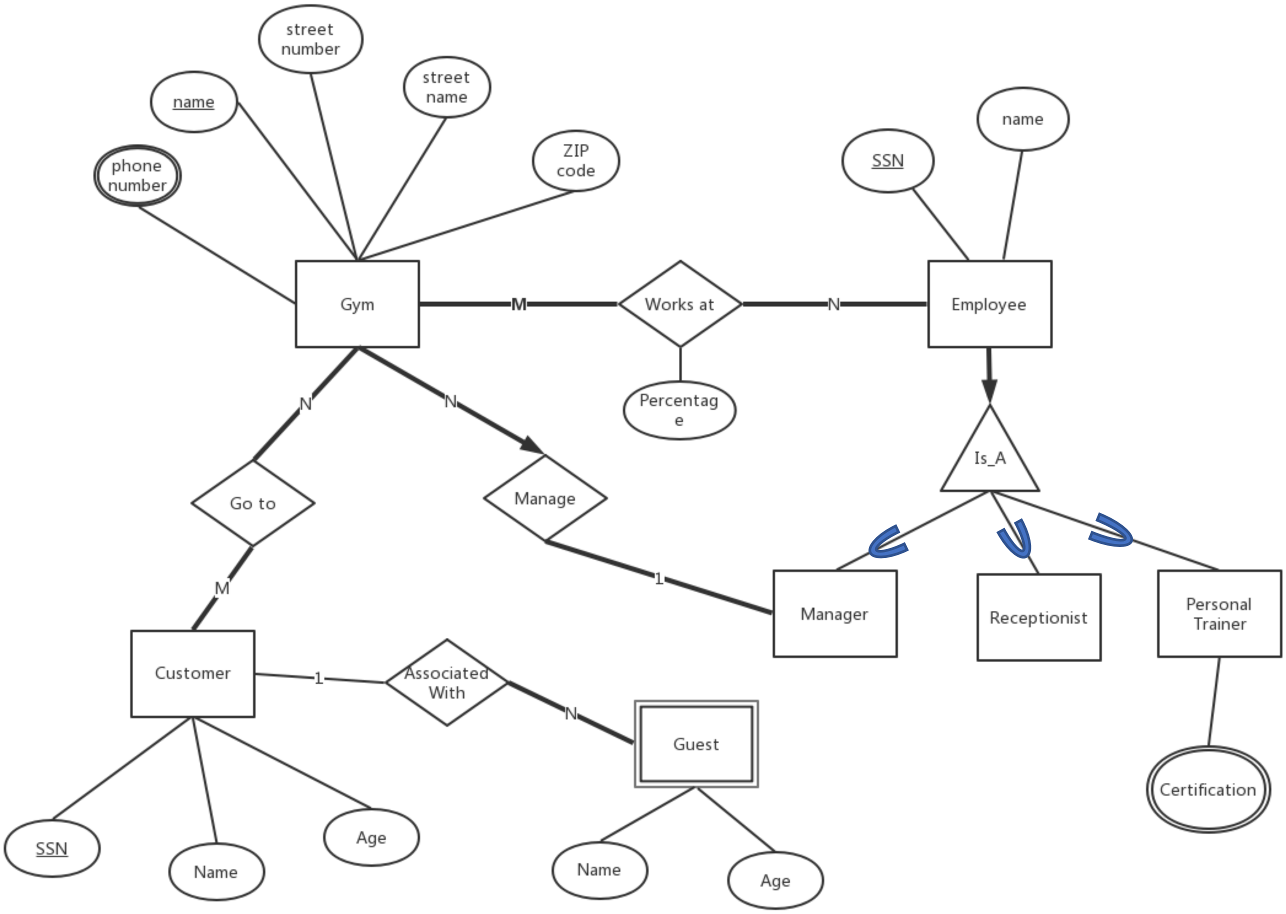
Homework1_1:



```
CREATE TABLE Customer
(
 SSN         VARCHAR(20) NOT NULL
   PRIMARY KEY,
 customerName VARCHAR(20) NULL,
 customAge   INT        NULL
)
 ENGINE = InnoDB;

CREATE TABLE Employee
(
 SSN         VARCHAR(20) NOT NULL
   PRIMARY KEY,
  employeeName VARCHAR(20) NULL
)
  ENGINE = InnoDB;

CREATE TABLE Goes_to
(
  gymName     VARCHAR(20) NOT NULL,
  customerSSN VARCHAR(20) NOT NULL,
  PRIMARY KEY (gymName, customerSSN),
  CONSTRAINT Goes_to_Gym_gymName_fk
  FOREIGN KEY (gymName) REFERENCES homework1_1.Gym (gymName),
  CONSTRAINT Goes_to_Customer_SSN_fk
  FOREIGN KEY (customerSSN) REFERENCES homework1_1.Customer (SSN)
)
  ENGINE = InnoDB;

CREATE INDEX Goes_to_Customer_SSN_fk
  ON Goes_to (customerSSN);

CREATE TABLE Guest
(
  guestName   VARCHAR(20) NOT NULL,
  guestAge    INT         NOT NULL,
  customerSSN VARCHAR(20) NOT NULL,
  PRIMARY KEY (guestName, guestAge),
  CONSTRAINT Guest_Customer_SSN_fk
  FOREIGN KEY (customerSSN) REFERENCES homework1_1.Customer (SSN)
)
  ENGINE = InnoDB;

CREATE INDEX Guest_Customer_SSN_fk
  ON Guest (customerSSN);

CREATE TABLE Gym
(
  gymName VARCHAR(20) DEFAULT '' NOT NULL
    PRIMARY KEY,
 strNo   INT                    NULL,
 strName VARCHAR(20)            NULL,
 zip     INT                    NULL
)
 ENGINE = InnoDB;

CREATE TABLE Manager
(
 employeeSSN VARCHAR(20) DEFAULT '' NOT NULL,
 gymName     VARCHAR(20) DEFAULT '' NOT NULL,
 PRIMARY KEY (employeeSSN, gymName),
 CONSTRAINT Manager_Employee_SSN_fk
 FOREIGN KEY (employeeSSN) REFERENCES homework1_1.Employee (SSN),
