

BLINKIT APP ANALYSIS SQL QUERIES

OBJECTIVE:

To utilize Oracle SQL queries for in-depth analysis of Blinkit's operations, focusing on order trends, delivery performance, and inventory management. This analysis aims to optimize data retrieval, identify key business insights, and enhance decision-making processes. By extracting and interpreting critical metrics, the goal is to improve operational efficiency, streamline supply chain management, and drive business growth.

Display all values from table:

```
SELECT
```

```
*
```

```
FROM
```

```
blinkit;
```

ITEMFATCONTENTS	ITEMIDENTIFIER	ITEMTYPE	OUTLETESTABLISHMENTYEAR	OUTLETIDENTIFIER	OUTLETLOCATIONTYPE
Regular	FDE56	Fruits and Vegetables	2020	OUT017	Tier 2
Regular	FDI07	Meat	2020	OUT017	Tier 2
Regular	FDU15	Meat	2020	OUT017	Tier 2
Regular	FDU27	Meat	2020	OUT017	Tier 2
Regular	FDN39	Meat	2020	OUT017	Tier 2
Regular	FDV10	Snack Foods	2020	OUT017	Tier 2
Regular	FDO45	Snack Foods	2020	OUT017	Tier 2
Regular	FDE23	Starchy Foods	2020	OUT017	Tier 2

1. Find total sales from the table:

```
SELECT
```

```
round(SUM(totalsales) / 1000000, 3) AS total_sales
```

```
FROM
```

```
blinkit;
```

```
TOTAL_SALES
```

```
1.202
```

2.Average Order Value:

```
select sum(total_price)/count(DISTINCT(order_id)) as Avg_Order from pizza_sales;
```

AVG_ORDER
38.23700044759929

3. Find no of rows in table:

```
SELECT
    COUNT (*)
FROM
    blinkit;
```

COUNT(*)
8523

4. Find total sales from the table:

```
SELECT
    round(SUM(totalsales) / 1000000, 3) AS total_sales
FROM
    blinkit;
```

TOTAL_SALES
1.202

Data cleaning in the table:

```
UPDATE blinkit
SET
    itemfatcontents = 'Regular'
WHERE
    itemfatcontents = 'reg';

UPDATE blinkit
SET
    itemfatcontents = 'Low Fat'
WHERE
    itemfatcontents = 'low fat';
```

5. Identify unique values form the table:

```

SELECT DISTINCT
    ( itemfatcontents )
FROM
    blinkit;

```

ITEMFATCONTENTS
Regular
Low Fat

6. Avg revenue per sale:

```

SELECT
    round(AVG(totalsales), 0) AS avg_sales
FROM
    blinkit;

```

AVG_SALES
141

7. Find total values for low-fat:

```

select
    trunc(SUM(totalsales), 1)
FROM Blinkit
WHERE
    itemfatcontents = 'Low Fat';

```

TRUNC(SUM(TOTALSALES),1)
776319.6

```

SELECT
    trunc(SUM(totalsales))
FROM
    blinkit
WHERE
    outletestablishmentyear = 2020;

```

```
TRUNC(SUM(TOTALSALES))
129103
```

8. Find average rating:

```
SELECT
    trunc(AVG(rating), 1)
FROM
    blinkit;
```

```
TRUNC(AVG(RATING),1)
3.9
```

9. Total sales by fat content:

```
SELECT
    itemfatcontents,
    round(SUM(totalsales)) AS sales,
    trunc(AVG(totalsales)) AS avg_sales,
    COUNT(itemfatcontents) AS count
FROM
    blinkit
WHERE
    outletestablishmentyear = 2022 group BY
    itemfatcontents
ORDER BY
    sales DESC;
```

ITEMFATCONTENTS	SALES	AVG_SALES	COUNT
Low Fat	84845	141	598
Regular	46633	141	330

10. Total sales by item type:

```

SELECT
    itemtype,
    trunc(SUM(totalsales)) AS sales
FROM
    blinkit
WHERE
    itemtype LIKE 'F%'
    OR outletlocationtype = 'Tier2'
GROUP BY
    itemtype
ORDER BY
    sales DESC;

```

ITEMTYPE	SALES
Fruits and Vegetables	178124
Frozen Foods	118558

11. Fat content outlet for total sales:

```

SELECT
    itemfatcontents,
    outletlocationtype,
    round(SUM(totalsales)) AS sales
FROM
    blinkit
GROUP BY
    itemfatcontents,
    outletlocationtype
ORDER BY
    outletlocationtype DESC;

```

ITEMFATCON...	OUTLETLOCATIONTYPE	SALES
Low Fat	Tier 3	306807
Regular	Tier 3	165326
Low Fat	Tier 2	254465
Regular	Tier 2	138686
Low Fat	Tier 1	215048
Regular	Tier 1	121350

12. Total sales by outlet establishment:

```

SELECT
    outletestablishmentyear AS year,
    round(SUM(totalsales)) AS sales,
    trunc(AVG(totalsales)) avg_sales,
    COUNT(*) AS totalcount
FROM
    blinkit
GROUP BY
    outletestablishmentyear
ORDER BY
    year ASC,
    sales DESC;

```

YEAR	SALES	AVG_SALES	TOTALCOUNT
1998	204522	139	1463
2000	131809	141	932
2010	132113	142	930
2011	78132	140	555
2012	130477	140	930
2015	130943	140	929
2017	133104	143	930
2020	129104	139	926
2022	131478	141	928

13. Percentage of sales outlet size:

```

SELECT
    outletsize,
    round(SUM(totalsales)) AS sales,
    round((SUM(totalsales) * 100) / SUM(SUM(totalsales)) OVER()) AS percentage_sales
FROM
    blinkit
GROUP BY
    outletsize;

```

OUTLETSIZE	SALES	PERCENTAGE_SALES
Medium	507896	42
High	248992	21
Small	444794	37

14. Sales by outlet location:

```

SELECT
    outletlocationtype,
    trunc(SUM(totalsales)) AS sales
FROM
    blinkit
GROUP BY
    outletlocationtype
ORDER BY
    sales DESC;

```

OUTLETLOCATIONTYPE	SALES
Tier 3	472133
Tier 2	393150
Tier 1	336397

15. All metric from outlet type:

```

SELECT
    outletlocationtype,
    round(SUM(totalsales)) AS sales,
    ( trunc(AVG(totalsales)) ) AS avg_sales,
    COUNT(*) AS total_count,
    round((SUM(totalsales) * 100) / SUM(SUM(totalsales)) OVER(),2) AS percentage_sales
FROM
    blinkit
GROUP BY
    outletlocationtype
ORDER BY
    sales DESC;

```

OUTLETLOCATIONTYPE	SALES	AVG_SALES	TOTAL_COUNT	PERCENTAGE_SALES
Tier 3	472133	140	3350	39.29
Tier 2	393151	141	2785	32.72
Tier 1	336398	140	2388	27.99