Database Programming with SQL
7-1: Oracle Equijoin and Cartesian Product

Objectives

- Name the Oracle proprietary joins and their ANSI/ISO SQL: 1999 counterparts
- Describe the purpose of join conditions
- Construct and execute a SELECT statement that results in a Cartesian product
- Construct and execute SELECT statements to access data from more than one table using an equijoin
- Construct and execute SELECT statements that add search conditions using the AND operator
- Apply the rule for using column aliases in a join statement

Vocabulary

Identify the vocabulary word for each definition below.

- Results from an invalid or omitted join condition; all combinations of rows are displayed Cartesian Product
- Values in a column in one table are equal to a value in another table; also called an inner join or simple join Equi-Join
- Connection command exclusive to a specific company Proprietary Join
- Gives a table another name to simplify queries and improve performance Alias
- Display data from two or more related tables Join

Try It / Solve It

1. Create a Cartesian product that displays the columns in the d_play_list_items and the d_track_listings in the DJs on Demand database.

```
SELECT *
FROM d_play_list_items, d_track_listings;
```

2. Correct the Cartesian product produced in question 1 by creating an equijoin using a common Column.

```
SELECT d_play_list_items.*, d_track_listings.*
FROM d_play_list_items
JOIN d_track_listings
ON d_play_list_items.track_id = d_track_listings.track_id;
```

3. Write a query to display the title, type, description, and artist from the DJs on Demand database.

```
SELECT title, type, description, artist
```

```
FROM d_track_listings;
```

4. Rewrite the query in question 3 to select only those titles with an ID of 47 or 48.

```
SELECT title, type, description, artist
FROM d_track_listings
WHERE id IN (47, 48);
```

5. Write a query that extracts information from three tables in the DJs on Demand database, the d_clients table, the d_events table, and the d_job_assignments table.

```
SELECT d_clients.client_name, d_events.event_name, d_events.event_date,
d_job_assignments.job_role
FROM d_clients
JOIN d_job_assignments
ON d_clients.client_id = d_job_assignments.client_id
JOIN d_events
ON d_job_assignments.event_id = d_events.event_id;
```

6. Create and execute an equijoin between DJs on Demand tables d_track_listings and d_cds. Return the song_id and the title only.

```
SELECT d_track_listings.song_id, d_track_listings.title
FROM d_track_listings
JOIN d_cds
ON d_track_listings.cd_id = d_cds.cd_id;
```

7. Mark T for the statements that are true and F for the statements that are false.

__F__ a. A join is a type of query that gets data from more than one table based on columns with the same name.

__T__ b. To join tables using an equijoin, there must be a common column in both tables and that column is usually a primary key in one of the tables.

__T__ c. A Cartesian product occurs because the query does not specify a WHERE clause.

__F__ d. Table aliases are required to create a join condition.

__T__ e. If a table alias is used for a table name in the FROM clause, it must be substituted for the table name throughout the SELECT statement.

__F__ f. Table alias must be only one character in length.

__T__ g. A simple join or inner join is the same as an equijoin.

8. What advantage does being able to combine data from multiple tables have for a business?

- Combining data from many sources allow for better views of operations, customer behaviors, and sales trend
- Can increase data accuracy. For example, by joining tables businesses make sure the data is consistent across other datasets and make sure data does not repeat

- Can improve reporting by combining data which allows businesses to make detailed reports and see different parts of their operations like sales and performances
- Can increase operational efficiency by integrating data from different departments within the business
- Can make strategic planning with access to combined data
- Can analyze trends by identifying patterns with the help of combining data

---

_

Database Programming with SQL
7-2: Oracle Nonequijoins and Outer Joins

Objectives

- Construct and execute a SELECT statement to access data from more than one table using a nonequijoin
- Create and execute a SELECT statement to access data from more than one table using an Oracle outer join

Try It / Solve It

1. Create a join based on the cost of the event between the DJs on Demand tables D_EVENTS and D_PACKAGES. Show the name of the event and the code for each event.

SELECT event_name, event_code, package_code
FROM d_events
JOIN d_packages
ON cost = cost;

2. Using the Oracle database, create a query that returns the employee last name, salary, and job-grade level based on the salary. Select the salary between the lowest and highest salaries.

SELECT last_name, salary, grade_level
FROM employees
JOIN job_grades
ON salary BETWEEN low_salary AND high_salary;

3. What condition requires the creation of a nonequijoin?

- Required when relationships exists between two tables
- When tables can not be established relationships through simple equality

4. Rewrite the following nonequijoin statement using the logical condition operators (AND, OR, NOT): WHERE a.ranking BETWEEN g.lowest_rank AND g.highest_rank

WHERE a.ranking >= g.lowest_rank AND a.ranking <= g.highest_rank

5. How do you know when to use a table alias and when not to use a table alias?

When working with multiple tables, handling long/complex tables, improving readability and maintainability

