DAY-4

Task 1: 1)Convert bookstore.xml into json(file is located in Presentation->XML folder)

Solution:


```
{
"bookstore": {
"book": [
{
 "title": "Harry Potter",
 "author": "J.K. Rowling",
 "price": 29.99,
"available": true
},
{
"title": "The Hobbit",
"author": "J.R.R. Tolkien",
"price": 19.99,
"available": false
}
]
}
}
```


2)Write a query to give inner join,left outer join,right outer join and full outer join(refer SQL_Assignments in Presentation folder)

Solution:

Inner join:

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all

rows from both the tables where the condition satisfies i.e value of the common field will be the same.

SQL LEFT JOIN

LEFT JOIN returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.

SQL RIGHT JOIN

RIGHT JOIN returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join.It is very similar to LEFT JOIN For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.

SQL FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.

```
mysql> create database day4
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> use day4;
Database changed

mysql> create table department(department_id int primary key,department_name varchar(200));
Query OK, 0 rows affected (0.02 sec)

mysql> insert into department values (10,"HR"),(20,"sales"),(30,"IT"),(40,"marketing");
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0


mysql> select * from department;
+---------------+-----------------+
```

```
| department_id | department_name |
+---------------+-----------------+
|            10 | HR              |
|            20 | sales           |
|            30 | IT              |
|            40 | marketing       |
+---------------+-----------------+
4 rows in set (0.00 sec)

mysql> CREATE TABLE employee (
    ->     employee_id INT,
    ->     first_name VARCHAR(255),
    ->     last_name VARCHAR(255),
    ->     department_id INT,
    ->                     FOREIGN KEY (department_id) REFERENCES
department(department_id)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> insert into employee values(1,"john","doe",10);
Query OK, 1 row affected (0.00 sec)

mysql>insertintoemployeevalues(2,"jane","smith",20),(3,"mike","jhonson",30),(
4,"emily","davis",40);
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
4 rows in set (0.01 sec)
mysql> select * from employee;
```

```
+-------------+------------+-----------+---------------+
| employee_id | first_name | last_name | department_id |
+-------------+------------+-----------+---------------+
|           1 | john       | doe       |            10 |
|           2 | jane       | smith     |            20 |
|           3 | mike       | jhonson   |            30 |
|           4 | emily      | davis     |            40 |
+-------------+------------+-----------+---------------+
4 rows in set (0.00 sec)

mysql> SELECT employee.employee_id, employee.first_name, employee.last_name, department.department_id, department.department_name
    -> FROM employee
    -> INNER JOIN department ON employee.department_id = department.department_id;
+-------------+------------+-----------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+---------------+-----------------+
|           1 | john       | doe       |            10 | HR              |
|           2 | jane       | smith     |            20 | sales           |
|           3 | mike       | jhonson   |            30 | IT              |
|           4 | emily      | davis     |            40 | marketing       |
+-------------+------------+-----------+---------------+-----------------+
4 rows in set (0.00 sec)

mysql> SELECT employee.employee_id, employee.first_name, employee.last_name, department.department_id, department.department_name
    -> FROM employee
    -> LEFT JOIN department ON employee.department_id = department.department_id;
+-------------+------------+-----------+---------------+-----------------+
```

| employee_id | first_name | last_name | department_id | department_name |

+-------------+------------+-----------+---------------+-----------------+

| 1 | john | doe | 10 | HR |

| 2 | jane | smith | 20 | sales |

| 3 | mike | jhonson | 30 | IT |

| 4 | emily | davis | 40 | marketing |

+-------------+------------+-----------+---------------+-----------------+

4 rows in set (0.00 sec)

mysql> SELECT employee.employee_id, employee.first_name, employee.last_name, department.department_id, department.department_name

-> FROM employee

-> RIGHT JOIN department ON employee.department_id = department.department_id;

+-------------+------------+-----------+---------------+-----------------+

| employee_id | first_name | last_name | department_id | department_name |

+-------------+------------+-----------+---------------+-----------------+

| 1 | john | doe | 10 | HR |

| 2 | jane | smith | 20 | sales |

| 3 | mike | jhonson | 30 | IT |

| 4 | emily | davis | 40 | marketing |

+-------------+------------+-----------+---------------+-----------------+

4 rows in set (0.00 sec)

mysql> SELECT employee.employee_id, employee.first_name, employee.last_name, department.department_id, department.department_name

