School of Computer Science

Web and Database Computing 2019

Lecture 31: SQL Ordering and Filtering Results

adelaide.edu.au

seek LIGHT

Ordering/Limiting results

Sorting results using ORDER BY

A common desire is for the results of a query to be sorted on a particular column.

- This can be achieved using ORDER BY

```
SELECT * FROM Customers
  ORDER BY Country;
```

- You can order Ascending (A-Z) or Descending (Z-A)
- You can also specify secondary and tertiary columns

```
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;
```

Restricting results

While most of the queries and databases we've been working with only have a few entries, some queries on larger databases could return thousands or millions of results.

- Large result sets can have performance and bandwidth consequences.
- We can limit the total number of results using LIMIT

```
SELECT * FROM Customers
  LIMIT 100;
```

- Use in conjunction with ORDER BY to ensure key results not omitted:

```
SELECT * FROM Customers
  ORDER BY Country ASC, CustomerName DESC
  LIMIT 100;
```

Removing Duplicates

Some queries may return duplicate results, especially where a Primary Key is not included in the columns returned.

- Often we wnat to exclude duplicate results.
- We can use the DISTINCT keyword to only return unique results.

```
SELECT DISTINCT Country FROM Customers;
```

- Can work with multiple columns:

```
SELECT DISTINCT Country, City FROM Customers;
```

Working With Aggregate Functions

Some functions rely on multiple rows to generate a result.

- E.g. COUNT, MAX, MIN, SUM, AVG
- These are known as aggregate functions

If you want to use an aggregate function as part of a WHERE clause however, this won't work as the conditions used in WHERE are evalueated row-by-row.

- E.g. Get a list of the cities where your customers live, but only if at least 10 of your customers live there.

Instead we can use the HAVING and GROUP BY clauses.

```
SELECT City
    FROM Customers
    GROUP BY City
    HAVING COUNT(CustomerID) >= 10;
```

Working With Aggregate Functions

The GROUP BY clause is used to group common values together in a column.

- These groups are necessary for aggregate functions to work.
- Once we have our groups, aggregate functions can be used on the data.

The HAVING clause replaces the WHERE clause for aggregate data.

- It is evaluated after the WHERE clause on grouped/aggregate data

```
SELECT City
    FROM Customers
    WHERE Country = 'Germany'
    GROUP BY City
    HAVING COUNT(CustomerID) >= 10;
```

Questions?


THE UNIVERSITY
of ADELAIDE

What's happening

Due:

- Prac Exercise 8 due Friday.

This week:

- Workshops
- Livestream tonight 8:30
- Optimisation

Further learning:

- Keep working on your group projects
- Follow the w3schools SQL tutorial