6. What kind of join would you use if you wanted to find data between a range of numbers?

Nonequijoins such as BETWEEN, < , >, <=, >=

7. You need to produce a report for Global Fast Foods showing customers and orders. A customer must be included on the report even if the customer has had no orders.

```
SELECT customer_id, customer_name, order_id, order_date, total_amount
FROM customers
LEFT OUTER JOIN orders ON customer_id = customer_id
ORDER BY customer_name;
```

8. Create a query of the Oracle database that shows employee last names, department IDs, and department names. Include all employees even if they are not assigned to a department.

```
SELECT last_name, department_id, department_name
FROM employees
LEFT OUTER JOIN departments
ON department_id = department_id
ORDER BY last_name;
```

9. Modify the query in problem 8 to return all the department IDs even if no employees are assigned to them.

```
SELECT last_name, department_id, department_name
FROM employees
RIGHT OUTER JOIN departments on department_id = department_id
ORDER BY department_id;
```

10. There are one or more errors in each of the following statements. Describe the errors and correct them.

a. WHERE e.department_id(+) = d.department_id (+);

```
WHERE e.department_id(+) = d.department_id;
```

b. SELECT e.employee id, e. last name, d. location id
FROM employees, departments
WHERE e.department_id = d.department_id(+);

```
SELECT e.employee_id, e.last_name, d.location_id
FROM employees e, departments d
WHERE e.department_id = d.department_id(+);
```

11. Create a query that will show all CD titles and song IDs in the DJs on Demand database even if there is no CD number in the track-listings table.

```
SELECT cd_id, title, song_id
FROM cds
LEFT OUTER JOIN Track_Listings  ON cd_id = cd_id
ORDER BY cd_id;
```

12. How many times has someone asked you: “What do you want to be when you grow up?” For most of us, the first thing that comes to mind is something like business manager, engineer, teacher, game designer, doctor, scientist, computer programmer, or accountant -- all pretty much traditional career choices. Have you ever thought about working in an odd job or nontraditional career?
There are people who are professional shoppers for busy executives, directors of zoos, recipe designers, insecticide chemists, golf-course designers, and turf managers. Picture yourself in a dream job or nontraditional career doing something that you think would be interesting, life fulfilling, and profitable.

Database Programming with SQL
8-1: Group Functions Practice Activities

Objectives

- Define and give an example of the seven group functions: SUM, AVG, COUNT, MIN, MAX, STDDEV, VARIANCE
- Construct and execute a SQL query using group functions
- Construct and execute group functions that operate only with numeric data types

Vocabulary

Identify the vocabulary word for each definition below.

- Calculates average value excluding nulls AVG( )
- Returns the number of rows with non-null values for the expression COUNT ( )
- For two sets of data with approximately the same mean, the greater the spread, the greater the standard deviation. (STDDEV( ))
- Operate on sets of rows to give one result per group Aggregate functions
- Returns minimum value ignoring nulls MIN ( )
- Used with columns that store numeric data to calculate the spread of data around the mean (VARIANCE( ))
- Calculates the sum ignoring null values SUM ( )
- Returns the maximum value ignoring nulls MAX ( )
- To gather into a sum or wholeAggregate

Try It / Solve It

1. Define and give an example of the seven group functions: AVG, COUNT, MAX, MIN, STDDEV, SUM, and VARIANCE.

AVG(): Calculates the average value of a column, excluding null values.
Example: SELECT AVG(salary) FROM employees;

COUNT(): Returns the number of rows with non-null values for an expression.
Example: SELECT COUNT(employee_id) FROM employees WHERE department_id = 50;

MAX(): Returns the maximum value from a column, ignoring null values.
Example: SELECT MAX(salary) FROM employees;

MIN(): Returns the minimum value from a column, ignoring null values.
Example: SELECT MIN(salary) FROM employees;

STDDEV(): Calculates the standard deviation, measuring the spread of data around the mean.
Example: SELECT STDDEV(salary) FROM employees;

SUM(): Calculates the sum of values in a column, excluding null values.
Example: SELECT SUM(salary) FROM employees WHERE department_id = 50;

VARIANCE(): Measures the variance, which represents the spread of data points from the mean.
Example: SELECT VARIANCE(salary) FROM employees;

2. Create a query that will show the average cost of the DJs on Demand events. Round to two decimal places.

```
SELECT ROUND(AVG(event_cost), 2) AS avg_cost
FROM dj_on_demand_events;
```

3. Find the average salary for Global Fast Foods staff members whose manager ID is 19.

```
SELECT AVG(salary) AS avg_salary FROM global_fast_foods_employees WHERE
manager_id = 19;
```

4. Find the sum of the salaries for Global Fast Foods staff members whose IDs are 12 and 9.

```
SELECT SUM(salary) AS total_salary FROM global_fast_foods_employees WHERE
employee_id IN (12, 9);
```

5. Using the Oracle database, select the lowest salary, the most recent hire date, the last name of the person who is at the top of an alphabetical list of employees, and the last name of the person who is at the bottom of an alphabetical list of employees. Select only employees who are in departments 50 or 60.

```
SELECT MIN(salary) "lowest salary", MAX(hire_date) "most recent hire date", MIN(last_name)
"top last name", MAX(last_name) "bottom last name" FROM employees WHERE
department_id in (50, 60);
```

6. Your new Internet business has had a good year financially. You have had 1,289 orders this year. Your customer order table has a column named total_sales. If you submit the following query, how many rows will be returned?
   SELECT sum(total_sales)
   FROM orders;

The query will return 1 row because SUM() is an aggregate function that returns a single value.

