

Week -3 Video Activity

Activity 5.1 – Natural JOIN

1. Write a SQL statement that is equivalent to the following statement without using NATURAL JOIN (that is explicitly specify the join attributes)

```
SELECT title, name
```

```
FROM books NATURAL JOIN publishers;
```

Answer:

```
SELECT title, name
```

```
FROM books b, publishers p
```

```
WHERE b.id = p.id;
```

Activity 5.2 – Union Compatibility

1. Create a table that is union, compatible with authors but which will contain data about cities

Answer:

Cities (id, name, population)

2. Write a relational algebra expression to find the union of Authors and cities

Answer:

Authors U Cities

3. Write one sentence describing the result of that relational algebra expression

Answer:

The result is valid but it does not make sense. It will interleave the attributes of cities table which are id(int), name(string), population(int) with the attributes of Authors table which are id(int), name(string), age(int).

Activity 5.3– UNION and UNION ALL

1. Write a SQL query that unions the Authors relation with itself

Answer:

(SELECT name FROM authors) UNION (SELECT name FROM authors);

2. Rewrite the query with UNION ALL

Answer:

(SELECT name FROM authors) UNION ALL (SELECT name FROM authors);

3. How do the answer differ?

Answer:

When we use **UNION**, each author name will appear only once. When we use **UNION ALL**, each author name will appear atleast twice.

Challenges– INTERSECT and INTERSECT ALL

1. Write a SQL query that intersects the Authors relation with itself

Answer:

(SELECT name FROM authors) INTERSECT (SELECT name FROM authors);

2. Rewrite the query with UNION ALL

Answer:

(SELECT name FROM authors) INTERSECT ALL (SELECT name FROM authors);

3. How do the answer differ?

Answer:

When we use **INTERSECT**, each author name will appear only once. When we use **INTERSECT ALL**, each author name will appear only once. The result of both query is identical.

Challenges– EXCEPT and EXCEPT ALL

1. Write a SQL query that expect the Authors relation with itself

Answer:

(SELECT name FROM authors) EXCEPT (SELECT name FROM authors);

2. Rewrite the query with UNION ALL

Answer:

(SELECT name FROM authors) EXCEPT ALL (SELECT name FROM authors);

3. How do the answer differ?

Answer:

When we use **INTERSECT**, each author name would not appear. When we use **INTERSECT ALL**, each author name would not appear. The result of both query returns nothing.

Activity 6.1 a– Outer JOIN

1. Write a query to find titles of Books and the names of their Authors; include Books that do not have an author

Answer:

```
SELECT b.title, a.name  
FROM books b LEFT OUTER JOIN authors a  
ON b.authorid = a.id;
```

2. Write one query to list the title of Books and the names of their Authors; include Books without an author and Authors without a book

Answer:

```
SELECT b.title, a.name  
FROM books b FULL OUTER JOIN authors a  
ON b.authorid = a.id;
```

Activity 6.1 b– Outer JOIN

1. Write a query to find the titles of Books that do not have an author

Answer:

```
(SELECT b.title FROM books b LEFT OUTER JOIN authors a
```

```
ON b.authorid = a.id)
```

```
EXCEPT
```

```
(SELECT title FROM books b JOIN authors a ON b.authorid = a.id);
```

Activity 6.1 c– Join on NULL Attribute

1. Run this query:

```
SELECT title, name
```

```
FROM books b INNER JOIN publishers p
```

```
ON b.pubid = p.id;
```

Answer:

```
su22adb20-> ON b.pubid = p.id;
              title                               |          name
-----+-----
It                               | Viking
Hamlet                           | HarperCollins
I Know Why the Caged Bird Sings | Penguin
A Suitable Boy                   | HarperCollins
The Joy Luck Club                | Putnam
Like Water for Chocolate        | PerfectionLearning
From Heaven Lake                 | Penguin
Kite Runner                      | Riverhead
The Vanishing Half               | Riverhead
September Love                  | AndrewsMcMeel
The Nickel Boys                  | Doubleday
The Alchemist                    | HarperCollins
Love and Misadventure            | AndrewsMcMeel
The Authenticity Project         | Penguin
(14 rows)
```

Total number of rows =14 rows

2. What happens when pubid is null?

Answer:

Rows from books where pubid is null (Tita's Diary) are not included in the result because null is not = to any pubid in publishers.

Activity 6.2a – GROUP BY/ HAVING

Write the following SQL queries:

1. Find the number of books by each author

a. Return authorid and count

Answer:

```
SELECT authorid, count(*)  
FROM books  
GROUP BY authorid ;
```

b. Return author name

Answer:

```
SELECT a.name  
FROM books b, authors a  
WHERE b.authorid = a.id  
GROUP BY a.name;
```

Activity 6.2b – GROUP BY/ HAVING

Write the following SQL queries:

1. Find the author that have written two or more books

a. Return authorid and count

Answer:

```
SELECT authorid  
FROM books b  
GROUP BY authorid  
HAVING count(*)>=2 ;
```

b. Return author name

Answer:

```
SELECT a.name  
FROM books b, authors a  
WHERE b.authorid = a.id  
GROUP BY a.name  
HAVING count(*)>=2;
```

Activity 6.2C – GROUP BY and Nulls

1. Insert tuples (106, Interlake, Red) and (106, Interlake) into the boats relation in your database.

Answer:

```
INSERT into boats values(106,'Interlake','Red');  
INSERT into boats values(106,'Interlake');
```

2. **SELECT * FROM boats;**
and look at the answer.

Answer:

```
su22adb20=> select * from boats;
bid |      bname      | color
-----+-----+-----
101 | Interlake | blue
102 | Interlake | red
103 | Clipper    | green
104 | Marine     | red
105 | Tubby      | purple
106 | Interlake  | Red
107 | Interlake  |
(7 rows)
```

3. For the following 3 queries: first, think about what you think the answer will be, then run the queries and see if you are correct.
SELECT name, count(*) from boats group by name;

```
su22adb20=> SELECT bname, count(*) from boats group by bname;
      bname      | count
-----+-----
Clipper          |      1
Tubby            |      1
Marine           |      1
Interlake        |      4
(4 rows)
```

SELECT name, count(color) from boats group by name;

```
su22adb20=> SELECT bname, count(color) from boats group by bname;
      bname      | count
-----+-----
Clipper          |      1
Tubby            |      1
Marine           |      1
Interlake        |      3
(4 rows)
```

Note: Count is 3 because color is null for one Interlake

SELECT name, count(distinct color) from boats group by name;

```
su22adb20=> SELECT bname, count(distinct color) from boats group by bname;
  bname  | count
-----+-----
Clipper  |      1
Interlake |      3
Marine   |      1
Tubby    |      1
(4 rows)
```

Note: There are red and blue Interlake boats. There is also an Interlake boat with a null color, but that boat does not get included in the count.