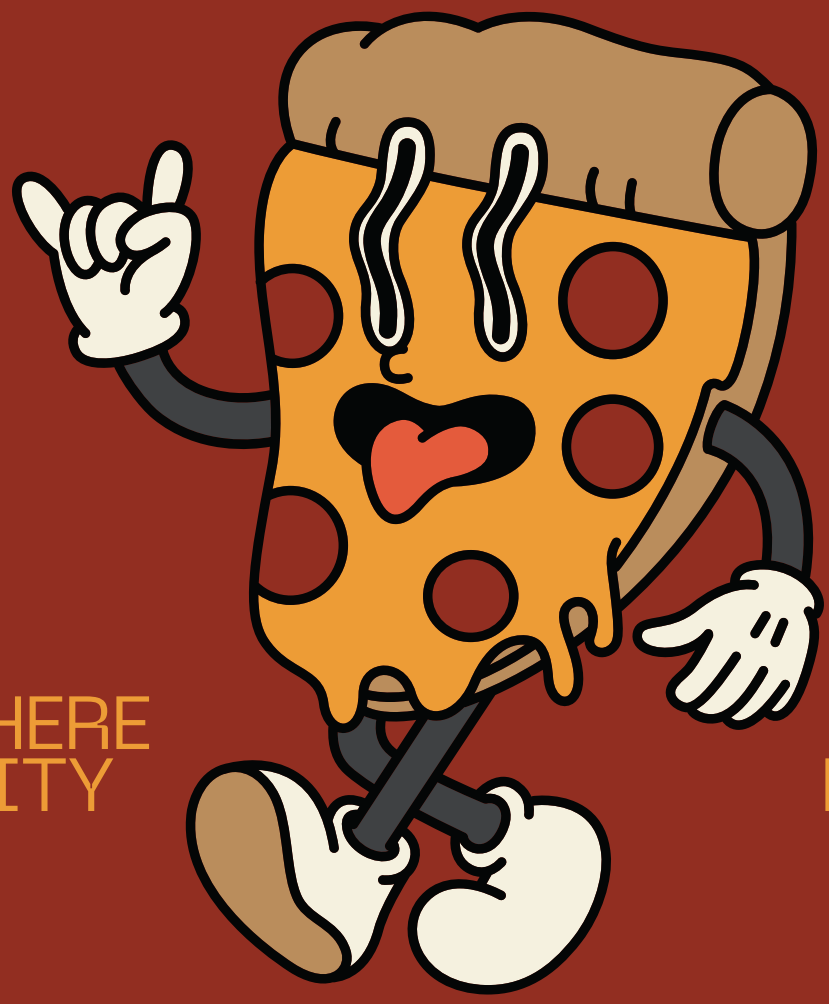
PIZZA SALES

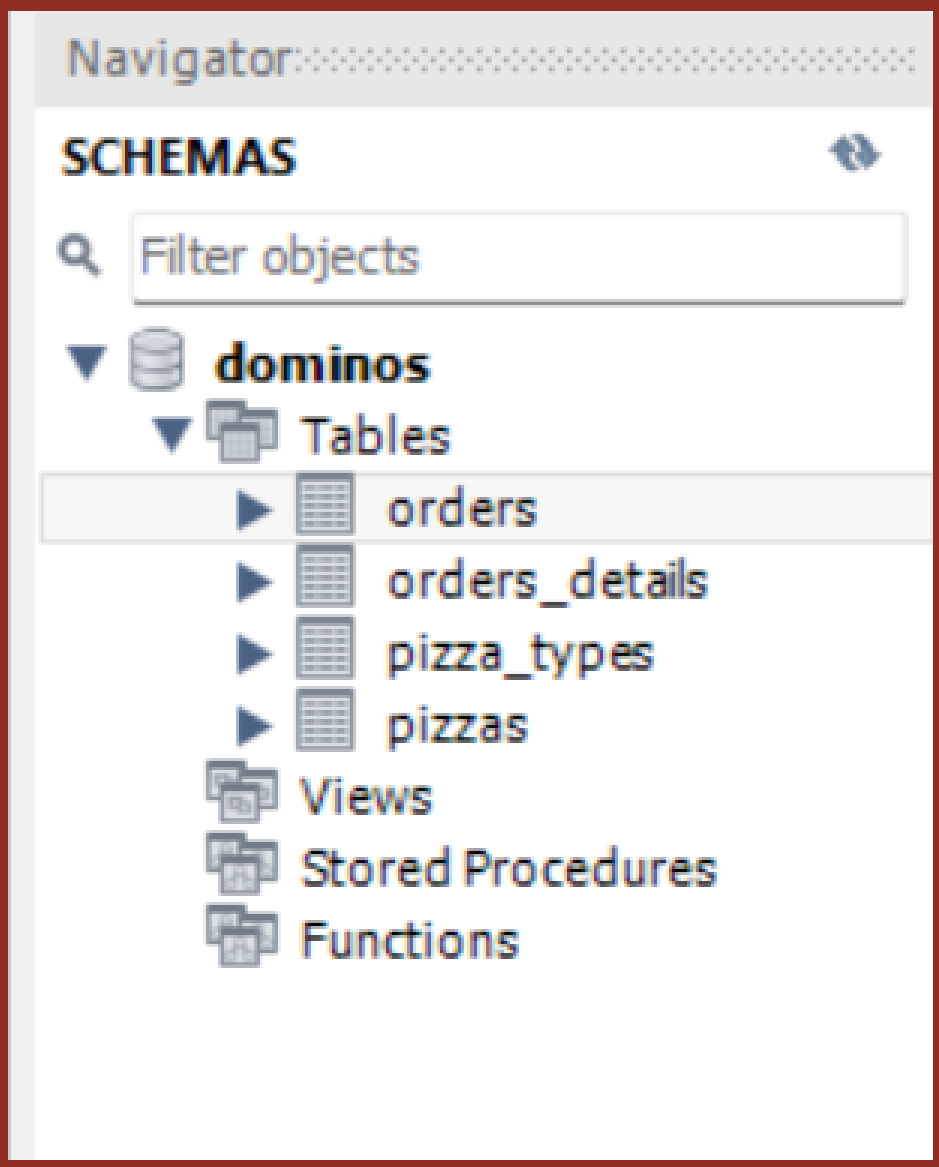
SQL

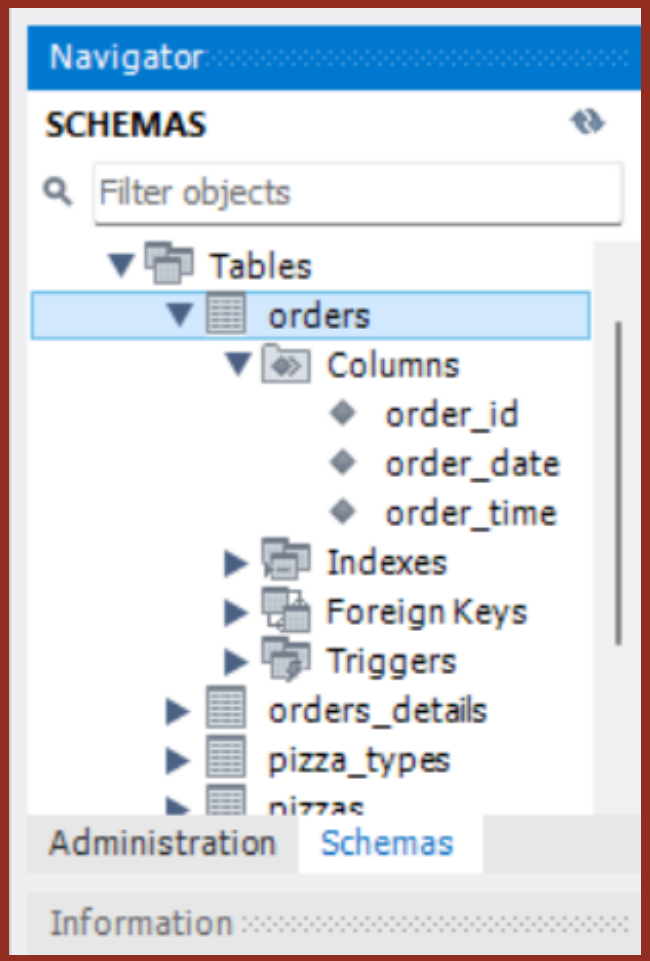


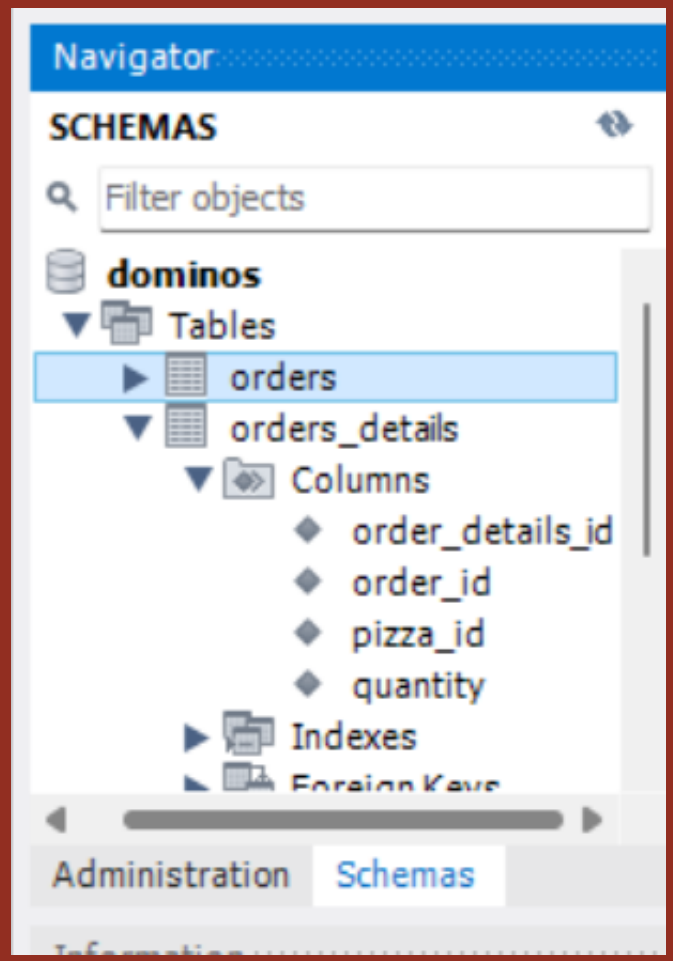
123 ANYWHERE
ST., ANY CITY

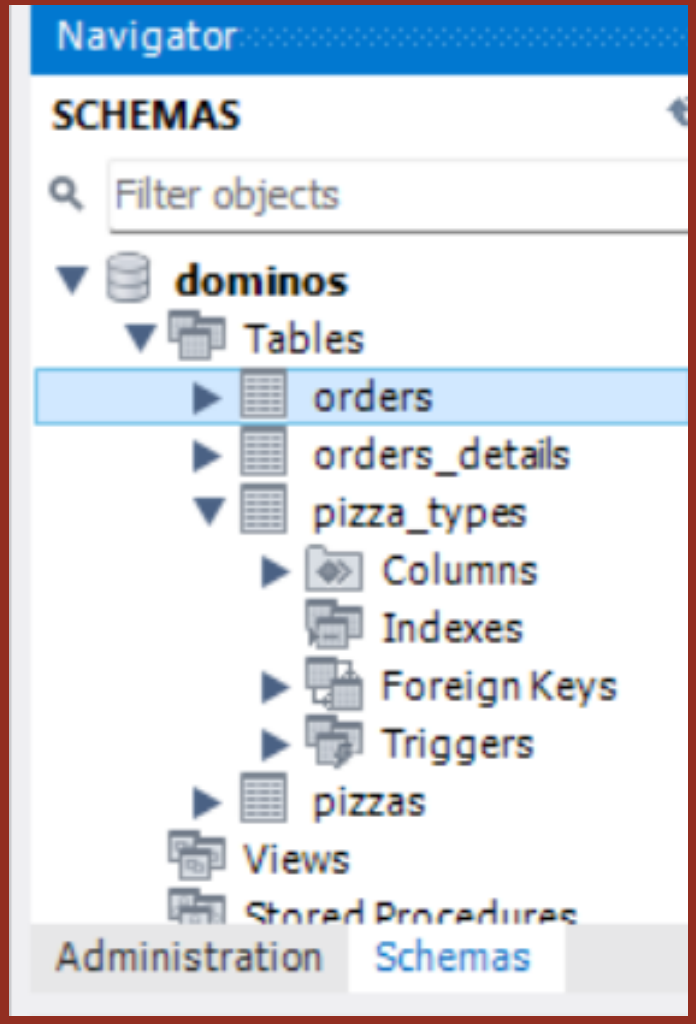
EVERY DAY
FROM 6PM TO 11 PM

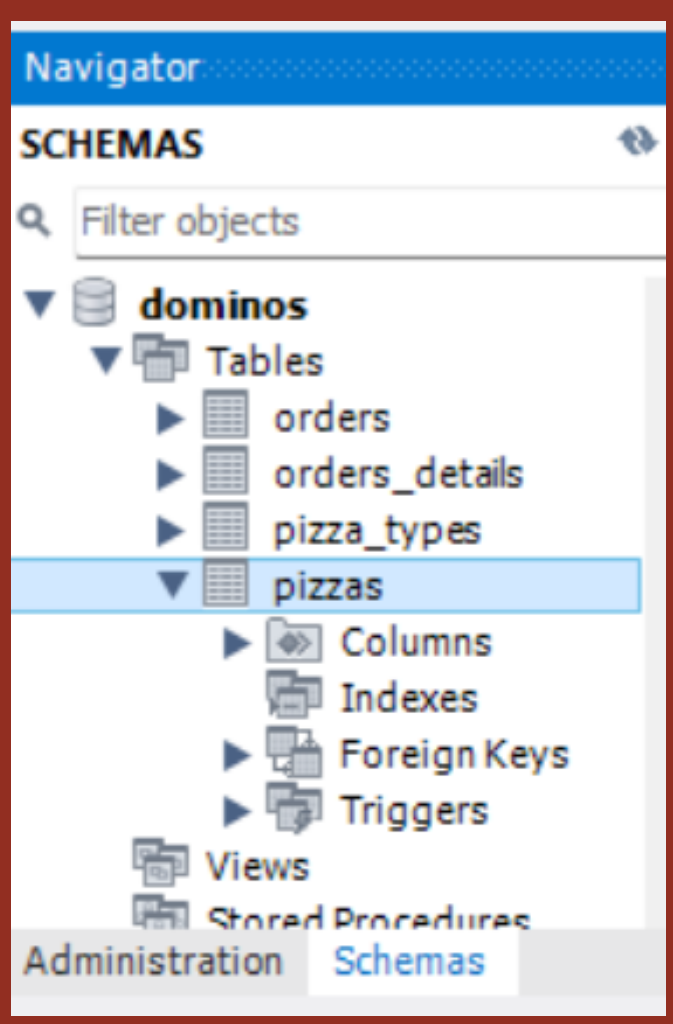
```
1 •    create database Dominos;
2
```




Navigator
SCHEMAS
Filter objects
Tables
orders
Columns
order_id
order_date
order_time
Indexes
Foreign Keys
Triggers
orders_details
pizza_types
pizzas
Administration
Schemas
Information


Navigator
SCHEMAS
Filter objects
dominos
Tables
orders
orders_details
Columns
order_details_id
order_id
pizza_id
quantity
Indexes
Administration
Schemas


Navigator
SCHEMAS
Filter objects
dominos
Tables
orders
orders_details
pizza_types
Columns
Indexes
Foreign Keys
Triggers
pizzas
Views
Administration
Schemas


Navigator
