**Database Programming with SQL  
7-1: Oracle Equijoin and Cartesian Product  
Practice Activities**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  
  Equijoin
* 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 table

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 \*

FROM d\_play\_list\_items p

JOIN d\_track\_listings t ON p.track\_id = t.track\_id;

3. Write a query to display the title, type, description, and artist from the DJs on Demand database.  
SELECT p.title, p.type, p.description, t.artist FROM d\_play\_list\_items p

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

SELECT p.title, p.type, p.description, t.artist

FROM d\_play\_list\_items p

WHERE p.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 c.client\_name, c.contact\_info, e.event\_name, e.event\_date, j.job\_role, j.assignment\_date

FROM d\_clients c

JOIN d\_events e ON c.client\_id = e.client\_id

JOIN d\_job\_assignments j ON e.event\_id = j.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 t.song\_id, t.title

FROM d\_track\_listings t

JOIN d\_cds c ON t.cd\_id = c.cd\_id;

7. Mark T for the statements that are true and F for the statements that are false.  
\_\_\_t\_ 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?

The advantage includes being able to comprehensively review data and view particular trends in data.

**Database Programming with SQL  
7-2: Oracle Nonequijoins and Outer Joins  
Practice Activities**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 e.event\_name, p.package\_code

FROM D\_EVENTS e

JOIN D\_PACKAGES p ON e.cost = p.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 employees.last\_name, employees.salary, job\_grades.grade\_level

FROM employees

JOIN job\_grades

ON employees.salary BETWEEN job\_grades.low\_sal AND job\_grades.high\_sal;

3. What condition requires the creation of a nonequijoin?  
joining of tables that are not equal.

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?  
SELECT e.last\_name FROM employees e;

Yes use table alias-when joining multiple tables

Don’t use table alias-when only one table is present

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

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 c.customer\_id, c.customer\_name, o.order\_id, o.order\_date, o.order\_amount

FROM Customers c

LEFT JOIN Orders o

ON c.customer\_id = o.customer\_id

ORDER BY c.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 e.last\_name, e.department\_id, d.department\_name

FROM employees e

LEFT JOIN departments d

ON e.department\_id = d.department\_id;

9. Modify the query in problem 8 to return all the department IDs even if no employees are assigned to them.  
SELECT e.last\_name, e.department\_id, d.department\_name

FROM departments d

LEFT JOIN employees e

ON e.department\_id = d.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 (+);  
b. SELECT e.employee id, e. last name, d. location id  
FROM employees, departments  
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 c.title AS cd\_title, t.song\_id

FROM d\_cds c

LEFT JOIN d\_track\_listings t

**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  
  group 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 whole  
  aggregate

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

SELECT AVG(salary) AS average\_salary

FROM employees;

SELECT COUNT(employee\_id) AS employee\_count

FROM employees;

SELECT MAX(salary) AS highest\_salary

FROM employees;

SELECT MIN(salary) AS lowest\_salary

FROM employees;

SELECT SUM(salary) AS total\_salary

FROM employees;

SELECT VARIANCE(salary) AS salary\_variance

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(cost), 2) AS average\_cost

FROM d\_events;

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

SELECT AVG(salary) AS average\_salary

FROM 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 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) AS lowest\_salary, MAX(hire\_date) AS most\_recent\_hire, (SELECT last\_name FROM employees WHERE department\_id IN (50, 60) ORDER BY last\_name ASC FETCH FIRST 1 ROWS ONLY) AS first\_employee, (SELECT last\_name FROM employees WHERE department\_id IN (50, 60) ORDER BY last\_name DESC FETCH FIRST 1 ROWS ONLY) AS last\_employee

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;

one

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?

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?

The oldest birth date will be returned; March 30, 1969

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(total\_sales) AS avg\_order\_total

FROM orders

WHERE order\_date BETWEEN TO\_DATE('2002-01-01', 'YYYY-MM-DD') AND TO\_DATE('2002-12-21', 'YYYY-MM-DD');

10. What was the hire date of the last Oracle employee hired?  
SELECT MAX(hire\_date) AS last\_hire\_date

FROM employees;

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

The value returned by the sum will be larger.

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(\*)

FROM D\_SONGS;

2. In how many different location types has DJs on Demand had venues?

SELECT COUNT(DISTINCT location\_type)

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 unique\_cd\_numbers

FROM d\_track\_listings;

4. How many of the DJs on Demand customers have email addresses?  
SELECT COUNT(email) AS customers\_with\_email

FROM d\_clients

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(\*)

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;  
4 & 2

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)

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.