

Ordering and Filtering

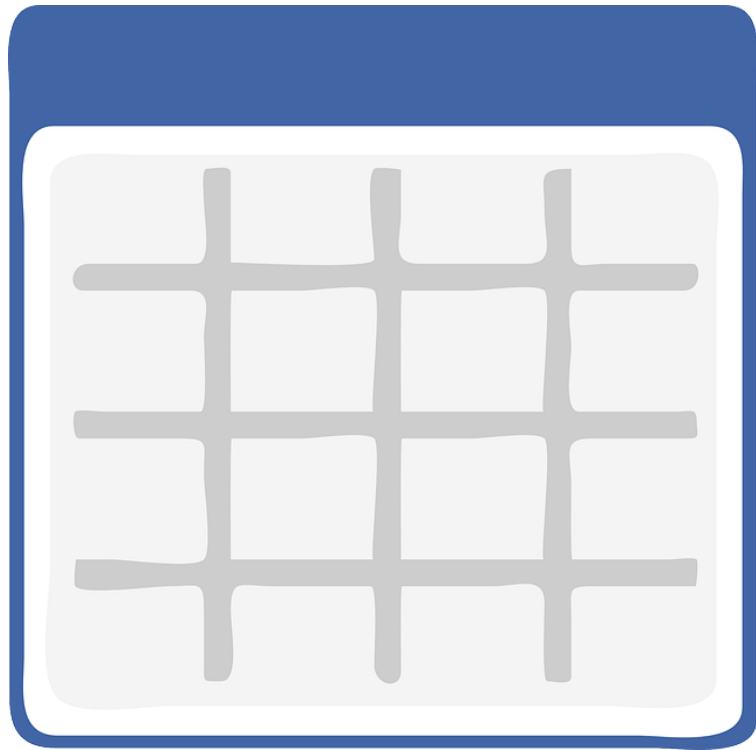
INTRODUCTION TO SQL SERVER



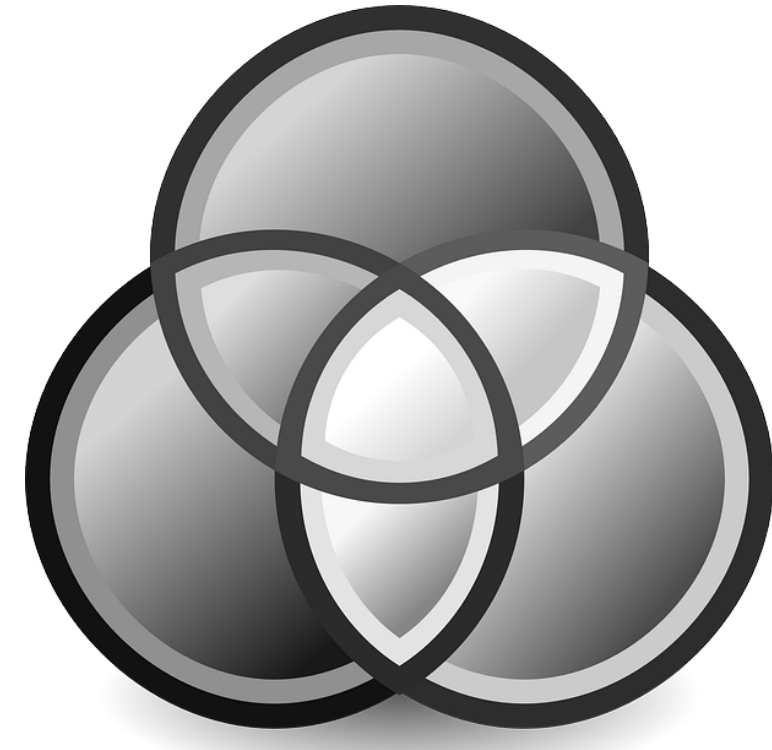
John MacKintosh
Instructor

Order! Order!

