

Session 6-4 Outer Join

Kyunghee Lee, PhD



WAYNE STATE
Mike Ilitch School of Business

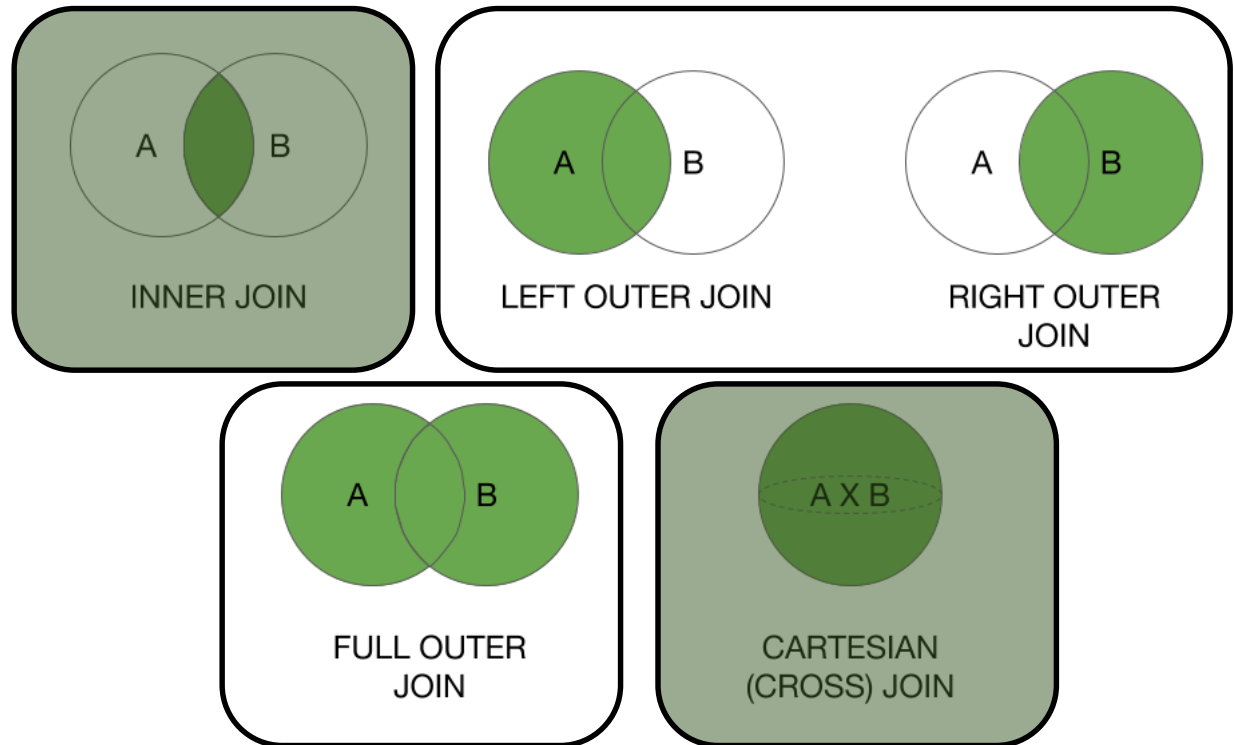
Types of Joins

Cartesian Joins

Inner Joins

- Equality Joins
- Non-equality Joins
- Self Joins

Outer Joins



Outer Joins

- Use outer joins to include rows that do not have a match in the other table
- Traditional method
 - In WHERE clause, include outer join operator (+) immediately after the column name of the table with missing rows to add NULL rows
- Join method
 - In FROM clause, use FULL, LEFT, or RIGHT with OUTER JOIN keywords



Outer Joins Example

- List *all* customers from CUSTOMERS table. If a customer has placed an order, include the corresponding order# as well. *If not, display only the customer info. and leave order# as blank.*

```
SELECT *  
FROM books b, order o  
WHERE b.customer# = o.customer#
```

