Solutions of ER to Relational Schema Tutorial Sheet

Convert the following questions into relational schemas based on the given scenario in the ER diagram of the question.

Solution Q1:

```
CREATE TABLE Ingredients (
  Ing_Id int(11) NOT NULL ,
  Name varchar(50) ,
  Quantity int (20),
  Unit varchar(20) ,
  PRIMARY KEY (Ing_Id)
);
CREATE TABLE Recipe (
  Recipe_Id int (11) NOT NULL,
  Name varchar (50),
  Instructions varchar (100),
  ServingQuantity varchar (20),
  PreparationTime varchar (20),
  Skill varchar (100),
  Notes varchar (100),
  Ing_Id int(11)
  PRIMARY KEY (Recipe_Id),
  Foreign Key (Ing_Id) references Ingredients (Ing_Id)
);
```

Solution Q2:

```
CREATE TABLE Faculty (
 Faculty_Id int(11) NOT NULL ,
 Name varchar(50) ,
 DOB date,
 Gender varchar(20) ,
 Course_ID int(20),
 Grade varchar(20),
 Salary decimal(10,2),
 Designation varchar(20),
 PRIMARY KEY (Faculty_ID)
);

CREATE TABLE Research Project (
  Project_Id int(11) NOT NULL ,
  Name varchar(50) ,
  Duration int (20),
 Faculty_ID int (20),
 Area varchar(20),
 NewAttribute varchar(100),
  PRIMARY KEY (Project_Id)
```

```
  FOREIGN KEY (Faculty_ID) REFERENCES Faculty(Faculty_ID)
);


CREATE TABLE Work (
        Project_ID int(11) NOT NULL ,
        Faculty_ID int(11) NOT NULL ,
       PRIMARY KEY (Project_ID,Faculty_ID),
    FOREIGN KEY (Project_ID) References Research Project ( Project_Id ),
    FOREIGN KEY (Faculty_ID) REFERENCES Faculty(Faculty_ID)
);
```

Solution Q3: (considering one to one relationship)

```
CREATE TABLE Attendees (
  StudentNo int(11) NOT NULL ,
 email varchar(20) ,
  Name varchar (20),
 Phone int (20),
  PRIMARY KEY (StudentNo)
);

CREATE TABLE exam (
 ExamNo int(11) NOT NULL ,
 Subject varchar(50) ,
StudentNo int(11),
PRIMARY KEY (ExamNo),
FOREIGN KEY (StudentNo) references Attendees (StudentNo)
);

CREATE TABLE Reportcard (
 RecordNo int(11) NOT NULL ,
 Subject varchar(50) ,
 Name varchar(20) ,
Score int,
ExamNo int(11),
 PRIMARY KEY (RecordNo),
FOREIGN KEY (ExamNo) References Exam (ExamNo),
);
```

Solution Q4:

```
CREATE TABLE Seller (
Seller_Id int(11) NOT NULL ,
 Name varchar(50) ,
Address varchar (20),
PRIMARY KEY (Seller_ID)
);

CREATE TABLE Qualification (
Degree_ID int(10) NOT NULL,
```

```
Seller_Id int(11) NOT NULL ,
PRIMARY KEY (Degree_ID),
Foreign key (Seller_Id) references Seller_Id(Seller)
);

CREATE TABLE Buyer (
Buyer_Id int(11) NOT NULL ,
 Name varchar(50) ,
Address varchar (50),
Country varchar (20),
City       varchar (20),
MobileNo  varchar (20),
LandlineNo  varchar (20),
PRIMARY KEY (Buyer_Id)
);


CREATE TABLE car (
  Car_id int(11) NOT NULL ,
  Seller_Id int(11) NOT NULL ,
Buyer_Id int NOT NULL,
 Serial Number varchar(20) ,
  Model_no varchar (20),
 Color varchar (20),
  year varchar (20),
  PRIMARY KEY (Car_Id),
  FOREIGN KEY (Seller_Id) References Seller (Seller_Id),
FOREIGN KEY (Buyer_Id) References Buyer (Buyer_Id)
);



CREATE TABLE Invoice (
 Invoice_Id int(11) NOT NULL ,
Seller_Id int(11) NOT NULL ,
Buyer_Id int(11) NOT NULL ,
Date varchar(50) ,
PRIMARY KEY (Invoice_ID),
FOREIGN KEY (Seller_Id) References Seller (Seller_Id),
FOREIGN KEY (Buyer_Id) References Buyer (Buyer_Id)
);
```

Solution Q5:

```
CREATE TABLE ATM (
  ATM_id int(11) NOT NULL ,
  Time varchar(20) ,
  Address varchar (20),
  PRIMARY KEY (ATM_Id)
```

```
);

CREATE TABLE Transaction (
ATM_id int(11) NOT NULL,
TransNo int(11) NOT NULL ,
 Amount varchar(20) ,
Type varchar (20),
PRIMARY KEY (ATM_id,TransNo),
FOREIGN KEY (ATM_Id) References ATM (ATM_Id)

);
```

