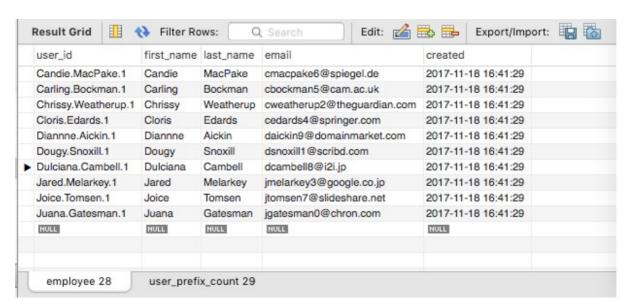
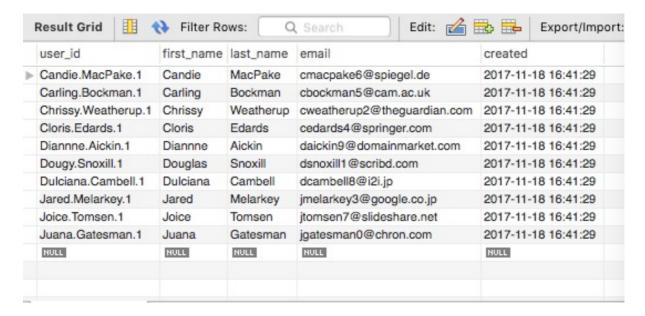
Huibo Zhao hz2480 COMS4111 SEC003 EC1 Nov.18th.2017

The ddl and test sql codes are appended at the end of this document.

Employee table after Initial load & Test 1:



Employee table after Test 2:



Result after Test 3:

		Time	Action	Response	Duration / Fetch Time
0	180	16:54:09	select * from employee LIMIT 0, 1000	10 row(s) returned	0.00043 sec / 0.000
0	181	16:54:09	select * from user_prefix_count LIMIT 0, 1000	10 row(s) returned	0.00080 sec / 0.0001
0	182	16:59:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.00069 sec
3	183	16:59:07	INSERT INTO employee (first_name,last_name,email) VALUES ('Jared','M	Error Code: 1062. Duplicate entry 'jmelarkey3@googl	0.0049 sec

Full text: 16:59:07 INSERT INTO employee (first_name,last_name,email) VALUES ('Jared','Melarkey','jmelarkey3@google.co.jp') Error Code: 1062. Duplicate entry 'jmelarkey3@google.co.jp' for key 'email' 0.0049 sec

Result after Test 4:

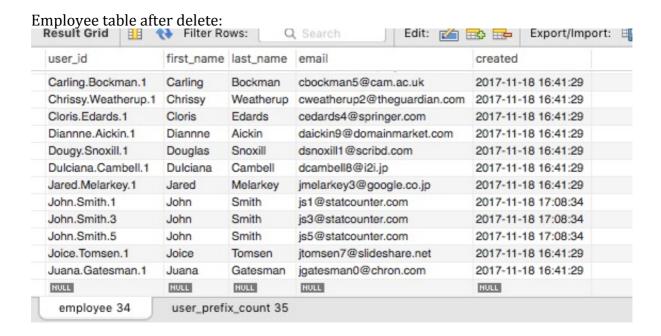
		Time	Action	Response	Duration / Fetch Time
0	182	16:59:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.00069 sec
8	183	16:59:07	INSERT INTO employee (first_name,last_name,email) VALUES ('Jared','M	Error Code: 1062. Duplicate entry 'jmelarkey3@googl	0.0049 sec
0	184	17:04:04	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.00064 sec
8	185	17:04:04	UPDATE employee SET user_id = 'foo' where employee.email = 'dsnoxill1	Error Code: 1644. Cannot change user_id	0.0017 sec

Full text: 17:04:04 UPDATE employee SET user_id = 'foo' where employee.email = 'dsnoxill1@scribd.com' Error Code: 1644. Cannot change user_id 0.0017 sec

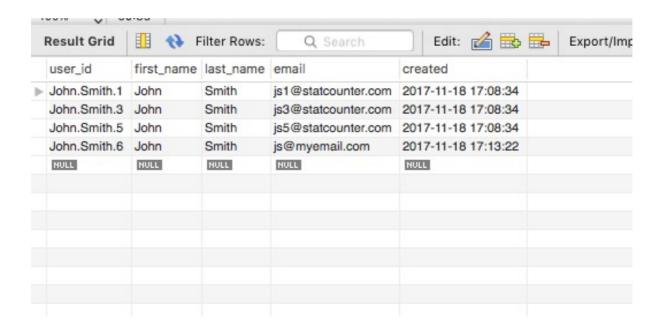
Test 5:

Employee table after first insert (5 John Smith):

user_id	first_name	last_name	email	created
Cloris.Edards.1	Cloris	Edards	cedards4@springer.com	2017-11-18 16:41:29
Diannne.Aickin.1	Diannne	Aickin	daickin9@domainmarket.com	2017-11-18 16:41:29
Dougy.Snoxill.1	Douglas	Snoxill	dsnoxill1@scribd.com	2017-11-18 16:41:29
Dulciana.Cambell.1	Dulciana	Cambell	dcambell8@i2i.jp	2017-11-18 16:41:29
Jared.Melarkey.1	Jared	Melarkey	jmelarkey3@google.co.jp	2017-11-18 16:41:29
John.Smith.1	John	Smith	js1@statcounter.com	2017-11-18 17:08:34
John.Smith.2	John	Smith	js2@statcounter.com	2017-11-18 17:08:34
John.Smith.3	John	Smith	js3@statcounter.com	2017-11-18 17:08:34
John.Smith.4	John	Smith	js4@statcounter.com	2017-11-18 17:08:34
John.Smith.5	John	Smith	js5@statcounter.com	2017-11-18 17:08:34
Joice.Tomsen.1	Joice	Tomsen	jtomsen7@slideshare.net	2017-11-18 16:41:29
Juana.Gatesman.1	Juana	Gatesman	jgatesman0@chron.com	2017-11-18 16:41:29
NULL	NULL	NULL	NULL	NULL



