SQL – Whole Table Aggregations

TRANSACTIONS

Transaction_ID	Customer_Id	Channel	Product	Price	Discount
1000123	60067	Web	Book	9.95	
1000124	12345	Store	Book	11.95	
1000125	23451	Store	DVD	14.95	
1000126	70436	Reseller	DVD	19.95	5
1000127	66772	Store	Magazine	3.25	
1000128	60067	Web	Book	29.95	
1000129	72045	Web	DVD	9.95	
1000130	82371	Reseller	Magazine	2.5	0.25
1000131	12345	Store	Book	7.95	

```
SELECT COUNT(*)
FROM TRANSACTIONS
```

9

OR

COUNT(*)
9

```
SELECT COUNT(*) AS NUM_ROWS
FROM TRANSACTIONS
```

NUM_ROWS
9

- NUM_ROWS is the 'Alias' for COUNT(*) and is designated using 'AS'

Be Boulder.
University of Colorado Boulder

SQL – Shorthand using ‘Aliases’

Column Aliases:

```
SELECT COUNT(*) AS NUM_ROWS
FROM TRANSACTIONS
```

Table Aliases:

```
SELECT CHANNEL, PRODUCT, PRICE
FROM TRANSACTIONS
```

OR

```
SELECT TRANSACTIONS.CHANNEL, TRANSACTIONS.PRODUCT,
       TRANSACTIONS.PRICE
FROM TRANSACTIONS
```

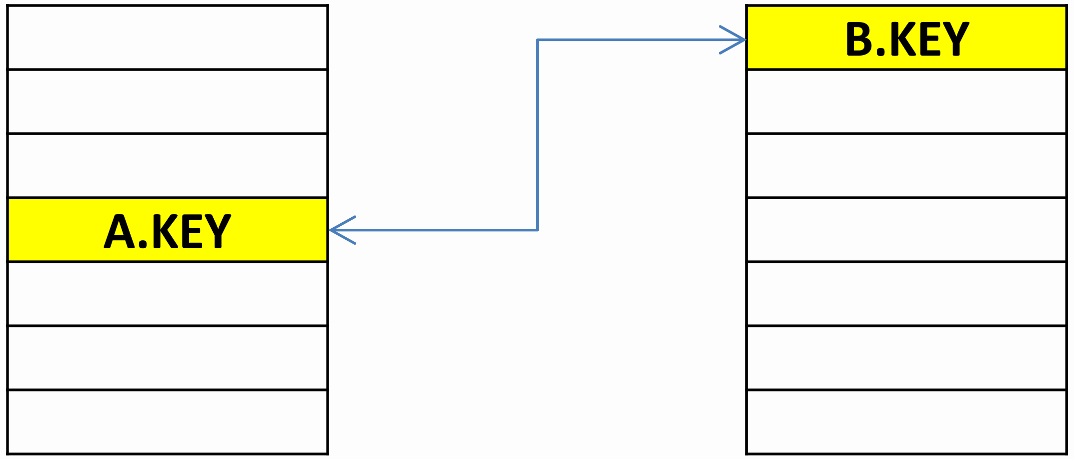
OR

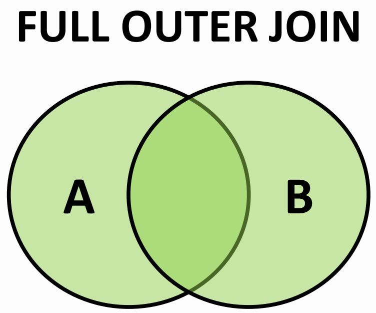
```
SELECT a.CHANNEL, a.PRODUCT, a.PRICE
FROM TRANSACTIONS a
```

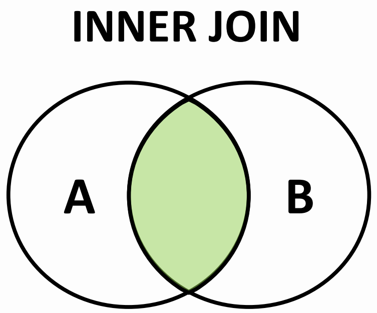
- In this case, ‘a’ is used as an alias for the table

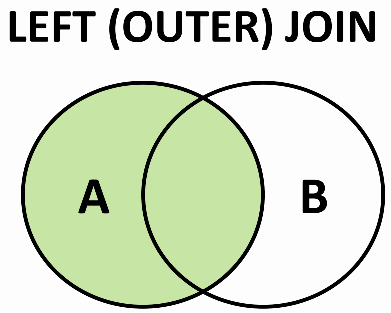
SQL – JOINing Tables

The real power of SQL is the ability to link tables across a relational database structure









SQL – JOIN Statements

```
SELECT a.FIELD_1, ..., a.FIELD_N, b.FIELD_1, ..., b.FIELD_N
FROM TABLE_1 a
FULL OUTER JOIN TABLE_2 b
ON a.KEY = b.KEY
```

