Session 6-2 Inner Join 1

Kyunghee Lee, PhD

WAYNE STATE
Mike Ilitch School of Business

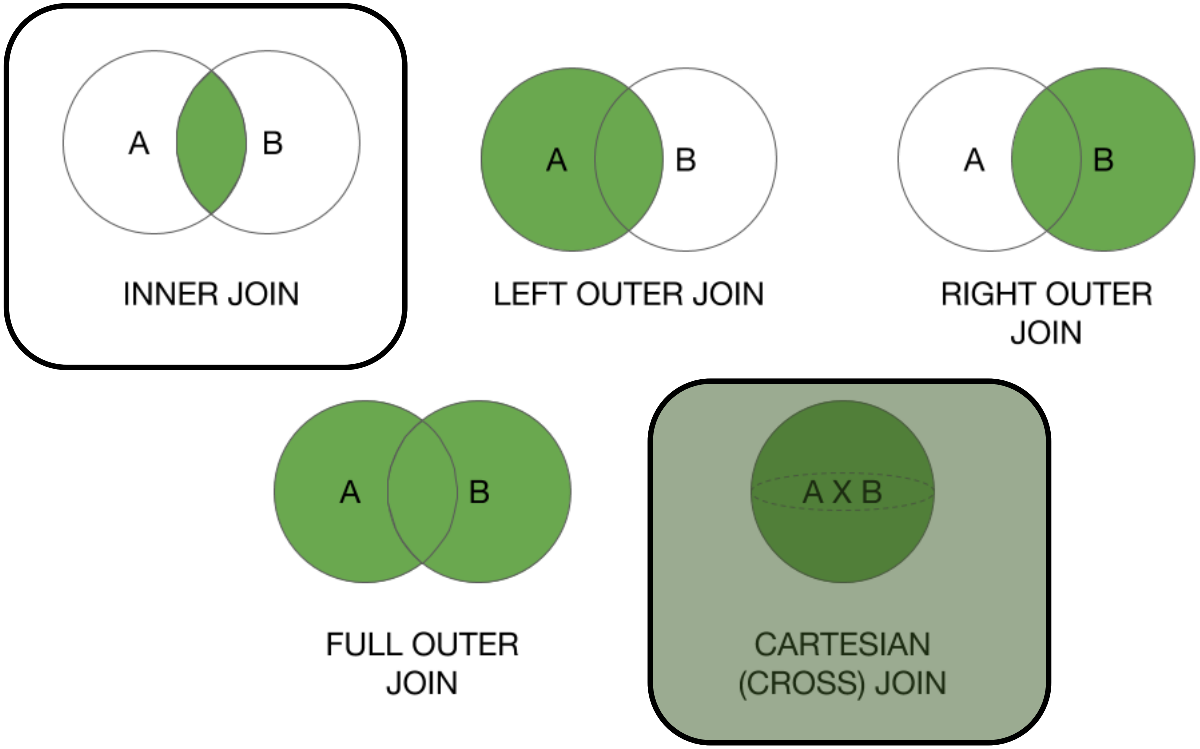
Types of Joins

Cartesian Joins

Inner Joins

- Equality Joins
- Non-equality Joins
- Self Joins

Outer Joins



Equality Joins

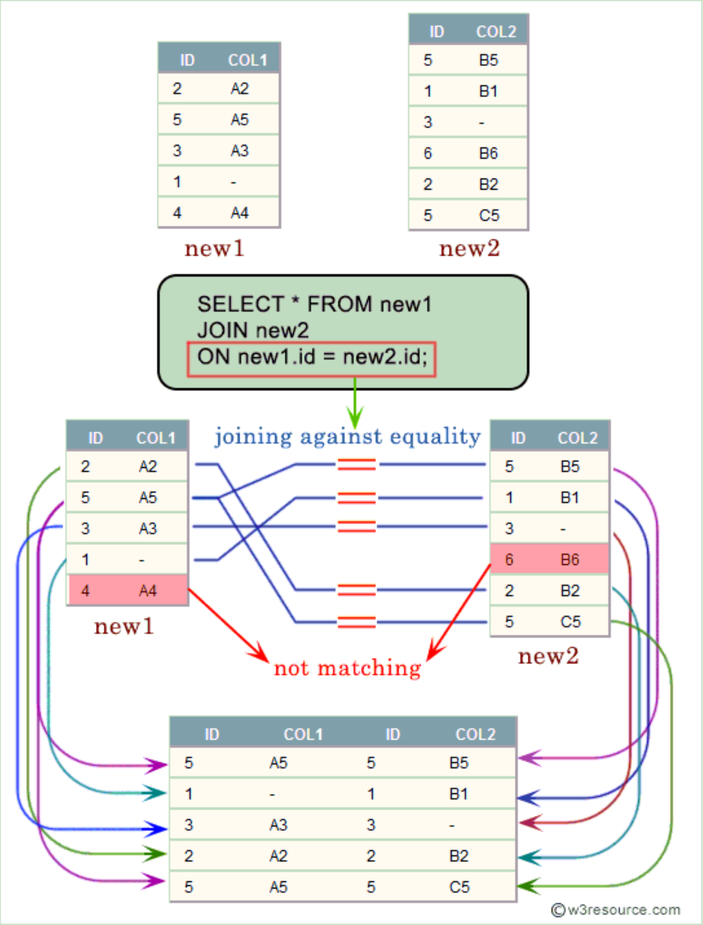
Link rows through equivalent data that exists in both tables

