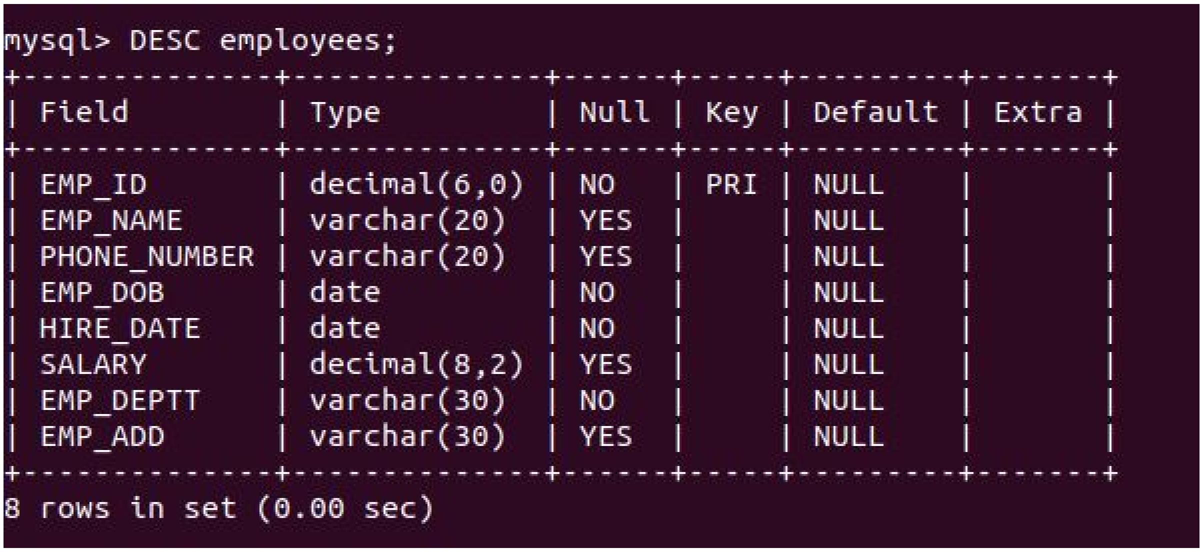
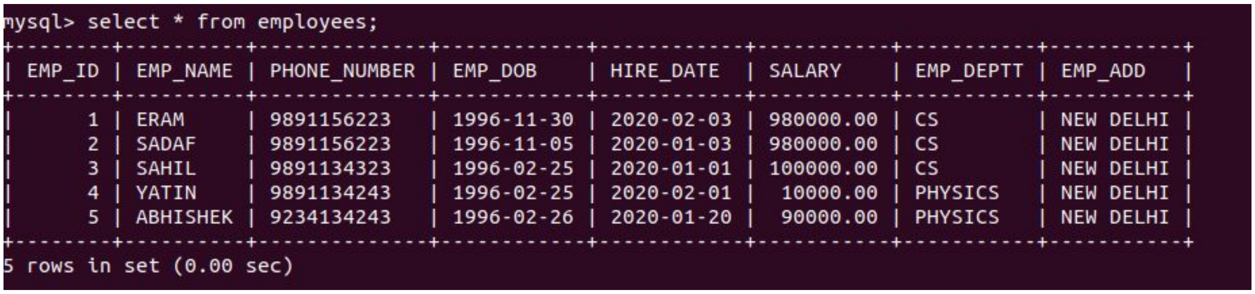
MySQL:

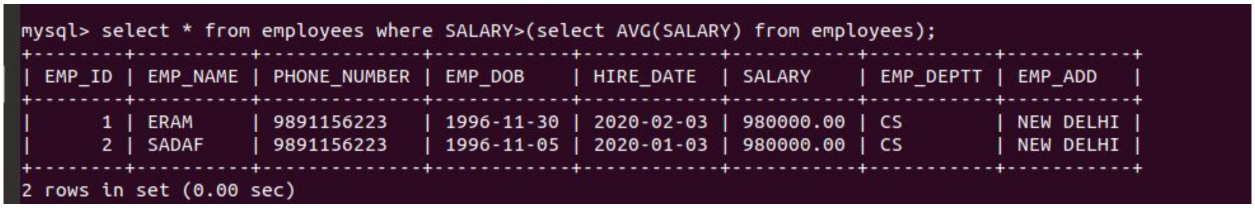


```
mysql> DESC employees;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| EMP_ID       | decimal(6,0) | NO   | PRI | NULL    |       |
| EMP_NAME     | varchar(20)  | YES  |     | NULL    |       |
| PHONE_NUMBER | varchar(20)  | YES  |     | NULL    |       |
| EMP_DOB      | date         | NO   |     | NULL    |       |
| HIRE_DATE    | date         | NO   |     | NULL    |       |
| SALARY       | decimal(8,2) | YES  |     | NULL    |       |
| EMP_DEPTT    | varchar(30)  | NO   |     | NULL    |       |
| EMP_ADD      | varchar(30)  | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```



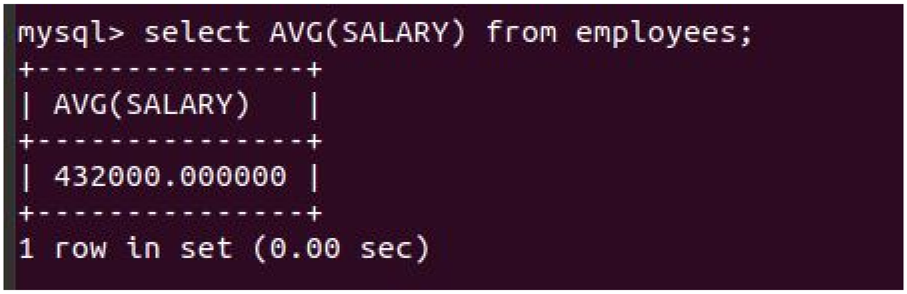
```
mysql> select * from employees;
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
| EMP_ID | EMP_NAME | PHONE_NUMBER | EMP_DOB    | HIRE_DATE  | SALARY    | EMP_DEPTT | EMP_ADD   |
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
|      1 | ERAM     | 9891156223   | 1996-11-30 | 2020-02-03 | 980000.00 | CS        | NEW DELHI |
|      2 | SADAF    | 9891156223   | 1996-11-05 | 2020-01-03 | 980000.00 | CS        | NEW DELHI |
