SQL Project: FoodDelivery Insights

Performed Ad-Hoc Analysis Using SQL Queries to Extract Actionable Business Insights from Food Delivery Data

```
select * from orders;
```

121 %

Results | Messages

	Order_id	Customer_code	Placed_at	Restaurant_id	Cuisine	Order_status	Promo_code_Name
1	OF1900191801	UFDDN1991918XUY1	2025-01-01 15:30:20.000	KMKMH6787	Lebanese	Delivered	Tasty50
2	OF1900191802	UFDDN1991918XUY1	2025-01-02 12:15:45.000	LEBANESE2	Lebanese	Delivered	NULL
3	OF1900191803	UFDDN1991918XUY1	2025-01-10 18:45:30.000	PIZZA123	Italian	Cancelled	HUNGRY20
4	OF1900191804	UFDDN1991918XUY1	2025-01-15 19:20:15.000	ITALIAN2	Italian	Delivered	NULL
5	OF1900191805	UFDDN1991918XUY1	2025-01-20 11:30:00.000	BURGER99	American	Delivered	NULL
6	OF1900191806	ABC1234567890XYZ	2025-01-01 08:45:00.000	AMERICAN2	American	Delivered	NEWUSER
7	OF1900191807	ABC1234567890XYZ	2025-01-05 13:20:00.000	TACO789	Mexican	Delivered	NULL
8	OF1900191808	DEF9876543210XYZ	2025-01-02 09:15:00.000	MEXICAN2	Mexican	Delivered	FIRSTORDER
9	OF1900191809	GHI5678901234XYZ	2025-01-03 14:30:00.000	SUSHI456	Japanese	Delivered	NEWUSER
10	OF1900191810	JKL3456789012XYZ	2025-01-04 12:00:00.000	JAPANESE2	Japanese	Delivered	FIRSTORDER
11	OF1900191811	MNO7890123456XYZ	2025-01-05 19:45:00.000	KMKMH6787	Lebanese	Delivered	NULL
12	OF1900191812	PQR1234567890ABC	2025-01-06 11:30:00.000	LEBANESE2	Lebanese	Delivered	NEWUSER
13	OF1900191813	STU9876543210ABC	2025-01-07 13:15:00.000	PIZZA123	Italian	Delivered	NULL
14	OF1900191814	VWX5678901234ABC	2025-01-08 18:00:00.000	ITALIAN2	Italian	Delivered	FIRSTORDER
15	OF1900191815	YZA3456789012ABC	2025-01-09 12:45:00.000	BURGER99	American	Delivered	NULL
16	OF1900191816	BCD7890123456ABC	2025-01-10 20:15:00.000	AMERICAN2	American	Delivered	NEWUSER
17	OF1900191817	EFG1234567890DEF	2025-01-11 09:30:00.000	TACO789	Mexican	Delivered	NULL
18	OF1900191818	HIJ9876543210DEF	2025-01-12 14:45:00.000	MEXICAN2	Mexican	Delivered	FIRSTORDER
19	OF1900191819	KLM5678901234DEF	2025-01-13 17:30:00.000	SUSHI456	Japanese	Delivered	NULL
20	OF1900191820	NOP3456789012DEF	2025-01-14 12:15:00.000	JAPANESE2	Japanese	Delivered	NULL
21	OF1900191821	QRS7890123456DEF	2025-01-15 19:00:00.000	KMKMH6787	Lebanese	Delivered	NEWUSER
22	OF1900191822	TUV1234567890GHI	2025-01-16 10:45:00.000	LEBANESE2	Lebanese	Delivered	NULL
23	OF1900191823	WXY9876543210GHI	2025-01-17 15:30:00.000	PIZZA123	Italian	Delivered	FIRSTORDER
24	OF1900191824	ZAB5678901234GHI	2025-01-18 18:15:00.000	ITALIAN2	Italian	Delivered	NULL
25	OF1900191825	CDE3456789012GHI	2025-01-19 11:00:00.000	BURGER99	American	Delivered	NULL
26	OF1900191826	FGH7890123456GHI	2025-01-20 20:45:00.000	AMERICAN2	American	Delivered	NEWUSER
27	OF1900191827	IJK1234567890JKL	2025-01-21 09:15:00.000	TACO789	Mexican	Delivered	NULL
28	OF1900191828	LMN9876543210JKL	2025-01-22 14:30:00.000	MEXICAN2	Mexican	Delivered	FIRSTORDER
29	OF1900191829	OPQ5678901234JKL	2025-01-23 17:45:00.000	SUSHI456	Japanese	Delivered	NULL
30	OF1900191830	RST3456789012JKL	2025-01-24 12:30:00.000	JAPANESE2	Japanese	Delivered	NULL
31	OF1900191831	UVW7890123456JKL	2025-01-25 19:15:00.000	KMKMH6787	Lebanese	Delivered	NEWUSER
32	OF1900191832	XYZ1234567890MNO	2025-01-26 10:00:00.000	LEBANESE2	Lebanese	Delivered	NULL
33	OF1900191833	ABC9876543210MNO	2025-01-27 15:15:00.000	PIZZA123	Italian	Delivered	FIRSTORDER
34	OF1900191834	DEF5678901234MNO	2025-01-28 18:30:00.000	ITALIAN2	Italian	Delivered	NULL