Result for select statement after second insert (Single John Smith):



Appendix:

DDL:

```
-- Table structure for table `Employee`
DROP TABLE IF EXISTS 'Employee';
CREATE TABLE 'Employee' (
`user_id` varchar(32) NOT NULL,
`first_name` varchar(32) NOT NULL,
`last_name` varchar(32) NOT NULL,
'email' varchar(32) NOT NULL,
 'created' datetime,
 PRIMARY KEY ('user_id'),
UNIQUE ('email')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Table structure for table 'user_prefix_count'
DROP TABLE IF EXISTS 'user prefix count';
CREATE TABLE `user_prefix_count` (
 'prefix' varchar(32) NOT NULL,
'max value' int(12) NOT NULL,
PRIMARY KEY ('prefix')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
# trigger before insert into 'employee'
DROP TRIGGER IF EXISTS 'trigger_insert_employee';
DELIMITER $$
CREATE TRIGGER 'trigger_insert_employee'
 BEFORE INSERT ON 'employee' FOR EACH ROW
BEGIN
                          `prefix_name`
      DECLARE
                                                           VARCHAR(32);
 DECLARE
             `existed`
                              int(1);
 DECLARE `count`
                              int(12);
 SET `prefix_name` =
                          CONCAT(New.`first_name`, '.', New.`last_name`);
 # check if `prefix_name` existed or not
```

```
SET 'existed' = (select exists (select * from 'user_prefix_count' where 'prefix' =
`prefix_name`));
 # if existed, we increment its max_value field
 # if not existed, we insert a new tuple to the `user_prefix_count` table
      IF 'existed' = 1 THEN
             SET `count` = (select `max_value` from `user_prefix_count` where `prefix`
= `prefix_name`) + 1;
    UPDATE `user_prefix_count` SET `max_value` = `count` where
`user_prefix_count`.`prefix` = `prefix_name`;
      ELSE
             SET `count` = 1;
             INSERT INTO 'user_prefix_count' (prefix,max_value) VALUES
('prefix_name',1);
      END IF;
 SET New.`user_id` =
                          CONCAT(New.`first_name`, '.', New.`last_name`,'.',`count`);
      SET New.`created` = CURRENT_TIMESTAMP;
END $$
DELIMITER;
# trigger before update on 'employee'
DROP TRIGGER IF EXISTS 'trigger_update_employee';
DELIMITER $$
CREATE TRIGGER `trigger_update_employee`
 BEFORE UPDATE ON 'employee' FOR EACH ROW
BEGIN
      SET NEW.`created` = Old.`created`;
      IF NOT NEW.`user_id` = Old.`user_id` THEN
             SIGNAL SQLSTATE '45001'
                    SET MESSAGE_TEXT = 'Cannot change user_id';
      end if;
END $$
DELIMITER;
```

Test Codes:

```
SET SQL SAFE UPDATES = 0;
INSERT INTO employee (first name, last name, email)
     VALUES ('Candie', 'MacPake', 'cmacpake6@spiegel.de');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Carling', 'Bockman', 'cbockman5@cam.ac.uk');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Chrissy','Weatherup','cweatherup2@theguardian.com');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Cloris', 'Edards', 'cedards4@springer.com');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Diannne','Aickin','daickin9@domainmarket.com');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Dougy', 'Snoxill', 'dsnoxill1@scribd.com');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Dulciana', 'Cambell', 'dcambell8@i2i.jp');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Jared','Melarkey','jmelarkey3@google.co.jp');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Joice', 'Tomsen', 'jtomsen7@slideshare.net');
INSERT INTO employee (first_name,last_name,email)
     VALUES ('Juana', 'Gatesman', 'jgatesman0@chron.com');
*/
#########
UPDATE employee SET first name = 'Douglas', 'created' = CURRENT TIMESTAMP
     where employee.first_name = 'Dougy' and employee.last_name = 'Snoxill';
#########
```

```
INSERT INTO employee (first_name,last_name,email)
    VALUES ('Jared','Melarkey','jmelarkey3@google.co.jp');
*/
#########
# UPDATE employee SET user id = 'foo' where employee.email =
'dsnoxill1@scribd.com';
########
INSERT INTO employee (first_name,last_name,email)
    VALUES ('John','Smith','js1@statcounter.com');
INSERT INTO employee (first name, last name, email)
    VALUES ('John', 'Smith', 'js2@statcounter.com');
INSERT INTO employee (first name,last name,email)
    VALUES ('John', 'Smith', 'js3@statcounter.com');
INSERT INTO employee (first_name,last_name,email)
    VALUES ('John', 'Smith', 'js4@statcounter.com');
INSERT INTO employee (first_name,last_name,email)
    VALUES ('John', 'Smith', 'js5@statcounter.com');
DELETE FROM employee WHERE 'email'='js2@statcounter.com';
DELETE FROM employee WHERE 'email'='js4@statcounter.com';
*/
INSERT INTO employee (first_name,last_name,email)
    VALUES ('John', 'Smith', 'js@myemail.com');
select * from employee where last_name = 'Smith';
#########
select * from employee;
select * from user prefix count;
```