ITC 5104 DATABASE DESIGN AND SQL

Lecture 6

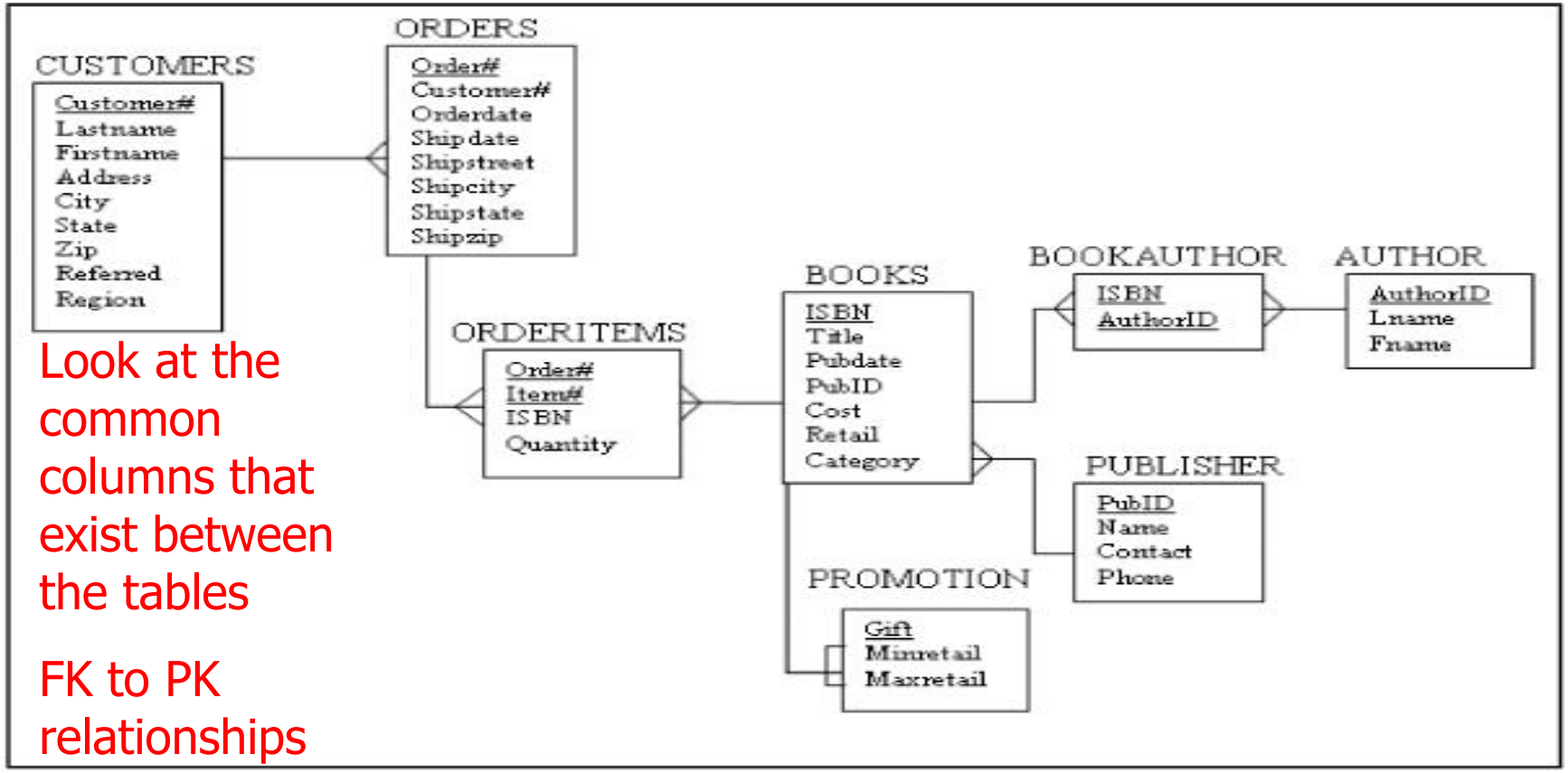
Chapter 9 Oracle 12c: SQL

Joining Data From Multiple Tables

Objectives

- Create a Cartesian product
- Create an equality join using the WHERE clause
- Create an equality join using the JOIN keyword
- Create a non-equality join using the WHERE clause
- Create a non-equality join using the JOIN … ON approach
- Create a self-join in the WHERE clause and with the JOIN keyword
- Distinguish between an inner join and an outer join
- Create an outer join using the WHERE clause
- Create an outer join using the OUTER keyword
- Join three or more tables

Equality Joins



Look at the common columns that exist between the tables

FK to PK relationships

Note: Underlines denote primary key columns

FIGURE 9-6 JustLee Book's table structure

Focus

- During the design phase, redundancy was reduced by structuring the data into multiple tables
- This chapter focuses on creating access paths to combine or join data that are stored in more than one table
- Traditionally in Oracle joins have been done in the WHERE clause, with Oracle 9i they adopted the ANSI-compliant joins
- In the ANSI SQL standard tables are joined in the FROM clause
- It also specifies the type of JOIN being performed
- Enhancements have been added in SQL 2003
- Does this mean that all SQL is the same regardless of the platform?
- This chapter focuses on adding join conditions, which are instructions in queries that combine data from more than one table

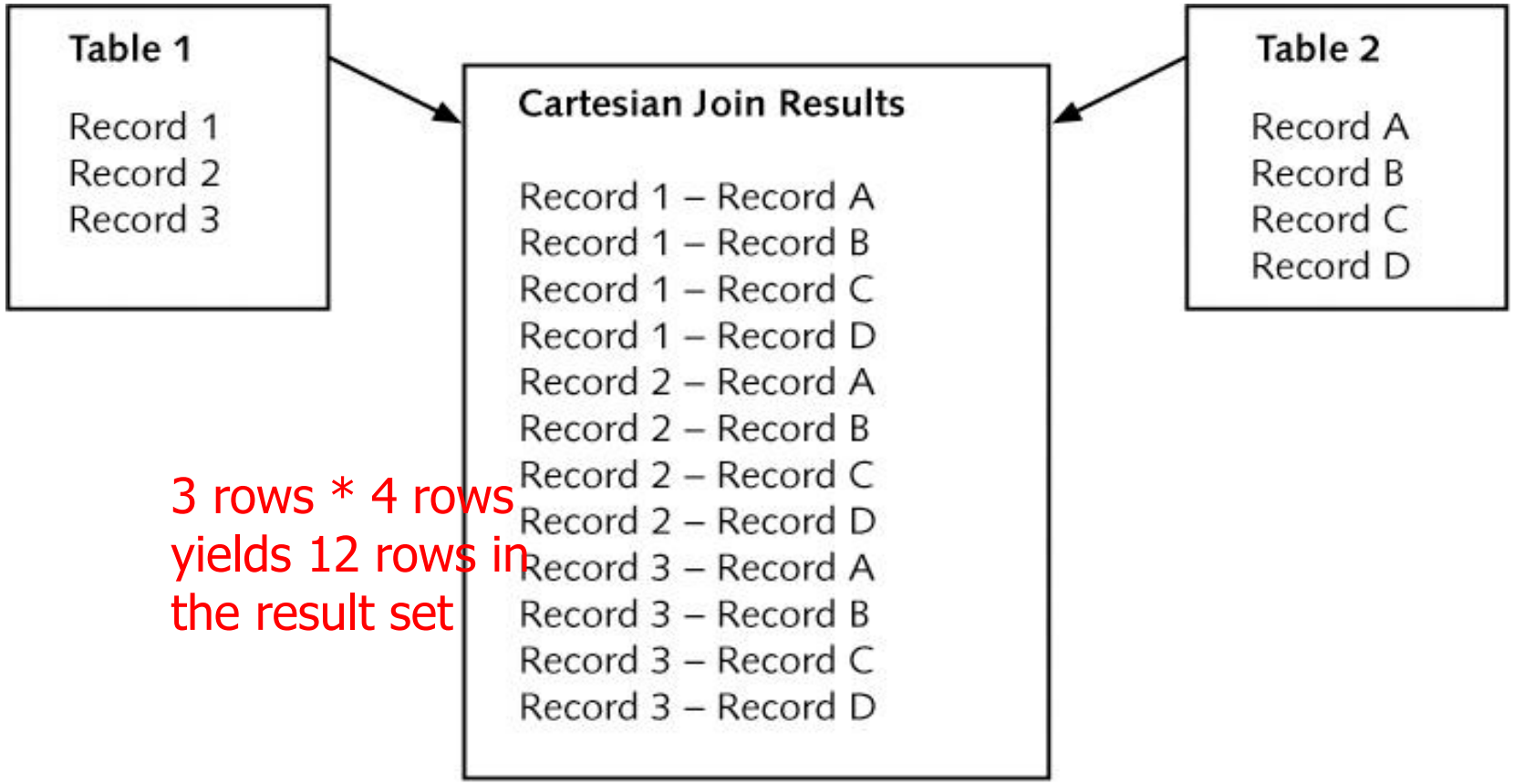
Focus

- The WHERE clause can then be used specifically for restricting rows being returned from the tables
- We will look at the traditional method using the WHERE clause approach then will also look at the SQL-99 approach using the JOIN keyword in the FROM clause
- You will need to understand both approaches to creating joins to support legacy Oracle systems

Cartesian Joins

- In a Cartesian join (also called a Cartesian product or a cross join), each record in the first table is matched with a record from the second table
- So, if you have m rows in the first table and n rows in the second table the result table will yield m*n rows
- Good for statistical analysis or to generate large amounts of data for testing

Cartesian Join



3 rows * 4 rows yields 12 rows in the result set

FIGURE 9-2 Results of a Cartesian join

Cartesian Joins – Traditional Method

- Consider the following: you have been asked to retrieve the publisher's name for each book in inventory
- First of all, let's look at the two tables in question

