




Pizza Sales Analysis

Using MYSQL





 **Project Goal :** Analyze pizza sales data to derive insightful information for the company.

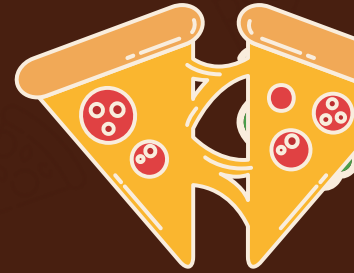
 **Data Handling :** Utilize MySQL to manage and analyze various aspects of the sales data.

 **Benefits for Product Team :**

1. Provide valuable insights into consumer preferences .
2. Identify sales trends.
3. Assess operational effectiveness.



TABLE OF CONTENTS



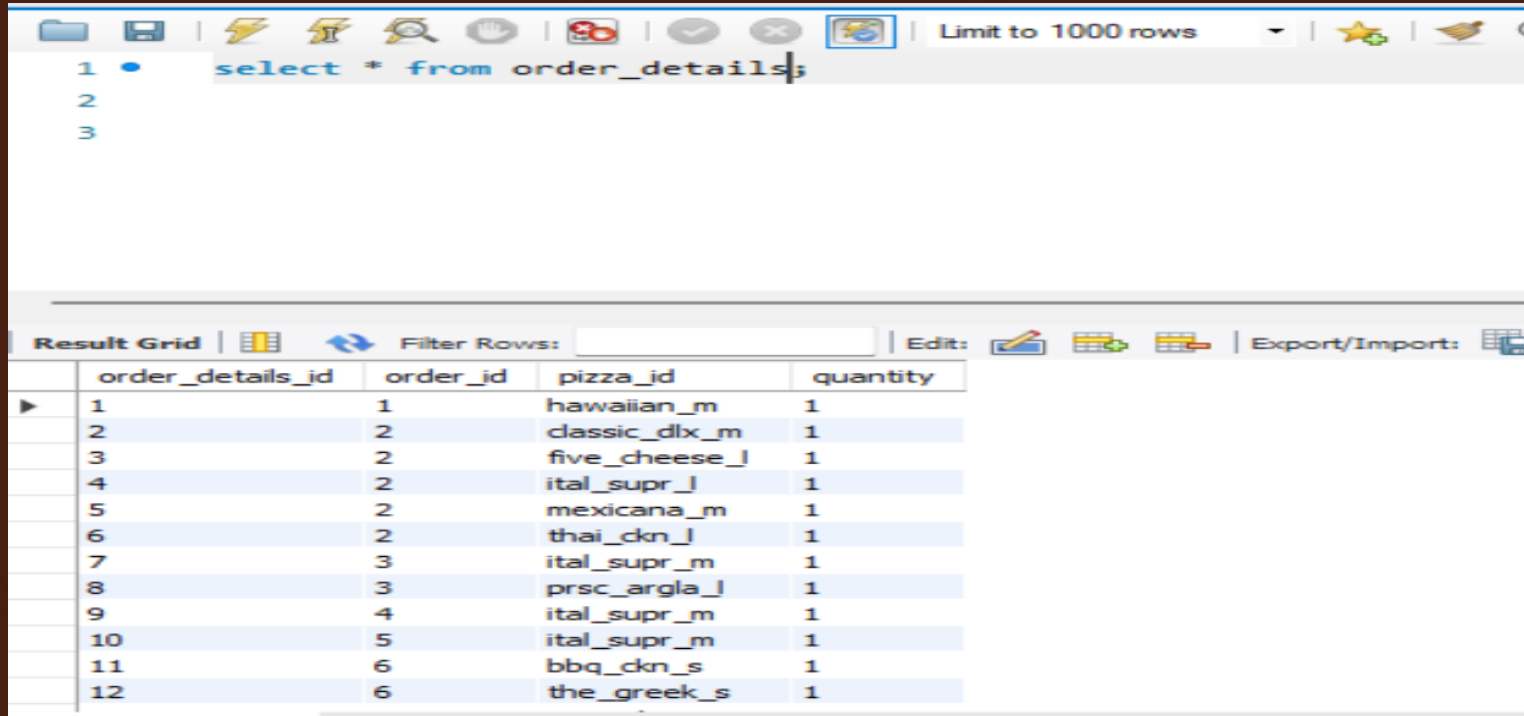
- Retrieve the total no of orders placed.
- calculate total revenue generated from pizza sales.
- Identify the highest prize of the pizza.
- Identify the most common pizzas size ordered.
- List the 5 most ordered pizzas types along with their quantities.
- join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- join relevant tables to find the category-wise distribution of pizzas
- Group the order by date and calculate the average number of pizzas ordered per day.
- Determine top 3 most ordered pizza types based on revenue.
- Determine top 3 most ordered pizza types based on revenue.



