Oracle SQL

Joining Data from Multiple Tables

Non-Equality Joins

- In WHERE clause, use any comparison operator other than the equal sign
- In FROM clause, use JOIN…ON keywords with a non-equivalent condition

Non-Equality Joins: WHERE Clause Example

Enter SQL Statement:

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

Non-Equality Joins: JOIN…ON Example

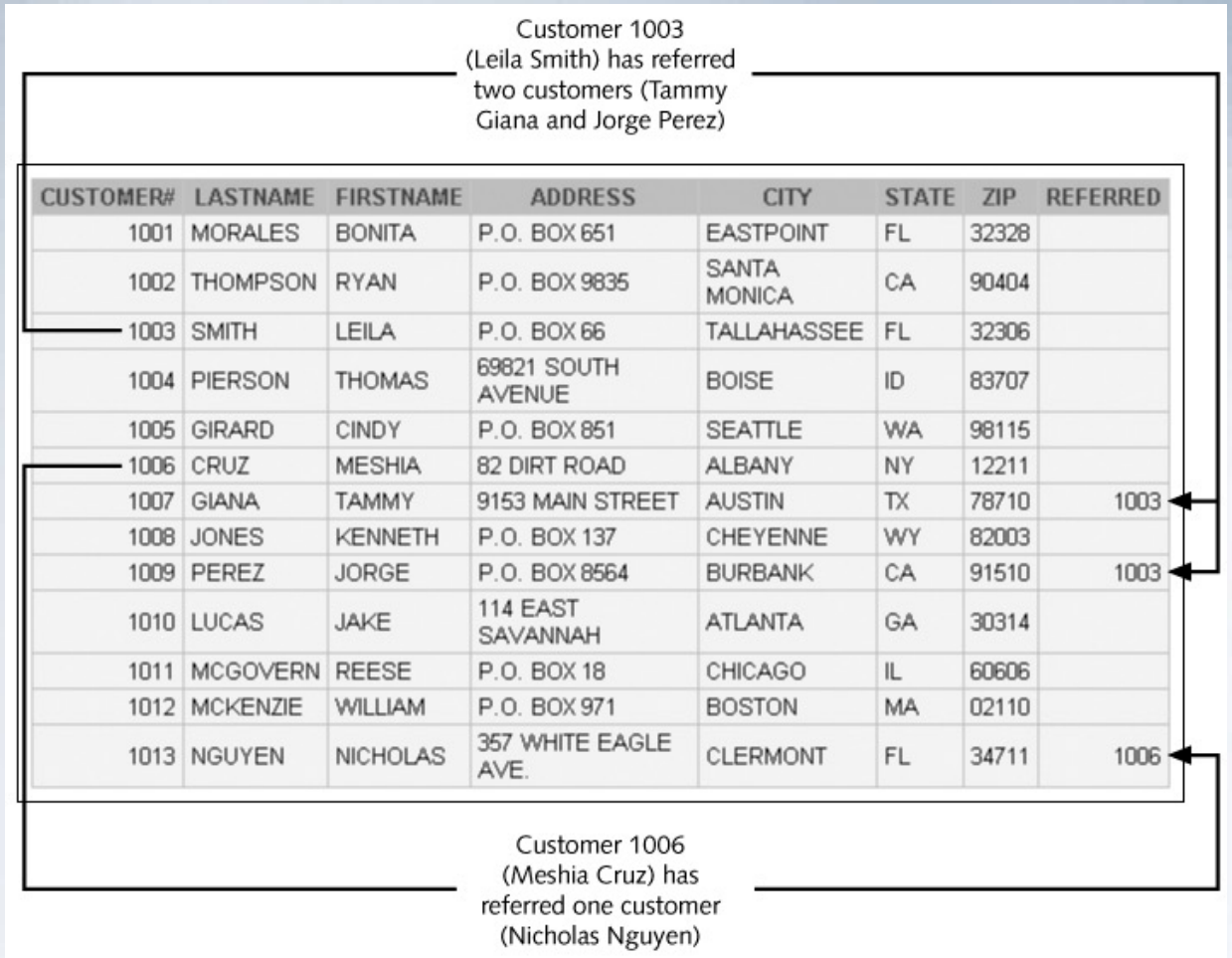
Enter SQL Statement:

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

Self-Joins

- Used to link a table to itself
- Requires the use of table aliases
- Requires the use of a column qualifier

Customer Table Example



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

Self-Joins: WHERE Clause Example

Enter SQL Statement:

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

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

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

Self-Joins: JOIN…ON Example

Enter SQL Statement:

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

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Results:

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

Outer Joins

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

Outer Joins: WHERE Clause Example

Enter SQL Statement:

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

Outer Joins: OUTER JOIN Keyword Example

Enter SQL Statement:

