



## Building a Fact Table

In SQL, a fact table is a table that stores quantitative data or "facts" about a business process or event. It typically contains numerical values, often aggregated or summarized, along with foreign keys that establish relationships with dimension tables. These dimension tables provide context to the facts stored in the fact table.

Here's a brief overview of key characteristics of a fact table in SQL:

1. **Quantitative Data:** Fact tables store measurable data, such as sales revenue, quantities sold, or units produced.
2. **Foreign Keys:** Fact tables contain foreign keys that establish relationships with dimension tables. These keys are used to join the fact table with dimension tables to provide context and additional information about the facts.
3. **Aggregated Data:** Fact tables often store aggregated or summarized data at different levels of granularity. For example, daily sales totals, monthly revenue, or yearly production figures.
4. **Central Role:** In a star schema or snowflake schema, the fact table serves as the central table, with dimension tables branching off it. This arrangement facilitates efficient querying and analysis of data.
5. **Performance Optimization:** Fact tables are typically designed to optimize query performance for analytical workloads. This may involve indexing strategies, partitioning, or denormalization techniques to improve data retrieval speed.



## Use cases of Fact Table:

Fact tables in SQL are essential components of data warehousing systems and are commonly used in various industries and applications for analytical purposes. Here are some typical use cases for fact tables:

### 1. Retail Sales Analysis:

- Tracking sales transactions, including sales revenue, quantity sold, discounts applied, and product information.
- Analyzing sales performance across different regions, stores, products, and time periods.
- Identifying trends, seasonality, and patterns in customer purchasing behavior.

### 2. Financial Analysis:

- Recording financial transactions such as deposits, withdrawals, investments, and loan payments.
- Calculating key financial metrics like revenue, expenses, profit margins, and return on investment (ROI).
- Analyzing financial performance, forecasting future trends, and monitoring financial health.

### **3. Healthcare Analytics:**

- Storing patient medical records, procedures, diagnoses, treatments, and outcomes.
- Analyzing healthcare utilization, patient demographics, disease prevalence, and treatment efficacy.
- Supporting population health management, clinical research, and healthcare quality improvement initiatives.

### **4. Supply Chain Management:**

- Tracking inventory levels, shipments, deliveries, and warehouse operations.
- Monitoring supplier performance, lead times, order fulfillment rates, and inventory turnover.
- Optimizing supply chain processes, reducing costs, and improving efficiency and responsiveness.

### **5. Marketing Campaign Analysis:**

- Recording campaign data, including impressions, clicks, conversions, and campaign costs.
- Evaluating campaign effectiveness, return on investment (ROI), and customer acquisition metrics.
- Segmentation analysis to identify target audiences, personalize marketing efforts, and optimize campaign strategies.

### **6. Telecommunications Billing:**

- Capturing call detail records (CDRs), including call duration, origin, destination, and call type.
- Billing analysis, including usage charges, tariffs, discounts, and customer billing statements.
- Identifying usage patterns, peak hours, and subscriber behavior for capacity planning and service optimization.

### **7. Manufacturing Performance Analysis:**

- Monitoring production metrics such as throughput, cycle times, downtime, and defect rates.
- Analyzing equipment performance, maintenance schedules, and production efficiency.
- Identifying opportunities for process improvement, cost reduction, and quality enhancement.

### **To begin with the Lab:**

1. The prerequisite for this lab is you should have an Azure SQL Database server running with sample data loaded.
2. Now you have to use SQL Server Management Studio (SSMS).
3. Here you are going to login with your SQL Database server and with your Synapse Workspace.