Solution Q6:

```
CREATE TABLE User (
  User_id int (11) NOT NULL ,
  fname varchar (20),
  lname varchar (20),
  dob varchar (20),
  aboutme varchar (100),
  profile_pic varchar (150),
  PRIMARY KEY (User_Id)
);

CREATE TABLE Blog (
bid varchar(20) NOT NULL ,
User_Id int(11) NOT NULL ,
title varchar(20) ,
content varchar (20),
comment varchar (100),
PRIMARY KEY (bid),
FOREIGN KEY (User_Id) References User (User_Id)
);

CREATE TABLE tags (
bid varchar(20) NOT NULL ,
tagno int(20)
PRIMARY KEY (tagno),
FOREIGN KEY (bid) References Blog (bid)
);

CREATE TABLE Writes (
        User_Id int(11) NOT NULL ,
        Bid int(11) NOT NULL ,
       PRIMARY KEY (User_Id,Bid),
    FOREIGN KEY (User_Id) References User (User_Id),
    FOREIGN KEY (bid) References Blog (bid)
);

CREATE TABLE Reads (
```

```
        User_Id int(11) NOT NULL ,
        Bid int(11) NOT NULL ,
       PRIMARY KEY (User_Id,Bid),
   FOREIGN KEY (User_Id) References User (User_Id),
   FOREIGN KEY (bid) References Blog (bid)
);
```

Solution Q7:

```
CREATE TABLE Member (
  Mem_id int(11) NOT NULL ,
  fname varchar(20) ,
  lname varchar(20) ,
  contribution varchar (30),
  PRIMARY KEY (Mem_Id)
);

CREATE TABLE Vendor (
vendor_id varchar(20) NOT NULL ,
Mem_id int(11),
name varchar(30) ,
location varchar (30),
PRIMARY KEY (vendor_id),
Foreign Key (Mem_id) references Member(Mem_id)
);

CREATE TABLE Committee(
id varchar(20) NOT NULL ,
vendor_id varchar(20),
com_name varchar(20) ,
chair varchar(30),
PRIMARY KEY (id),
Foreign Key (vendor_id) references Vendor(vendor_id)
);

CREATE TABLE Serve (
        id int(11) NOT NULL ,
        Mem_id int(11) NOT NULL ,
       PRIMARY KEY (Id,Mem_id),
   FOREIGN KEY (Id) References Committee (Id),
   FOREIGN KEY (Mem_id) References Member (Mem_id)
);
```

Solution Q8:

```
CREATE TABLE Location(
lid varchar(20) NOT NULL ,
address varchar(50) ,
name varchar(30)
```

```
PRIMARY KEY (lid)
);


CREATE TABLE Studio(
id varchar(20) NOT NULL ,
lid varchar(20) NOT NULL,
address varchar(50) ,
name varchar(30),
size  varchar(20),
PRIMARY KEY (id),
FOREIGN KEY (lid) References  Location(lid)
);


CREATE TABLE Movie (
title varchar (20) NOT NULL,
id int (11) NOT NULL,
type varchar(20),
year varchar (30),
producer varchar (30),
genre varchar (30),
length varchar (30)
PRIMARY KEY (title),
FOREIGN KEY (id) References Studio(id)
);

CREATE TABLE Movie_Stars (
  id int (11) NOT NULL,
  title varchar(50),
  contract varchar(20),
  salary int,
  name varchar (20),
  birthday varchar (20),
  address varchar (30),
  PRIMARY KEY (Id),
Foreign key (title) references Movie (title)
);
```

Solution Q9:

```
CREATE TABLE EventOrganizer (
  id int (11) NOT NULL,
  name varchar (20),
  email varchar (20),
  address varchar (30),
  mobile no varchar ,
  PRIMARY KEY (id)
```

```
);

CREATE TABLE Payment (
Payment_id varchar (20) NOT NULL,
id int (11) NOT NULL,
amount decimal (8,2),
date varchar (30),
PRIMARY KEY (Payment_id),
FOREIGN KEY (id) References  EventOrganizer(id)

);

CREATE TABLE Booking(
b_id varchar(20) NOT NULL ,
id int (11) NOT NULL,
book_date varchar(50) ,
book_amount varchar(30),
booktype varchar(20),
PRIMARY KEY (b_id),
FOREIGN KEY (id) References  EventOrganizer(id)

);

CREATE TABLE Event(
id varchar(20) NOT NULL ,
type varchar(50) ,
date varchar(25),
b_id varchar(20)
PRIMARY KEY (id),
FOREIGN KEY (b_id) References  Booking(b_id)
);
```

Solution Q10:

```
CREATE TABLE Pilot (
  id int (11) NOT NULL,
  fname varchar (20),
  lname varchar (20),
  email varchar (20),
  address varchar (30),
  mobile no varchar,
  PRIMARY KEY (id)
);

CREATE TABLE Flight (
FlightNo varchar (20) NOT NULL,
arrivaltime varchar (30),
arrivaldate varchar (30),
departuredate varchar (30),
departuretime varchar (30),
```

```
PRIMARY KEY (FlightNo)
);

CREATE TABLE Passenger(
id varchar(20) NOT NULL ,
email varchar(50) ,
mobile No varchar(25),
name  varchar(20),
PRIMARY KEY (id)
);

CREATE TABLE Booking(
b_id varchar(20) NOT NULL ,
date varchar(20),
id varchar(20) NOT NULL ,
FlightNo varchar (20) NOT NULL,
amount varchar(30),
PRIMARY KEY (b_id),
FOREIGN KEY (id) References Passenger (id),
FOREIGN KEY (FlightNo) References Flight (FlightNo)

);

CREATE TABLE Flies (
      id int (11) NOT NULL,
      FlightNo varchar (20) NOT NULL,
      PRIMARY KEY (Id, FlightNo),
   FOREIGN KEY (Id) References Pilot (Id),
   FOREIGN KEY (FlightNo) References Flight (FlightNo)
);
```