TITLE:- REFERENTIAL INTEGRITY

THEORY:-

Referential integrity is a database constraint that ensures the relationship between two tables remain consistent by enforcing foreign key constraint. In this lab, we are going to explore referential integrity in action.

1.) Creating two tables:-

a) Employees

```
create table employees (
    id int primary key,
    name varchar (50),
    dept_id int
);
```

b) Department

```
create table department (
    did int primary key,
    dname varchar (50)
);
```

2.) Adding referential integrity:-

Referential integrity can be added by adding update and delete rules on the foreign key column.

For example:- a) Cascade

```
alter table employees add foreign key (dept_id)
references department (did)
on delete cascade
on update cascade;
```

=> There are three types of rules i.e. Cascade, Set null, and no action.

3.) Inserting data:-

a) Into department table:-

```
insert into department values
(10,'IT'),
(20, 'HR'),
(30, 'INFRA');
```

b) Into employees table:-

```
insert into employees values
(101, 'Anurag', 10),
(102, 'pranaya', 20),
(103, 'Hina', 30);
```

4.) Testing:-

a) Update:-

```
update department set did=100 where did=10;
```

→ Updates Anurag's dept-id column with 100 as we have set 'cascade' rule

b) Delete:-

~~dete~~ delete from department
where did=100;

→ deletes 'IT' department and 'Anurag' row from employees table.

For set null

Step-1: delete employees table

→ drop table employees;

step-2: Truncate department table

→ truncate table department;

Step-3: Insert data into department table as in 3.ba

Step-4: Create table 'employees' as in 1.a and insert data as in 3.b

Step-5: Add referentral integrity set null

```
-> alter table employees add foreign key (dept-id)
   references department (did)
   on delete set null
   on update set null;
```

Step-6: Testing

-> Update:-

~~Udpade~~

```
update department set did=100 where did=10;
```

- Sets the value of employee table having dept-id 10 to null

-> delete:-

```
delete from department where did=200;
```

- Sets the value of employee table having dept-id 20 to null.

For no action

i) Repeat step-1 to 4 from above

ii) Add no action referentral integrity

```
-> alter table employees add foreign key (dept-id)
   references department (did)
   on delete no action
   on update no action;
```

iii) Testing

-> Update:-

- update deportment set did = 300 where did = 30;

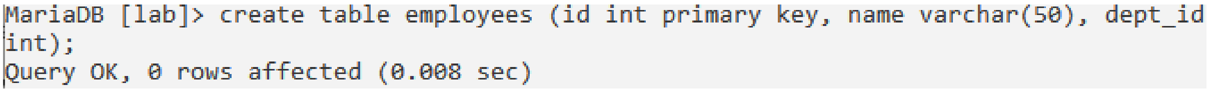
-> Delete:-

- delete from department where did = 30;

Both of these actions give error and the value is not updated.

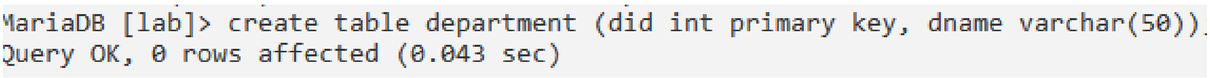
CONCLUSION:-

In this lab, we have explored the way we can implement referential integrity and examined the behaviour of different rules, i.e. no action, cascade and set null.



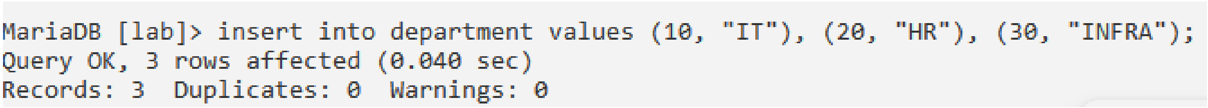
```
MariaDB [lab]> create table employees (id int primary key, name varchar(50), dept_id
int);
Query OK, 0 rows affected (0.008 sec)
```

