Exercise 7.2.1

A) CREATE TABLE Movies(
   title CHAR(50),
   year INT CHECK (year > 1915),
   length INT,
   genre CHAR(50),
   studioName CHAR(50),
   producerC# INT
);

B) CREATE TABLE Movies(
   title CHAR(50),
   year INT,
   length INT CHECK (length >= 60 AND length <= 250),
   genre CHAR(50),
   studioName CHAR(50),
   producerC# INT
);

Exercise 7.2.3

A) CREATE TABLE StarsIn(
   movieTitle CHAR(50),
   movieYear INT,
   starName Char(50),
   CHECK (movieYear >= (SELECT birthdate
           FROM MovieStar
           WHERE StarsIN.starName = MovieStar.name))
);

B) CREATE TABLE Studio(
   name INT,
   address CHAR(50),
   presC# INT,
   CHECK (address NOT IN (SELECT address
           FROM Studio))
);

Exercise 7.4.1

A) CREATE ASSERTION noMakerlaptops CHECK (NOT EXISTS (SELECT maker FROM Product WHERE type = 'PC'))
AND
    maker IN (SELECT maker FROM Product WHERE type = 'Laptop'))
);

B) CREATE ASSERTION laptopSpeed CHECK ( NOT EXISTS ( SELECT maker FROM Product p1, PC
WHERE speed > ALL ( SELECT Laptop.speed FROM Product p2, Laptop
WHERE p1.maker = p2.maker AND p2.model = Laptop.model) )
);

Exercise 7.5.2
CREATE TRIGGER PCpriceTRIG INSTEAD of UPDATE ON PC REFERENCING OLD ROW AS OldTuple NEW ROW AS NewTuple FOR EACH ROW WHEN (NewTuple.price <= ALL(SELECT price FROM PC WHERE speed = newTuple.speed) UPDATE PC SET OldTuple.price = NewTuple.price;