Matching tables on joining cond

- Based on a matching key (PK-FK)

Output: Intersection of two tables

- Only include those matched
- Non-matched rows excluded



ID	COL1
2	A2
5	A5
3	A3
1	-
4	A4

new1

ID	COL2
5	B5
1	B1
3	-
6	B6
2	B2
5	C5

new2

```
SELECT * FROM new1
JOIN new2
ON new1.id = new2.id;
```

ID	COL1
2	A2
5	A5
3	A3
1	-
4	A4

new1

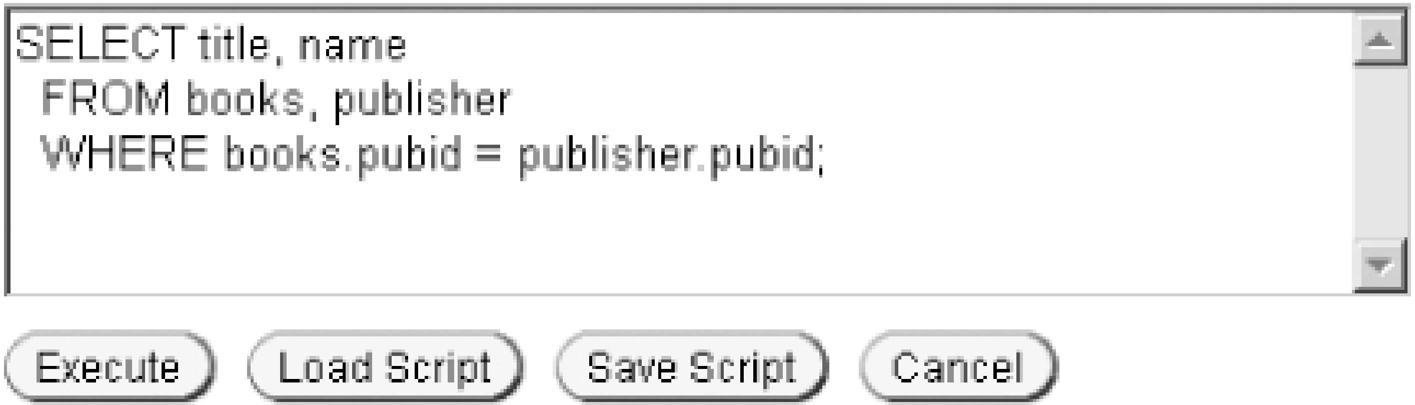
ID	COL2
5	B5
1	B1
3	-
6	B6
2	B2
5	C5

new2

ID	COL1	ID	COL2
5	A5	5	B5
1	-	1	B1
3	A3	3	-
2	A2	2	B2
5	A5	5	C5

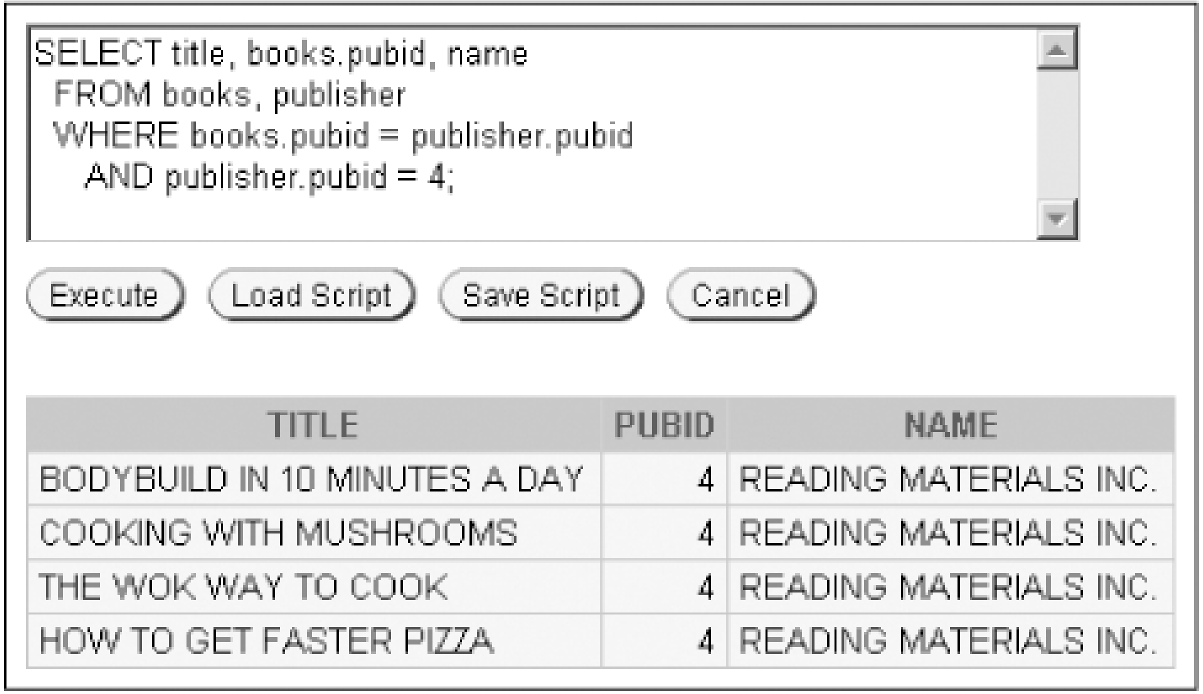
©w3resource.com

Equality Joins: WHERE Clause Example



```
SELECT title, name
  FROM books, publisher
  WHERE books.pubid = publisher.pubid;
```

