Murach Chapter 4 Part 3

How to Retrieve Data From Two or More Tables

Week 3, Lecture 6

Knowledge Points in this lecture

- SELF JOIN
- In-Class Practice about
  - SELF JOIN
- OUTER JOIN
  - LEFT, RIGHT, FULL
- DIFFERENCE between OUTER JOIN and INNER JOIN
- In-Class Practice about
  - INNER JOIN, OUTER JOIN

SELF JOIN

- Join a table with itself
- Must use table aliases in FROM clause
- Must qualify each column name with the table alias
- Used to compare and combine different rows in the same table.
- Example in next slide

A self-join that returns vendors from the same city and state

```
SELECT v1.vendor_name, v2.vendor_name
FROM vendors v1 JOIN vendors v2
     ON (v1.vendor_city = v2.vendor_city) AND
        (v1.vendor_state = v2.vendor_state) AND
        (v1.vendor_id <> v2.vendor_id)
```

The result set

	VENDOR_NAME	VENDOR_NAME_1
1	Reiter's Scientific & Pro Books	National Information Data Ctr
2	Register of Copyrights	National Information Data Ctr
3	Reiter's Scientific & Pro Books	Register of Copyrights
4	National Information Data Ctr	Register of Copyrights
5	Office Depot	Jobtrak
6	Ford Motor Credit Company	Jobtrak
7	American Express	Jobtrak
8	Opamp Technical Books	Jobtrak
9	State of California	California Chamber Of Commerce
10	Franchise Tax Board	California Chamber Of Commerce
11	Pacific Bell	California Chamber Of Commerce

(1758 rows selected)

SELF JOIN vs 1-TABLE SELECT Query

- In a 1-TABLE SELECT query like below,

  SELECT column_list FROM table1 WHERE search_condition;

  - A single row is checked against the search condition in WHERE clause. If it makes the condition evaluate to true, then the row is returned by the query.
  - We cannot compare two different rows in the table source.
- In a SELF-JOIN,
  - A physical table is treated as two logically different tables.
  - RDBMS can then compare two different rows in the same physical table, but in two logically different tables.

A self-join that returns vendors from cities in common with other vendors

```
SELECT DISTINCT v1.vendor_name, v1.vendor_city,
    v1.vendor_state
FROM vendors v1 JOIN vendors v2
    ON (v1.vendor_city = v2.vendor_city) AND
       (v1.vendor_state = v2.vendor_state) AND
       (v1.vendor_id <> v2.vendor_id)
ORDER BY v1.vendor_state, v1.vendor_city
```

The result set

	VENDOR_NAME	VENDOR_CITY	VENDOR_STATE
1	AT&T	Phoenix	AZ
2	Computer Library	Phoenix	AZ
3	Wells Fargo Bank	Phoenix	AZ
4	Aztek Label	Anaheim	CA
5	Blue Shield of California	Anaheim	CA
6	ASC Signs	Fresno	CA
7	Abbey Office Furnishings	Fresno	CA
8	BFI Industries	Fresno	CA

(84 rows selected)

In-Class Practice

- Topics covered in the practice
  - SELF-JOIN, OUTER JOIN
- See details in file:
  - Wk3-MurachCh4-InClassPractice-P2.sql

OUTER JOIN

- Used to return the EXTRA DATA in one table that DON'T have a MATCH in the other table, IN ADDITION to what's returned by the corresponding INNER JOIN.
- OUTER JOIN gets
  - The data returned by the corresponding INNER JOIN:
    - i.e. ALL rows that satisfy the join condition
  - PLUS the EXTRA data in one or both tables
    - i.e. UNMATACHED rows in one or both tables
- NULL values are used for the missing data.

OUTER JOIN

- Two cases when values don’t match
  1. One value is NULL, the other value is not a NULL.
     - E.g. 10 and NULL are not a match
  2. NEITHER value is NULL. But they are not equal.
     - E.G. 10 and 20 are not a match

Types of OUTER JOIN

OUTER JOIN TYPE	EXPLICIT SYNTAX	QUERY RESULT SET
LEFT OUTER JOIN	SELECT … FROM t1 LEFT JOIN t2	Extra rows in t1 on the LEFT-side that don't have a match in t2 on the right-side, PLUS all rows returned by the INNER JOIN: t1 JOIN t2
RIGHT OUTER JOIN	SELECT … FROM t1 RIGHT JOIN t2	Extra rows in t2 on the RIGHT-side that don't have a match in t1 on the left-side, PLUS all rows returned by the INNER JOIN: t1 JOIN t2
FULL OUTER JOIN	SELECT … FROM t1 FULL JOIN t2	Extra rows in t1 on the LEFT-side that don't have a match in t2 on the right-side, PLUS Extra rows in t2 on the RIGHT-side that don't have a match in t1 on the left-side, PLUS all rows returned by the INNER JOIN: t1 JOIN t2

