [Nikita-SQL-Database].chinook.Album ab on art.ArtistId
= ab.ArtistId
join [Nikita-SQL-Database].chinook.Track tr on tr.AlbumId  = ab.AlbumId
join [Nikita-SQL-Database].chinook.InvoiceLine involi on involi.TrackId  = tr.TrackId
group by art.Name
order by SalesByArtist desc;
```



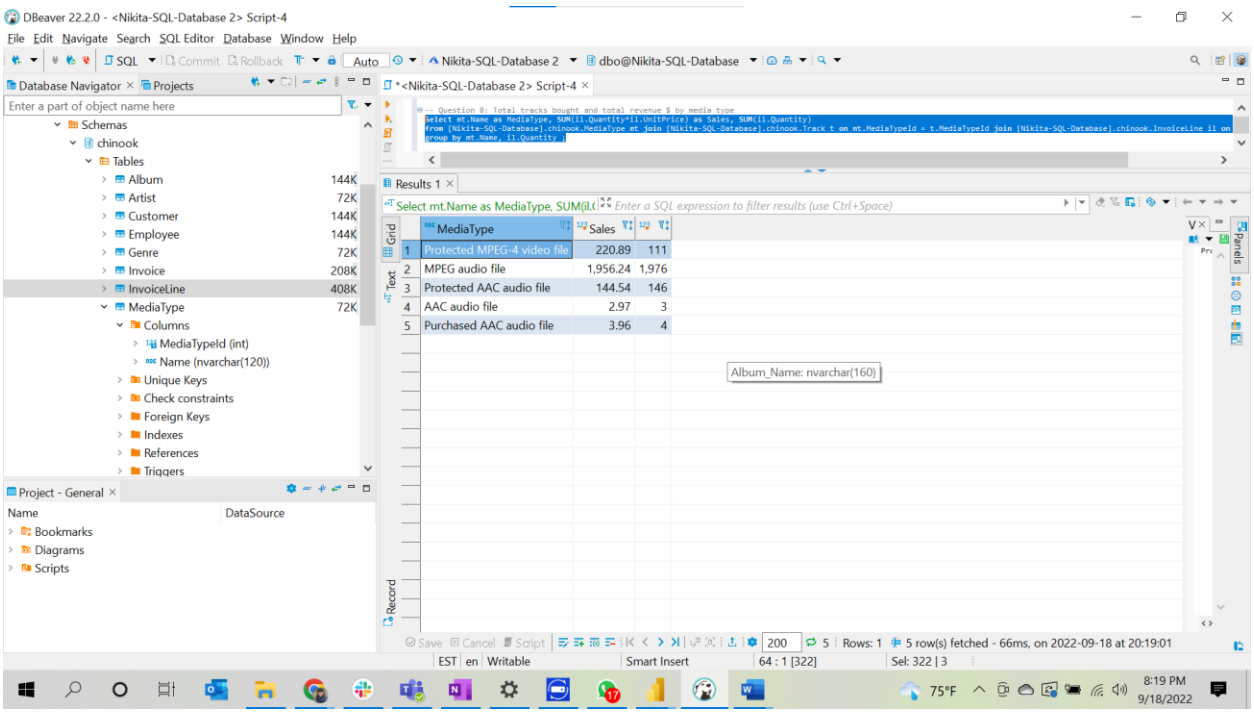
```
--Question 6. Total sales $ by albums

Select ab.Title as Album_Name,sum(involi.UnitPrice * involi.Quantity) SalesByAlbum
from [Nikita-SQL-Database].chinook.Album  ab join chinook.Track tr on ab.AlbumId = tr.AlbumId
join [Nikita-SQL-Database].chinook.InvoiceLine involi on involi.TrackId = tr.TrackId
group by ab.Title
order by SalesByAlbum desc;
```



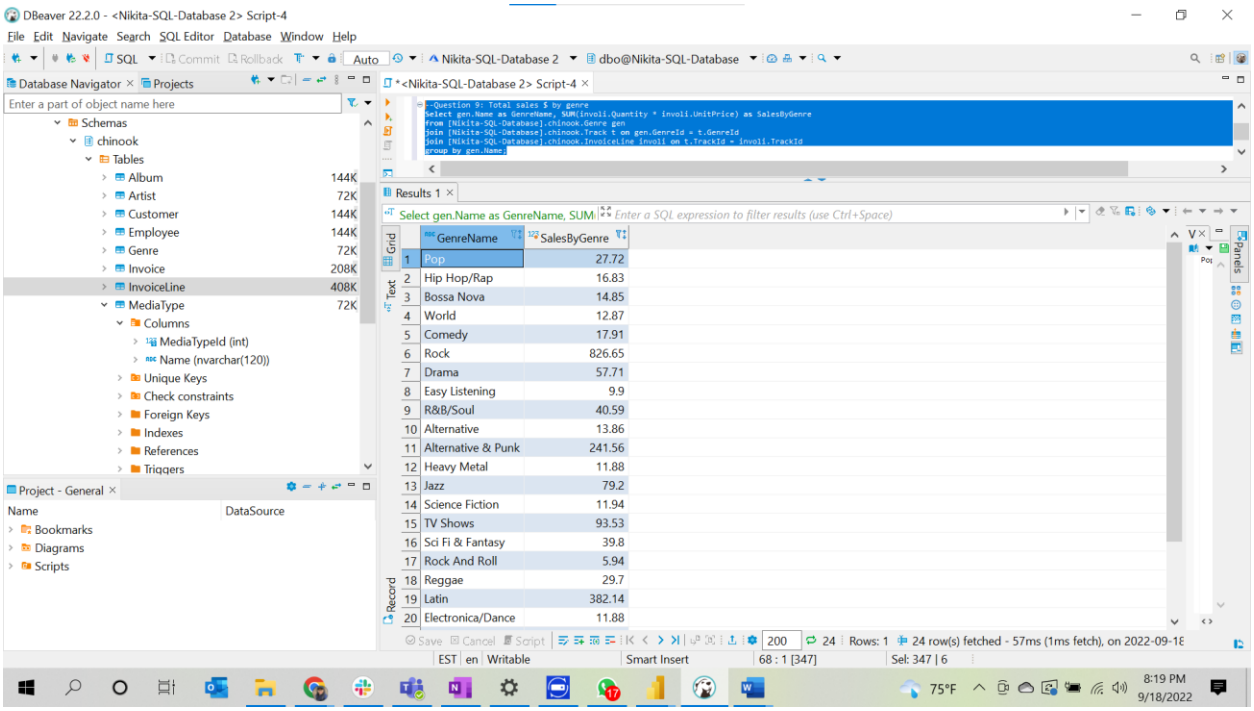

```
--Question 7. Total sales $ by salesperson (employee)

Select (emp.LastName  + ', ' + emp.FirstName  ) as SalesPerson, sum(involi.UnitPrice * involi.Quantity)
SalesBySalesperson
from [Nikita-SQL-Database].chinook.Employee emp join [Nikita-SQL-Database].chinook.Customer cust on
emp.EmployeeId = cust.SupportRepId
join [Nikita-SQL-Database].chinook.Invoice invo on invo.CustomerId = cust.CustomerId
join [Nikita-SQL-Database].chinook.InvoiceLine involi on involi.InvoiceId = invo.InvoiceId
group by (emp.LastName  + ', ' + emp.FirstName  )
order by SalesBySalesperson desc;
```



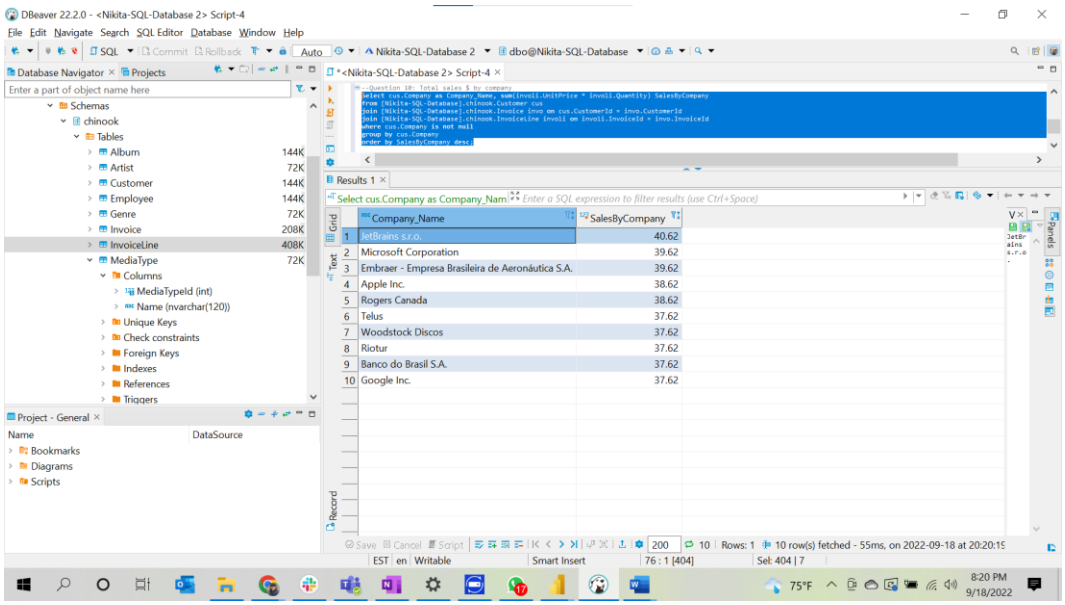

