
Pizza Sales Management System
Optimizing Pizza Sales and Inventory Management

Prepared By: Ravi Mishra

Institute- IT. VEDANT

Date: November 11, 2024

Project Overview:

This project aims to develop a comprehensive Pizza Sales Management System to streamline and optimize pizza sales, inventory tracking, and customer management. The system utilizes SQL for database management, ensuring data integrity and easy access to information, ultimately enhancing operational efficiency and customer satisfaction.

➤ Creating a database

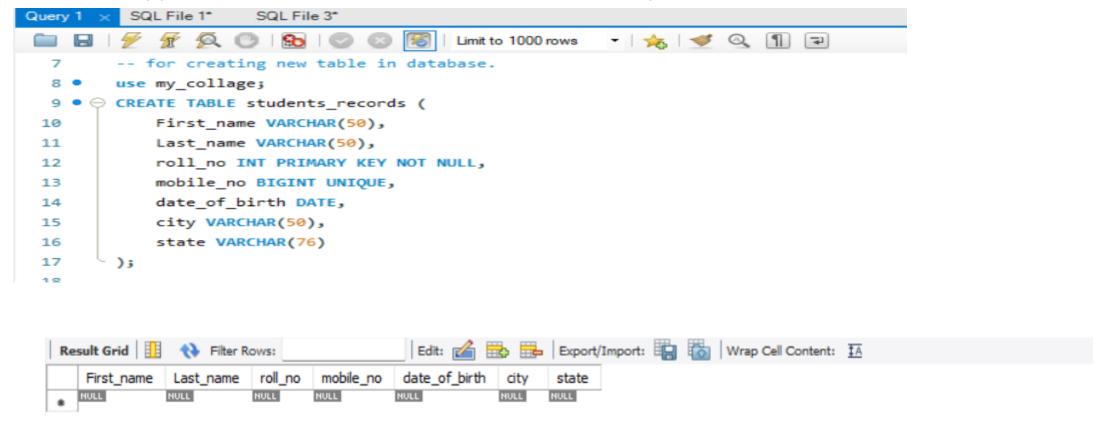
Database can be created using below query where you can store tables

```
Query 1 × SQL File 1* SQL File 3*

| SQL File 1* SQL File 3* | SQL File
```

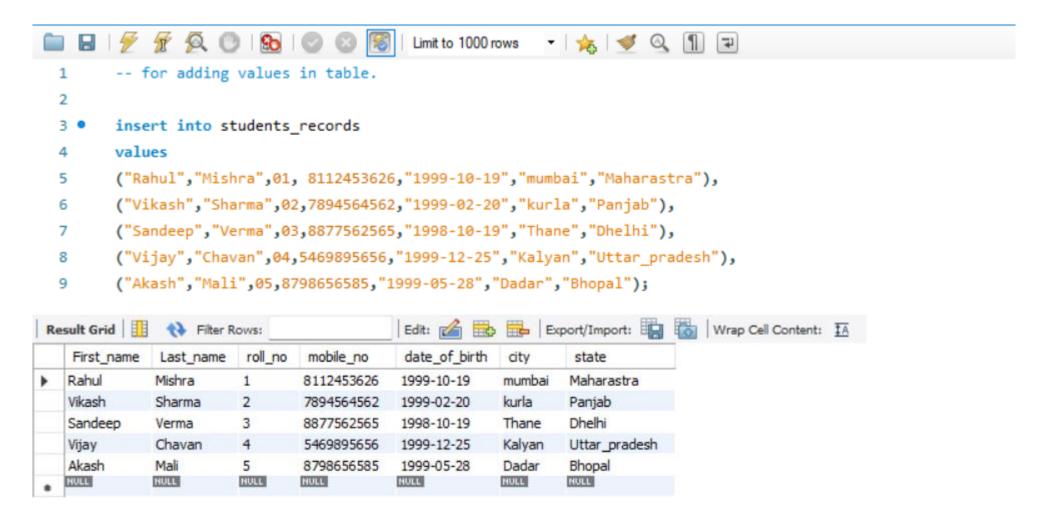
➤ Creating a table using constraints

Table is created to store data in it and we should assign datatype to each column in table. Constraint is a rule applied to column in a table to ensure the accuracy



