***SUBMITTED BY: - PARTH PARASHAR***

***PSU ID: -923928157***

***Activity- 5.1 Natural Join***

**Problem statement: -** Rewrite the SQL query without using natural join

Select title, name from books NATURAL JOIN publishers;

**Solution: -** The equivalent query for the problem is given below

select title, name from books NATURAL JOIN publishers;

The screenshot below provides the result for the same

Text

Description automatically generated

***Activity-5.2 Union Compatibility***

**Problem Statement: -**

Create a union compatible table with Authors

Write Relational Algebra expression to find union of authors and cities

Describe the result of relational algebra expression

**Solution: -**

The structure and contents of the authors table is given below: -

Text

Description automatically generated

The new table with structure and contents is given below: -

Text

Description automatically generated

These tables are both union compatible as both have three columns and both have same data types.

The relational Algebra expression is given below: -

Authors U Cities

The result will not make much sense. The only thing it will do is interlink the authors’ tables attributes with the attributes of the Cities tables.

***Activity-5.3 Union and Union All Activity: -***

**Problem Statement: -**

1. SQL query for authors union author
2. Rewrite 1 with union all
3. Difference between the answers

**Answer: -**

1. (Select name from authors) union (select name from authors);

The output of this query is given in the screenshot provided below: -

A picture containing text

Description automatically generated

1. (Select name from authors) union all (select name from authors);

The output for this query is given in the screenshot provided below: -

A picture containing text

Description automatically generated

1. The difference between both the answers is that Union does not include any duplicates where the result generated from union all can have duplicate values. Therefore, the number of rows in the result of union all is more than that of union.

**Challenge: -**

1. **Intersect**

(Select name from authors) intersect (select name from authors);

The screenshot for the same is provided below: -

A picture containing text

Description automatically generated

1. **Intersect all**

(select name from authors) intersect all (select name from authors);

The screenshot for this query is given below: -

Text

Description automatically generated with low confidence

1. **Except**

(select name from authors) except (select name from authors);

The screenshot for this query is given below: -

Text

Description automatically generated

1. **Except All**

(select name from authors) except all (select name from authors);

The screenshot for this query is given below: -

Text

Description automatically generated

***ACTIVITY-6.1.A Outer Join Activity***

**Problem Statement: -** Write SQL queries for the following: -

1. To find the titles of the books and the names of their authors which includes the books that do not have an author
2. To list the titles of the books and the names of their authors including books without an author and authors without a book

**Solution: -**

1. The query for this problem is: -

select b.title, a.name from books b left outer join authors a on b.authorid = a.id;

The screenshot for the result of this query is attached below: -

Text

Description automatically generated

1. The query for this problem is: -

select b.title, a.name from books b full outer join authors a on b.authorid = a.id;

The screenshot containing the result for this query is given below: -

Text

Description automatically generated

***ACTIVITY-6.1.B Outer Join Activity-II***

**Problem statement: -**

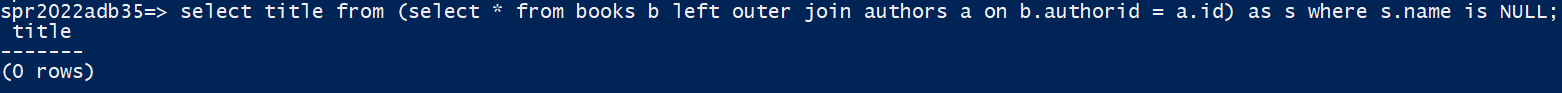
Write a query to find titles of books that do not have an author

**Solution: -**

The query that will be used is: -

select title from (select \* from books b left outer join authors a on b.authorid = a.id) as s where s.name is NULL;

The screenshot contains the result of this query



As we can see from the screenshot that the result of this query is 0 rows.

This is because each title has an author which can be confirmed by doing a select \* on both books and authors tables.

Text

Description automatically generated

***Activity-6.1.C Join on Null Attributes***

**Problem Statement: -**

1. Run the query
2. What happens when pubid is null

**Solution: -**

1. Select title, name from books b inner join publishers p on b.pubid = p.id;

Text

Description automatically generated

1. When pubid is null, then that book is not printed in the result because publisher id is the primary key for the publishers table and therefore, it cannot have null values. Since publishers’ Id cannot be null for publishers table, therefore, the equality condition will not hold and hence it will not print it in the result.

Also

Null compared with null will give null.

This is also given in the screenshot below: -

***Text

Description automatically generated***

***Activity-6.2.A Group By/Having Clause***

**Problem Statement: -**

Find the number of books by each author

1. Return author id and count
2. Return author name

**Solution: -**

1. select authorid, count(\*) from books group by authorid;

The screenshot depicting the result of this query is given below: -

A picture containing text

Description automatically generated

1. select a.name from books b, authors a where b.authorid = a.id group by a.name;

The screenshot depicting the result of this query is given below: -

A picture containing text

Description automatically generated

***ACTIVITY 6.2.B- Group By/ Having clause***

**Problem statement: -**

Find the authors that have written two or more books

1. return authored and count
2. return author name

**Solution: -**

1. select authorid from books group by authorid having count(\*) >=2;

The screenshot depicting the result of this query is given below: -

Text

Description automatically generated with medium confidence

1. select a.name from books b, authors a where b.authorid = a.id group by a.name having count(\*) >= 2;

The screenshot depicting the result of this query is given below: -

Text

Description automatically generated

***ACTIVITY 6.2.C – Group by and nulls***

**Problem statement: -**

Run the following queries and print their result after inserting tuples in the boats database.

**Solution: -**

**Step-1: - Insertion into the boats table**

The following screenshot contains the queries used to insert tuples into the boats table

Text

Description automatically generated

Here, we can see that the query-2 provided for inserting into the boats table has an error. This is because in the slides, bid column’s values is repeated. Since bid is a primary key, it cannot be the same for two tuples. Also, the value of colour is null for the entry with bid of 107.

**Step-2: - Verifying the insertion**

The insertion can be verified by printing all the values present in the boats table as given in the screenshot below: -

A picture containing table

Description automatically generated

**Step-3: -** Running the queries given and printing the results along with comments

1. select bname, count(\*) from boats group by bname;

The result of this query is given in the screenshot below: -

Text

Description automatically generated

This is according to my expectation as count is on the group by clause of bname.

1. select bname, count(color) from boats group by bname;

The result of this query is given in the screenshot provided below: -

Graphical user interface, text

Description automatically generated

This is in accordance of my expectations as the count function was on the color and it will not take NULL value. Hence the number of ‘Interlake’ is 3.

1. select bname, count(distinct color) from boats group by bname;

The result of this query is given in the screenshot provided below: -

Graphical user interface, text

Description automatically generated

This is in accordance with what I was expecting as it will see distinct colors count.