

Cleaning Process followed for blinkit_dataset

- 1) Datatype changed from string to date in blinkit_customer_feedback table

```
ALTER TABLE blinkit_customer_feedback
ADD COLUMN feedback_date_converted DATE;
set SQL_safe_updates=0;
UPDATE blinkit_customer_feedback
SET feedback_date_converted = STR_TO_DATE(feedback_date, '%Y-%m-%d');
alter table blinkit_customer_feedback drop column feedback_date;
alter table blinkit_customer_feedback rename column feedback_date_converted to feedback_date;
```

- 2) Checking for null values in every column in every table

```
select count(*) from blinkit_customers where avg_order_value is null
```

- 3) Blank column in blinkit_delivery_performance table is replaced with '-'

```
update blinkit_delivery_performance
set reasons_if_delayed='-'
where reasons_if_delayed='' and delivery_status='On Time';
```

- 4) Deleting unnecessary columns in tables

```
ALTER TABLE blinkit_customers
DROP COLUMN email,
DROP COLUMN phone,
DROP COLUMN address;
```

- 5) Updating columns with primary and foreign keys to the tables to provide interconnection

```
1 • alter table blinkit_products
2   add constraint primary key(product_id);
3
4 • alter table blinkit_orders
5   add constraint primary key(order_id);
6
7 • alter table blinkit_customers
8   add constraint primary key(customer_id);
9
10 • alter table blinkit_delivery_performance
11   add constraint primary key(delivery_partner_id);

13 • alter table blinkit_inventorynew
14   add constraint foreign key(product_id) references blinkit_products(product_id) on delete cascade on update cascade;
15
16 • alter table blinkit_order_items
17   add constraint foreign key(product_id) references blinkit_products(product_id) on delete cascade on update cascade;
18
19 • alter table blinkit_order_items
20   add constraint foreign key(order_id) references blinkit_orders(order_id) on delete cascade on update cascade;
21
22 • alter table blinkit_orders
23   add constraint foreign key(customer_id) references blinkit_customers(customer_id) on delete cascade on update cascade;
24
25 • alter table blinkit_customer_feedback
26   add constraint foreign key(customer_id) references blinkit_customers(customer_id) on delete cascade on update cascade;
27
28 • alter table blinkit_customer_feedback
29   add constraint foreign key(order_id) references blinkit_orders(order_id) on delete cascade on update cascade;

31 • alter table blinkit_delivery_performance
32   add constraint foreign key(order_id) references blinkit_orders(order_id) on delete cascade on update cascade;
33
34 • alter table blinkit_orders
35   add constraint foreign key(delivery_partner_id)
36   references blinkit_delivery_performance(delivery_partner_id) on delete cascade on update cascade;
```

Insights and KPI's of Blinkit_dataset

- 1) What products blinkit selling (unique products)?

S_NO	product_name	S_NO	product_name
1	Onions	23	Iced Tea
2	Potatoes	24	Mango Drink
3	Tomatoes	25	Lemonade
4	Bananas	26	Frozen Pizza
5	Mangoes	27	Instant Noodles
6	Carrots	28	Frozen Veget...
7	Spinach	29	Ice Cream
8	Cheese	30	Frozen Biryani
9	Curd	31	Rice
10	Butter	32	Pulses
11	Eggs	33	Sugar
12	Bread	34	Salt
13	Milk	35	Wheat Flour
14	Cereal	36	Dish Soap
15	Biscuits	37	Toilet Cleaner
16	Nuts	38	Detergent
17	Chips	39	Shampoo
18	Popcorn	40	Toothpaste
19	Cookies	41	Lotion
20	Chocolates	42	Soap
21	Orange Juice	43	Baby Wipes
22	Cola	44	Baby Food
23	Iced Tea	45	Diapers
24	Mango Drink	46	Cat Food
		47	Pet Treats
		48	Dog Food
		49	Pain Reliever
		50	Cough Syrup
		51	Vitamins

