Database Systems: Homework #3 Solution

1.

(a)

$A + B$	A^2	B^2
1	0	1
5	4	9
1	0	1
6	4	16
7	9	16

(b)

A	B
0	1
0	1
2	3
2	4
3	4

(c)

A	B	C
2	3	4
2	3	4
0	1	$\perp$
0	1	$\perp$
2	4	$\perp$
3	4	$\perp$

(d)

A	B	C
2	3	4
2	3	4
$\perp$	0	1
$\perp$	2	4
$\perp$	2	5
$\perp$	0	2

(e)

A	B	C
2	3	4
2	3	4
0	1	⊥
0	1	⊥
2	4	⊥
3	4	⊥
⊥	0	1
⊥	2	4
⊥	2	5
⊥	0	2

2.

(a)

To deal with this, we need to set up a foreign key by Default for movie title as follows:

```
CREATE TABLE StarsIn (
        movieTitle      CHAR (80),
        movieYear       INT,
        starName        CHAR (30),
        PRIMARY KEY (movieTitle, movieYear, starName),
        FOREIGN KEY (movieTitle)        REFERENCES      Movies (title) );
```

(b)

To deal with the violations, we need to set up a foreign key for star name, and set it as cascade as follows:

```
CREATE TABLE StarsIn (
        movieTitle      CHAR (80),
        movieYear       INT,
        starName        CHAR (30),
        PRIMARY KEY (movieTitle, movieYear, starName),
        FOREIGN KEY (starName)  REFERENCES      MovieStar (name)
                                ON DELETE  CASCADE );
```

3.

(a)

When creating the table, the attribute ‘studioName’ should be written as follows:

```
studioName    CHAR(30)    CHECK ( studioName IN (‘Disney’, ‘Fox’,
                          ‘MGM’, ‘Paramount’) )
```

4.

(a)

The constraint for star name can be written as follows while creating the Table ‘MovieStar’:

```
CREATE TABLE MovieStar (
    name       CHAR(30)    CHECK ( name NOT IN (SLELECT name
               FROM MovieExec) ),
    address    VARCHAR(255),
    gender     CHAR(1),
    birthdate  DATE,
    PRIMARY KEY (name)
);
```

(b)

The constraint for studio name can be specified as follows when creating the Table ‘Studio’:

```
CREATE TABLE Studio (
    name       CHAR(30)    CHECK ( name IN (SELECT studioName
               FROM Movies) ),
    address    VARCHAR(255),
    presC#     INT,
    PRIMARY KEY (name)
);
```

5.

(a)

```
CREATE ASSERTION HigherPrice CHECK (
    NOT EXISTS (SELECT      model
                FROM        Laptop
                WHERE       price ≤ ALL  (SELECT price
                                          FROM PC
                                          WHERE  PC.ram < laptop.ram)
                )
);
```

(b)

```
CREATE ASSERTION ModelType CHECK (
    EXISTS (SELECT t1.model
            FROM Product p, PC t1
            WHERE p.model = t1.model AND p.type = 'pc')

            UNION ALL

            (SELECT t2.model
             FROM   Product p, Laptop t2
             WHERE p.model = t2.model AND p.type = 'laptop')

            UNION ALL

            (SELECT t3.model
             FROM   Product p, Printer t3
             WHERE p.model = t3.model AND p.type = 'printer')
);
```