Database Programming with SQL

7-1: Oracle Equijoin and Cartesian Product

- **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
- **Alias** - Gives a table another name to simplify queries and improve performance
- **Join** - Display data from two or more related tables

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 common column.

```
SELECT *

FROM d_play_list_items dpi

JOIN d_track_listings dtl

ON dpi.common_column = dtl.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 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 c.client_name, e.event_name, ja.job_assignment

FROM d_clients c

JOIN d_events e ON c.client_id = e.client_id
JOIN d_job_assignments ja ON e.event_id = ja.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 dtl.song_id, cds.title

FROM d_track_listings dtl

JOIN d_cds cds ON dtl.song_id = cds.song_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?

- **Data Integration**: Businesses can combine related data (clients, events, and job assignments) to get comprehensive insights.
- **Efficiency**: Reduces the need to manually search across multiple tables, saving time and improving decision-making.
- **Enhanced Reporting**: Allows businesses to create richer, more detailed reports by accessing data from different sources.

7-2: Oracle Nonequijoins and Outer Joins

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 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 needed when there is no direct equality condition between the columns being joined, but instead a condition like BETWEEN, >, <, etc., is used.

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 are joining multiple tables, and the same column names exist in both tables.**
- **You want to simplify queries and improve readability.**
- **It is optional but improves query clarity.**

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

You would use a nonequijoin when you need to find data between a range of numbers.

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 (+);

```
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
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 c.title, t.song_id
FROM cds c
LEFT JOIN track_listings t ON c.cd_number = t.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

A dream nontraditional career of mine that is completely unrelated to the career I am pursuing would be a *Historical Consultant for Film and TV Productions*. I love learning about history and my genre of choice for movies and TV shows is always Historical Fiction/Period Pieces. After I watch a movie/show I tend to find myself researching the historical accuracy of the content afterwards. A historical consultant for film and television productions works closely with directors, set designers, costume departments, and scriptwriters to ensure historical accuracy in period dramas, documentaries, and even fantasy shows. The role involves researching different time periods, advising on everything from the architecture of ancient buildings to the authenticity of props and clothing. This career path would be both creative and fulfilling because I could bring true stories to life as accurately as possible. I would have the opportunity to work on exciting projects, travel to historic sites for research, and see my efforts reflected in film and TV productions viewed by millions. It would also be making history more accessible and engaging to a broader audience which I think is important especially in education to turn a stereotypically boring subject into an exciting visual.

8-1: Group Functions

- **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
- 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\
- **AGGREGATE -** To gather into a sum or whole

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

   ⍰ AVG: Average value
   ⍰ COUNT: Number of rows
   ⍰ SUM: Total of values
   ⍰ MIN: Smallest value
   ⍰ MAX: Largest value
   ⍰ STDDEV: Standard deviation
   ⍰ VARIANCE: Spread around the mean

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,

    MIN(last_name) AS first_alphabetical,

    MAX(last_name) AS last_alphabetical

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;
```

```
SELECT SUM(total_sales)

FROM orders;
```

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 inclusion of hourly wages in the same query as salaried employees could inflate the averages. Hourly wages and salaries should likely be separated or normalized.

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?

SELECT MIN(birthdate)

FROM employees;

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

FROM 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 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 SUM(operating_cost) will be larger than AVG(operating_cost) because the sum aggregates all costs, while the average divides the sum by the number of entries.

12. Refer to the DJs on Demand database D_EVENTS table:
Which code is valid as part of an SQL query?

b. SELECT SUM(cost)
d. SELECT AVG(cost) AS "Expense"

8-2: Count, Distinct, NVL

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

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 different_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_songs, COUNT(DISTINCT cd_number) AS distinct_cds

FROM d_track_listings;
```

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

```
SELECT COUNT(email_address) AS customers_with_email

FROM d_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) AS partners_with_privilege

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 shoe color values and COUNT(DISTINCT shoe_color) will return the number of unique non-null shoe color values.

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

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