

MYSQL Queries

See all the data imported:

```
SELECT * FROM blinkit_data
```

Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility
Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.1000
Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.0089
Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.0258
Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.0422
Low Fat	DR125	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.0339
Low Fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.0056
Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.0983
Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.0269
Low Fat	FDW20	Fruits and Vegetables	2000	OUT013	Tier 3	High	Supermarket Type1	0.0241
Low Fat	NCU41	Health and Hygiene	2017	OUT035	Tier 2	Small	Supermarket Type1	0.0520

```
SHOW COLUMNS FROM blinkit_data;
```

Field	Type	Null	Key	Default	Extra
Item Fat Content	text	YES		NULL	
Item Identifier	text	YES		NULL	
Item Type	text	YES		NULL	
Outlet Establishment Year	int	YES		NULL	
Outlet Identifier	text	YES		NULL	
Outlet Location Type	text	YES		NULL	
Outlet Size	text	YES		NULL	
Outlet Type	text	YES		NULL	
Item Visibility	double	YES		NULL	
Item Weight	double	YES		NULL	
Sales	double	YES		NULL	
Rating	int	YES		NULL	

```
CHANGE COLUMN NAME `transaction_id` to transaction_id
```

```
ALTER TABLE blinkit_data
```

```
CHANGE COLUMN `item Fat Content` item_fat_content VARCHAR(50);
```

```
UPDATE blinkit_data
```

```
SET Item_Fat_Content =
```

```
CASE
```

```
WHEN Item_Fat_Content IN ('LF', 'low fat') THEN 'Low Fat'
```

```
WHEN Item_Fat_Content = 'reg' THEN 'Regular'
```

```
ELSE Item_Fat_Content  
END;
```

DATA HAS BEEN CLEANED OR NOT

```
SELECT DISTINCT Item_Fat_Content FROM blinkit_data;
```

Result Grid	
	Item_Fat_Content
▶	Regular
	Low Fat

A. KPI's

1. TOTAL SALES:

```
Select sum(Sales) AS Total_Sales  
From blinkit_data
```

Result Grid	
	Total_Sales
▶	1247309.3020000043

```
SELECT CAST(SUM(Sales) / 1000000.0 AS DECIMAL(10,2)) AS Total_Sales_Million  
FROM blinkit_data;
```

Result Grid	
	Total_Sales_Million
▶	1.25

2. AVERAGE SALES

```
SELECT  
CAST(AVG(Sales) AS SIGNED) AS Avg_Sales  
FROM blinkit_data;
```

Result Grid		Filter Rows:
	Avg_Sales	
▶	141	

3. NO OF ITEMS

```
SELECT COUNT(*) AS No_of_Orders
FROM blinkit_data;
```

4. AVG RATING

```
SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg_Rating
FROM blinkit_data;
```

Result Grid		Filter Rows:
	Avg_Rating	
▶	4.1	

B. Total Sales by Fat Content:

```
SELECT
`Item_Fat_Content`,
CAST(SUM(`Sales`) AS DECIMAL(10,2)) AS Total_Sales
FROM blinkit_data
```

Result Grid		Filter Rows:	Export:	Wrap Cell
	Item_Fat_Content	Total_Sales		
▶	Regular	443172.75		
	Low Fat	804136.55		

C. Total Sales by Item Type

```
SELECT
```

```

`Item Type`,
ROUND(SUM(`Sales`), 2) AS Total_Sales
FROM blinkit_data
GROUP BY `Item Type`
ORDER BY Total_Sales DESC;

```

	Item Type	Total_Sales
▶	Fruits and Vegetables	184057.01
	Snack Foods	182958.43
	Household	139916.94
	Frozen Foods	125040.75
	Dairy	105531.19
	Canned	3386.24
	Baking Goods	83765.58
	Health and Hygiene	70868.04
	Soft Drinks	62005.18
	Meat	60292.27
	Breads	36766.63
	Hard Drinks	31913.99

	Item Type	Total_Sales
	Canned	93386.24
	Baking Goods	83765.58
	Health and Hygiene	70868.04
	Soft Drinks	62005.18
	Meat	60292.27
	Breads	36766.63
	Hard Drinks	31913.99
	Others	23467.96
	Starchy Foods	22876.45
	Breakfast	16017.14
	Seafood	8445.5

D. Fat Content by Outlet for Total Sales

```

SELECT
`Outlet Location Type`,
ROUND(SUM(CASE WHEN `Item_Fat_Content` = 'Low Fat' THEN `Sales` ELSE 0 END), 2) AS Low_Fat,
ROUND(SUM(CASE WHEN `Item_Fat_Content` = 'Regular' THEN `Sales` ELSE 0 END), 2) AS Regular
FROM blinkit_data
GROUP BY `Outlet Location Type`

```

```
ORDER BY `Outlet Location Type`;
```

	Outlet Location Type	Low_Fat	Regular
▶	Tier 1	212759.85	125090.33
	Tier 2	313301.17	170396.42
	Tier 3	278075.53	147686

E. Total Sales by Outlet Establishment

```
SELECT  
    `Outlet Establishment Year`,  
    ROUND(SUM(`Sales`), 2) AS Total_Sales  
FROM blinkit_data  
GROUP BY `Outlet Establishment Year`  
ORDER BY `Outlet Establishment Year`;
```

	Outlet Establishment Year	Total_Sales
▶	2000	164256.83
	2010	164607.63
	2011	98290.92
	2012	173242.55
	2015	159547.61
	2017	164347.86
	2020	159802.13
	2022	163213.77

F. Percentage of Sales by Outlet Size

```
SELECT  
    `Outlet Size`,  
    ROUND(SUM(`Sales`), 2) AS Total_Sales,  
    ROUND(  
        SUM(`Sales`) * 100.0 / SUM(SUM(`Sales`)) OVER (),  
        2  
) AS Sales_Percentage  
FROM blinkit_data  
GROUP BY `Outlet Size`
```

```
ORDER BY Total_Sales DESC;
```

The screenshot shows a database query results grid titled "Result Grid". The columns are "Outlet Size", "Total_Sales", and "Sales_Percentage". The data rows are:

Outlet Size	Total_Sales	Sales_Percentage
Medium	481373.27	38.59
Small	458335.55	36.75
High	307600.48	24.66

G. Sales by Outlet Location

```
SELECT  
    `Outlet Location Type`,  
    ROUND(SUM(`Sales`), 2) AS Total_Sales  
FROM blinkit_data  
GROUP BY `Outlet Location Type`  
ORDER BY Total_Sales DESC;
```

The screenshot shows a database query results grid titled "Result Grid". The columns are "Outlet Location Type" and "Total_Sales". The data rows are:

Outlet Location Type	Total_Sales
Tier 2	483697.6
Tier 3	425761.53
Tier 1	337850.18

H. All Metrics by Outlet Type:

```
SELECT  
    `Outlet Type`,  
    ROUND(SUM(`Sales`), 2) AS Total_Sales,  
    ROUND(AVG(`Sales`), 0) AS Avg_Sales,  
    COUNT(*) AS No_Of_Items,  
    ROUND(AVG(`Rating`), 2) AS Avg_Rating,  
    ROUND(AVG(`Item Visibility`), 2) AS Item_Visibility  
FROM blinkit_data  
GROUP BY `Outlet Type`  
ORDER BY Total_Sales DESC;
```

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

	Outlet Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating	Item_Visibility
▶	Supermarket Type1	985804.61	141	6990	4.05	0.06
	Supermarket Type2	163213.77	142	1146	4.04	0.06
	Grocery Store	98290.92	140	700	4.08	0.1