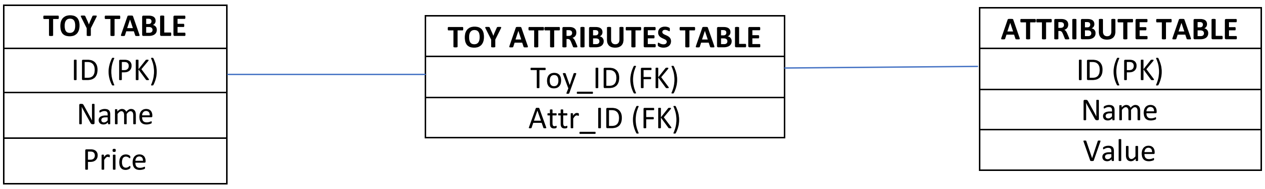
Question 1

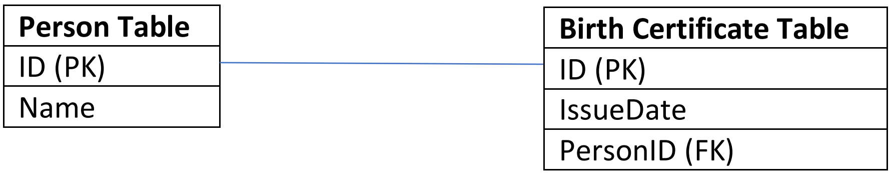


This design would ensure that storage in the database is maximised due to a lack of replication. We would create one table to store all toys and one table to store all attributes and use and one to many table in the middle to link the two. By doing this we can create as many toys as we want and place them all in the toy table and as many attributes as we like and store them in the attributes table. We can then use their ids to link a toy to an attribute and vice versa. This makes the database easy to use and maintain, keeps joins simplistic and offers scalability if new toys or attribute need to be added.

Question 2

A)

Person <- (1 , 1) -> Has <- (1 , 1) -> Birth Certificate



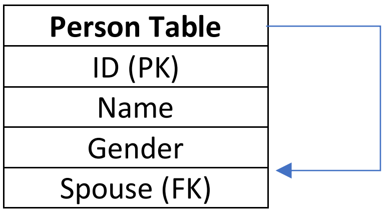
```
CREATE TABLE Person (
        Id int AUTO INCREMENT PRIMARY KEY,
        Name varchar(20),
        BirthCertID int,

)

CREATE TABLE BirthCertificate (
        Id int AUTO INCREMENT PRIMARY KEY,
        IssueDate DATE,
        PersonID int NOT NULL,
        CONSTRAINT person_fk(PersonID) REFERENCES Person(Id)
)
```

B)

Person <- (0 , 1) -> Is Married To <- (0 , 1) -> Person

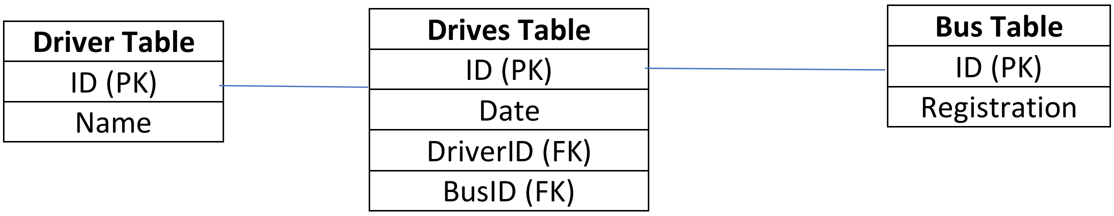


Person Table
ID (PK)
Name
Gender
Spouse (FK)

```
CREATE TABLE Person (
            Id int AUTO INCREMENT PRIMARY KEY,
             Name varchar(20),
            Gender varchar(1),
            SpouseID int,
            CONSTRAINT spouse_fk(SpouseID) REFERENCES Person (Id)
)
```

C)

Bus driver <- (0 , *) -> Drives <- (0 , *) -> Bus



Driver Table
ID (PK)
Name

Drives Table
ID (PK)
Date
DriverID (FK)
BusID (FK)

Bus Table
ID (PK)
Registration

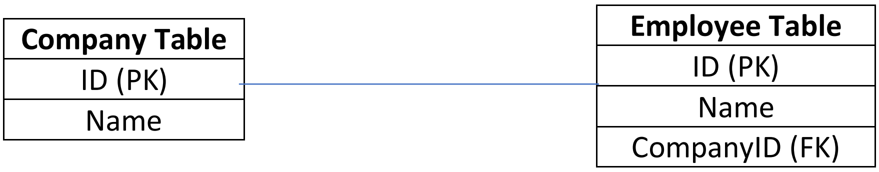
```
CREATE TABLE Driver (
            Id int AUTO INCREMENT PRIMARY KEY,
             Name varchar(20),
)

CREATE TABLE Bus (
            Id int AUTO INCREMENT PRIMARY KEY,
            Registration varchar(20),
)

CREATE TABLE Drives (
            Id int AUTO INCREMENT PRIMARY KEY,
            Date DATE,
            DriverID int,
            BusID int,
            CONSTRAINT driver_fk(DriverID) REFERENCES Driver (Id),
            CONSTRAINT bus_fk(BusID) REFERENCES Bus(Id)
)
```

Yes, this is possible to model it with a relational model and to create with an oracle create table statement. If we take the example of a company and an employee for example. A company must have at least one employee, but can also have many employees. Also, an employee must belong to the company for them to be an employee.