Lists of Tables

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Tables_in_pizza_hut			
▶	order_details			
	orders			
	pizza_types			
	pizzas			

Order_details table



The screenshot shows a database query tool interface. At the top, a toolbar contains various icons for file operations, execution, and viewing. A text input field contains the SQL query: `select * from order_details;`. Below the query, a 'Result Grid' section displays the query results in a table format. The table has four columns: `order_details_id`, `order_id`, `pizza_id`, and `quantity`. The results show 12 rows of data, representing individual pizza orders.

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1
11	6	bbq_ckn_s	1
12	6	the_greek_s	1



Orders table



SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL

Limit to 1000 rows

```
1 • select * from orders;  
2  
3
```

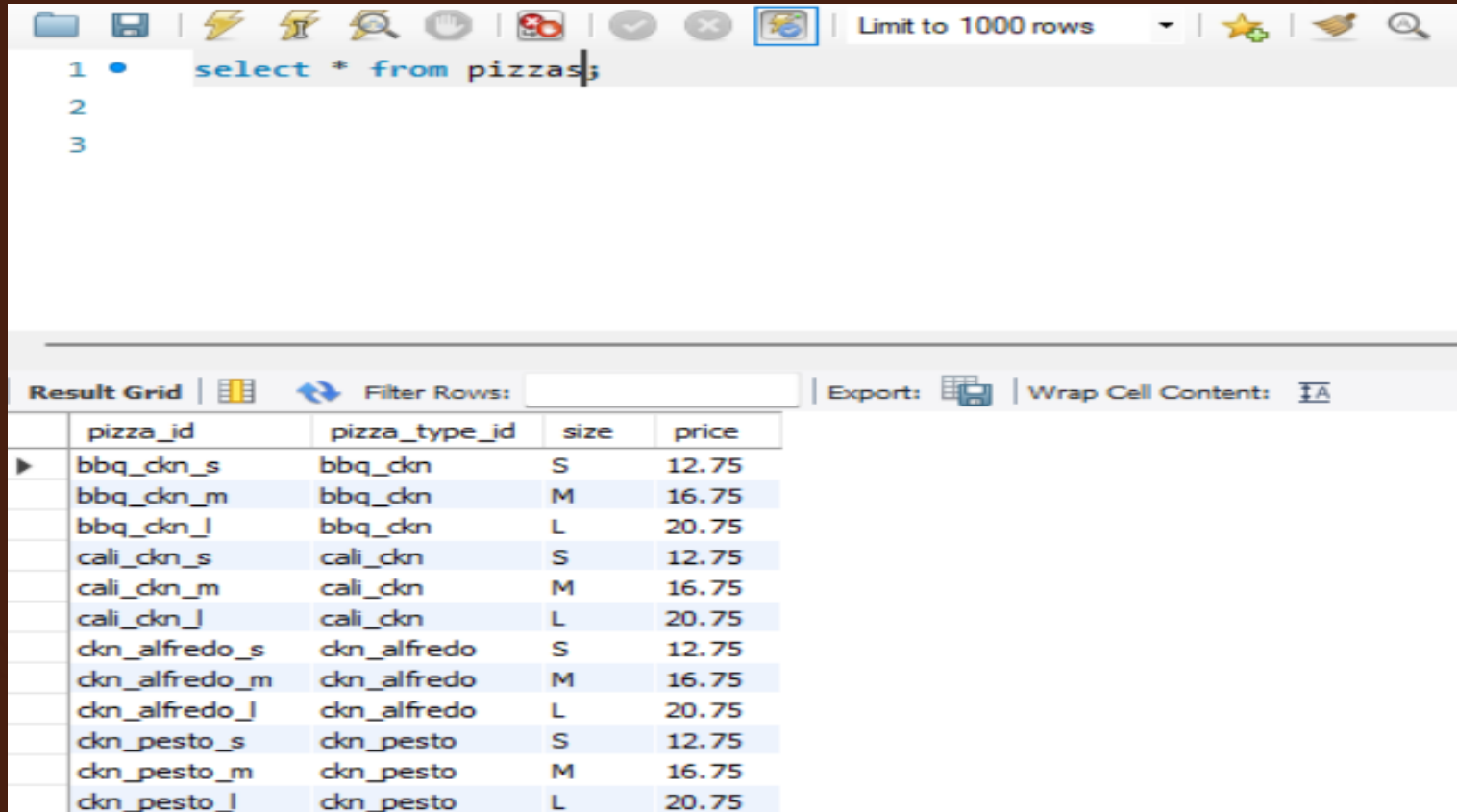
Result Grid | Filter Rows: | Edit: | Export/Im