Figure 1 create employee table



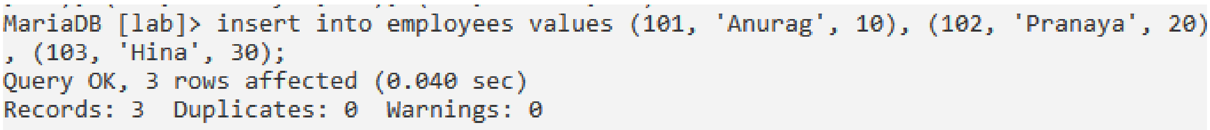
```
MariaDB [lab]> create table department (did int primary key, dname varchar(50));
Query OK, 0 rows affected (0.043 sec)
```

Figure 2 create department table



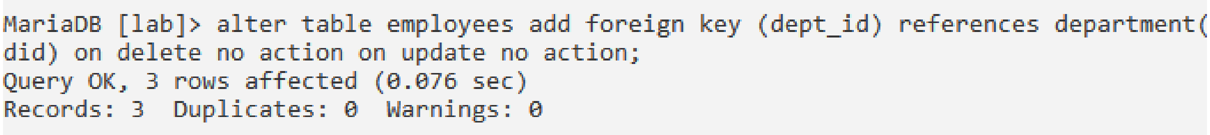
```
MariaDB [lab]> insert into department values (10, "IT"), (20, "HR"), (30, "INFRA");
Query OK, 3 rows affected (0.040 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Figure 3 insert values into department table



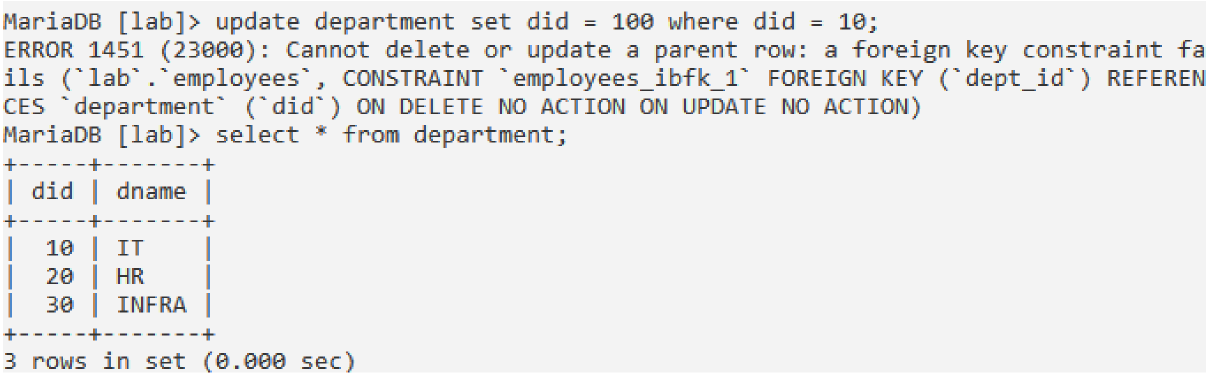
```
MariaDB [lab]> insert into employees values (101, 'Anurag', 10), (102, 'Pranaya', 20)
, (103, 'Hina', 30);
Query OK, 3 rows affected (0.040 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Figure 4 insert values into employees table



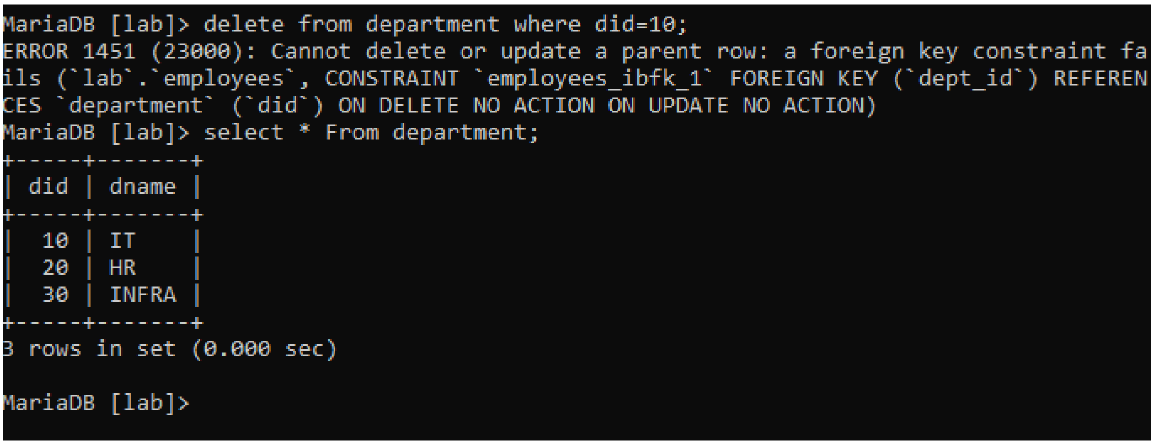
```
MariaDB [lab]> alter table employees add foreign key (dept_id) references department(
did) on delete no action on update no action;
Query OK, 3 rows affected (0.076 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Figure 5 add foreign key with no action rule



