Assignment #2

By: Daria Yip

Question 1

```
mysql> CREATE DATABASE Hollywood;
Query OK, 1 row affected (0.00 sec)

mysql> Use Hollywood;
Database changed
mysql> CREATE TABLE Movie (
    -> mID integer AUTO_INCREMENT primary key, title text, year int, check (year < 2016), director text);
Query OK, 0 rows affected (0.05 sec)

mysql> Describe Movie;
+----------+---------+------+-----+---------+----------------+
| Field    | Type    | Null | Key | Default | Extra          |
+----------+---------+------+-----+---------+----------------+
| mID      | int(11) | NO   | PRI | NULL    | auto_increment |
| title    | text    | YES  |     | NULL    |                |
| year     | int(11) | YES  |     | NULL    |                |
| director | text    | YES  |     | NULL    |                |
+----------+---------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE Reviewer (
    -> rID integer auto_increment primary key, name text);
Query OK, 0 rows affected (0.02 sec)

mysql> Describe Reviewer;
+-------+---------+------+-----+---------+----------------+
| Field | Type    | Null | Key | Default | Extra          |
+-------+---------+------+-----+---------+----------------+
| rID   | int(11) | NO   | PRI | NULL    | auto_increment |
| name  | text    | YES  |     | NULL    |                |
+-------+---------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

mysql> CREATE TABLE Rating (
    -> rID Integer, mID Integer, stars Integer, ratingDate timestamp default now());
Query OK, 0 rows affected (0.02 sec)

mysql> Describe Rating;
+------------+-----------+------+-----+-------------------+-------+
| Field      | Type      | Null | Key | Default           | Extra |
+------------+-----------+------+-----+-------------------+-------+
| rID        | int(11)   | YES  |     | NULL              |       |
| mID        | int(11)   | YES  |     | NULL              |       |
| stars      | int(11)   | YES  |     | NULL              |       |
| ratingDate | timestamp | NO   |     | CURRENT_TIMESTAMP |       |
+------------+-----------+------+-----+-------------------+-------+
4 rows in set (0.00 sec)

mysql> ALTER TABLE Rating ADD CONSTRAINT fk_mID FOREIGN KEY (mID) REFERENCES Movie(mID);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Rating ADD CONSTRAINT fk_rID FOREIGN KEY (rID) REFERENCES Reviewer(rID);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> Describe Rating;
+------------+-----------+------+-----+-------------------+-------+
| Field      | Type      | Null | Key | Default           | Extra |
+------------+-----------+------+-----+-------------------+-------+
| rID        | int(11)   | YES  | MUL | NULL              |       |
| mID        | int(11)   | YES  | MUL | NULL              |       |
| stars      | int(11)   | YES  |     | NULL              |       |
| ratingDate | timestamp | NO   |     | CURRENT_TIMESTAMP |       |
+------------+-----------+------+-----+-------------------+-------+
4 rows in set (0.00 sec)
```

Question 2

1.

```
mysql> UPDATE hiking
    -> SET distance = 10.50, est_time = 10.50
    -> WHERE trail = 'East Mesa Loop';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> SELECT * from hiking;
+----------------+--------------------+----------+----------+
| trail          | area               | distance | est_time |
+----------------+--------------------+----------+----------+
| East Mesa Loop | Cuyamaca Mountains |     10.5 |     10.5 |
+----------------+--------------------+----------+----------+
1 row in set (0.00 sec)

mysql> INSERT INTO hiking(trail, distance) VALUES ('Oak Canyon', 3.00);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM hiking;
+----------------+--------------------+----------+----------+
| trail          | area               | distance | est_time |
+----------------+--------------------+----------+----------+
| East Mesa Loop | Cuyamaca Mountains |     10.5 |     10.5 |
| Oak Canyon     | NULL               |        3 |     NULL |
+----------------+--------------------+----------+----------+
2 rows in set (0.00 sec)
```