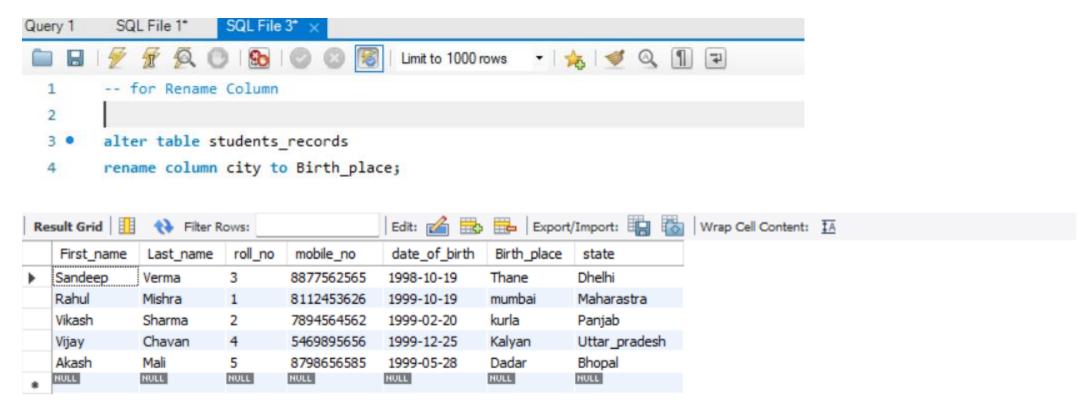
➤ Inserting value in table

For inserting data in table, apply the below query and then add values in the column.



➤ Renaming a column in a table:

For renaming column name first we have to apply alter table command then apply rename query.



➤ Drop and truncate table:

Drop table means whole table will be dropped including data as well as table structure and truncate table means only data will be deleted but the structure will remain same

```
22  -- drop And truncate Column
23
24      drop table students_records;
25      truncate table students_records;
```

➤ Delete command:

Delete command is used for deleting records based on specific conditions.

```
-- delete command

29

30 • delete from students_records
where First_name ="Vijay";
```

➤Update function:

Update is used for updating records in the table, example of update query is shown below

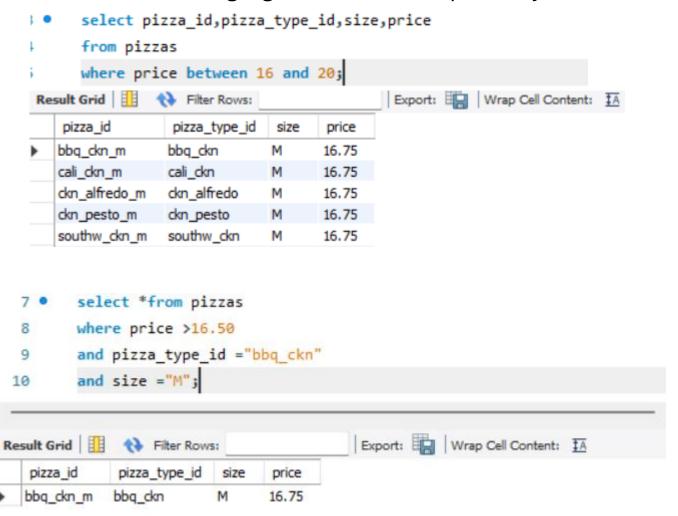
```
set sql_safe_updates=0;

update my_collage.students_records
set state = "Delhi" where state="Dhelhi";
```

Re	sult Grid	♦ Filter R	lows:		Edit: 🚄 🖶	Export	/Import:
	First_name	Last_name	roll_no	mobile_no	date_of_birth	Birth_place	state
•	Rahul	Mishra	1	8112453626	1999-10-19	mumbai	Maharastra
	Vikash	Sharma	2	7894564562	1999-02-20	kurla	Panjab
	Sandeep	Verma	3	8877562565	1998-10-19	Thane	Delhi
	Vijay	Chavan	4	5469895656	1999-12-25	Kalyan	Uttar_pradesh
	Akash	Mali	5	8798656585	1999-05-28	Dadar	Bhopal
	NULL	NULL	NULL	NULL	NULL	HULL	NULL

➤ Comparison and logical operator

Comparison and logical operators are used for comparing the values and using logical functions respectively.



