1. Convert bookstore .xml into .json

XML Data

```
<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 Data

```
{
  "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

a) INNER JOIN

The INNER JOIN will return records that have matching values in both tables.

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employee e
INNER JOIN department d
ON e.department_id = d.department_id;
```

b) LEFT OUTER JOIN

The LEFT OUTER JOIN will return all records from the left table (employee), and the matched records from the right table (department). The result is NULL from the right side if there is no match.

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employee e
LEFT OUTER JOIN department d
```

```
ON e.department_id = d.department_id;
```

c) RIGHT OUTER JOIN

The RIGHT OUTER JOIN will return all records from the right table (department), and the matched records from the left table (employee). The result is NULL from the left side if there is no match

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employee e
RIGHT OUTER JOIN department d
ON e.department_id = d.department_id;
```

d) FULL OUTER JOIN

The FULL OUTER JOIN will return all records when there is a match in either left (employee) or right (department) table records.

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name

FROM employee e

FULL OUTER JOIN department d

ON e.department_id = d.department_id;
```

3. Write a query to find duplicate records

a) Based on first_name

```
SELECT first_name, COUNT(*)
FROM employees
GROUP BY first_name
HAVING COUNT(*) > 1;
```

b) Based on email

```
SELECT email, COUNT(*)
FROM employees
GROUP BY email
HAVING COUNT(*) > 1;
```

c)Based on `first_name` and `last_name`

```
SELECT first_name, last_name, COUNT(*)
FROM employees
GROUP BY first_name, last_name
HAVING COUNT(*) > 1;
```

d)Based on the first_name and email

```
SELECT first_name, email, COUNT(*)
  FROM employees
GROUP BY first_name, email
HAVING COUNT(*) > 1;
```