

# BlinkIt Data Analysis using SQL

**1. Total Sales** : The overall revenue generated from all items sold.

## UPDATE blinkit\_grocery\_data

*SET item\_fat\_content =*

## CASE

*WHEN item\_fat\_content = 'Regular' OR item\_fat\_content = 'reg' THEN 'regular'*

*WHEN item\_fat\_content = 'Low Fat' OR item\_fat\_content = 'LF' THEN 'low\_fat'*

*ELSE item\_fat\_content*

*END;*

```
SELECT item_fat_content FROM blinkit_grocery_data;
```

The screenshot shows a SQL editor window titled "blinkit\_sql". The query being executed is as follows:

```

1 • UPDATE blinkit_grocery_data
2   SET item_fat_content =
3     CASE
4       WHEN item_fat_content = 'Regular' OR item_fat_content = 'reg' THEN 'regular'
5       WHEN item_fat_content = 'Low Fat' OR item_fat_content = 'LF' THEN 'low_fat'
6       ELSE item_fat_content
7     END;
8
9 • select item_fat_content from blinkit_grocery_data;
10

```

Below the query, the "Result Grid" is displayed, showing the results of the SELECT statement. The grid has two columns: "item\_fat\_content" and "item\_id". The first row is expanded, showing the values "regular" and "1".

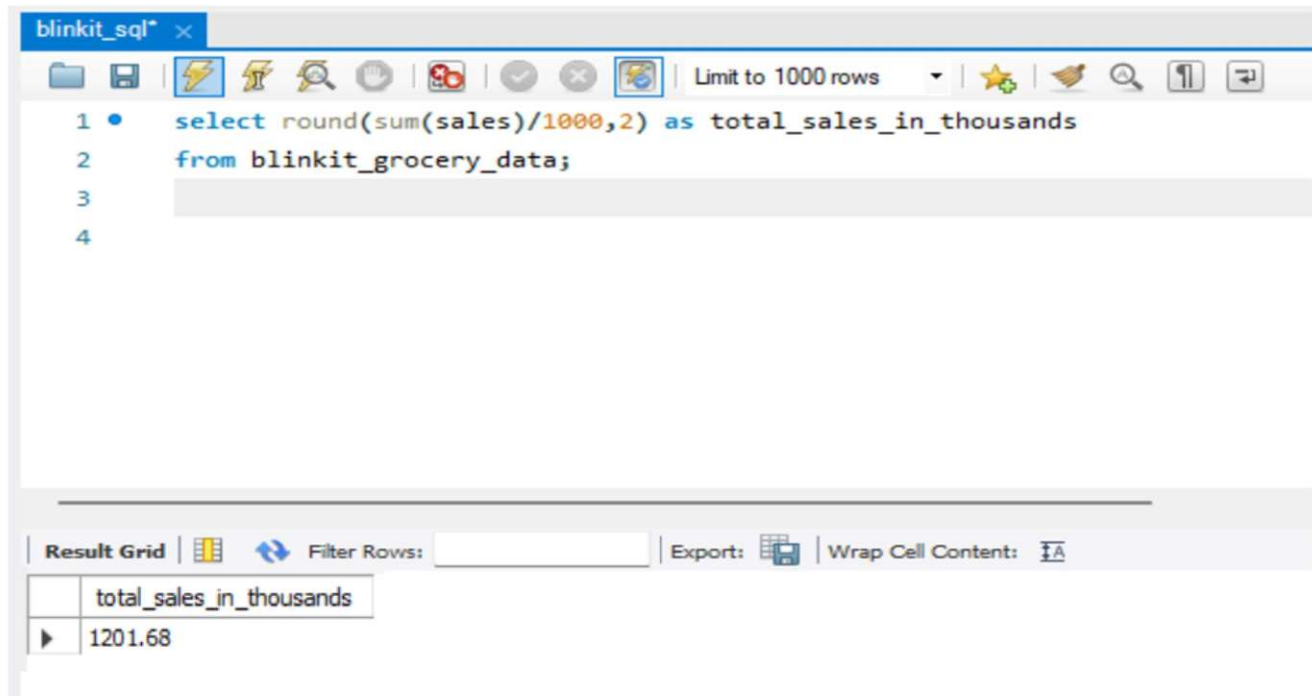
item_fat_content	item_id
regular	1
low_fat	2
regular	3
regular	4
low_fat	5
low_fat	6
low_fat	7
low_fat	8
low_fat	9
low_fat	10