7. You were asked to create a report of the average salaries for all employees in each division of the company. Some employees in your company are paid hourly instead of by salary. When you ran the report, it seemed as though the averages were not what you expected—they were much higher than you thought! What could have been the cause?

The cause could be that hourly wages were included with annual salaries in the same column without adjusting for hourly wages to annual equivalents.

8. Employees of Global Fast Foods have birth dates of July 1, 1980, March 19, 1979, and March 30, 1969. If you select MIN(birthdate), which date will be returned?

March 30, 1969 because it is the earliest date.

9. Create a query that will return the average order total for all Global Fast Foods orders from January 1, 2002, to December 21, 2002.

SELECT AVG(order_total) AS avg_order_total
FROM global_fast_foods_orders
WHERE order_date
BETWEEN '2002-01-01' AND '2002-12-21';

10. What was the hire date of the last Oracle employee hired?

SELECT MAX(hire_date) AS last_hire_date FROM oracle_employees;

11. In the following SELECT clause, which value returned by the SELECT statement will be larger?
    SELECT SUM(operating_cost), AVG(operating_cost)
    SELECT SUM(operating_cost)

12. Refer to the DJs on Demand database D_EVENTS table:
Which code is valid as part of an SQL query?
___F____a. FROM event_date
____T___b. SELECT SUM(cost)
___F____c. SELECT SUM(event_date)

____T___d. SELECT AVG(cost) AS "Expense"
____F___e. WHERE MIN(id) = 100
____F___f. SELECT MAX(AVG(cost))
____T___g. SELECT MIN(event_date)

____

_

Database Programming with SQL
8-2: Count, Distinct, NVL Practice Activities

Objectives

- Construct and execute a SQL query using the COUNT group function
- Use DISTINCT and the NVL function with group functions

Vocabulary

Identify the vocabulary word for each definition below.

- Returns the number of non-null values in the expression column COUNT
- The keyword used to return only non-duplicate values or combinations of non-duplicate values in a query. DISTINCT
- Returns the number of unique non-null values in the expression column. COUNT DISTINCT

Try It / Solve It

1. How many songs are listed in the DJs on Demand D_SONGS table?
   SELECT COUNT(*) AS total_songs FROM d_songs;
2. In how many different location types has DJs on Demand had venues?
   SELECT COUNT(DISTINCT location_type) AS distinct_location_types FROM d_venues;
3. The d_track_listings table in the DJs on Demand database has a song_id column and a cd_number column. How many song IDs are in the table and how many different CD numbers are in the table?
   SELECT COUNT(song_id) AS total_song_ids,
   COUNT(DISTINCT cd_number) AS distinct_cd_numbers
   FROM d_track_listings;
4. How many of the DJs on Demand customers have email addresses?
   SELECT COUNT(*) AS customers_with_email
   FROM d_customers
   WHERE email IS NOT NULL;
5. Some of the partners in DJs on Demand do not have authorized expense amounts

(auth_expense_amt). How many partners do have this privilege?

```
SELECT COUNT(*) AS partners_with_auth_expense
FROM d_partners
WHERE auth_expense_amt IS NOT NULL;
```

6. What values will be returned when the statement below is issued?

```
SELECT COUNT(shoe_color), COUNT(DISTINCT shoe_color)
FROM shoes;
```

   COUNT(shoe_color) will return the number of non-null values in the shoe_color column. COUNT(DISTINCT shoe_color) will return the number of distinct (unique) non-null values in the shoe_color column.

7. Create a query that will convert any null values in the auth_expense_amt column on the DJs on Demand D_PARTNERS table to 100000 and find the average of the values in this column. Round the result to two decimal places.

```
SELECT ROUND(AVG(NVL(auth_expense_amt, 100000)), 2) AS avg_expense_amt
FROM d_partners;
```

8. Which statement(s) is/are True about the following SQL statement:

```
SELECT AVG(NVL(selling_bonus, 0.10))
FROM bonuses;
```

__F__ a. The datatypes of the values in the NVL clause can be any datatype except date data.

__T___ b. If the selling_bonus column has a null value, 0.10 will be substituted.

__T___ c. There will be no null values in the selling_bonus column when the average is calculated.

___F__ d. This statement will cause an error. There cannot be two functions in the SELECT Statement.

9. Which of the following statements is/are TRUE about the following query?

```
SELECT DISTINCT colors, sizes
FROM items;
```

___F__ a. Each color will appear only once in the result set.

__F___ b. Each size will appear only once in the result set.

___T__ c. Unique combinations of color and size will appear only once in the result set.

__F___ d. Each color and size combination will appear more than once in the result set.