

Murach Chapter 4

How to Retrieve Data From Two or More Tables

Week 3, Lecture 5, Part 1

Knowledge Points in this lecture

- Basics about JOIN
- CROSS JOIN
- INNER JOIN
- Table Alias
- Column qualification
- Search condition and join condition
- Schema
- Ad hoc query
- Multiple JOIN conditions
- Explicit syntax of INNER JOIN
- Join 3 tables

Basics About JOIN

- Join
 - Combine columns from 2 or more tables into the same result set based on join conditions
 - Typically combine data based on primary key – foreign key relationship
- Join conditions
 - Typically use operator =
 - But can use any operator
- Explicit syntax of JOIN
 - Explicitly use keyword JOIN in the FROM clause of a SELECT statement

CROSS JOIN

- CROSS JOIN
 - Cartesian product of multiple tables listed in the FROM clause
 - FROM t1 CROSS JOIN t2
 - Combine each row in table t1 with each row in table t2
 - Generates $|t1| \times |t2|$ number of combined rows in the query result, where $|t1|$, $|t2|$ are number of rows in t1, t2
- Given $T1 = \{2, 3\}$, $T2 = \{3, 5\}$, the Cartesian product of T1 and T2:
 $T1 \times T2 = \{(2,3), (2, 5), (3, 3), (3, 5)\}$

CROSS JOIN

- Examples – Print all information about any combination of any vendor with any invoice

```
SELECT vendors.*, invoices.*  
FROM vendors CROSS JOIN invoices;
```

```
SELECT *  
FROM vendors CROSS JOIN invoices;
```

INNER JOIN

- INNER JOIN
 - Only combined rows that satisfy the join condition are included in the query result (i.e. result set)
 - INNER keyword is optional and typically omitted

A SELECT statement that joins two tables

```
SELECT invoice_number, vendor_name
FROM vendors INNER JOIN invoices
      ON vendors.vendor_id = invoices.vendor_id
ORDER BY invoice_numbe
```

The result set

	INVOICE_NUMBER	VENDOR_NAME
1	0-2058	Malloy Lithographing Inc
2	0-2060	Malloy Lithographing Inc
3	0-2436	Malloy Lithographing Inc
4	1-200-5164	Federal Express Corporation

(114 rows selected)

Table Alias, Column Qualification

Table Alias

- Alternative table name assigned to tables listed in FROM clause
- Used to shorten long names, and/or qualify column names
- Can not use the original table name once the table is assigned an alias

Column Qualification

- Use a table name or table alias and a dot operator to indicate where the column come from
- E.g. `vendors.vendor_id`, `v.vendor_id`

An inner join with aliases for all tables

```
SELECT invoice_number, vendor_name, invoice_due_date,  
       (invoice_total - payment_total - credit_total)  
       AS balance_due  
FROM vendors v JOIN invoices i  
     ON v.vendor_id = i.vendor_id  
WHERE (invoice_total - payment_total - credit_total) > 0  
ORDER BY invoice_due_date DESC
```

The result set

	INVOICE_NUMBER	VENDOR_NAME	INVOICE_DUE_DATE	BALANCE_DUE	
1	40318	Data Reproductions Corp	20-JUL-14	21842	
2	39104	Data Reproductions Corp	20-JUL-14	85.31	
3	0-2436	Malloy Lithographing Inc	17-JUL-14	10976.06	

(40 rows selected)

An inner join with an alias for only one table

```
SELECT invoice_number, line_item_amt,  
       line_item_description  
FROM invoices JOIN invoice_line_items line_items  
   ON invoices.invoice_id = line_items.invoice_id  
WHERE account_number = 540  
ORDER BY invoice_date
```

The result set

	INVOICE_NUMBER	LINE_ITEM_AMT	LINE_ITEM_DESCRIPTION
1	97/553B	313.55	Card revision
2	97/553	904.14	DB2 Card decks
3	97/522	765.13	SCMD Flyer

(8 rows selected)

Join Condition and Search Condition

- Search Condition
 - Condition that involves columns from one table
 - Typically in a WHERE clause
- Join condition
 - Condition that compares columns from two different tables
 - Typically in a FROM clause:
 - FROM table1 JOIN table2 ON (join-condition)
- See examples in next slide

A Query with Search Condition and Join Condition

```
SELECT invoice_number, line_item_amt,  
       line_item_description  
FROM invoices JOIN invoice_line_items line_items  
  ON invoices.invoice_id = line_items.invoice_id  
WHERE account_number = 540  
ORDER BY invoice_date
```

Join condition

Search condition

The result set

	INVOICE_NUMBER	LINE_ITEM_AMT	LINE_ITEM_DESCRIPTION
1	97/553B	313.55	Card revision
2	97/553	904.14	DB2 Card decks
3	97/522	765.13	SCMD Flyer

(8 rows selected)

