

## a) Total sales \$

- Using SalesOrderHeader (ties to customer)

```
Select sum(SubTotal) as TotalSales_SalesOrderHeader
From [SalesLT].[SalesOrderHeader]
```

The screenshot shows the Microsoft Azure portal interface for the DADABI database. The left sidebar contains navigation options like Overview, Activity log, Tags, and Settings. The main area displays the 'Query editor (preview)' for the 'DADABI (dadabi-sqlserver/DADABI)' database. The query editor shows a SQL query: `Select sum(SubTotal) as TotalSales_SalesOrderHeader From [SalesLT].[SalesOrderHeader]`. The results pane shows a single row with the value 865433.1171. The status bar at the bottom indicates 'Query succeeded | 0s'.

Total rows: 1

- Using SalesOrderDetail (ties to product)

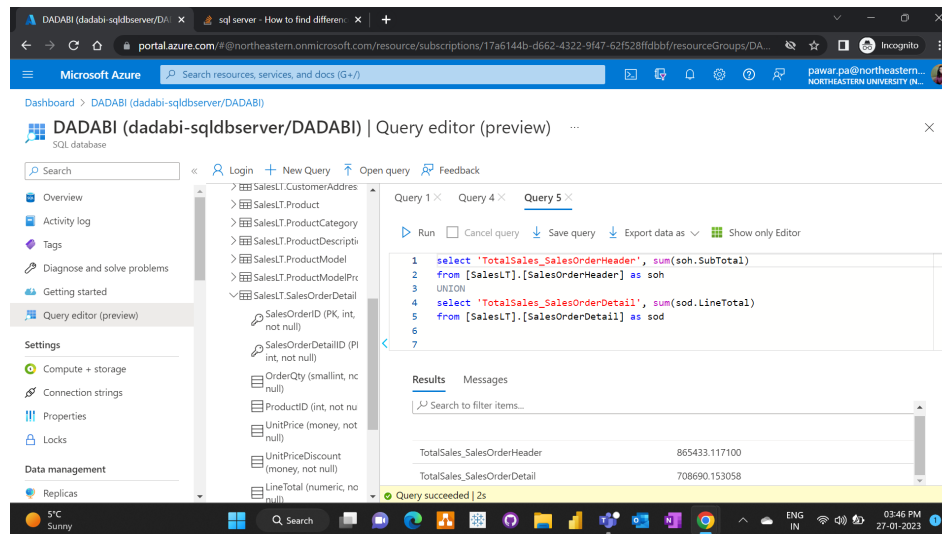
```
Select sum(LineTotal) as TotalSales_SalesOrderDetail
From [SalesLT].[SalesOrderDetail]
```

The screenshot shows the Microsoft Azure portal interface for the DADABI database. The left sidebar contains navigation options like Overview, Activity log, Tags, and Settings. The main area displays the 'Query editor (preview)' for the 'DADABI (dadabi-sqlserver/DADABI)' database. The query editor shows a SQL query: `Select sum(LineTotal) as TotalSales_SalesOrderDetail From [SalesLT].[SalesOrderDetail]`. The results pane shows a single row with the value 708690.153058. The status bar at the bottom indicates 'Query succeeded | 6s'.