*CREATE DATABASE blinkit;*

*SELECT*

*ROUND(SUM(sales)/1000,2) as total\_sales\_in\_thousands*

*FROM blinkit\_grocery\_data;*



The screenshot shows a SQL query editor window titled "blinkit\_sql". The query is as follows:

```
1 • select round(sum(sales)/1000,2) as total_sales_in_thousands
2   from blinkit_grocery_data;
3
4
```

The editor has a toolbar with various icons and a "Limit to 1000 rows" dropdown. Below the query editor, there is a "Result Grid" section with the following data:

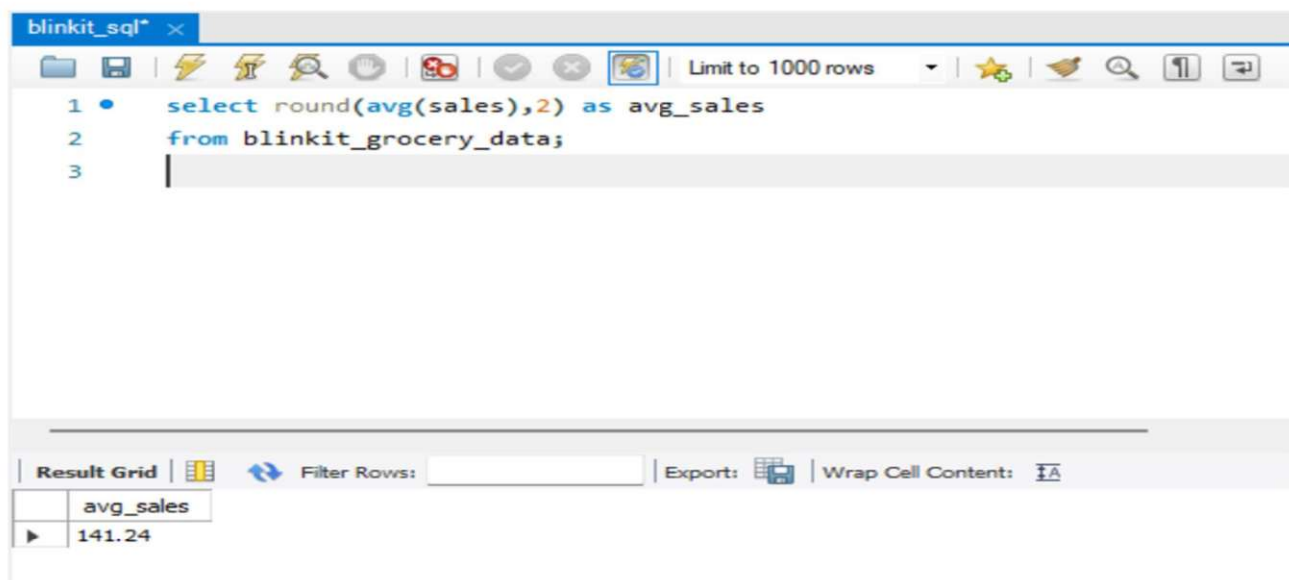
	total_sales_in_thousands
▶	1201.68

## ***2. Average Sales : The average revenue sale.***

*SELECT*

*ROUND(AVG(sales),2) as avg\_sales*

*FROM blinkit\_grocery\_data;*



The screenshot shows a SQL query editor window titled "blinkit\_sql". The query is as follows:

```
1 • select round(avg(sales),2) as avg_sales
2   from blinkit_grocery_data;
3
```

The editor has a toolbar with various icons and a "Limit to 1000 rows" dropdown. Below the query editor, there is a "Result Grid" section with the following data:

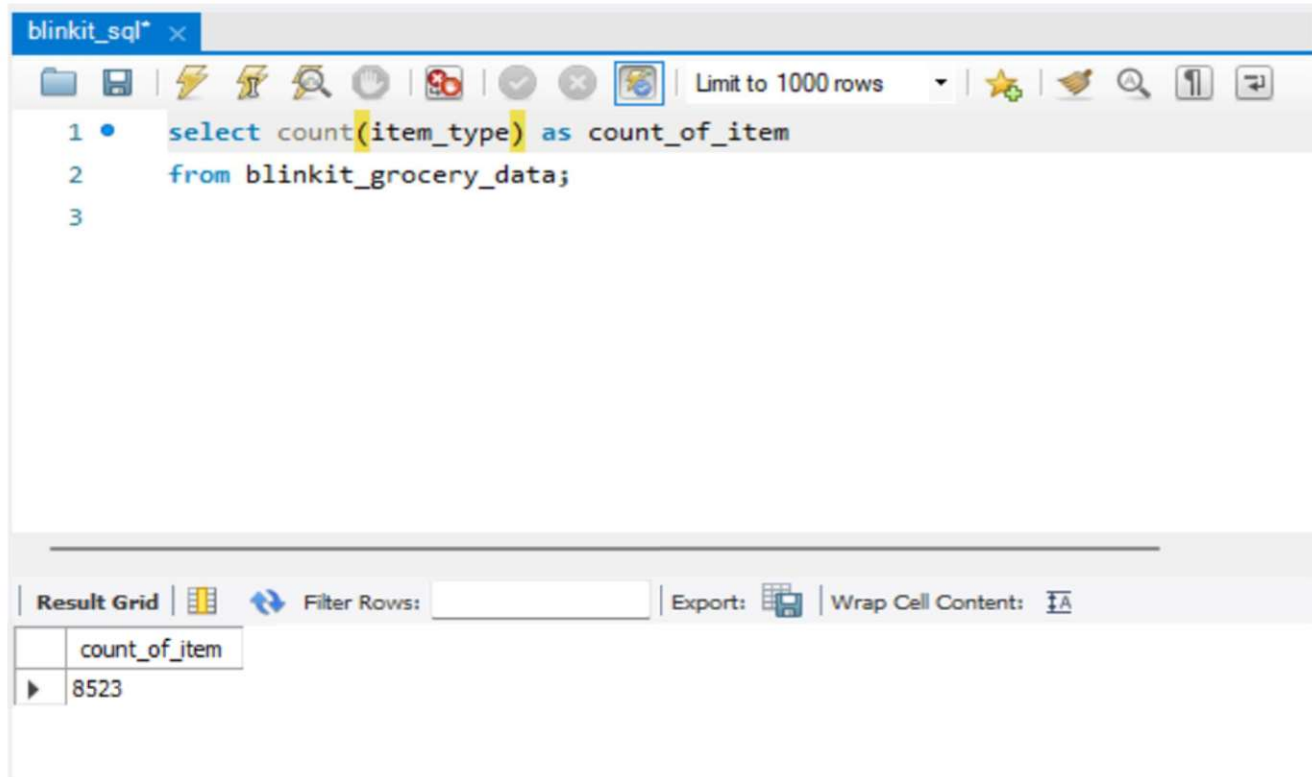
	avg_sales
▶	141.24

### ***3. Number of Items : The total count of different items sold.***

*SELECT*

*COUNT(item\_type) AS count\_of\_item*

*FROM blinkit\_grocery\_data;*



The screenshot shows a SQL query editor window titled "blinkit\_sql". The query is: `select count(item_type) as count_of_item from blinkit_grocery_data;`. The result is displayed in a table with one row: 

count_of_item
8523

. The editor includes a toolbar with various icons and a "Limit to 1000 rows" dropdown.

### ***4. Average Rating : The average customer rating for items sold.***

*SELECT*

*ROUND(AVG(rating),2) AS avg\_rating*

*FROM blinkit\_grocery\_data;*

The screenshot shows a SQL query editor window titled "blinkit\_sql". The query is as follows:

```
1 • select round(avg(rating),2) as avg_rating
2   from blinkit_grocery_data;
3
```

Below the query editor, the "Result Grid" is displayed with the following data:

avg_rating
3.96

## **5. Total Sales by Fat Content:**

*SELECT*

*ROUND(SUM(sales)/1000,2) AS sales\_in\_thousands, item\_fat\_content*

*FROM blinkit\_grocery\_data*

*GROUP BY item\_fat\_content;*

The screenshot shows a SQL query editor window titled "blinkit\_sql". The query is as follows:

```
1 • select round(sum(sales)/1000,2) as sales_in_thousands, item_fat_content
2   from blinkit_grocery_data
3  group by item_fat_content;
4
5
6
```

Below the query editor, the "Result Grid" is displayed with the following data:

sales_in_thousands	item_fat_content
352.64	regular
644.52	low_fat

## **6. Total Sales by Item Type:**

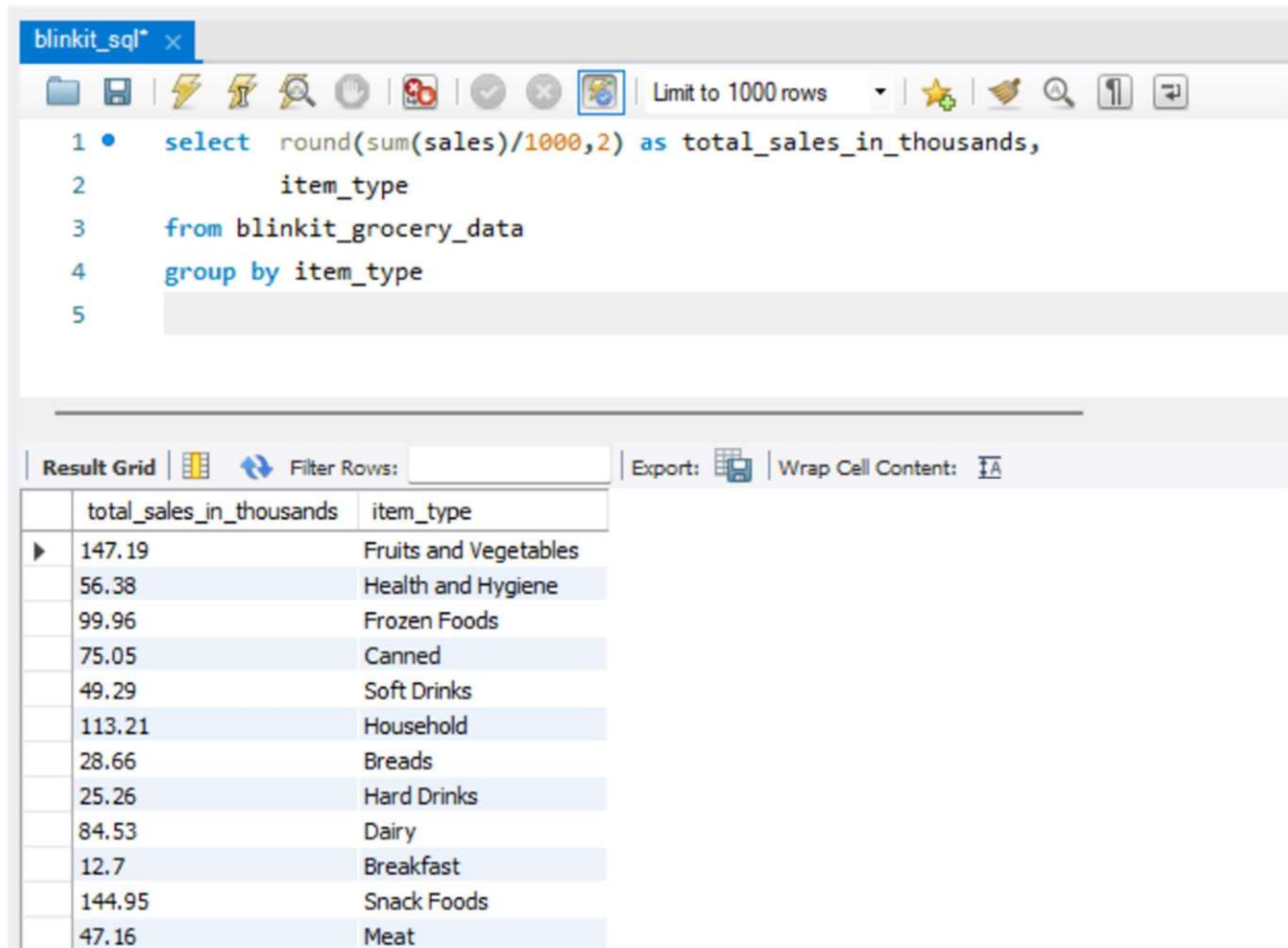
*SELECT*

*ROUND(SUM(sales)/1000,2) AS total\_sales\_in\_thousands,*

*item\_type*

*FROM blinkit\_grocery\_data*

*GROUP BY item\_type*



The screenshot shows a SQL query editor window titled 'blinkit\_sql\*'. The query is as follows:

```
1 • select round(sum(sales)/1000,2) as total_sales_in_thousands,
2       item_type
3 from blinkit_grocery_data
4 group by item_type
5
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field and an 'Export' button. The results are displayed in a table with two columns: 'total\_sales\_in\_thousands' and 'item\_type'.

total_sales_in_thousands	item_type
147.19	Fruits and Vegetables
56.38	Health and Hygiene
99.96	Frozen Foods
75.05	Canned
49.29	Soft Drinks
113.21	Household
28.66	Breads
25.26	Hard Drinks
84.53	Dairy
12.7	Breakfast
144.95	Snack Foods
47.16	Meat

## **7.Fat Content by Outlet for Total Sales:**

*SELECT*

*item\_fat\_content,*

*outlet\_type,*

*ROUND(SUM(sales)/1000,2) AS total\_sales\_thousands*

*FROM blinkit\_grocery\_data*

*GROUP BY item\_fat\_content, outlet\_type;*



blinkit\_sql\* x

Limit to 1000 rows