```
-- Question 8: Total tracks bought and total revenue $ by media type
Select mt.Name as MediaType, SUM(il.Quantity*il.UnitPrice) as Sales, SUM(il.Quantity)
from [Nikita-SQL-Database].chinook.MediaType mt join [Nikita-SQL-Database].chinook.Track t on
mt.MediaTypeId = t.MediaTypeId join [Nikita-SQL-Database].chinook.InvoiceLine il on t.TrackId = il.TrackId
group by mt.Name, il.Quantity ;
```



```sql
--Question 9: Total sales $ by genre
Select gen.Name as GenreName, SUM(involi.Quantity * involi.UnitPrice) as SalesByGenre
from [Nikita-SQL-Database].chinook.Genre gen
join [Nikita-SQL-Database].chinook.Track t on gen.GenreId = t.GenreId
join [Nikita-SQL-Database].chinook.InvoiceLine involi on t.TrackId = involi.TrackId
group by gen.Name;
```

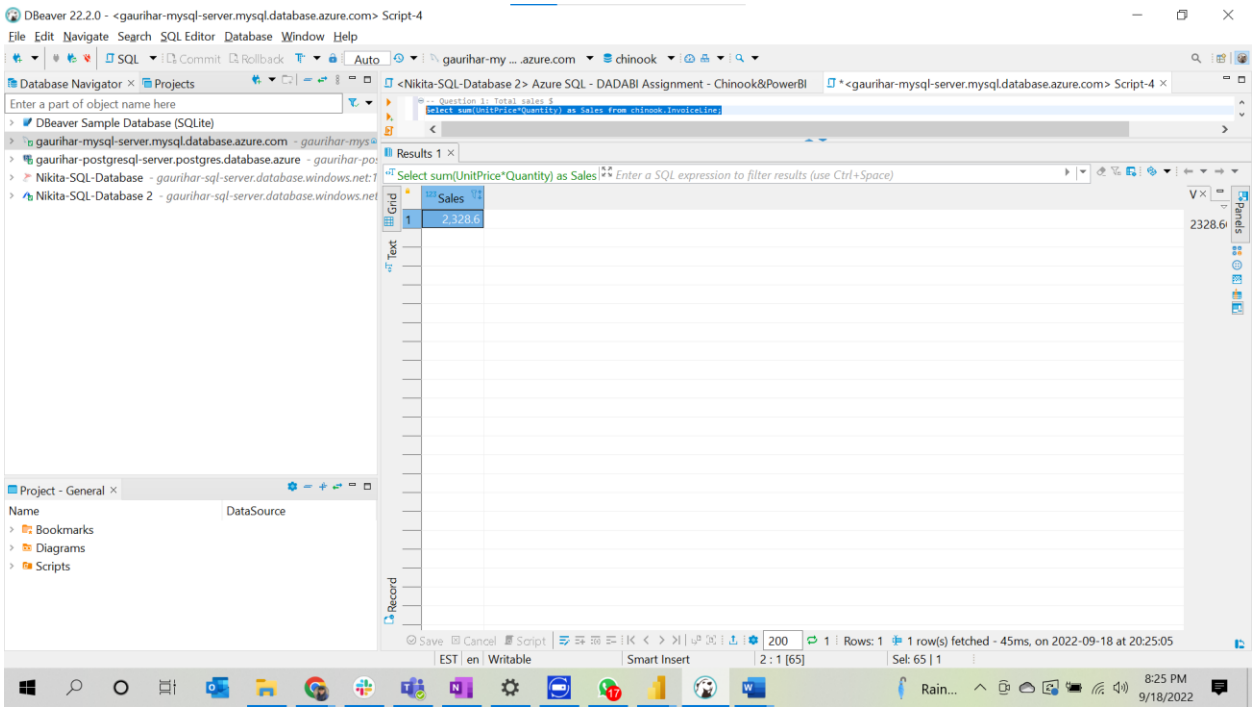



```sql
--Question 10: Total sales $ by company
Select cus.Company as Company_Name, sum(involi.UnitPrice * involi.Quantity) SalesByCompany
from [Nikita-SQL-Database].chinook.Customer cus
join [Nikita-SQL-Database].chinook.Invoice invo on cus.CustomerId = invo.CustomerId
join [Nikita-SQL-Database].chinook.InvoiceLine involi on involi.InvoiceId = invo.InvoiceId
where cus.Company is not null
group by cus.Company
order by SalesByCompany desc;
```

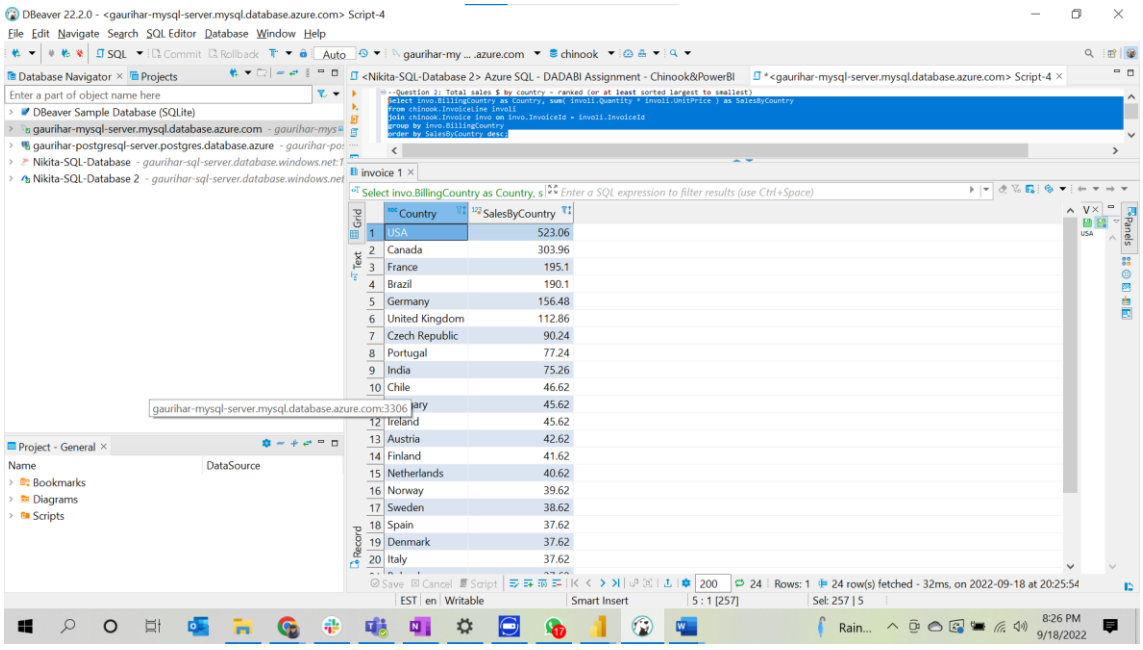


2. MySQL Database Server

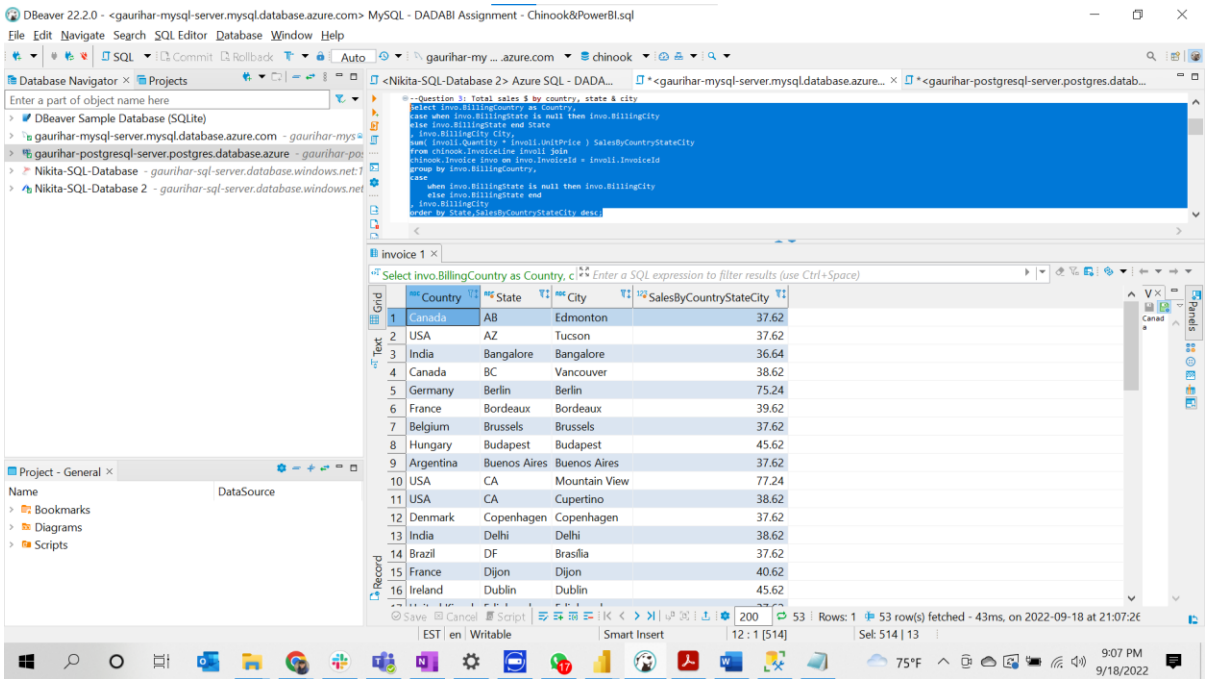
```
-- Question 1: Total sales $

