Practice 7-1

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 common exclusive toa specific company
alias	Gives a table another name to simplify queries and improve performance
Join conditions	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 d_play_list_items.event_id "event id in playlist",
   d_play_list_items.song_id "song id in playlist",
   d_play_list_items.comments "comments in playlist",
   d_track_listings.song_id "song id in tracklist", d_track_listings.cd_number
   "cd number in tracklist",  d_track_listings.track "track in tracklist"
   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.event_id "event id in playlist",
   d_play_list_items.song_id "song id in playlist",
   d_play_list_items.comments "comments in playlist",
   d_track_listings.song_id "song id in tracklist", d_track_listings.cd_number
   "cd number in tracklist",  d_track_listings.track "track in tracklist"
   FROM  d_play_list_items, d_track_listings
   WHERE d_play_list_items.song_id = d_track_listings.song_id;
   ```

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

   ```
   SELECT  d_songs.title,  d_songs.type_code type, d_types.description
   FROM d_songs, d_types
   WHERE d_songs.type_code = d_types.code;
   ```

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

```
SELECT  d_songs.title,  d_songs.type_code type, d_types.description
FROM d_songs, d_types
WHERE d_songs.type_code = d_types.code AND d_songs.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.email AS "d_clients - email" , d_clients.phone AS
"d_clients - phone" , d_clients.last_name AS "d_clients - last_name" ,
d_clients.first_name AS "d_clients - first_name"  , d_clients.client_number
AS "d_clients - client_number"  , d_events.id AS "d_events - id",
d_events.name AS "d_events - name", d_events.event_date AS "d_events -
event_date", d_events.description AS "d_events - description",
d_events.cost AS "d_events - cost", d_events.venue_id AS "d_events -
venue_id", d_events.package_code AS "d_events - package_code",
d_events.theme_code AS "d_events - theme_code",
d_events.client_number AS "d_events - client_number",
d_job_assignments.partner_id AS "d_job_assignments - partner_id",
d_job_assignments.event_id AS "d_job_assignments - event_id",
d_job_assignments.job_date AS "d_job_assignments - job_date",
d_job_assignments.status AS "d_job_assignments - status"

FROM d_clients, d_events, d_job_assignments
WHERE d_clients.client_number = d_events.client_number AND
d_events.id =  d_job_assignments.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 "song id in tracklist", d_cds.title "title in
cds"
    FROM d_track_listings, d_cds
    WHERE d_track_listings.cd_number = d_cds.cd_number;
```

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 aliases 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?

   To get a more comprehensive result. It is useful for retrieving, modifying, and updating data.

Practice 7-2

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.
   Nonequi join only:

```
SELECT  d_events.name, d_packages.code
FROM d_events, d_packages
WHERE(d_events.cost BETWEEN d_packages.low_range AND
d_packages.high_range);
```

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,job_grades
   WHERE employees.salary BETWEEN job_grades.lowest_sal AND job_grades.highest_sal;
   ```

3. What condition requires the creation of a nonequijoin?

   When there is no exact match (=) between columns of two tables, but still there is a relation hidden.

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?

   Alias for table becomes must when query targets recursive relationships Always use table aliases when joining multiple tables to improve query readability and avoid ambiguity. Consider aliases as a tool to document complex queries.

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

   Nonequi join

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 f_customers.first_name ||' '|| f_customers.last_name "Customer Name", f_orders.order_number, f_orders.order_total, f_orders.order_date
   FROM f_customers, f_orders
   WHERE  f_customers.id = f_orders.cust_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 employees.last_name, employees.department_id,
departments.department_name
FROM employees, departments
WHERE employees.department_id = departments.department_id(+);
```

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

```
SELECT employees.last_name, employees.department_id,
departments.department_name
FROM employees, departments
WHERE employees.department_id(+) = departments.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 (+);
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 d_cds.title , d_track_listings.song_id
FROM  d_cds, d_track_listings
```

```
WHERE d_cds.cd_number = d_track_listings.cd_number(+) ;
```

8-1

AVG	Calculates average value excluding nulls
COUNT	Returns the number of rows with non-nulls 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

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

```
        SELECT  AVG(salary) || ' is Average Salary of ' || COUNT(salary) || ' employees' ||
        (COUNT(*) - COUNT(salary)) || ' rows are skipped in  Average Salary calculation.'
        "Example"
        FROM EMPLOYEES;

                SELECT 'The standard deviation of ' || COUNT(salary) || ' salaries in employees
table is ' || ROUND(STDDEV(salary), 4) ||'. The variance of '|| COUNT(salary) || ' salaries in
employees table is '|| ROUND(VARIANCE(salary), 4) ||'.' "Example3"
        FROM EMPLOYEES;

SELECT 'The sum of ' || COUNT(salary) || ' salaries in employees table is ' || SUM(salary) ||'.'
"Example4"
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 TO_CHAR(ROUND(AVG(salary),2), '$999999.99') as "Average Salary"
   FROM f_staffs
   WHERE manager_id = 19;
   ```

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

   ```
   SELECT TO_CHAR(ROUND(SUM(salary),2), '$999999.99') as "Total Salary"
   FROM f_staffs
   WHERE id in (12, 19);
   ```

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;
   ```

   1

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?

```
SELECT AVG(NVL(salary, hourly_rate* hrs_worked_in_yr  ))
```

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

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 'Average of ' || COUNT(order_number) || '  orders is : ' || AVG(NVL(order_total, 0))
as "Average"
FROM f_orders
WHERE order_date BETWEEN TO_DATE('January 1, 2002', 'fmMonth DD, YYYY') AND
TO_DATE('December 21, 2002', 'fmMonth DD, YYYY')
```

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

```
SELECT MAX(hire_date) as "the last"
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 must be be ‘equal or greater than’ average

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

FALSE a. FROM event_date
TRUE b. SELECT SUM(cost)
FALSE c. SELECT SUM(event_date)
FALSE d. SELECT AVG(cost) AS "Expense"
FALSE e. WHERE MIN(id) = 100
FALSE f. SELECT MAX(AVG(cost))
TRUE g. SELECT MIN(event_date)

8-2

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

1. How many songs are listed in the DJs on Demand D_SONGS table?

```
SELECT COUNT(*)
FROM d_songs;
```

6

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

```
SELECT COUNT(DISTINCT  loc_type)
FROM d_venues;
```

4

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 "song with dupes", COUNT(distinct cd_number)  "cd no. distinct"
FROM d_track_listings
```

5 and 4

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

```
SELECT COUNT(email) "count with email"
FROM d_clients;
```

3

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(*) - COUNT(auth_expense_amt)) "Free from limit count"
FROM d_partners;
```

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

ID	type	Shoe_color
456	oxford	brown
463	sandal	tan
262	heel	black
433	slipper	tan

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

4 and 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 TO_CHAR(ROUND(AVG(NVL(auth_expense_amt,100000)),2), '$999999.99')
FROM d_partners;
```

$166666.67

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

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

FALSE a. The datatypes of the values in the NVL clause can be any datatype except date data.
TRUE b. If the selling_bonus column has a null value, 0.10 will be substituted.

TRUE c. There will be no null values in the selling_bonus column when the average is calculated.
FALSE 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;
   FALSE a. Each color will appear only once in the result set.
   FALSE b. Each size will appear only once in the result set.
   TRUE c. Unique combinations of color and size will appear only once in the result set.
   FALSE d. Each color and size combination will appear more than once in the result set.