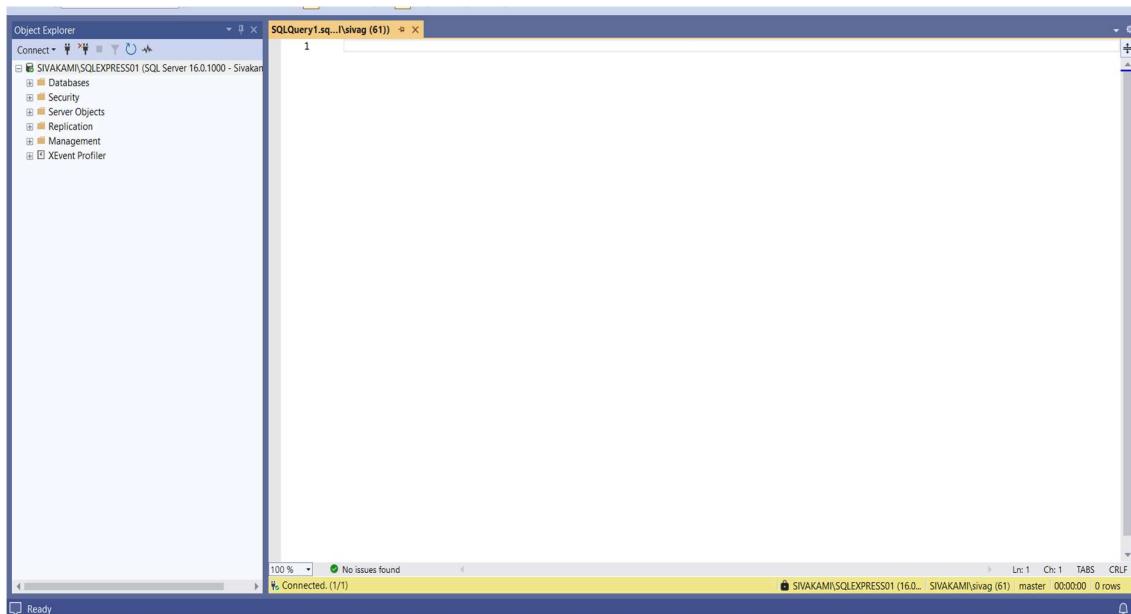


SQL Server Query Execution Documentation

Project Title: Texture Tales – SQL Query Documentation

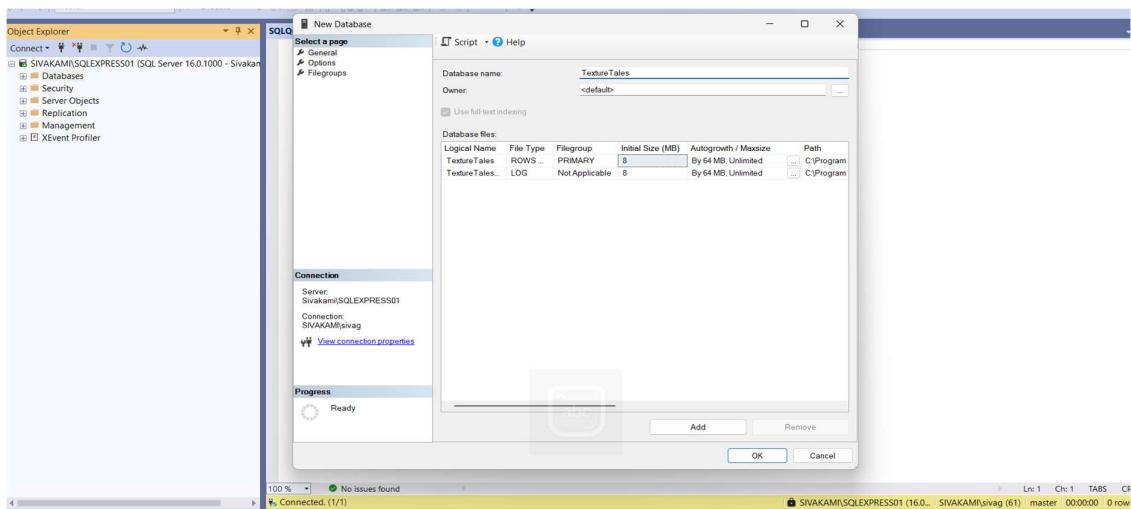
✓ Step 1: Open SQL Server Management Studio

1. Click **Start** and open **SQL Server Management Studio**.
2. Connect to the SQL Server using your server name and credentials.