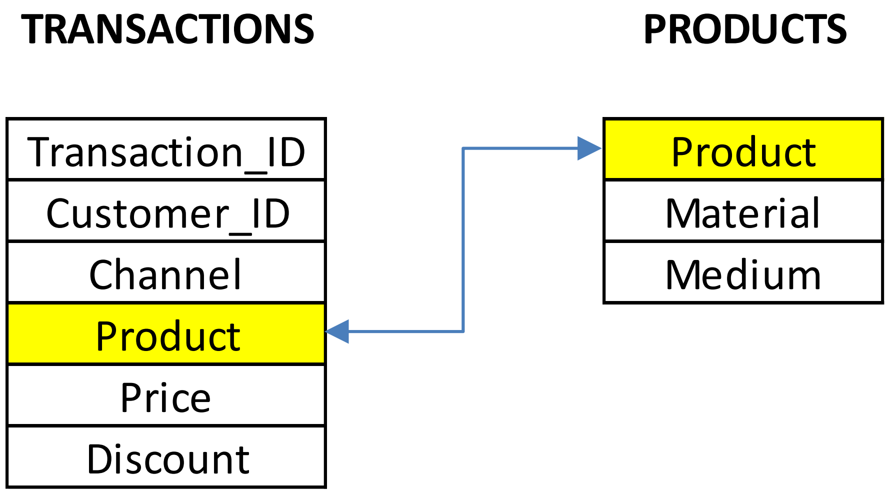
```
SELECT a.FIELD_1, ..., a.FIELD_N, b.FIELD_1, ..., b.FIELD_N
FROM TABLE_1 a
INNER JOIN TABLE_2 b
ON a.KEY = b.KEY
```

```
SELECT a.FIELD_1, ..., a.FIELD_N, b.FIELD_1, ..., b.FIELD_N
FROM TABLE_1 a
LEFT JOIN TABLE_2 b
ON a.KEY = b.KEY
```

SQL – Identifying the JOIN field

Transaction_ID	Customer_Id	Channel	Product	Price	Discount
1000123	60067	Web	Book	9.95	
1000124	12345	Store	Book	11.95	
1000125	23451	Store	DVD	14.95	
1000126	70436	Reseller	DVD	19.95	5
1000127	66772	Store	Magazine	3.25	
1000128	60067	Web	Book	29.95	
1000129	72045	Web	DVD	9.95	
1000130	82371	Reseller	Magazine	2.5	0.25
1000131	12345	Store	Book	7.95	

Product	Material	Medium
Book	Stock Paper	Visual
DVD	Plastic	Audiovisual
Magazine	Glossy Paper	Visual
CD	Plastic	Audio
Newspaper	Newsprint	Visual
MP3	Digital	Audio



SQL – JOIN Statements

Let’s say I want more information about the products that were actually purchased:

```
SELECT a.*, b.*
FROM TRANSACTIONS a
LEFT JOIN PRODUCTS b
ON a.PRODUCT = b.PRODUCT
```

Transaction_ID	Customer_Id	Channel	Product	Price	Discount	Material	Medium
1000123	60067	Web	Book	9.95		Stock Paper	Visual
1000124	12345	Store	Book	11.95		Stock Paper	Visual
1000125	23451	Store	DVD	14.95		Plastic	Audiovisual
1000126	70436	Reseller	DVD	19.95	5	Plastic	Audiovisual
1000127	66772	Store	Magazine	3.25		Glossy Paper	Visual
1000128	60067	Web	Book	29.95		Stock Paper	Visual
1000129	72045	Web	DVD	9.95		Plastic	Audiovisual
1000130	82371	Reseller	Magazine	2.5	0.25	Glossy Paper	Visual
1000131	12345	Store	Book	7.95		Stock Paper	Visual

SQL – JOIN Statements

Why not an INNER JOIN?

```
SELECT a.*, b.*
FROM TRANSACTIONS a
INNER JOIN PRODUCTS b
ON a.PRODUCT = b.PRODUCT
```

- In this case, the query would actually return the same result
- However, if a product were missing from the PRODUCT table, those transactions would be eliminated
- Sometimes this is desirable, sometimes not, depending on the question you are trying to answer!