	order_id	order_date	order_time
▶	1	2015-01-01	11:38:36
	2	2015-01-01	11:57:40
	3	2015-01-01	12:12:28
	4	2015-01-01	12:16:31
	5	2015-01-01	12:21:30
	6	2015-01-01	12:29:36
	7	2015-01-01	12:50:37
	8	2015-01-01	12:51:37
	9	2015-01-01	12:52:01
	10	2015-01-01	13:00:15
	11	2015-01-01	13:02:59
	12	2015-01-01	13:04:41





Pizzas table



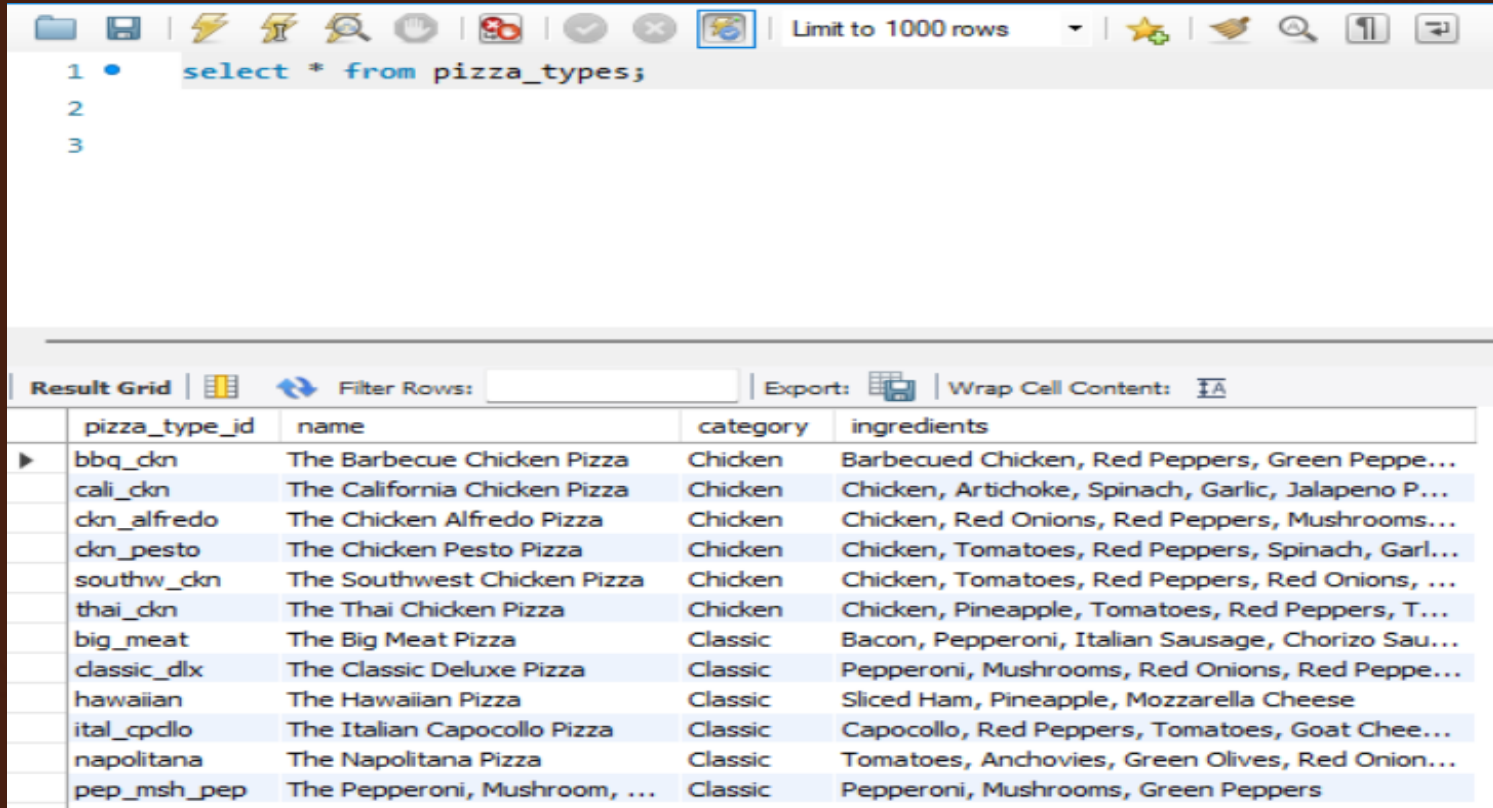
The image shows a database client interface. At the top, there is a toolbar with various icons for file operations, execution, and search. Below the toolbar, a SQL query is entered in a text area: `select * from pizzas;`. The query is executed, and the results are displayed in a table below. The table has four columns: `pizza_id`, `pizza_type_id`, `size`, and `price`. The results show 12 rows of data, representing different pizza combinations of type, size, and price.