Cartesian Joins – Traditional Method

0.078 seconds

```
SELECT *
FROM books;

SELECT *
FROM publisher;
```

Query Result | Script Output

Task completed in 0.078 seconds

ISBN	TITLE	PUBDATE	PUBID	COST
1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75
0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2
4981341710	BUILDING A CAR WITH TOOTHPICKS	18-MAR-06	2	37.8
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4
3437212490	COOKING WITH MUSHROOMS	28-FEB-04	4	12.5
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8
9959789321	E-BUSINESS THE EASY WAY	01-MAR-06	2	37.9
2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48
0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19
8117949391	BIG BEAR AND LITTLE DOVE	08-NOV-05	5	5.32
0132149871	HOW TO GET FASTER PIZZA	11-NOV-06	4	17.85
9247381001	HOW TO MANAGE THE MANAGER	09-MAY-03	1	15.4
2147428890	SHORTEST POEMS	01-MAY-05	5	21.85

14 rows selected

PUBID	NAME	CONTACT	PHONE
1	PRINTING IS US	TOMMIE SEYMOUR	000-714-8321
2	PUBLISH OUR WAY	JANE TOMLIN	010-410-0010
3	AMERICAN PUBLISHING	DAVID DAVIDSON	800-555-1211
4	READING MATERIALS INC.	RENEE SMITH	800-555-9743
5	REED-N-RITE	SEBASTIAN JONES	800-555-8284

14 rows in books table and 5 rows in the publisher table

Cartesian Joins – Traditional Method

```
SELECT title, name
FROM books, publisher;
```

Script Output | Query Result

Fetched 50 rows in 0.016 seconds

	TITLE	NAME
1	BODYBUILD IN 10 MINUTES A DAY	PRINTING IS US
2	REVENGE OF MICKEY	PRINTING IS US
3	BUILDING A CAR WITH TOOTHPICKS	PRINTING IS US
4	DATABASE IMPLEMENTATION	PRINTING IS US
5	COOKING WITH MUSHROOMS	PRINTING IS US
6	HOLY GRAIL OF ORACLE	PRINTING IS US
7	HANDCRANKED COMPUTERS	PRINTING IS US
8	E-BUSINESS THE EASY WAY	PRINTING IS US
9	PAINLESS CHILD-REARING	PRINTING IS US
10	THE WOK WAY TO COOK	PRINTING IS US
11	BIG BEAR AND LITTLE DOVE	PRINTING IS US
12	HOW TO GET FASTER PIZZA	PRINTING IS US
13	HOW TO MANAGE THE MANAGER	PRINTING IS US
14	SHORTEST POEMS	PRINTING IS US
15	BODYBUILD IN 10 MINUTES A DAY	PUBLISH OUR WAY
16	REVENGE OF MICKEY	PUBLISH OUR WAY
17	BUILDING A CAR WITH TOOTHPICKS	PUBLISH OUR WAY
18	DATABASE IMPLEMENTATION	PUBLISH OUR WAY
19	COOKING WITH MUSHROOMS	PUBLISH OUR WAY
20	HOLY GRAIL OF ORACLE	PUBLISH OUR WAY
21	HANDCRANKED COMPUTERS	PUBLISH OUR WAY
22	E-BUSINESS THE EASY WAY	PUBLISH OUR WAY
23	PAINLESS CHILD-REARING	PUBLISH OUR WAY
24	THE WOK WAY TO COOK	PUBLISH OUR WAY
25	BIG BEAR AND LITTLE DOVE	PUBLISH OUR WAY
26	HOW TO GET FASTER PIZZA	PUBLISH OUR WAY
27	HOW TO MANAGE THE MANAGER	PUBLISH OUR WAY
28	SHORTEST POEMS	PUBLISH OUR WAY
29	BODYBUILD IN 10 MINUTES A DAY	AMERICAN PUBLISHING

```
SELECT title, name
FROM books, publisher;
```

Script Output | Query Result

All Rows Fetched: 70 in 0.015 seconds

	TITLE	NAME
42	SHORTEST POEMS	AMERICAN PUBLISHING
43	BODYBUILD IN 10 MINUTES A DAY	READING MATERIALS INC.
44	REVENGE OF MICKEY	READING MATERIALS INC.
45	BUILDING A CAR WITH TOOTHPICKS	READING MATERIALS INC.
46	DATABASE IMPLEMENTATION	READING MATERIALS INC.
47	COOKING WITH MUSHROOMS	READING MATERIALS INC.
48	HOLY GRAIL OF ORACLE	READING MATERIALS INC.
49	HANDCRANKED COMPUTERS	READING MATERIALS INC.
50	E-BUSINESS THE EASY WAY	READING MATERIALS INC.
51	PAINLESS CHILD-REARING	READING MATERIALS INC.
52	THE WOK WAY TO COOK	READING MATERIALS INC.
53	BIG BEAR AND LITTLE DOVE	READING MATERIALS INC.
54	HOW TO GET FASTER PIZZA	READING MATERIALS INC.
55	HOW TO MANAGE THE MANAGER	READING MATERIALS INC.
56	SHORTEST POEMS	READING MATERIALS INC.
57	BODYBUILD IN 10 MINUTES A DAY	REED-N-RITE
58	REVENGE OF MICKEY	REED-N-RITE
59	BUILDING A CAR WITH TOOTHPICKS	REED-N-RITE
60	DATABASE IMPLEMENTATION	REED-N-RITE
61	COOKING WITH MUSHROOMS	REED-N-RITE
62	HOLY GRAIL OF ORACLE	REED-N-RITE
63	HANDCRANKED COMPUTERS	REED-N-RITE
64	E-BUSINESS THE EASY WAY	REED-N-RITE
65	PAINLESS CHILD-REARING	REED-N-RITE
66	THE WOK WAY TO COOK	REED-N-RITE
67	BIG BEAR AND LITTLE DOVE	REED-N-RITE
68	HOW TO GET FASTER PIZZA	REED-N-RITE
69	HOW TO MANAGE THE MANAGER	REED-N-RITE
70	SHORTEST POEMS	REED-N-RITE

The result set returned 70 rows (14 * 5 = 70). Look at the output. Do you notice anything peculiar?

Cartesian Joins – Traditional Method

- Although there were only 14 records in the books table of the database, 70 rows of output were seen in the result set
- If you see results like this, you should be suspicious
- The problem with the output is you have specified the columns to be retrieved from the two tables, but you have not specified the common field between the two tables that will join the two tables together
- Every row of one table is joined to every row of the second table
- The database does not know how the tables should be joined, so it joins every row with every possible combination of records

Cartesian Joins – JOIN Method

```
1 SELECT title, name
2 FROM books CROSS JOIN publisher;
```

Script Output | Query Result

Fetched 50 rows in 0.062 seconds

	TITLE	NAME
1	BODYBUILD IN 10 MINUTES A DAY	PRINTING IS US
2	REVENGE OF MICKEY	PRINTING IS US
3	BUILDING A CAR WITH TOOTHPICKS	PRINTING IS US
4	DATABASE IMPLEMENTATION	PRINTING IS US
5	COOKING WITH MUSHROOMS	PRINTING IS US
6	HOLY GRAIL OF ORACLE	PRINTING IS US
7	HANDCRANKED COMPUTERS	PRINTING IS US
8	E-BUSINESS THE EASY WAY	PRINTING IS US
9	PAINLESS CHILD-REARING	PRINTING IS US
10	THE WOK WAY TO COOK	PRINTING IS US
11	BIG BEAR AND LITTLE DOVE	PRINTING IS US
12	HOW TO GET FASTER PIZZA	PRINTING IS US
13	HOW TO MANAGE THE MANAGER	PRINTING IS US
14	SHORTEST POEMS	PRINTING IS US
15	BODYBUILD IN 10 MINUTES A DAY	PUBLISH OUR WAY
16	REVENGE OF MICKEY	PUBLISH OUR WAY
17	BUILDING A CAR WITH TOOTHPICKS	PUBLISH OUR WAY
18	DATABASE IMPLEMENTATION	PUBLISH OUR WAY
19	COOKING WITH MUSHROOMS	PUBLISH OUR WAY
20	HOLY GRAIL OF ORACLE	PUBLISH OUR WAY
21	HANDCRANKED COMPUTERS	PUBLISH OUR WAY
22	E-BUSINESS THE EASY WAY	PUBLISH OUR WAY
23	PAINLESS CHILD-REARING	PUBLISH OUR WAY
24	THE WOK WAY TO COOK	PUBLISH OUR WAY
25	BIG BEAR AND LITTLE DOVE	PUBLISH OUR WAY
26	HOW TO GET FASTER PIZZA	PUBLISH OUR WAY
27	HOW TO MANAGE THE MANAGER	PUBLISH OUR WAY
28	SHORTEST POEMS	PUBLISH OUR WAY
29	BODYBUILD IN 10 MINUTES A DAY	AMERICAN PUBLISHING

```
1 SELECT title, name
2 FROM books CROSS JOIN publisher;
```