- 2) Find top 10 most sold-out products?

```
select product_name,count(blinkit_order_items.product_id) Sales
from blinkit_order_items
join
blinkit_products
on blinkit_order_items.product_id=blinkit_products.product_id
group by blinkit_products.product_name
order by Sales desc limit 10;
```

product_name	Sales
Pet Treats	233
Toilet Cleaner	205
Cough Syrup	190
Lotion	188
Dish Soap	184
Vitamins	183
Baby Wipes	176
Cat Food	152
Pulses	137
Bread	135

- 3) Find top 10 least sold-out products?

```
select product_name,count(blinkit_order_items.product_id) Sales
from blinkit_order_items
join
blinkit_products
on blinkit_order_items.product_id=blinkit_products.product_id
group by blinkit_products.product_name
order by Sales limit 10;
```

product_name	Sales
Spinach	21
Lemonade	21
Milk	28
Instant Noodles	35
Diapers	43
Cereal	44
Bananas	45
Cookies	47
Mangoes	49
Ice Cream	50

- 4) Total sales from march 2023 to November 2024

```
select count(order_id) as Total_orders from blinkit_orders
```

count(order_id)
5000

- 5) Total revenue generated

```
select round(sum(order_total)) as Revenue_generated from blinkit_orders
```

Revenue_generated
11009308

- 6) Average order value

```
select round(sum(order_total)/count(order_id) )
as Average_order_value from blinkit_orders
```

Average_order_value
2202

7) Month wise sales count, revenue generated and profit gain

```
select monthname(o.order_date) as Month,
count(distinct o.order_id) as Sales,
round(sum(o.order_total)) as Revenue_generated,
round(sum(i.unit_price * i.quantity)) as Expenditure,
(round(sum(o.order_total))-round(sum(i.unit_price * i.quantity))) as Profit
from blinkit_orders o join blinkit_order_items i
on o.order_id=i.order_id
group by (monthname(order_date)),month(order_date)
order by (month(order_date));
```

Month	Sales	Revenue_generated	Expenditure	Profit
January	270	560424	273091	287333
February	252	545090	260130	284960
March	371	816061	351651	464410
April	479	1093100	464638	628462
May	539	1182377	534442	647935
June	480	1044303	472830	571473
July	500	1140752	489429	651323
August	536	1169667	554768	614899
September	509	1089813	508610	581203
October	501	1116073	519723	596350
November	295	635941	289474	346467
December	268	615709	253628	362081

8) Which time has highest orders and lowest orders like peak hours

```
select hour(order_date) as Peak_hours,
count(distinct order_id) as Sales
from blinkit_orders
group by hour(order_date) order by Sales desc limit 5;
```

Peak_hours	Sales
8	240
13	239
19	228
20	227
18	226

```
select hour(order_date) as weak_hours,
count(distinct order_id) as Sales
from blinkit_orders
group by hour(order_date) order by Sales asc limit 5;
```

weak_hours	Sales
9	181
11	184
3	189
12	193
21	195

9) Which day has most and least sales

```
select dayname(order_date) as weak_hours,
count(distinct order_id) as Sales
from blinkit_orders
group by dayname(order_date) order by Sales desc ;
```

weak_hours	Sales
Wednesday	744
Sunday	723
Thursday	720
Friday	715
Saturday	706
Tuesday	705
Monday	687

10) Orders delivered by delivery status

```
select distinct delivery_status,count(order_id) as total_sales
from blinkit_orders group by delivery_status;
```

delivery_status	total_sales
On Time	3470
Slightly Delayed	1037
Significantly Delayed	493

11) Payment ways done on orders

payment_method	count(distinct order_id)
Card	1285
Cash	1257
UPI	1214
Wallet	1244

12) Average delivery delay Time

<pre>SELECT round(AVG(TIMESTAMPDIFF(MINUTE, promised_delivery_time, actual_delivery_time))) AS avg_delay_minutes FROM blinkit_orders WHERE actual_delivery_time > promised_delivery_time;</pre>	<div>avg_delay_minutes</div> <div>9</div>
--	---

13) Average promised delivery time from ordered time

<pre>SELECT round(AVG(TIMESTAMPDIFF(MINUTE, order_date, promised_delivery_time))) AS avg_promised_delivery_time FROM blinkit_orders;</pre>	<div>avg_promised_delivery_time</div> <div>15</div>
--	---