	pizza_id	pizza_type_id	size	price
▶	bbq_ckn_s	bbq_ckn	S	12.75
	bbq_ckn_m	bbq_ckn	M	16.75
	bbq_ckn_l	bbq_ckn	L	20.75
	cali_ckn_s	cali_ckn	S	12.75
	cali_ckn_m	cali_ckn	M	16.75
	cali_ckn_l	cali_ckn	L	20.75
	ckn_alfredo_s	ckn_alfredo	S	12.75
	ckn_alfredo_m	ckn_alfredo	M	16.75
	ckn_alfredo_l	ckn_alfredo	L	20.75
	ckn_pesto_s	ckn_pesto	S	12.75
	ckn_pesto_m	ckn_pesto	M	16.75
	ckn_pesto_l	ckn_pesto	L	20.75





Pizza_types table



The screenshot shows a database query interface. At the top, there is a toolbar with various icons for file operations, search, and execution. Below the toolbar, a SQL query is entered in a text area: `select * from pizza_types;`. The query is numbered 1, 2, and 3. Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The main part of the result grid is a table with 5 columns: `pizza_type_id`, `name`, `category`, and `ingredients`. The table contains 12 rows of data, each representing a different pizza type. The first row is highlighted with a blue background.

	<code>pizza_type_id</code>	<code>name</code>	<code>category</code>	<code>ingredients</code>
▶	<code>bbq_chn</code>	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
	<code>cali_chn</code>	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
	<code>chn_alfredo</code>	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
	<code>chn_pesto</code>	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
	<code>southw_chn</code>	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
	<code>thai_chn</code>	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
	<code>big_meat</code>	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
	<code>classic_dlx</code>	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppe...
	<code>hawaiian</code>	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
	<code>ital_cpdllo</code>	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Chee...
	<code>napolitana</code>	The Napolitana Pizza	Classic	Tomatoes, Anchovies, Green Olives, Red Onion...
	<code>pep_msh_peg</code>	The Pepperoni, Mushroom, ...	Classic	Pepperoni, Mushrooms, Green Peppers





Retrieve the total no of orders placed.

