# COMS 4111 - Introduction to Databases
# Extra Credit Assignment 1

Test 1

```
mysql> SELECT * FROM employee;
Empty set (0.00 sec)

mysql> SELECT * FROM user_prefix_count;
Empty set (0.00 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Candie', 'MacPake', 'cmacpake6@spiegel.de');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Carling', 'Bockman', 'cbockman5@cam.ac.uk');
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Chrissy', 'Weatherup', 'cweatherup2@theguardian.com');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Cloris', 'Edards', 'cedards4@springer.com');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Dianne', 'Aickin', 'daickin9@domainmarket.com');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Dougy', 'Snoxill', 'dsnoxill1@scribd.com');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Dulciana', 'Cambell', 'dcambell8@i2i.jp');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Jared', 'Melarkey', 'jmelarkey3@google.co.jp');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Joice', 'Tomsen', 'jtomsen7@slideshare.net');
Query OK, 1 row affected (0.15 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('Juana', 'Gatesman', 'jgatesman0@chron.com');
Query OK, 1 row affected (0.08 sec)

mysql> SELECT * FROM `eca1`.`employee`;
+------------+-----------+----------------------------+---------------------+---------------------+
| first_name | last_name | email                      | created             | user_id             |
+------------+-----------+----------------------------+---------------------+---------------------+
| Candie     | MacPake   | cmacpake6@spiegel.de       | 2017-11-13 17:00:51 | CANDIE.MACPAKE.1    |
| Carling    | Bockman   | cbockman5@cam.ac.uk        | 2017-11-13 17:00:59 | CARLING.BOCKMAN.1   |
| Chrissy    | Weatherup | cweatherup2@theguardian.com | 2017-11-13 17:01:10 | CHRISSY.WEATHERUP.1 |
| Cloris     | Edards    | cedards4@springer.com      | 2017-11-13 17:01:16 | CLORIS.EDARDS.1     |
| Dianne     | Aickin    | daickin9@domainmarket.com  | 2017-11-13 17:01:22 | DIANNE.AICKIN.1     |
| Dougy      | Snoxill   | dsnoxill1@scribd.com       | 2017-11-13 17:01:30 | DOUGY.SNOXILL.1     |
| Dulciana   | Cambell   | dcambell8@i2i.jp           | 2017-11-13 17:01:37 | DULCIANA.CAMBELL.1  |
| Jared      | Melarkey  | jmelarkey3@google.co.jp    | 2017-11-13 17:01:44 | JARED.MELARKEY.1    |
| Joice      | Tomsen    | jtomsen7@slideshare.net    | 2017-11-13 17:01:49 | JOICE.TOMSEN.1      |
| Juana      | Gatesman  | jgatesman0@chron.com       | 2017-11-13 17:01:55 | JUANA.GATESMAN.1    |
+------------+-----------+----------------------------+---------------------+---------------------+
10 rows in set (0.00 sec)
```

The first 2 SELECTs are executed to show that there is no data in both the tables initially. Following the SELECTs, are the INSERTs corresponding to the data given in the test statement. The final SELECT query shows the inserted tuples. The tuples demonstrate the correctness of the `Before_INSERT` trigger which generates the `user_id` for a person using the `first_name` and the `last_name`.

Test 2a

```
mysql> SET @time = Now();
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @time;
+---------------------+
| @time               |
+---------------------+
| 2017-11-13 17:05:41 |
+---------------------+
1 row in set (0.00 sec)

mysql> UPDATE `eca1`.`employee` SET `first_name`='Douglas', `created` = NOW() WHERE user_id = 'DOUGY.SNOXILL.1';
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM `eca1`.`employee` WHERE user_id = 'DOUGY.SNOXILL.1';
+------------+-----------+----------------------+---------------------+-----------------+
| first_name | last_name | email                | created             | user_id         |
+------------+-----------+----------------------+---------------------+-----------------+
| Douglas    | Snoxill   | dsnoxill1@scribd.com | 2017-11-13 17:01:30 | DOUGY.SNOXILL.1 |
+------------+-----------+----------------------+---------------------+-----------------+
1 row in set (0.00 sec)
```

(@time variable is set to provide reference for the time when the following update query is executed) The UPDATE query doesn't update the `created` property. Only the `first_name` field is updated and any update to `created` field is ignored. This query demonstrates the correctness of the `Before_UPDATE` trigger.

Test 2b

```
mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Gatesman', 'jgatesman0@chron.com');
ERROR 1062 (23000): Duplicate entry 'jgatesman0@chron.com' for key 'email_UNIQUE'
mysql> SELECT * FROM `eca1`.`employee`;
+------------+-----------+----------------------------+---------------------+---------------------+
| first_name | last_name | email                      | created             | user_id             |
+------------+-----------+----------------------------+---------------------+---------------------+
| Candie     | MacPake   | cmacpake6@spiegel.de       | 2017-11-13 17:00:51 | CANDIE.MACPAKE.1    |
| Carling    | Bockman   | cbockman5@cam.ac.uk        | 2017-11-13 17:00:59 | CARLING.BOCKMAN.1   |
| Chrissy    | Weatherup | cweatherup2@theguardian.com | 2017-11-13 17:01:10 | CHRISSY.WEATHERUP.1 |
| Cloris     | Edards    | cedards4@springer.com      | 2017-11-13 17:01:16 | CLORIS.EDARDS.1     |
| Dianne     | Aickin    | daickin9@domainmarket.com  | 2017-11-13 17:01:22 | DIANNE.AICKIN.1     |
| Douglas    | Snoxill   | dsnoxill1@scribd.com       | 2017-11-13 17:01:30 | DOUGY.SNOXILL.1     |
| Dulciana   | Cambell   | dcambell8@i2i.jp           | 2017-11-13 17:01:37 | DULCIANA.CAMBELL.1  |
| Jared      | Melarkey  | jmelarkey3@google.co.jp    | 2017-11-13 17:01:44 | JARED.MELARKEY.1    |
| Joice      | Tomsen    | jtomsen7@slideshare.net    | 2017-11-13 17:01:49 | JOICE.TOMSEN.1      |
| Juana      | Gatesman  | jgatesman0@chron.com       | 2017-11-13 17:01:55 | JUANA.GATESMAN.1    |
+------------+-----------+----------------------------+---------------------+---------------------+
10 rows in set (0.00 sec)
```

