

# Querying Relational Database

UNION

JOIN 2 Tables

JOIN Multiple Tables

*Matthew Morris*

Git: *Morrisdata*

*MatthewMorris.DA@gmail.com*



# Previously in Data Analytics

FILTERS = , !=, >, <

IN, NOT IN, BETWEEN, LIKE, NOT LIKE

SUM, MIN, MAX, COUNT

GROUP BY, HAVING

COMMENTING



---

**QUERY A RELATIONAL DATABASE**

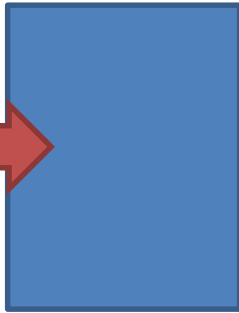
---

**UNION**



SELECT  
FROM  
JOIN  
ON  
WHERE  
GROUP BY  
HAVING  
UNION  
ORDER BY  
LIMIT

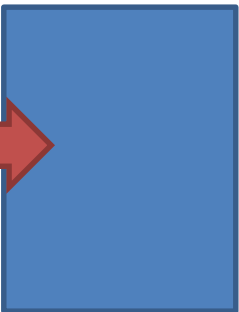
**FY17**



```
SELECT fy, pd, store_name, week1,  
week2, week3 week4  
FROM FY17
```

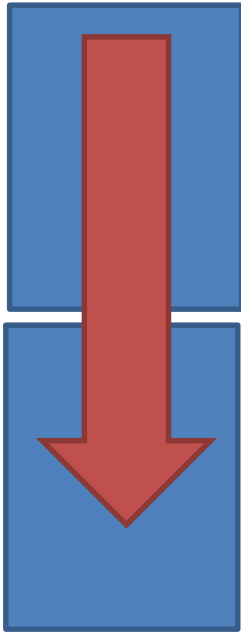
**UNION**

**FY18**



```
SELECT fy, pd, store_name, week1,  
week2, week3 week4  
FROM FY18
```

**FY17**



**FY18**

```
SELECT fy, pd, store_name, week1,  
week2, week3 week4  
FROM FY17  
UNION  
SELECT fy, pd, store_name, week1,  
week2, week3 week4  
FROM FY18
```

**COLUMNS**  
**CONDITIONS**  
**UNION and UNION ALL**  
**ORDER BY**

---

QUERY A RELATIONAL DATABASE

---

# JOIN 1 Table



# Joins



Querying a Relational Database

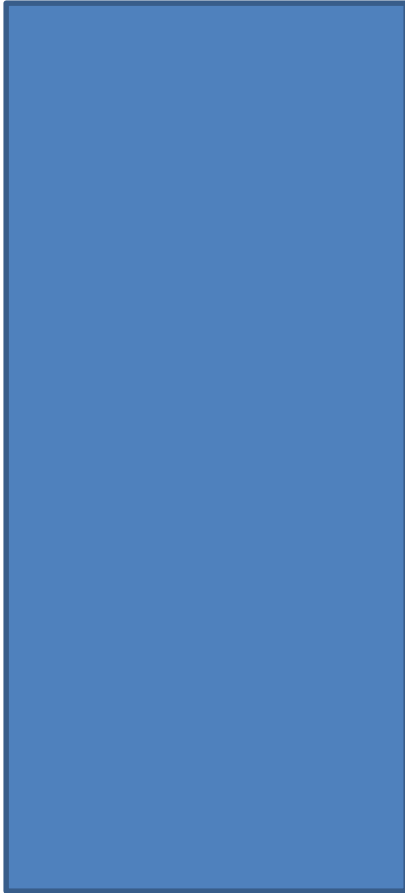


# Joins



Query a Relational Database

**LEFT/PRIMARY**



**RIGHT/SECONDARY**



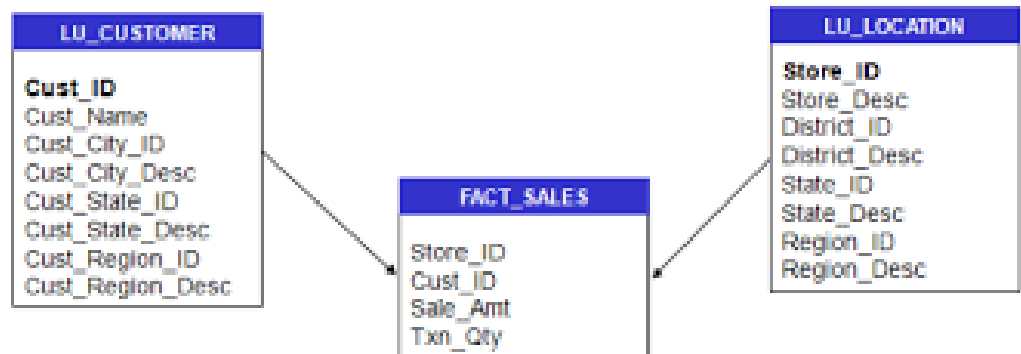
What table is the transaction table?

If you wanted to link on the lowest level of detail to the other tables what fields would you use?

# Joins

SALES			
<u>FIELD</u>	<u>TYPE</u>	<u>LENGTH</u>	
ID	PK	1	
ARTIST	Char	25	
SONG	Char	225	
ALBUM	Char	225	

**Create a rough sketch with how  
The Iowa Liquor Sales Database  
Would JOIN**



**Querying a Relational Database**

```
SELECT a.item, b.description, a.sales
```

```
FROM sales a
```

```
JOIN products b
```

```
ON a.item=b.item
```

1. Create separate queries to join each table to Sales
  - a. Products to Sales
  - b. County to Sales
  - c. Stores to Sales
2. Use this as an opportunity to bring fields in from both tables.
3. Try out some aggregations or Wild card searches. Stretch with an Aggregate and a Group by

```
SELECT c.field, a.field, b.field, a.field, c.field
```

```
FROM table1 a
```

```
JOIN table2 b
```

```
ON a.field=b.field
```

```
JOIN table3 c
```

```
ON a.field=c.field
```



Using Sales as the primary table, create links to the all of the other tables in the Iowa liquor database. Result should be 1 query with several JOINS.

Bring back county from the county table, store from the stores table, name from the stores table, case\_cost from the products table and total from the sales table. Limit to 1000.

```
SELECT b.field1, a.field2, a.field3, c.field4  
FROM table1 a  
INNER JOIN table2 b  
ON a.field1 = b.field1  
INNER JOIN table3 c  
ON a.field1 = c.field1  
LIMIT 100
```

# Q & A

---

*“Sometimes questions are more important  
than answers.”*

– Nancy Willard

# Conclusion

Find the lowest level detail between tables.

Place the transactional or largest table on the left



# EXIT TICKET

**CLASS :** Querying a Relational Database

**QUESTION:**

**Why do you need an alias when using Joins?**