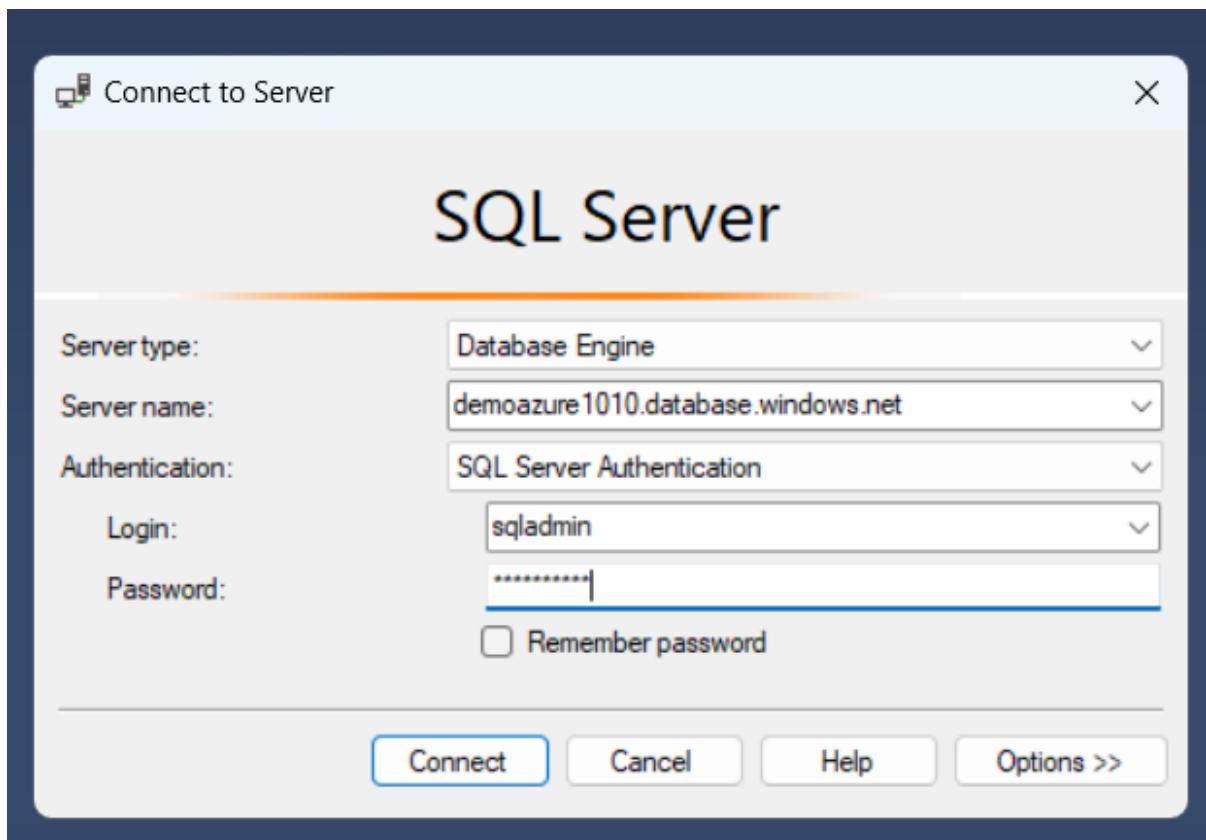
4. First login with your SQL Database Server for that in Azure Portal navigate to your Database.
5. From its overview you need to copy the server's name.

Azure Portal screenshot showing the 'Essentials' section for a database server. The 'Server name' field is highlighted with a red box. Other visible fields include Resource group, Status, Location, Subscription, Subscription ID, Tags, and various connection and pricing details.

Essentials		JSON View
Resource group ( <a href="#">move</a> )	: demo-resource-group	Server name : <a href="#">demoazure1010.database.windows.net</a>
Status	: Online	Elastic pool : <a href="#">No elastic pool</a>
Location	: North Europe	Connection strings : <a href="#">Show database connection strings</a>
Subscription ( <a href="#">move</a> )	: Azure Pass - Sponsorship	Pricing tier : Basic
Subscription ID	: e41df6f3-2d66-416f-9924-552b6cda27ec	Earliest restore point : 2024-04-15 07:05 UTC
Tags ( <a href="#">edit</a> )	: Add tags	

Getting started    Monitoring    Properties    Features    Notifications (0)    Integrations    Tutorials