```

1 • select item_fat_content,
2         outlet_type,
3         round(sum(sales)/1000,2) as total_sales_thousands
4 from blinkit_grocery_data
5 group by item_fat_content, outlet_type;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	item_fat_content	outlet_type	total_sales_thousands
▶	regular	Supermarket Type1	279.66
	low_fat	Supermarket Type2	84.84
	low_fat	Supermarket Type1	507.89
	low_fat	Grocery Store	51.79
	regular	Grocery Store	26.35
	regular	Supermarket Type2	46.63

## **8. Total Sales by Outlet Establishment:**

**SELECT**

*outlet\_establishment\_year,*

*ROUND(SUM(sales)/1000,2) AS total\_sales\_thousands*

*FROM blinkit\_grocery\_data*

*GROUP BY outlet\_establishment\_year*

*ORDER BY outlet\_establishment\_year asc;*

blinkit\_sql\*

Limit to 1000 rows

```

1 • select outlet_establishment_year,
2         round(sum(sales)/1000,2) as total_sales_thousands
3 from blinkit_grocery_data
4 group by outlet_establishment_year
5 order by outlet_establishment_year asc;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	outlet_establishment_year	total_sales_thousands
▶	2011	78.13
	2012	130.48
	2014	131.81
	2015	130.94
	2016	132.11
	2017	133.1
	2020	129.1
	2022	131.48

## 9. Percentage of Sales by Outlet Size:

*SELECT*

*outlet\_size,*

*ROUND(SUM(sales) \* 100.0 / (SELECT SUM(sales) FROM blinkit\_grocery\_data),2) AS  
percentage\_sales\_by\_outlet\_size*

*FROM*

*blinkit\_grocery\_data*

*GROUP BY*

*outlet\_size;*

blinkit\_sql x

Limit to 1000 rows

```
1 SELECT
2     outlet_size,
3     round(SUM(sales) * 100.0 / (SELECT SUM(sales) FROM blinkit_grocery_data),2) AS percentage_sales_by_outlet_size
4 FROM
5     blinkit_grocery_data
6 GROUP BY
7     outlet_size;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	outlet_size	percentage_sales_by_outlet_size
▶	Medium	37.83
	Small	37.2
	High	24.97

## **10. Sales by Outlet Location:**

***SELECT***

***outlet\_location\_type,***

***ROUND(SUM(sales) / 1000, 2) AS sales\_in\_thousands***

***FROM***

***blinkit\_grocery\_data***

***GROUP BY***

***outlet\_location\_type;***



The screenshot shows a SQL IDE window titled 'blinkit\_sql'. The query editor contains the following SQL code:

```

1 • SELECT
2     outlet_location_type,
3     ROUND(SUM(sales) / 1000, 2) AS sales_in_thousands
4 FROM
5     blinkit_grocery_data
6 GROUP BY
7     outlet_location_type;

```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has two columns: 'outlet\_location\_type' and 'sales\_in\_thousands'. The results are as follows:

outlet_location_type	sales_in_thousands
Tier 1	262.59
Tier 3	341.42
Tier 2	393.15

## **11. All Metrics by Outlet Type**

**SELECT**

*outlet\_type,*

*ROUND(SUM(sales)/1000, 2) AS sales\_in\_thousands,*

*round(avg(sales),2) as avg\_sales,*

*item\_type,*

*round(avg(rating),2)*

**FROM**

*blinkit\_grocery\_data*

**GROUP BY**

*outlet\_type, item\_type;*

blinkit\_sql\*



```

1 • SELECT
2     outlet_type,
3     ROUND(SUM(sales) / 1000, 2) AS sales_in_thousands,
4     round(avg(sales),2) as avg_sales,
5     item_type,
6     round(avg(rating),2) as avg_rating
7 FROM
8     blinkit_grocery_data
9 GROUP BY
10    outlet_type,item_type;

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	outlet_type	sales_in_thousands	avg_sales	item_type	avg_rating
▶	Supermarket Type1	117.43	145.88	Fruits and Vegetables	3.93
	Supermarket Type2	7.34	126.55	Health and Hygiene	3.91
	Supermarket Type1	79.35	138.72	Frozen Foods	3.95
	Supermarket Type1	59.65	140.02	Canned	4.00
	Supermarket Type1	39	130.01	Soft Drinks	3.90
	Grocery Store	4.99	134.95	Health and Hygiene	3.92
	Supermarket Type1	89.14	149.32	Household	4.01
	Supermarket Type1	44.05	131.49	Health and Hygiene	3.99