```
MariaDB [lab]> update department set did = 100 where did = 10;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fa
ils (`lab`.`employees`, CONSTRAINT `employees_ibfk_1` FOREIGN KEY (`dept_id`) REFEREN
CES `department` (`did`) ON DELETE NO ACTION ON UPDATE NO ACTION)
MariaDB [lab]> select * from department;
+-----+-------+
| did | dname |
+-----+-------+
|  10 | IT    |
|  20 | HR    |
|  30 | INFRA |
+-----+-------+
3 rows in set (0.000 sec)
```

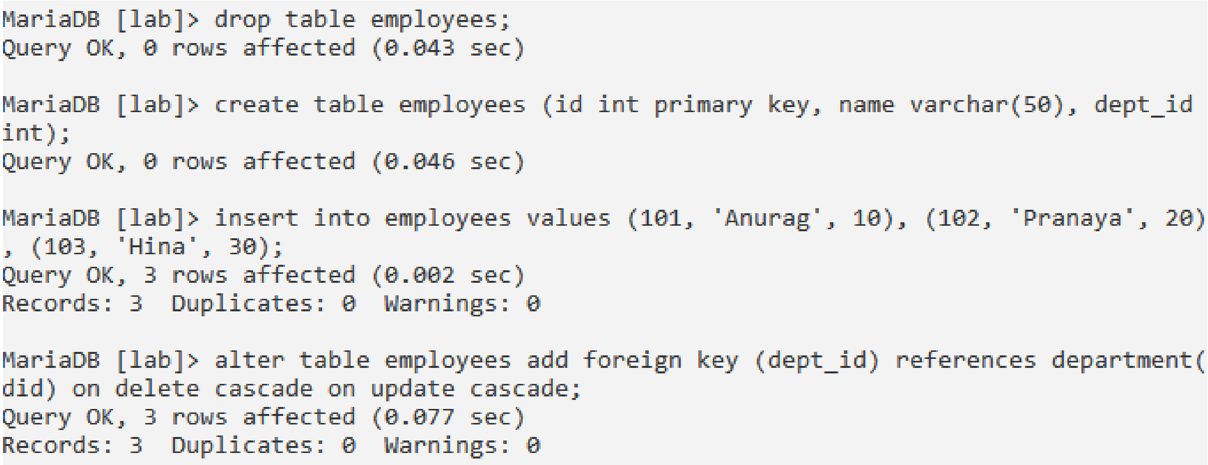
Figure 6 we couldn't update the data



```
MariaDB [lab]> delete from department where did=10;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fa
ils (`lab`.`employees`, CONSTRAINT `employees_ibfk_1` FOREIGN KEY (`dept_id`) REFEREN
CES `department` (`did`) ON DELETE NO ACTION ON UPDATE NO ACTION)
MariaDB [lab]> select * From department;
+-----+-------+
| did | dname |
+-----+-------+
|  10 | IT    |
|  20 | HR    |
|  30 | INFRA |
+-----+-------+
3 rows in set (0.000 sec)

