



Domino's Pizza

FOR MONTH OF JANUARY 2024

OBJECTIVE:

1. To design a relational database to capture important data that dominos generates, this will help them to track orders for efficient and accurate workflow.
2. To create Dashboard to monitor business performance.

DATA WE NEED TO COLLECT:

Customer name, Customer address, Item name, Item prize and Quantity.

TASK TO PERFORM:

The data we got have lot of repetition in various rows. To fix this we will perform Normalization to reduce redundancy.

- We will create additional tables to store above data and create identifiers to access this data.
- Then we will create a table that will contain all this data together.
- Create a query to access all relevant fields from all this tables that will help in billing and monitoring workflow.
- Create a dashboard to show daily operation, sales and workflow of January 2024.

➤ **STEP 1:**

For this we will split details of customer name and customer address and then create respective Id columns for all the tables to access this data. Now store this data details by creating new tables 'customer' and 'address'

As per dominos menu they have different categories, types and sizes. So, we will create an 'item' table to store all details of their menu data into this table.

customer table:

Field	Type	Null	Key	Default	Extra
customer_id	int(3)	NO	PRI	NULL	
customer_first_name	varchar(20)	NO		NULL	
customer_last_name	varchar(20)	NO		NULL	

address table:

Field	Type	Null	Key	Default	Extra
address_id	int(3)	NO	PRI	NULL	
address_1	varchar(50)	NO		NULL	
address_2	varchar(50)	YES		NULL	
address_city	varchar(30)	NO		NULL	
address_pincode	int(6)	NO		NULL	

Item table:

Field	Type	Null	Key	Default	Extra
item_id	varchar(10)	NO	PRI	NULL	
item_name	varchar(30)	NO		NULL	
item_category	varchar(30)	NO		NULL	
item_size	varchar(10)	NO		NULL	
item_price	decimal(5,2)	NO		NULL	
item_description	varchar(250)	YES		NULL	

➤ **STEP 2:**

Now we will create 'order' table which will contain all required data from customer, address and items table. Ensure that all the three tables have relationship with 'order' table to ease access of data.

This table will contain all the data that we needed to collect.

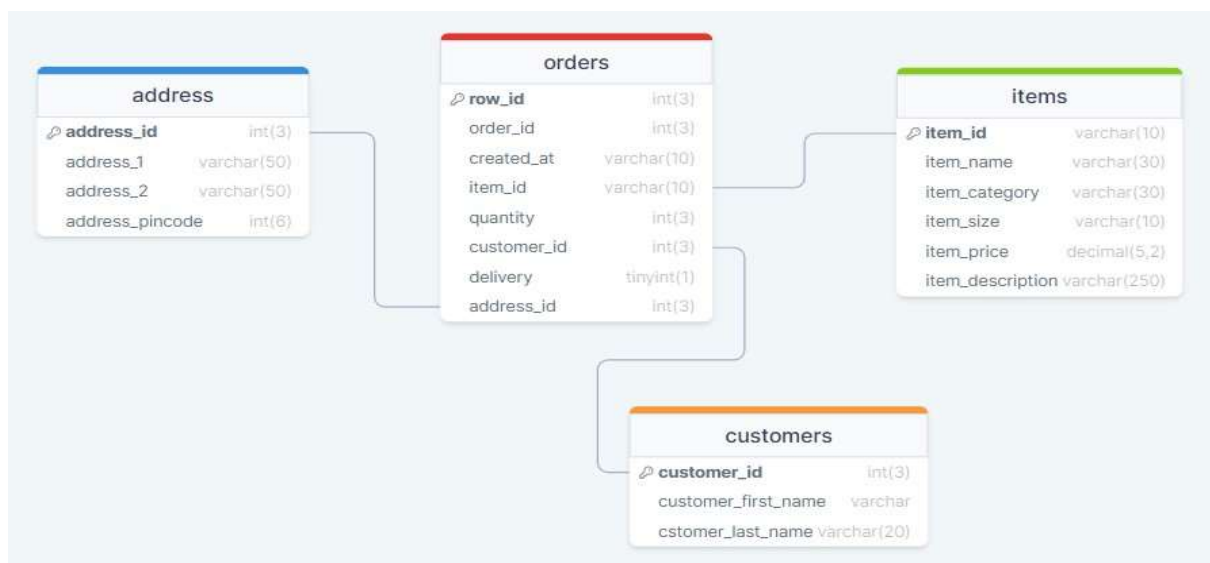
Here row_id is a primary key and created_at is current_timestamp() which will save data and time when data(rows) will be entered.