Select sum(UnitPrice*Quantity) as Sales from chinook.InvoiceLine;
```



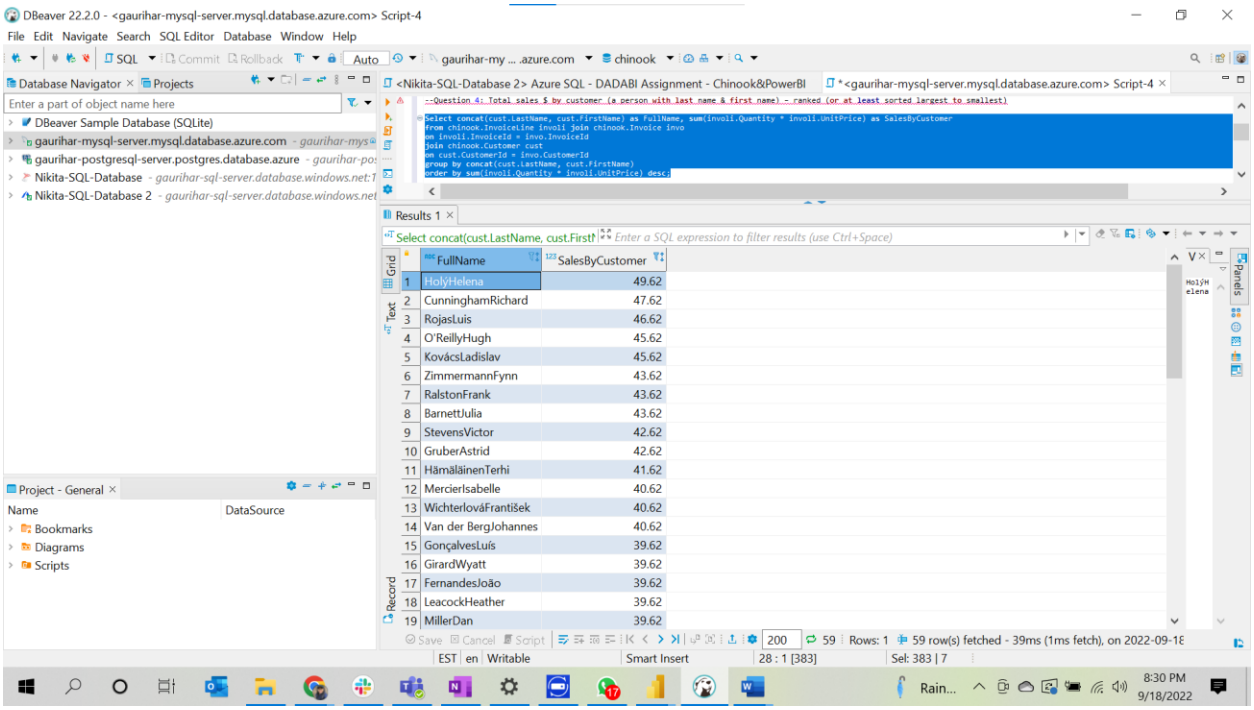

```
--Question 2: Total sales $ by country - ranked (or at least sorted largest to smallest)

Select invo.BillingCountry as Country, sum( involi.Quantity * involi.UnitPrice ) as SalesByCountry
from chinook.InvoiceLine involi
join chinook.Invoice invo on invo.InvoiceId = involi.InvoiceId
group by invo.BillingCountry
order by SalesByCountry desc;
```



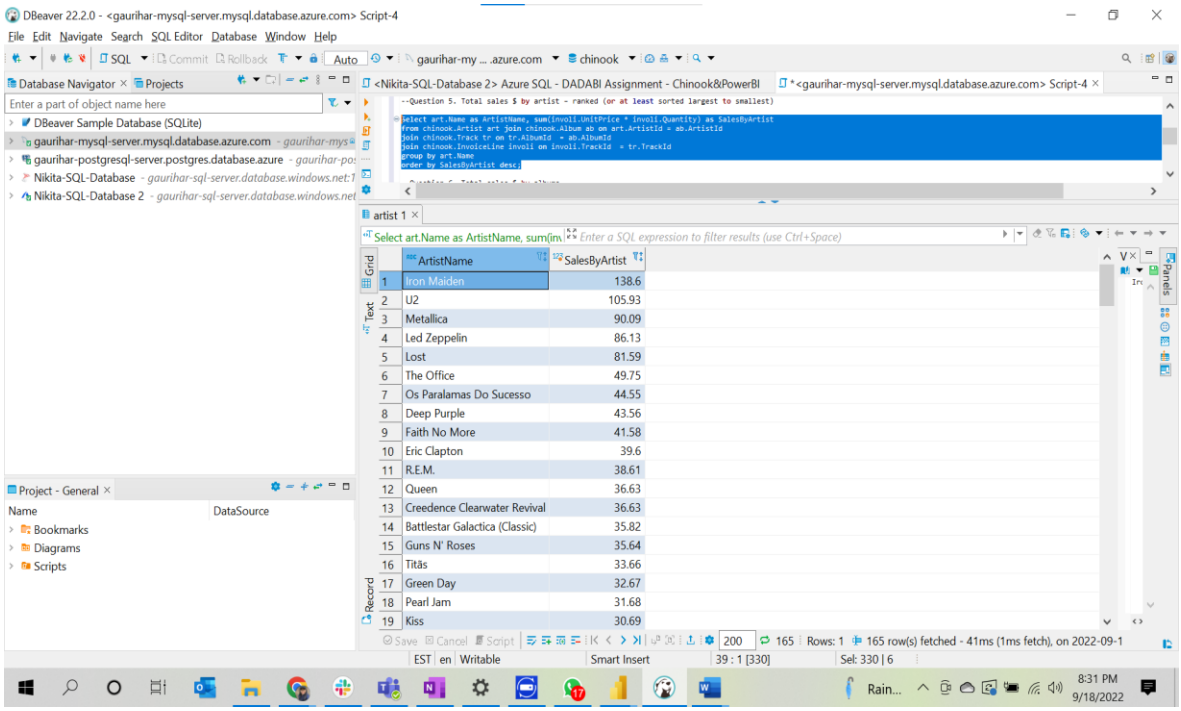
```sql
--Question 3: Total sales $ by country, state & city