14) Average delivery time from ordered to delivered

<pre>SELECT round(AVG(TIMESTAMPDIFF(MINUTE, order_date, actual_delivery_time))) AS avg_total_delivery_time FROM blinkit_orders;</pre>	<div>avg_total_delivery_time</div> <div>19</div>
---	--

15) Total profit generated

<pre>select (round(sum(o.order_total)-(sum(i.unit_price * i.quantity)))) as Total_Profit from blinkit_orders o join blinkit_order_items i on o.order_id=i.order_id;</pre>	<div>Total_Profit</div> <div>6036893</div>
---	--

16) Highest Sold-out categories and lowest sold-out categories

<pre>select p.category, count(distinct o.order_id) as Sales from blinkit_order_items o join blinkit_products p on o.product_id=p.product_id group by p.category order by Sales desc;</pre>	<table> <tr> <th>category</th><th>Sales</th></tr> <tr><td>Dairy & Breakfast</td><td>566</td></tr> <tr><td>Household Care</td><td>509</td></tr> <tr><td>Pet Care</td><td>501</td></tr> <tr><td>Fruits & Vegetables</td><td>492</td></tr> <tr><td>Snacks & Munchies</td><td>483</td></tr> <tr><td>Pharmacy</td><td>481</td></tr> <tr><td>Personal Care</td><td>454</td></tr> <tr><td>Grocery & Staples</td><td>449</td></tr> <tr><td>Cold Drinks & Juices</td><td>375</td></tr> <tr><td>Instant & Frozen Food</td><td>356</td></tr> <tr><td>Baby Care</td><td>334</td></tr> </table>	category	Sales	Dairy & Breakfast	566	Household Care	509	Pet Care	501	Fruits & Vegetables	492	Snacks & Munchies	483	Pharmacy	481	Personal Care	454	Grocery & Staples	449	Cold Drinks & Juices	375	Instant & Frozen Food	356	Baby Care	334
category	Sales																								
Dairy & Breakfast	566																								
Household Care	509																								
Pet Care	501																								
Fruits & Vegetables	492																								
Snacks & Munchies	483																								
Pharmacy	481																								
Personal Care	454																								
Grocery & Staples	449																								
Cold Drinks & Juices	375																								
Instant & Frozen Food	356																								
Baby Care	334																								

17) Profit generated by campaigns

<pre>select campaign_name, (sum(revenue_generated)-sum(spend)) as Profit from blinkit_marketing_performance group by campaign_name order by Profit desc;</pre>	<table> <tr> <th>campaign_name</th><th>Profit</th></tr> <tr><td>Referral Program</td><td>1873357.0900000002</td></tr> <tr><td>Email Campaign</td><td>1791055.5499999993</td></tr> <tr><td>Weekend Special</td><td>1771840.54000000024</td></tr> <tr><td>New User Discount</td><td>1770405.35999999957</td></tr> <tr><td>App Push Notification</td><td>1765380.79999999952</td></tr> <tr><td>Membership Drive</td><td>1734881.45000000011</td></tr> <tr><td>Category Promotion</td><td>1731872.36999999987</td></tr> <tr><td>Flash Sale</td><td>1724399.19999999958</td></tr> <tr><td>Festival Offer</td><td>1710376.77</td></tr> </table>	campaign_name	Profit	Referral Program	1873357.0900000002	Email Campaign	1791055.5499999993	Weekend Special	1771840.54000000024	New User Discount	1770405.35999999957	App Push Notification	1765380.79999999952	Membership Drive	1734881.45000000011	Category Promotion	1731872.36999999987	Flash Sale	1724399.19999999958	Festival Offer	1710376.77
campaign_name	Profit																				
Referral Program	1873357.0900000002																				
Email Campaign	1791055.5499999993																				
Weekend Special	1771840.54000000024																				
New User Discount	1770405.35999999957																				
App Push Notification	1765380.79999999952																				
Membership Drive	1734881.45000000011																				
Category Promotion	1731872.36999999987																				
Flash Sale	1724399.19999999958																				
Festival Offer	1710376.77																				

