BlinkIt

PERFORMANCE ANALYSIS

DATASET SOURCE: Kaggle.com

MYSQL QUERIES FOR DATA CLEANING AND CALCULATIONS

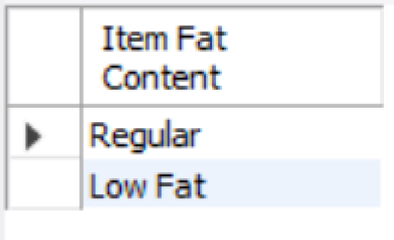
#To check / tally number of rows from imported data

```
SELECT * FROM blinkitdb.`blinkit grocery data`;

SELECT COUNT(*) FROM blinkitdb.`blinkit grocery data`;
```

#Normalise text

```
update blinkitdb.`blinkit grocery 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;
```

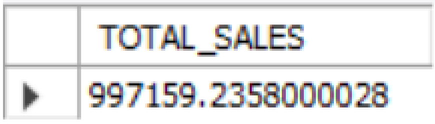
```
Select distinct(`Item Fat Content`) from blinkitdb.`blinkit grocery data`;
```



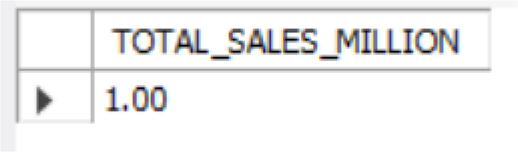
	Item Fat Content
▶	Regular
	Low Fat

#Calculate Total sales and change value to Million and 2 decimals

```
select sum(Sales) AS TOTAL_SALES from blinkitdb.`blinkit grocery data`;

select CAST(sum(Sales)/1000000 AS DECIMAL(10,2)) AS TOTAL_SALES_MILLION from
blinkitdb.`blinkit grocery data`;
```



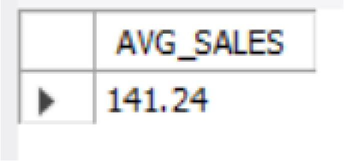
	TOTAL_SALES
▶	997159.2358000028



	TOTAL_SALES_MILLION
▶	1.00

#Average Sales

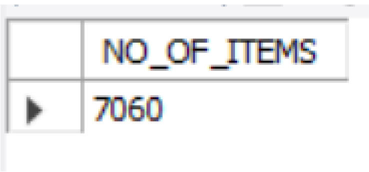
```
select CAST(AVG(Sales) AS DECIMAL(10,2)) AS AVG_SALES from blinkitdb.`blinkit grocery
data`;
```



	AVG_SALES
▶	141.24

#Count number of Items

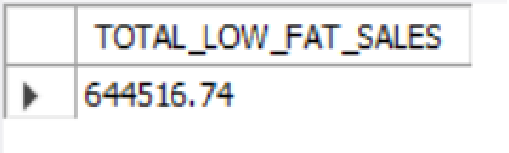
```
Select count(*) AS NO_OF_ITEMS from blinkitdb.`blinkit grocery data`;
```



	NO_OF_ITEMS
▶	7060

#Total Sales for low fat items

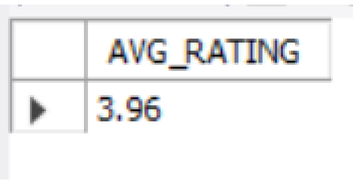
```
select cast(sum(Sales) AS DECIMAL(10,2)) AS TOTAL_LOW_FAT_SALES from blinkitdb.`blinkit
grocery data` WHERE `Item Fat Content` = 'Low Fat';
```



	TOTAL_LOW_FAT_SALES
▶	644516.74

#Average Rating

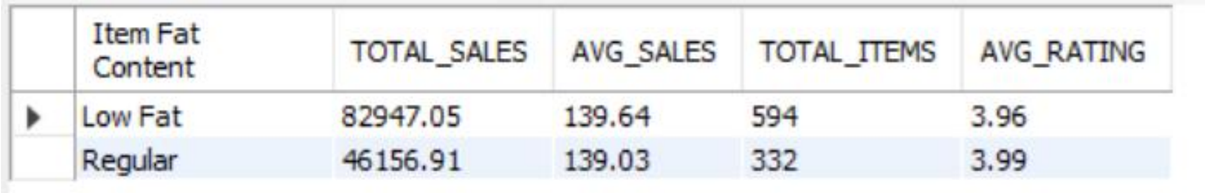
```
SELECT CAST(AVG(Rating) as DECIMAL(10,2)) AS AVG_RATING from blinkitdb.`blinkit grocery
data`;
```