Q1: TOP OUTLETS BY CUISINE TYPE WITHOUT USING LIMIT AND TOP FUNCTIONS

```sql
--1- TOP outlets by cuisine type without using limit and top function
with cte as (
select Cuisine, Restaurant_id, count(*) as no_of_orders
from orders
group by Cuisine, Restaurant_id
)
select Cuisine, Restaurant_id, no_of_orders from (
select *,
row_number() over (partition by Cuisine order by no_of_orders desc) as rn
from cte) A
where rn = 1
```

121 %

Results | Messages

	Cuisine	Restaurant_id	no_of_orders
1	American	BURGER99	8
2	Italian	PIZZA123	10
3	Japanese	SUSHI456	6
4	Lebanese	KMKMH6787	10
5	Mexican	TACO789	7

Q2: FIND THE DAILY NEW CUSTOMER COUNT FROM THE LAUNCH DATE (EVERYDAY HOW MANY NEW CUSTOMERS ARE WE ACQUIRING)

```sql
--2- Find the daily new customer count from the launch date (everyday how many new customers are we acquiring)
with cte as(
select Customer_code, cast(MIN(Placed_at)as date) as first_order_date
from orders
group by customer_code
)
select first_order_date, count(*) as no_of_new_customers
from cte
group by first_order_date
order by first_order_date;
```

Results | Messages

	first_order_date	no_of_new_customers
1	2025-01-01	2
2	2025-01-02	1
3	2025-01-03	1
4	2025-01-04	1
5	2025-01-05	3
6	2025-01-06	1
7	2025-01-07	1
8	2025-01-08	1
9	2025-01-09	1
10	2025-01-10	3
11	2025-01-11	1
12	2025-01-12	1
13	2025-01-13	1
14	2025-01-14	1
15	2025-01-15	2
16	2025-01-16	1
17	2025-01-17	1
18	2025-01-18	1
19	2025-01-19	1
20	2025-01-20	2
21	2025-01-21	1
22	2025-01-22	1
23	2025-01-23	1
24	2025-01-24	1
25	2025-01-25	1
26	2025-01-26	1
27	2025-01-27	1
28	2025-01-28	1
29	2025-01-29	1
30	2025-01-30	1
31	2025-01-31	4
32	2025-02-01	2
33	2025-02-05	1
34	2025-02-10	1
35	2025-03-20	2

Q3: COUNT OF ALL THE USERS WHO WERE ACQUIRED IN JAN 2025 AND ONLY PLACED ONE ORDER IN JAN AND DID NOT PLACE ANY OTHER ORDER.

```sql
/* 3- Count of all the users who were acquired in jan 2025 and only placed one order in jan and did not place any
other order. */
with cte as(
select *,
count(*) over (partition by customer_code) as no_of_orders
from orders
where Datepart(month, Placed_at) = 1
)
select Customer_code from(
select * from cte
where no_of_orders = 1) A
where Customer_code not in (select Customer_code from orders where Datepart(month, Placed_at) != 1)
```

Results | Messages

	Customer_code
1	BCD7890123456ABC
2	DEF5678901234MNO
3	EFG1234567890DEF
4	FGH7890123456GHI
5	GHI3456789012MNO
6	GHI5678901234XYZ
7	HIJ9876543210DEF
8	IJK1234567890JKL
9	JAN_ONLY_ORDER1
10	JAN_ONLY_ORDER2
11	JKL3456789012XYZ
12	JKL7890123456MNO
13	KLM5678901234DEF
14	LMN9876543210JKL
15	MNO1234567890PQR
16	MNO7890123456XYZ
17	NOP3456789012DEF
18	OPQ5678901234JKL
19	PQR1234567890ABC
20	PQR9876543210PQR
21	QRS7890123456DEF
22	RST3456789012JKL
23	SINGLE_ORDER_JAN
24	STU5678901234PQR
25	STU9876543210ABC
26	TUV1234567890GHI
27	VWX3456789012PQR
28	VWX5678901234ABC
29	WXY9876543210GHI
30	XYZ1234567890MNO
31	YZA3456789012ABC
32	ZAB5678901234GHI

Q4: LIST ALL THE CUSTOMERS WITH NO ORDERS IN THE LAST 7 DAYS BUT ACQUIRED ONE MONTH AGO WITH THEIR FIRST ORDER ON PROMO

```sql
--4 - List all the customers with no orders in the last 7 days but acquired one month ago with their first order
--on promo
with cte as(
select Customer_code, min(placed_at) as first_order_date, max(placed_at) as latest_order_date
from orders
group by Customer_code)
select c.*, o.promo_code_name as first_order_promo from cte c
inner join orders o on c.customer_code = o.customer_code and c.first_order_date = o.placed_at
where latest_order_date < dateadd(DAY, -7, getdate())
and first_order_date <  dateadd(MONTH, -1, getdate()) and o.promo_code_name is not null
```