2.

```
mysql> UPDATE hiking
    -> SET area = 'Mission Trails Regional Park', est_time = 2
    -> WHERE trail = 'Oak Canyon';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * from hiking;
+---------------+-----------------------------+----------+----------+
| trail         | area                        | distance | est_time |
+---------------+-----------------------------+----------+----------+
| East Mesa Loop | Cuyamaca Mountains          |     10.5 |     10.5 |
| Oak Canyon     | Mission Trails Regional Park |        3 |        2 |
+---------------+-----------------------------+----------+----------+
2 rows in set (0.00 sec)
```

3.

```
mysql> DELETE FROM hiking WHERE distance > 5;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM hiking;
+------------+-----------------------------+----------+----------+
| trail      | area                        | distance | est_time |
+------------+-----------------------------+----------+----------+
| Oak Canyon | Mission Trails Regional Park |        3 |        2 |
+------------+-----------------------------+----------+----------+
1 row in set (0.00 sec)
```

4.

```
mysql> CREATE TABLE rating(trail varchar(50), difficulty int);
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+----------------------------+
| Tables_in_cind110A2Script1 |
+----------------------------+
| hiking                     |
| rating                     |
+----------------------------+
2 rows in set (0.00 sec)
```

5.

```
mysql> ALTER TABLE hiking ADD COLUMN trailID decimal(10, 5) primary key;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

6.

```
mysql> ALTER TABLE rating ADD COLUMN trailID decimal(10, 5);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE rating ADD CONSTRAINT fk_trailID FOREIGN KEY (trailID) REFERENCES hiking(trailID);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe hiking;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| trail    | char(50)      | YES  |     | NULL    |       |
| area     | char(50)      | YES  |     | NULL    |       |
| distance | float         | YES  |     | NULL    |       |
| est_time | float         | YES  |     | NULL    |       |
| trailID  | decimal(10,5) | NO   | PRI | NULL    |       |
+----------+---------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql> describe rating;
+------------+---------------+------+-----+---------+-------+
| Field      | Type          | Null | Key | Default | Extra |
+------------+---------------+------+-----+---------+-------+
| trail      | varchar(50)   | YES  |     | NULL    |       |
| difficulty | int(11)       | YES  |     | NULL    |       |
| trailID    | decimal(10,5) | YES  | MUL | NULL    |       |
+------------+---------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Provision for taking large values including decimals was shown in the data type decimal(10, 5).

7.

```
mysql> DROP TABLE rating;
Query OK, 0 rows affected (0.00 sec)
```

Question 3

1.

```
CREATE DATABASE Question3;
USE Question3;

CREATE TABLE Salesman(salesman_id int primary key, name varchar(50), city varchar(30), commission decimal(3, 2));

CREATE TABLE Customer(customer_id int primary key, cust_name varchar(50), city varchar(30), grade int, salesman_id int)
ALTER TABLE Customer ADD CONSTRAINT fk_salesman_id FOREIGN KEY (salesman_id) REFERENCES Salesman(salesman_id);

CREATE TABLE OrderTable(Order_No int primary key, Purch_Amt decimal(6,2), Ord_Date date, customer_id int, salesman_id int);
ALTER TABLE OrderTable ADD CONSTRAINT fk_customer_id FOREIGN KEY (customer_id) REFERENCES Customer(customer_id);
ALTER TABLE OrderTable ADD CONSTRAINT fk_salesman_id2 FOREIGN KEY (salesman_id) REFERENCES Salesman(salesman_id);
```

#	name	cust_name	city
1	James Hoog	Nick Rimando	New York
2	James Hoog	Brad Davis	New York
3	Nail Knite	Fabian Johns	Paris
4	Pit Alex	Brad Guzan	London
5	Pit Alex	Julian Green	London
6	Mc Lyon	Fabian Johns	Paris

2.

```
SELECT OrderTable.Order_No, OrderTable.Purch_Amt, Customer.cust_name, Customer.city from Customer inner join OrderTable WHERE OrderTable.Purch_Amt between 500 and 2000;
```

#	Order_No	Purch_Amt	cust_name	city
1	70007	948.50	Brad Guzan	London
2	70007	948.50	Nick Rimando	New York
3	70007	948.50	Jozy Altidor	Moscow
4	70007	948.50	Fabian Johns	Paris
5	70007	948.50	Graham Zusi	California
6	70007	948.50	Brad Davis	New York