A SELECT statement that uses a left outer join

```
SELECT vendor_name, invoice_number, invoice_total
FROM vendors LEFT JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
ORDER BY vendor_name
```

The result set

	VENDOR_NAME	INVOICE_NUMBER	INVOICE_TOTAL
1	ASC Signs	(null)	(null)
2	AT&T	(null)	(null)
3	Abbey Office Furnishings	203339-13	17.5
4	American Booksellers Assoc	(null)	(null)
5	American Express	(null)	(null)

(202 rows selected)

The Departments table

	DEPARTMENT_NUMBER	DEPARTMENT_NAME
1	1	Accounting
2	2	Payroll
3	3	Operations
4	4	Personnel
5	5	Maintenance

The Employees table

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_NUMBER
1	Smith	Cindy	2
2	Jones	Elmer	4
3	Simonian	Ralph	2
4	Hernandez	Olivia	1
5	Aaronsen	Robert	2
6	Watson	Denise	6
7	Hardy	Thomas	5
8	O'Leary	Rhea	4
9	Locario	Paulo	6

Column department_number in Employees table

- NOT a Foreign key referencing column department_number in Departments table

A left outer join

```
SELECT department_name AS dept_name,
    d.department_number AS dept_no,
    last_name
FROM departments d
    LEFT OUTER JOIN employees e
    ON d.department_number = e.department_number
ORDER BY department_name
```

	DEPT_NAME	DEPT_NO	LAST_NAME
1	Accounting	1	Hernandez
2	Maintenance	5	Hardy
3	Operations	3	(null)
4	Payroll	2	Simonian
5	Payroll	2	Aaronsen
6	Payroll	2	Smith
7	Personnel	4	Jones
8	Personnel	4	O'Leary

Extra data from the LEFT table without a match from the right table

A right outer join

```
SELECT department_name AS dept_name,
    e.department_number AS dept_no,
    last_name
FROM departments d
    RIGHT OUTER JOIN employees e
    ON d.department_number = e.department_number
ORDER BY department_name
```

Extra data from the RIGHT table without a match from the left table

	DEPT_NAME	DEPT_NO	LAST_NAME
1	Accounting	1	Hernandez
2	Maintenance	5	Hardy
3	Payroll	2	Aaronsen
4	Payroll	2	Simonian
5	Payroll	2	Smith
6	Personnel	4	Jones
7	Personnel	4	O'Leary
8	(null)	6	Locario
9	(null)	6	Watson

A full outer join

```
SELECT department_name AS dept_name,
    d.department_number AS d_dept_no,
    e.department_number AS e_dept_no,
    last_name
FROM departments d
    FULL OUTER JOIN employees e
    ON d.department_number = e.department_number
ORDER BY department_name
```

	DEPT_NAME	D_DEPT_NO	E_DEPT_NO	LAST_NAME
1	Accounting	1	1	Hernandez
2	Maintenance	5	5	Hardy
3	Operations	3	(null)	(null)
4	Payroll	2	2	Simonian
5	Payroll	2	2	Smith
6	Payroll	2	2	Aaronsen
7	Personnel	4	4	Jones
8	Personnel	4	4	O'Leary
9	(null)	(null)	6	Locario
10	(null)	(null)	6	Watson

The Departments table

	DEPARTMENT_NUMBER	DEPARTMENT_NAME
1	1	Accounting
2	2	Payroll
3	3	Operations
4	4	Personnel
5	5	Maintenance

The Employees table

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_NUMBER
1	Smith	Cindy	2
2	Jones	Elmer	4
3	Simonian	Ralph	2
4	Hernandez	Olivia	1
5	Aaronsen	Robert	2
6	Watson	Denise	6
7	Hardy	Thomas	5
8	O'Leary	Rhea	4
9	Locario	Paulo	6