The screenshot shows a SQL IDE interface with the following components:

- Menu Bar:** Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Icons for SQL, database, and other tools.
- Tab Bar:** SQL File 3*, SQL File 4*, SQL File 5*, SQL File 6*, SQL File 7*, SQL File 8*, SQL File 9*
- Editor:** Contains the SQL query:

```
1  -- Retrieve the total no of orders placed.  
2  
3  •  select * from orders;  
4  •  select count(order_id) from orders;
```
- Left Panel (Object Explorer):** Lists database objects including **pizza_hut** (Tables, Views, Stored Procedures, Functions), pizzas, student, and sys.
- Bottom Panel (Result Grid):** Shows the execution result of the query:

count(order_id)
21350





calculate total revenue generated from pizza sales



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar displays a 'SCHEMAS' tree with a search filter and a list of databases: ankita, ccdb, pizza_hut, pizzas, student, and sys. The 'pizza_hut' database is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main editor window, titled 'SQL File 4*', contains the following SQL query:

```
1  -- calculate total revenue generated from pizza sales.
2
3  SELECT
4      *
5  FROM
6      pizzas;
7  SELECT
8      ROUND(SUM(order_details.quantity * pizzas.price),
9             2)as Total_Sales
10 FROM
11     order_details
12     JOIN
13     pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

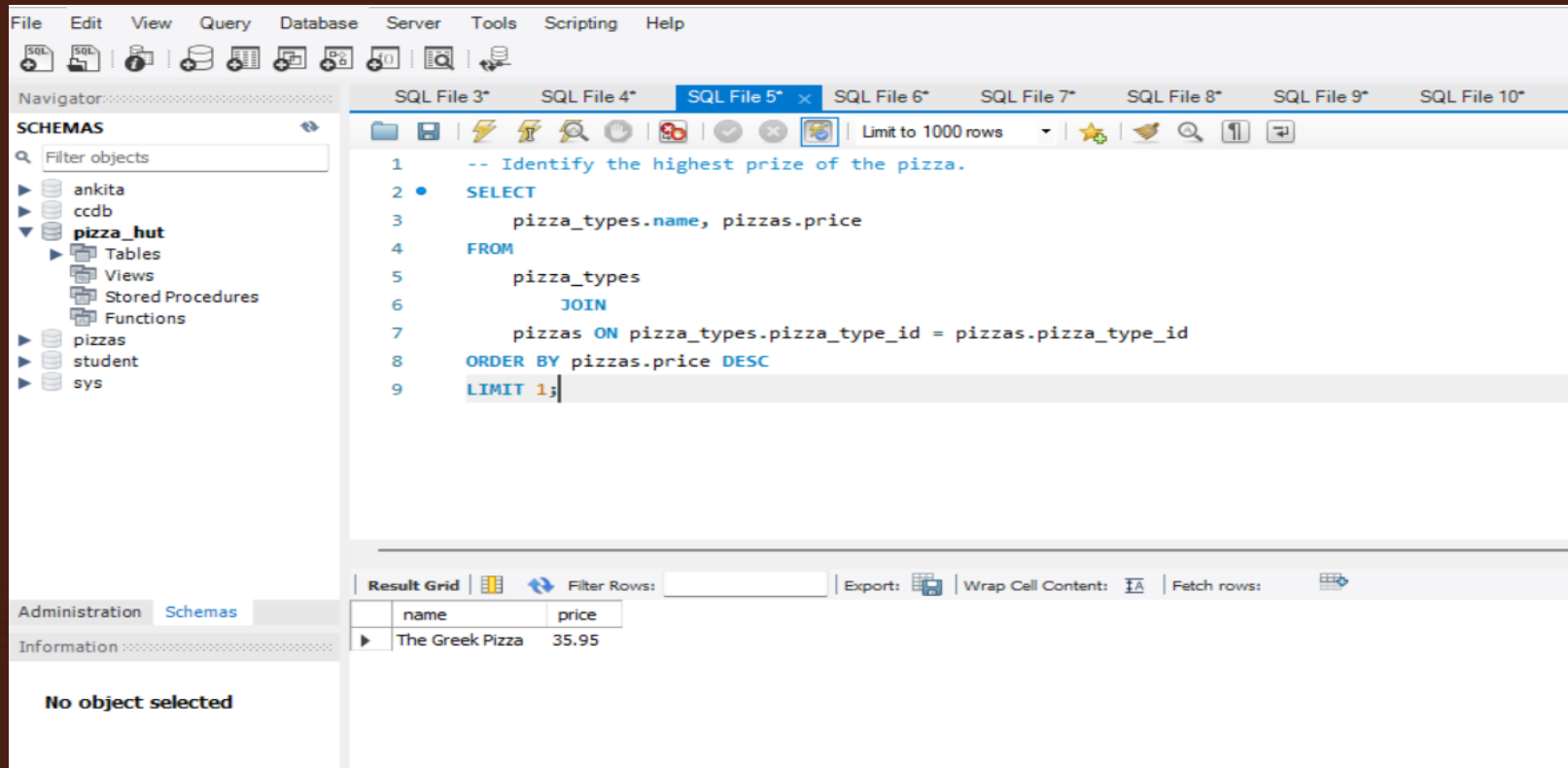
Below the query editor, the 'Result Grid' tab is active, displaying the query results in a table:

Total_Sales
817860.05

The bottom status bar indicates 'No object selected'.



Identify the highest prize of the pizza.



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The Navigator pane on the left displays a schema tree for 'pizza_hut' containing Tables, Views, Stored Procedures, and Functions. The main editor shows a SQL query in 'SQL File 5*' that selects the name and price of pizzas, ordered by price in descending order, limited to 1 row. The Result Grid at the bottom shows the query results.