|      3 | SAHIL    | 9891134323   | 1996-02-25 | 2020-01-01 | 100000.00 | CS        | NEW DELHI |
|      4 | YATIN    | 9891134243   | 1996-02-25 | 2020-02-01 |  10000.00 | PHYSICS   | NEW DELHI |
|      5 | ABHISHEK | 9234134243   | 1996-02-26 | 2020-01-20 |  90000.00 | PHYSICS   | NEW DELHI |
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
5 rows in set (0.00 sec)
```

1. Fetch employee's Name, address, Phone, Dob, Department, salary whose salary is greater than the average salary.



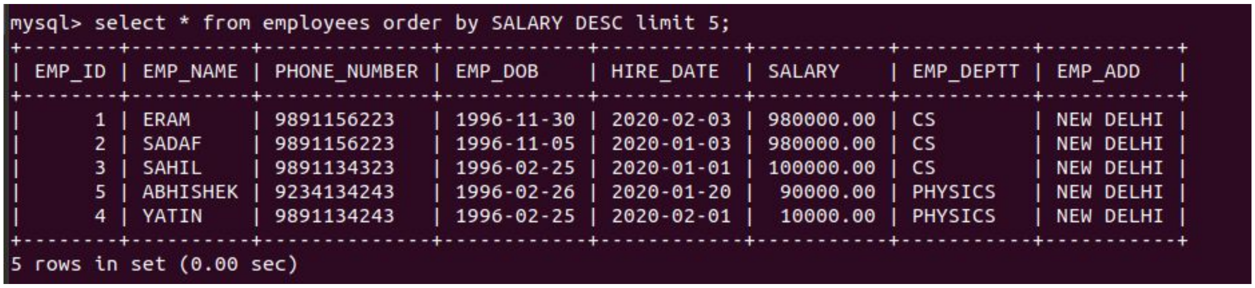
```
mysql> select * from employees where SALARY>(select AVG(SALARY) from employees);
+--------+----------+--------------+------------+------------+-----------+-----------+------------+
| EMP_ID | EMP_NAME | PHONE_NUMBER | EMP_DOB    | HIRE_DATE  | SALARY    | EMP_DEPTT | EMP_ADD    |
+--------+----------+--------------+------------+------------+-----------+-----------+------------+
|      1 | ERAM     | 9891156223   | 1996-11-30 | 2020-02-03 | 980000.00 | CS        | NEW DELHI  |
|      2 | SADAF    | 9891156223   | 1996-11-05 | 2020-01-03 | 980000.00 | CS        | NEW DELHI  |
+--------+----------+--------------+------------+------------+-----------+-----------+------------+
2 rows in set (0.00 sec)
```