-> FROM employee

-> LEFT JOIN department ON employee.department_id = department.department_id

-> UNION

-> SELECT employee.employee_id, employee.first_name, employee.last_name, department.department_id, department.department_name

-> FROM employee

-> RIGHT JOIN department ON employee.department_id = department.department_id;

```
+-------------+------------+-----------+--------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+--------------+-----------------+
|           1 | john       | doe       |           10 | HR              |
|           2 | jane       | smith     |           20 | sales           |
|           3 | mike       | jhonson   |           30 | IT              |
|           4 | emily      | davis     |           40 | marketing       |
+-------------+------------+-----------+--------------+-----------------+
```

4 rows in set (0.01 sec)

3)Write a query to find duplicate records(refer SQL_Assignments in Presentation folder)

Solution

mysql> create table employee1(employee_id int,firstname varchar(100),lastname varchar(100),email varchar(100));

Query OK, 0 rows affected (0.02 sec)

mysql> insert into employee1 values(1,"john","doe","john.doe@example.com");

Query OK, 1 row affected (0.00 sec)

mysql> insert into employee1 values(2,"jane","smith","jane.smith@example.com");

Query OK, 1 row affected (0.00 sec)

mysql> insert into employee1 values(3,"john","doe","john.doe@example.com");

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into employee1 values(4,"emily","davis","emily.davis@example.com");
Query OK, 1 row affected (0.00 sec)
mysql> select * from employee1;
+-------------+-----------+----------+------------------------+
| employee_id | firstname | lastname | email                  |
+-------------+-----------+----------+------------------------+
|           1 | john      | doe      | john.doe@example.com   |
|           2 | jane      | smith    | jane.smith@example.com |
|           3 | john      | doe      | john.doe@example.com   |
|           4 | emily     | davis    | emily.davis@example.com |
+-------------+-----------+----------+------------------------+
4 rows in set (0.00 sec)
mysql> SELECT first_name, COUNT(*)
    -> FROM employee
    -> GROUP BY first_name
    -> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
mysql> SELECT firstname, COUNT(*)
    -> FROM employee1
    -> GROUP BY firstname
    -> HAVING COUNT(*) > 1;
+-----------+----------+
| firstname | COUNT(*) |
+-----------+----------+
| john      |        2 |
+-----------+----------+
1 row in set (0.00 sec)
```

```
mysql> SELECT email, COUNT(*)
    -> FROM employee1
    -> GROUP BY email
    -> HAVING COUNT(*) > 1;
+---------------------+----------+
| email               | COUNT(*) |
+---------------------+----------+
| john.doe@example.com |        2 |
+---------------------+----------+
1 row in set (0.00 sec)
mysql> SELECT email,firstname COUNT(*)
    -> FROM employee1
    -> GROUP BY email,firstname
    -> HAVING COUNT(*) > 1;
FROM employee1
GROUP BY email,firstname
HAVING COUNT(*) > 1' at line 1
mysql> SELECT email,firstname, COUNT(*)
    -> FROM employee1
    -> GROUP BY email,firstname,
    -> HAVING COUNT(*) > 1;
mysql> SELECT firstname, lastname, COUNT(*)
    -> FROM employee1
    -> GROUP BY firstname, lastname
    -> HAVING COUNT(*) > 1;
+-----------+----------+----------+
| firstname | lastname | COUNT(*) |
```

```
+-----------+----------+----------+
| john      | doe      |        2 |
+-----------+----------+----------+
1 row in set (0.00 sec)
mysql> SELECT firstname,email, COUNT(*)
    -> FROM employee1
    -> GROUP BY firstname,email
    -> HAVING COUNT(*) > 1;
+-----------+---------------------+----------+
| firstname | email               | COUNT(*) |
+-----------+---------------------+----------+
| john      | john.doe@example.com |        2 |
+-----------+---------------------+----------+
1 row in set (0.00 sec)
```