WHERE Clause Supports Join and Other Conditions

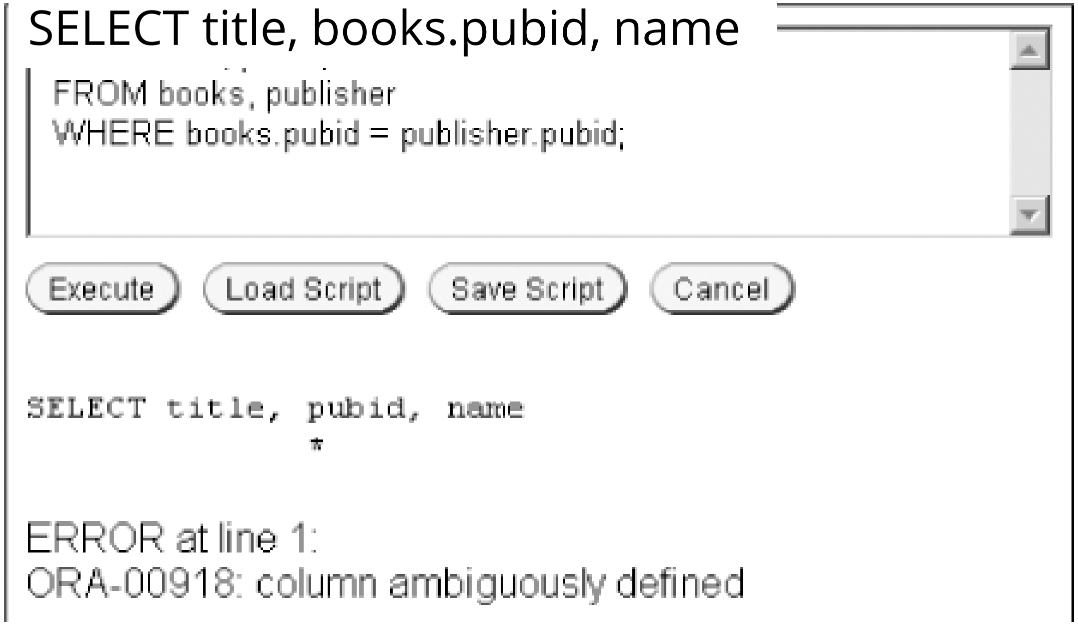


```
SELECT title, books.pubid, name
  FROM books, publisher
  WHERE books.pubid = publisher.pubid
     AND publisher.pubid = 4;
```

TITLE	PUBID	NAME
BODYBUILD IN 10 MINUTES A DAY	4	READING MATERIALS INC.
COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
THE WOK WAY TO COOK	4	READING MATERIALS INC.
HOW TO GET FASTER PIZZA	4	READING MATERIALS INC.

Column Ambiguously Defined?

- Columns in both tables must be qualified



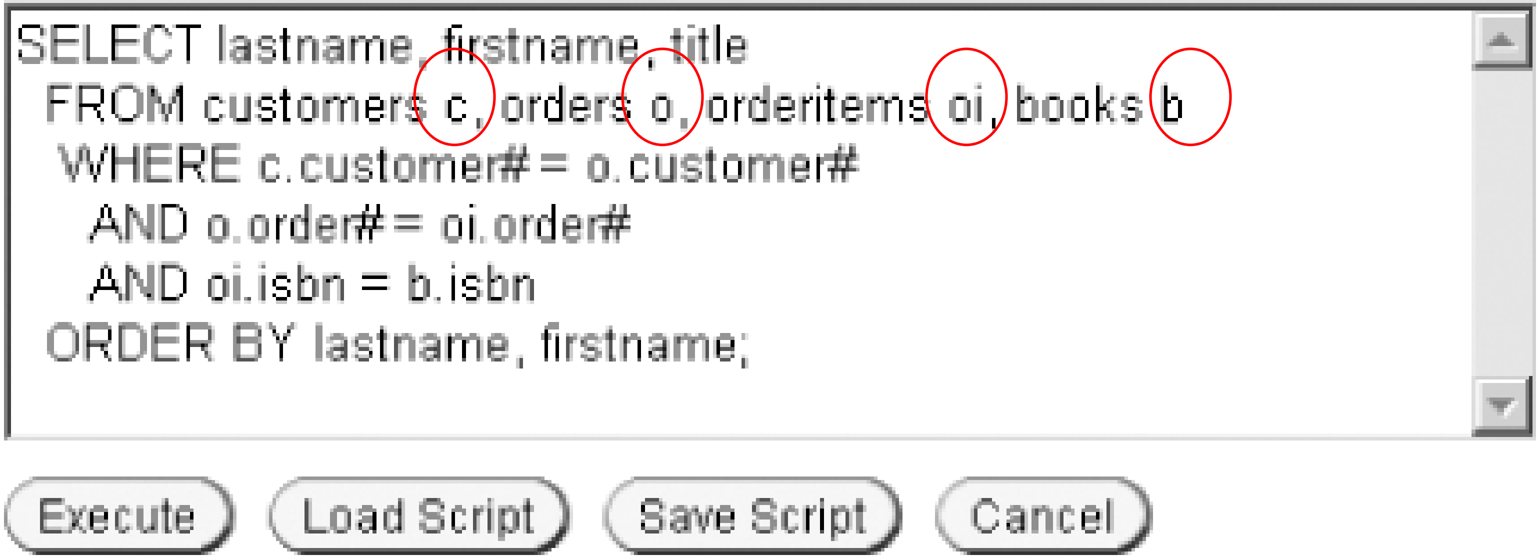
```
SELECT title, books.pubid, name
FROM books, publisher
WHERE books.pubid = publisher.pubid;
```

```
SELECT title, pubid, name
              *

ERROR at line 1:
ORA-00918: column ambiguously defined
```