➤ Arithmetic operators:

Arithmetic operators are used to find various operations such as sum, max, min, avg, etc

```
SELECT
        SUM(price) AS sum_of_total_price,
        AVG(price) AS avg_price,
        MIN(price) AS min_price,
        MAX(price) AS max_price
    FROM
        pizzas;
                                           Export: Wrap Cell Content: IA
Result Grid
              Filter Rows:
   sum_of_total_price avg_price
                              min_price
                                       max_price
 1578.3
                   16.440625
                             9.75
                                       35.95
```

► Math functions

In math function we can do operations such as absolute, floor, ceil, round.

```
21 .
           SELECT
22
           ABS(- 500) AS whole on,
           CEIL(- 5.6) AS ceil_result,
23
           FLOOR(- 5.6) AS floor_result,
24
25
           CEIL(5.6) AS ceil_result_1,
           FLOOR(5.6) AS floor result 1,
26
27
           ROUND(5.4) AS round_result,
           ROUND(5.6) AS round_result_1,
28
           POWER(2, 5) AS power,
29
           SQRT(775) AS sqrt;
30
```

Res	sult Grid 🛮	N Filter	Rows:		Export: Wra	ap Cell Content:	<u>‡A</u>		
	whole_on	ceil_result	floor_result	ceil_result_1	floor_result_1	round_result	round_result_1	power	sqrt
•	500	-5	-6	6	5	5	6	32	27.83882181415011

► Like function

It is used to find the word starting with , ending with or containing some letter in it.

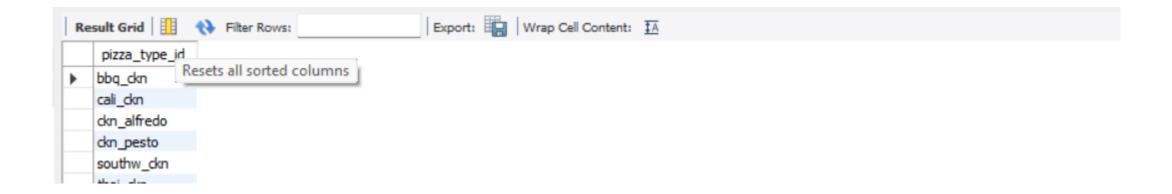
```
select pizza_type_id,name
from pizza_types
where pizza_type_id like ("c%")
and pizza_type_id like ("%o");
```



▶ Distinct:

Distinct keyword is used to retrieve unique values from a specified column.

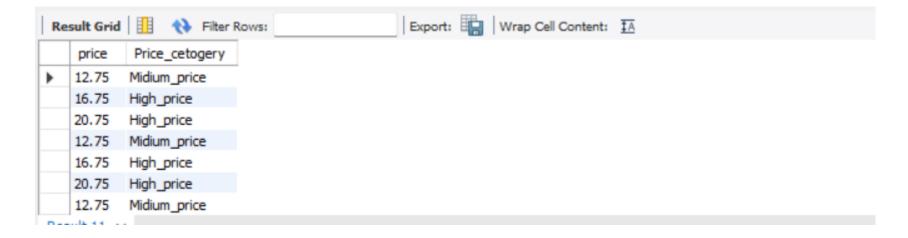
```
9 • select distinct pizza_type_id
10 from pizza_types;
```



➤ Case statement:

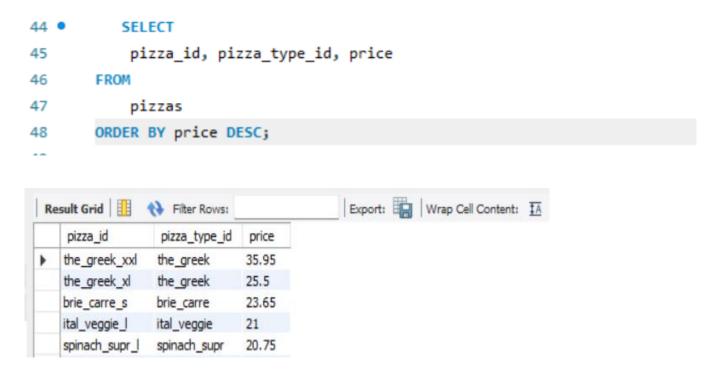
The case statement gives us result according to given conditions, example of case statement is given below