Column department_number in Employees table

- NOT a Foreign key referencing column department_number in Departments table

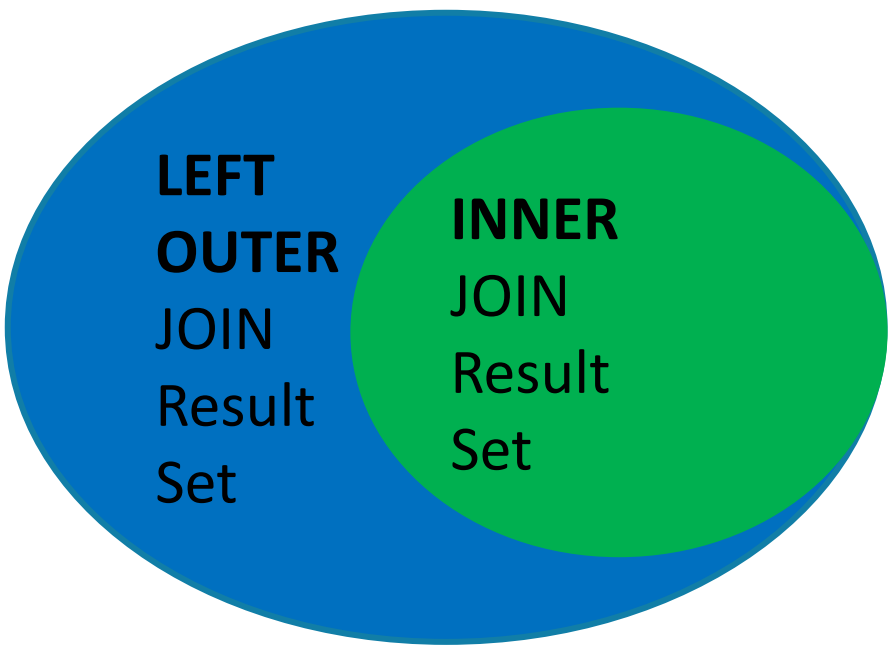
The Projects table

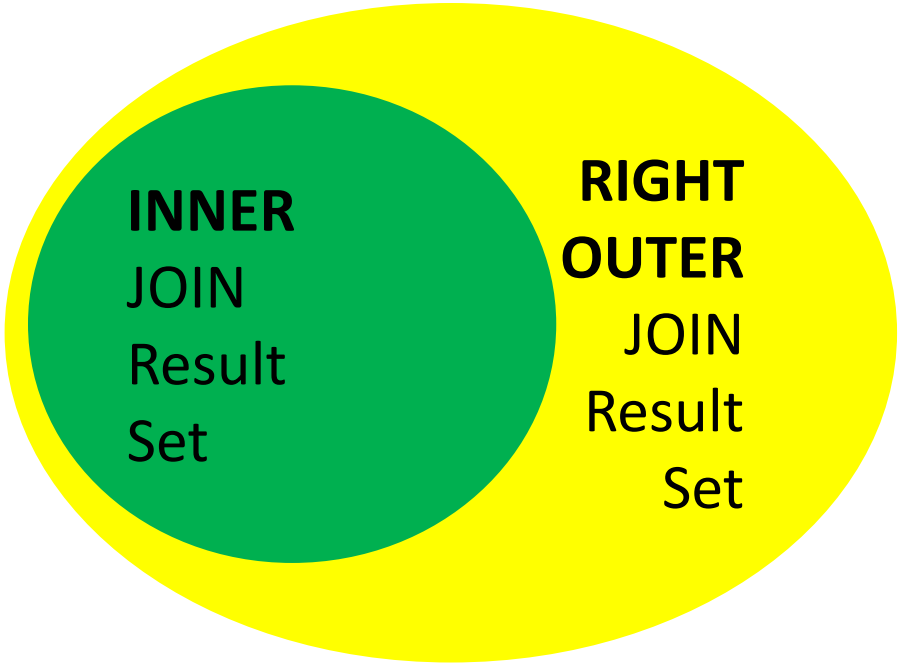
	PROJECT_NUMBER	EMPLOYEE_ID
1	P1011	8
2	P1011	4
3	P1012	3
4	P1012	1
5	P1012	5
6	P1013	6
7	P1013	9
8	P1014	10