Joining More Than Two Tables

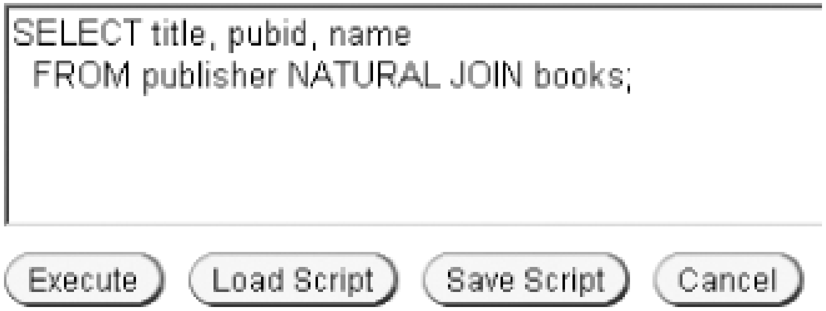
- Alias
- Joining 4 tables requires 3 join conditions



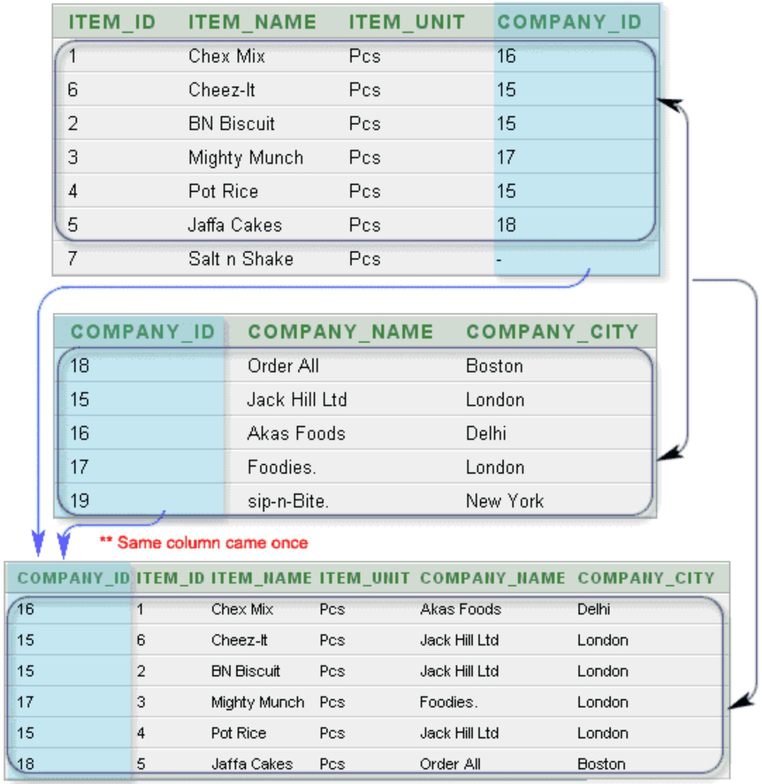
```
SELECT lastname, firstname, title
 FROM customers c, orders o, orderitems oi, books b
  WHERE c.customer# = o.customer#
    AND o.order# = oi.order#
    AND oi.isbn = b.isbn
 ORDER BY lastname, firstname;
```

WAYNE STATE
Mike Ilitch School of Business

Equality Joins: NATURAL JOIN



```
SELECT title, pubid, name
  FROM publisher NATURAL JOIN books;
```