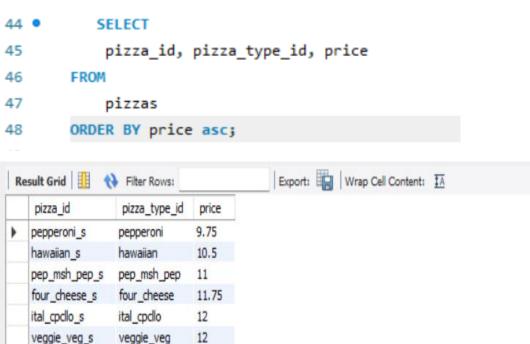
```
34 •
           SELECT
35
           price,
36
           CASE
               WHEN price > 16 THEN 'High price'
37
               WHEN price >= 10 THEN 'Midium price'
38
               WHEN price < 10 THEN 'Low'
39
           END AS Price cetogery
40
41
       FROM
           pizzas;
42
```



➤Order by:

Order by is nothing but arranging the values in the column in Ascending order or descending order.





Retrieve the total number of orders placed.

Joins

In SQL, joins are used to combine rows from two or more tables based on a related column between them.

Here are the most common types of joins:

- Inner Join
- Left Join (Left Outer Join)
- Right Join (Right Outer Join)
- Full Join (Full Outer Join)
- Cross Join
- Self Join

LEFT JOIN

A **LEFT JOIN** in SQL is used to retrieve all records from the left table (the first table in the join statement), and the matched records from the right table (the second table in the join statement).

```
4 • select * from orders_details
5 left join pizzas
6 on pizzas.pizza_id = orders_details.pizza_id;
```

Re	sult Grid 🔢 🔌	Filter Row	/S:	Exp	ort: Wrap	Cell Content: TA	Fetch	rows:
	order_details_id	order_id	pizza_id	quantity	pizza_id	pizza_type_id	size	price
•	1	1	hawaiian_m	1	hawaiian_m	hawaiian	M	13.25
	2	2	classic_dlx_m	1	classic_dlx_m	classic_dlx	M	16
	3	2	five_cheese_l	1	five_cheese_l	five_cheese	L	18.5
	4	2	ital_supr_l	1	ital_supr_l	ital_supr	L	20.75
	5	2	mexicana_m	1	mexicana_m	mexicana	M	16
	6	2	thai_ckn_l	1	thai_ckn_l	thai_ckn	L	20.75

RIGHT JOIN

A **RIGHT JOIN** (or right outer join) in SQL is used to retrieve all records from the right table (the second table in the join statement), and the matched records from the left table (the first table in the join statement).

```
9 • select *from orders_details
10    right join pizzas
11    on pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid				1	Export:	Wrap Cell Conten	t: <u>‡A</u>	Fetch rows:
	order_details_id	order_id	pizza_id	quantity	pizza_id	pizza_type_id	size	price
١	8895	3899	cali_ckn_l	1	cali_ckn_l	cali_ckn	L	20.75
	8881	3892	cali_ckn_l	1	cali_ckn_l	cali_ckn	L	20.75
	8880	3892	bbq_ckn_m	1	bbq_dkn_m	bbq_ckn	M	16.75
	8869	3887	bbq_ckn_s	1	bbq_dkn_s	bbq_ckn	S	12.75
	8868	3887	bbq_ckn_m	1	bbq_ckn_m	bbq_ckn	M	16.75
	8861	3886	cali_ckn_l	2	cali_ckn_l	cali_ckn	L	20.75

Cross Join

It Returns the Cartesian product of both tables, meaning it combines each row of the first table with each row of the second table.

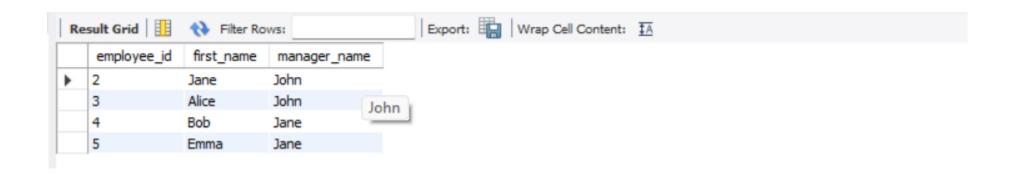
```
13 • select *from orders_details
14     cross join pizzas
15     on pizzas.pizza_id = orders_details.pizza_id;
```

	order_details_id	order_id	pizza_id	quantity	pizza_id	pizza_type_id	size	price
١	1	1	hawaiian_m	1	hawaiian_m	hawaiian	М	13.25
	2	2	classic_dlx_m	1	classic_dlx_m	classic_dlx	M	16
	3	2	five_cheese_l	1	five_cheese_l	five_cheese	L	18.5
	4	2	ital_supr_l	1	ital_supr_l	ital_supr	L	20.75
	5	2	mexicana_m	1	mexicana_m	mexicana	M	16
	6	2	thai_ckn_l	1	thai_ckn_l	thai_ckn	L	20.75

Self Join

A **self join** is a join where a table is joined with itself. This can be useful when you need to compare rows within the same table or create hierarchical queries. To perform a self join, you use table aliases to represent the table multiple times in the query.

