**1)Convert bookstore.xml into json**

**XML File**

<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>

**JSON File**

{

"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**

Employee Table

===========

employee\_id first\_name last\_name department\_id

1 John Doe 10

2 Jane Smith 20

3 Mike Johnson 30

4 Emily Davis 10

Department Table

=========

department\_id department\_name

10 HR

20 Sales

30 IT

40 Marketing

**DESCRIPTION:**

**INNER JOIN:**

An INNER JOIN in MySQL combines rows from two or more tables based on a related column between them. It returns only the rows that have matching values in both tables.

**SELECTe.employee\_id,e.first\_name, e.last\_name,e.department\_id,d.department\_name**

**FROM Employees e**

**INNER JOIN Department d ON e.department\_id=d.department\_id;**

| **employee\_id** | **first\_name** | **last\_name** | **department\_id** | **department\_name** |
| --- | --- | --- | --- | --- |
| 1 | John | Doe | 10 | HR |
| 2 | Jane | Smith | 20 | Sales |
| 3 | Mike | Johnson | 30 | IT |
| 4 | Emily | Davis | 10 | HR |

**LEFT OUTER JOIN:**

Returns all rows from the left table, and the matched rows from the right table. If no match is found, NULL values are returned for columns from the right table.

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.department\_id, d.department\_name**

**FROM Employee AS e**

**LEFT JOIN Department AS d ON e.department\_id = d.department\_id;**

| **employee\_id** | **first\_name** | **last\_name** | **department\_id** | **department\_name** |
| --- | --- | --- | --- | --- |
| 1 | John | Doe | 10 | HR |
| 2 | Jane | Smith | 20 | Sales |
| 3 | Mike | Johnson | 30 | IT |
| 4 | Emily | Davis | 10 | HR |

**RIGHT OUTER JOIN:**

Returns all rows from the right table, and the matched rows from the left table. If no match is found, NULL values are returned for columns from the left table.

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.department\_id, d.department\_name**

**FROM Employee AS e**

**RIGHT JOIN Department AS d ON e.department\_id = d.department\_id;**

| **employee\_id** | **first\_name** | **last\_name** | **department\_id** | **department\_name** |
| --- | --- | --- | --- | --- |
| 1 | John | Doe | 10 | HR |
| 2 | Jane | Smith | 20 | Sales |
| 3 | Mike | Johnson | 30 | IT |
| 4 | Emily | Davis | 10 | HR |
| NULL | NULL | NULL | 40 | Marketing |

**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 on the side that does not have a match

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.department\_id, d.department\_name**

**FROM Employee AS e**

**LEFT JOIN Department AS d ON e.department\_id = d.department\_id**

**UNION**

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.department\_id, d.department\_name**

**FROM Employee AS e**

**RIGHT JOIN Department AS d ON e.department\_id = d.department\_id;**

| **employee\_id** | **first\_name** | **last\_name** | **department\_id** | **department\_name** |
| --- | --- | --- | --- | --- |
| 1 | John | Doe | 10 | HR |
| 2 | Jane | Smith | 20 | Sales |
| 3 | Mike | Johnson | 30 | IT |
| 4 | Emily | Davis | 10 | HR |
| NULL | NULL | NULL | 40 | Marketing |

**3)** **Write a query to find duplicate records**

| **employee\_id** | **first\_name** | **last\_name** | **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 |

**Find Duplicate records**

**1) Based on firstName**

**2) based on email**

**3) Based on firstname and Last Name**

**4) Based on firstname and Email**

**1) Based on firstName:**

SELECT first\_name, COUNT(\*) FROM Employee GROUP BY first\_name HAVING COUNT(\*) > 1;

| **first\_name** | **COUNT(\*)** |
| --- | --- |
| John | 2 |

**2) based on email**

SELECT email, COUNT(\*) FROM Employee GROUP BY email HAVING COUNT(\*) > 1;

| **email** | **COUNT(\*)** |
| --- | --- |
| john.doe@example.com | 2 |

**3) Based on firstname and Last Name**

SELECT first\_name, last\_name, COUNT(\*) FROM Employee GROUP BY first\_name, last\_name HAVING COUNT(\*) > 1;

| **first\_name** | **last\_name** | **COUNT(\*)** |
| --- | --- | --- |
| John | Doe | 2 |

**4) Based on firstname and email**

SELECT first\_name, email, COUNT(\*) FROM Employee GROUP BY first\_name, email HAVING COUNT(\*) > 1;

| **first\_name** | **email** | **COUNT(\*)** |
| --- | --- | --- |
| John | john.doe@example.com | 2 |