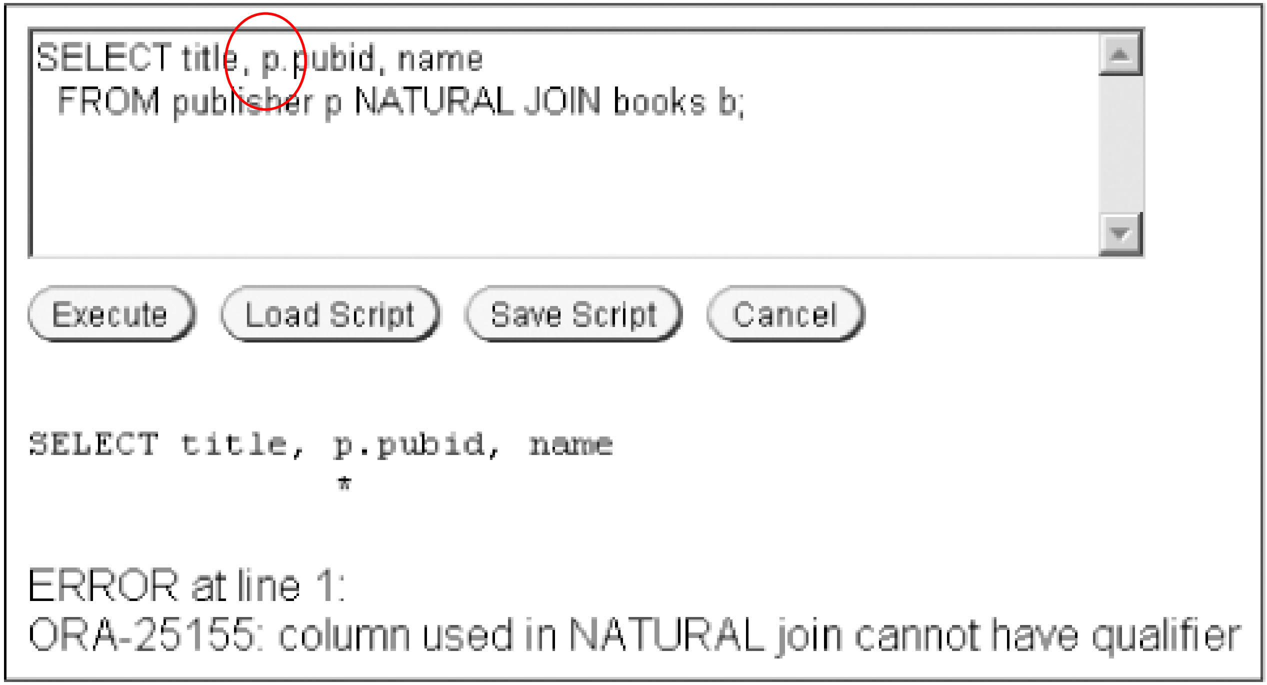
ITEM_ID	ITEM_NAME	ITEM_UNIT	COMPANY_ID
1	Chex Mix	Pcs	16
6	Cheez-It	Pcs	15
2	BN Biscuit	Pcs	15
3	Mighty Munch	Pcs	17
4	Pot Rice	Pcs	15
5	Jaffa Cakes	Pcs	18
7	Salt n Shake	Pcs	-

COMPANY_ID	COMPANY_NAME	COMPANY_CITY
18	Order All	Boston
15	Jack Hill Ltd	London
16	Akas Foods	Delhi
17	Foodies.	London
19	sip-n-Bite.	New York

** Same column came once

COMPANY_ID	ITEM_ID	ITEM_NAME	ITEM_UNIT	COMPANY_NAME	COMPANY_CITY
16	1	Chex Mix	Pcs	Akas Foods	Delhi
15	6	Cheez-It	Pcs	Jack Hill Ltd	London
15	2	BN Biscuit	Pcs	Jack Hill Ltd	London
17	3	Mighty Munch	Pcs	Foodies.	London
15	4	Pot Rice	Pcs	Jack Hill Ltd	London
18	5	Jaffa Cakes	Pcs	Order All	Boston

No Qualifiers Needed with NATURAL JOIN



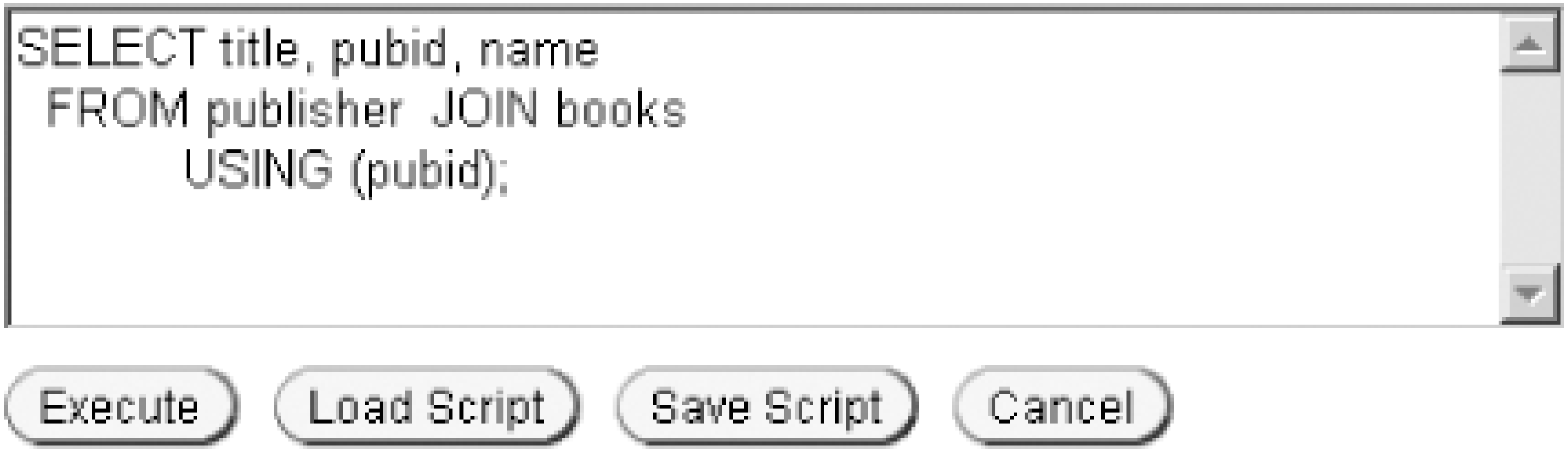
```
SELECT title, p.pubid, name
  FROM publisher p NATURAL JOIN books b;
```

```
SELECT title, p.pubid, name
              *
```