Item_id, customer_id and address_id are foreign keys connecting link to their respective primary keys.

Order table:

Field	Type	Null	Key	Default	Extra
row_id	int(3)	NO	PRI	NULL	
order_id	int(3)	NO		NULL	
created_at	timestamp	NO		current_timestamp()	
item_id	varchar(10)	NO	MUL	NULL	
quantity	int(3)	NO		NULL	
customer_id	int(3)	NO	MUL	NULL	
delivery	tinyint(1)	NO		NULL	
address_id	int(3)	NO	MUL	NULL	

Relationship between all the tables:



SELECT * FROM orders;

row_id	order_id	created_at	item_id	quantity	customer_id	delivery	address_id
1	101	2024-01-01 09:25:43	it_004	2	4	1	4
2	101	2024-01-01 11:00:35	it_045	1	4	1	4
3	101	2024-01-01 13:02:48	it_043	1	4	1	4
4	102	2024-01-01 13:54:59	it_024	2	7	1	7
5	102	2024-01-01 14:45:31	it_044	4	7	1	7
6	103	2024-01-01 16:30:40	it_040	1	11	0	11
7	103	2024-01-01 17:49:00	it_051	1	11	0	11

➤ STEP 3:

Using a query to create a table which will help in accessing the data for either billing or creating a dashboard.

```
CREATE TABLE order_data AS
SELECT o.order_id,
i.item_category,
i.item_name,
o.quantity,
i.item_price,
o.created_at,
a.address_1,
a.address_2,
a.address_city,
a.address_pincode,
o.delivery
FROM orders o
LEFT JOIN items i ON o.item_id = i.item_id
LEFT JOIN address a ON o.address_id = a.address_id;
```

Sample of 'order_data' table:

```
SELECT * FROM order_data;
```

From order_id: 101

order_id	item_category	item_name	quantity	item_price	created_at	address_1	address_2	address_city	address_pincode	delivery
101	Veg_pizza	Farm_house	2	229.00	2024-01-01 09:25:43	44 Elm Circle	Belapur	Mumbai	400614	1
101	Beverage	Mirinda	1	30.00	2024-01-01 11:00:35	44 Elm Circle	Belapur	Mumbai	400614	1
101	Beverage	Pepsi	1	30.00	2024-01-01 13:02:48	44 Elm Circle	Belapur	Mumbai	400614	1
102	Non-veg_pizza	Chicken_golden_delight	2	699.00	2024-01-01 13:54:59	77 Cedar Avenue	Ghansoli	Mumbai	400701	1
102	Beverage	7UP	4	30.00	2024-01-01 14:45:31	77 Cedar Avenue	Ghansoli	Mumbai	400701	1
103	Non-veg_pizza	Indi_chicken_tikka	1	319.00	2024-01-01 16:30:40	21 Cedar Park	Belapur	Mumbai	400614	0
103	Side	Lava_cake	1	99.00	2024-01-01 17:49:00	21 Cedar Park	Belapur	Mumbai	400614	0
104	Veg_pizza	Peppy_Paneer	2	599.00	2024-01-01 19:25:48	22 Oak Lane	Airoli	Mumbai	400708	1
104	Topping	Extra_cheese	1	20.00	2024-01-01 19:45:50	22 Oak Lane	Airoli	Mumbai	400708	1
104	Topping	Extra_veggies	1	20.00	2024-01-01 20:16:12	22 Oak Lane	Airoli	Mumbai	400708	1
104	Beverage	Pepsi	4	30.00	2024-01-01 20:16:23	22 Oak Lane	Airoli	Mumbai	400708	1
104	Non-veg_pizza	Chicken_dominator	1	319.00	2024-01-01 20:32:39	22 Oak Lane	Airoli	Mumbai	400708	1
105	Veg_pizza	Margherita	1	199.00	2024-01-02 08:55:37	99 Redwood Lane	Airoli	Mumbai	400708	0