Script Output | Query Result

All Rows Fetched: 70 in 0.016 seconds

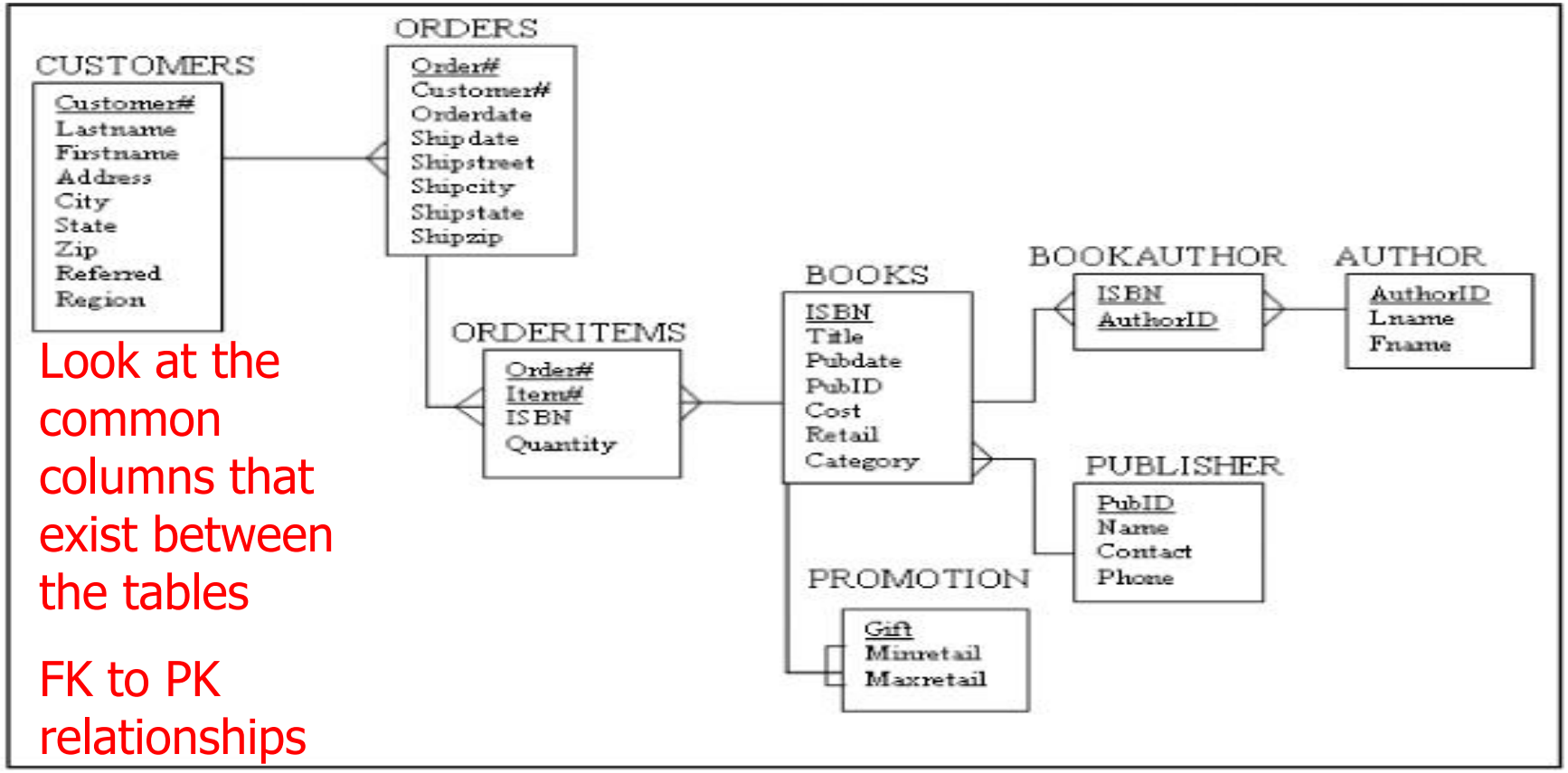
	TITLE	NAME
42	SHORTEST POEMS	AMERICAN PUBLISHING
43	BODYBUILD IN 10 MINUTES A DAY	READING MATERIALS INC.
44	REVENGE OF MICKEY	READING MATERIALS INC.
45	BUILDING A CAR WITH TOOTHPICKS	READING MATERIALS INC.
46	DATABASE IMPLEMENTATION	READING MATERIALS INC.
47	COOKING WITH MUSHROOMS	READING MATERIALS INC.
48	HOLY GRAIL OF ORACLE	READING MATERIALS INC.
49	HANDCRANKED COMPUTERS	READING MATERIALS INC.
50	E-BUSINESS THE EASY WAY	READING MATERIALS INC.
51	PAINLESS CHILD-REARING	READING MATERIALS INC.
52	THE WOK WAY TO COOK	READING MATERIALS INC.
53	BIG BEAR AND LITTLE DOVE	READING MATERIALS INC.
54	HOW TO GET FASTER PIZZA	READING MATERIALS INC.
55	HOW TO MANAGE THE MANAGER	READING MATERIALS INC.
56	SHORTEST POEMS	READING MATERIALS INC.
57	BODYBUILD IN 10 MINUTES A DAY	REED-N-RITE
58	REVENGE OF MICKEY	REED-N-RITE
59	BUILDING A CAR WITH TOOTHPICKS	REED-N-RITE
60	DATABASE IMPLEMENTATION	REED-N-RITE
61	COOKING WITH MUSHROOMS	REED-N-RITE
62	HOLY GRAIL OF ORACLE	REED-N-RITE
63	HANDCRANKED COMPUTERS	REED-N-RITE
64	E-BUSINESS THE EASY WAY	REED-N-RITE
65	PAINLESS CHILD-REARING	REED-N-RITE
66	THE WOK WAY TO COOK	REED-N-RITE
67	BIG BEAR AND LITTLE DOVE	REED-N-RITE
68	HOW TO GET FASTER PIZZA	REED-N-RITE
69	HOW TO MANAGE THE MANAGER	REED-N-RITE
70	SHORTEST POEMS	REED-N-RITE

The result set also returns 70 rows (14 * 5 = 70). This is the ANSI JOIN method using the CROSS JOIN keywords

Cartesian Joins – JOIN Method

- Notice the syntax on the previous slide. The two tables are joined in the FROM clause by the keywords CROSS JOIN
- There are no commas separating any parts of the FROM clause as there are in the traditional method
- If you were to place a comma in the FROM clause, you would receive an error

Equality Joins



Look at the common columns that exist between the tables

FK to PK relationships

Note: Underlines denote primary key columns

FIGURE 9-6 JustLee Book's table structure

Equality Joins

- Previously, the query returned a Cartesian join since the software did not know which column the two tables had in common
- The most common type of join used in the workplace uses two or more tables with equivalent data stored in a common column
- Such joins are called equality joins. These can also be referred to as equijoins, inner joins or simple joins
- A common column is a column with equivalent data that exists in two or more tables

Equality Joins

- BOOKS and PUBLISHER have a *common column* called pubid (a primary key to foreign key relationship exists between the two tables)
- When you want a list of publishers for each book in the BOOKS table, you want to match the publisher ID in the BOOKS table with the publisher ID in the PUBLISHER table
- The result set will be a list where there is a match between the two pubid columns stored in each table

Equality Joins – Traditional Method

- The traditional method that avoids the Cartesian product uses the WHERE clause
- The WHERE clause is used to define the access path Oracle 11g needs in order to join the tables correctly

Equality Joins – Traditional Method

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

Script Output | Query Result

All Rows Fetched: 14 in 0.016 seconds

	TITLE	NAME
1	REVENGE OF MICKEY	PRINTING IS US
2	HOW TO MANAGE THE MANAGER	PRINTING IS US
3	E-BUSINESS THE EASY WAY	PUBLISH OUR WAY
4	BUILDING A CAR WITH TOOTHPICKS	PUBLISH OUR WAY
5	HOLY GRAIL OF ORACLE	AMERICAN PUBLISHING
6	DATABASE IMPLEMENTATION	AMERICAN PUBLISHING
7	HANDCRANKED COMPUTERS	AMERICAN PUBLISHING
8	COOKING WITH MUSHROOMS	READING MATERIALS INC.
9	THE WOK WAY TO COOK	READING MATERIALS INC.
10	HOW TO GET FASTER PIZZA	READING MATERIALS INC.
11	BODYBUILD IN 10 MINUTES A DAY	READING MATERIALS INC.
12	PAINLESS CHILD-REARING	REED-N-RITE
13	BIG BEAR AND LITTLE DOVE	REED-N-RITE
14	SHORTEST POEMS	REED-N-RITE

Equality Joins – Traditional Method

- The WHERE clause lets Oracle 11g know the BOOKS and the PUBLISHER table are related by the pubid column
- The equal sign is used to specify that the contents of the Pubid column of each table must be exactly equal for the rows to be joined and returned in the results
- Notice that the pubid column names are prefixed with the name of each table. If the table names were omitted, you will get an error message
- This is known as a column qualifier. For example, publisher.pubid is the pubid of the PUBLISHER table and books.pubid is the pubid of the BOOKS table

Equality Joins – Traditional Method

```
1 SELECT title, name
2 FROM books, publisher
3 WHERE pubid = pubid;
```

Script Output | Query Result

Executing:SELECT title, nameFROM books, publisherWHERE pubid = pubid in 0.032 seconds

```
ORA-00918: column ambiguously defined
00918. 00000 - "column ambiguously defined"
*Cause:
*Action:
Error at Line: 3 Column: 14
```

The system does not know which pubid column is which, since both tables have the same column defined

Equality Joins – Traditional Method

- If you wanted the pubid column to appear in the result set, it would also have to be qualified, since there are two of them defined in the SELECT statement
- It does not matter which pubid is used since both of the values for the pubid are equal either can be used, on slide 24 the books.pubid value is used, why? Very simple books is a shorter word than publisher.pubid
- Could as easily used publisher.pubid the query would return the same results

Equality Joins – Traditional Method

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

Script Output | Query Result

Executing:SELECT title, pubid, nameFROM books, publisherWHERE books.pubid = publisher.pubid in 0.032 seconds

```
ORA-00918: column ambiguously defined
00918. 00000 - "column ambiguously defined"
*Cause:
*Action:
Error at Line: 1 Column: 14
```

Equality Joins – Traditional Method

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

Script Output | Query Result

All Rows Fetched: 14 in 0.015 seconds

	TITLE	PUBID	NAME
1	REVENGE OF MICKEY	1	PRINTING IS US
2	HOW TO MANAGE THE MANAGER	1	PRINTING IS US
3	E-BUSINESS THE EASY WAY	2	PUBLISH OUR WAY
4	BUILDING A CAR WITH TOOTHPICKS	2	PUBLISH OUR WAY
5	HOLY GRAIL OF ORACLE	3	AMERICAN PUBLISHING
6	DATABASE IMPLEMENTATION	3	AMERICAN PUBLISHING
7	HANDCRANKED COMPUTERS	3	AMERICAN PUBLISHING
8	COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
9	THE WOK WAY TO COOK	4	READING MATERIALS INC.
10	HOW TO GET FASTER PIZZA	4	READING MATERIALS INC.
11	BODYBUILD IN 10 MINUTES A DAY	4	READING MATERIALS INC.
12	PAINLESS CHILD-REARING	5	REED-N-RITE
13	BIG BEAR AND LITTLE DOVE	5	REED-N-RITE
14	SHORTEST POEMS	5	REED-N-RITE

Equality Joins – Traditional Method

- If you want to limit the results to a certain publisher, you can use the WHERE clause, as was done in the previous chapter
- These conditions are specified in the same WHERE clause used to join the two tables together
- The same logical operators discussed in the previous chapter can be used here as well
- See examples on the next two slides

Equality Joins – Traditional Method

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

Script Output | Query...

