

Assignment 2
Michael McAllister
Student Number:

QUESTION 1.

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

```
mysql> DESCRIBE Reviewer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| rID   | int  | NO   | PRI | NULL    | auto_increment |
| name  | text | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| rID        | int  | NO   | MUL | NULL    |                |
| mID        | int  | NO   | MUL | NULL    |                |
| stars      | int  | YES  |     | NULL    |                |
| ratingDate | date | YES  |     | curdate() | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> SHOW CREATE TABLE Movie;
```

```
| Movie | CREATE TABLE `Movie` (
  `mID` int NOT NULL AUTO_INCREMENT,
  `title` text,
  `year` int DEFAULT NULL,
  `director` text,
  PRIMARY KEY (`mID`),
  CONSTRAINT `Movie_chk_1` CHECK ((`year` < 2017))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

QUESTION 2

1.

```
mysql> UPDATE hiking
-> SET distance=10.50, est_time =10.50
-> WHERE trail='East Mesa Loop';
```

```
mysql> INSERT INTO hiking(trail,distance) VALUES
-> ('Oak Canyon', 3.00);
```

2.

```
mysql> UPDATE hiking
-> SET area = 'Mission Trails Regional Park', est_time = 2.00
-> WHERE trail = 'Oak Canyon';
Query OK, 0 rows affected (0.00 sec)
```

3.

```
mysql> DELETE FROM hiking WHERE distance >5;
Query OK, 1 row affected (0.03 sec)
```

4.

```
mysql> CREATE TABLE rating
-> (trail CHAR(50),
-> difficulty INT);
Query OK, 0 rows affected (0.30 sec)
```

5.

```
mysql> ALTER TABLE hiking
-> ADD COLUMN trailID FLOAT NOT NULL PRIMARY KEY;
Query OK, 0 rows affected (0.36 sec)
```

6.

```
mysql> ALTER TABLE rating
-> ADD COLUMN trailID FLOAT;
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE rating
-> ADD CONSTRAINT FK_trailID
-> FOREIGN KEY (trailID) REFERENCES hiking(trailID);
```

7.

```
mysql> DROP TABLE rating;
```

QUESTION 3

1.

```
108 • SELECT Salesman.name, Customer.cust_name, Customer.city
109 FROM Customer, Salesman
110 WHERE Customer.city = Salesman.city
```

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

2.

```
1 • SELECT Orders.order_no, Orders.purch_amt, Customer.cust_name, Customer.city
2 FROM Customer, Orders
3 WHERE Customer.customerID = Orders.customerID AND Orders.purch_amt BETWEEN 500 and 2000
```

#	order_no	purch_amt	cust_name	city
1	70007	948.5	Graham Zusi	California
2	70010	1983.43	Fabian Johns	Paris

3.

```
• SELECT Customer.cust_name AS "Customer Name", Salesman.name AS "Salesman"
  FROM Customer
  INNER JOIN Salesman
    ON Customer.salesmanID = Salesman.salesmanID;
```

#	Customer Name	Salesman
1	Nick Rimando	James Hoog
2	Graham Zusi	Nail Knite
3	Brad Guzan	Pit Alex
4	Fabian Johns	Mc Lyon
5	Brad Davis	James Hoog
6	Geoff Camero	Lauson Hen
7	Julian Green	Nail Knite
8	Jozy Altidor	Paul Adam

4.

```
SELECT Customer.cust_name,  
       Salesman.name,  
       Salesman.commission  
FROM Customer  
INNER JOIN Salesman  
  ON Customer.salesmanID = Salesman.salesmanID  
WHERE Salesman.commission > 0.12;
```

#	cust_name	name	commission
1	Nick Rimando	James Hoog	0.15
2	Graham Zusi	Nail Knite	0.13
3	Fabian Johns	Mc Lyon	0.14
4	Brad Davis	James Hoog	0.15
5	Julian Green	Nail Knite	0.13
6	Jozy Altidor	Paul Adam	0.13

5.

- ```
SELECT Customer.cust_name,
 Salesman.name,
 Salesman.commission
FROM Customer
INNER JOIN Salesman
 ON Customer.salesmanID = Salesman.salesmanID
WHERE Salesman.commission > 0.12 AND Customer.city <> Salesman.city;
```

| # | cust_name    | name       | commission |
|---|--------------|------------|------------|
| 1 | Graham Zusi  | Nail Knite | 0.13       |
| 2 | Julian Green | Nail Knite | 0.13       |
| 3 | Jozy Altidor | Paul Adam  | 0.13       |

6.

