Create the given table below.

Table name : Sales

Fields

sales_id int PRIMARY KEY
book_name
book_price
book_amount

Execute the following query.

```
INSERT INTO sales VALUES(NULL, 'DBMS', 1000.00 , 1);
INSERT INTO sales VALUES(NULL, 'DBMS', 1200.00 , 3);
INSERT INTO sales VALUES(NULL, 'PIT', 600.00 , 4);
INSERT INTO sales VALUES(NULL, 'DBMS', 1500.00 , 2);
INSERT INTO sales VALUES(NULL, 'PIT', 900.00 , 2);
INSERT INTO sales VALUES(NULL, 'JAVA', 2000.00 , 3);
```

Update

```
UPDATE table_name
SET column_name1 = new-value1,
    column_name2=new-value2, ...
[WHERE Clause]
```

- Activity 01

Update book name JAVA into MIT.

```
UPDATE sales
SET book_name = 'MIT'
WHERE book_name = 'JAVA';
```

ORDER BY Clause

to sort the records in ascending or descending order

```
SELECT column1, column2,                                    ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

- Activity 02

```
SELECT * FROM Sales
ORDER BY book_name;
```

- Activity 03

```
SELECT * FROM Sales
ORDER BY book_name DESC;
```

GROUP BY Clause

to collect data from multiple records and group the result by one or more column. It is generally used in a SELECT statement.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

- Activity 04

```
SELECT COUNT(sales_id),book_name
FROM Sales
GROUP BY book_name;
```

- Activity 05

```
SELECT book_price, COUNT(*)
FROM   Sales
GROUP BY book_price;
```

- Activity 06

Write SQL query to find the average book price of the books form the table "Sales".

Having Clause

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
```

```
HAVING condition
ORDER BY column_name(s);
```

List the number of books in for each book name. Only include book name with more than 3 sales:

- Activity 07

```
SELECT COUNT(sales_id),book_name
FROM Sales
GROUP BY book_name
HAVING COUNT(sales_id) > 3;
```

- Activity 08

```
SELECT COUNT(sales_id),book_name
FROM Sales
GROUP BY book_name
HAVING COUNT(sales_id)> 3
ORDER BY COUNT(CustomerID) DESC;
```

MySQL Joins

There are three types of MySQL joins:

- MySQL INNER JOIN (or sometimes called simple join)
- MySQL LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

Order table

OrderID	CustomerID	OrderData
1111	5	1995-09-18
1112	4	1995-10-11
1113	1	2000-01-02
1114	2	2000-03-14
1115	3	1992-12-05

Customer table

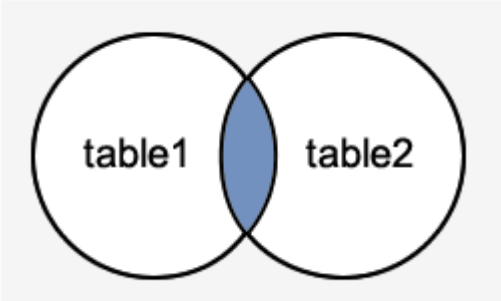
CustomerID	CustomerName	ContactName	City
1	Jayani Bogaha	Isuru Pathirana	Colombo
2	Sandun Isuru	Sanduni Isara	Colombo
3	Piyumi Gamage	Gethmi Gamage	Matara
4	Nethma Samadhi	Samudi Ishara	Kandy

- Activity 09

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

INNER JOIN

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```



- Activity 10

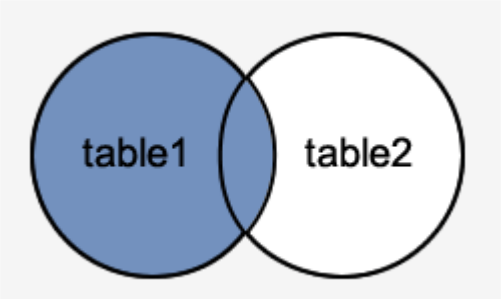
```
SELECT Customers.CustomerName, Order.OrderID
FROM Customers
INNER JOIN Order
ON Customers. CustomerID = Orders.CustomerID;
```

- Activity 11

```
SELECT Customers.CustomerName, Order.OrderID
FROM Customers
INNER JOIN Order
ON Customers. CustomerID = Orders.CustomerID;
GROUP BY OrderData;
```

LEFT JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records (if any) from the right table (table2).



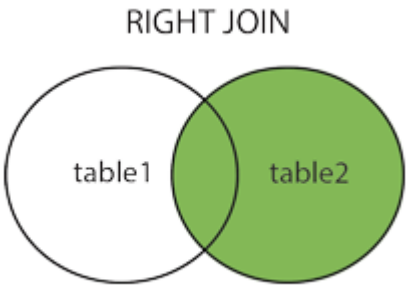

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

- Activity 12

```
SELECT Customers.CustomerName,Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

RIGHT JOIN

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records (if any) from the left table (table1).




```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

"Employees" table

EmployeeID	FirstName	LastName	BirthDate
1	Sanath	Peter	12/8/1995
2	Malshi	Sathsara	13/2/2000
3	Sumudu	Bandara	18/12/1993

- Activity 13

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
FROM Orders
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
ORDER BY Orders.Order
```