till order_id: 262

261	Topping	Extra_cheese	2	20.00	2024-01-31 19:17:00	120 Pine Avenue	Ghansoli	Mumbai	400701	0
262	Non-veg_pizza	chicken_fiesta	4	459.00	2024-01-31 19:30:00	09 Sequoia Lane	Kopar Khairane	Mumbai	400709	1
262	Veg_pizza	Deluxe_veggie	2	699.00	2024-01-31 20:10:02	09 Sequoia Lane	Kopar Khairane	Mumbai	400709	1
262	Beverage	7UP	2	30.00	2024-01-31 21:00:10	09 Sequoia Lane	Kopar Khairane	Mumbai	400709	1
262	Beverage	Lipton_icetea	1	20.00	2024-01-31 23:17:04	09 Sequoia Lane	Kopar Khairane	Mumbai	400709	1

➤ STEP 4:

DASHBOARD

SLICER (ORDER ID):

Used to select Order_id from the list to filter visualization.

KPIs:

- TOTAL ORDERS: Shows total no. of ordered placed.
- TOTAL QUANTITY: Shows no. of ordered got placed.
- TOTAL SALES: Shows total revenue.
- AVG ORDER VALUE: Shows average revenue per order.

PIE CHART (DELIVERY STATUS):

Shows distribution of delivery status either 'Yes' or 'No'. Yes, means customer opted for home delivery and No means they decided to dine in.

CLUSTERED BAR CHART (TOP PROFITABLE ITEMS):

Shows top revenue generating items

MAP CHART:

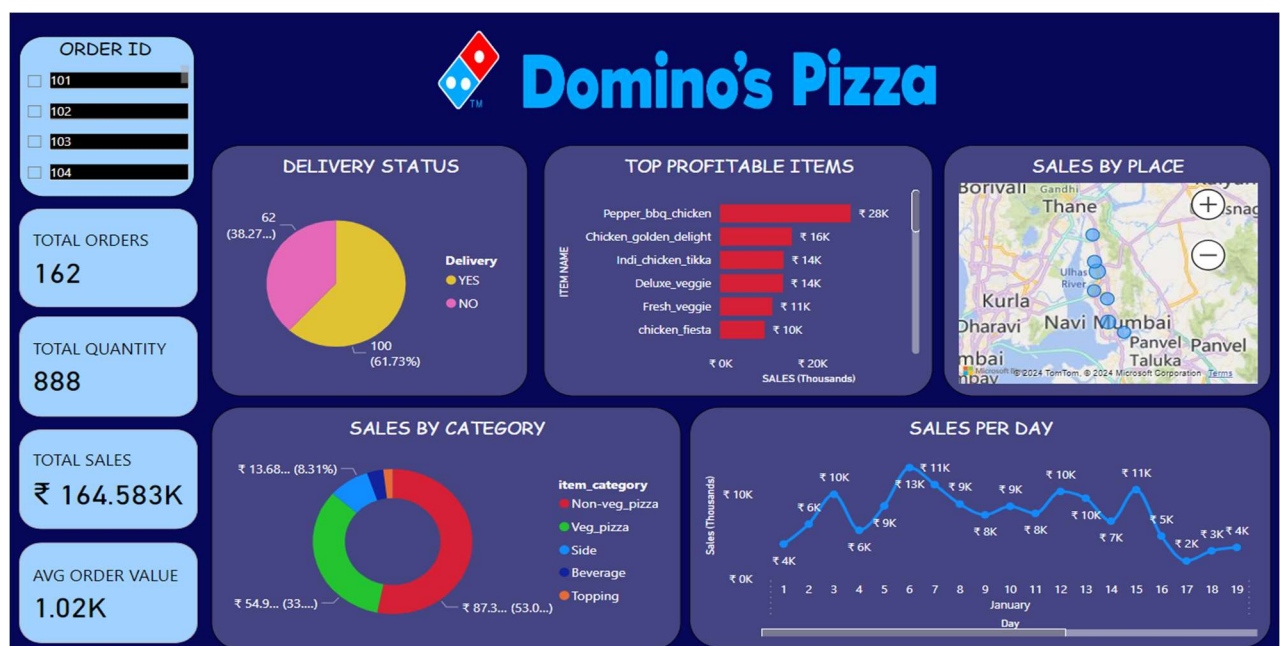
Highlight bubbles in location which shows sales volumes.