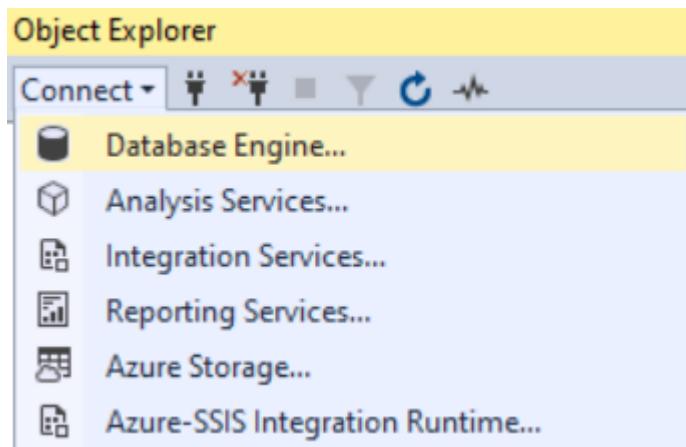
6. Now paste the server here in SSMS then give your login details and click on connect.



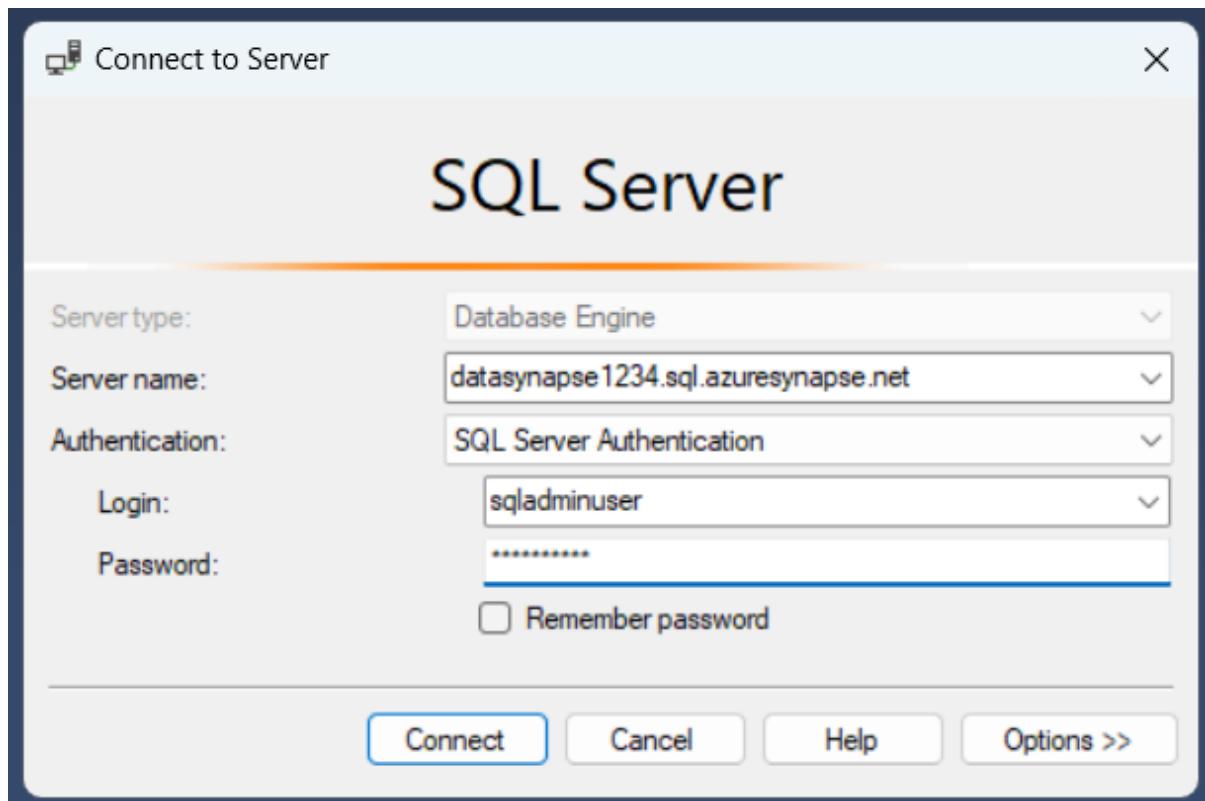
7. Once you are connected then navigate to Azure Portal again and go to Synapse workspace.
8. Now here you will see Dedicated SQL endpoint, copy it to clip board.