MariaDB [lab]>
```

Figure 7 we couldn't delete the data



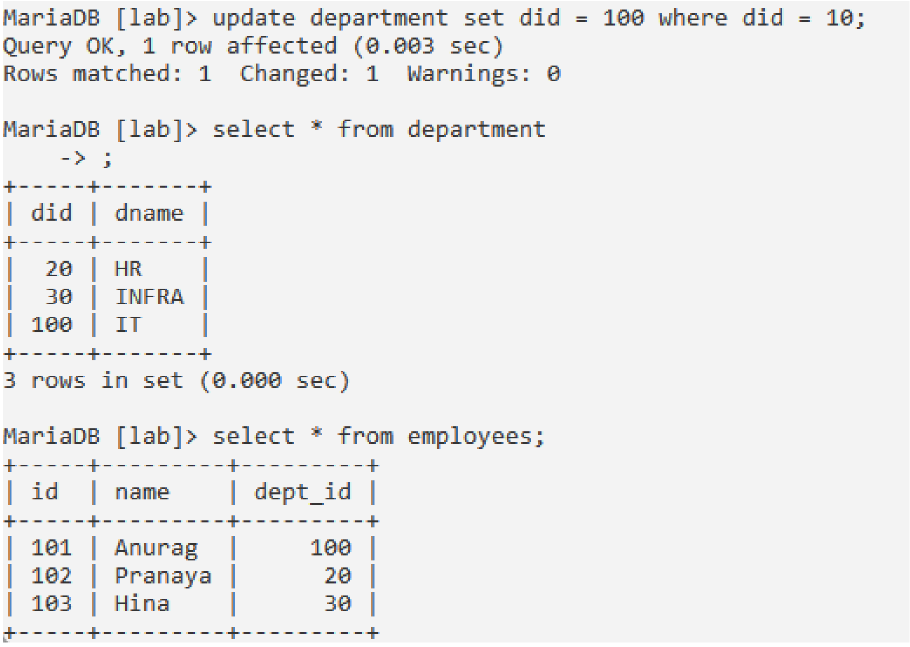
```
MariaDB [lab]> drop table employees;
Query OK, 0 rows affected (0.043 sec)

MariaDB [lab]> create table employees (id int primary key, name varchar(50), dept_id
int);
Query OK, 0 rows affected (0.046 sec)

MariaDB [lab]> insert into employees values (101, 'Anurag', 10), (102, 'Pranaya', 20)
, (103, 'Hina', 30);
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [lab]> alter table employees add foreign key (dept_id) references department(
did) on delete cascade on update cascade;
Query OK, 3 rows affected (0.077 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Figure 8 drop employee table, re-create it, and add foreign key with cascade rule

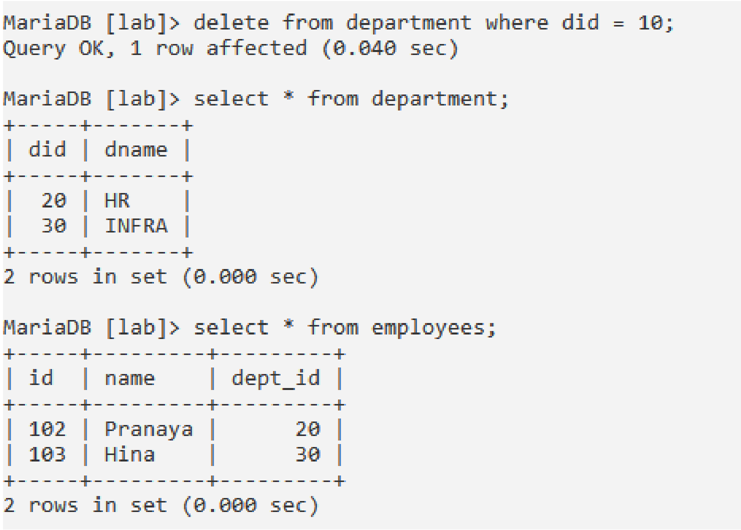


```
MariaDB [lab]> update department set did = 100 where did = 10;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [lab]> select * from department
    -> ;
+-----+-------+
| did | dname |
+-----+-------+
|  20 | HR    |
|  30 | INFRA |
| 100 | IT    |
+-----+-------+
3 rows in set (0.000 sec)

MariaDB [lab]> select * from employees;
+-----+---------+---------+
| id  | name    | dept_id |
+-----+---------+---------+
| 101 | Anurag  |     100 |
| 102 | Pranaya |      20 |
| 103 | Hina    |      30 |
+-----+---------+---------+
```

