Yaoxi Luo
Prof. Scott T. Leutenegger
COMP - 3421
Assignment 5
Feb. 22, 2016

******************************************************

Question1:

```
create table Orders(
        Order_id int,
        Cus_id int,
        Emp_id int,
        OrderDate date NOT NULL,
        Primary Key (Order_id, Cus_id, Emp_id),
        Foreign Key (Cus_id) references Customer(Cus_id),
        Foreign Key (Emp_id) references Employee(Emp_id));
```

Output:

```
mysql> insert into Orders values (1,1,1000001,'2015-03-11');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key
constraint fails (`newtoysale`.`orders`, CONSTRAINT `orders_ibfk_2`
FOREIGN KEY (`Emp_id`) REFERENCES `Employee` (`Emp_id`))
```

******************************************************

Question2:

Description:

```
/*
This procedure named oneCusOrders. The user can use it to check one
Customer have how many orders. There is one input parameter, which is
Customer id. Also, there is one output, which is the total number of
the orders for that one customer.
*/

drop procedure if exists oneCusOrders ;

delimiter //
create procedure oneCusOrders (IN inCusID INT, OUT outSum INT)
begin
  select count(*) INTO outSum
  from Orders O
  where O.Cus_id = inCusID ;
end //
delimiter ;

set @OrderSum = 0 ;

select 'CALL oneCusOrders(1234567,@OrderSum):' as ' ' ;
CALL oneCusOrders(1234567, @OrderSum) ;
select @OrderSum ;
```

```
Output:
mysql> source oneCusOrders;
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+---------------------------------------+
|                                       |
+---------------------------------------+
| CALL oneCusOrders(1234567,@OrderSum): |
+---------------------------------------+
1 row in set, 1 warning (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+-----------+
| @OrderSum |
+-----------+
|         2 |
+-----------+
1 row in set (0.00 sec)

******************************************************
Question3:
Description:
/*
The trigger named UnitPriceChange_trigger. This trigger will run when
the user changes the UnitSellingPrice in OrderLine whatever increasing
or decreasing UnitSellingPrice. Then, the trigger will create a table
named log to record the changes.
*/

drop trigger if exists UnitPriceChange_trigger;

DELIMITER $$
create trigger UnitPriceChange_trigger AFTER UPDATE ON OrderLine
FOR EACH ROW
BEGIN
  IF NEW.UnitSellingPrice <> OLD.UnitSellingPrice THEN
    insert into log values (NEW.Invent_id, OLD.UnitSellingPrice,
NEW.UnitSellingPrice) ;
  END IF ;
END; $$
DELIMITER ;


drop table if exists log ;
create table log (
  Invent_id integer,
  oldUnitSellingPrice integer,
  newUnitSellingPrice integer
  );

select 'before update: select count(*) from log:' as '';
select count(*) from log;

select * from OrderLine
where Order_id = 2345688;

update OrderLine
set UnitSellingPrice = UnitSellingPrice * 1.7
where Order_id = 2345688;

select * from OrderLine
where Order_id = 2345688;

select 'after update: select count(*) from log:' as '';
select count(*) from log;
select * from log;

Output:
mysql> drop trigger if exists UnitPrice_trigger;
Query OK, 0 rows affected (0.00 sec)

mysql> source UnitPrice_trigger;
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

+------------------------------------------+
|                                          |
+------------------------------------------+
| before update: select count(*) from log: |
+------------------------------------------+
1 row in set (0.00 sec)

+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)

+----------+-----------+----------+------------------+
| Order_id | Invent_id | Quantity | UnitSellingPrice |
+----------+-----------+----------+------------------+
|  2345688 |     33264 |       12 |               37 |
|  2345688 |    119983 |       16 |               94 |
+----------+-----------+----------+------------------+
2 rows in set (0.00 sec)

Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

+----------+-----------+----------+------------------+
| Order_id | Invent_id | Quantity | UnitSellingPrice |
+----------+-----------+----------+------------------+
|  2345688 |     33264 |       12 |               63 |
|  2345688 |    119983 |       16 |              160 |
+----------+-----------+----------+------------------+
2 rows in set (0.00 sec)

+-----------------------------------------+
|                                         |
+-----------------------------------------+
| after update: select count(*) from log: |
+-----------------------------------------+
1 row in set (0.00 sec)

+----------+
| count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)

+-----------+---------------------+---------------------+
| Invent_id | oldUnitSellingPrice | newUnitSellingPrice |
+-----------+---------------------+---------------------+
|     33264 |                  37 |                  63 |
|    119983 |                  94 |                 160 |
+-----------+---------------------+---------------------+
2 rows in set (0.00 sec)

******************************************************
Question4:
1) a selection on a single relation;
select 'Speed of no index on Quantity in OrderLine:' as ' ' ;
select count(*) from OrderLine where Quantity = 20;

select 'Creating an index on Quantity in OrderLine:' as ' ' ;
create index indexOrderLineQuantity on OrderLine (Quantity) using
btree;
```

```
select 'Speed of Creating an index on Quantity in OrderLine:' as ' ' ;
select count(*) from OrderLine where Quantity = 20;

drop index indexOrderLineQuantity on OrderLine;

Output:
mysql> source indexOrderLineQuantity;
+---------------------------------------------+
|                                             |
+---------------------------------------------+
| Speed of no index on Quantity in OrderLine: |
+---------------------------------------------+
1 row in set, 1 warning (0.00 sec)

+----------+
| count(*) |
+----------+
|   250261 |
+----------+
1 row in set (3.45 sec)

+---------------------------------------------+
|                                             |
+---------------------------------------------+
| Creating an index on Quantity in OrderLine: |
+---------------------------------------------+
1 row in set, 1 warning (0.00 sec)

Query OK, 0 rows affected (8.63 sec)
Records: 0  Duplicates: 0  Warnings: 0

+------------------------------------------------------+
|                                                      |
+------------------------------------------------------+
| Speed of Creating an index on Quantity in OrderLine: |
+------------------------------------------------------+
1 row in set, 1 warning (0.00 sec)

+----------+
| count(*) |
+----------+
|   250261 |
+----------+
1 row in set (0.05 sec)

Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

2) a selection that involves a join.