Resource group ( <a href="#">move</a> ) <a href="#">demo-resource-group</a>	Networking <a href="#">Show firewall settings</a>
Status Succeeded	Primary ADLS Gen2 account URL <a href="https://demosynapse12.dfs.core.windows.net">https://demosynapse12.dfs.core.windows.net</a>
Location North Europe	Primary ADLS Gen2 file system demofilesystem12
Subscription ( <a href="#">move</a> ) <a href="#">Azure Pass - Sponsorship</a>	SQL admin username sqladminuser
Subscription ID e41df6f3-2d66-416f-9924-552b6cda27ec	SQL Microsoft Entra admin <a href="mailto:live.com#behal.ritesh@gmail.com">live.com#behal.ritesh@gmail.com</a>
Managed virtual network No	Dedicated SQL endpoint datasynapse1234.sql.azuresynapse.net
Managed Identity object ID 556a84d9-ccf1-4263-99b2-4a4fd1bc33e8	Serverless SQL endpoint datasynapse1234-on-demand.sql.azuresynapse.net
Workspace web URL <a href="https://web.azuresynapse.net?workspace=%2bsubscriptions%2fe41df...">https://web.azuresynapse.net?workspace=%2bsubscriptions%2fe41df...</a>	Development endpoint <a href="https://datasynapse1234.dev.azuresynapse.net">https://datasynapse1234.dev.azuresynapse.net</a>

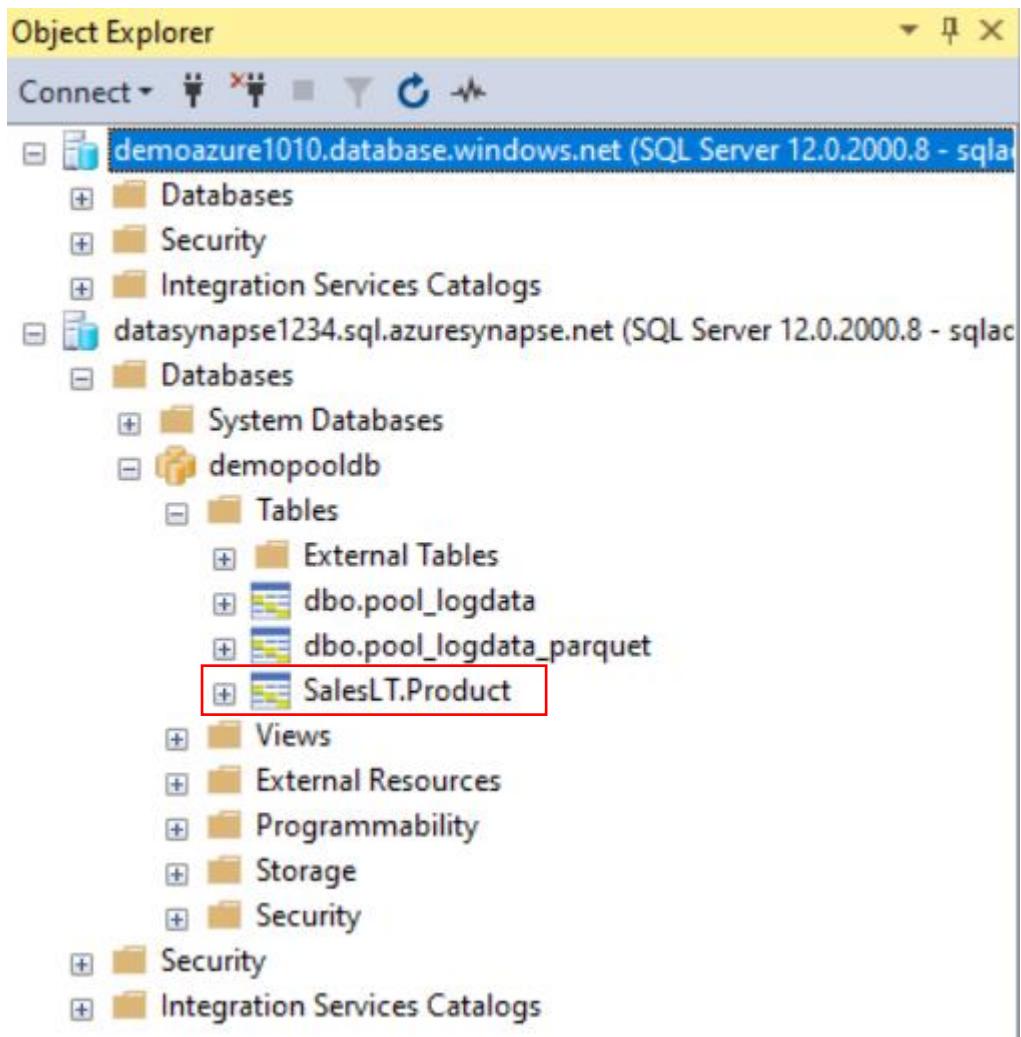
9. Now come back to your SSMS and click on connect choose database engine.