 CONSTRAINT Manager_Gym_gymName_fk
 FOREIGN KEY (gymName) REFERENCES homework1_1.Gym (gymName)
)
 ENGINE = InnoDB;

CREATE INDEX Manager_Gym_gymName_fk
 ON Manager (gymName);

CREATE TABLE PersonalTrainer
(
 employeeSSN   VARCHAR(20) DEFAULT ''  NOT NULL,
 certification VARCHAR(20) DEFAULT ' ' NOT NULL,
 PRIMARY KEY (employeeSSN, certification),
 CONSTRAINT PersonalTrainer_Employee_SSN_fk
 FOREIGN KEY (employeeSSN) REFERENCES homework1_1.Employee (SSN)
)
 ENGINE = InnoDB;

CREATE TABLE Receptionist
(
 employeeSSN VARCHAR(20) NOT NULL
   PRIMARY KEY,
 CONSTRAINT Receptionist_Employee_SSN_fk
 FOREIGN KEY (employeeSSN) REFERENCES homework1_1.Employee (SSN)
```

```
)
  ENGINE = InnoDB;

CREATE TABLE Works_at
(
  gymName     VARCHAR(20) NULL,
  employeeSSN VARCHAR(20) NULL,
  percentage  FLOAT       NULL,
  CONSTRAINT Works_at_Gym_gymName_fk
  FOREIGN KEY (gymName) REFERENCES homework1_1.Gym (gymName),
  CONSTRAINT Works_at_Employee_SSN_fk
  FOREIGN KEY (employeeSSN) REFERENCES homework1_1.Employee (SSN)
)
  ENGINE = InnoDB;

CREATE INDEX Works_at_Gym_gymName_fk
  ON Works_at (gymName);

CREATE INDEX Works_at_Employee_SSN_fk
  ON Works_at (employeeSSN);

CREATE TABLE phoneNumber
(
  phoneNumber VARCHAR(20) DEFAULT '' NOT NULL,
  gymName     VARCHAR(20) DEFAULT '' NOT NULL,
  PRIMARY KEY (phoneNumber, gymName),
  CONSTRAINT phoneNumber_Gym_gymName_fk
  FOREIGN KEY (gymName) REFERENCES homework1_1.Gym (gymName)
)
  ENGINE = InnoDB;

CREATE INDEX phoneNumber_Gym_gymName_fk
  ON phoneNumber (gymName);
```