	AVG_RATING
▶	3.96

#Total Sales by Fat Content

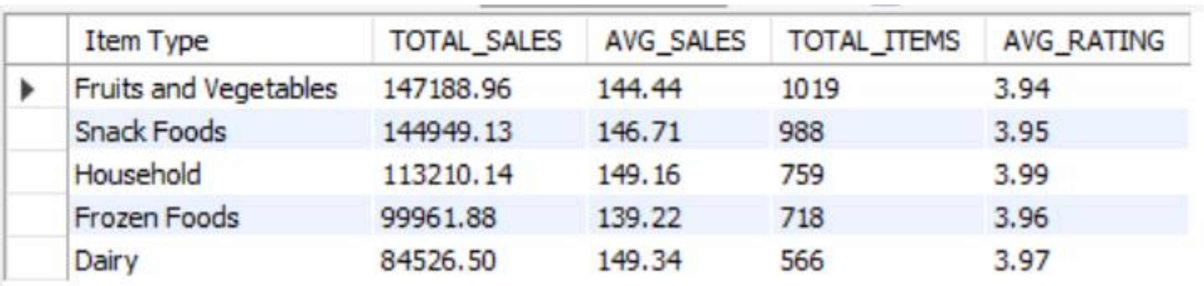
```
select `Item Fat Content`, CAST(SUM(Sales) AS DECIMAL(10,2)) AS TOTAL_SALES ,

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 blinkitdb.`blinkit grocery data`
WHERE `Outlet Establishment Year`=2020 group by `Item Fat Content`

order by TOTAL_SALES desc;
```



	Item Fat Content	TOTAL_SALES	AVG_SALES	TOTAL_ITEMS	AVG_RATING
▶	Low Fat	82947.05	139.64	594	3.96
	Regular	46156.91	139.03	332	3.99

#Total Sales by Item Type

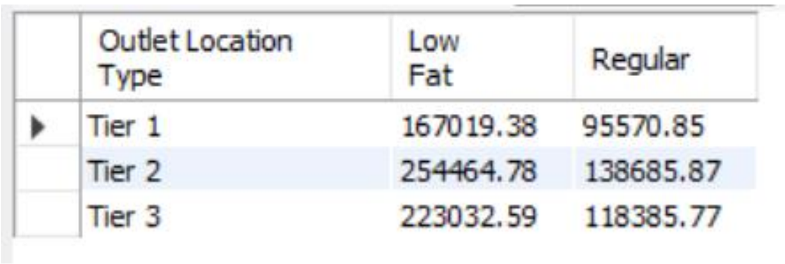
```
select ` Item Type` ,  CAST(SUM(Sales) AS DECIMAL(10,2)) AS TOTAL_SALES ,

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 blinkitdb.` blinkit grocery data`

group by ` Item Type` order by TOTAL_SALES desc  LIMIT 5;
```



	Item Type	TOTAL_SALES	AVG_SALES	TOTAL_ITEMS	AVG_RATING
▶	Fruits and Vegetables	147188.96	144.44	1019	3.94
	Snack Foods	144949.13	146.71	988	3.95
	Household	113210.14	149.16	759	3.99
	Frozen Foods	99961.88	139.22	718	3.96
	Dairy	84526.50	149.34	566	3.97

#Fat Content by Outlet for Total Sales (3 columns- outlet type, low fat, regular)

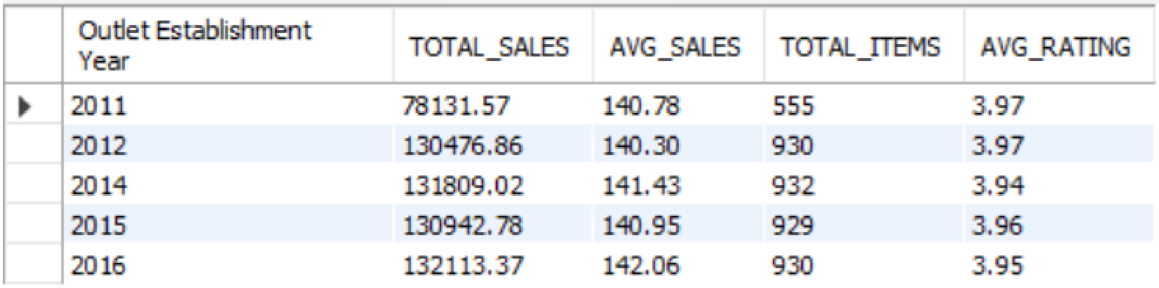
```
SELECT ` Outlet Location Type` ,

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 blinkitdb.` blinkit grocery data`

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



	Outlet Location Type	Low Fat	Regular
▶	Tier 1	167019.38	95570.85
	Tier 2	254464.78	138685.87
	Tier 3	223032.59	118385.77

#Total Sales by Outlet Establishment

