Bresia Prudente
CS 480 - HW 3
3/15/2016

1)(a)

```
SELECT DISTINCT c.Customer_ID, c.Customer_Name
FROM Customers c
JOIN Bookings b ON c.Customer_ID=b.Customer_ID
JOIN Rooms r ON b.Room_ID=r.Room_ID
WHERE r.price=200 OR r.price=150;
```

(b)

```
 SELECT DISTINCT c.Customer_ID, c.Customer_Name
FROM Customers c
JOIN Bookings b ON c.Customer_ID=b.Customer_ID
WHERE b.Room_ID NOT IN (
        SELECT r.Room_ID
         FROM Rooms r
         WHERE r.price=100
);
```

(c)

```
SELECT c.Customer_ID, c.Customer_Name, COUNT(*) AS totalCount
FROM Customers c
WHERE Customer_ID IN (
        SELECT b.id
        FROM Bookings b
        JOIN Rooms r ON b.Room_ID=r.Room_ID
)
GROUP BY c.Customer_Name
HAVING COUNT(*) >= 2;
```

(d)

```
SELECT c.Customer_ID, c.Customer_Name, COUNT(*) AS totalCount
FROM Customers c
GROUP BY c.Customer_Name
HAVING COUNT(*) >= 3;
```

(e)

```
SELECT c.Customer_ID, c.Customer_Name, COUNT(*) AS totalCount
FROM Customers c
WHERE Customer_ID IN (
        SELECT b.id
        FROM Bookings b
        JOIN Rooms r ON b.Room_ID=r.Room_ID
)
```

```
GROUP BY c.Customer_Name
HAVING COUNT(*) >= 2;
```

2) (a)

```
SELECT e.E_ID, e.Ename
FROM Employee e
JOIN Works w ON e.E_ID=w.E_ID
JOIN Airplane a ON w.A_ID=a.A_ID
WHERE a.FlyingRange>=5000;
```

(b)

```
SELECT e.E_ID, e.Ename
FROM Employee e
JOIN Works w ON e.E_ID=w.E_ID
JOIN Airplane a ON w.A_ID=a.A_ID
WHERE a.FlyingRange>=5000
HAVING COUNT(*)>2;
```

(c)

```
DELETE FROM Airplane
WHERE FlyingRange>40000000;
```

This will also remove values associated with the key A_ID in the Works table.

(d)

```
INSERT INTO Airplane(A_ID, Model, FlyingRange, TotalMiles)
VALUES
(2345, 'Boeing 737', 3000, 0);
```

This will update the Airplane table and also update values associated with the key A_ID in the Works table.

(e)

```
UPDATE Employee
SET Stipend=0
WHERE Ename='Manfred von Richthofen';
```

This will not affect any of the other tables. Only the Employee table will be affected.

3)(a)

Assuming that the users are connecting to the database locally:

```
CREATE USER Alice@localhost IDENTIFIED BY 'alice_pass';
CREATE USER Bob@localhost IDENTIFIED BY  'bob_pass';
CREATE USER Jane@localhost IDENTIFIED BY 'jane_pass';
```

(b)
GRANT ALL PRIVILEGES ON DDL.* TO Alice@localhost;

(c)
GRANT SELECT ON DDL.* TO Bob@localhost;

(d)
GRANT SELECT, UPDATE ON DDL.* TO Jane@localhost;
REVOKE SELECT, UPDATE ON DDL.instructor FROM Jane@localhost;
GRANT DELETE, INSERT ON DDL.instructor TO Jane@localhost;

4)(a)

E_ID	A_ID	Ename	Stipend
3215	4567	Jane Zimmer	80000

(b)

FlyingRange
4000

(c)

E_ID
1234
4567

(d)

E_ID	A_ID	Model	Flying Range	TotalMiles
null	1234	Boeing 737	3000	30000000
3215	4567	Boeing 737	4000	10000000
4312	7890	Airbus A300	4000	0

5)(a)

```
CREATE FUNCTION get_total_budget(limit) RETURNS INT
        BEGIN
                DECLARE result INT;
                SET result=0;
                SELECT SUM(budget) AS budgetTotal FROM department;

                IF limit > budget THEN
                        SET result=limit;
                ELSEIF limit < budget THEN
                        SET result=budget;
                ELSE
                        SET result=0;
                END IF;
                RETURN(result);
```

```
    END
```

(b)

```
CREATE FUNCTION get_total_credits(dept_name) RETURNS INT
    BEGIN
        DECLARE total_credits INT;
        SET total_credits=0;

        SELECT SUM(tot_cred) INTO total_credits
        FROM student
        JOIN takes ON student.ID=takes.ID
        WHERE takes.semester='Fall' AND takes.year=2009;

        RETURN(total_credits);
    END
```

(c)

```
SELECT get_total_budget(1000);
```

budgetTotal
595000

```
SELECT get_total_credits('Physics');
```

SUM(tot_cred)
56
46