Database Programming

Join Clauses

Objectives

This lesson covers the following objectives:

- Construct and execute a join with the ANSI-99 USING and ON clauses
- Construct and execute an ANSI-99 query that joins three tables

Purpose

As you add more commands to your database vocabulary, you will be better able to design queries that return the desired result. The purpose of a join is to bind data together, across tables, without repeating all of the data in every table. Why ask for more data than you really need?

USING Clause

In a natural join, if the tables have columns with the same names but different data types, the join causes an error.

To avoid this situation, the join clause can be modified with a USING clause.

The USING clause specifies the columns that should be used for the equijoin.

USING Clause (cont.)

The query shown is an example of the USING clause. The columns referenced in the USING clause should not have a qualifier (table name or alias) anywhere in the SQL statement.

```
SELECT client_number, first_name, last_name, event_date
FROM d_clients JOIN d_events
USING (client_number);
```

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	EVENT_DATE
5922	Hiram	Peters	14-MAY-2004
6133	Lauren	Vigil	28-APR-2004

USING Clause (cont.)

The USING clause allows us to use WHERE to restrict rows from one or both tables:

```
SELECT client_number, first_name, last_name, event_date
FROM d_clients JOIN d_events
USING (client_number)
WHERE last_name = 'Peters';
```

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	EVENT_DATE
5922	Hiram	Peters	14-MAY-2004

ON Clause

What if the columns to be joined have different names, or if the join uses non-equality comparison operators such as <, >, or BETWEEN ?

We can’t use USING, so instead we use an ON clause. This allows a greater variety of join conditions to be specified.

The ON clause also allows us to use WHERE to restrict rows from one or both tables.

ON Clause Example

In this example, the ON clause is used in a self-join where the same table is given two different references. In the employees table, some employees are also managers. The self-join is used to select those employees who are also managers.

```
SELECT e.last_name as "EMP", m.last_name as "MGR"
FROM employees e JOIN employees m
ON (e.manager_id = m.employee_id);
```

EMP	MGR
Hartstein	King
Zlotkey	King
Mourgos	King
De Haan	King
Kochhar	King
Higgins	Kochhar
……	……

ON Clause with WHERE Clause

Here is the same query with a WHERE clause to restrict the rows selected.

```
SELECT e.last_name as "EMP", m.last_name as "MGR"
FROM employees e JOIN employees m
ON (e.manager_id = m.employee_id)
WHERE e.last_name like 'H%';
```

EMP	MGR
Hartstein	King
Higgins	Kochhar
Hunold	De Haan

Joining Three Tables

Both USING and ON can be used to join three or more tables.

Suppose we need a report of our clients, their events, and the themes for those events ? We need to join three tables: d_clients, d_events, and d_themes.

Joining Three Tables Example

```
SELECT last_name, event_date, t.description
FROM d_clients c JOIN d_events e
USING (client_number)
JOIN d_themes t
ON (e.theme_code = t.code);
```

LAST_NAME	EVENT_DATE	DESCRIPTION
Peters	14-MAY-2004	Tropical
Vigil	28-APR-2004	Tropical

Join Comparison

Comparing Oracle Proprietary Joins with ANSI/ISO SQL: 1999 Joins

Oracle Proprietary Join	ANSI/ISO SQL: 1999 Equivalent
Cartesian Product	Cross Join
Equijoin	Natural Join (if the join columns have the same name and data type) USING clause (if the columns have the same name but different data types) ON clause (if the columns have different names)
Non-equijoin	ON clause

Terminology

Key terms used in this lesson included:

- ON clause
- USING clause

Summary

In this lesson, you should have learned how to:

- Construct and execute a join with the ANSI-99 USING and ON clauses
- Construct and execute an ANSI-99 query that joins three tables