INNER JOIN only returns what's in common



WAYNE STATE

Mike Ilitch School of Business

Outer Joins

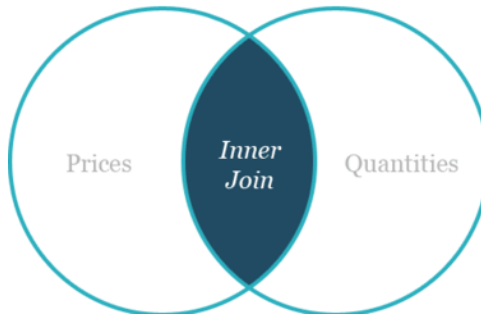


TABLE 1: PRICES		TABLE 2: QUANTITIES	
PRODUCT	PRICE	PRODUCT	QUANTITY
Potatoes	\$3	Potatoes	45
Avocados	\$4	Avocados	63
Kiwis	\$2	Kiwis	19
Onions	\$1	Onions	20
Melons	\$5	Melons	66
Oranges	\$5	Broccoli	27
Tomatoes	\$6	Squash	92

```
SELECT Prices.*, Quantities.Quantity
FROM Prices INNER JOIN Quantities
ON Prices.Product = Quantities.Product;
```

QUERY RESULT FOR INNER JOIN		
PRODUCT	PRICE	QUANTITY
Potatoes	\$3	45
Avocados	\$4	63
Kiwis	\$2	19
Onions	\$1	20
Melons	\$5	66

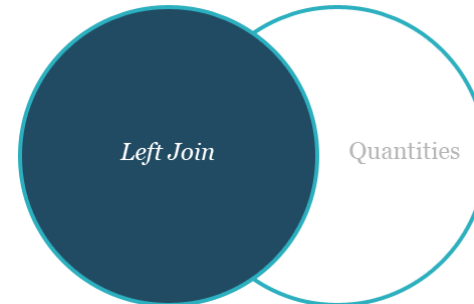


TABLE 1: PRICES		TABLE 2: QUANTITIES	
PRODUCT	PRICE	PRODUCT	QUANTITY
Potatoes	\$3	Potatoes	45
Avocados	\$4	Avocados	63
Kiwis	\$2	Kiwis	19
Onions	\$1	Onions	20
Melons	\$5	Melons	66
Oranges	\$5	Broccoli	27
Tomatoes	\$6	Squash	92

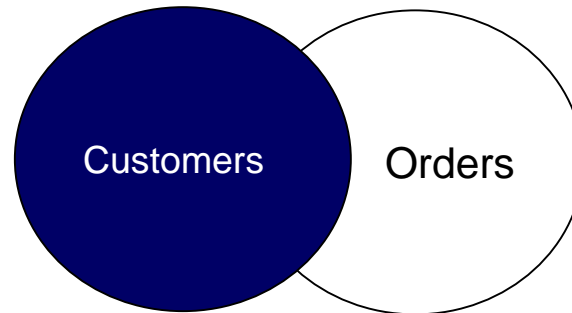
```
SELECT Prices.*, Quantities.Quantity
FROM Prices LEFT OUTER JOIN Quantities
ON Prices.Product = Quantities.Product;
```

QUERY RESULT FOR LEFT OUTER JOIN		
PRODUCT	PRICE	QUANTITY
Potatoes	\$3	45
Avocados	\$4	63
Kiwis	\$2	19
Onions	\$1	20
Melons	\$5	66
Oranges	\$5	NULL
Tomatoes	\$6	NULL



WAYNE STATE
Mike Ilitch School of Business

(Left) Outer Joins: WHERE Clause Example



```
SELECT lastname, firstname, order#  
FROM customers c, orders o  
WHERE c.customer# = o.customer#(+)  
ORDER BY c.customer#;
```

Execute

Load Script

Save Script

Cancel

Include (+) immediately after the column name of the table with missing rows to add NULL rows



WAYNE STATE

Mike Ilitch School of Business

(Left) Outer Joins: WHERE Clause Example

```
select lastname, firstname, order#  
from customers c, orders o  
where c.customer# = o.customer#(+)  
order by c.customer#
```