All Rows Fetched: 4 in 0.015 seconds

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

Equality Joins – Traditional Method

```
SELECT b.title, p.pubid, p.name
FROM books b, publisher p
WHERE b.pubid = p.pubid
AND (b.cost < 15 OR p.pubid = 1)
ORDER BY title;
```

Script Output | Query Result

All Rows Fetched: 4 in 0.031 seconds

	TITLE	PUBID	NAME
1	BIG BEAR AND LITTLE DOVE	5	REED-N-RITE
2	COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
3	HOW TO MANAGE THE MANAGER	1	PRINTING IS US
4	REVENGE OF MICKEY	1	PRINTING IS US

Equality Joins – Traditional Method

- The SELECT clause specifies the columns to be displayed. It also includes a table alias for publisher p
- A period separates the table alias from the column name
- The table alias in the FROM clause works like a column alias by temporarily giving a different name to a table
- A table alias *reduces the memory requirements* and *reduces the number of keystrokes* needed
- If a table alias is assigned in the FROM clause, it must be used whenever the table is referenced in that SQL statement

Equality Joins – Traditional Method

- The WHERE clause includes not only the access path to join both the BOOKS and PUBLISHER tables together, it also contains any other search conditions using the AND and OR logical operators
- The statement concludes with an ORDER BY clause to display the results in a sorted order

Equality Joins – JOIN Method

- Although there are three approaches you can use to create an equality join that uses the JOIN keyword we will focus on the one that will function with all SQL variants :
  - JOIN … ON
- We will focus in on the JOIN … ON method since it works in all cases the other two only work in Oracle and we will not address these
- We will also look a the traditional method used by Oracle since it works with other vendors

Equality Joins – JOIN … ON Method

- There may be instances where tables have common columns, but the columns names are not the same
- When related columns have different names, the software will not know how the columns are related and you will get an error message
- When there are no commonly named columns to use in a join, use the JOIN keyword in the FROM clause, then add the ON clause to specify which fields are related. ON goes immediately after the FROM clause

Equality Joins – JOIN … ON Method

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

Script Output | Query...

All Rows Fetched: 14 in 0.032 seconds

	TITLE	PUBID	NAME
1	REVENGE OF MICKEY	1	PRINTING IS US
2	HOW TO MANAGE THE MANAGER	1	PRINTING IS US
3	E-BUSINESS THE EASY WAY	2	PUBLISH OUR WAY
4	BUILDING A CAR WITH TOOTHPICKS	2	PUBLISH OUR WAY
5	HOLY GRAIL OF ORACLE	3	AMERICAN PUBLISHING
6	DATABASE IMPLEMENTATION	3	AMERICAN PUBLISHING
7	HANDCRANKED COMPUTERS	3	AMERICAN PUBLISHING
8	COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
9	THE WOK WAY TO COOK	4	READING MATERIALS INC.
10	HOW TO GET FASTER PIZZA	4	READING MATERIALS INC.
11	BODYBUILD IN 10 MINUTES A DAY	4	READING MATERIALS INC.
12	PAINLESS CHILD-REARING	5	REED-N-RITE
13	BIG BEAR AND LITTLE DOVE	5	REED-N-RITE
14	SHORTEST POEMS	5	REED-N-RITE

Since no columns that join tables in this database have different names, these columns were joined using this method just to demonstrate it

Equality Joins – INNER JOIN … ON Method

```
1  SELECT b.title, b.pubid, p.name
2  FROM books b INNER JOIN publisher p
3  ON b.pubid = p.pubid;
4
```

Script Output | Query...

All Rows Fetched: 14 in 0.031 seconds

	TITLE	PUBID	NAME
1	REVENGE OF MICKEY	1	PRINTING IS US
2	HOW TO MANAGE THE MANAGER	1	PRINTING IS US
3	E-BUSINESS THE EASY WAY	2	PUBLISH OUR WAY
4	BUILDING A CAR WITH TOOTHPICKS	2	PUBLISH OUR WAY
5	HOLY GRAIL OF ORACLE	3	AMERICAN PUBLISHING
6	DATABASE IMPLEMENTATION	3	AMERICAN PUBLISHING
7	HANDCRANKED COMPUTERS	3	AMERICAN PUBLISHING
8	COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
9	THE WOK WAY TO COOK	4	READING MATERIALS INC.
10	HOW TO GET FASTER PIZZA	4	READING MATERIALS INC.
11	BODYBUILD IN 10 MINUTES A DAY	4	READING MATERIALS INC.
12	PAINLESS CHILD-REARING	5	REED-N-RITE
13	BIG BEAR AND LITTLE DOVE	5	REED-N-RITE
14	SHORTEST POEMS	5	REED-N-RITE

The optional keyword INNER can be used with the JOIN keyword in the SELECT statement here as well

Equality Joins – JOIN Method

- There are three approaches you can use to create an equality join that uses the JOIN keyword:
  - NATURAL JOIN
  - JOIN … USING
  - JOIN … ON
- We will focus in on the JOIN … ON method since it works in all cases and is used by other SQL variants from other vendors
- We will also look a the traditional method used by Oracle
- Examples of the others will be shown

Using the NATURAL JOIN

- The NATURAL JOIN keywords instruct Oracle 11g to list the title of each book in the books table and the corresponding publisher ID and the publisher name
- Since the BOOKS and PUBLISHERS table contain the PUBID column it is a common column and it should be used to relate the two tables
- When using the NATURAL JOIN you are not required to specify columns the two tables have in common
- The NATURAL keyword implies that the two specified tables have at least one column in common with the same name and datatype
- Oracle 11g compares the two tables and uses the common columns to join the tables

Using the NATURAL JOIN

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

Script Output | Query Result

All Rows Fetched: 14 in 0.031 seconds

	TITLE	PUBID	NAME
1	REVENGE OF MICKEY	1	PRINTING IS US
2	HOW TO MANAGE THE MANAGER	1	PRINTING IS US
3	E-BUSINESS THE EASY WAY	2	PUBLISH OUR WAY
4	BUILDING A CAR WITH TOOTHPICKS	2	PUBLISH OUR WAY
5	HOLY GRAIL OF ORACLE	3	AMERICAN PUBLISHING
6	DATABASE IMPLEMENTATION	3	AMERICAN PUBLISHING
7	HANDCRANKED COMPUTERS	3	AMERICAN PUBLISHING
8	COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
9	THE WOK WAY TO COOK	4	READING MATERIALS INC.
10	HOW TO GET FASTER PIZZA	4	READING MATERIALS INC.
11	BODYBUILD IN 10 MINUTES A DAY	4	READING MATERIALS INC.
12	PAINLESS CHILD-REARING	5	REED-N-RITE
13	BIG BEAR AND LITTLE DOVE	5	REED-N-RITE
14	SHORTEST POEMS	5	REED-N-RITE

Using the NATURAL JOIN

```
1  SELECT title, p.pubid, name
2  FROM books NATURAL JOIN publisher;
3
```

Script Output | Query Result

Executing:SELECT title, p.pubid, nameFROM books NATURAL JOIN publisher in 0.125 seconds

```
ORA-00904: "P"."PUBID": invalid identifier
00904. 00000 -  "%s: invalid identifier"
*Cause:
*Action:
Error at Line: 1 Column: 14
```

You cannot qualify the column with a table alias, these are not permitted when the NATURAL JOIN is used

Using the NATURAL JOIN

- This warning is included in the text book on page 298
- Oracle does not recommend using this method, it can produce some unexpected results
- When using with more than two tables, and where there are more than multiple columns with the same name the NATURAL JOIN will do joins on all matching column names
- If a column named DESCRIPTION were added to both the BOOKS and PUBLISHERS tables with unrelated data the NATURAL JOIN would also attempt to join the tables on this column too
- You can end up with Cartesian and partial Cartesians and a result
- So it is recommended to use the alternate methods if there are more than one set of columns existing between the tables
- The other methods explicitly specify the columns being joined

Using the JOIN … USING

- As with the NATURAL JOIN, a column referenced by a USING clause cannot contain a column qualifier anywhere in the SELECT statement
- The column referenced in the USING clause must be enclosed in parentheses
- The other columns can be referenced by a column qualifier
- The USING clause requires a commonly named column to perform the join

Using the JOIN … USING

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

Script Output | Query...