Figure 9 updating department table also updates employee table

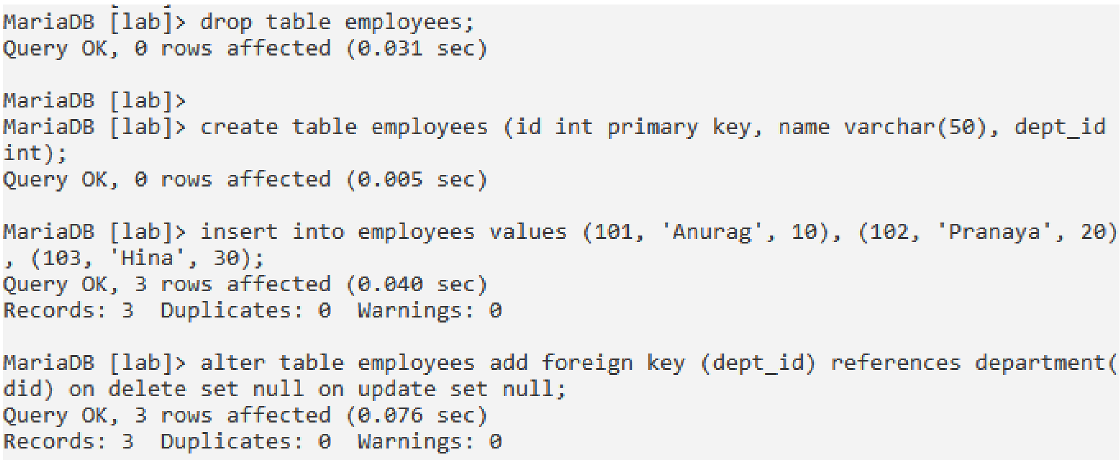


```
MariaDB [lab]> delete from department where did = 10;
Query OK, 1 row affected (0.040 sec)

MariaDB [lab]> select * from department;
+-----+-------+
| did | dname |
+-----+-------+
|  20 | HR    |
|  30 | INFRA |
+-----+-------+
2 rows in set (0.000 sec)

MariaDB [lab]> select * from employees;
+-----+---------+---------+
| id  | name    | dept_id |
+-----+---------+---------+
| 102 | Pranaya |      20 |
| 103 | Hina    |      30 |
+-----+---------+---------+
2 rows in set (0.000 sec)
```

