Deerwalk Institute Of Technology
Advance Database Management System

Lab: 4

Submitted By:
Name: Sagar Giri
Roll No. 205

Submitted To:
Rishikesh Katuwal

Date:

1. maintain a table which tracks the total count of records of another table.

- Let’s create two tables (**student** and **studentAudit**)
- **student** table keeps record of student and **studentAudit** table keeps record of no of entries in student table.

-- Table structure for table `student`

```
CREATE TABLE `student` (
  `id` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
  `address` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Table structure for table `studentAudit`

```
CREATE TABLE `studentAudit` (
  `record_count` int(11) NOT NULL,
  `time_stamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

– Insert some dummy data in the `student` table

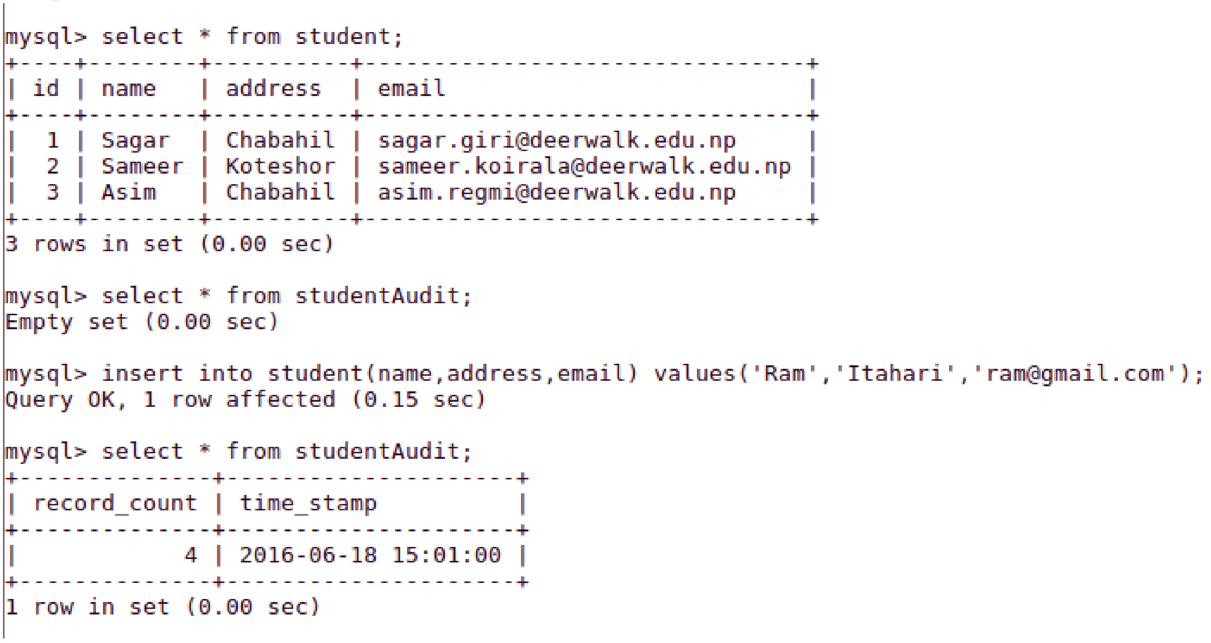
```
INSERT INTO `student` (`id`, `name`, `address`, `email`) VALUES
(1, 'Sagar', 'Chabahil', 'sagar.giri@deerwalk.edu.np'),
(2, 'Sameer', 'Koteshor', 'sameer.koirala@deerwalk.edu.np'),
(3, 'Asim', 'Chabahil', 'asim.regmi@deerwalk.edu.np'),
```

Now, create a trigger that inserts record count of `student` table into `studentAudit` table while inserting any record in the `student` table.

```
DELIMITER $$
CREATE TRIGGER `audit_record`
AFTER INSERT ON `student`
FOR EACH ROW
INSERT INTO studentAudit(record_count) SELECT COUNT(*) FROM `student`
$$
DELIMITER ;
```

Now whenever we insert values in `student` table, `studentAudit` table gets updated.

Output:



```
mysql> select * from student;
+----+--------+----------+------------------------------+
| id | name   | address  | email                        |
+----+--------+----------+------------------------------+
|  1 | Sagar  | Chabahil | sagar.giri@deerwalk.edu.np   |
|  2 | Sameer | Koteshor | sameer.koirala@deerwalk.edu.np |
|  3 | Asim   | Chabahil | asim.regmi@deerwalk.edu.np   |
+----+--------+----------+------------------------------+
3 rows in set (0.00 sec)

mysql> select * from studentAudit;
Empty set (0.00 sec)

mysql> insert into student(name,address,email) values('Ram','Itahari','ram@gmail.com');
Query OK, 1 row affected (0.15 sec)

mysql> select * from studentAudit;
+--------------+---------------------+
| record_count | time_stamp          |
+--------------+---------------------+
|            4 | 2016-06-18 15:01:00 |
+--------------+---------------------+
1 row in set (0.00 sec)
```

2. Create a trigger which doesn't allow to update numeric values smaller than older values.

First, create a dummy table:

```
CREATE TABLE `employee` (
  `id` int(11) NOT NULL,
  `salary` int(11) NOT NULL,
  `name` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

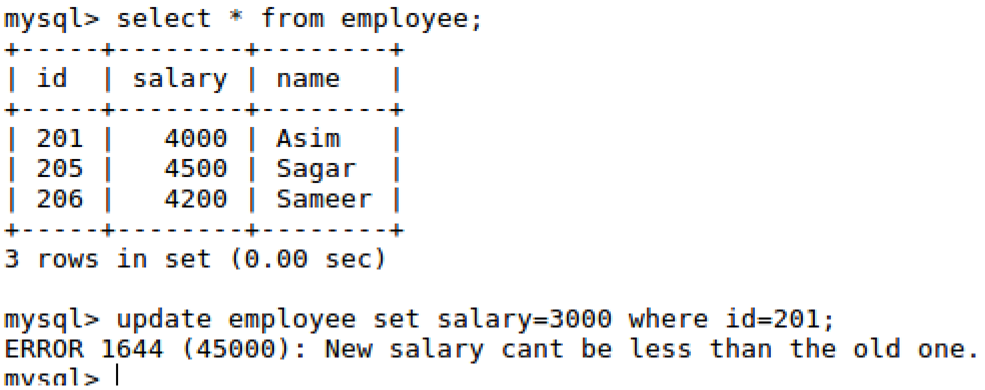
Now, insert some dummy value:

```
insert into employee values(205, 4500, 'Sagar');
insert into employee values(206, 4200, 'Sameer');
insert into employee values(201, 4000, 'Asim');
```

Create a trigger that doesn’t allow user to update salary less than the old one:

```
delimiter //
drop trigger if exists salary_update;
create trigger salary_update
before update on employee
FOR EACH ROW
      BEGIN
            IF(new.salary < old.salary)
            THEN
                  signal sqlstate '45000'
                  set message_text = 'New salary cant be less than the old one.';
            END IF;
      END //
delimiter ;
```

Check output:



```
mysql> select * from employee;
+-----+--------+--------+
| id  | salary | name   |
+-----+--------+--------+
| 201 |   4000 | Asim   |
| 205 |   4500 | Sagar  |
| 206 |   4200 | Sameer |
+-----+--------+--------+
3 rows in set (0.00 sec)

mysql> update employee set salary=3000 where id=201;
ERROR 1644 (45000): New salary cant be less than the old one.
mysql> |
```