Blinkit SQL Queries

1.

```
Execute
SQLQuery1.sql - D...O6Q163\Admin (77))*
    SELECT * FROM blinkit_data
```

2.

```
SQLQuery1.sql - D...O6Q163\Admin (77))*
    SELECT COUNT(Item_Identifier) AS No_of_items
    FROM [blinkit_data]
```

121 %

Results | Messages

	No_of_items
1	8523

CLEANING PART

3.

```
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

SELECT DISTINCT(Item_Fat_Content) FROM blinkit_data
```

121 %

Results | Messages

	Item_Fat_Content
1	Low Fat
2	Regular

BASICS KPI'S

4.

```sql
SELECT CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)),'M')
AS Total_sales_Millions
FROM blinkit_data
```

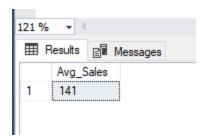|	Total_sales_Millions
1	1.20M

5.

```sql
SELECT CAST(AVG(Sales) AS DECIMAL(10,0)) AS Avg_Sales FROM blinkit_data
```

121 %

	Avg_Sales
1	141

6.

```sql
SELECT CAST(AVG(Rating) AS DECIMAL(10,2))
AS  Avg_rating
FROM blinkit_data
```

121 %

	Avg_rating
1	3.97

7.

```sql
SELECT Item_Fat_Content,
    CONCAT(CAST(SUM(Sales)/1000 AS DECIMAL(10,2)),'K') AS Total_Sales ,
    CAST(AVG(Sales) AS DECIMAL(10,2)) AS Avg_Sales,
    CONCAT(CAST(COUNT(*)/1000 AS DECIMAL(10,2)),'K') AS No_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
    FROM blinkit_data
    GROUP BY Item_Fat_Content
    ORDER BY Total_Sales
```
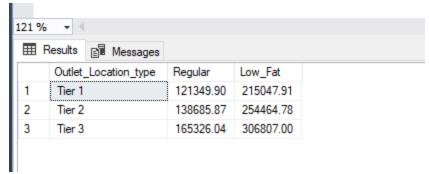
	Item_Fat_Content	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Regular	425.36K	141.50	3.00K	3.97
2	Low Fat	776.32K	140.71	5.00K	3.97

8.

```sql
SELECT Item_Type,
    SUM(Sales) AS Total_Sales ,
    CAST(AVG(Sales) AS DECIMAL(10,2)) AS Avg_Sales,
    COUNT(*)AS No_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
    FROM blinkit_data
    GROUP BY Item_Type
    ORDER BY Total_Sales
```

121 %

	Item_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Seafood	9077.86996841431	141.84	64	3.96
2	Breakfast	15596.6964912415	141.79	110	3.93
3	Starchy Foods	21880.0274009705	147.84	148	3.92
4	Others	22451.8916893005	132.85	169	3.95
5	Hard Drinks	29334.6805801392	137.08	214	3.91
6	Breads	35379.1197090149	140.95	251	3.88
7	Soft Drinks	58514.1668949127	131.49	445	3.92
8	Meat	59449.8637466431	139.88	425	4.02
9	Health and Hygiene	68025.8388824463	130.82	520	3.99
10	Baking Goods	81894.7364845276	126.38	648	3.98
11	Canned	90706.7288208008	139.76	649	3.99
12	Dairy	101276.461353302	148.50	682	3.97
13	Frozen Foods	118558.881378174	138.50	856	3.97
14	Household	135976.52520752	149.42	910	4.00
15	Snack Foods	175433.922431946	146.19	1200	3.95
16	Fruits and Vegeta...	178124.080921173	144.58	1232	3.96

9.

```
SQLQuery1.sql - D...O6Q163\Admin (76))*  ⊡ ✕
    ⊟SELECT Outlet_Location_type,
            ISNULL([Regular],0) AS Regular,
            ISNULL([Low Fat],0) AS Low_Fat
    FROM
    (
        SELECT Outlet_Location_Type , Item_Fat_content,
        CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
        FROM blinkit_data
        GROUP BY Outlet_Location_Type , Item_Fat_content
    ) AS SourceCode
    PIVOT
    (
    SUM(Total_Sales)
    FOR Item_Fat_Content IN ([Regular],[Low Fat])
    ) AS PivotTable
    ORDER BY Outlet_Location_Type
```

