



# Database Programming with SQL

6-2

Join Clauses



# Objectives

This lesson covers the following objectives:

- Construct and execute a join with the ANSI-99 USING Clause
- Construct and execute a join with the ANSI-99 ON Clause
- 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 join.

# USING Clause

- 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 first_name, last_name, department_id, department_name
FROM employees JOIN departments USING (department_id);
```

FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Jennifer	Whalen	10	Administration
Michael	Hartstein	20	Marketing
Pat	Fay	20	Marketing
...	...	...	...

# USING Clause

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

```
SELECT first_name, last_name, department_id, department_name
FROM employees JOIN departments USING (department_id)
WHERE last_name = 'Higgins';
```

FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Shelley	Higgins	110	Accounting

# Aliases

- Working with lengthy column and table names can be cumbersome.
- Fortunately, there is a way to shorten the syntax using aliases.
- To distinguish columns that have identical names but reside in different tables, use table aliases.
- A table alias is similar to a column alias; it renames an object within a statement.
- It is created by entering the new name for the table just after the table name in the from-clause.



# Table Aliases

- Table aliases are used in the query below.

```
SELECT last_name, e.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
      AND department_id = 80;
```

LAST_NAME	JOB_ID	JOB_TITLE
Zlotkey	SA_MAN	Sales Manager
Abel	SA_REP	Sales Representative
Taylor	SA_REP	Sales Representative

- When column names are not duplicated between two tables, you do not need to add the table name or alias to the column name.

# Table Aliases

- If a table alias is used in the FROM clause, then that table alias must be substituted for the table name throughout the SELECT statement.
- Using the name of a table in the SELECT clause that has been given an alias in the FROM clause will result in an error.

```
SELECT last_name, employees.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
      AND department_id = 80;
```



ORA-00904: "EMPLOYEES"."JOB\_ID": invalid identifier

# 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 to join the employees table with the jobs table.

```
SELECT last_name, job_title
FROM employees e JOIN jobs j
  ON (e.job_id = j.job_id);
```

- A join ON clause is required when the common columns have different names in the two tables.

LAST_NAME	JOB_TITLE
King	President
Kochhar	Administration Vice President
De Haan	Administration Vice President
Whalen	Administration Assistant
Higgins	Accounting Manager
Gietz	Public Accountant
Zlotkey	Sales Manager
Abel	Sales Representative
Taylor	Sales Representative
...	

# ON Clause Example

- When using an ON clause on columns with the same name in both tables, you need to add a qualifier (either the table name or alias) otherwise an error will be returned. The example above uses table aliases as a qualifier `e.job_id = j.job_id`, but could also have been written using the table names (`employees.job_id = jobs.job_id`).

```
SELECT last_name, job_title
FROM employees e JOIN jobs j
  ON (e.job_id = j.job_id);
```

LAST_NAME	JOB_TITLE
King	President
Kochhar	Administration Vice President
De Haan	Administration Vice President
Whalen	Administration Assistant
Higgins	Accounting Manager
Gietz	Public Accountant
Zlotkey	Sales Manager
Abel	Sales Representative
Taylor	Sales Representative
...	

# ON Clause with WHERE Clause

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

```
SELECT last_name, job_title  
FROM employees e JOIN jobs j  
  ON (e.job_id = j.job_id)  
WHERE last_name LIKE 'H%';
```

LAST_NAME	JOB_TITLE
Higgins	Accounting Manager
Hunold	Programmer
Hartstein	Marketing Manager

# ON Clause with non-equality operator

- Sometimes you may need to retrieve data from a table that has no corresponding column in another table.
- Suppose we want to know the `grade_level` for each employees salary.
- The `job_grades` table does not have a common column with the `employees` table.
- Using an ON clause allows us to join the two tables

**job\_grades table**

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

# ON Clause with non-equality operator

```
SELECT last_name, salary, grade_level, lowest_sal, highest_sal
FROM employees JOIN job_grades
    ON(salary BETWEEN lowest_sal AND highest_sal);
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Vargas	2500	A	1000	2999
Matos	2600	A	1000	2999
Davies	3100	B	3000	5999
Rajs	3500	B	3000	5999
Lorentz	4200	B	3000	5999
Whalen	4400	B	3000	5999
Mourgos	5800	B	3000	5999
Fay	6000	C	6000	9999
...				



# Joining Three Tables

- Both USING and ON can be used to join three or more tables.
- Suppose we need a report of our employees, their department, and the city where the department is located?
- We need to join three tables: employees, departments and locations.



# Joining Three Tables Example

```
SELECT last_name, department_name AS "Department", city
FROM employees JOIN departments USING (department_id)
      JOIN locations USING (location_id);
```



LAST_NAME	Department	CITY
Hartstein	Marketing	Toronto
Fay	Marketing	Toronto
Zlotkey	Sales	Oxford
Abel	Sales	Oxford
Taylor	Sales	Oxford
Hunold	IT	Southlake
Ernst	IT	Southlake
Lorentz	IT	Southlake
Mourgos	Shipping	South San Francisco
...		

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

