

Answer 1)

A)

```
mysql> insert into hiking values
-> ('East Mesa Loop','Cuyamaca Mountains',10.50,5.50),
-> ('Oak Canyon',NULL,3.00,NULL);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

B)

```
mysql> update hiking
-> set area='Mission Trail Regional Park',est_time=2.00
-> where trail='Oak Canyon';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

C)

```
mysql> delete from hiking where distance > 5.0;
Query OK, 2 rows affected (0.01 sec)
```

D)

```
mysql> create table rating
-> (trail CHAR(50),
-> difficulty INT);
Query OK, 0 rows affected (0.02 sec)
```

E) DROP table rating;

F)

```
mysql> alter table hiking
-> add column trail_id INT PRIMARY KEY;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table rating
-> add column trail_id INT,
-> add constraint fk_trailid foreign key (trail_id) references
-> hiking (trail_id);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

QUESTION 2)

```
mysql> create table Rating (rid INT, mid INT, stars int, ratingDate datetime DEF
AULT CURRENT_TIMESTAMP, FOREIGN KEY (rid) REFERENCES Review (rid), FOREIGN KEY (
mid) REFERENCES Movie (mid));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table Movie
-> (mid INT AUTO_INCREMENT PRIMARY KEY,
-> title text,
-> year INT CHECK (year < 2016),
-> director text);
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table Review
-> ( rid INT AUTO_INCREMENT PRIMARY KEY,
-> name text);
Query OK, 0 rows affected (0.02 sec)
```

Question 3

- a) SELECT C.cust_name , S.name, city
FROM Customer AS C, Salesman AS S
WHERE C.salesman_id = S.salesman_id AND C.city = S.city;
- b) SELECT O.Order_No, O.Purch_Amt,C.cust_name,C.city
FROM Order AS O, Customer AS C

WHERE O.customer_id = C.customer_id AND Purch_Amt BETWEEN 500 AND 2000;

- c) Write an SQL statement to find out which salesmen are working for which customer

```
SELECT c.cust_name,s.name
FROM Customer AS c, Salesman AS s,
WHERE c.salesman_id = s.salesman_id
```

- d) Write an SQL statement to find the list of customers who appointed a salesman for their jobs whose commission is more than 12%

```
SELECT c.cust_name, s.name
FROM Customer AS c, Salesman AS s
WHERE c.saleman_id = s.salesman_id AND s.commision > .12 ;
```

- e) Write an SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in the same city where the customer lives and gets a commission about 12%

```
SELECT c.cust_name
FROM Customer AS c, Salesman AS s
WHERE c.salesman_id = s.salesman_id AND c.city<> s.city AND s.commission > 0.12;
```

- f) Write an SQL statement to find the details of an order ie ordernumber, order date, amount of order, which customer gives the order and which salesman work for the customer and how much commission he gets from an order

```
SELECT o.Order_No, o.Order_Date,o.Purch_Amt, c.cust_name,c.grade,
s.names.commission
FROM Order AS o, Salesman AS s, Customer AS c
WHERE o.Customer_id = c.customer_id AND o.salesman_id =s.salesman_id;
```

- g) SELECT * FROM Orders
NATURAL JOIN customer
NATURAL JOIN salesman;

Question 4)

A)

- 1) INSERT: Insert the new values For example (CS119,1,2016-5-5.5, 42A,Harnoor Singh, 416-524-2466) into SEAT_RESERVATION table
- 2) Modify: Modify the LEG_INSTANCE table with the values we added in the insert command such that Flight_Number=CS119 AND Leg_Number=1 and DATE=2016-5-5 and set Number_of_Available_seats=Number_of_Available_seats-1
- 3) These steps should be repeated for each LEG of the flight which is reserved by the customer

B) The insert operation into SEAT_RESERVATION will check all the key, entity integrity and referential integrity constraints for the relation. For example: Checking if the number of seats in each LEG_INSTANCE of the flight is greater than 1 before doing any reservation and that the SEAT_NUMBER being reserved in SEAT_RESERVATION table is available

* Entity Integrity: Checking that either of Flight_Number,Leg_number,Date and Seat_number in the SEAT_RESERVATION is not NULL

* Key Integrity: Checking that Seat_number that we are trying to insert in not already used in the SEAT_RESERVATION table

* Referential Integrity: Checking leg_number that we are inserting does not conflict with any leg_number entry in the LEG_INSTANCE table

*DOMAIN constraint: Checking that the new insert command follows the proper date format

c) * Entity Integrity: Checking that either of Flight_Number,Leg_number,Date and Seat_number in the SEAT_RESERVATION is not NULL

* Key Integrity: Checking that Seat_number that we are trying to insert in not already used in the SEAT_RESERVATION table

* Referential Integrity: Checking leg_number that we are inserting does not conflict with any leg_number entry in the LEG_INSTANCE table

*DOMAIN constraint: Checking that the new insert command follows the proper date format

d)

LEG_INSTANCE.Departure_airport_code → AIRPORT

FORMAT:

(Attribute Departure_airport_code of table LEG_INSTANCE) references primary key of AIRPORT table.

LEG_INSTANCE.Arrival_airport_code → AIRPORT

LEG_INSTANCE.Airport_id → AIRPORT

FLIGHT_LEG.FLIGHT_NUMBER → FLIGHT

FLIGHT_LEG.DEPARTURE_AIRPORT_CODE → AIRPORT

LEG_INSTANCE.(FLIGHT_NUMBER,LEG_NUMBER) → FLIGHT_LEG

LEG_INSTANCE.DEPARTURE_AIRPORT_CODE → AIRPORT

LEG_INSTANCE.ARRIVAL_AIRPORT_CODE → AIRPORT

LEG_INSTANCE.AIRPORT_ID → AIRPLANE

FARES.FLIGHT_NUMBER → FLIGHT

CAN_LAND.AIRPORT_TYPE_NAME → AIRPLANE_TYPE

Question 5)

STUDENT(Ssn, Name, Major, Bdate)

COURSE(Course#, Cname, Dept)

ENROLL (Ssn Foreign Key REFERENCES STUDENT(Ssn), Course# FOREIGN KEY REFERENCES COURSE(Course#), Quarter, Grade)

BOOK_ADOPTION (Course# FOREIGN KEY REFERENCES COURSE(Course#), Quarter, Book_isbn Foreign Key References TEXT (Book_ISBN))

TEXT(Book_isbn, Book_title, Publisher, Author)