121 % ▾ ◂

⊞ Results 🗎 Messages

	Outlet_Location_type	Regular	Low_Fat
1	Tier 1	121349.90	215047.91
2	Tier 2	138685.87	254464.78
3	Tier 3	165326.04	306807.00

10.

```
SQLQuery2.sql - D...O6Q163\Admin (61))*  ⊡ ✕   SQLQuery1.sql - D...O6Q163\Admin (76))*
    ⊟SELECT Outlet_Establishment_Year,
    CAST(SUM(Sales) AS DECIMAL(10,2))AS Total_Sales,
    CAST(AVG(Sales) AS DECIMAL(10,2)) AS Avg_sales,
    COUNT(*) AS Total_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
    FROM blinkit_data
    GROUP BY Outlet_Establishment_Year
    ORDER BY Outlet_Establishment_Year
```
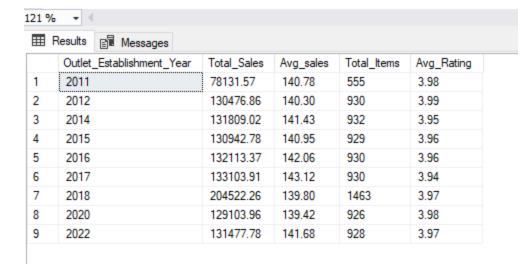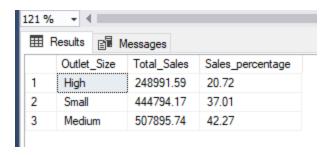
121 % ▼ ◄

	Outlet_Establishment_Year	Total_Sales	Avg_sales	Total_Items	Avg_Rating
1	2011	78131.57	140.78	555	3.98
2	2012	130476.86	140.30	930	3.99
3	2014	131809.02	141.43	932	3.95
4	2015	130942.78	140.95	929	3.96
5	2016	132113.37	142.06	930	3.96
6	2017	133103.91	143.12	930	3.94
7	2018	204522.26	139.80	1463	3.97
8	2020	129103.96	139.42	926	3.98
9	2022	131477.78	141.68	928	3.97

11.

```sql
SELECT Outlet_Size,
CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST((SUM(Sales)*100 / SUM(SUM(Sales)) OVER()) AS DECIMAL(10,2)) AS Sales_percentage
FROM blinkit_data
GROUP BY Outlet_Size
ORDER BY Total_Sales
```

121 % ▼ ◄

	Outlet_Size	Total_Sales	Sales_percentage
1	High	248991.59	20.72
2	Small	444794.17	37.01
3	Medium	507895.74	42.27

SQLQuery3.sql - D...O6Q163\Admin (62))* ⊕ ✕ SQLQuery2.sql - D...O6Q163\Admin (61))*   SQLQuery1.sql - D...O6Q163\Admin (76))*

12.

```sql
SELECT Outlet_Location_Type,
CAST(SUM(Sales) AS DECIMAL(10,2))AS Total_Sales,
CAST((SUM(Sales)*100 / SUM(SUM(Sales)) OVER()) AS DECIMAL(10,2)) AS Sales_percentage,
CAST(AVG(Sales) AS DECIMAL(10,2)) AS Avg_sales,
COUNT(*) AS Total_Items,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Location_Type
ORDER BY Total_Sales
```

121 % ▼ ◄

	Outlet_Location_Type	Total_Sales	Sales_percentage	Avg_sales	Total_Items	Avg_Rating
1	Tier 1	336397.81	27.99	140.87	2388	3.98
2	Tier 2	393150.65	32.72	141.17	2785	3.96
3	Tier 3	472133.03	39.29	140.94	3350	3.96

13.

```sql
SELECT Outlet_Type,
CAST(SUM(Sales) AS DECIMAL(10,2))AS Total_Sales,
CAST((SUM(Sales)*100 / SUM(SUM(Sales)) OVER()) AS DECIMAL(10,2)) AS Sales_percentage,
CAST(AVG(Sales) AS DECIMAL(10,2)) AS Avg_sales,
COUNT(*) AS Total_Items,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Type
ORDER BY Total_Sales
```

121 %

▦ Results 📇 Messages

	Outlet_Type	Total_Sales	Sales_percentage	Avg_sales	Total_Items	Avg_Rating
1	Supermarket Type3	130714.67	10.88	139.80	935	3.95
2	Supermarket Type2	131477.78	10.94	141.68	928	3.97
3	Grocery Store	151939.15	12.64	140.29	1083	3.99
4	Supermarket Type1	787549.89	65.54	141.21	5577	3.96