Figure 10 deleting an item from department also deletes from employee table



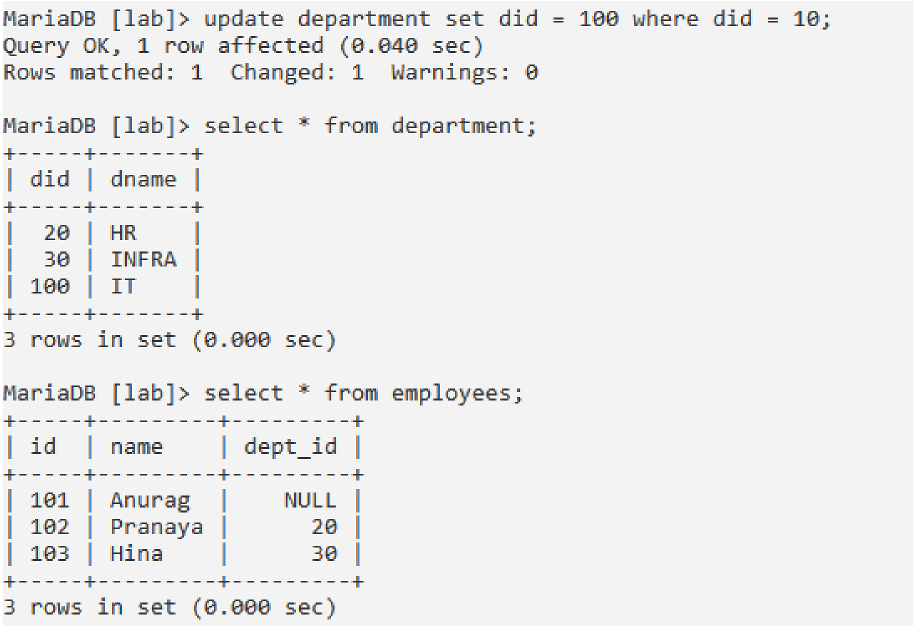
```
MariaDB [lab]> drop table employees;
Query OK, 0 rows affected (0.031 sec)

MariaDB [lab]>
MariaDB [lab]> create table employees (id int primary key, name varchar(50), dept_id
int);
Query OK, 0 rows affected (0.005 sec)

MariaDB [lab]> insert into employees values (101, 'Anurag', 10), (102, 'Pranaya', 20)
, (103, 'Hina', 30);
Query OK, 3 rows affected (0.040 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [lab]> alter table employees add foreign key (dept_id) references department(
did) on delete set null on update set null;
Query OK, 3 rows affected (0.076 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Figure 11 adding foreign key with set null type

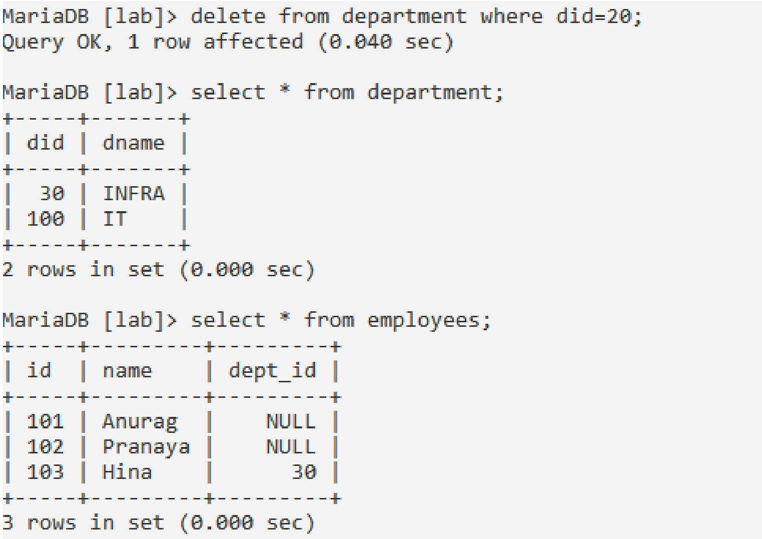


```
MariaDB [lab]> update department set did = 100 where did = 10;
Query OK, 1 row affected (0.040 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [lab]> select * from department;
+-----+-------+
| did | dname |
+-----+-------+
|  20 | HR    |
|  30 | INFRA |
| 100 | IT    |
+-----+-------+
3 rows in set (0.000 sec)

MariaDB [lab]> select * from employees;
+-----+---------+---------+
| id  | name    | dept_id |
+-----+---------+---------+
| 101 | Anurag  |    NULL |
| 102 | Pranaya |      20 |
| 103 | Hina    |      30 |
+-----+---------+---------+
3 rows in set (0.000 sec)
```

Figure 12 updating a record in department table sets null in the foreign key field in employee table



```
MariaDB [lab]> delete from department where did=20;
Query OK, 1 row affected (0.040 sec)

MariaDB [lab]> select * from department;
+-----+-------+
| did | dname |
+-----+-------+
|  30 | INFRA |
| 100 | IT    |
+-----+-------+
2 rows in set (0.000 sec)

MariaDB [lab]> select * from employees;
+-----+---------+---------+
| id  | name    | dept_id |
+-----+---------+---------+
| 101 | Anurag  |    NULL |
| 102 | Pranaya |    NULL |
| 103 | Hina    |      30 |
+-----+---------+---------+
3 rows in set (0.000 sec)
```

Figure 13 deleting a record in department table sets null in the foreign key field in employee table