Total rows: 1

- Comparison querying with SalesOrderHeader vs SalesOrderDetail

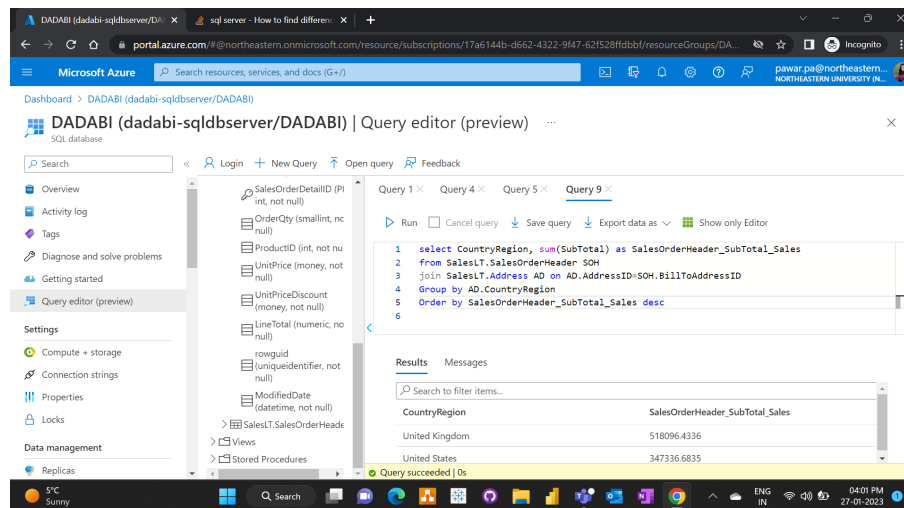
```
select 'TotalSales_SalesOrderHeader', sum(soh.SubTotal)
from [SalesLT].[SalesOrderHeader] as soh UNION
select 'TotalSales_SalesOrderDetail', sum(sod.LineTotal)
from [SalesLT].[SalesOrderDetail] as sod
```



Total rows: 2

- b) Total sales \$ by country – ranked/sorted (highest to lowest)

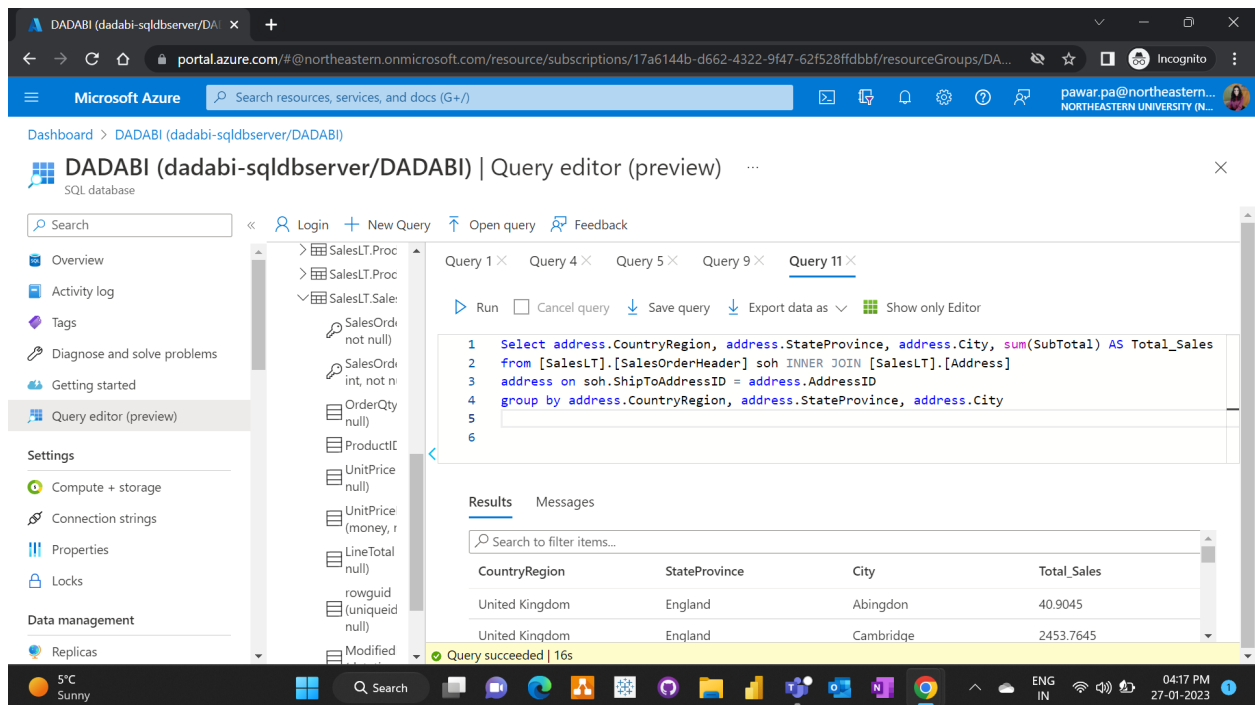
```
select CountryRegion, sum(SubTotal) as SalesOrderHeader_SubTotal_Sales
from SalesLT.SalesOrderHeader SOH
join SalesLT.Address AD on AD.AddressID=SOH.BillToAddressID
Group by AD.CountryRegion
Order by SalesOrderHeader_SubTotal_Sales desc
```