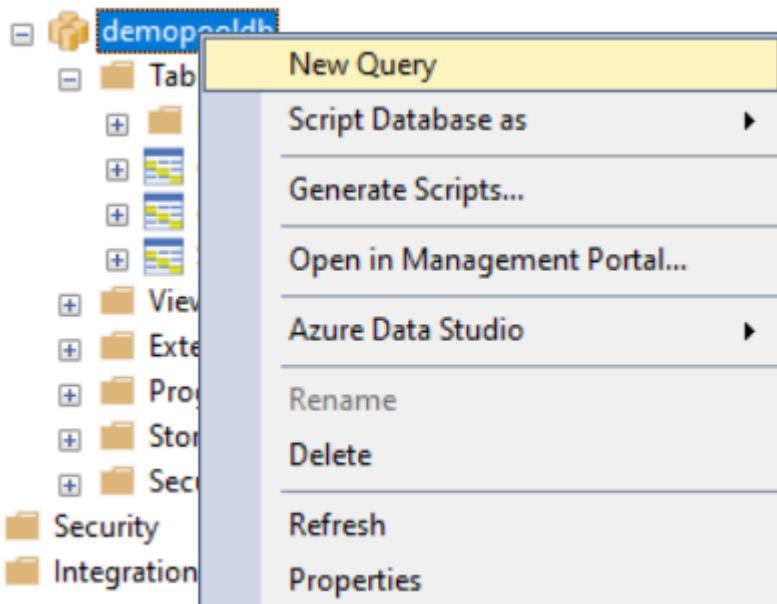
10. Now here you must paste the endpoint and in login details you can get the username from Synapse Workspace overview and then write your password.  
 11. Then click on connect.



12. Below you can see your SQL database and Synapse workspace. Within your Synapse workspace, you can see your tables.
13. First we are going to drop SalesLT table.



14. Now right click on demopooldb and choose New Query.



15. Below you can see that the tables has been dropped successfully.

SQLQuery1.sql - dat... (sqladminuser (0)) \* X

```
DROP TABLE [SalesLT].[Product]
```