121 %

Results | Messages

	Customer_code	first_order_date	latest_order_date	first_order_promo
1	ABC1234567890XYZ	2025-01-01 08:45:00.000	2025-01-05 13:20:00.000	NEWUSER
2	ABC9876543210MNO	2025-01-27 15:15:00.000	2025-03-15 15:15:00.000	FIRSTORDER
3	BCD7890123456ABC	2025-01-10 20:15:00.000	2025-01-10 20:15:00.000	NEWUSER
4	DEF9876543210XYZ	2025-01-02 09:15:00.000	2025-03-02 09:15:00.000	FIRSTORDER
5	FGH7890123456GHI	2025-01-20 20:45:00.000	2025-01-20 20:45:00.000	NEWUSER
6	GHI5678901234XYZ	2025-01-03 14:30:00.000	2025-01-03 14:30:00.000	NEWUSER
7	HIJ9876543210DEF	2025-01-12 14:45:00.000	2025-01-12 14:45:00.000	FIRSTORDER
8	JAN_ONLY_ORDER1	2025-01-15 13:30:00.000	2025-01-15 13:30:00.000	NEWUSER
9	JAN_ONLY_ORDER2	2025-01-20 18:45:00.000	2025-01-20 18:45:00.000	FIRSTORDER
10	JKL3456789012XYZ	2025-01-04 12:00:00.000	2025-01-04 12:00:00.000	FIRSTORDER
11	JKL7890123456MNO	2025-01-30 20:00:00.000	2025-01-30 20:00:00.000	NEWUSER
12	LMN9876543210JKL	2025-01-22 14:30:00.000	2025-01-22 14:30:00.000	FIRSTORDER
13	NO_ORDER_LAST7_1	2025-02-01 12:15:00.000	2025-02-01 12:15:00.000	NEWUSER
14	NO_ORDER_LAST7_2	2025-02-05 19:30:00.000	2025-02-05 19:30:00.000	FIRSTORDER
15	NO_ORDER_RECENT	2025-02-10 12:30:00.000	2025-02-15 18:00:00.000	NEWUSER
16	PQR1234567890ABC	2025-01-06 11:30:00.000	2025-01-06 11:30:00.000	NEWUSER
17	PQR9876543210PQR	2025-01-31 14:00:00.000	2025-01-31 14:00:00.000	FIRSTORDER
18	PROMO_FIRST_ONLY	2025-02-01 11:45:00.000	2025-02-10 17:30:00.000	WELCOME50
19	QRS7890123456DEF	2025-01-15 19:00:00.000	2025-01-15 19:00:00.000	NEWUSER
20	SINGLE_ORDER_JAN	2025-01-10 19:00:00.000	2025-01-10 19:00:00.000	FIRSTORDER

Q5: GROWTH TEAM IS PLANNING TO CREATE A TRIGGER THAT WILL TARGET CUSTOMERS AFTER THEIR EVERY THIRD ORDER WITH A PERSONALIZED COMMUNICATION AND THEY HAVE ASKED YOU TO CREATE A QUERY FOR THIS.

```sql
--5- Growth team is planning to create a trigger that will target customers after their every third order with a
-- personalized communication and they have asked you to create a query for this.
with cte as(
select *,
row_number() over (partition by Customer_code order by placed_at) as order_number
from orders)
select * from cte
where order_number % 3 = 0 and cast(placed_at as date) = '2025-03-31'

--cast(placed_at as date) = cast(getdate()as date) FOR REAL TIME DATA
```

121 %

Results | Messages

	Order_id	Customer_code	Placed_at	Restaurant_id	Cuisine	Order_status	Promo_code_Name	order_number
1	OF1900191864	LAST_ORDER_7DAYS	2025-03-31 16:30:00.000	KMKMH6787	Lebanese	Delivered	NULL	3
2	OF1900191870	MULTI_CUISINE_CUST	2025-03-31 14:45:00.000	PIZZA123	Italian	Delivered	NULL	6

Q6: LIST CUSTOMERS WHO PLACED MORE THAN 1 ORDER AND ALL THEIR ORDERS ON A PROMO ONLY

```sql
--6.) list customers who placed more than 1 order and all their orders on a promo only
select Customer_code
from orders
group by Customer_code
having count(*) > 1 and count(*) = count(Promo_code_Name)
```

121 %

Results | Messages

	Customer_code
1	DEF9876543210XYZ
2	UVW7890123456JKL

Q7: WHAT PERCENT OF CUSTOMERS WERE ORGANICALLY AQUIRED IN JAN 2025. (PLACED THEIR FIRST ORDER WITHOUT PROMO CODE)

```sql
--7.) What percent of customers were organically aquired in jan 2025. (placed their first order without promo code)
with cte as(
select *,
row_number() over (partition by customer_code order by placed_at) as rn
from orders
where month(placed_at) = 1
)
select
COUNT(case when rn = 1 and promo_code_name is null then customer_code end)*100.0/count(distinct customer_code)
as percent_of_customers
from cte
```

121 %

Results | Messages

	percent_of_customers
1	43.902439024390