Homework1_2

1.

```
SELECT sname
FROM Suppliers
WHERE EXISTS(
   SELECT Catalog.sid from Catalog WHERE Catalog.sid=Suppliers.sid
);
```

2.

```
SELECT DISTINCT sid
FROM Catalog C1
WHERE C1.cost>(
 SELECT AVG(C2.cost)
 FROM Catalog C2
 WHERE C1.pid=C2.pid
);
```

3.

```
SELECT P.pid,S.sname
FROM Suppliers S, Parts P,Catalog C
WHERE C.sid=S.sid
AND C.pid=P.pid
AND C.cost=(SELECT MAX(C1.cost)
          FROM Catalog C1
          WHERE C1.pid=C.pid
)
```

4.

```
SELECT DISTINCT C.sid
FROM Catalog C
WHERE NOT EXISTS(
 SELECT  *
 FROM Parts P
 WHERE P.pid=C.pid
 AND P.color<>'red'
)
```

5.

```
SELECT DISTINCT C.sid
FROM Catalog C
WHERE EXISTS(
  SELECT  *
  FROM Parts P
  WHERE P.pid=C.pid
  AND P.color='red'
)
UNION
SELECT DISTINCT C.sid
FROM Catalog C
WHERE EXISTS(
  SELECT  *
  FROM Parts P
  WHERE P.pid=C.pid
  AND P.color='green'
)
```

Solution 2:

```
SELECT DISTINCT C.sid
From Catalog C,Parts P
WHERE C.pid=P.pid
AND P.color='red'
UNION
SELECT DISTINCT C.sid
FROM Catalog C1, Parts P1
WHERE  C1.pid=P1.pid
AND P1.color='green'
```

6.

```
SELECT P.pname ,MAX(C.cost) AS MAXCost
FROM Suppliers S,Catalog C,Parts P
WHERE P.pid=C.pid
AND C.sid=S.sid
AND P.color IN ('red','green');
```

Homework1_3

1.
```sql
SELECT M.MovieName
FROM MovieSupplier MS, Movies M,Suppliers S
WHERE M.MovieID=MS.MovieID
AND S.SupplierID=MS.SupplierID
AND S.SupplierName IN('Ben''s Video','Video Clubhouse');
```

2.
```sql
SELECT M.MovieName
FROM Movies M,Rentals R,Inventory I
WHERE I.TapeIID=R.TapeIID
AND I.MovieID=M.MovieID
AND R.Duration >= ALL(SELECT Duration FROM Rentals);
```

3.
```sql
SELECT S.SupplierName
FROM Suppliers S
WHERE S.SupplierID NOT IN (
 SELECT MS.SupplierID
 FROM MovieSupplier MS,Inventory I
 WHERE NOT EXISTS(
  SELECT  *
  FROM Inventory I1,MovieSupplier MS1
  WHERE MS1.MovieID=I1.MovieID
  AND MS.SupplierID=MS1.SupplierID
  AND I1.MovieID=I.MovieID
 )
);
```

4.
```sql
SELECT S.SupplierName, COUNT(DISTINCT MovieID)
FROM Suppliers S, Movies M, MovieSupplier MS
where S.SupplierID=MS.SupplierID
AND M.MovieID=MS.MovieID
GROUP BY  S.SupplierName
```

5.

```
SELECT M.MovieName
FROM Movies M,Orders O
WHERE O.MovieID=M.MovieID
GROUP BY  M.MovieName
HAVING SUM(Copies)>4;
```

6.

```
SELECT  C.LastName,C.FirstnAME
FROM Rentals R,Movies M,Inventory I, Customers C
WHERE R.TapeIID=I.TapeIID
AND C.CustID=R.CustomerID
AND I.MovieID=M.MovieID
AND MovieName='Kung Fu Panda'
UNION
SELECT  C.LastName,C.FirstnAME
FROM Rentals R1,Movies M1,Inventory I1, Customers C1,MovieSupplier MS1,Suppliers S1
WHERE R1.TapeIID=I1.TapeIID
AND C1.CustID=R1.CustomerID
AND I1.MovieID=M1.MovieID
AND MS1.MovieID=M1.MovieID
AND MS1.SupplierID=S1.SupplierID
AND S1.SupplierName='Palm Video'
```

7.

```
SELECT M.MovieName
FROM Inventory I1,Inventory I2,Movies M
WHERE I1.MovieID=M.MovieID
AND I1.MovieID=I2.MovieID
AND I1.TapeIID<>I2.TapeIID;
```

8.

```
SELECT C.FirstnAME,C.LastName
FROM Customers C,Rentals R
WHERE C.CustID=R.CustomerID
AND Duration>=5;
```

9.

```
SELECT S.SupplierName
FROM Suppliers S, Movies M, MovieSupplier MS
WHERE MS.MovieID=M.MovieID
AND MS.SupplierID=S.SupplierID
AND M.MovieName='Cinderella 2015'
AND MS.Price<=ALL(
 SELECT MovieSupplier.Price
 FROM MovieSupplier , Movies
 WHERE MovieSupplier.MovieID=Movies.MovieID
 AND Movies.MovieName='Cinderella 2015'
);
```

10.

```
SELECT M.MovieName
FROM Movies
WHERE MovieID NOT IN (
 SELECT MovieID
 FROM Inventory
);
```

Homework1_4

a) 4->2>3

b) 4>3>1.5

c) 4>1.5