2. Fetch Paid average salary for the current month.



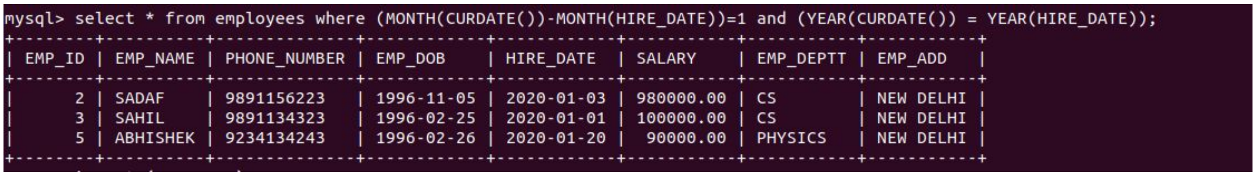
```
mysql> select AVG(SALARY) from employees;
+---------------+
| AVG(SALARY)   |
+---------------+
| 432000.000000 |
+---------------+
1 row in set (0.00 sec)
```

3. Fetch the top 5 Employee's Name, Phone, Salary month, Paid salary for the current month



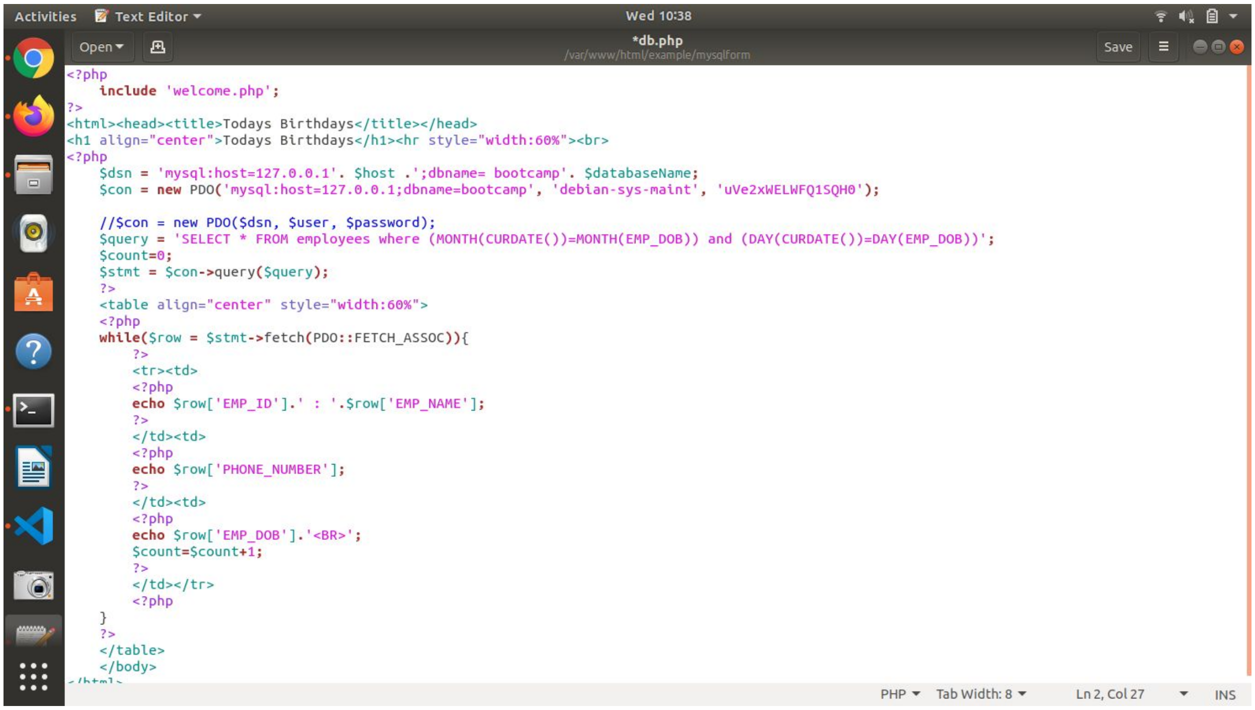
```
mysql> select * from employees order by SALARY DESC limit 5;
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
| EMP_ID | EMP_NAME | PHONE_NUMBER | EMP_DOB    | HIRE_DATE  | SALARY    | EMP_DEPTT | EMP_ADD   |
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
|      1 | ERAM     | 9891156223   | 1996-11-30 | 2020-02-03 | 980000.00 | CS        | NEW DELHI |
|      2 | SADAF    | 9891156223   | 1996-11-05 | 2020-01-03 | 980000.00 | CS        | NEW DELHI |
|      3 | SAHIL    | 9891134323   | 1996-02-25 | 2020-01-01 | 100000.00 | CS        | NEW DELHI |
|      5 | ABHISHEK | 9234134243   | 1996-02-26 | 2020-01-20 |  90000.00 | PHYSICS   | NEW DELHI |