```
SELECT Orders.order_no,
 Orders.ord_date,
 Orders.purch_amt,
 Customer.cust_name AS "Customer Name",
 Salesman.name AS "Salesman",
 Salesman.commission
FROM Orders
INNER JOIN Customer
 ON Orders.customerID=Customer.customerID
INNER JOIN Salesman
 ON Orders.salesmanID=Salesman.salesmanID;
```

| #  | order_no | ord_date   | purch_amt | Customer Name | Salesman   | commission |
|----|----------|------------|-----------|---------------|------------|------------|
| 1  | 70013    | 2012-04-25 | 3045.6    | Nick Rimando  | James Hoog | 0.15       |
| 2  | 70008    | 2012-09-10 | 5760      | Nick Rimando  | James Hoog | 0.15       |
| 3  | 70002    | 2012-10-05 | 65.26     | Nick Rimando  | James Hoog | 0.15       |
| 4  | 70007    | 2012-09-10 | 948.5     | Graham Zusi   | Nail Knite | 0.13       |
| 5  | 70001    | 2012-10-05 | 150.5     | Graham Zusi   | Nail Knite | 0.13       |
| 6  | 70009    | 2012-09-10 | 270.65    | Brad Guzan    | Pit Alex   | 0.11       |
| 7  | 70010    | 2012-10-10 | 1983.43   | Fabian Johns  | Mc Lyon    | 0.14       |
| 8  | 70005    | 2012-07-27 | 2400.6    | Brad Davis    | James Hoog | 0.15       |
| 9  | 70003    | 2012-10-10 | 2480.4    | Geoff Camero  | Lauson Hen | 0.12       |
| 10 | 70004    | 2012-08-17 | 110.5     | Geoff Camero  | Lauson Hen | 0.12       |
| 11 | 70012    | 2012-06-27 | 250.45    | Julian Green  | Nail Knite | 0.13       |
| 12 | 70011    | 2012-08-17 | 75.29     | Jozy Altidor  | Paul Adam  | 0.13       |

7.

- ```
SELECT *
FROM Orders
NATURAL JOIN Salesman
NATURAL JOIN Customer;
```

#	salesmanID	customerID	city	order_no	purch_amt	ord_date	name	commission	cust_name	grade
1	5005	3001	London	70009	270.65	2012-09-10	Pit Alex	0.11	Brad Guzan	NULL
2	5001	3002	New York	70002	65.26	2012-10-05	James Hoog	0.15	Nick Rimando	100
3	5001	3007	New York	70005	2400.6	2012-07-27	James Hoog	0.15	Brad Davis	200
4	5001	3002	New York	70008	5760	2012-09-10	James Hoog	0.15	Nick Rimando	100
5	5006	3004	Paris	70010	1983.43	2012-10-10	Mc Lyon	0.14	Fabian Johns	300
6	5001	3002	New York	70013	3045.6	2012-04-25	James Hoog	0.15	Nick Rimando	100

Question 4.

The schema likely has the following four foreign keys:

- the attribute SSN of relation ENROLL that references relation STUDENT
- the attribute Course# in relation ENROLL that references relation COURSE
- the attribute Course# in relation BOOK_ADOPTION that references relation COURSE
- the attribute Book_ISBN of relation BOOK_ADOPTION that references relation TEXT.

For all 'possible' Foreign Keys depending on the nature of the schema there are more possible answers. A foreign key has to be the same domain as the referencing primary key. From the schema it appears there are 3 primary keys: Ssn, Course#, Book_isbn (it is also possible that Ssn and Bdate is a composite primary key in the Student table, but theoretically any of the integer fields can be a primary key as long as they don't break the uniqueness constraint).

All 3 primary keys appear to have a numeric(integer) domain so can only be the referenced column for other foreign key numeric domain attributes, these possible integer numeric domains appear 7 times. We can therefore take all possible permutations of these 7 attributes as 'possible' foreign keys and possible primary keys. This is under the assumption that all the other attributes are not numeric and the primary keys are integers.

$$\begin{aligned}P(n, r) &= P(7, 2) \\&= \frac{7!}{(7 - 2)!} \\&= 42\end{aligned}$$