Schema

- A DB user and the collection of DB objects owned by that user
 - schema = DB user
 - schema name = DB user name
- More technically, a set of metadata for all DB objects belonging to a DB user
- Examples -- schemas used in Murach book chapters
 - AP (Account Payable)
 - OM (Order Management)
 - EX (Examples)
- Created by DDL (Data Definition Language) statements

Schema and Database Objects

- Must log into the DB server as a DB user before accessing any DB object like a table
- Do NOT need schema name (i.e. user name) when accessing your own data
- Must use schema name (i.e. user name) for accessing objects in other schemas

The syntax of a table name
that's **qualified** with a **schema** name

`schema_name.table_name`

A SQL statement that grants the **SELECT**
permission in the **OM** schema to the **AP** schema

```
GRANT SELECT ON customers TO ap
```

A join with a table from another schema

```
SELECT vendor_name, customer_last_name,  
       customer_first_name, vendor_state AS state,  
       vendor_city AS city  
FROM vendors v  
     JOIN om.customers c  
     ON v.vendor_zip_code = c.customer_zip  
ORDER BY state, city
```

Join ap's vendors to customers in om user
(Run it as user ap)

The result set

	VENDOR_NAME	CUSTOMER_LAST_NAME	CUSTOMER_FIRST_NAME	STATE	CITY
1	Wells Fargo Bank	Marissa	Kyle	AZ	Phoenix
2	Aztek Label	Irvin	Ania	CA	Anaheim
3	Lou Gentile's Flower Basket	Damien	Deborah	CA	Fresno
4	Shields Design	Damien	Deborah	CA	Fresno
5	Costco	Neftaly	Thalia	CA	Fresno
6	Costco	Holbrooke	Rashad	CA	Fresno
7	Gary McKeighan Insurance	Holbrooke	Rashad	CA	Fresno
8	Zylka Design	Neftaly	Thalia	CA	Fresno
9	Zylka Design	Holbrooke	Rashad	CA	Fresno

(37 rows)

Ad Hoc Query

- Join data from multiple tables based on relationships that are not stored in the database
- Do not join data from multiple tables based on primary key and foreign key relationships.
- Example in the previous slide is an Ad Hoc query

An inner join with two conditions

```
SELECT invoice_number, invoice_date,  
       invoice_total, line_item_amt  
FROM invoices i JOIN invoice_line_items li  
  ON (i.invoice_id = li.invoice_id) AND  
     (i.invoice_total > li.line_item_amt)  
ORDER BY invoice_number
```

The result set

	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL	LINE_ITEM_AMT
1	97/522	30-APR-14	1962.13	765.13
2	97/522	30-APR-14	1962.13	1197
3	I77271-001	05-JUN-14	662	75.6
4	I77271-001	05-JUN-14	662	58.4

(6 rows selected)

The explicit syntax for an inner join

```
SELECT select_list
FROM table_1
    [INNER] JOIN table_2
        ON join_condition_1
    [[INNER] JOIN table_3
        ON join_condition_2] ...
```

- []: Optional
 - You don't have to use INNER keyword in a JOIN.
 - Optionally, an inner join can include 3 or more tables in FROM clause.
- Explicit syntax
 - Explicitly use the keyword JOIN or INNER JOIN

The explicit syntax for an inner join that uses table aliases

```
SELECT select_list
FROM table_1 n1
    [INNER] JOIN table_2 n2
        ON n1.column_name operator n2.column_name
    [[INNER] JOIN table_3 n3
        ON n2.column_name operator n3.column_name] ...
```

- []: Optional
 - You don't have to use INNER keyword in a JOIN.
 - Optionally, an inner join can include 3 or more tables in FROM clause.
- Explicit syntax
 - Explicitly use the keyword JOIN or INNER JOIN

A SELECT statement that joins **three** tables

```
SELECT vendor_name, i.invoice_id, invoice_number,  
       invoice_date, invoice_total, line_item_amt  
FROM vendors v  
     JOIN invoices i ON (v.vendor_id = i.vendor_id)  
     JOIN invoice_line_items li  
         ON (i.invoice_id = li.invoice_id) AND  
           (i.invoice_total > li.line_item_amt)  
ORDER BY invoice_number
```

The result set

	INVOICE_ID	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL	LINE_ITEM_AMT	VENDOR_NAME
1	19	97/522	30-APR-14	1962.13	1197	Zylka Design
2	19	97/522	30-APR-14	1962.13	765.13	Zylka Design
3	100	I77271-001	05-JUN-14	662	478	Wells Fargo Bank
4	100	I77271-001	05-JUN-14	662	75.6	Wells Fargo Bank
5	100	I77271-001	05-JUN-14	662	50	Wells Fargo Bank
6	100	I77271-001	05-JUN-14	662	58.4	Wells Fargo Bank

(6 rows selected)