|      4 | YATIN    | 9891134243   | 1996-02-25 | 2020-02-01 |  10000.00 | PHYSICS   | NEW DELHI |
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
5 rows in set (0.00 sec)
```

4. Fetch the employee's name, phone, dob, department name, salary who join last month.



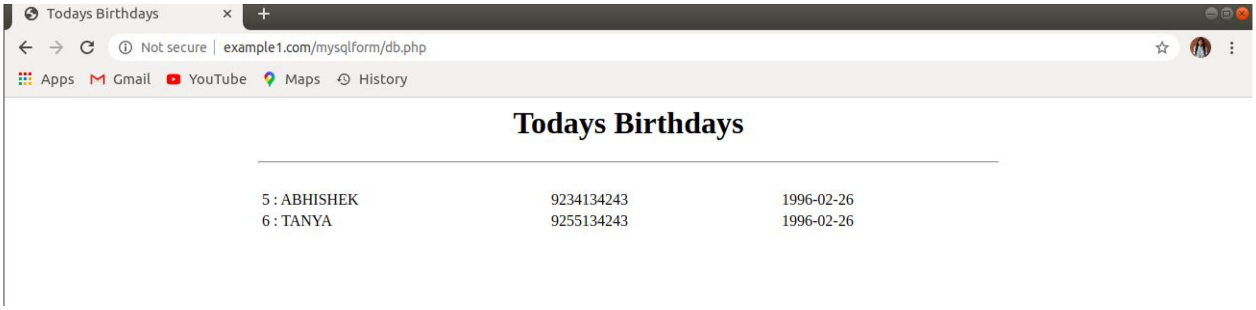
```
mysql> select * from employees where (MONTH(CURDATE())-MONTH(HIRE_DATE))=1 and (YEAR(CURDATE()) = YEAR(HIRE_DATE));
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
| EMP_ID | EMP_NAME | PHONE_NUMBER | EMP_DOB    | HIRE_DATE  | SALARY    | EMP_DEPTT | EMP_ADD   |
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
|      2 | SADAF    | 9891156223   | 1996-11-05 | 2020-01-03 | 980000.00 | CS        | NEW DELHI |
|      3 | SAHIL    | 9891134323   | 1996-02-25 | 2020-01-01 | 100000.00 | CS        | NEW DELHI |
|      5 | ABHISHEK | 9234134243   | 1996-02-26 | 2020-01-20 |  90000.00 | PHYSICS   | NEW DELHI |
+--------+----------+--------------+------------+------------+-----------+-----------+-----------+
```

