**Programming 7&8**

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

Vocabulary  
Identify the vocabulary word for each definition below.

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

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, d\_track\_listings t

WHERE p.some\_common\_column = t.some\_common\_column;

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

SELECT title, type, description, artist

FROM djs\_on\_demand;

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 djs\_on\_demand

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 c.client\_name, e.event\_name, j.job\_title

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, c.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.  
\_\_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 multiple tables allows businesses to get a comprehensive view of related data, improving reporting, decision-making, and operational efficiency.

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

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, e.event\_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 e.last\_name, e.salary, g.grade\_level

FROM employees e

JOIN job\_grades g ON e.salary BETWEEN g.lowest\_salary AND g.highest\_salary;

3. What condition requires the creation of a nonequijoin?

A nonequijoin is required when the join condition involves inequalities (<, >, !=, etc.) rather than equality (=). This often happens when ranges of values (e.g., salary ranges or date ranges) are compared.

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?

* You use a table alias when:

1. You are joining multiple tables with similar column names.
2. You want to make your SQL query easier to read and more concise.

* Avoid using aliases when they don't improve clarity or when there's no risk of column name ambiguity.

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

A nonequijoin is used when you want to match data based on a range of values. This often involves BETWEEN or <, > conditions.

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\_name, o.order\_id

FROM customers c

LEFT JOIN orders o ON c.customer\_id = o.customer\_id;

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, d.department\_id, d.department\_name

FROM employees e

RIGHT JOIN departments d 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(+);

* 1. **Error:** The + operator is used incorrectly. It should only appear on one side of the join condition.

WHERE e.department\_id(+) = d.department\_id;

B **Error:** department\_id(+) is incorrectly placed on the right side, and there are spacing issues with the column names.

SELECT e.employee\_id, e.last\_name, d.location\_id

FROM employees e

LEFT JOIN departments d ON 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.cd\_title, tl.song\_id

FROM cds cd

LEFT JOIN track\_listings tl ON cd.cd\_number = tl.cd\_number;

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. Use Internet resources to explore your idea. Write a brief description of the job to share with the class.

As someone who loves animals and being in nature, I think being a wildlife biologist would be an amazing career. A wildlife biologist studies animals and their habitats, watching how they behave and interact in the wild. They often work outdoors, sometimes in remote areas, to collect data on different species, which helps us understand animal populations better and create strategies to protect them. Wildlife biologists can work for government agencies, research centers, or conservation organizations, all aiming to protect endangered species and their ecosystems. I find this career really fulfilling because it’s not only exciting but also meaningful. It allows me to contribute to the protection of wildlife and the environment, which is something I care deeply about.

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

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **AVG** | Calculates average value excluding nulls |
| **Count** | Returns the number of rows with non-null values for the expression |
| **STDDEV** | For two sets of data with approximately the same mean, the greater the spread, the greater the standard deviation. |
| **Group Functions** | Operate on sets of rows to give one result per group |
| **Min** | Returns minimum value ignoring nulls |
| **Variance** | Used with columns that store numeric data to calculate the spread of data around the mean |
| **Sum** | Calculates the sum ignoring null values |
| **Max** | Returns the maximum value ignoring nulls |
| **Aggregate** | To gather into a sum or whole |

Try It / Solve It

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

* AVG Example: SELECT AVG(salary) FROM employees;
* COUNT Example: SELECT COUNT(employee\_id) FROM employees WHERE department\_id = 50;
* MAX Example: SELECT MAX(salary) FROM employees;
* MIN Example: SELECT MIN(salary) FROM employees;
* STDDEV Example: SELECT STDDEV(salary) FROM employees;
* SUM Example: SELECT SUM(salary) FROM employees WHERE department\_id = 50;
* VARIANCE 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(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), MAX(hire\_date), MIN(last\_name), MAX(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;

**1 row** will be returned because SUM() is an aggregate function that returns a single result.

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 averages might be skewed because hourly employees were included in the calculation. They likely earn less than salaried employees, and if they were excluded, the average would be higher.

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 query will return **March 30, 1969** as 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 average\_order\_total

FROM orders

WHERE order\_date BETWEEN '01-JAN-2002' AND '21-DEC-2002';

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)

SUM(operating\_cost) will return a larger value than AVG(operating\_cost) because the sum adds all costs, while the average divides the total by the number of rows.

12. Refer to the DJs on Demand database D\_EVENTS table:  
Which code is valid as part of an SQL query?  
\_\_\_\_\_\_\_a. FROM event\_date  
\_\_\_\_\_\_\_b. SELECT SUM(cost)  
\_\_\_\_\_\_\_c. SELECT SUM(event\_date)  
\_\_\_\_\_\_\_d. SELECT AVG(cost) AS "Expense"  
\_\_\_\_\_\_\_e. WHERE MIN(id) = 100  
\_\_\_\_\_\_\_f. SELECT MAX(AVG(cost))  
\_\_\_\_\_\_\_g. SELECT MIN(event\_date)

**Valid SQL Codes**:

* b. SELECT SUM(cost)
* d. SELECT AVG(cost) AS "Expense"
* g. SELECT MIN(event\_date)

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

Vocabulary  
Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **Count** | Returns the number of non-null values in the expression column |
| **Distinct** | The keyword used to return only non-duplicate values or combinations of non-duplicate values in a query. |
| **Count(Distinct)** | Returns the number of unique non-null values in the expression column. |

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 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), COUNT(DISTINCT cd\_number) FROM d\_track\_listings;

4. How many of the DJs on Demand customers have email addresses?

SELECT COUNT(email\_address) FROM customers WHERE email\_address 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(auth\_expense\_amt) 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;

* The first COUNT(shoe\_color) will return the number of non-null values in the shoe\_color column.
* The second COUNT(DISTINCT shoe\_color) will return the number of unique non-null shoe colors.

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;  
\_\_\_\_\_ a. The datatypes of the values in the NVL clause can be any datatype except date data.  
\_\_\_\_\_ b. If the selling\_bonus column has a null value, 0.10 will be substituted.  
\_\_\_\_\_ c. There will be no null values in the selling\_bonus column when the average is calculated.  
\_\_\_\_\_ d. This statement will cause an error. There cannot be two functions in the SELECT  
statement.

* b. If the selling\_bonus column has a null value, 0.10 will be substituted.
* c. There will be no null values in the selling\_bonus column when the average is calculated.

9. Which of the following statements is/are TRUE about the following query?  
SELECT DISTINCT colors, sizes  
FROM items;  
\_\_\_\_\_ a. Each color will appear only once in the result set.  
\_\_\_\_\_ b. Each size will appear only once in the result set.  
\_\_\_\_\_ c. Unique combinations of color and size will appear only once in the result set.  
\_\_\_\_\_ d. Each color and size combination will appear more than once in the result set

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