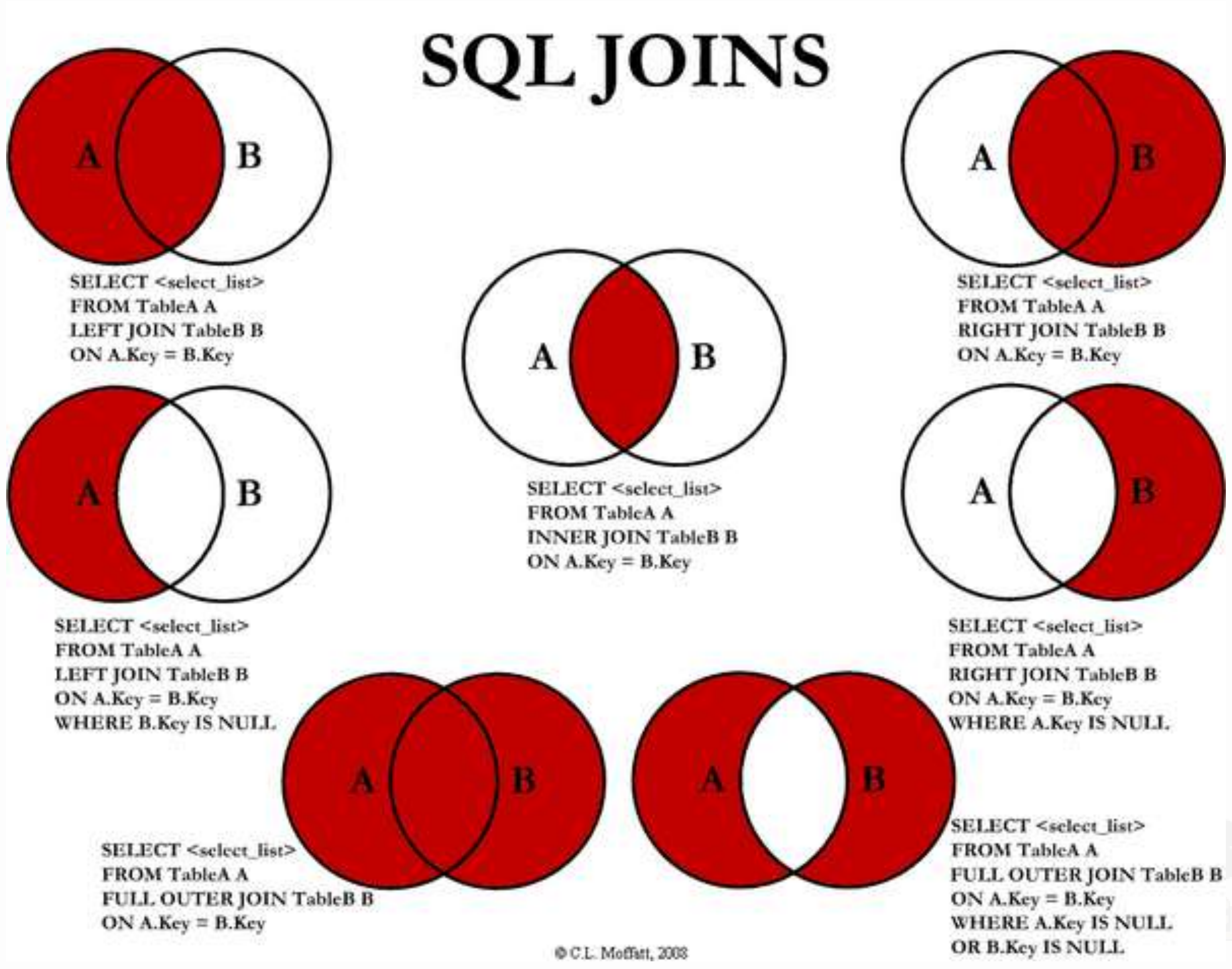
SQL – JOIN Statements

Why not a FULL OUTER JOIN?

```
SELECT a.*, b.*
FROM TRANSACTIONS a
FULL OUTER JOIN PRODUCTS b
ON a.PRODUCT = b.PRODUCT
```

Transaction_ID	Customer_Id	Channel	Product	Price	Discount	Material	Medium
1000123	60067	Web	Book	9.95		Stock Paper	Visual
1000124	12345	Store	Book	11.95		Stock Paper	Visual
1000125	23451	Store	DVD	14.95		Plastic	Audiovisual
1000126	70436	Reseller	DVD	19.95	5	Plastic	Audiovisual
1000127	66772	Store	Magazine	3.25		Glossy Paper	Visual
1000128	60067	Web	Book	29.95		Stock Paper	Visual
1000129	72045	Web	DVD	9.95		Plastic	Audiovisual
1000130	82371	Reseller	Magazine	2.5	0.25	Glossy Paper	Visual
1000131	12345	Store	Book	7.95		Stock Paper	Visual
			Newspaper			Newsprint	Visual
			MP3			Digital	Audio

SQL – More JOINing Logic



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

SQL – JOIN Statements

Extended Example:

- Return average price of products by Medium
- Exclude Resellers
- Only include Medium values where average price > 10
- Sort results from highest to lowest average price

```
SELECT b.MEDIUM, AVG(a.PRICE) AS AVG_PRICE
FROM TRANSACTIONS a
LEFT JOIN PRODUCTS b
ON a.PRODUCT = b.PRODUCT
WHERE a.CHANNEL <> 'RESELLER'
GROUP BY b.MEDIUM
HAVING AVG_PRICE > 12.50
ORDER BY AVG_PRICE DESC
```

Medium	AVG_PRICE
Visual	12.61
Audiovisual	12.45

Be Boulder.

University of Colorado Boulder