CPSC 5021 In-Class Exercise (Stored Procedure)

Question 1:

Create a stored procedure to

(1) count the order in a specific order status such as shipped, resolved, cancelled, on hold, disputed or in process (the value of order status is passed into the procedure when the procedure is invoked)

(2) change all of the orders in the status "in process" to "shipped".

The structure and data of the table "orders" is shown below.

orderNumber	status
1	shipped
2	resolved
3	shipped
4	cancelled
5	in process
6	in process

Invoke the procedure to check whether it works.

Sol:

```
create table orders
(orderNumber int,
status varchar(20));


insert into orders
values (1, 'shipped'),
(2, 'resolved'),
(3, 'shipped'),
(4, 'cancelled'),
(5, 'in process'),
(6, 'in process');
```

```
DELIMITER |
CREATE PROCEDURE countOrderByStatus(
IN orderStatus VARCHAR(20),OUT total INT)
BEGIN
SELECT count(orderNumber)
INTO total
FROM orders
WHERE status = orderStatus;
UPDATE orders
SET status = 'shipped'
where status = 'in process';
END
|
```

```
set @status = 'shipped';
CALL countOrderByStatus(@status,@total);
select @status, @total;
```

Question 2:

Create a stored procedure to insert a student record (including ID, lastName, firstName, email, classYear, and major) only if classYear < 2015. The structure of the table STUDENT is shown below. You can pick a suitable data type for each attribute.

STUDENT(ID, lastName, firstName, email, classYear, major).

Invoke the stored procedure to check whether it works or not.

```
create table STUDENT
(ID char(11),
lastName varchar(20),
firstName varchar(20),
email varchar(50),
classYear char(4),
major varchar(15));
```

```
delimiter |
CREATE PROCEDURE insertStudents(id varchar(11), lname
varchar(20), fname varchar(20), email varchar(50), classyear
char(4), major varchar(15))
BEGIN
IF classyear < 2015 THEN
   INSERT INTO STUDENT
   VALUES (id, lname, fname, email,classyear,major);
END IF;
END
|
```

```
call insertStudents('11111111112', 'Amy', 'Han', 'jdoe@usna.edu', 2016, 'FIN');
```

Question 3:

Suppose you want to create a large number of dataset for a table and your table structure looks like this:

```
CREATE TABLE IF NOT EXISTS dictionary (
  id int(11) AUTO_INCREMENT,
  word varchar(100) NOT NULL,
  meaning varchar(300) NOT NULL,
  PRIMARY KEY (id)
);
```

Now you have to add 100 dummy data ("a", "a meaning") to this table. Create a stored procedure to finish the task.

Sol:

```
delimiter |

CREATE PROCEDURE sampleProc()

BEGIN

 DECLARE x  INT;

 SET x = 1;

 WHILE x  <= 100 DO

        insert into dictionary

  set word = 'a',

        meaning = 'a meaning';

 SET  x = x + 1;

 END WHILE;

END

|
```

```
call SampleProc();
```

Question 4:

Suppose you want to create a large number of dataset for a table and your table structure looks like this:

```
CREATE TABLE dataset (
  id int(11),
  word varchar(100) NOT NULL,
  meaning varchar(300) NOT NULL,
  PRIMARY KEY (id)
);
```

Now you have to add 98 dummy data (1, “a”, “a meaning”), (2, “a”, “a meaning”), … (100, “a”, “a meaning”) to this table. The value of “id” starts from 1 until 100, but 60 and 80 are skipped. The values of “word” and “meaning” are always “a” and “a meaning”.

Create a stored procedure to finish the task.

```
delimiter |
CREATE PROCEDURE loopiterate()
BEGIN
 DECLARE x  INT;
 SET x = 0;
 loop_label: loop
  IF x >= 100 THEN
  LEAVE loop_label;
  END IF;
  SET  x = x + 1;
  IF (x = 60) or (x = 80) THEN
  ITERATE loop_label;
  ELSE
  insert into dataset VALUES(x, 'a','a meaning');
  END IF;
 END loop;
END
|
```

```
call loopiterate ();
```