<pre>SELECT campaign_name, SUM(spend) AS total_spent, SUM(revenue_generated) AS total_revenue, SUM(revenue_generated) - SUM(spend) AS profit, ROUND(SUM(revenue_generated) / NULLIF(SUM(spend), 0), 2) AS overall_roas FROM blinkit_marketing_performance GROUP BY campaign_name ORDER BY profit DESC;</pre>	
--	--

campaign_name	total_spent	total_revenue	profit	overall_roas
Referral Program	1818025.50999999995	3691382.60000000015	1873357.0900000002	2.03
Email Campaign	1810729.67	3601785.21999999993	1791055.5499999993	1.99
Weekend Special	1799611.28999999982	3571451.83000000005	1771840.54000000024	1.98
New User Discount	1833454.81000000024	3603860.1699999998	1770405.35999999957	1.97
App Push Notification	1788989.20000000016	3554369.99999999967	1765380.79999999952	1.99
Membership Drive	1790069.80000000003	3524951.25000000014	1734881.45000000011	1.97
Category Promotion	1850583.00000000014	3582455.37	1731872.36999999987	1.94
Flash Sale	1831687.82000000015	3556087.0199999997	1724399.19999999958	1.94
Festival Offer	1796687.14000000001	3507063.91	1710376.77	1.95

ROAS=Return on Advertising Spend (ROAS=Revenue from Ads/Ad Spend)

18) Customers registered from march 2023 to November 2024

```
select min(registration_date) as Date_from,
count(distinct customer_id) as cutomers_registered,
max(registration_date)as date_upto from blinkit_customers;
```

Date_from	cutomers_registered	date_upto
2023-03-16 00:00:00	2500	2024-11-04 00:00:00

19) Sales area wise

```
select ifnull(a.`State/UT`, 'Data Missing') as Place, count(distinct o.order_id) as Total_sales
from blinkit_customers c join blinkit_orders o
on c.customer_id=o.customer_id
left join area_to_state1 a
on a.Area=c.area
group by a.`State/UT` order by Total_sales desc limit 10;
```

Place	Total_sales
Data Missing	861
Uttar Pradesh	529
Andhra Pradesh	463
Maharashtra	410
West Bengal	335
Tamil Nadu	312
Bihar	291
Madhya Pradesh	246
Gujarat	204
Rajasthan	155

20) Feedback Rating category wise

```
SELECT
feedback_category,
SUM(CASE WHEN sentiment = 'Positive' THEN 1 ELSE 0 END) AS Positives,
SUM(CASE WHEN sentiment = 'Neutral' THEN 1 ELSE 0 END) AS Neutrals,
SUM(CASE WHEN sentiment = 'Negative' THEN 1 ELSE 0 END) AS Negatives
FROM blinkit_customer_feedback
GROUP BY feedback_category;
```

feedback_category	Positives	Neutrals	Negatives
Delivery	420	446	405
App Experience	404	421	388
Customer Service	418	431	417
Product Quality	378	440	432

21) People given negative feedback

```
select feedback_text, count(feedback_text) people_mentioned
from blinkit_customer_feedback
group by feedback_text order by people_mentioned desc;
```

feedback_text	people_mentioned
Delivery was late and I was unhappy.	182
Product was damaged during delivery.	178
Not worth the price I paid.	175
I had a bad experience.	169
The order was incorrect.	166
Very disappointed with the quality.	164
Taste was not as expected.	163
Items were missing from my order.	158
The packaging was poor.	146
Customer service was not helpful.	141

22) Margin percentage by product category

```
select category, round(avg(margin_percentage)) as avg_margin_percentage
from blinkit_products group by category
order by avg_margin_percentage desc;
```

category	avg_margin_percentage
Instant & Frozen Food	40
Personal Care	35
Pet Care	35
Snacks & Munchies	35
Baby Care	30
Cold Drinks & Juices	30
Fruits & Vegetables	25
Household Care	25
Dairy & Breakfast	20
Pharmacy	20
Grocery & Staples	15