```
1  -- Identify the highest prize of the pizza.
2  •  SELECT
3      pizza_types.name, pizzas.price
4  FROM
5      pizza_types
6      JOIN
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8  ORDER BY pizzas.price DESC
9  LIMIT 1;
```

Result Grid

name	price
The Greek Pizza	35.95

Information: No object selected



Identify the most common pizzas size ordered.



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar displays a 'CHEMAS' tree with a search filter and a list of databases: ankita, ccdb, pizza_hut, pizzas, student, and sys. The 'pizza_hut' database is expanded, showing Tables, Views, Stored Procedures, and Functions. The main editor window shows a SQL query in 'SQL File 6*' with line numbers 1 through 11. The query is:
1 -- Identify the most common pizzas size ordered.
2
3 SELECT
4 pizzas.size,
5 COUNT(order_details.order_details_id) AS order_count
6 FROM
7 pizzas
8 JOIN
9 order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC;
The bottom panel shows the 'Result Grid' with columns 'size' and 'order_count'. The results are:
size order_count
L 18526
M 15385
S 14137
XL 544
XXL 28
The 'Administration' tab is active, and the 'Schemas' sub-tab is selected. The status bar at the bottom indicates 'No object selected'.

```
-- Identify the most common pizzas size ordered.

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

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28





List the 5 most ordered pizzas types along with their quantities.



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The Navigator pane on the left shows a database schema with tables, views, stored procedures, and functions. The main editor displays a SQL query in 'SQL File 7'. The query is as follows:

```
1  -- List the 5 most ordered pizzas types along with their quantities.
2
3  •  SELECT
4      pizza_types.name, SUM(order_details.quantity) AS quantity
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9      JOIN
10     order_details ON order_details.pizza_id = pizzas.pizza_id
11  GROUP BY pizza_types.name
12  ORDER BY quantity DESC
13  LIMIT 5;
```

The Result Grid at the bottom shows the following data:

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





Determine the distribution of orders by hour of the day.



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar displays a 'SCHEMAS' tree with a filter 'Filter objects'. The 'pizza_hut' schema is expanded, showing tables: order_details, orders, pizza_types, and pizzas. The main editor shows a SQL query in 'SQL File 8*':

```
2
3 • SELECT
4     pizza_types.category,
5     SUM(order_details.quantity) AS quantity
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.category
13 ORDER BY quantity DESC;
14
```

The bottom panel shows the 'Result Grid' with a 'Filter Rows' field and 'Export' and 'Wrap Cell Content' options. The results are displayed in a table:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

The bottom status bar indicates 'Schema: pizza_hut'.





join relevant tables to find the category-wise distribution of pizzas



The screenshot shows a SQL IDE interface with a schema browser on the left and a query editor on the right. The schema browser displays the 'pizza_hut' schema with tables: order_details, orders, pizza_types, pizzas, Views, Stored Procedures, Functions, pizzas, student, and sys. The query editor contains the following SQL code:

```
-- Determine the distribution of orders by hour of the day.
select hour(order_time), count(order_id) from orders
group by hour(order_time);
```