5. Create a Program in PHP that will fetch and show employee details whose DOB is today?



```
<?php
    include 'welcome.php';
?>
<html><head><title>Todays Birthdays</title></head>
<h1 align="center">Todays Birthdays</h1><hr style="width:60%"><br>
<?php
    $dsn = 'mysql:host=127.0.0.1'. $host .';dbname= bootcamp'. $databaseName;
    $con = new PDO('mysql:host=127.0.0.1;dbname=bootcamp', 'debian-sys-maint', 'uVe2xWELWFQ1SQH0');

    //$con = new PDO($dsn, $user, $password);
    $query = 'SELECT * FROM employees where (MONTH(CURDATE())=MONTH(EMP_DOB)) and (DAY(CURDATE())=DAY(EMP_DOB))';
    $count=0;
    $stmt = $con->query($query);
    ?>
    <table align="center" style="width:60%">
    <?php
    while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
        ?>
        <tr><td>
        <?php
        echo $row['EMP_ID'].' : '.$row['EMP_NAME'];
        ?>
        </td><td>
        <?php
        echo $row['PHONE_NUMBER'];
        ?>
        </td><td>
        <?php
        echo $row['EMP_DOB'].'<BR>';
        $count=$count+1;
        ?>
        </td></tr>
        <?php
    }
    ?>
    </table>
    </body>
</html>
```



Todays Birthdays

```
5 : ABHISHEK          9234134243          1996-02-26
6 : TANYA             9255134243          1996-02-26
```

7. What are the advantage & disadvantage of Index?

Advantages of MySQL Indexes

1- Indexes make search queries much faster.

2- Indexes like primary key index and unique index help to avoid duplicate row data.

3- Full-text indexes in MySQL, users have the opportunity to optimize searching against even large amounts of text located in any field indexed as such.

Disadvantages of MySQL indexes

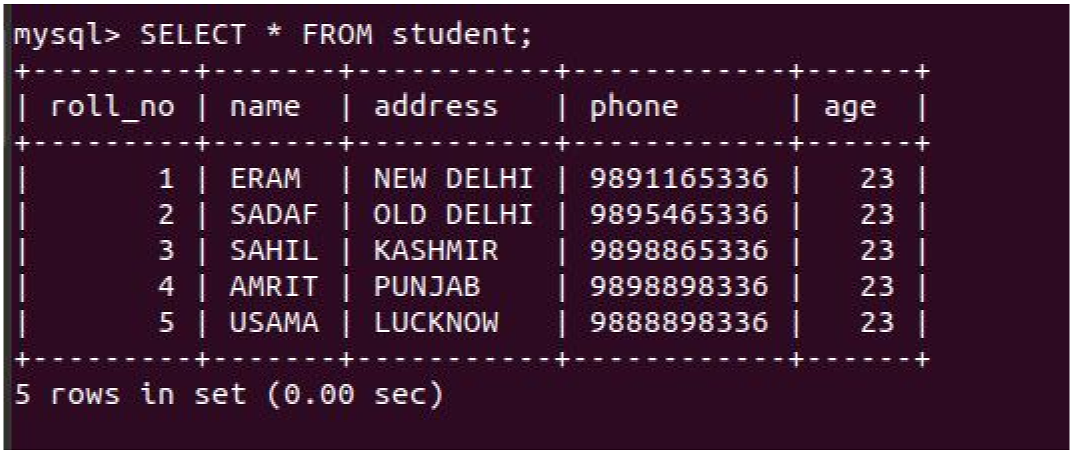
Actually a separate file created when a new index created on the table column. that file stored only the field you're interested in sorting on. So when we create index, it takes up disk space. but because of creating index on every column in every possible combination, the index file would grow much more quickly than the data file. In the case when a table is of large table size, the index file could reach the operating system's maximum file size.