Select invo.BillingCountry as Country,
case when invo.BillingState is null then invo.BillingCity
else invo.BillingState end State
, invo.BillingCity City,
sum( involi.Quantity * involi.UnitPrice ) SalesByCountryStateCity
from chinook.InvoiceLine involi join
chinook.Invoice invo on invo.InvoiceId = involi.InvoiceId
group by invo.BillingCountry,
case
    when invo.BillingState is null then invo.BillingCity
    else invo.BillingState end
, invo.BillingCity
order by State,SalesByCountryStateCity desc;
```



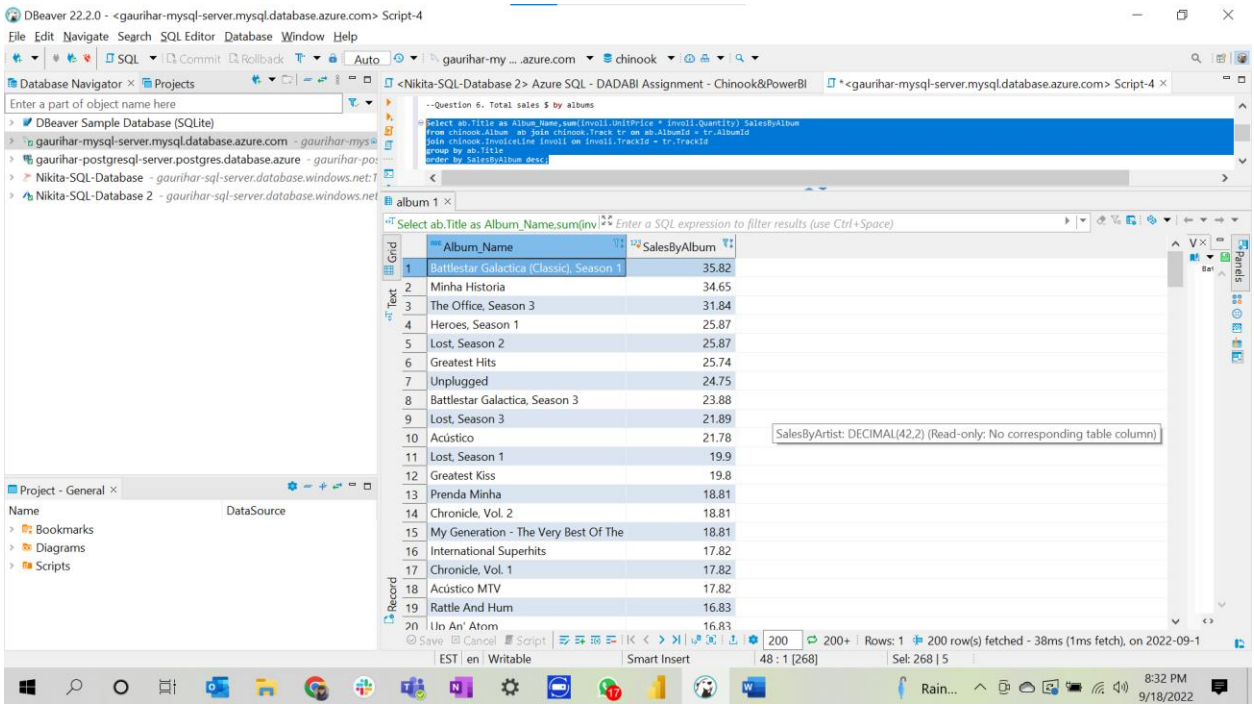

```sql
--Question 4: Total sales $ by customer (a person with last name & first name) - ranked (or at least sorted largest to smallest)

Select concat(cust.LastName, cust.FirstName) as FullName, sum(involi.Quantity * involi.UnitPrice) as SalesByCustomer
from chinook.InvoiceLine involi join chinook.Invoice invo
on involi.InvoiceId = invo.InvoiceId
join chinook.Customer cust
on cust.CustomerId = invo.CustomerId
group by concat(cust.LastName, cust.FirstName)
order by sum(involi.Quantity * involi.UnitPrice) desc;
```



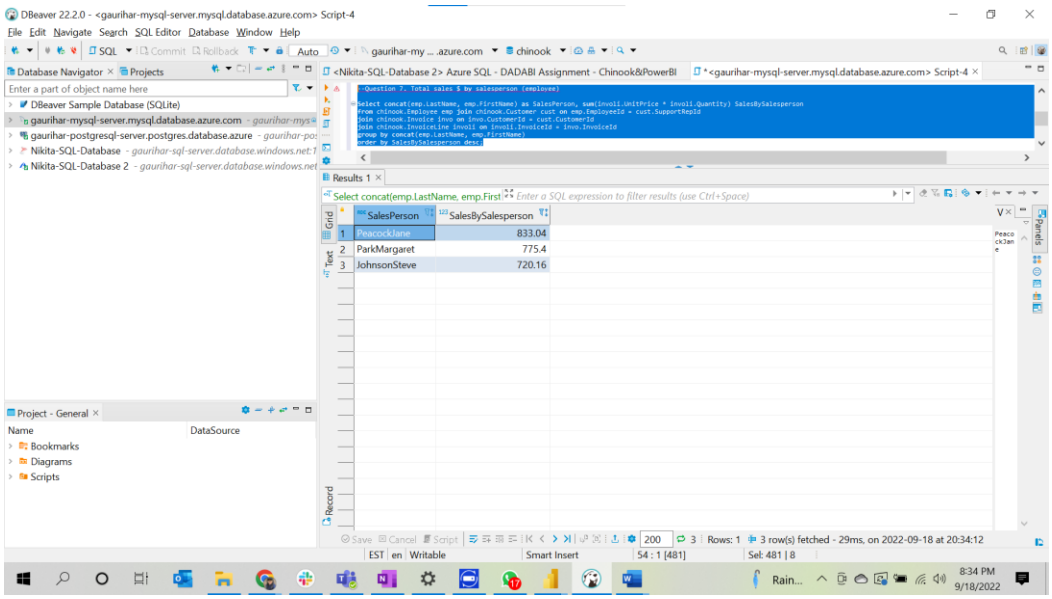

```
--Question 5. Total sales $ by artist - ranked (or at least sorted largest to smallest)

Select art.Name as ArtistName, sum(involi.UnitPrice * involi.Quantity) as SalesByArtist
from chinook.Artist art join chinook.Album ab on art.ArtistId = ab.ArtistId
join chinook.Track tr on tr.AlbumId  = ab.AlbumId
join chinook.InvoiceLine involi on involi.TrackId  = tr.TrackId
group by art.Name
order by SalesByArtist desc;
```



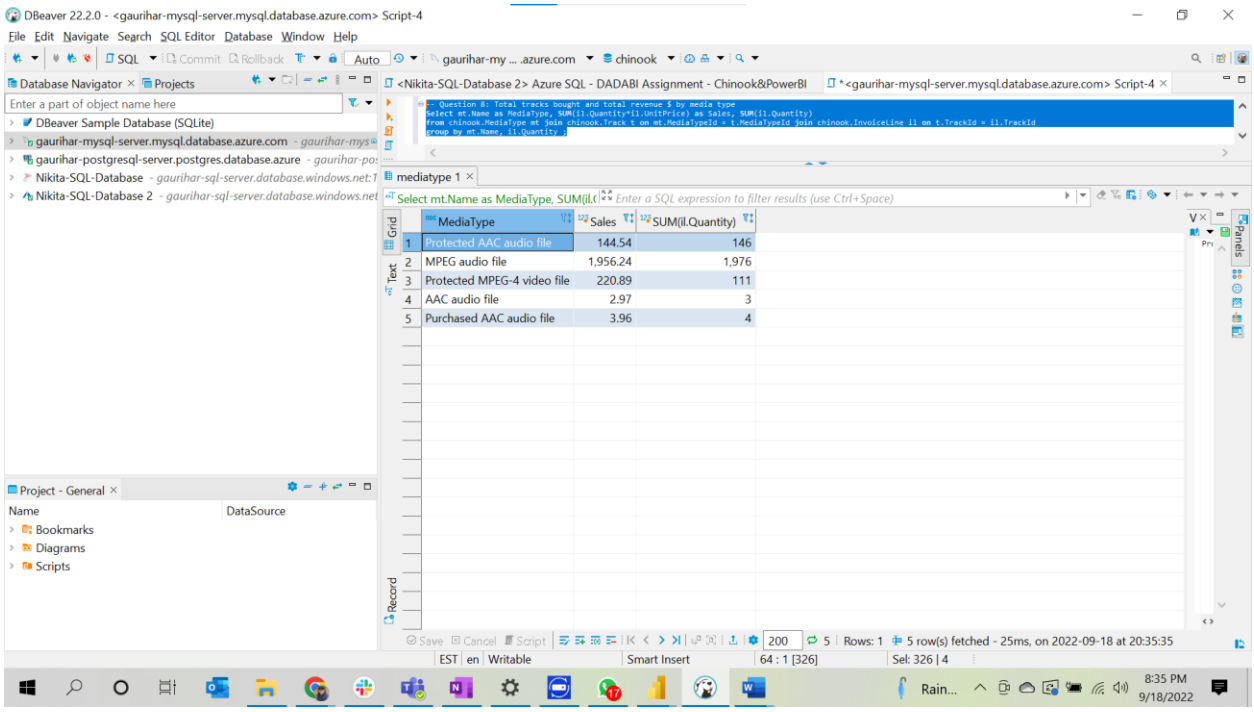
```
--Question 6. Total sales $ by albums