The results are displayed in a table with the following data:

hour(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1





Group the order by date and calculate the average number of pizzas ordered per day



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar contains a 'Navigator' pane with a search box and a tree view of schemas. The 'pizza_hut' schema is expanded, showing tables (order_details, orders, pizza_types, pizzas), views, stored procedures, and functions. The main editor displays a SQL query in 'SQL File 10*':

```
1  -- join relevant table stp find the category-wise distribution of pizzas
2
3  •  select category , count(name) from pizza_types
4     group by category;
```

Below the editor is a 'Result Grid' tab with a 'Filter Rows' input and an 'Export' button. The result grid shows the following data:

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

The bottom of the IDE shows 'Administration' and 'Information' tabs. The 'Administration' tab is active, showing 'Schema: pizza_hut'.





Determine top 3 most ordered pizza types based on revenue.



File Edit View Query Database Server Tools Scripting Help

SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* x SQL File 12*

Navigator: Schemas Filter objects

- ankita
- ccdb
- pizza_hut**
 - Tables
 - order_details
 - orders
 - pizza_types
 - pizzas
 - Views
 - Stored Procedures
 - Functions
- pizzas
- student
- sys

```
1 -- Group the order by date and calculate the average number of pizzas ordered per day.
2
3 SELECT
4     AVG(quantity)
5 FROM
6     (SELECT
7         orders.order_date, SUM(order_details.quantity) AS quantity
8     FROM
9         orders
10    JOIN order_details ON orders.order_id = order_details.order_id
11   GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

AVG(quantity)
138.4749

Administration Schemas Information

Schema: pizza_hut





Determine top 3 most ordered pizza types based on revenue.



The screenshot shows a SQL IDE interface with a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help) and a toolbar. The left sidebar displays a 'SCHEMAS' tree with a search filter 'Filter objects'. The tree includes databases like 'ankita', 'ccdb', and 'pizza_hut'. Under 'pizza_hut', there are 'Tables' (order_details, orders, pizza_types, pizzas), 'Views', 'Stored Procedures', 'Functions', and other databases like 'pizzas', 'student', and 'sys'. The main editor shows a SQL query in 'SQL File 12*' with line numbers 1 through 10. The query is:
1 -- Determine top 3 most ordered pizza types based on revenue.
2
3 • select pizza_types.name ,
4 sum(order_details.quantity * pizzas.price) as revenue
5 from pizza_types join pizzas
6 on pizzas.pizza_type_id = pizza_types.pizza_type_id
7 join order_details
8 on order_details.pizza_id=pizzas.pizza_id
9 group by pizza_types.name order by revenue desc limit 3;
10
The bottom section shows the 'Result Grid' with columns 'name' and 'revenue'. It contains three rows of data. The 'Administration' tab is active, and the 'Schema: pizza_hut' is selected at the bottom left.

```
1 -- Determine top 3 most ordered pizza types based on revenue.
2
3 • select pizza_types.name ,
4 sum(order_details.quantity * pizzas.price) as revenue
5 from pizza_types join pizzas
6 on pizzas.pizza_type_id = pizza_types.pizza_type_id
7 join order_details
8 on order_details.pizza_id=pizzas.pizza_id
9 group by pizza_types.name order by revenue desc limit 3;
10
```

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

Schema: pizza_hut



Key Insights



1. Total Orders: 21,350
2. Total Revenue: \$817,860.05
3. Highest Priced Pizza: The Greek Pizza at \$35.95
4. Most Common Pizza Size Ordered: Large (18,526 orders).
5. Top 5 Most Ordered Pizzas : -
 - * The Classic Deluxe Pizza (2,453)
 - * The Barbecue Chicken Pizza (2,432)
 - * The Hawaiian Pizza (2,422)
 - * The Pepperoni Pizza (2,418)
 - * The Thai Chicken Pizza (2,371)
6. Top Revenue-Generating Pizza: The Thai Chicken Pizza at \$43,434.25
7. Order Distribution by Hour:
8. Peak hours: 12:00 (2,520 orders), 13:00 (2,455 orders), 18:00 (2,399 orders)



THANK YOU