ERROR at line 1:
ORA-25155: column used in NATURAL join cannot have qualifier

FIGURE 9-14 Column qualifier error with a NATURAL JOIN

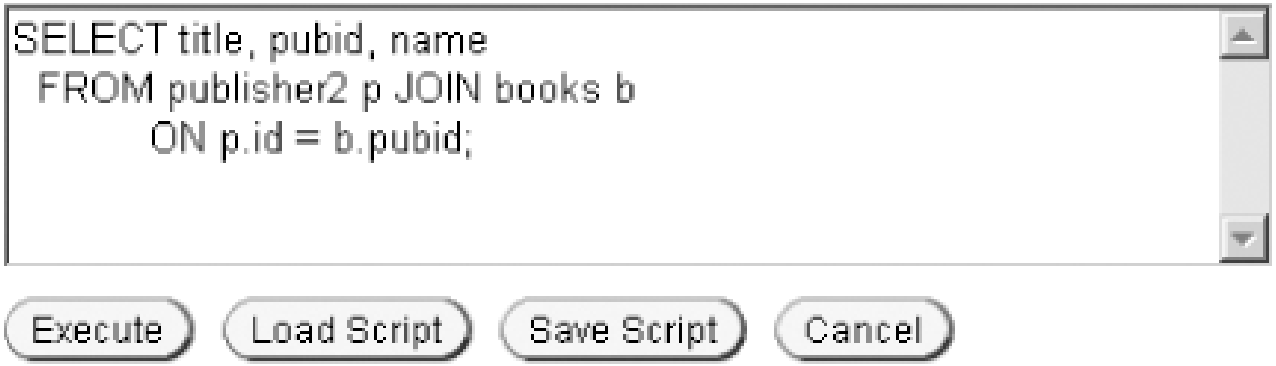
Equality Joins: JOIN…USING



```
SELECT title, pubid, name
  FROM publisher  JOIN books
           USING (pubid);
```

Equality Joins: JOIN...ON

- Required if column names are different



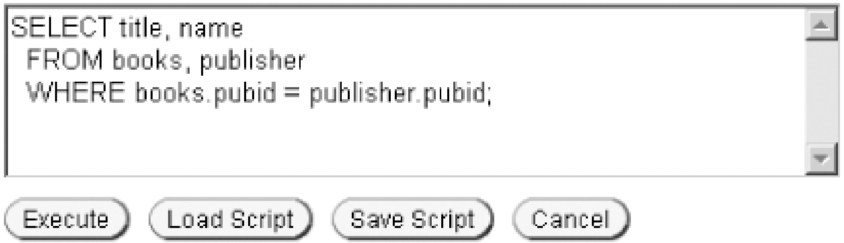
```
SELECT title, pubid, name
  FROM publisher2 p JOIN books b
         ON p.id = b.pubid;
```

JOIN Keyword Overview

- Use JOIN…USING when tables have one or more columns in common
- Use JOIN…ON when same named columns are not involved or a condition is needed to specify a relationship other than equivalency (next section)
- Using the JOIN keyword frees the WHERE clause for exclusive use in restricting rows

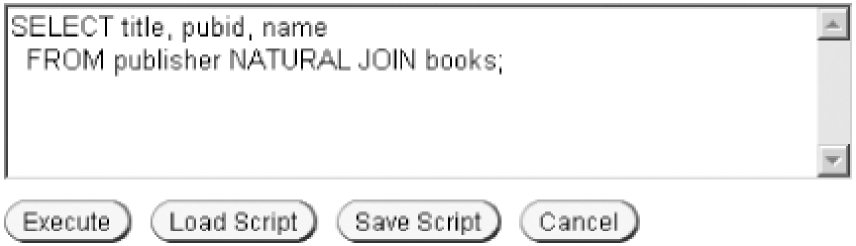
Equality Join Overview

Traditional Method – WHERE clause



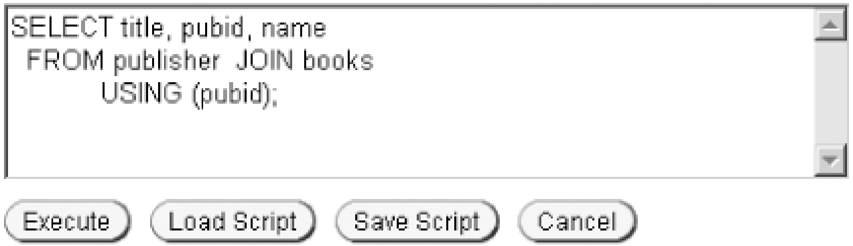
```
SELECT title, name
 FROM books, publisher
 WHERE books.pubid = publisher.pubid;
```