All Rows Fetched: 14 in 0.016 seconds

	TITLE	PUBID	NAME
1	REVENGE OF MICKEY	1	PRINTING IS US
2	HOW TO MANAGE THE MANAGER	1	PRINTING IS US
3	E-BUSINESS THE EASY WAY	2	PUBLISH OUR WAY
4	BUILDING A CAR WITH TOOTHPICKS	2	PUBLISH OUR WAY
5	HOLY GRAIL OF ORACLE	3	AMERICAN PUBLISHING
6	DATABASE IMPLEMENTATION	3	AMERICAN PUBLISHING
7	HANDCRANKED COMPUTERS	3	AMERICAN PUBLISHING
8	COOKING WITH MUSHROOMS	4	READING MATERIALS INC.
9	THE WOK WAY TO COOK	4	READING MATERIALS INC.
10	HOW TO GET FASTER PIZZA	4	READING MATERIALS INC.
11	BODYBUILD IN 10 MINUTES A DAY	4	READING MATERIALS INC.
12	PAINLESS CHILD-REARING	5	REED-N-RITE
13	BIG BEAR AND LITTLE DOVE	5	REED-N-RITE
14	SHORTEST POEMS	5	REED-N-RITE

Using the WHERE Clause to Restrict Rows

- It is possible that ambiguity can exist when referencing columns with the JOIN … ON keywords. Oracle 11g permits the use of column qualifiers to avoid ambiguity
- Using the ON clause in a SELECT statement allows you to use the WHERE clause exclusively for restricting the rows to be included in the results
- This can improve the readability of complex SELECT statements for anyone not familiar with the traditional method of joining tables
- This is the logic behind the ANSI committee, they wanted the table joins to be done in the FROM clause and the WHERE clause to be used exclusively for restricting rows

Using the WHERE Clause to Restrict Rows

```
SELECT b.title, b.pubid, p.name
FROM books b JOIN publisher p
ON b.pubid = p.pubid
WHERE b.pubid = 4;
```

Script Output | Query...

All Rows Fetched: 4 in 0.016 seconds

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

WHERE used with the JOIN … ON keyword

Non-Equality Joins

- With an equality join, the data value of the stored record in the common column for the first table must match the data value in the second table
- There are many cases where the values will not match
- A non-equality join is used when the related columns cannot be joined through an equal sign
- For example, the shipping fee charged by a freight company is based on the weight of an item. In most cases, the fee is based on a range of weights. 1 to 5 pounds would be a certain fee

Non-Equality Joins

- A non-equality join enables you to store the minimum value for a range in one column and the maximum value for a range in another column
- Instead of finding a column-to-column match, you use the non-equality join to determine whether the weight of the item falls between the minimum and maximum ranges of the columns
- If the join finds a matching range for the item, the corresponding shipping fee is returned in the results
- Once a year, Just Lee Books offers a promotion in which customers receive a gift based on the value of each book purchased

Non-Equality Joins

- I f you look at the relationship that exists between the PROMOTION table and the BOOKS table you will notice there is no common column
- The RETAIL column of the BOOKS table is used as a comparison with the MINRETAIL and MAXRETAIL columns of the PROMOTION table

Non-Equality Joins - Traditional

```
SELECT b.title, p.gift, b.retail, p.minretail, p.maxretail
FROM books b,  promotion p
WHERE b.retail BETWEEN p.minretail AND p.maxretail;
```

Script Output | Query...

All Rows Fetched: 14 in 0.016 seconds

	TITLE	GIFT	RETAIL	MINRETAIL	MAXRETAIL
1	BIG BEAR AND LITTLE DOVE	BOOKMARKER	8.95	0	12
2	COOKING WITH MUSHROOMS	BOOK LABELS	19.95	12.01	25
3	REVENGE OF MICKEY	BOOK LABELS	22	12.01	25
4	HANDCRANKED COMPUTERS	BOOK LABELS	25	12.01	25
5	THE WOK WAY TO COOK	BOOK COVER	28.75	25.01	56
6	HOW TO GET FASTER PIZZA	BOOK COVER	29.95	25.01	56
7	BODYBUILD IN 10 MINUTES A DAY	BOOK COVER	30.95	25.01	56
8	HOW TO MANAGE THE MANAGER	BOOK COVER	31.95	25.01	56
9	SHORTEST POEMS	BOOK COVER	39.95	25.01	56
10	E-BUSINESS THE EASY WAY	BOOK COVER	54.5	25.01	56
11	DATABASE IMPLEMENTATION	BOOK COVER	55.95	25.01	56
12	BUILDING A CAR WITH TOOTHPICKS	FREE SHIPPING	59.95	56.01	999.99
13	HOLY GRAIL OF ORACLE	FREE SHIPPING	75.95	56.01	999.99
14	PAINLESS CHILD-REARING	FREE SHIPPING	89.95	56.01	999.99

Non-Equality Joins

- The BETWEEN … AND operator is used in the WHERE clause to determine the range in which the retail price of the book falls
- Then, depending on the minimum retail and maximum retail values in the PROMOTION table, the corresponding gift is returned for each purchase
- In order for this to work properly, you must make sure that none of the values in the PROMOTION table overlap. Also make sure that all possible values are contained in the various ranges of the PROMOTION table

Non-Equality Joins – JOIN Method

- A non-equality join using the JOIN keyword has the same syntax as an equality join with the JOIN keyword
- The only difference is the equal sign is not used to specify the relationship in the ON clause

Non-Equality Joins – JOIN Method

```
1 SELECT b.title, p.gift, b.retail, p.minretail, p.maxretail
2 FROM books b JOIN promotion p
3 ON b.retail BETWEEN p.minretail AND p.maxretail;
```

Script Output | Query...

All Rows Fetched: 14 in 0.016 seconds

	TITLE	GIFT	RETAIL	MINRETAIL	MAXRETAIL
1	BIG BEAR AND LITTLE DOVE	BOOKMARKER	8.95	0	12
2	COOKING WITH MUSHROOMS	BOOK LABELS	19.95	12.01	25
3	REVENGE OF MICKEY	BOOK LABELS	22	12.01	25
4	HANDCRANKED COMPUTERS	BOOK LABELS	25	12.01	25
5	THE WOK WAY TO COOK	BOOK COVER	28.75	25.01	56
6	HOW TO GET FASTER PIZZA	BOOK COVER	29.95	25.01	56
7	BODYBUILD IN 10 MINUTES A DAY	BOOK COVER	30.95	25.01	56
8	HOW TO MANAGE THE MANAGER	BOOK COVER	31.95	25.01	56
9	SHORTEST POEMS	BOOK COVER	39.95	25.01	56
10	E-BUSINESS THE EASY WAY	BOOK COVER	54.5	25.01	56
11	DATABASE IMPLEMENTATION	BOOK COVER	55.95	25.01	56
12	BUILDING A CAR WITH TOOTHPICKS	FREE SHIPPING	59.95	56.01	999.99
13	HOLY GRAIL OF ORACLE	FREE SHIPPING	75.95	56.01	999.99
14	PAINLESS CHILD-REARING	FREE SHIPPING	89.95	56.01	999.99

Exercise Before we Begin

- Write a query to display the book PUBID, TITLE the NAME of the publisher and the CONTACT at the publisher. Only books published by pubid 2 or 4 are to be displayed. The results are to be sorted by the title of the books.
- Do this query two ways, the traditional method and the JOIN … ON method.

Self-Joins

- Sometimes, the data in a table references other data stored in the same table
- For example, customers who refer a new customer to Just Lee Books receive a discount certificate for a future purchase
- The Referred column of the CUSTOMERS table stores the customer number of the individual who referred the new customer
- The Referred column actually relates to other rows in the same table

Self-Joins

Customer 1003 (Leila Smith) has referred two customers (Tammy Giana and Jorge Perez)

CUSTOMER#	LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	REFERRED
1001	MORALES	BONITA	P.O. BOX 651	EASTPOINT	FL	32328	
1002	THOMPSON	RYAN	P.O. BOX 9835	SANTA MONICA	CA	90404	
1003	SMITH	LEILA	P.O. BOX 66	TALLAHASSEE	FL	32306	
1004	PIERSON	THOMAS	69821 SOUTH AVENUE	BOISE	ID	83707	
1005	GIRARD	CINDY	P.O. BOX 851	SEATTLE	WA	98115	
1006	CRUZ	MESHIA	82 DIRT ROAD	ALBANY	NY	12211	
1007	GIANA	TAMMY	9153 MAIN STREET	AUSTIN	TX	78710	1003
1008	JONES	KENNETH	P.O. BOX 137	CHEYENNE	WY	82003	
1009	PEREZ	JORGE	P.O. BOX 8564	BURBANK	CA	91510	1003
1010	LUCAS	JAKE	114 EAST SAVANNAH	ATLANTA	GA	30314	
1011	MCGOVERN	REESE	P.O. BOX 18	CHICAGO	IL	60606	
1012	MCKENZIE	WILLIAM	P.O. BOX 971	BOSTON	MA	02110	
1013	NGUYEN	NICHOLAS	357 WHITE EAGLE AVE.	CLERMONT	FL	34711	1006

Customer 1006 (Meshia Cruz) has referred one customer (Nicholas Nguyen)

FIGURE 9-22 Two columns of the same table are related (Partial table shown)

Self-Joins

- You must join table to itself to retrieve the information you need
- This is referred to as a self-join
- You can create the self-join either with the WHERE clause, or using the JOIN with the ON clause

Self-Joins – Traditional Method

- To perform a self-join, the CUSTOMERS table must be listed twice in the FROM clause
- You must treat it as if it is two separate tables
- You must assign each specification of the CUSTOMER table a different table alias
- The query uses the table alias "c" to identify the containing customer and the alias "r" to identify the table storing the individual who referred the new customer
- Since the table aliases are different, Oracle 11g is able to select records within the same table while executing the query

Self-Joins – Traditional Method

```
SELECT r.firstname, r.lastname, c.lastname "Referred"
FROM customers c, customers r
WHERE c.referred = r.customer#;
```

Script Output | Query Result

All Rows Fetched: 5 in 0.078 seconds

	FIRSTNAME	LASTNAME	Referred
1	LEILA	SMITH	GIANA
2	LEILA	SMITH	PEREZ
3	LEILA	SMITH	SMITH
4	MESHIA	CRUZ	NGUYEN
5	JAKE	LUCAS	DAUM