```
select e1.employee_id, e1.first_name,e2.first_name as manager_name
from employees e1
join employees e2
on e1.manager_id = e2.employee_id;
```

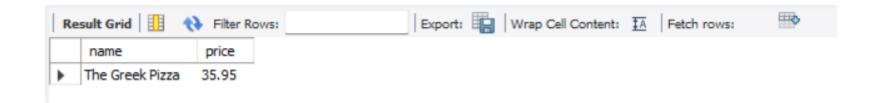


Calculate the total revenue generated from pizza sales.



Identify the highest-priced pizza.

```
52 •
       SELECT
           pizza_types.name, pizzas.price
53
54
       FROM
55
           pizza_types
56
               JOIN
57
           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
       ORDER BY price DESC
58
       LIMIT 1;
59
```



Identify the most common pizza size ordered.

```
SELECT

pizzas.size,

COUNT(orders_details.order_details_id) AS total_orders

FROM

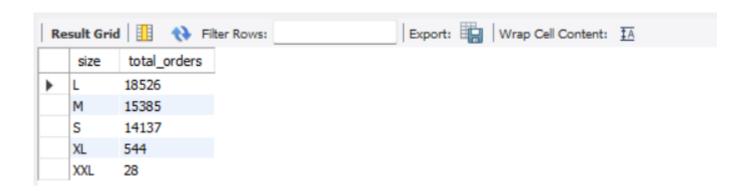
pizzas

JOIN

orders_details ON pizzas.pizza_id = orders_details.pizza_id

GROUP BY pizzas.size

ORDER BY total_orders DESC;
```



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

```
SELECT
3 •
           pizza types.name,
 5
           SUM(orders_details.quantity) AS total_order_quantity
 6
       FROM
           pizza_types
               JOIN
9
           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
               JOIN
10
           orders_details ON orders_details.pizza_id = pizzas.pizza_id
11
       GROUP BY pizza_types.name
12
13
       ORDER BY total order quantity DESC
       LIMIT 5;
14
```

	esult Grid 1	1 10 2 10 10 10 10 10 10 10 10 10 10 10 10 10	Export:	-	Wrap Cell Content:	-	Discound of the Sale	
	name	total_order_quantity						
١	The Classic Deluxe Pizza	2453						
	The Barbecue Chicken Pizza	2432						
	The Hawaiian Pizza	2422						
	The Pepperoni Pizza	2418						
	The Thai Chicken Pizza	2371						

Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
           pizza_types.category,
           SUM(orders details.quantity) AS Total order quentity
       FROM
 6
           pizza_types
               JOIN
           pizzas ON pizza types.pizza type id = pizzas.pizza type id
               JOIN
10
           orders_details ON orders_details.pizza_id = pizzas.pizza_id
11
12
       GROUP BY pizza types.category
13
       ORDER BY Total_order_quentity DESC;
  Result Grid
                                              Export: Wrap Cell Content: $\overline{A}$
                Filter Rows:
     category
              Total_order_quentity
              14888
     Classic
     Supreme
              11987
     Veggie
              11649
     Chicken
              11050
```

Subquery

A subquery, also known as an inner query or nested query, is a query within another SQL query. It is used to perform operations in multiple steps by breaking down the complex query into simpler parts.

Retrieve the details of the pizza with the highest price.