- Tables comprise of rows and columns
- Queries return *sets*, or *subsets*



- Sets have no inherent order
- If order is important, use `ORDER BY`



```
SELECT TOP (10) prod_id, year_intro
FROM products
-- Order in ascending order
ORDER BY year_intro, product_id;
```

```
+-----+-----+
| product_id | year_intro |
+-----+-----+
| 36         | 1981      |
| 37         | 1982      |
| 38         | 1983      |
| 39         | 1984      |
| 40         | 1984      |
| 41         | 1984      |
| 52         | 1985      |
| 43         | 1986      |
| 44         | 1987      |
| 54         | 1987      |
+-----+-----+
```

```
SELECT TOP (10) product_id, year_intro
FROM products
-- Order year_intro in descending order
ORDER BY year_intro DESC, product_id;
```

```
+-----+-----+
| product_id | year_intro |
+-----+-----+
| 158        | 2015      |
| 173        | 2015      |
| 170        | 2014      |
| 171        | 2014      |
| 172        | 2014      |
| 144        | 2013      |
| 146        | 2013      |
| 147        | 2013      |
| 148        | 2013      |
| 149        | 2013      |
+-----+-----+
```

```
SELECT
  TOP (10) channels,
  year_intro
FROM products
-- Order in different directions
ORDER BY
  year_intro DESC,
  channels;
```

```
+-----+-----+
| channels | year_intro |
+-----+-----+
| 35       | 2015      |
| 74       | 2015      |
| 29       | 2014      |
| 45       | 2014      |
| 48       | 2014      |
| 12       | 2013      |
| 13       | 2013      |
| 14       | 2013      |
| 22       | 2013      |
| 24       | 2013      |
+-----+-----+
```

```
SELECT
  TOP (10) channels,
  year_intro
FROM products
-- Both columns in descending order
ORDER BY
  year_intro DESC,
  channels DESC;
```

```
+-----+-----+
| channels | year_intro |
+-----+-----+
| 74       | 2015      |
| 35       | 2015      |
| 48       | 2014      |
| 45       | 2014      |
| 29       | 2014      |
| 837      | 2013      |
| 642      | 2013      |
| 561      | 2013      |
| 491      | 2013      |
| 198      | 2013      |
+-----+-----+
```

```
SELECT city_id, name_alias
FROM invoice
-- Ordering text (Ascending order)
ORDER BY name_alias;
```

city_id	name_alias
48	Amsterdam
59	Bangalore
36	Berlin
38	Berlin
42	Bordeaux
23	Boston
13	Brasília
8	Brussels
45	Budapest
56	Buenos Aires

```
SELECT city_id, name_alias
FROM invoice
-- Ordering text (Descending order)
ORDER BY name_alias DESC;
```

city_id	name_alias
33	Yellowknife
32	Winnipeg
49	Warsaw
7	Vienne
15	Vancouver
27	Tucson
29	Toronto
2	Stuttgart
51	Stockholm
55	Sydney

What if we only wanted to return rows that met certain criteria?

```
SELECT customer_id, total
FROM invoice
WHERE total > 15;
```

First 3 customers with invoice value > 15

```
+-----+-----+
| customer_id | total   |
+-----+-----+
| 57          | 17.91   |
| 7           | 18.86   |
| 45          | 21.86   |
+-----+-----+
```

```
-- Rows with points greater than 10
WHERE points > 10

-- Rows with points less than 10
WHERE points < 10

-- Rows with points greater than or equal to 10
WHERE points >= 10

-- Rows with points less than or equal to 20
WHERE points <= 20

-- Character data type
WHERE country = 'Spain'

-- Date data type
WHERE event_date = '2012-01-02'
```



```
SELECT customer_id, total
FROM invoice
-- Testing for non-equality
WHERE total <> 10;
```

```
+-----+-----+
| customerid | total |
|-----+-----|
| 2          | 1.98  |
| 4          | 3.96  |
| 8          | 5.94  |
| 14         | 8.91  |
| 23         | 13.86 |
| 37         | 0.99  |
+-----+-----+
```

Between

```
SELECT customer_id, total
FROM invoice
WHERE total BETWEEN 20 AND 30;
```

customerid	total
45	21.86
46	21.86
26	23.86
6	25.86

```
SELECT customer_id, total
FROM invoice
WHERE total NOT BETWEEN 20 AND 30;
```

customerid	total
2	1.98
4	3.96
8	5.94
14	8.91

What is NULL?

- NULL indicates there is no value for that record
- NULLs help highlight gaps in our data

SELECT

TOP (6) total,
billing_state

FROM invoice

WHERE billing_state **IS** NULL;

```
+-----+-----+
| total | billing_state |
+-----+-----+
| 1.98  | NULL         |
| 3.96  | NULL         |
| 5.94  | NULL         |
| 0.99  | NULL         |
| 1.98  | NULL         |
| 1.98  | NULL         |
+-----+-----+
```

SELECT

TOP (6) total,
billing_state

FROM invoice

WHERE billing_state **IS NOT** NULL;

```
+-----+-----+
| total | billing_state |
+-----+-----+
| 8.91  | AB            |
| 13.96 | MA            |
| 5.94  | Dublin        |
| 0.99  | CA            |
| 1.98  | WA            |
| 1.98  | CA            |
+-----+-----+
```