

1) Convert bookstore.xml into json

Given XML code is:

```
<bookstore>

  <book>

    <title>Harry Potter</title>

    <author>J.K. Rowling</author>

    <price>29.99</price>

    <available>true</available>

  </book>

  <book>

    <title>The Hobbit</title>

    <author>J.R.R. Tolkien</author>

    <price>19.99</price>

    <available>false</available>

  </book>

</bookstore>
```

Given XML to Json:

```
{
  "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

- INNER JOIN

```

mysql> SELECT e.Emp_Id,e.First_name,e.Last_name,d.Dept_name
-> FROM Employee e
-> INNER JOIN Dept d
-> ON e.Dept_Id=d.Dept_Id;

```

```

+-----+-----+-----+-----+
| Emp_Id | First_name | Last_name | Dept_name |
+-----+-----+-----+-----+
| 1 | Kiran | Dedavath | HR |
| 2 | Srivyshnavi | Kummera | Sales |
| 3 | Ruchitha | Kanaparthi | IT |
| 4 | Prathyusha | Pisati | Marketing |
+-----+-----+-----+-----+

```

4 rows in set (0.00 sec)

- LEFT OUTER JOIN

A left outer join returns all rows from the left table and the matched rows from the right table. If there is no match,the result is NULL on the side of the right table.

```

mysql> SELECT e.Emp_Id,e.First_name,e.Last_name,d.Dept_name
-> FROM Employee e

```

-> LEFT OUTER JOIN Dept d

-> ON e.Dept_Id=d.Dept_Id;

```
+-----+-----+-----+-----+
| Emp_Id | First_name | Last_name | Dept_name |
+-----+-----+-----+-----+
| 1 | Kiran | Dedavath | HR |
| 2 | Srivyshnavi | Kummera | Sales |
| 3 | Ruchitha | Kanaparthi | IT |
| 4 | Prathyusha | Pisati | Marketing |
+-----+-----+-----+-----+
```

4 rows in set (0.00 sec)

- RIGHT OUTER JOIN

A right outer join returns all rows from the right table and the matched rows from the left table .If there is no match,the result is NULL on the side of left table

```
mysql> SELECT e.Emp_Id,e.First_name,e.Last_name,d.Dept_name
```

-> FROM Employee e

-> RIGHT OUTER JOIN Dept d

-> ON e.Dept_Id=d.Dept_Id;

```
+-----+-----+-----+-----+
| Emp_Id | First_name | Last_name | Dept_name |
+-----+-----+-----+-----+
| 1 | Kiran | Dedavath | HR |
| 2 | Srivyshnavi | Kummera | Sales |
| 3 | Ruchitha | Kanaparthi | IT |
| 4 | Prathyusha | Pisati | Marketing |
| NULL | NULL | NULL | CSE |
+-----+-----+-----+-----+
```

5 rows in set (0.00 sec)

- FULL OUTER JOIN

A full outer join returns all rows when there is a match in either left or right table. If there is no match, the result is NULL from side that does not have a match.

```
mysql> SELECT e.Emp_Id,e.First_name,e.Last_name,d.Dept_name
-> FROM Employee e
-> LEFT JOIN Dept d
-> ON e.Dept_Id=d.Dept_Id
-> UNION
-> SELECT e.Emp_Id,e.First_name,e.Last_name,d.Dept_name
-> FROM Employee e
-> RIGHT JOIN Dept d
-> ON e.Dept_Id=d.Dept_Id
-> ;
```

```
+-----+-----+-----+-----+
| Emp_Id | First_name | Last_name | Dept_name |
+-----+-----+-----+-----+
| 1 | Kiran | Dedavath | HR |
| 2 | Srivyshnavi | Kummera | Sales |
| 3 | Ruchitha | Kanaparthi | IT |
| 4 | Prathyusha | Pisati | Marketing |
| NULL | NULL | NULL | CSE |
+-----+-----+-----+-----+
```

5 rows in set (0.01 sec)

3) Write a query to find duplicate records

- Based on first name

```
mysql> SELECT First_name,Count(*)
-> FROM Employee e
-> GROUP BY First_name
```

-> HAVING COUNT(*) > 1;

```
+-----+-----+
| First_name | Count(*) |
+-----+-----+
| Kiran      | 2        |
+-----+-----+
```

1 row in set (0.00 sec)

- Based on Email

mysql> SELECT Email,Count(*)

-> FROM Employee

-> GROUP BY Email

-> HAVING COUNT(*) > 1;

```
+-----+-----+
| Email    | Count(*) |
+-----+-----+
| kiran@123 | 2        |
+-----+-----+
```

1 row in set (0.00 sec)

- Based on First name and last name

mysql> SELECT First_name,Last_name,Count(*)

-> FROM Employee

-> GROUP BY First_name,Last_name

-> HAVING COUNT(*) > 1;

```
+-----+-----+-----+
| First_name | Last_name | Count(*) |
+-----+-----+-----+
| Kiran      | Yshu      | 2        |
+-----+-----+-----+
```

1 row in set (0.00 sec)

- Based on First name and Email

```
mysql> SELECT First_name,Email,Count(*)
```

```
-> FROM Employee
```

```
-> GROUP BY First_name,Email
```

```
-> HAVING COUNT(*) > 1;
```

```
+-----+-----+-----+  
| First_name | Email   | Count(*) |  
+-----+-----+-----+  
| Kiran     | kiran@123 | 2 |
```

```
+-----+-----+-----+
```

```
1 row in set (0.00 sec)
```