Join Method 1 – NATURAL JOIN



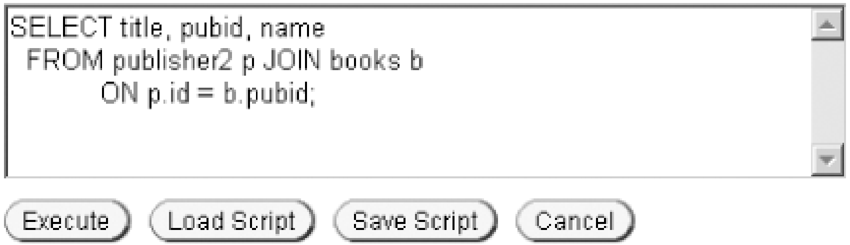
```
SELECT title, pubid, name
 FROM publisher NATURAL JOIN books;
```

Join Method 2 – JOIN…USING



```
SELECT title, pubid, name
 FROM publisher  JOIN books
          USING (pubid);
```

Join Method 3 – JOIN…ON



```
SELECT title, pubid, name
 FROM publisher2 p JOIN books b
          ON p.id = b.pubid;
```

Wrap-up: Table Join Checklist

1. How many tables to be combined
2. How many join conditions required
3. Find pairs of tables based on their common keys

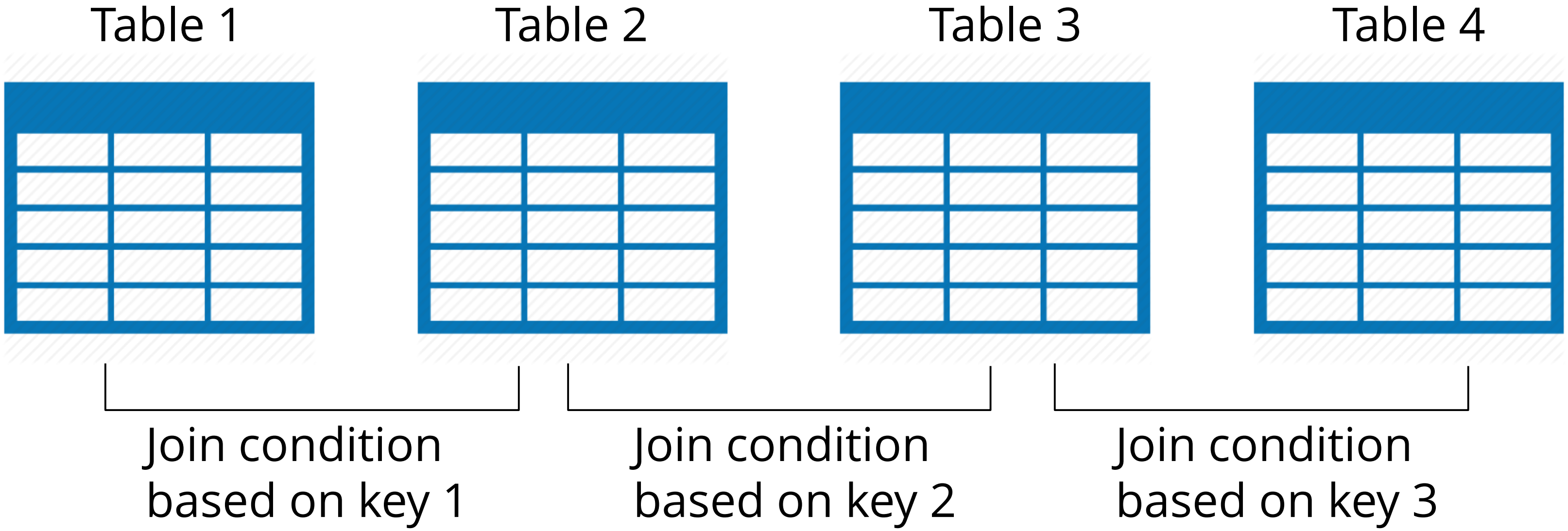


Table 1

Table 2

Table 3

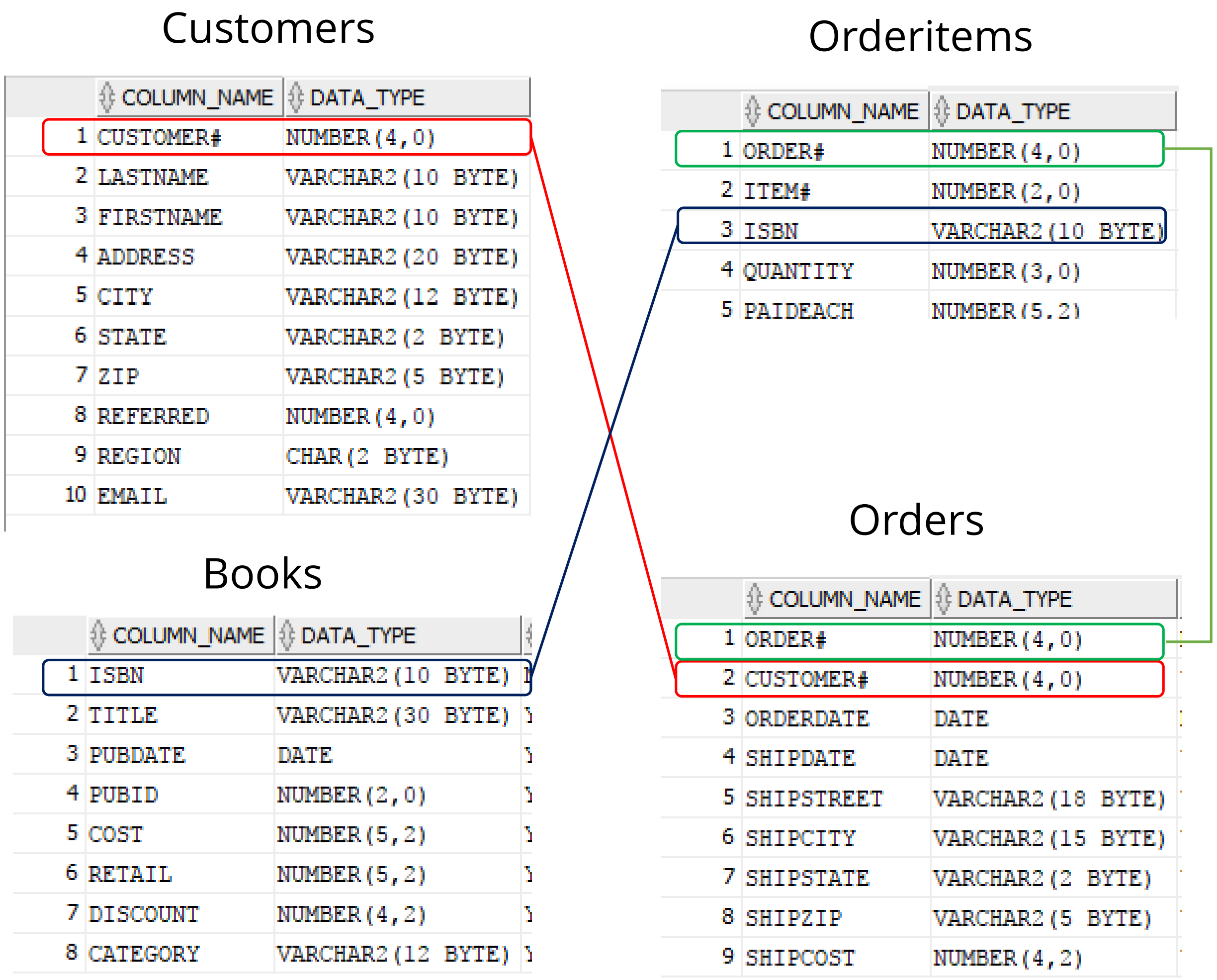
Table 4

Join condition based on key 1

Join condition based on key 2

Join condition based on key 3

Wrap-up: Table Join Checklist (cont’d)



Customers

	COLUMN_NAME	DATA_TYPE
1	CUSTOMER#	NUMBER(4,0)
2	LASTNAME	VARCHAR2(10 BYTE)
3	FIRSTNAME	VARCHAR2(10 BYTE)
4	ADDRESS	VARCHAR2(20 BYTE)
5	CITY	VARCHAR2(12 BYTE)
6	STATE	VARCHAR2(2 BYTE)
7	ZIP	VARCHAR2(5 BYTE)
8	REFERRED	NUMBER(4,0)
9	REGION	CHAR(2 BYTE)
10	EMAIL	VARCHAR2(30 BYTE)

Orderitems