Company <- (1 , *) -> Employs <- (1 , 1) -> Employee

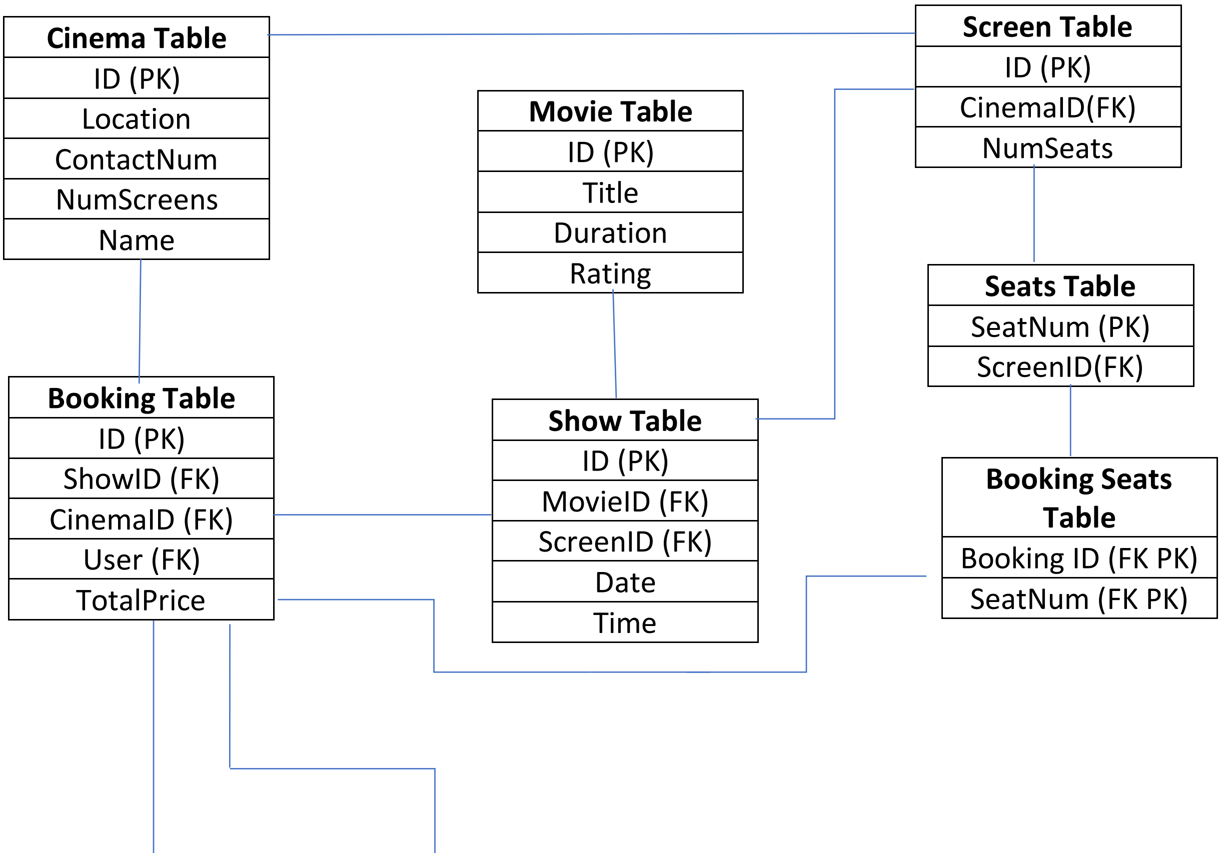


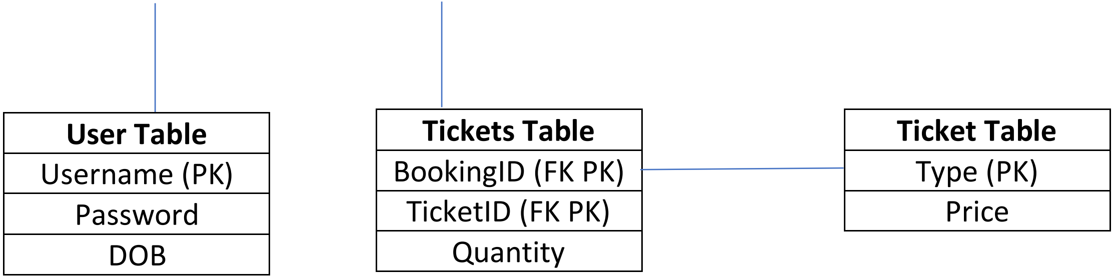
```
CREATE TABLE Company (
        Id int AUTO INCREMENT PRIMARY KEY,
        Name varchar(20),
        BirthCertID int,

)

CREATE TABLE Employee (
        Id int AUTO INCREMENT PRIMARY KEY,
        Name varchar(20),
        CompanyID int NOT NULL,
        CONSTRAINT company_fk(CompanyID) REFERENCES Company(Id)
)
```

Question 3





User Table
Username (PK)
Password
DOB

Tickets Table
BookingID (FK PK)
TicketID (FK PK)
Quantity

Ticket Table
Type (PK)
Price