Self-Joins – JOIN Method

- Regardless of which method is used, the same concept applies: to make it appear that two different tables are being joined together using table aliases
- Using the JOIN … ON approach to create a self-join will still allow you to place row restrictions in a WHERE clause

Self-Joins – JOIN Method

```
1  SELECT r.firstname, r.lastname, c.lastname "Referred"
2  FROM customers c JOIN customers r
3  ON c.referred = r.customer#;
```

Script Output | Query Result

All Rows Fetched: 5 in 0.015 seconds

	FIRSTNAME	LASTNAME	Referred
1	LEILA	SMITH	GIANA
2	LEILA	SMITH	PEREZ
3	LEILA	SMITH	SMITH
4	MESHIA	CRUZ	NGUYEN
5	JAKE	LUCAS	DAUM

Outer Joins

- When performing equality, non-equality and self-joins, a row is returned if there was a corresponding record in each table queried
- These types of joins are all classified as inner joins because records are only returned if a match is found in each table
- As we saw earlier, the default keyword INNER can be included with the JOIN keyword to specify that only the records having a matching row in the corresponding table should be returned in the results

Outer Joins

- Suppose you want a list of all customers and the order numbers of any order the customer has placed
- The CUSTOMER table contains a list of all customers who have ever placed an order
- The ORDERS table lists just the current month's orders or any unfilled orders from the previous month
- An inner join may not give you the exact results you are looking for because a customer might not have placed a recent order
- The inner join drops any non-matching rows

Outer Joins

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

Script Output | Query Result

All Rows Fetched: 21 in 0.109 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	FALAH	KENNETH	1004
2	FALAH	KENNETH	1015
3	GIANA	TAMMY	1014
4	GIANA	TAMMY	1007
5	GIRARD	CINDY	1009
6	GIRARD	CINDY	1000
7	JONES	KENNETH	1020
8	LEE	JASMINE	1013
9	LUCAS	JAKE	1001
10	LUCAS	JAKE	1011
11	MCGOVERN	REESE	1002
12	MONTIASA	GREG	1019
13	MONTIASA	GREG	1005
14	MORALES	BONITA	1018
15	MORALES	BONITA	1003
16	NELSON	BECCA	1012
17	PIERSON	THOMAS	1008
18	SCHELL	STEVE	1017
19	SMITH	JENNIFER	1010
20	SMITH	LEILA	1016
21	SMITH	LEILA	1006

Only the matched rows appear in this query

It does not show any customers who have not recently placed an order, they are omitted from the query since no match is found between the column in the two tables

Outer Joins

- The previous query identifies any customer who has placed an order that is stored in the ORDERS table
- However, it does not list customers who have not recently placed an order
- The request is to list all customers, so you need to modify the query
- When you need to include records in the results of a joining query that exist in one table but do not have a corresponding row in the other table, you need an OUTER JOIN

Outer Joins

- The keywords OUTER JOIN instructs Oracle 11g to include a record in the output even though there is no matching record in the corresponding table
- Oracle 11g will join the row from one table to a NULL record in the other table
- An outer join can be created either in the WHERE clause with an ***outer join operator*** (+) or by using the OUTER JOIN keywords

Outer Joins – Traditional Method

If you look at the output you will notice in the ORDER# column there are values of (null)

These indicate customers that have not placed orders, hence the absence of the ORDER# in the column

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

Script Output | Query...

All Rows Fetched: 27 in 0.031 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	CRUZ	MESHIA	(null)
2	DAUM	MICHELL	(null)
3	FALAH	KENNETH	1004
4	FALAH	KENNETH	1015
5	GIANA	TAMMY	1007
6	GIANA	TAMMY	1014
7	GIRARD	CINDY	1009
8	GIRARD	CINDY	1000
9	JONES	KENNETH	1020
10	LEE	JASMINE	1013
11	LUCAS	JAKE	1001
12	LUCAS	JAKE	1011
13	MCGOVERN	REESE	1002
14	MCKENZIE	WILLIAM	(null)
15	MONTIASA	GREG	1005
16	MONTIASA	GREG	1019
17	MORALES	BONITA	1018
18	MORALES	BONITA	1003
19	NELSON	BECCA	1012
20	NGUYEN	NICHOLAS	(null)
21	PEREZ	JORGE	(null)
22	PIERSON	THOMAS	1008
23	SCHELL	STEVE	1017
24	SMITH	JENNIFER	1010
25	SMITH	LEILA	1016
26	SMITH	LEILA	1006
27	THOMPSON	RYAN	(null)

Outer Joins – Traditional Method

- If a customer is in the CUSTOMERS table but has not placed an order, the ORDERS table will lack the corresponding row – or it will be the deficient table, the table with the missing data
- The join operator (+) is placed immediately after the portion of the joining condition in the WHERE clause that references the deficient ORDERS table
- The CUSTOMERS table contains all existing customers but the ORDERS table only has the customers that have recently placed an order
- If I also want to see any customers that have not recently placed an order the outer join marker (+) is placed against the table that is lacking the customers, this is the ORDERS table

Outer Joins – Traditional Method

- There are two rules to remember when working with traditional outer joins:
  - The outer join operator can only be used for one table in the joining condition. The outer join operator cannot be used on both sides of the joining condition at the same time
  - A condition that includes the outer join operator cannot use the IN or the OR operator because that would imply that a row should be shown if it matches a row in the other table or if it matches another given condition

Outer Joins – JOIN Method

- If you use the JOIN keyword, you can create a left, right or full outer join
- The JOIN keyword alone, by default, indicates an inner join
- For an outer join, you must include the keyword RIGHT, LEFT or FULL with the JOIN keyword to identify the type of join
- The OUTER keyword can also be included but it is optional and can be omitted

Outer Joins – JOIN Method

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

Script Output | Query...

All Rows Fetched: 27 in 0.031 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	CRUZ	MESHIA	(null)
2	DAUM	MICHELL	(null)
3	FALAH	KENNETH	1004
4	FALAH	KENNETH	1015
5	GIANA	TAMMY	1007
6	GIANA	TAMMY	1014
7	GIRARD	CINDY	1009
8	GIRARD	CINDY	1000
9	JONES	KENNETH	1020
10	LEE	JASMINE	1013
11	LUCAS	JAKE	1001
12	LUCAS	JAKE	1011
13	MCGOVERN	REESE	1002
14	MCKENZIE	WILLIAM	(null)
15	MONTIASA	GREG	1005
16	MONTIASA	GREG	1019
17	MORALES	BONITA	1018
18	MORALES	BONITA	1003
19	NELSON	BECCA	1012
20	NGUYEN	NICHOLAS	(null)
21	PEREZ	JORGE	(null)
22	PIERSON	THOMAS	1008
23	SCHELL	STEVE	1017
24	SMITH	JENNIFER	1010
25	SMITH	LEILA	1016
26	SMITH	LEILA	1006
27	THOMPSON	RYAN	(null)

This query recreates the original query, but this one uses the outer join operator

LEFT OUTER JOIN says the table on the left has all the customer# and the one on the right is missing the customer# so the null keyword is inserted for any customer on the left that does not have an order

Outer Joins – JOIN Method

- The table listed on the left side of the joining condition given in the FROM clause has the unmatched records
- If the RIGHT OUTER JOIN were used with the balance of the query remaining the same, you would have seen any orders that did not have a customer since the customers would then be deficient of the data

Outer Joins – JOIN Method

Since all orders have customers, no null values for customers are indicated

Good way to check if there were any orders that did not have customers attached to them

Run Statement (Ctrl+Enter)

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

Script Output | Query...

All Rows Fetched: 21 in 0.032 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	FALAH	KENNETH	1015
2	FALAH	KENNETH	1004
3	GIANA	TAMMY	1007
4	GIANA	TAMMY	1014
5	GIRARD	CINDY	1009
6	GIRARD	CINDY	1000
7	JONES	KENNETH	1020
8	LEE	JASMINE	1013
9	LUCAS	JAKE	1011
10	LUCAS	JAKE	1001
11	MCGOVERN	REESE	1002
12	MONTIASA	GREG	1005
13	MONTIASA	GREG	1019
14	MORALES	BONITA	1018
15	MORALES	BONITA	1003
16	NELSON	BECCA	1012
17	PIERSON	THOMAS	1008
18	SCHELL	STEVE	1017
19	SMITH	JENNIFER	1010
20	SMITH	LEILA	1016
21	SMITH	LEILA	1006

Outer Joins – JOIN Method

However, you can reverse the tables used in the FROM clause and then use the RIGHT OUTER JOIN to retrieve the original results, ORDERS again is the deficient table and the table on the right contains all possible values for customer

This is not exactly the same query the two tables have been switched

```
1 SELECT c.lastname, c.firstname, o.order#
2 FROM orders o RIGHT OUTER JOIN customers c
3 ON c.customer# = o.customer#
4 ORDER BY c.lastname, c.firstname;
```

Script Output | Query Result

All Rows Fetched: 27 in 0.016 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	CRUZ	MESHIA	(null)
2	DAUM	MICHELL	(null)
3	FALAH	KENNETH	1004
4	FALAH	KENNETH	1015
5	GIANA	TAMMY	1007
6	GIANA	TAMMY	1014
7	GIRARD	CINDY	1009
8	GIRARD	CINDY	1000
9	JONES	KENNETH	1020
10	LEE	JASMINE	1013
11	LUCAS	JAKE	1001
12	LUCAS	JAKE	1011
13	MCGOVERN	REESE	1002
14	MCKENZIE	WILLIAM	(null)
15	MONTIASA	GREG	1005
16	MONTIASA	GREG	1019
17	MORALES	BONITA	1018
18	MORALES	BONITA	1003
19	NELSON	BECCA	1012
20	NGUYEN	NICHOLAS	(null)
21	PEREZ	JORGE	(null)
22	PIERSON	THOMAS	1008
23	SCHELL	STEVE	1017
24	SMITH	JENNIFER	1010
25	SMITH	LEILA	1016
26	SMITH	LEILA	1006
27	THOMPSON	RYAN	(null)