SCHEMAS
Filter objects
dominos
Tables
orders
orders_details
pizza_types
pizzas
Columns
Indexes
Foreign Keys
Triggers
Views
Administration
Schemas

Ques-> Retrive the total number of orders placed.

Sol->

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Output

Result Grid | Filter Rows:

	total_orders
▶	21350

Ques-> CALCULATE THE TOTAL REVENUE GENRATED FROM PIZZA SALES.ALTER

Sol->

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

OUtput->

Result Grid

total_sales
817860.05

Ques-> Identify the highest-priced pizza.

Sol->

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Output->

Result Grid | Filter Rov

	name	price
▶	The Greek Pizza	35.95

Ques-> Identify the most common pizza size ordered.

Sol->

```sql
SELECT
    quantity, COUNT(order_details_id)
FROM
    orders_details
GROUP BY quantity;
```

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Output->

size	order_count
L	18526
M	15385
S	14137
XL	544

Ques-> List the top 5 most ordered pizza types along with their quantities.

Sol->

```sql
select pizza_types.name,
sum(orders_details.quantity)as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by quantity desc limit 5
```

Output->

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Ques->Determine the distribution of orders by hour of the day.

Sol->

```
select hour(order_time)as hour ,
count(order_id)as order_count from  orders
group by hour(order_time);
```

Output - >

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

Ques-> Join relevant tables to find the category-wise distribution of pizzas.

Sol->

```
select category,count(name) from pizza_types
group by category;
```

Output->

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Ques->Group the orders by date and calculate the average number of pizzas ordered per day.

Sol->

```sql
select round(avg(quantity),0)from
(select orders.order_date ,  sum(orders_details.quantity) as quantity
from orders join orders_details
on orders.order_id = orders_details.order_id
group by orders.order_date) as order_quantity;
```

Output->

round(avg(quantity),0)
138

Ques-> Determine the top 3 most ordered pizza types based on revenue

Sol->

```sql
select pizza_types.name,
sum(orders_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3 ;
```

Output->

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Ques-> Analyze the cumulative revenue generated over time.

sol->

```sql
select order_date,
sum(revenue) over (order by order_date)as cum_revenue from
(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as sales;
```

Output->

order_date	cum_revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05