	COLUMN_NAME	DATA_TYPE
1	ORDER#	NUMBER(4,0)
2	ITEM#	NUMBER(2,0)
3	ISBN	VARCHAR2(10 BYTE)
4	QUANTITY	NUMBER(3,0)
5	PAIDEACH	NUMBER(5,2)

Books

	COLUMN_NAME	DATA_TYPE
1	ISBN	VARCHAR2(10 BYTE)
2	TITLE	VARCHAR2(30 BYTE)
3	PUBDATE	DATE
4	PUBID	NUMBER(2,0)
5	COST	NUMBER(5,2)
6	RETAIL	NUMBER(5,2)
7	DISCOUNT	NUMBER(4,2)
8	CATEGORY	VARCHAR2(12 BYTE)

Orders

	COLUMN_NAME	DATA_TYPE
1	ORDER#	NUMBER(4,0)
2	CUSTOMER#	NUMBER(4,0)
3	ORDERDATE	DATE
4	SHIPDATE	DATE
5	SHIPSTREET	VARCHAR2(18 BYTE)
6	SHIPCITY	VARCHAR2(15 BYTE)
7	SHIPSTATE	VARCHAR2(2 BYTE)
8	SHIPZIP	VARCHAR2(5 BYTE)
9	SHIPCOST	NUMBER(4,2)

WAYNE STATE
Mike Ilitch School of Business

Wrap-up: 4. Writing Join Conditions

- WHERE
  - `WHERE customers.customer# = orders.customer# AND orders.order# = orderitems.order# AND …`
- NATURAL JOIN
  - `FROM customers NATURAL JOIN orders NATURAL JOIN orderitems …`
- JOIN USING
  - `FROM customers JOIN orders USING (customer#) JOIN orderitems USING (order#) …`
- JOIN ON
  - `FROM customers JOIN orders ON customers.customer#=orders.customer# JOIN orderitems ON orders.order#=orderitems.order# …`