The INSERT query tries to insert a new tuple ('John', 'Gatesman') with the same `email` as the tuple('Juana','Gatesman'). The query fails because the `UNIQUE` constraint is enforced on the `email` field.

Test 3

```
mysql> UPDATE `eca1`.`employee` SET user_id='DOGE.SNOXILL.729' WHERE user_id='DOUGY.SNOXILL.1';
ERROR 1643 (02222): Can't change user_id. It is immutable!
mysql> UPDATE `eca1`.`employee` SET user_id='DOGE.SNOXILL.729' WHERE first_name='Douglas' and last_name='Snoxill';
ERROR 1643 (02222): Can't change user_id. It is immutable!
mysql> SELECT * FROM `eca1`.`employee`;
+------------+-----------+---------------------------+---------------------+--------------------+
| first_name | last_name | email                     | created             | user_id            |
+------------+-----------+---------------------------+---------------------+--------------------+
| Candie     | MacPake   | cmacpake6@spiegel.de      | 2017-11-13 17:00:51 | CANDIE.MACPAKE.1   |
| Carling    | Bockman   | cbockman5@cam.ac.uk       | 2017-11-13 17:00:59 | CARLING.BOCKMAN.1  |
| Chrissy    | Weatherup | cweatherup2@theguardian.com | 2017-11-13 17:01:10 | CHRISSY.WEATHERUP.1 |
| Cloris     | Edards    | cedards4@springer.com     | 2017-11-13 17:01:16 | CLORIS.EDARDS.1    |
| Dianne     | Aickin    | daickin9@domainmarket.com | 2017-11-13 17:01:22 | DIANNE.AICKIN.1    |
| Douglas    | Snoxill   | dsnoxill1@scribd.com      | 2017-11-13 17:01:30 | DOUGY.SNOXILL.1    |
| Dulciana   | Cambell   | dcambell8@i2i.jp          | 2017-11-13 17:01:37 | DULCIANA.CAMBELL.1 |
| Jared      | Melarkey  | jmelarkey3@google.co.jp   | 2017-11-13 17:01:44 | JARED.MELARKEY.1   |
| Joice      | Tomsen    | jtomsen7@slideshare.net   | 2017-11-13 17:01:49 | JOICE.TOMSEN.1     |
| Juana      | Gatesman  | jgatesman0@chron.com      | 2017-11-13 17:01:55 | JUANA.GATESMAN.1   |
+------------+-----------+---------------------------+---------------------+--------------------+
10 rows in set (0.00 sec)
```

Both the UPDATE queries try to update the `user_id` of the tuple(`Douglas`,`Snoxill`). The queries fail because `user_id` is meant to be immutable because the `max_value` count doesn't change on deletion. Thus, each tuple gets a unique `user_id`, generated at the time of its insertion, that won't change throughout its life, hence, the manual update raises an error.

Test 5

```
mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Smith', 'js1@statcounter.com');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Smith', 'js2@statcounter.com');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Smith', 'js3@statcounter.com');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Smith', 'js4@statcounter.com');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Smith', 'js5@statcounter.com');
Query OK, 1 row affected (0.08 sec)

mysql> DELETE FROM `eca1`.`employee` WHERE `email`='js2@statcounter.com';
Query OK, 1 row affected (0.08 sec)

mysql> DELETE FROM `eca1`.`employee` WHERE `email`='js4@statcounter.com';
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `eca1`.`employee` (`first_name`, `last_name`, `email`) VALUES ('John', 'Smith', 'js@myemail.com');
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM `eca1`.`employee` WHERE `last_name`='Smith';
+------------+-----------+---------------------+---------------------+-------------+
| first_name | last_name | email               | created             | user_id     |
+------------+-----------+---------------------+---------------------+-------------+
| John       | Smith     | js1@statcounter.com | 2017-11-13 17:11:44 | JOHN.SMITH.1 |
| John       | Smith     | js3@statcounter.com | 2017-11-13 17:11:53 | JOHN.SMITH.3 |
| John       | Smith     | js5@statcounter.com | 2017-11-13 17:12:06 | JOHN.SMITH.5 |
| John       | Smith     | js@myemail.com      | 2017-11-13 17:12:29 | JOHN.SMITH.6 |
+------------+-----------+---------------------+---------------------+-------------+
4 rows in set (0.00 sec)
```

The INSERTs and DELETES correspond to the records given in the test statement. The final SELECT shows that the state of the data is consistent with the expected state of the data.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ END ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~