```
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 blinkitdb.` blinkit grocery data`

group by ` Outlet Establishment Year` order by ` Outlet Establishment Year`   LIMIT 5;
```



	Outlet Establishment Year	TOTAL_SALES	AVG_SALES	TOTAL_ITEMS	AVG_RATING
▶	2011	78131.57	140.78	555	3.97
	2012	130476.86	140.30	930	3.97
	2014	131809.02	141.43	932	3.94
	2015	130942.78	140.95	929	3.96
	2016	132113.37	142.06	930	3.95

#Percentage Sales by Outlet Size

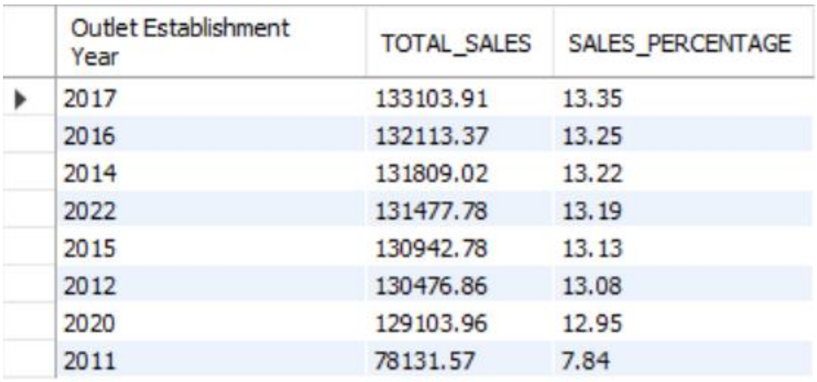
SELECT `Outlet Establishment Year`,

CAST(SUM(Sales) AS DECIMAL(10,2)) AS TOTAL_SALES,

CAST(SUM(Sales) * 100.0 / SUM(SUM(Sales)) OVER () AS DECIMAL(10,2)) AS SALES_PERCENTAGE

FROM blinkitdb.`blinkit grocery data`

GROUP BY `Outlet Establishment Year` ORDER BY TOTAL_SALES DESC;

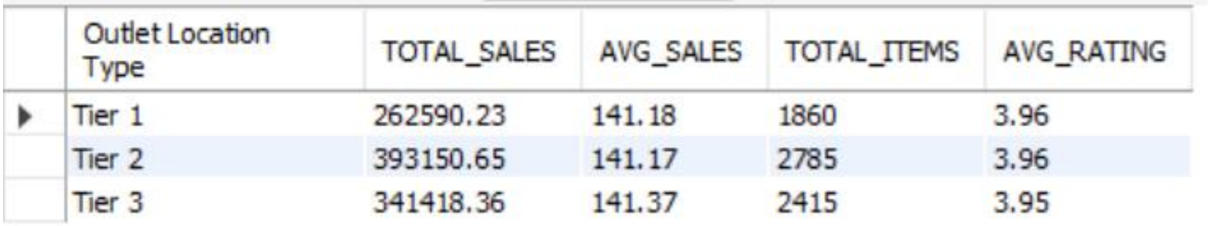


	Outlet Establishment Year	TOTAL_SALES	SALES_PERCENTAGE
▶	2017	133103.91	13.35
	2016	132113.37	13.25
	2014	131809.02	13.22
	2022	131477.78	13.19
	2015	130942.78	13.13
	2012	130476.86	13.08
	2020	129103.96	12.95
	2011	78131.57	7.84

#Sales by Outlet Location

Select `Outlet Location Type`,

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 blinkitdb.`blinkit grocery data` GROUP BY `Outlet Location Type` ORDER BY `Outlet Location Type`;



	Outlet Location Type	TOTAL_SALES	AVG_SALES	TOTAL_ITEMS	AVG_RATING
▶	Tier 1	262590.23	141.18	1860	3.96
	Tier 2	393150.65	141.17	2785	3.96
	Tier 3	341418.36	141.37	2415	3.95

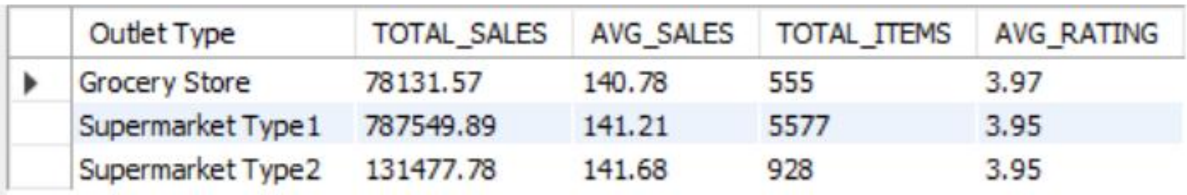
#All metrics by Outlet Type

Select `Outlet Type`, 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 blinkitdb.`blinkit grocery data`

GROUP BY `Outlet Type` ORDER BY `Outlet Type`;



	Outlet Type	TOTAL_SALES	AVG_SALES	TOTAL_ITEMS	AVG_RATING
▶	Grocery Store	78131.57	140.78	555	3.97
	Supermarket Type1	787549.89	141.21	5577	3.95
	Supermarket Type2	131477.78	141.68	928	3.95