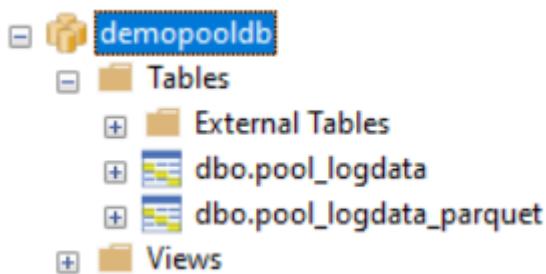
150 %

Messages

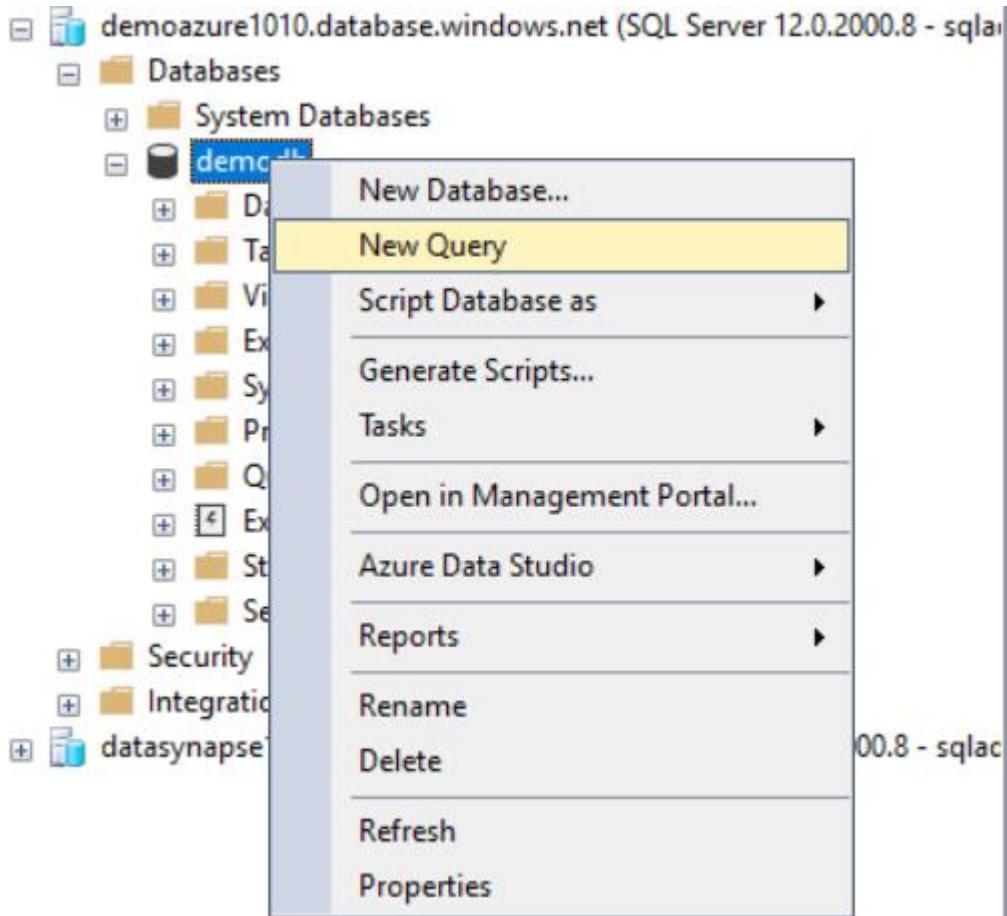
Commands completed successfully.

Completion time: 2024-04-16T16:13:45.2065446+05:30

16. Now if you will expand demopooldb you can see that table is gone.



17. Now expand your SQL Database and select your database then right click on it. After that choose New query.



18. There you are going to create a view for the Sales order detail table and sales order header table.
19. If you do a select command for this view then you can see the data respectively.

SQLQuery2.sql - de...odb (sqladmin (65)) -> SQLQuery1.sql - dat... (sqladminuser (0))

```

CREATE VIEW [Sales_Fact_View]
AS
SELECT dt.[ProductID],dt.[SalesOrderID],dt.[OrderQty],dt.[UnitPrice],hd.[OrderDate],hd.[CustomerID]
FROM [SalesLT].[SalesOrderDetail] dt
LEFT JOIN [SalesLT].[SalesOrderHeader] hd
ON dt.[SalesOrderID]=hd.[SalesOrderID]

SELECT * FROM [Sales_Fact_View]