Select ab.Title as Album_Name,sum(involi.UnitPrice * involi.Quantity) SalesByAlbum
from chinook.Album  ab join chinook.Track tr on ab.AlbumId = tr.AlbumId
join chinook.InvoiceLine involi on involi.TrackId = tr.TrackId
group by ab.Title
order by SalesByAlbum desc;
```



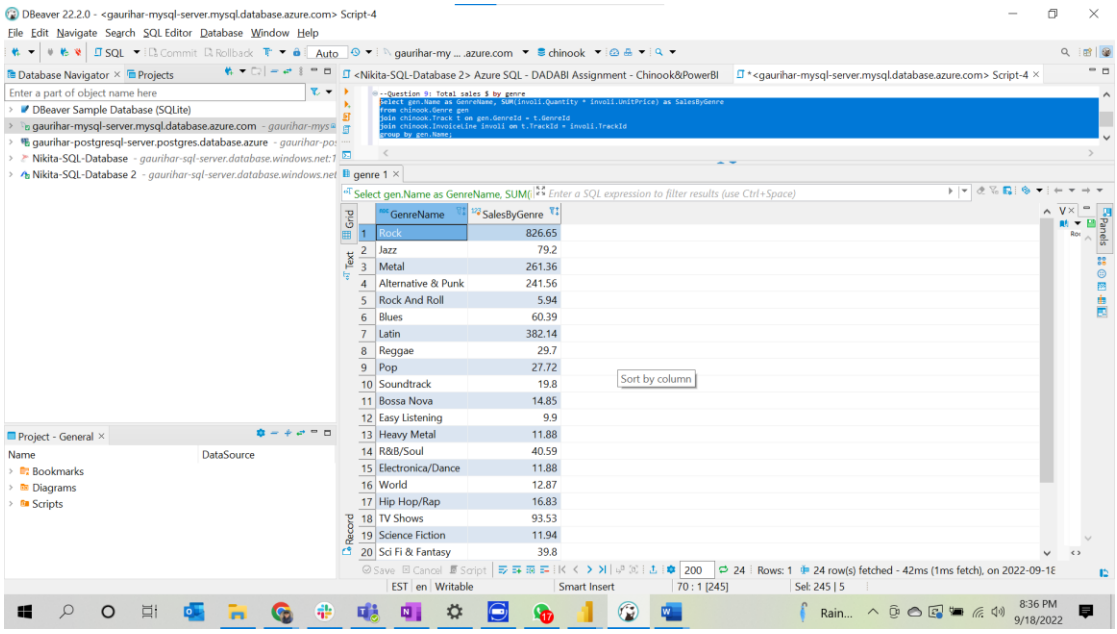

```
--Question 7. Total sales $ by salesperson (employee)

Select concat(emp.LastName, emp.FirstName) as SalesPerson, sum(involi.UnitPrice * involi.Quantity) SalesBySalesperson
from chinook.Employee emp join chinook.Customer cust on emp.EmployeeId = cust.SupportRepId
join chinook.Invoice invo on invo.CustomerId = cust.CustomerId
join chinook.InvoiceLine involi on involi.InvoiceId = invo.InvoiceId
group by concat(emp.LastName, emp.FirstName)
order by SalesBySalesperson desc;
```



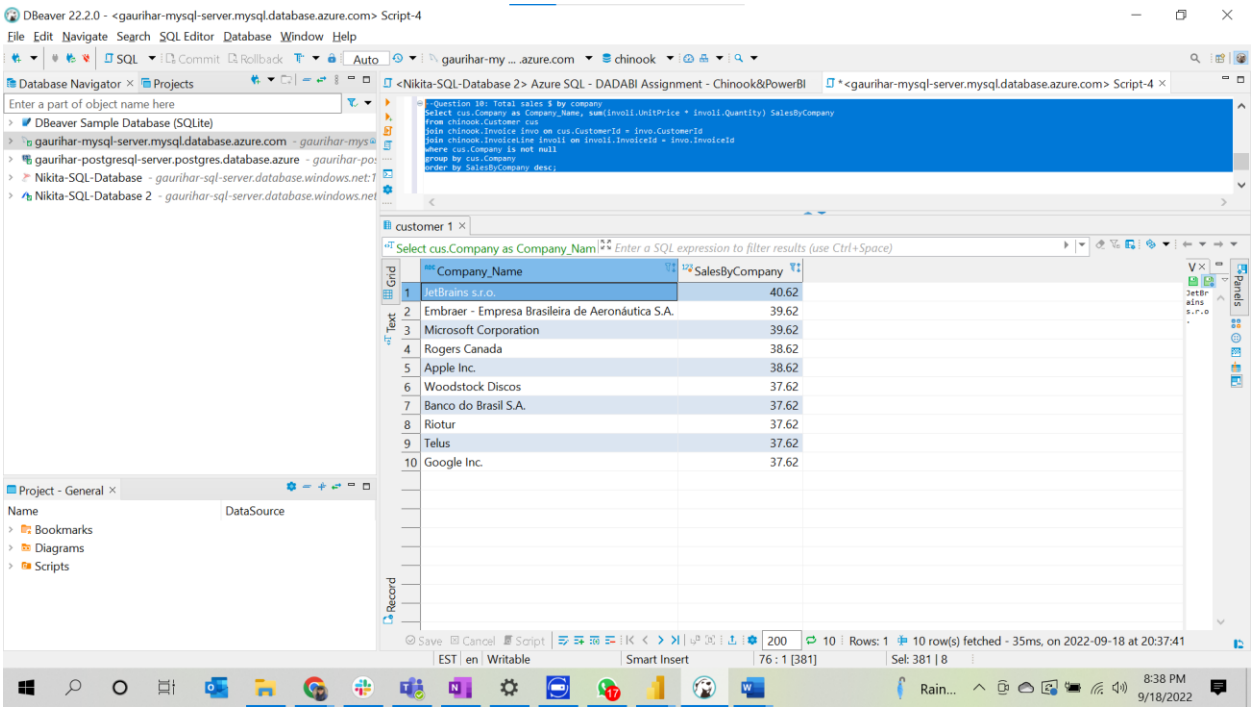
```sql
-- Question 8: Total tracks bought and total revenue $ by media type

Select mt.Name as MediaType, SUM(il.Quantity*il.UnitPrice) as Sales, SUM(il.Quantity)
from chinook.MediaType mt join chinook.Track t on mt.MediaTypeId = t.MediaTypeId join chinook.InvoiceLine il on t.TrackId = il.TrackId
group by mt.Name, il.Quantity ;
```




```sql
--Question 9: Total sales $ by genre

Select gen.Name as GenreName, SUM(involi.Quantity * involi.UnitPrice) as SalesByGenre
from chinook.Genre gen
join chinook.Track t on gen.GenreId = t.GenreId
join chinook.InvoiceLine involi on t.TrackId = involi.TrackId
group by gen.Name;
```

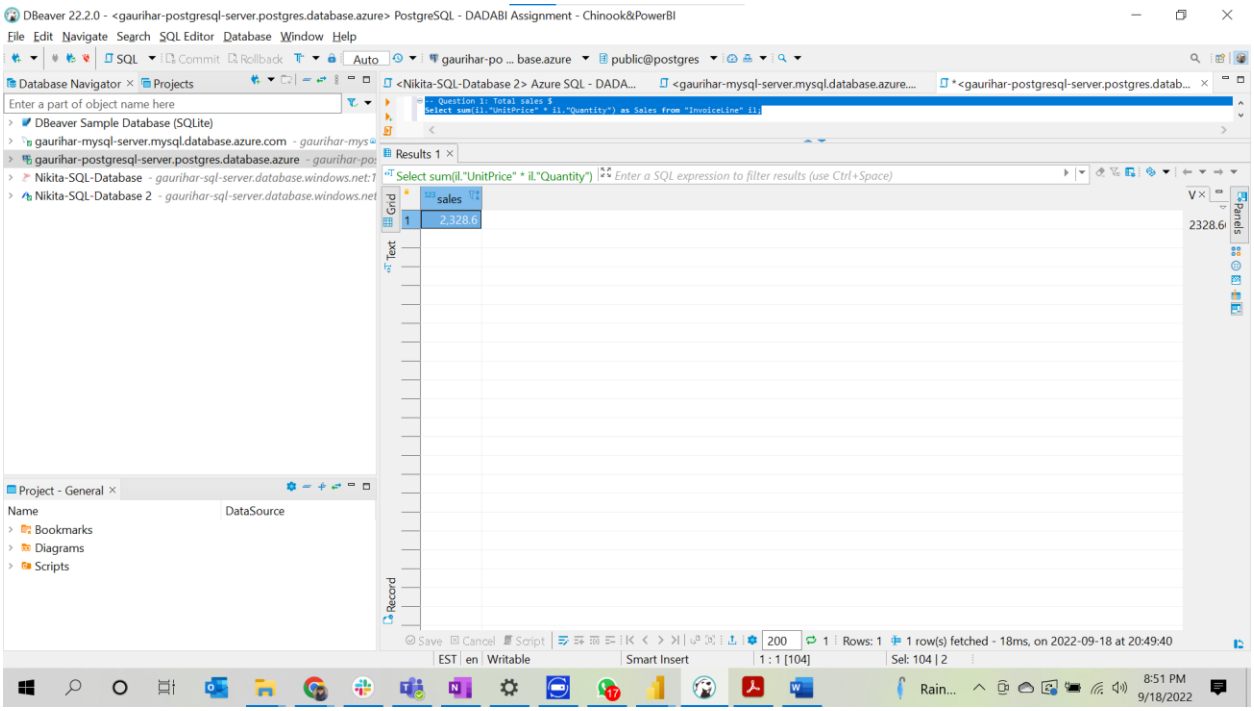