Outer Joins – JOIN Method

As mentioned previously, the OUTER keyword is optional and can be omitted. Here is the same query executed without the OUTER keyword

Any time you see the words RIGHT, LEFT or FULL used in a join it will be an OUTER JOIN

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

Script Output | Query Result

All Rows Fetched: 27 in 0.031 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	CRUZ	MESHIA	(null)
2	DAUM	MICHELL	(null)
3	FALAH	KENNETH	1004
4	FALAH	KENNETH	1015
5	GIANA	TAMMY	1007
6	GIANA	TAMMY	1014
7	GIRARD	CINDY	1009
8	GIRARD	CINDY	1000
9	JONES	KENNETH	1020
10	LEE	JASMINE	1013
11	LUCAS	JAKE	1001
12	LUCAS	JAKE	1011
13	MCGOVERN	REESE	1002
14	MCKENZIE	WILLIAM	(null)
15	MONTIASA	GREG	1005
16	MONTIASA	GREG	1019
17	MORALES	BONITA	1018
18	MORALES	BONITA	1003
19	NELSON	BECCA	1012
20	NGUYEN	NICHOLAS	(null)
21	PEREZ	JORGE	(null)
22	PIERSON	THOMAS	1008
23	SCHELL	STEVE	1017
24	SMITH	JENNIFER	1010
25	SMITH	LEILA	1016
26	SMITH	LEILA	1006
27	THOMPSON	RYAN	(null)

Outer Joins – JOIN Method

- Substituting the FULL JOIN keywords would instruct Oracle 11g to return records from either table if there is no matching record in the other table
- When using the traditional method, the full outer join is not available

Outer Joins – JOIN Method

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

Script Output | Query...

All Rows Fetched: 27 in 0.046 seconds

	LASTNAME	FIRSTNAME	ORDER#
1	CRUZ	MESHIA	(null)
2	DAUM	MICHELL	(null)
3	FALAH	KENNETH	1004
4	FALAH	KENNETH	1015
5	GIANA	TAMMY	1014
6	GIANA	TAMMY	1007
7	GIRARD	CINDY	1009
8	GIRARD	CINDY	1000
9	JONES	KENNETH	1020
10	LEE	JASMINE	1013
11	LUCAS	JAKE	1001
12	LUCAS	JAKE	1011
13	MCGOVERN	REESE	1002
14	MCKENZIE	WILLIAM	(null)
15	MONTIASA	GREG	1005
16	MONTIASA	GREG	1019
17	MORALES	BONITA	1003
18	MORALES	BONITA	1018
19	NELSON	BECCA	1012
20	NGUYEN	NICHOLAS	(null)
21	PEREZ	JORGE	(null)
22	PIERSON	THOMAS	1008
23	SCHELL	STEVE	1017
24	SMITH	JENNIFER	1010
25	SMITH	LEILA	1016
26	SMITH	LEILA	1006
27	THOMPSON	RYAN	(null)

The results shown are the same since we already know there are no orders that do not have a customer, only customers that have no orders

Joining Three or More Tables

- There are times that you will need to retrieve data from three or more tables
- For example, if you wanted to retrieve the name of each book that has been purchased by each customer
- You would need to know the name of the customer and the orders that the customer has placed (CUSTOMERS and ORDERS)
- Then, the ISBN number of each book on each line of the order (ORDERS and ORDERITEMS), then the name of each book on each line (ORDERITEMS and BOOKS)
- This requires 4 tables in total

Joining Three or More Tables – Traditional Method

```
SELECT c.lastname, c.firstname, b.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
AND category = 'COMPUTER'
ORDER BY lastname, firstname;
```

Query Result

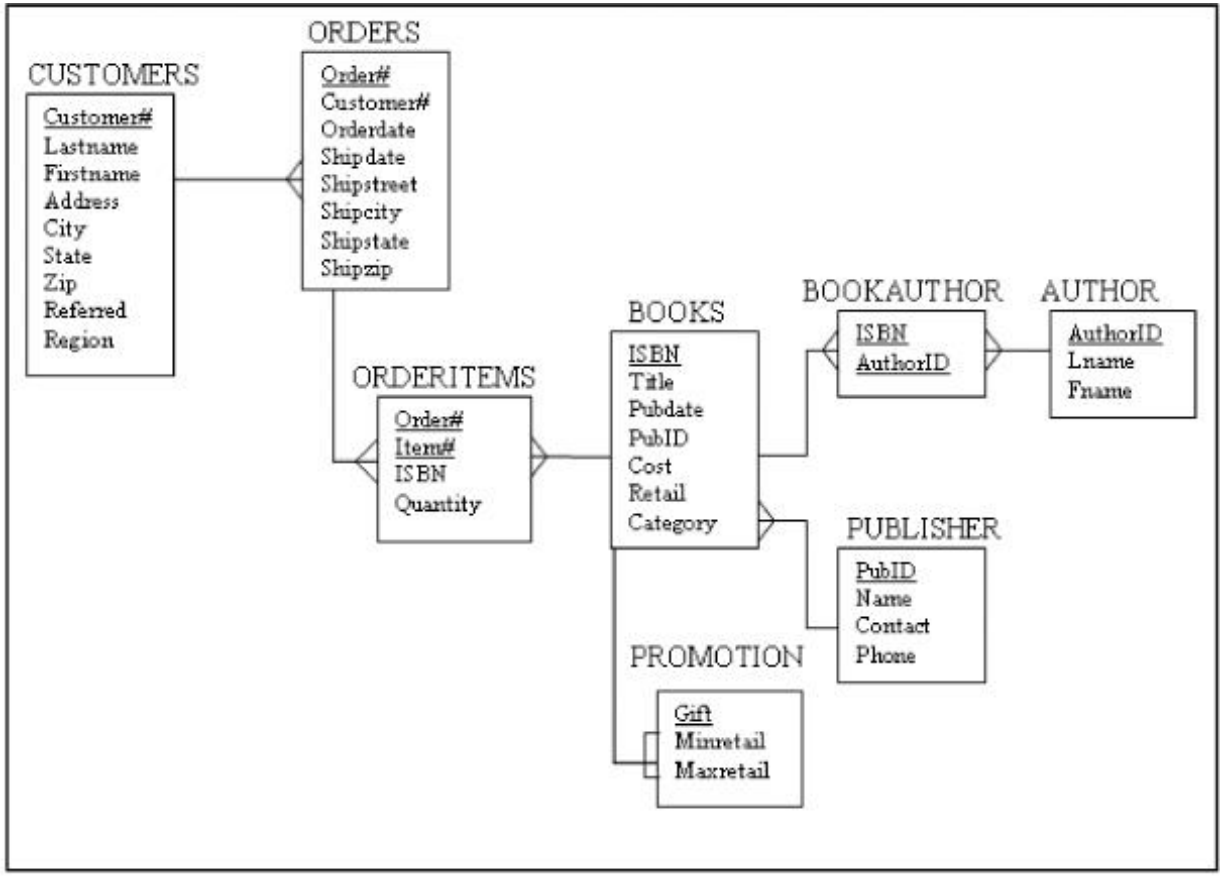
All Rows Fetched: 10 in 0.031 seconds

	LASTNAME	FIRSTNAME	TITLE
1	GIANA	TAMMY	DATABASE IMPLEMENTATION
2	GIANA	TAMMY	E-BUSINESS THE EASY WAY
3	GIANA	TAMMY	HOLY GRAIL OF ORACLE
4	LEE	JASMINE	DATABASE IMPLEMENTATION
5	MCGOVERN	REESE	DATABASE IMPLEMENTATION
6	MORALES	BONITA	DATABASE IMPLEMENTATION
7	MORALES	BONITA	DATABASE IMPLEMENTATION
8	NELSON	BECCA	HANDCRANKED COMPUTERS
9	SMITH	JENNIFER	DATABASE IMPLEMENTATION
10	SMITH	LEILA	E-BUSINESS THE EASY WAY

Joining Three or More Tables – Traditional Method

- The WHERE clause is used to define the access path that Oracle 11g should take to relate the various tables
- Notice the three conditions that are required to establish the relationships among the four tables
- There will always be one less condition than the number of tables being joined; 4 tables, and 3 conditions
- Look back at the SELECT statement to see the tables and the relationships that are used
- Compare this to the schema shown on the next slide, you will see to retrieve the lastname, firstname and the tilte you need to follow through the 4 tables in order to retrieve the three column values

Joining Three or More Tables



Note: Underlines denote primary key columns

FIGURE 9-6 JustLee Book's table structure

Joining Three or More Tables – JOIN Method

- To accomplish the joining of three or more tables, we will use a JOIN … ON method
- Here we will join the four tables as before
- Every time the JOIN keyword is used, it must have the ON keyword supplied with it
- The first join appears as we have described already, then the third join is added to it then finally the fourth join condition is added after the third one

Joining Three or More Tables – JOIN Method

- Although this appears to be correct, it is flawed
- The AND clause should be a WHERE clause to keep the syntax as it was intended
- FROM clause for tables
- WHERE clause to apply restrictions

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

Query Result

All Rows Fetched: 10 in 0.047 seconds

	LASTNAME	FIRSTNAME	TITLE
1	GIANA	TAMMY	DATABASE IMPLEMENTATION
2	GIANA	TAMMY	E-BUSINESS THE EASY WAY
3	GIANA	TAMMY	HOLY GRAIL OF ORACLE
4	LEE	JASMINE	DATABASE IMPLEMENTATION
5	MCGOVERN	REESE	DATABASE IMPLEMENTATION
6	MORALES	BONITA	DATABASE IMPLEMENTATION
7	MORALES	BONITA	DATABASE IMPLEMENTATION
8	NELSON	BECCA	HANDCRANKED COMPUTERS
9	SMITH	JENNIFER	DATABASE IMPLEMENTATION
10	SMITH	LEILA	E-BUSINESS THE EASY WAY