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

Left Outer joins explained

id	title	category
1	ASSASSIN'S CREED: EMBERS	Animations
2	Real Steel(2012)	Animations
3	Alvin and the Chipmunks	Animations
4	The Adventures of Tin Tin	Animations
5	Safe (2012)	Action
6	Safe House(2012)	Action
7	GIA	18+
8	Deadline 2009	18+
9	The Dirty Picture	18+
10	Marley and me	Romance

id	first_name	last_name	movie_id
1	Adam	Smith	1
2	Ravi	Kumar	2
3	Susan	Davidson	5
4	Jenny	Adrianna	8
6	Lee	Pong	10

title	first_name	last_name
ASSASSIN'S CREED: EMBERS	Adam	Smith
Real Steel(2012)	Ravi	Kumar
Safe (2012)	Susan	Davidson
Deadline(2009)	Jenny	Adrianna
Marley and me	Lee	Pong
Alvin and the Chipmunks	NULL	NULL
The Adventures of Tin Tin	NULL	NULL
Safe House(2012)	NULL	NULL
GIA	NULL	NULL
The Dirty Picture	NULL	NULL

Note: Null is returned for non-matching rows on right

Right Outer Join

first_name	last_name	title
Adam	Smith	ASSASSIN'S CREED: EMBERS
Ravi	Kumar	Real Steel(2012)
Susan	Davidson	Safe (2012)
Jenny	Adrianna	Deadline(2009)
Lee	Pong	Marley and me
NULL	NULL	Alvin and the Chipmunks
NULL	NULL	The Adventures of Tin Tin
NULL	NULL	Safe House(2012)
NULL	NULL	GIA
NULL	NULL	The Dirty Picture

Note: Null is returned for non-matching rows on left

Set Operators

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

Set Operator	Description
UNION	Returns the results of both queries and removes duplicates
UNION ALL	Returns the results of both queries but includes duplicates
INTERSECT	Returns only the rows included in the results of both queries
MINUS	Subtracts the second query's results if they're also returned in the first query's results

Set Operators: UNION Example

Enter SQL Statement:

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

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

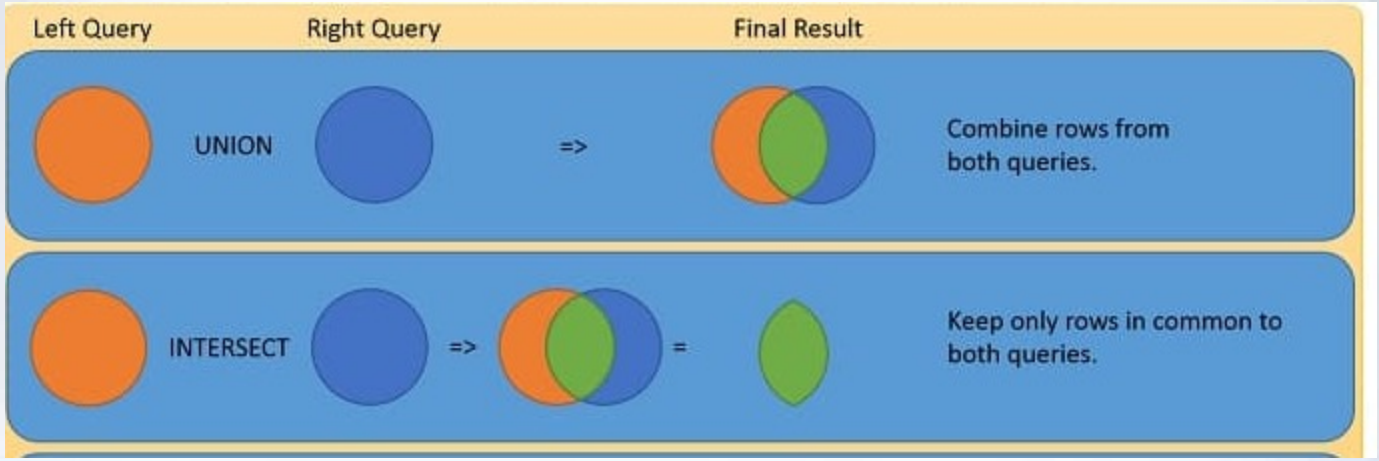
Results:

	AUTHORID
1	B100
2	F100
3	J100
4	K100
5	R100

Set Operators: INTERSECT Example

Enter SQL Statement:

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



Set Operators: MINUS Example

Enter SQL Statement:

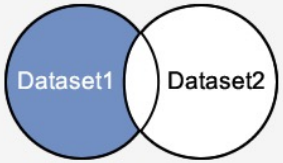
```
SELECT customer#
 FROM customers
MINUS
SELECT customer#
 FROM orders;
```

Results | Script Output | Explain | Autotrace | DBMS Output

Results:

	CUSTOMER#
1	1002
2	1006
3	1009
4	1012
5	1013
6	1016

Minus Query



Explanation: The MINUS query will return the records in the blue shaded area. These are the records that exist in Dataset1 and not in Dataset2.

Summary

- Data stored in multiple tables regarding a single entity can be linked together through the use of joins
- A Cartesian join between two tables returns every possible combination of rows from the tables; the resulting number of rows is always *m* * *n*
- An equality join is created when the data joining the records from two different tables are an exact match
- A non-equality join establishes a relationship based upon anything other than an equal condition
- Self-joins are used when a table must be joined to itself to retrieve needed data

Summary (continued)

- Inner joins are categorized as being equality, non-equality, or self-joins
- An outer join is created when records need to be included in the results without having corresponding records in the join tables
  - The record is matched with a NULL record so it will be included in the output
- Set operators such as UNION, UNION ALL, INTERSECT, and MINUS can be used to combine the results of multiple queries