```sql
--Question 10: Total sales $ by company

Select cus.Company as Company_Name, sum(involi.UnitPrice * involi.Quantity) SalesByCompany
from chinook.Customer cus
join chinook.Invoice invo on cus.CustomerId = invo.CustomerId
join chinook.InvoiceLine involi on involi.InvoiceId = invo.InvoiceId
where cus.Company is not null
group by cus.Company
order by SalesByCompany desc;
```

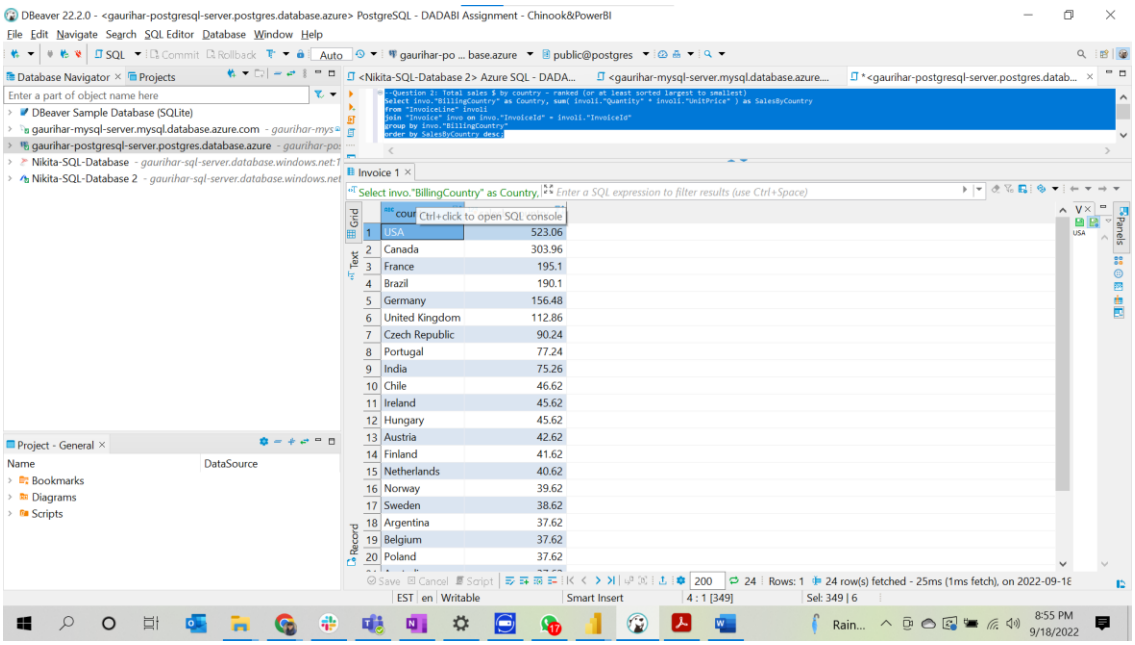


3. PostGre SQL Server

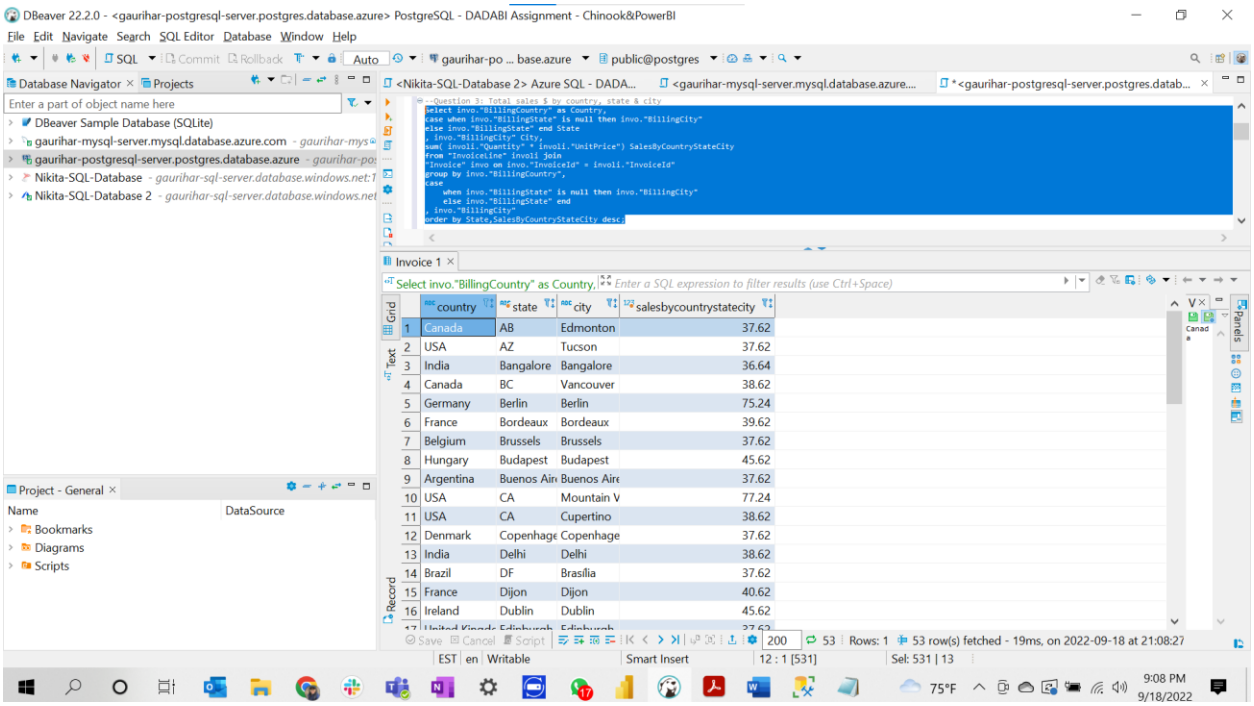
```
-- Question 1: Total sales $

Select sum(il."UnitPrice" * il."Quantity") as Sales from "InvoiceLine" il;
```



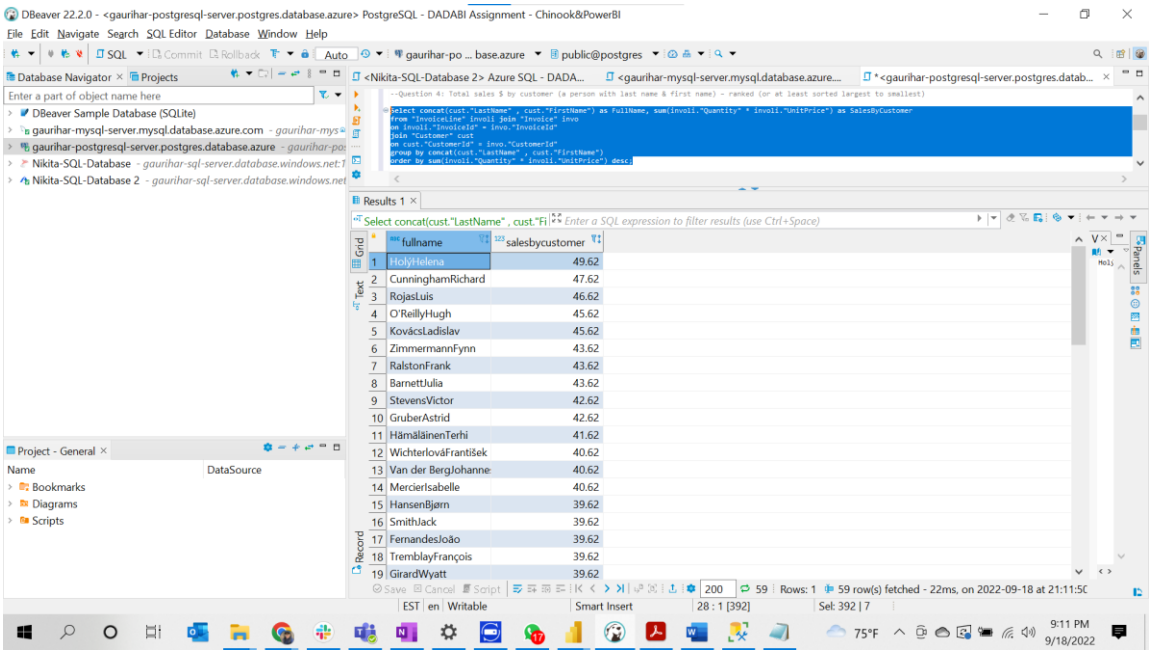