Joining Three or More Tables – JOIN Method

Worksheet | Query Builder

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

Query Result

SQL | All Rows Fetched: 10 in 0.063 seconds

	LASTNAME	FIRSTNAME	TITLE
1	GIANA	TAMMY	DATABASE IMPLEMENTATION
2	GIANA	TAMMY	E-BUSINESS THE EASY WAY
3	GIANA	TAMMY	HOLY GRAIL OF ORACLE
4	LEE	JASMINE	DATABASE IMPLEMENTATION
5	MCGOVERN	REESE	DATABASE IMPLEMENTATION
6	MORALES	BONITA	DATABASE IMPLEMENTATION
7	MORALES	BONITA	DATABASE IMPLEMENTATION
8	NELSON	BECCA	HANDCRANKED COMPUTERS
9	SMITH	JENNIFER	DATABASE IMPLEMENTATION
10	SMITH	LEILA	E-BUSINESS THE EASY WAY

- This is the correct method to use a condition
- JOINs are done in the FROM clause and the restrictions are accomplished in the WHERE clause

Joining Three or More Tables – JOIN Method

- The same rule applies as with the traditional join; the number of joins required is one less than the number of tables to be joined
- In the ANSI join, you join two tables, then add the third table to the previous two, then add the fourth table to the previous 3
- It does provide a more structured method when selecting data from more than 2 tables, I believe it is harder to omit any of tables and join conditions because you use them in the order they appear in the schema
- The traditional method is more free-from, there is no order that has to be adhered to when specifying the tables, and there is no order to the join clauses when listing them

Joining Tables

- Doing table joins for the first time will cause you some confusion, do not panic it happens to everyone
- The schema diagram is the tool you will use since it shows you how the tables are interconnected or joined together
- Use this as a guide and then look for the foreign key to primary key connections established between the tables
- For the next assignment do not wait till the last minute you probably will have some issues
- The next lab will have you do some practice sessions with multiple tables

Oracle Explain Plans

Worksheet | Query Builder

```
1 SELECT c.customer#, lastname, firstname, o.order#, shipdate
2 from customers c, orders o
3 WHERE c.customer# = o.customer#
4 AND c.customer# = 1014;
```

Query Result | Explain Plan

SQL | 0.188 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	4
NESTED LOOPS			1	4
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1	1
INDEX	CUSTOMERS_CUSTOMER#_PK	UNIQUE SCAN	1	0
Access Predicates				
C.CUSTOMER#=1014				
TABLE ACCESS	ORDERS	FULL	1	3
Filter Predicates				
O.CUSTOMER#=1014				

2 tables joined with traditional method

Oracle Explain Plans

Worksheet | Query Builder

```
1 SELECT c.customer#, lastname, firstname, o.order#, shipdate
2 from customers c JOIN orders o
3 ON c.customer# = o.customer#
4 AND c.customer# = 1014;
```

Query Result | Explain Plan

SQL | 0.172 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	4
NESTED LOOPS			1	4
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1	1
INDEX	CUSTOMERS_CUSTOMER#_PK	UNIQUE SCAN	1	0
Access Predicates				
C.CUSTOMER#=1014				
TABLE ACCESS	ORDERS	FULL	1	3
Filter Predicates				
O.CUSTOMER#=1014				

2 tables joined with ANSI JOIN … ON

Oracle Explain Plans

4 table joined with JOIN … ON

Worksheet | Query Builder

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

Script Output | Query Result | Explain Plan

SQL | 0.063 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY
SELECT STATEMENT			
SORT		ORDER BY	
HASH JOIN			
Access Predicates			
C.CUSTOMER#=O.CUSTOMER#			
NESTED LOOPS			
NESTED LOOPS			
STATISTICS COLLECTOR			
HASH JOIN			
Access Predicates			
O.ORDER#=OI.ORDER#			
NESTED LOOPS			
STATISTICS COLLEC			
MERGE JOIN			
TABLE ACCE	BOOKS	BY INDEX ROWID	
Filter Predicates			
B.CATEGORY='COMPUTER'			
INDEX	BOOKS_ISBN_PK	FULL SCAN	
SORT		JOIN	
Access Predicates			
OI.ISBN=B.ISBN			
Filter Predicates			

Oracle Explain Plan Using Plan Table

Worksheet | Query Builder

```
EXPLAIN  PLAN FOR
SELECT c.lastname, c.firstname, b.title
FROM customers c JOIN orders o
ON c.customer# = o.customer#
JOIN orderitems oi
ON o.order# = oi.order#
JOIN books b
ON oi.isbn = b.isbn
WHERE category = 'COMPUTER'
ORDER BY lastname, firstname;

SELECT *
FROM   TABLE(DBMS_XPLAN.DISPLAY);
```

Script Output | Query Result

SQL | All Rows Fetched: 30 in 0.141 seconds

PLAN_TABLE_OUTPUT

```
Plan hash value: 3047123743

-----------------------------------------------------------------------------------------------
| Id  | Operation                        | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                 |               |     5 |   430 |    13  (16)| 00:00:01 |
|   1 |  SORT ORDER BY                   |               |     5 |   430 |    13  (16)| 00:00:01 |
|*  2 |   HASH JOIN                      |               |     5 |   430 |    12   (9)| 00:00:01 |
|*  3 |    HASH JOIN                     |               |     5 |   340 |     9  (12)| 00:00:01 |
|   4 |     MERGE JOIN                   |               |     5 |   300 |     6  (17)| 00:00:01 |
|*  5 |      TABLE ACCESS BY INDEX ROWID| BOOKS         |     2 |    90 |     2   (0)| 00:00:01 |
|   6 |       INDEX FULL SCAN            | BOOKS_ISBN_PK |    14 |       |     1   (0)| 00:00:01 |
|*  7 |      SORT JOIN                   |               |    32 |   480 |     4  (25)| 00:00:01 |
|   8 |       TABLE ACCESS FULL          | ORDERITEMS    |    32 |   480 |     3   (0)| 00:00:01 |
```

Oracle Explain Plan Using Plan Table

Script Output × Query Result ×

SQL | All Rows Fetched: 30 in 0.141 seconds

```
PLAN_TABLE_OUTPUT
Plan hash value: 3047123743

-------------------------------------------------------------------------------------------------
| Id  | Operation                        | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                 |               |     5 |   430 |    13  (16)| 00:00:01 |
|   1 |  SORT ORDER BY                   |               |     5 |   430 |    13  (16)| 00:00:01 |
|*  2 |   HASH JOIN                      |               |     5 |   430 |    12   (9)| 00:00:01 |
|*  3 |    HASH JOIN                     |               |     5 |   340 |     9  (12)| 00:00:01 |
|   4 |     MERGE JOIN                   |               |     5 |   300 |     6  (17)| 00:00:01 |
|*  5 |      TABLE ACCESS BY INDEX ROWID| BOOKS         |     2 |    90 |     2   (0)| 00:00:01 |
|   6 |       INDEX FULL SCAN            | BOOKS_ISBN_PK |    14 |       |     1   (0)| 00:00:01 |
|*  7 |      SORT JOIN                   |               |    32 |   480 |     4  (25)| 00:00:01 |
|   8 |       TABLE ACCESS FULL          | ORDERITEMS    |    32 |   480 |     3   (0)| 00:00:01 |
|   9 |     TABLE ACCESS FULL            | ORDERS        |    21 |   168 |     3   (0)| 00:00:01 |
|  10 |    TABLE ACCESS FULL             | CUSTOMERS     |    20 |   360 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("C"."CUSTOMER#"="O"."CUSTOMER#")
   3 - access("O"."ORDER#"="OI"."ORDER#")
   5 - filter("B"."CATEGORY"='COMPUTER')
   7 - access("OI"."ISBN"="B"."ISBN")
       filter("OI"."ISBN"="B"."ISBN")
```

Oracle Explain Plans

Worksheet | Query Builder

```
SELECT c.customer#, lastname, firstname, o.order#, shipdate, title
from customers c, orders o, orderitems oe, books b
WHERE c.customer# = o.customer#
AND oe.order# = o.order#
AND oe.isbn = b.isbn
AND c.customer# = 1014;
```

Query Result | Explain Plan

SQL | 0.219 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	8
HASH JOIN			2	8
Access Predicates				
OE.ISBN=B.ISBN				
NESTED LOOPS			2	8
NESTED LOOPS			2	8
STATISTICS COLLECTOR				
HASH JOIN			2	6
Access Predicates				
OE.ORDER#=O.ORDER#				
NESTED LOOPS			2	6
STATISTICS COLLECTOR				
NESTED LOOPS			1	4
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1	1
INDEX	CUSTOMERS_CUSTOMER#_PK	UNIQUE SCAN	1	0
Access Predicates				
C.CUSTOMER#=1014				
TABLE ACCESS	ORDERS	FULL	1	3
Filter Predicates				
O.CUSTOMER#=1014				
TABLE ACCESS	ORDERITEMS	BY INDEX ROWID BATCHED	2	2
INDEX	ORDERITEMS_PK	RANGE SCAN	2	1
Access Predicates				
OE.ORDER#=O.ORDER#				
TABLE ACCESS	ORDERITEMS	FULL	2	2
INDEX	BOOKS_ISBN_PK	UNIQUE SCAN	1	0
Access Predicates				
OE.ISBN=B.ISBN				
TABLE ACCESS	BOOKS	BY INDEX ROWID	1	1
TABLE ACCESS	BOOKS	FULL	1	1