23) Category wise products maximum shelf-life days

```
select category,max(shelf_life_days)
as Maximum_shelf_life_days
from blinkit_products group by category
order by max(shelf_life_days) desc
```

category	Maximum_shelf_life_days
Baby Care	365
Personal Care	365
Household Care	365
Pharmacy	365
Pet Care	365
Grocery & Staples	365
Instant & Frozen Food	180
Cold Drinks & Juices	180
Snacks & Munchies	90
Dairy & Breakfast	7
Fruits & Vegetables	3

24) Most common delivery delay reasons

```
select distinct reasons_if_delayed
from blinkit_delivery_performance
```

reasons_if_delayed
Traffic
-

25) Average customer rating

```
select round(avg(rating),1) as Average_customer_rating
from blinkit_customer_feedback;
```

Average_customer_rating
3.3

26) Customer ratings

```
select rating,count(rating) as people_count
from blinkit_customer_feedback
group by rating order by rating desc;
```

rating	people_count
5	816
4	1708
3	1398
2	538
1	540

27) Stock received vs damaged

```
select p.product_name,p.category,
sum(i.stock_received) as received_stock,
sum(i.damaged_stock) as damaged_stock
from blinkit_products p join
blinkit_inventorynew i
on p.product_id=i.product_id
group by p.product_name,p.category
order by damaged_stock desc;
```

product_name	category	received_stock	damaged_stock
Pet Treats	Pet Care	1311	51
Pulses	Grocery & Staples	970	47
Dish Soap	Household Care	945	46
Lotion	Personal Care	1038	46
Toilet Cleaner	Household Care	950	38
Detergent	Household Care	664	35
Vitamins	Pharmacy	1140	34
Cat Food	Pet Care	723	34
Bread	Dairy & Breakfast	741	34
Orange Juice	Cold Drinks & Juices	764	33
Onions	Fruits & Vegetables	649	32
Wheat Flour	Grocery & Staples	618	32
Potatoes	Fruits & Vegetables	800	31
Frozen Biryani	Instant & Frozen ...	778	29
Pain Reliever	Pharmacy	670	28
Biscuits	Snacks & Munchies	782	26
Cough Syrup	Pharmacy	1025	26
Eggs	Dairy & Breakfast	767	26
Sugar	Grocery & Staples	579	26
Cola	Cold Drinks & Juices	756	25
Baby Wipes	Baby Care	695	24

Popcorn	Snacks & Munchies	571	22
Dog Food	Pet Care	678	21
Mango Drink	Cold Drinks & Juices	457	21
Cheese	Dairy & Breakfast	535	20
Butter	Dairy & Breakfast	425	19
Toothpaste	Personal Care	464	18
Soap	Personal Care	528	18
Nuts	Snacks & Munchies	437	17
Frozen Veget...	Instant & Frozen ...	352	17
Curd	Dairy & Breakfast	457	17
Baby Food	Baby Care	448	16
Chocolates	Snacks & Munchies	449	15
Shampoo	Personal Care	481	15
Frozen Pizza	Instant & Frozen ...	553	14
Instant Noodles	Instant & Frozen ...	213	14
Chips	Snacks & Munchies	425	14
Bananas	Fruits & Vegetables	318	13
Tomatoes	Fruits & Vegetables	326	13
Carrots	Fruits & Vegetables	563	13
Rice	Grocery & Staples	355	12
Ice Cream	Instant & Frozen ...	350	11

Salt	Grocery & Staples	347	11
Diapers	Baby Care	210	11
Cookies	Snacks & Munchies	331	11
Iced Tea	Cold Drinks & Juices	341	10
Mangoes	Fruits & Vegetables	237	8
Cereal	Dairy & Breakfast	227	7
Spinach	Fruits & Vegetables	117	4
Milk	Dairy & Breakfast	111	2
Lemonade	Cold Drinks & Juices	108	2

28) Efficient way of marketing

```
select channel, round(sum(revenue_generated)-sum(spend)) as Profit
from blinkit_marketing_performance
group by channel order by profit desc;
```

channel	Profit
Email	4191844
SMS	3940042
Social Media	3880052
App	3861632