DONUT CHART (SALES BY CATEGORY):

Shows distribution of sales volume and sales percentage by item category.

LINE CHART (SALES PER DAY):

Shows sales occurred per day in the month of January 2024.



ORDER DETAILS

SLICER (ORDER ID):

Used to filter and choose order_id to get customer data and collect revenue.

NEW CARDS:

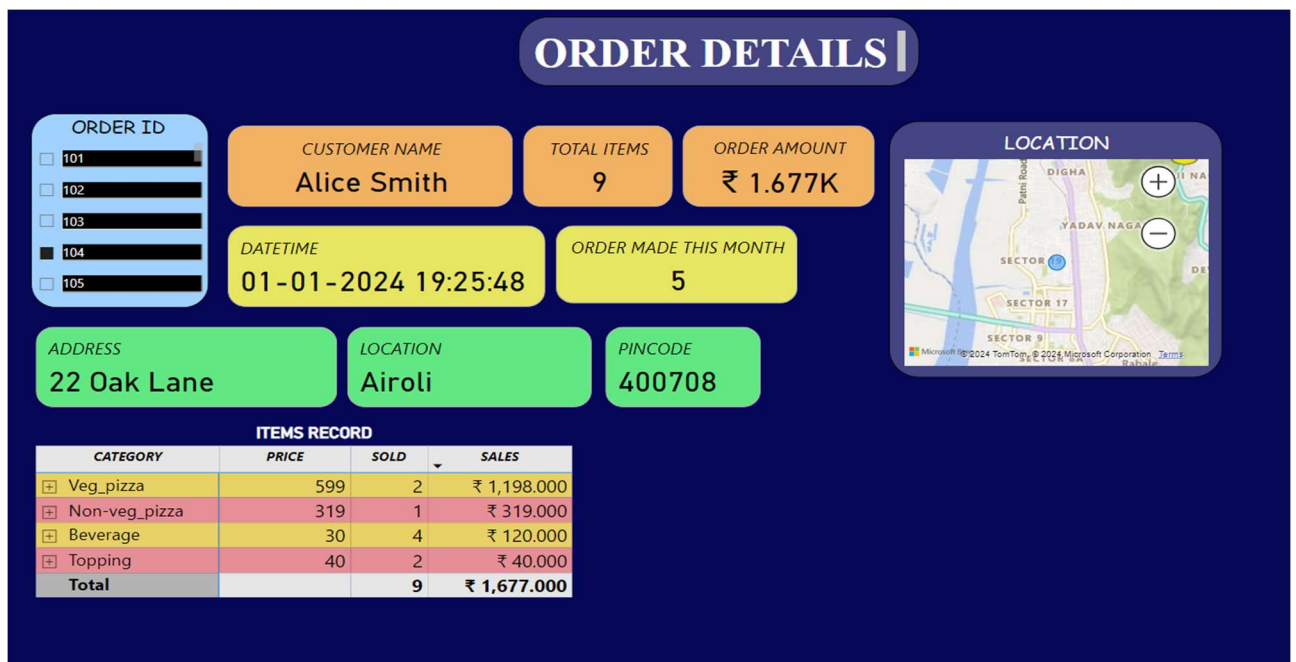
Shows Customer name, Total items bought, Order amount, Date & time of order, Customer complete address for delivery, No. of order by customer this month

MAP (LOCATION):

Display of proper address of customer with pin code.

ITEM RECORDS:

Mini statement and drill through to check Item's category, Item's name, Item's price, No. of items sold and Total sales(revenue) gained through that order.



NARRATION (for January 2024):

January 6th, 2024 marked a significant day in sales performance, with a notable achievement of 8.03% of total sales. Additionally, with total sales amounting to ₹164,583.

Standing out prominently amidst the diverse menu items was Pepper_bbq_chicken, commanding a substantial total sales figure of ₹28,440 by a staggering 2,330.77%, with Pepsi recording the lowest total sales at ₹1,170.

This wide sales spectrum highlights the diverse market dynamics at play.