

# SQL

## SELECTING MORE

Alish Bista - 08/25/2025

COUNT

# COUNT()

We'll Be Counting Stars.... And More

- Counts the number of records
- COUNT(\*) shows total number of rows in a field
- COUNT(field\_name) shows number of records with a value in a field

```
SELECT COUNT(*) AS total_records  
FROM orders
```

DISTINCT

# DISTINCT

## The Chosen Ones

- It removes duplicates to return **unique** values

```
SELECT o_orderpriority  
FROM orders;
```

```
SELECT DISTINCT o_orderpriority  
FROM orders;
```

**FILTER**

# FILTER

I LIKE IT! MINUS!!!

- Filters records based on conditions
- **Comparison Operators:** <, >, =, <=, >=, <>
- **Criteria Combination:** AND, OR, BETWEEN
- **Text Pattern:** LIKE, NOT LIKE, IN
- **Wildcards:** \_, %

NULL

# NULL

## Much Ado About Nothing

- NULL values are missing values
- Common in real world data
- Keywords in SQL for handling them:
  - IS NULL
  - IS NOT NULL

# AGGREGATES

# AGGREGATES

All Together Now

- **Numerical Fields**

- AVG()
- SUM()

- **Various Data Types**

- MIN()
- MAX()
- COUNT()

**ROUND**

# ROUND

## Approximately Speaking

- **Syntax:** `ROUND(number_to_round, decimal_places)`
- Rounding to whole number:  
`ROUND(number_to_round)`  
`ROUND(number_to_round, 0)`
- Rounding with negative parameter:  
`ROUND(number_to_round, - nearest_multiple_of_ten)`

# SORTING

# SORTING

## First Things First

- Adds order to the result set
- Keywords that help to provide order:
  - ORDER BY
  - ASC, DESC

# GROUPING

# GROUPING

## Birds of a Feather

- Generates aggregates for groups coming from one field or combination of many fields
- Keyword used: **GROUP BY**
- Declaration is required whenever using aggregate functions in combination with other fields

**HAVING**

# HAVING

## Double Check

- Filtering for grouped records
- Where is WHERE? WHERE filters in the context of individual records
- Uses Comparison operators similar to WHERE