Total rows: 2

### c) Total sales \$ by country, state & city

```
Select address.CountryRegion, address.StateProvince, address.City,  
sum(SubTotal) AS Total_Sales  
from [SalesLT].[SalesOrderHeader] soh INNER JOIN [SalesLT].[Address]  
address on soh.ShipToAddressID = address.AddressID  
group by address.CountryRegion, address.StateProvince, address.City
```



The screenshot shows the DADABI query editor interface. The query editor displays the following SQL query:

```
1 Select address.CountryRegion, address.StateProvince, address.City, sum(SubTotal) AS Total_Sales  
2 from [SalesLT].[SalesOrderHeader] soh INNER JOIN [SalesLT].[Address]  
3 address on soh.ShipToAddressID = address.AddressID  
4 group by address.CountryRegion, address.StateProvince, address.City  
5  
6
```

The results section shows the following data:

| CountryRegion  | StateProvince | City      | Total_Sales |
|----------------|---------------|-----------|-------------|
| United Kingdom | England       | Abingdon  | 40.9045     |
| United Kingdom | England       | Cambridge | 2453.7645   |

The status bar indicates "Query succeeded | 16s".

Total rows: 29

### d) Total sales \$ by customer (person) – ranked/sorted (highest to lowest)

```
Select concat(CUS.LastName, ',', CUS.FirstName) as  
CustomerPerson, sum(SubTotal) as TotalSales  
from [SalesLT].[SalesOrderHeader] SOH inner join [SalesLT].[Customer] CUS  
ON SOH.CustomerID = CUS.CustomerID  
group by CUS.LastName, CUS.FirstName  
order by sum(SubTotal) desc
```

Microsoft Azure portal showing the DADABI (dadabi-sqlldbserver/DADABI) Query editor. The query editor displays a SQL query that concatenates customer names and sums subtotals. The results table shows two rows: Eminhizer, Terry with a total sales of 108561.8317, and Sunkammuralli, Krishna with a total sales of 98278.6910. The status bar indicates 'Query succeeded | 7s'.

Total rows: 32

e) Total sales \$ & by customer (company) – ranked/sorted (highest to lowest)

```
select CompanyName as CustomerCompany , sum(SubTotal) as Total_Sales
from [SalesLT].[SalesOrderHeader] SOH inner join [SalesLT].[Customer] c
on SOH.CustomerID = c.CustomerID
group by CompanyName
order by sum(SubTotal) desc
```

Microsoft Azure portal showing the DADABI Query editor. The editor displays a SQL query for Query 16, which selects CustomerCompany and Total\_Sales from SalesLT.SalesOrderHeader and SalesLT.Customer. The results table shows two rows: Action Bicycle Specialists with a total sales of 108561.8317, and Metropolitan Bicycle Supply with a total sales of 98278.6910. The status bar indicates 'Query succeeded | 4s'.

Total rows: 32

#### f) Sales \$ by product category hierarchy – Product & vGetAllCategories

```

Select PC.ParentProductCategoryName as 'ParentProductCategoryName',
PC.ProductCategoryName as
'ProductCategoryName', p.name as 'ProductName', sum(SOD.linetotal) as
'SOD_LineTotal_Sales'
from [SalesLT].[vgetallcategories] as PC
join [SalesLT].[Product] as P
on P.productcategoryid = PC.productcategoryid
join [SalesLT].[SalesOrderDetail] as SOD
on P.productid = SOD.productid
group by PC.ParentProductCategoryName, PC.ProductCategoryName, P.name

```

Microsoft Azure portal showing the DADABI (dadabi-sqlserver/DADABI) Query editor. The interface includes a sidebar with navigation options like Overview, Activity log, and Settings. The main area displays a SQL query in the editor and its results in a table. The query selects product category names, product names, and the sum of line totals from the SalesLT.Product and SalesLT.SalesOrderDetail tables. The results table shows two rows: one for 'Accessories' with 'Bike Racks' and a total of 2304.000000, and another for 'Accessories' with 'Bottles and Cages' and a total of 158.206952. A status bar at the bottom indicates 'Query succeeded | 8s'.