The index also slow down the speed of writing queries, such as INSERT, UPDATE and DELETE. AS MySQL has to internally maintain the pointers to the inserted rows in the actual data file, so there is a performance price to pay in case of above said writing queries because every time a record is changed, the indexes must be updated.

So Indexes are important to speed in large MySQL databases. it doesn't matter how small your table, a 100000-row table scan will never be fast. So If you have a site with a 100000-row table, you should really spend time analyzing possible indexes and possibly consider rewriting queries to optimize your application.

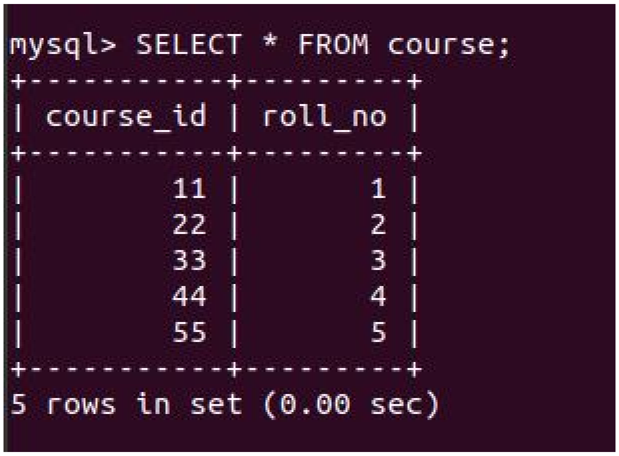
8. Create two tables in MYSQL that shows working of INNER, LEFT, RIGHT.

STUDENT TABLE



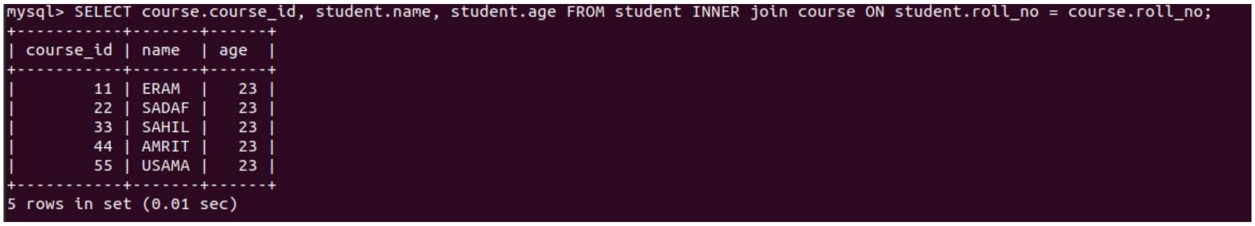
```
mysql> SELECT * FROM student;
+---------+-------+-----------+------------+------+
| roll_no | name  | address   | phone      | age  |
+---------+-------+-----------+------------+------+
|       1 | ERAM  | NEW DELHI | 9891165336 |   23 |
|       2 | SADAF | OLD DELHI | 9895465336 |   23 |
|       3 | SAHIL | KASHMIR   | 9898865336 |   23 |
|       4 | AMRIT | PUNJAB    | 9898898336 |   23 |
|       5 | USAMA | LUCKNOW   | 9888898336 |   23 |
+---------+-------+-----------+------------+------+
5 rows in set (0.00 sec)
```

COURSE TABLE



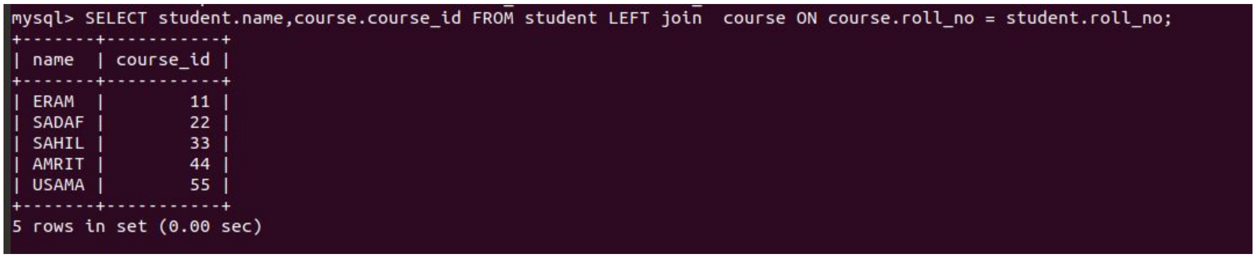
```
mysql> SELECT * FROM course;
+-----------+---------+
| course_id | roll_no |
+-----------+---------+
|        11 |       1 |
|        22 |       2 |
|        33 |       3 |
|        44 |       4 |
|        55 |       5 |
