#DAY-4

1)Converting bookstore.xml into json:

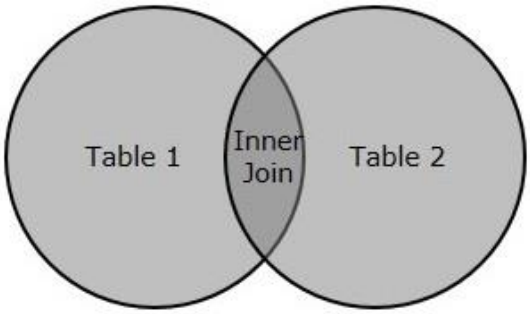
Input:bookstore.xml

Output:bookstore.json

2) Write a query to give inner join,left outer join,right outer join and full outer join:

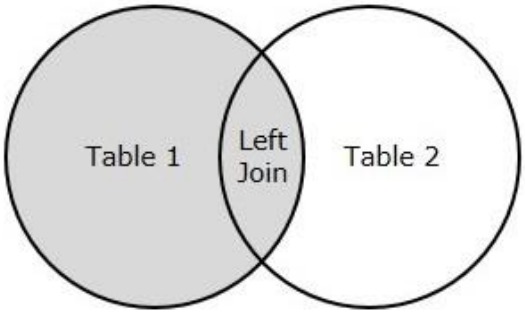
INNER JOIN:

Inner Join: It is a type of join that combines multiple tables by retrieving records that have matching values in both tables

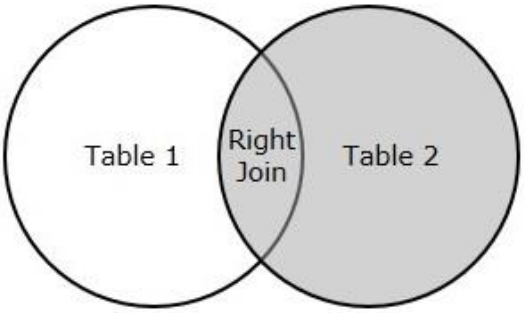


OUTER JOIN:

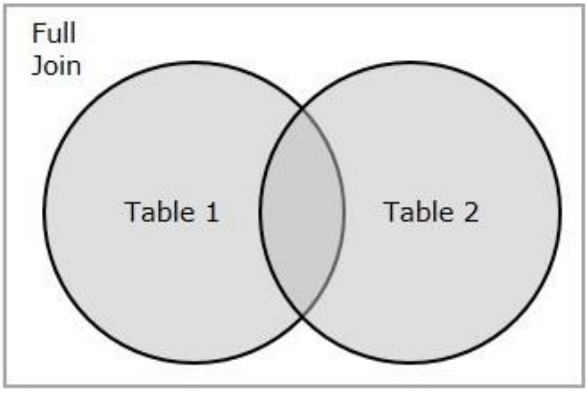
Left (Outer) Join: Retrieves all the records from the first table, Matching records from the second table and NULL values in the unmatched rows.



Right (Outer) Join: Retrieves all the records from the second table, Matching records from the first table and NULL values in the unmatched rows.



Full (Outer) Join: Retrieves records from both the tables and fills the unmatched values with NULL.



Employee table:

```
+--------+--------+-----------+---------+
| emp_id | f_name | l_name    | dept_id |
+--------+--------+-----------+---------+
|      1 | pisati | prathyusha |      10 |
|      2 | pisati | anusha     |      20 |
|      3 | pisati | prathyusha |      30 |
|      4 | chotu  | reddy      |      30 |
+--------+--------+-----------+---------+
```

Department table:

```
+---------+-----------+
| dept_id | dept_name |
+---------+-----------+
|      10 | HR        |
|      20 | sales     |
|      30 | IT        |
|      40 | Marketing |
```

Inner join:

mysql> select * from employee e inner join department d on e.dept_id=d.dept_id;

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

| emp_id | f_name | l_name | dept_id | dept_id | dept_name |

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

| 1 | pisati | prathyusha | 10 | 10 | HR |

| 2 | pisati | anusha | 20 | 20 | sales |

| 3 | pisati | prathyusha | 30 | 30 | IT |

| 4 | chotu | reddy | 30 | 30 | IT |

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

Left Outer join:

mysql> select * from employee e left outer join department d on e.dept_id=d.dept_id;

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

| emp_id | f_name | l_name | dept_id | dept_id | dept_name |

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

| 1 | pisati | prathyusha | 10 | 10 | HR |

| 2 | pisati | anusha | 20 | 20 | sales |

| 3 | pisati | prathyusha | 30 | 30 | IT |

| 4 | chotu | reddy | 30 | 30 | IT |

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

Right outer join:

mysql> select * from employee e right outer join department d on e.dept_id=d.dept_id;

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

| emp_id | f_name | l_name | dept_id | dept_id | dept_name |

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

| 1 | pisati | prathyusha | 10 | 10 | HR |

| 2 | pisati | anusha | 20 | 20 | sales |

| 4 | chotu | reddy | 30 | 30 | IT |

| 3 | pisati | prathyusha | 30 | 30 | IT |

| NULL | NULL | NULL | NULL | 40 | Marketing |

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

Full outer join:

Note:Not all SQL databases support FULL OUTER JOIN

Directly.For databases that do not support it,we can use UNION of LEFT JOIN and RIGHT JOIN to achieve the same result.

mysql> select * from employee e left outer join department d on e.dept_id=d.dept_id union select * from employee e right outer join department d on e.dept_id=d.dept_id;

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

| emp_id | f_name | l_name | dept_id | dept_id | dept_name |

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

| 1 | pisati | prathyusha | 10 | 10 | HR |

| 2 | pisati | anusha | 20 | 20 | sales |

| 3 | pisati | prathyusha | 30 | 30 | IT |

| 4 | chotu | reddy | 30 | 30 | IT |

| NULL | NULL | NULL | NULL | 40 | Marketing |

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

3)Write a query to find duplicate records:

Employees table:

```
+--------+--------+-----------+-----------------+
| emp_id | f_name | l_name    | email           |
+--------+--------+-----------+-----------------+
|      1 | pisati | prathyusha | pisati@gmail.com |
|      2 | pisati | anusha    | anu@gmail.com    |
|      3 | pisati | prathyusha | pisati@gmail.com |
|      4 | chotu  | reddy     | chotu@gmail.com  |
+--------+--------+-----------+-----------------+
```

1)Based on first name:

```
mysql> SELECT f_name,count(*) from employees e group by f_name having count(*)>1;
+--------+----------+
| f_name | count(*) |
+--------+----------+
| pisati |        3 |
+--------+----------+
```

2)Based on email:

mysql> SELECT email,count(*) from employees e group by email having count(*)>1;

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

| email | count(*) |

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

| pisati@gmail.com | 2 |

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

3)Based on first name and last name:

mysql> SELECT f_name,l_name,count(*) from employees e group by f_name,l_name having count(*)>1;

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

| f_name | l_name | count(*) |

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

| pisati | prathyusha | 2 |

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

4)Based on first name and email:

mysql> SELECT f_name,email,count(*) from employees e group by f_name,email having count(*)>1;

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

| f_name | email | count(*) |

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

| pisati | pisati@gmail.com | 2 |

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