Total rows: 142

g) Sales \$ by product name – ranked/sorted (highest to lowest)

```

Select P.Name, sum(S.LineTotal) AS SOD_LineTotal
from SalesLT.Product P
inner join SalesLT.SalesOrderDetail S on P.ProductID=S.ProductID
group by P.Name
order by SUM(s.LineTotal) DESC

```

Microsoft Azure portal showing the DADABI (dadabi-sqlldbserver/DADABI) Query editor. The query editor displays a SQL query for Query 19, which selects product names and their total sales. The results table shows two rows: 'Touring-1000 Blue, 60' with a total of 37191.492000 and 'Mountain-200 Black, 42' with a total of 37178.838000. The status bar indicates the query succeeded in 8 seconds.

```

1 Select P.Name, sum(S.LineTotal) AS SOD_LineTotal
2 from SalesLT.Product P
3 inner join SalesLT.SalesOrderDetail S on P.ProductID=S.ProductID
4 group by P.Name
5 order by SUM(s.LineTotal) DESC
6

```

| Name                   | SOD_LineTotal |
|------------------------|---------------|
| Touring-1000 Blue, 60  | 37191.492000  |
| Mountain-200 Black, 42 | 37178.838000  |

Query succeeded | 8s

Total rows: 142

#### h) Sales \$ by Company (Reseller)

```

Select C.CompanyName, sum(LineTotal) as SOD_LineTotal
from SalesLT.SalesOrderHeader H
inner join SalesLT.Customer C on H.CustomerID= C.CustomerID
inner join SalesLT.SalesOrderDetail O on H.SalesOrderID = O.SalesOrderID
group by C.CompanyName
order by sum(LineTotal) desc

```

Microsoft Azure portal showing the DADABI Query editor. The query editor displays a SQL query for Query 20:

```

1 Select C.CompanyName, sum(LineTotal) as SOD_LineTotal
2 from SalesLT.SalesOrderHeader H
3 inner join SalesLT.Customer C on H.CustomerID= C.CustomerID
4 inner join SalesLT.SalesOrderDetail O on H.SalesOrderID = O.SalesOrderID
5 group by C.CompanyName
6 order by sum(LineTotal) desc

```

The results table shows the following data:

| CompanyName                 | SOD_LineTotal |
|-----------------------------|---------------|
| Action Bicycle Specialists  | 89869.276314  |
| Metropolitan Bicycle Supply | 79589.616024  |

The status bar indicates "Query succeeded | 5s".

Total rows: 32

#### i) Product Category Sales \$ by Company (Reseller)

```

select C.CompanyName as 'CompanyName', PC.Name as 'CategoryName',
sum(sod.LineTotal) as
'SOD_LineTotal_Sales'
from [SalesLT].[ProductCategory] as PC
join [SalesLT].[Product] as P
on P.productcategoryid = PC.productcategoryid
join [SalesLT].[SalesOrderDetail] as sod
on sod.productid = P.productid
join [SalesLT].[SalesOrderHeader] as soh
on soh.salesorderid = sod.salesorderid
join [SalesLT].[Customer] as C
on C.customerID = soh.customerID
group by C.CompanyName,pc.Name
order by sum(sod.LineTotal) desc

```



Microsoft Azure portal interface showing the DADABI (dadabi-sqldbserver/DADABI) Query editor (preview). The interface includes a sidebar with navigation options (Overview, Activity log, Tags, Diagnose and solve problems, Getting started, Query editor (preview), Settings, Compute + storage, Connection strings, Properties, Locks, Data management, Replicas) and a main area for writing and running SQL queries.

The query editor shows a SQL query being executed:

```
9 on soh.salesorderid = sod.salesorderid
10 join [SalesLT].[Customer] as C
11 on C.customerID = soh.customerID
12 group by C.CompanyName,pc.Name
13 order by sum(sod.LineTotal) desc
14
15
```

The results section displays the output of the query:

| CompanyName                | CategoryName  | SOD_LineTotal_Sales |
|----------------------------|---------------|---------------------|
| Action Bicycle Specialists | Touring Bikes | 76613.651796        |
| Bulk Discount Store        | Road Bikes    | 70597.284000        |

The status bar indicates "Query succeeded | 2s".

Total rows: 195