

Module 2 - Lecture 2

Introduction to ordering, grouping, and other functions



REVIEW

- Databases
 - What they good for?
 - What is the type we are going to be using?
- SQL
 - DDL
 - DML
 - DCL
- PostgreSQL & DbVisualizer
- SELECT statements



Ordering Results

A result set can be sorted using the **ORDER BY** syntax.

- Sort columns must exist in the table being queried or can be aliased columns.
- Multiple column names can be provided which assigns a priority sort

```
SELECT    column1, column2
FROM      table
ORDER BY column1 [ASC | DESC],
           column2 [ASC | DESC];
```



Limiting Results

A result set can be limited to N results using the **LIMIT** syntax.

```
SELECT    column1, column2  
FROM      table  
LIMIT     { number | ALL } ;
```



String Concatenation

We can concatenate the values across multiple columns into a single field.

- This is done with the || operator. NOTE: this is different than OR.

```
SELECT      (column1 || ', ' || column2)
FROM        table;
```



Aggregate Functions

We can aggregate the values across multiple rows into a single result.

- **ABS** returns the absolute value of a numeric result
- **AVG** returns the average value of a numeric column
- **SUM** returns the total sum of a numeric column
- **COUNT** returns the number of rows matching criteria
- **MIN** returns the smallest value of the selected column
- **MAX** returns the largest value of the selected column

```
SELECT  AVG (expression)  FROM table;
```



Grouping Results

Grouping data is the process of combining columns with duplicate values.

- The **GROUP BY** clause can be used in conjunction with a SELECT statement and aggregate functions to collect data across multiple records.

```
SELECT      column1, AVG (expression)  
FROM        table  
[WHERE]      [...]  
GROUP BY    column1;
```



BONUS



Subqueries

A **subquery** is referred to as an inner query and can provide the results of one query as input to another.

- Often used in the WHERE clause
- Can only return one item in the SELECT clause

```
SELECT column1, ... , [column-n]
FROM table1
WHERE column1 {IN|NOT IN} (SELECT column1
                           FROM table2
                           [WHERE] [...]);
```



QUESTIONS?