```
--Question 2: Total sales $ by country - ranked (or at least sorted largest to smallest)
Select invo."BillingCountry" as Country, sum( involi."Quantity" * involi."UnitPrice" ) as SalesByCountry
from "InvoiceLine" involi
join "Invoice" invo on invo."InvoiceId" = involi."InvoiceId"
group by invo."BillingCountry"
order by SalesByCountry desc;
```



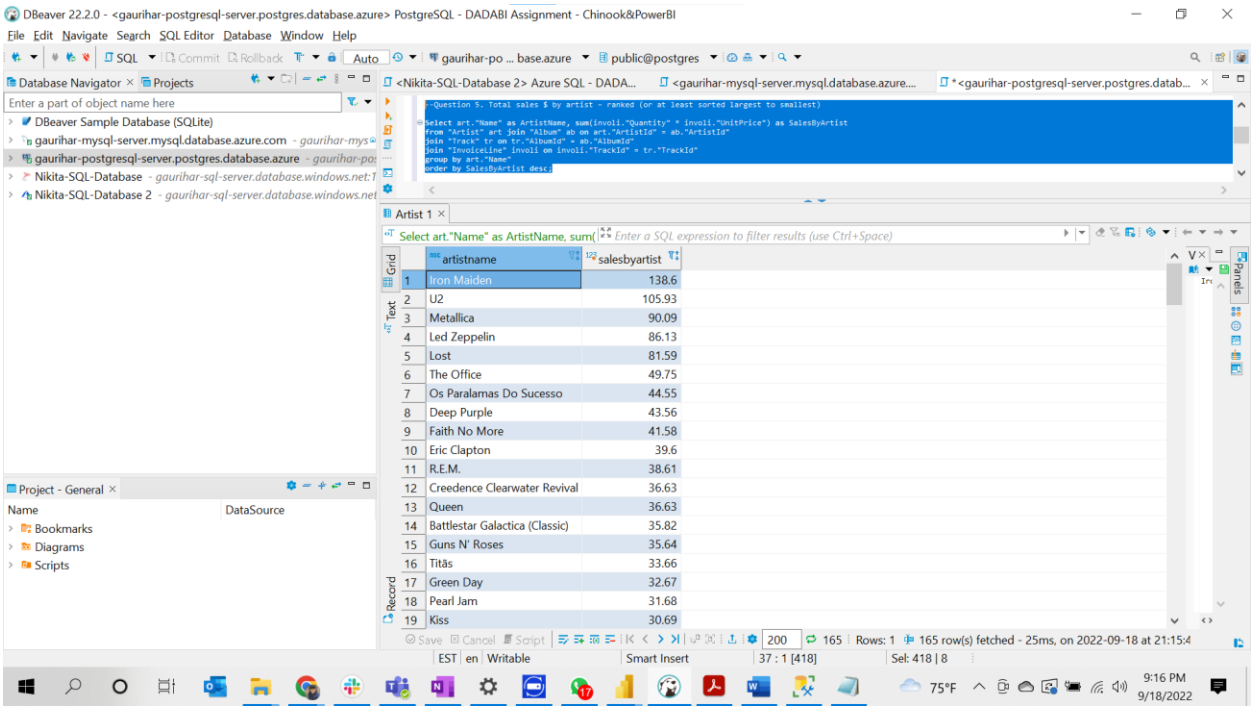
```sql
--Question 3: Total sales $ by country, state & city
Select invo."BillingCountry" as Country,
case when invo."BillingState" is null then invo."BillingCity"
else invo."BillingState" end State
, invo."BillingCity" City,
sum( involi."Quantity" * involi."UnitPrice") SalesByCountryStateCity
from "InvoiceLine" involi join
"Invoice" invo on invo."InvoiceId" = involi."InvoiceId"
group by invo."BillingCountry",
case
    when invo."BillingState" is null then invo."BillingCity"
    else invo."BillingState" end
, invo."BillingCity"
order by State,SalesByCountryStateCity desc;
```



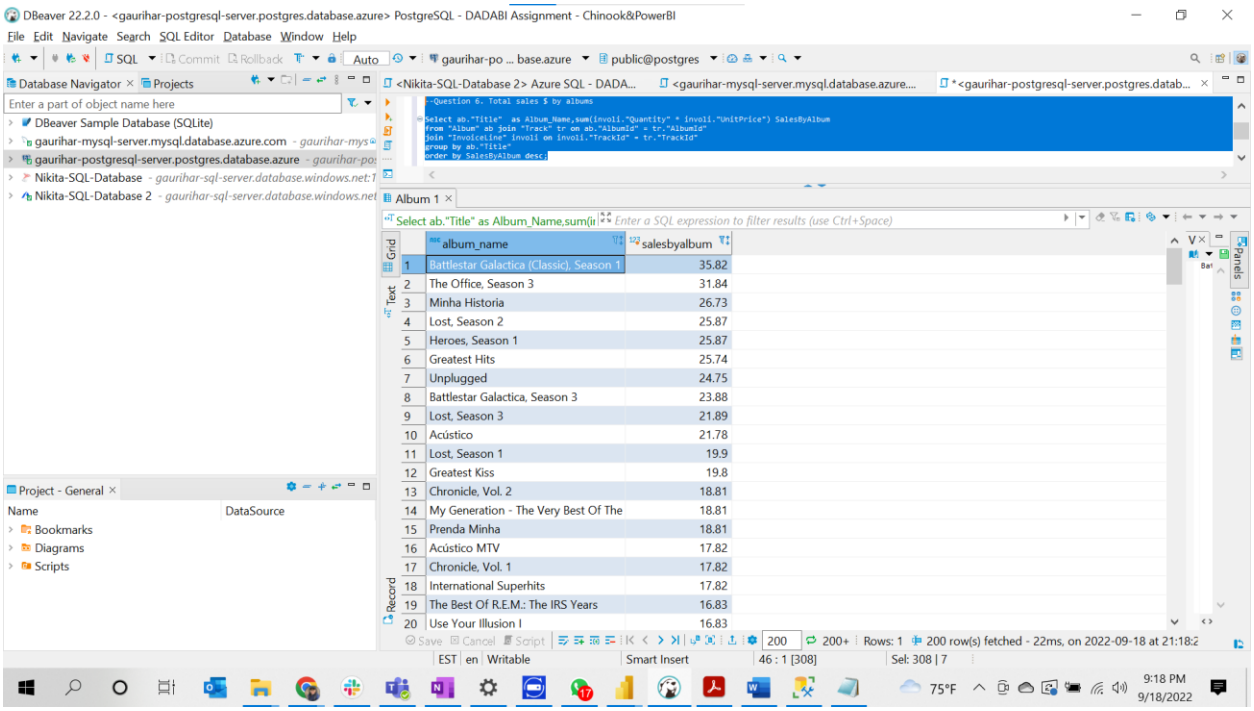

```sql
--Question 4: Total sales $ by customer (a person with last name & first name) - ranked (or at least sorted largest to smallest)


