DAY 3 CONVERTING XML CODE TO JSON CODE

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

SQL 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.

The syntax for SQL INNER JOIN is:

```
SELECT table1.column1,table1.column2,table2.column1,....
FROM table1
INNER JOIN table2
ON  table1.matching_column = table2.matching_column;
```

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.

The syntax of LEFT JOIN in SQL is:

SELECT table1.column1,table1.column2,table2.column1,....
FROM table1
LEFT JOIN table2
ON table1.matching_column = table2.matching_column;

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.

The syntax of RIGHT JOIN in SQL is:

SELECT table1.column1,table1.column2,table2.column1,....
FROM table1
RIGHT JOIN table2
ON table1.matching_column = table2.matching_column;

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.

The syntax of SQL FULL JOIN is:

SELECT table1.column1,table1.column2,table2.column1,....
FROM table1
FULL JOIN table2
ON table1.matching_column = table2.matching_column;

QUERIES ON INNER ,OUTER JOINS AND DUPLICATES OF DEPARTMENT AND EMPLOYEE TABLES AS FOLLOWS :

1) mysql> SELECT *FROM DEPARTMENT;

```
+---------+-------------+
| dept_id | dept_name   |
+---------+-------------+
|       1 | ENGINEERING |
|       2 | EEE         |
|       3 | AIML        |
|       4 | DS          |
|       5 | CSE         |
|       6 | MARKETING   |
```

2) mysql> select *from employee;

```
+--------+---------+--------+--------------------+---------+
| emp_id | f_name  | l_name | email_id   | dept_id |
+--------+---------+--------+--------------------+---------+
|      1 | john    | doe    | john@123.com       |       1 |
|      2 | sangam  | joshna | sangam@321.com     |       4 |
|      3 | david   | brown  | david brown@152.com |       5 |
|      4 | jyoshna | sangam | jyo@176.com        |       2 |
|      5 | male    | jeevan | jkumar@889.com     |       4 |
|      6 | john    | doe    | NULL               |       1 |
+--------+---------+--------+--------------------+---------+
```

3) mysql> select e.emp_id, e.f_name, e.l_name, d.dept_name from employee e **INNER JOIN** department d ON e.dept_id=d.dept_id;

```
+--------+---------+--------+-------------+
| emp_id | f_name  | l_name | dept_name   |
+--------+---------+--------+-------------+
|      1 | john    | doe    | ENGINEERING |
|      2 | sangam  | joshna | DS          |
|      3 | david   | brown  | CSE         |
|      4 | jyoshna | sangam | EEE         |
|      5 | male    | jeevan | DS          |
|      6 | john    | doe    | ENGINEERING |
+--------+---------+--------+-------------+
```

4) select e.emp_id,e.f_name,e.l_name,d.dept_name from employee e **LEFT OUTER JOIN** department d on e.dept_id=d.dept_id;

```
+--------+---------+--------+-------------+
| emp_id | f_name  | l_name | dept_name   |
+--------+---------+--------+-------------+
|      1 | john    | doe    | ENGINEERING |
|      2 | sangam  | joshna | DS          |
```

| 3 | david | brown | CSE |

| 4 | jyoshna | sangam | EEE |

| 5 | male | jeevan | DS |

| 6 | john | doe | ENGINEERING |

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

5) select e.emp_id,e.f_name,e.l_name,d.dept_name from employee e **RIGHT OUTER JOIN** department d on e.dept_id=d.dept_id;

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

| emp_id | f_name | l_name | dept_name |

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

| 1 | john | doe | ENGINEERING |

| 6 | john | doe | ENGINEERING |

| 4 | jyoshna | sangam | EEE |

| NULL | NULL | NULL | AIML |

| 2 | sangam | joshna | DS |

| 5 | male | jeevan | DS |

| 3 | david | brown | CSE |

| NULL | NULL | NULL | MARKETING |

6) select f_name,count(*)from employee **GROUP BY** f_name **having** count(*)>1;

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

| f_name | count(*) |

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

| john | 2 |

7) select f_name,l_name,count(*)from employee **GROUP BY** f_name,l_name **having** count(*)>1;

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

| f_name | l_name | count(*) |

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

| john | doe | 2 |

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