+-----------+---------+
5 rows in set (0.00 sec)
```

1- INNER JOIN



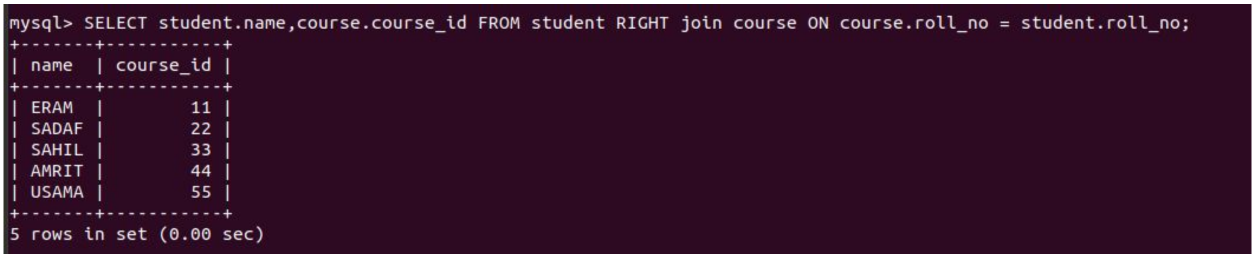
```
mysql> SELECT course.course_id, student.name, student.age FROM student INNER join course ON student.roll_no = course.roll_no;
+-----------+-------+------+
| course_id | name  | age  |
+-----------+-------+------+
|        11 | ERAM  |   23 |
|        22 | SADAF |   23 |
|        33 | SAHIL |   23 |
|        44 | AMRIT |   23 |
|        55 | USAMA |   23 |
+-----------+-------+------+
5 rows in set (0.01 sec)
```

2- LEFT JOIN



```
mysql> SELECT student.name,course.course_id FROM student LEFT join  course ON course.roll_no = student.roll_no;
+-------+-----------+
| name  | course_id |
+-------+-----------+
| ERAM  |        11 |
| SADAF |        22 |
| SAHIL |        33 |
| AMRIT |        44 |
| USAMA |        55 |
+-------+-----------+
5 rows in set (0.00 sec)
```

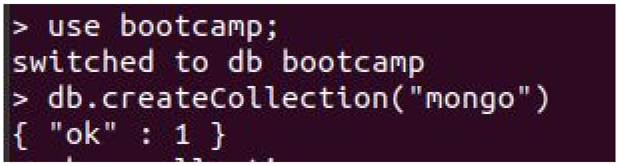
3- RIGHT JOIN



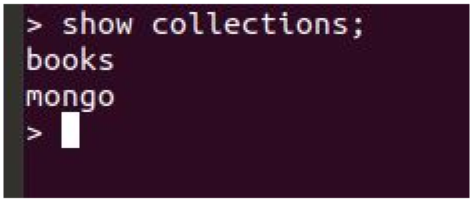
```
mysql> SELECT student.name,course.course_id FROM student RIGHT join course ON course.roll_no = student.roll_no;
+-------+-----------+
| name  | course_id |
+-------+-----------+
| ERAM  |        11 |
| SADAF |        22 |
| SAHIL |        33 |
| AMRIT |        44 |
| USAMA |        55 |
+-------+-----------+
5 rows in set (0.00 sec)
```

MONGO:

9. create a database "Bootcamp" with table "mongo".



```
> use bootcamp;
switched to db bootcamp
> db.createCollection("mongo")
{ "ok" : 1 }
```



```
> show collections;
books
mongo
>
```

10. Write a PHP program to perform Create, Read, Update and Delete Operations.