Select concat(cust."LastName" , cust."FirstName") as FullName, sum(involi."Quantity" * involi."UnitPrice") as SalesByCustomer
from "InvoiceLine" involi join "Invoice" invo
on involi."InvoiceId" = invo."InvoiceId"
join "Customer" cust
on cust."CustomerId" = invo."CustomerId"
group by concat(cust."LastName" , cust."FirstName")
order by sum(involi."Quantity" * involi."UnitPrice") desc;
```



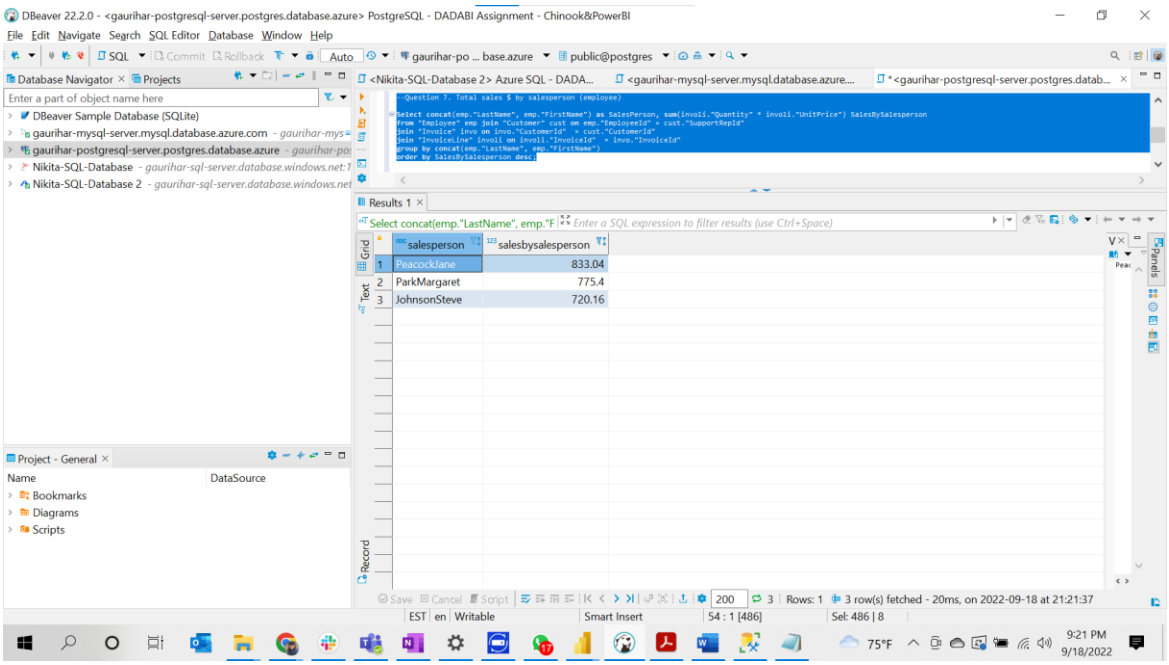

```
--Question 5. Total sales $ by artist - ranked (or at least sorted largest to smallest)

Select art."Name" as ArtistName, sum(involi."Quantity" * involi."UnitPrice") as SalesByArtist
from "Artist" art join "Album" ab on art."ArtistId" = ab."ArtistId"
join "Track" tr on tr."AlbumId" = ab."AlbumId"
join "InvoiceLine" involi on involi."TrackId" = tr."TrackId"
group by art."Name"
order by SalesByArtist desc;
```



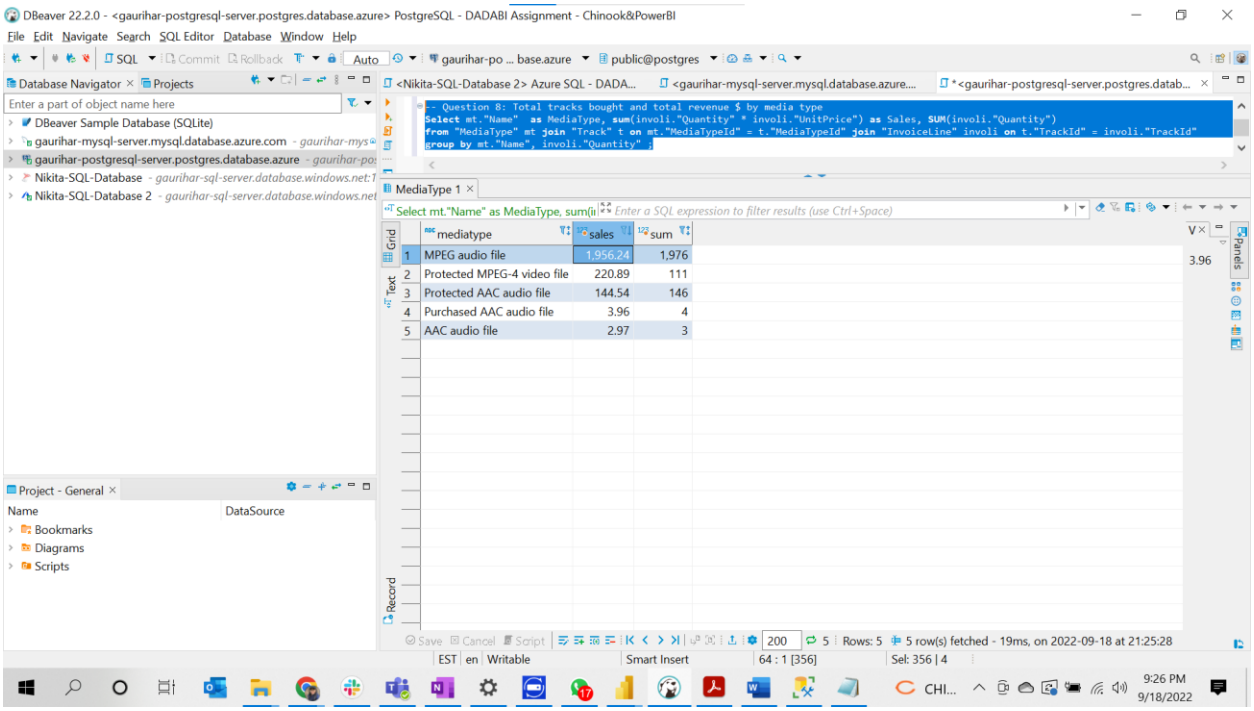
```
--Question 6. Total sales $ by albums

Select ab."Title"  as Album_Name,sum(involi."Quantity" * involi."UnitPrice") SalesByAlbum
from "Album" ab join "Track" tr on ab."AlbumId" = tr."AlbumId"
join "InvoiceLine" involi on involi."TrackId" = tr."TrackId"
group by ab."Title"
order by SalesByAlbum desc;
```



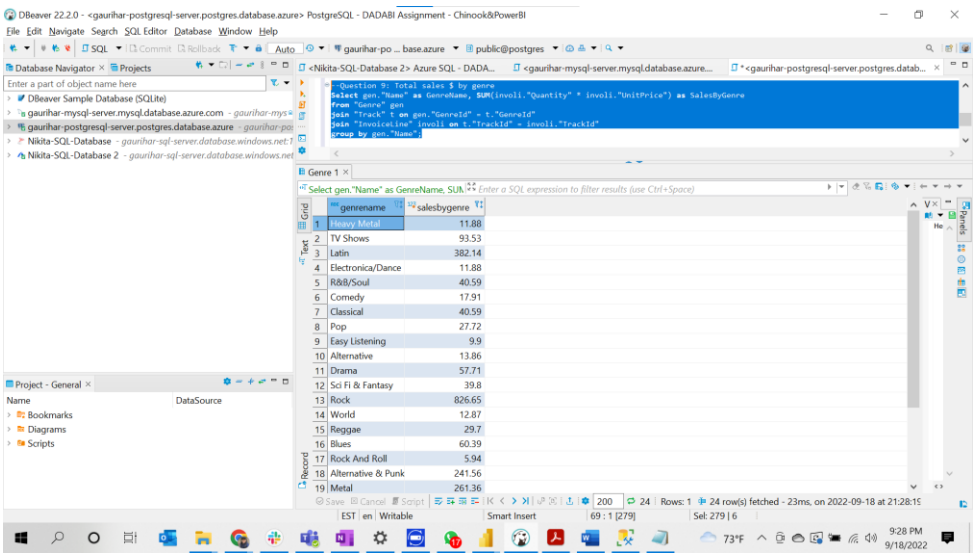

```
--Question 7. Total sales $ by salesperson (employee)

Select concat(emp."LastName", emp."FirstName") as SalesPerson, sum(involi."Quantity" * involi."UnitPrice") SalesBySalesperson
from "Employee" emp join "Customer" cust on emp."EmployeeId" = cust."SupportRepId"
join "Invoice" invo on invo."CustomerId"  = cust."CustomerId"
join "InvoiceLine" involi on involi."InvoiceId"  = invo."InvoiceId"
group by concat(emp."LastName", emp."FirstName")
order by SalesBySalesperson desc;
```



```sql
-- Question 8: Total tracks bought and total revenue $ by media type
Select mt."Name"  as MediaType, sum(involi."Quantity" * involi."UnitPrice") as Sales,
SUM(involi."Quantity")
from "MediaType" mt join "Track" t on mt."MediaTypeId" = t."MediaTypeId" join "InvoiceLine" involi on
t."TrackId" = involi."TrackId"
group by mt."Name", involi."Quantity" ;
```




```sql
--Question 9: Total sales $ by genre

Select gen."Name" as GenreName, SUM(involi."Quantity" * involi."UnitPrice") as SalesByGenre
from "Genre" gen
join "Track" t on gen."GenreId" = t."GenreId"
join "InvoiceLine" involi on t."TrackId" = involi."TrackId"
group by gen."Name";
```



```sql
--Question 10: Total sales $ by company
Select cus."Company" as Company_Name, sum(involi."Quantity" * involi."UnitPrice") SalesByCompany
from "Customer" cus
join "Invoice" invo on cus."CustomerId" = invo."CustomerId"
join "InvoiceLine" involi on involi."InvoiceId"  = invo."InvoiceId"
where cus."Company" is not null
group by cus."Company"
order by SalesByCompany desc;
```