Column employee_id in Projects table

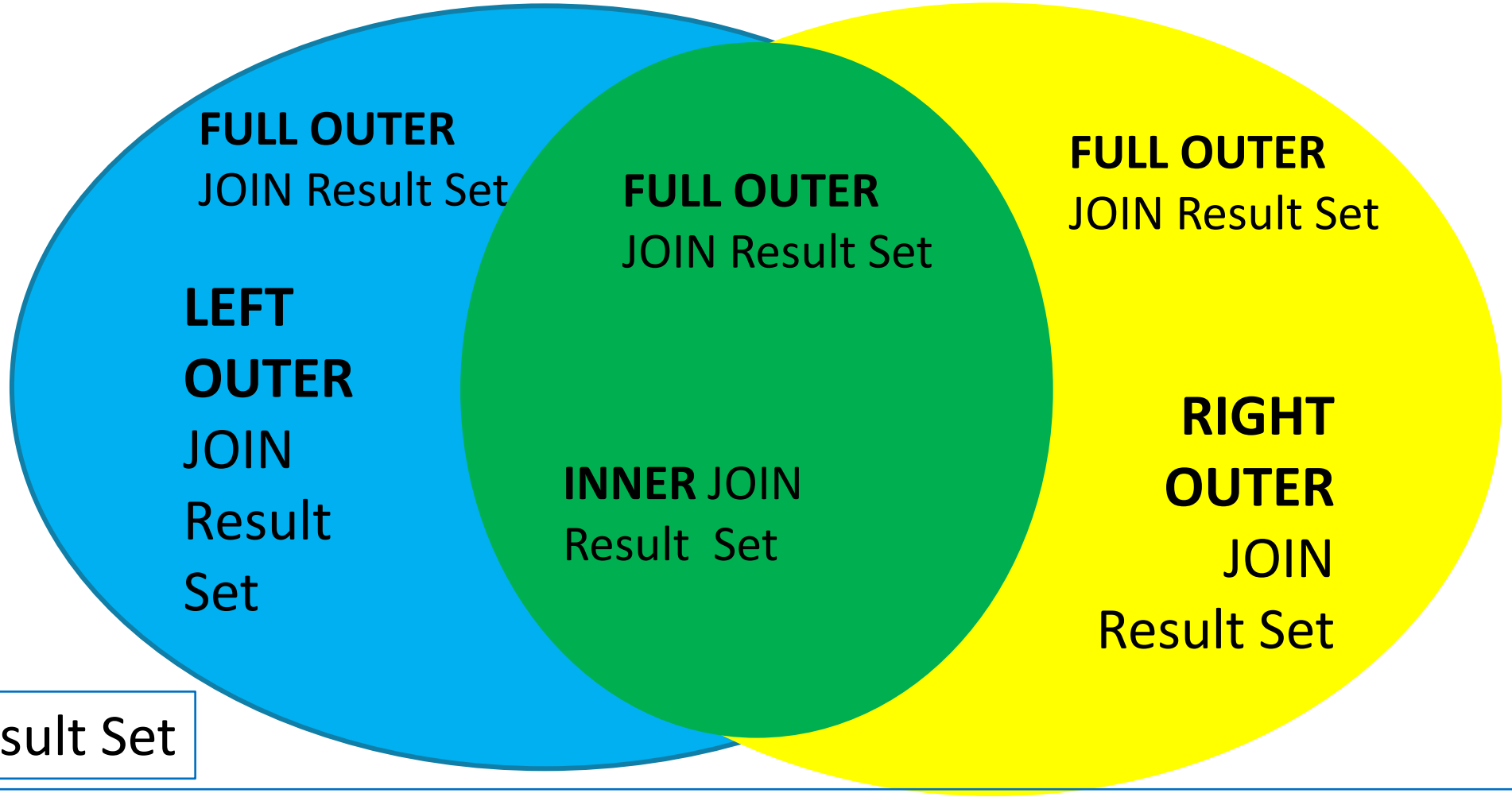
- NOT a Foreign key referencing column employee_id in Employees table?

Outer Join and Inner Join





- LEFT OUTER JOIN result set
  = INNER JOIN result set + Extra Data in LEFT table & NULLs
- RIGHT OUTER JOIN result set
  = INNER JOIN result set + NULLs & Extra Data in RIGHT table



RS: Result Set

FULL OUTER JOIN RS = INNER join RS + LEFT OUTER Join RS + RIGHT OUTER Join RS

= INNER JOIN result set + Extra Data in LEFT table & NULLs

+ NULLs & Extra Data in RIGHT table

19

In-Class Practice

- Topics covered in the practice
  - Self Join, OUTER JOIN
- See details in file:
  - Wk3-MurachCh4-InClassPractice-P2.sql