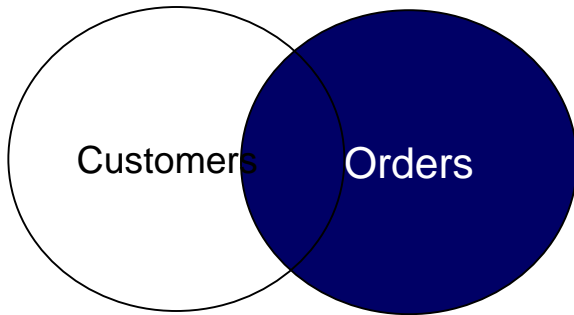
Execute Load Script Save Script Cancel

LASTNAME	FIRSTNAME	ORDER#
MORALES	BONITA	1018
MORALES	BONITA	1003
THOMPSON	RYAN	
SMITH	LEILA	1006
SMITH	LEILA	1016
PIERSON	THOMAS	1008
GIRARD	CINDY	1000
GIRARD	CINDY	1009
CRUZ	MESHIA	
IANA	TAMMY	1014
IANA	TAMMY	1007
JONES	KENNETH	1020
PEREZ	JORGE	1025
PEREZ	JORGE	1021
LUCAS	JAKE	1011
LUCAS	JAKE	1001
MCGOVERN	REESE	1002
MCKENZIE	WILLIAM	
NGUYEN	NICHOLAS	
LEE	JASMINE	1013
SHELL	STEVE	1017
DAUM	MICHELL	
NELSON	BECCA	1012
MONTIASA	GREG	1005

- Rows that only exist in Customer table
- Missing rows are added



(Right) Outer Joins: WHERE Clause Example



```
select lastname, firstname, order#
from customers c, orders o
where c.customer#(+) = o.customer#
order by c.customer#
```

Execute Load Script Save Script Cancel

LASTNAME	FIRSTNAME	ORDER#
MORALES	BONITA	1018
MORALES	BONITA	1003
SMITH	LEILA	1016
SMITH	LEILA	1006
PIERSON	THOMAS	1008
GIRARD	CINDY	1009
GIRARD	CINDY	1000
IANA	TAMMY	1014
IANA	TAMMY	1007
JONES	KENNETH	1020
PEREZ	JORGE	1025
PEREZ	JORGE	1021
LUCAS	JAKE	1011
LUCAS	JAKE	1001
MCGOVERN	REESE	1002
LEE	JASMINE	1013
SHELL	STEVE	1017
NELSON	BECCA	1012
MONTIASA	GREG	1019
MONTIASA	GREG	1005
SMITH	JENNIFER	1010
FALAH	KENNETH	1015
FALAH	KENNETH	1004

23 rows selected.



WAYNE STATE
Mike Ilitch School of Business

Outer Joins: OUTER JOIN Keyword Example

```
SELECT lastname, firstname, order#  
FROM customers c LEFT OUTER JOIN orders o  
    ON c.customer# = o.customer#  
ORDER BY c.customer#;
```

Execute

Load Script

Save Script

Cancel



WAYNE STATE

Mike Ilitch School of Business

Set Operators

- Used to combine the results of two or more SELECT statements

SET OPERATOR	DESCRIPTION
UNION	Returns the results of the combined SELECT statements and suppresses duplicates
UNION ALL	Returns the results of the combined SELECT statements but does not suppress duplicates
INTERSECT	Returns only the rows included in the results of both SELECT statements
MINUS	Removes the results of the second query that are also found in the first query and displays only the rows that are uniquely returned by only the first query

FIGURE 9-29 List of set operators

Set Operators: UNION Example

```
SELECT authorid
FROM books JOIN bookauthor USING(isbn)
WHERE category = 'FAMILY LIFE'
UNION
SELECT authorid
FROM books JOIN bookauthor USING(isbn)
WHERE category = 'CHILDREN';
```

Execute Load Script Save Script Cancel

AUTHORID
B100
F100
J100
K100
R100



WAYNE STATE

Mike Ilitch School of Business

Set Operators: INTERSECT

Example

```
SELECT customer# FROM customers  
INTERSECT  
SELECT customer# FROM orders;
```

Execute Load Script Save Script Cancel



Set Operators: MINUS Example

SELECT customer# FROM customers
MINUS
SELECT customer# FROM orders;

Execute Load Script Save Script Cancel

CUSTOMER#
1002
1006
1009
1012
1013
1016

6 rows selected.

select customer# from orders
where customer# = 1002

Execute Load Script Save Script Cancel

no rows selected



WAYNE STATE

Mike Ilitch School of Business

Wrap-up

- Outer Join
 - LEFT
 - RIGHT
- Set operators
 - UNION
 - INTERSECT
 - MINUS