```

	ProductID	SalesOrderID	OrderQty	UnitPrice	OrderDate	CustomerID	TaxAmt
1	836	71774	1	356.898	2008-06-01 00:00:00.000	29847	70.4279
2	822	71774	1	356.898	2008-06-01 00:00:00.000	29847	70.4279
3	907	71776	1	63.90	2008-06-01 00:00:00.000	30072	6.3048
4	905	71780	4	218.454	2008-06-01 00:00:00.000	30113	3073.4952
5	983	71780	2	461.694	2008-06-01 00:00:00.000	30113	3073.4952
6	988	71780	6	112.998	2008-06-01 00:00:00.000	30113	3073.4952
7	748	71780	2	818.70	2008-06-01 00:00:00.000	30113	3073.4952
8	990	71780	1	323.994	2008-06-01 00:00:00.000	30113	3073.4952
9	926	71780	1	149.874	2008-06-01 00:00:00.000	30113	3073.4952
10	743	71780	1	809.76	2008-06-01 00:00:00.000	30113	3073.4952
11	782	71780	4	1376.994	2008-06-01 00:00:00.000	30113	3073.4952
12	918	71780	2	158.43	2008-06-01 00:00:00.000	30113	3073.4952
13	780	71780	4	1391.994	2008-06-01 00:00:00.000	30113	3073.4952
14	937	71780	1	48.594	2008-06-01 00:00:00.000	30113	3073.4952
15	867	71780	6	41.994	2008-06-01 00:00:00.000	30113	3073.4952
16	985	71780	1	112.998	2008-06-01 00:00:00.000	30113	3073.4952
17	989	71780	2	323.994	2008-06-01 00:00:00.000	30113	3073.4952
18	991	71780	3	323.994	2008-06-01 00:00:00.000	30113	3073.4952
19	992	71780	1	323.994	2008-06-01 00:00:00.000	30113	3073.4952
20	993	71780	2	323.994	2008-06-01 00:00:00.000	30113	3073.4952

Query executed successfully. demoazure1010.database.windows.net | sqladmin (65) | demodb | 00:00:00 | 542 rows

20. After that you are going to create a new table using the Select statement.

The screenshot shows the SQL Server Management Studio interface with two tabs open: 'SQLQuery2.sql - de...odb (sqladmin (65))' and 'SQLQuery1.sql - dat... (sqladminuser (0))'. The 'Results' tab is selected, displaying the output of the following two queries:

```
SELECT [ProductID],[SalesOrderID],[CustomerID],[OrderQty],[UnitPrice],[OrderDate],[TaxAmt]
INTO [FactSales]
FROM [Sales_Fact_View]

SELECT * FROM [FactSales]
```

The results grid shows data from the 'FactSales' table, with columns: ProductID, SalesOrderID, CustomerID, OrderQty, UnitPrice, OrderDate, and TaxAmt. The data consists of 542 rows, starting with row 1 and ending with row 542. The first few rows are:

	ProductID	SalesOrderID	CustomerID	OrderQty	UnitPrice	OrderDate	TaxAmt
1	836	71774	29847	1	356.898	2008-06-01 00:00:00.000	70.4279
2	822	71774	29847	1	356.898	2008-06-01 00:00:00.000	70.4279
3	907	71776	30072	1	63.90	2008-06-01 00:00:00.000	6.3048
4	905	71780	30113	4	218.454	2008-06-01 00:00:00.000	3073.4952
5	983	71780	30113	2	461.694	2008-06-01 00:00:00.000	3073.4952
6	988	71780	30113	6	112.998	2008-06-01 00:00:00.000	3073.4952
7	748	71780	30113	2	818.70	2008-06-01 00:00:00.000	3073.4952
8	990	71780	30113	1	323.994	2008-06-01 00:00:00.000	3073.4952
9	926	71780	30113	1	149.874	2008-06-01 00:00:00.000	3073.4952
10	743	71780	30113	1	809.76	2008-06-01 00:00:00.000	3073.4952
11	782	71780	30113	4	1376.994	2008-06-01 00:00:00.000	3073.4952
12	918	71780	30113	2	158.43	2008-06-01 00:00:00.000	3073.4952
13	780	71780	30113	4	1391.994	2008-06-01 00:00:00.000	3073.4952
14	937	71780	30113	1	48.594	2008-06-01 00:00:00.000	3073.4952
15	867	71780	30113	6	41.994	2008-06-01 00:00:00.000	3073.4952
16	985	71780	30113	1	112.998	2008-06-01 00:00:00.000	3073.4952
17	989	71780	30113	2	323.994	2008-06-01 00:00:00.000	3073.4952
18	991	71780	30113	3	323.994	2008-06-01 00:00:00.000	3073.4952
19	992	71780	30113	1	323.994	2008-06-01 00:00:00.000	3073.4952
20	993	71780	30113	2	323.994	2008-06-01 00:00:00.000	3073.4952

At the bottom left, there is a green checkmark icon followed by the text 'Query executed successfully.' On the right side, there is a status bar showing 'demoazure1010.database.wind...' and other connection details.