Result 10

3.

```
SELECT Salesman.name, Customer.cust_name from Customer inner join Salesman on Customer.salesman_id = Salesman.salesman_id;
```

#	name	cust_name
1	Pit Alex	Brad Guzan
2	James Hoog	Nick Rimando
3	Paul Adam	Jozy Altidor
4	Mc Lyon	Fabian Johns
5	Nail Knite	Graham Zusi
6	James Hoog	Brad Davis

4.

```
SELECT Customer.cust_name, Salesman.name, Customer.city, Salesman.city, Salesman.commission from Salesman, Customer WHERE Salesman.commission > 0.12;
```

#	cust_name	name	city	city	commission
1	Brad Guzan	James Hoog	London	New York	0.15
2	Nick Rimando	James Hoog	New York	New York	0.15
3	Jozy Altidor	James Hoog	Moscow	New York	0.15
4	Fabian Johns	James Hoog	Paris	New York	0.15
5	Graham Zusi	James Hoog	California	New York	0.15
6	Brad Davis	James Hoog	New York	New York	0.15

Result 17

5.

```
SELECT Customer.cust_name, Salesman.name, Customer.city, Salesman.city, Salesman.commission from Salesman, Customer WHERE Salesman.commission > 0.12 And
NOT Customer.city = Salesman.city;
```

#	cust_name	name	city	city	commission
1	Brad Guzan	James Hoog	London	New York	0.15
2	Jozy Altidor	James Hoog	Moscow	New York	0.15
3	Fabian Johns	James Hoog	Paris	New York	0.15
4	Graham Zusi	James Hoog	California	New York	0.15
5	Julian Green	James Hoog	London	New York	0.15
6	Geoff Camero	James Hoog	Berlin	New York	0.15

Result 20

6.

```
SELECT OrderTable.Order_No, OrderTable.Ord_Date, OrderTable.Purch_Amt, Customer.cust_name, Salesman.name, Salesman.commission
from ((OrderTable inner join Customer ON Customer.customer_id = OrderTable.customer_id) INNER JOIN Salesman ON Salesman.salesman_id = OrderTable.salesman_id);
```

#	Order_No	Ord_Date	Purch_Amt	cust_name	name	commission
1	70002	2012-10-05	65.26	Nick Rimando	James Hoog	0.15
2	70008	2012-09-10	5760.00	Nick Rimando	James Hoog	0.15
3	70013	2012-04-25	3045.60	Nick Rimando	James Hoog	0.15
4	70005	2012-07-27	2400.60	Brad Davis	James Hoog	0.15
5	70001	2012-10-05	150.50	Graham Zusi	Nail Knite	0.13
6	70007	2012-09-10	948.50	Graham Zusi	Nail Knite	0.13

7.

```
SELECT * FROM OrderTable natural join Customer natural join Salesman;
```

#	salesman_id	city	customer_id	Order_No	Purch_Amt	Ord_Date	cust_name	grade	name	commission
1	5005	London	3001	70009	270.65	2012-09-10	Brad Guzan	NULL	Pit Alex	0.11
2	5001	New York	3002	70002	65.26	2012-10-05	Nick Rimando	100	James Hoog	0.15
3	5001	New York	3002	70008	5760.00	2012-09-10	Nick Rimando	100	James Hoog	0.15
4	5001	New York	3002	70013	3045.60	2012-04-25	Nick Rimando	100	James Hoog	0.15
5	5006	Paris	3004	70010	1983.43	2012-10-10	Fabian Johns	300	Mc Lyon	0.14
6	5001	New York	3007	70005	2400.60	2012-07-27	Brad Davis	200	James Hoog	0.15

Question 4

```
'
FINAL ANSWER:
STUDENT: SSN is the PK
COURSE: N/A
ENROLL: SSN is a FK that references STUDENT.SSN PK, COURSE# is a FK that references Course.Course# PK.
BOOK ADOPTION: COURSE# is a FK that references Course.Course# PK, Book ISBN is a FK that references Text.Book ISBN PK
TEXT: N/A

'
```

Therefore, Student only has a PK, but no FK it references. ENROLL and BOOK ADOPTION have FKs that are referenced, as mentioned in the answer above. Text and Course have no FKs referenced.