```
select 'Speed of no index on Quantity in OrderLine:' as ' ' ;
select count(*)
from OrderLine O1, Orders O2
where O1.Order_id = O2.Order_id and O1.Quantity = 20;

select 'Creating an index on Quantity in OrderLine:' as ' ' ;
create index indexOrderLineQuantity on OrderLine (Quantity) using
btree;

select 'Speed of Creating an index on Quantity in OrderLine:' as ' ' ;
select count(*)
from OrderLine O1, Orders O2
where O1.Order_id = O2.Order_id and O1.Quantity = 20;

drop index indexOrderLineQuantity on OrderLine;

Output:
mysql> source indexOrderLineQuantity2;
+---------------------------------------------+
|                                             |
+---------------------------------------------+
| Speed of no index on Quantity in OrderLine: |
+---------------------------------------------+
1 row in set, 1 warning (0.00 sec)

+----------+
| count(*) |
+----------+
|   250261 |
+----------+
1 row in set (16.19 sec)

+---------------------------------------------+
|                                             |
+---------------------------------------------+
| Creating an index on Quantity in OrderLine: |
+---------------------------------------------+
1 row in set, 1 warning (0.00 sec)

Query OK, 0 rows affected (9.58 sec)
Records: 0  Duplicates: 0  Warnings: 0

+------------------------------------------------------+
|                                                      |
+------------------------------------------------------+
| Speed of Creating an index on Quantity in OrderLine: |
+------------------------------------------------------+
1 row in set, 1 warning (0.00 sec)

+----------+
| count(*) |
+----------+
|   250261 |
+----------+
1 row in set (16.58 sec)

Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```