✓ Step 2: Create Database in SSMS:

Right-click on Databases → **click New Database...** → **enter the name (e.g., texture)** → **click OK.**



✓ Step 3: Run SQL Queries

📌 Query 1: Total Quantity Sold for All Products

The screenshot shows the Object Explorer on the left with a connection to SIVAKAMISQLEXPRESS01. The SQL Query1.sql window on the right contains the following code:

```
131
132 /* 1.Total quantity sold for all products */
133
134
135 /* SELECT SUM(qty) AS total_quantity_sold
136
137 FROM sales;
138
139
140
141
142
143
144
145
146
147
148
149
```

The Results tab shows the output:

total_quantity_sold
145

📌 Query 2: Average Discount Value Per Transaction

The screenshot shows the Object Explorer on the left with a connection to SIVAKAMISQLEXPRESS01. The SQL Query1.sql window on the right contains the following code:

```
141
142 /* 2.Average discount value per transaction */
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
```

The Results tab shows the output:

avg_discount_per_transaction
231

📌 Query 3: Top Selling Product for Each Segment

The screenshot shows the SQL Server Management Studio interface with two tabs open: 'TextFile_Tale...\\sivag (77)*' and 'SQLQuery1.sql - M\\sivag (75)'. The left pane displays the Object Explorer with the database 'SIVAKAMI\\SQLEXPRESS01' selected. The right pane contains the query code and its results.

```
153 /* 3.Top selling product for each segment*/
154
155 WITH SegmentSales AS (
156     SELECT
157         pd.segment_name,
158         pd.product_name,
159         s.prod_id,
160         SUM(s.qty) AS total_quantity,
161         ROW_NUMBER() OVER (
162             PARTITION BY pd.segment_name
163             ORDER BY SUM(s.qty) DESC
164         ) AS rn
165     FROM sales s JOIN product_details pd ON s.prod_id = pd.product_id
166     GROUP BY pd.segment_name, pd.product_name, s.prod_id
167 )
168     SELECT segment_name, product_name, prod_id, total_quantity
169     FROM SegmentSales
170     WHERE rn = 1;
```

The results grid shows the top selling products for each segment:

segment_name	product_name	prod_id	total_quantity
1 Jacket	Khaki Suit Jacket - Womens	d5e9ef	15
2 Jeans	Navy Oversized Jeans - Womens	c4d532	20
3 Shirt	Blue Polo Shirt - Mens	2a2353	12
4 Socks	Navy Solid Socks - Mens	f084eb	18

At the bottom, a message bar indicates: 'Query executed successfully.' and 'SIVAKAMI\\SQLEXPRESS01 (16.0.0.0) - SIVAKAMI\\sivag (77) | TextureTales | 00:00:00 | 4 rows'.

📌 Query 4: Top 3 Products by Total Revenue (Before Discount)

The screenshot shows the SQL Server Management Studio interface with two tabs open: 'TextFile_Tale...\\sivag (77)*' and 'SQLQuery1.sql - M\\sivag (75)'. The left pane displays the Object Explorer with the database 'SIVAKAMI\\SQLEXPRESS01' selected. The right pane contains the query code and its results.

```
172
173
174
175
176 /* 4.Top 3 products by total revenue before discount */
177
178 SELECT TOP 3
179     pd.product_name,
180     s.prod_id,
181     SUM(s.qty * s.price) AS total_revenue
182     FROM sales s
183     JOIN product_details pd ON s.prod_id = pd.product_id
184     GROUP BY pd.product_name, s.prod_id
185     ORDER BY total_revenue DESC;
```

The results grid shows the top 3 products by total revenue:

product_name	prod_id	total_revenue
1 Blue Polo Shirt - Mens	2a2353	684
2 Navy Solid Socks - Mens	f084eb	648
3 Grey Fashion Jacket - Womens	9ed847	486

At the bottom, a message bar indicates: 'Query executed successfully.' and 'SIVAKAMI\\SQLEXPRESS01 (16.0.0.0) - SIVAKAMI\\sivag (77) | TextureTales | 00:00:00 | 3 rows'.

📌 Query 5: Total Quantity, Revenue & Discount per Segment

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists databases like SIVAKAMI\SQLEXPRESS01, System Databases, and TextureTales. The main area contains a query window titled 'SQLQuery1sql_Misivag (75)'. The code is as follows:

```
/* 5. total quantity, total revenue (before discount), and total discount for each segment */
SELECT pd.segment_name,
       SUM(s.qty) AS total_quantity,
       SUM(s.qty * s.price) AS total_revenue,
       SUM(s.qty * s.discount) AS total_discount
FROM sales s
JOIN product_details pd ON s.prod_id = pd.product_id
GROUP BY pd.segment_name;
```

Below the code, the results pane shows a table with four rows of data:

	segment_name	total_quantity	total_revenue	total_discount
1	Jacket	37	1078	521
2	Jeans	48	738	753
3	Shirt	28	1204	450
4	Socks	32	934	587

The status bar at the bottom indicates 'Query executed successfully.' and shows the session details: SIVAKAMI\SQLEXPRESS01 (16.0.1000.6) - SIVAKAMI\misivag (77) - TextureTales | 00:00:00 | 4 rows.