<https://github.com/PLP-Database-Design/week-4-assignment-advanced-sql-queries-and-aggregations-Machuge27.git>  
  
1. Data Manipulation

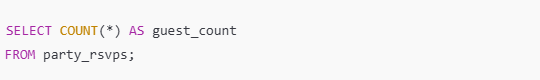
**🔄 Advanced SQL Queries and Aggregations 🛠️**

🌟 Welcome to Level-Up SQL: Advanced Queries Edition! 🌟

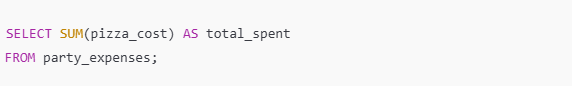
Today, we’ll channel your inner data detective and teach you how to uncover insights, spot trends, and flex your SQL superpowers with aggregation, grouping, and filtering. Let's make data analysis fun and a little extra!

**1. Aggregation Functions: The VIP Guests**

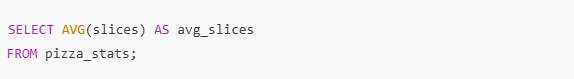
Think of aggregation functions as the cool tricks to sum, count, or average your data. They’re the foundation of your SQL powers.

COUNT: Ever wondered how many people RSVP’d to your party? Use COUNT to get the headcount.  


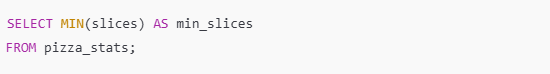
* Output: Total number of guests. 🎉

SUM: Add up your expenses, like calculating how much you spent on pizza.  


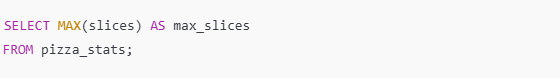
* Output: Your bank account might cry. 💸

AVG: What’s the average pizza slice eaten per guest?  


* Output: The truth about your hungry friends. 🍕
* MIN & MAX:
* Who ate the fewest slices?



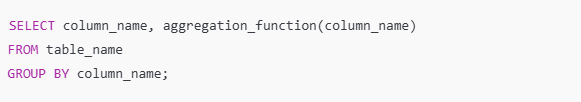
* Who’s the slice king/queen?



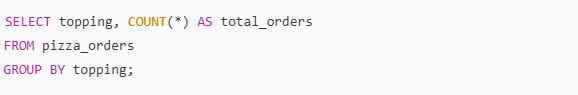
**2. GROUP BY: Squad Goals**

GROUP BY is like dividing your data into squads based on a common factor. Let’s say you’re analyzing pizza orders grouped by topping.

* Syntax breakdown:



* Example: Total pizzas sold per topping.

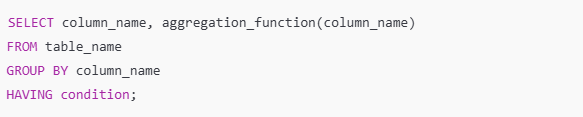


💡 Pro Tip: Without GROUP BY, SQL would mix up the squads, and nobody wants that chaos.

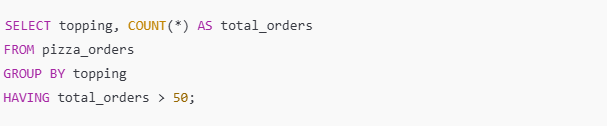
**3. HAVING: The Party Filter**

Now that you’ve grouped your data, you might want to filter out the underperforming squads. Enter HAVING, the bouncer for your grouped data.

* Syntax breakdown:



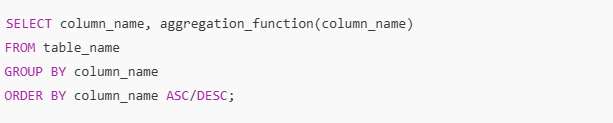
* Example: Only show toppings with more than 50 orders.

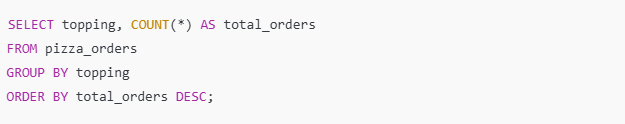


**4. ORDER BY: The Setlist**

You’ve got your squads and filtered them down to the best ones. Now, let’s sort them! Use ORDER BY to organize your results.

* Syntax breakdown:



* Example: Sort toppings by popularity (descending).  
  

2. Handling NULL Values.

**🌟 Understanding NULL Values in Databases**

In the realm of databases, a **NULL value** represents the absence of any data in a field. It's crucial to understand that NULL is not the same as zero or spaces; it signifies that no value was provided for a particular field. Let's dive deeper into what NULL values mean and how to work with them effectively.



**🤔 What is a NULL Value?**

* **Definition:** A field with a NULL value is essentially empty, indicating no data entry. This often occurs when a field is optional, and no value is provided during record creation or updating.
* **Key Distinction:** NULL is different from zero (0) or a field containing spaces. While zero and spaces are actual values, NULL signifies that the field was left blank intentionally.

**🧪 How to Test for NULL Values**

Testing for NULL values requires specific operators because traditional comparison operators like =, <, or <> don't work with NULLs. Instead, use the following operators:

* **IS NULL:** Checks if a field is NULL.
* **IS NOT NULL:** Checks if a field contains any value other than NULL.



**IS NULL Syntax**

To retrieve records where a specific column has a NULL value, use this query:

SELECT column\_names

FROM table\_name

WHERE column\_name IS NULL;

**IS NOT NULL Syntax**

To find records where a column has any value other than NULL, use this query:

SELECT column\_names

FROM table\_name

WHERE column\_name IS NOT NULL;

**📋 Summary**

* **IS NULL Operator:** Returns true if the specified value is NULL.
* **IS NOT NULL Operator:** Returns true if the specified value is not NULL.

Understanding and handling NULL values is essential for accurate data analysis and manipulation. By mastering these techniques, you can ensure your queries return precise results, leading to better insights and decision-making in your database management tasks. Keep practising these concepts to enhance your database skills! 🌟

3. <https://youtu.be/-F47R8uAV2U>

https://youtu.be/OB2leB2iZ6U