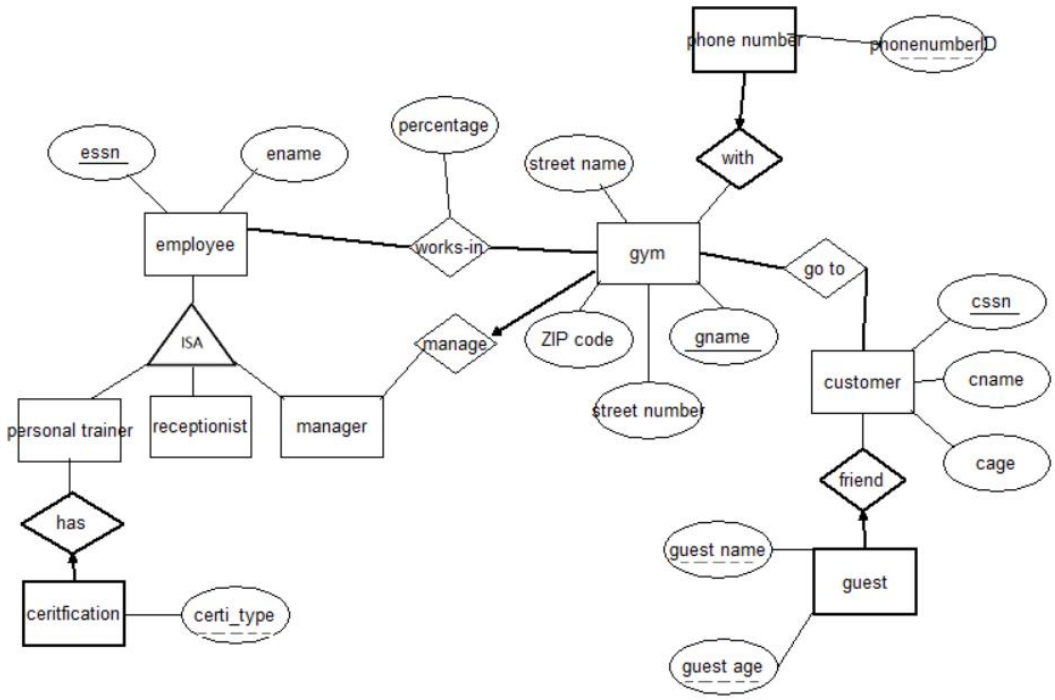
HW1

Ruid: 182007396 Name: Xinyu Lyu

1.

Answer:




```
CREATE TABLE employee(
essn CHAR(11),
ename CHAR(20),
PRIMARY KEY (essn),
UNIQUE(essn)
);
CREATE TABLE works_in
(essn CHAR(11),
gname CHAR(20),
percentage CHAR(20),
PRIMARY KEY (essn,gname,percentage),
FOREIGN KEY (essn) REFERENCE employee,
FOREIGN KEY (gname) REFERENCE gym
);
```

```
CREATE TABLE customer
(cssn CHAR(11),
cname CHAR(20),
cage CHAR（11）,
PRIMARY KEY (cssn),
UNIQUE(cssn)
);
CREATE TABLE cus_guest(
guestname CHAR(20)
guestage CHAR(11),
cssn CHAR(11) NOT NULL,
PRIMARY KEY (guestname,guestage ,cssn),
FOREIGN KEY (cssn) REFERENCE customer,
UNIQUE (guestname,guestage)
);
CREATE TABLE gym_manage(
gname CHAR(20),
streetnumber CHAR(20),
streetname CHAR(20),
ZIPcode CHAR(20),
essn CHAR(11) NOT NULL,
PRIMARY KEY (gname),
FOREIGN KEY (essn) REFERENCE manager,
UNIQUE (gname)
);
CREATE TABLE goto(
cssn CHAR(11) NOT NULL,
gname CHAR(20) NOT NULL,
PRIMARY KEY (cssn,gname),
FOREIGN KEY (cssn) REFERENCE customer,
FOREIGN KEY(gname) REFERENCE gym
);
CREATE TABLE gym_phonenumber(
gname CHAR(20) NOT NULL,
phonenumberID CHAR(20),
PRIMARY KEY (gname,phonenumberID),
FOREIGN KEY (gname) REFERENCE gym
);
```

```
CREATE TABLE personal trainer (
essn CHAR (11),
PRIMARY KEY (essn),
FOREIGN KEY (essn) REFERENCE employee
);
CREATE TABLE trainer_certification(
essn CHAR(11) NOT NULL,
certification CHAR (45),
PRIMARY KEY (essn,certi_type),
FOREIGN KEY (essn) REFERENCE trainer
);
CREATE TABLE receptionist(
essn CHAR (11),
PRIMARY KEY (essn),
FOREIGNER KEY(essn) REFERENCE employee
);
```

2.

Answer:

```
(1)  SELECT S.sname
FROM Suppliers S
WHERE NOT EXISTS
((SELECT P.pid
FROM Parts P)
EXCEPT
(SELECT C.pid
FROM Catalog C
WHERE C.sid =S.sid));

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

(3)  SELECT P.pid,S.name
FROM Catalog C, Suppliers S,Parts P
```

```
WHERE C.pid=P.pid
AND    C.sid=S.sid
WHERE C.cost=(SELECT MAX(C2.cost)
FROM Catalog C2
WHERE C2.pid=P.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 S.sid

```
FROM Suppliers S ,Parts P, Catalog C
WHERE S.sid=C.sid
AND C.pid = P.pid
AND P.color = 'Red'
UNION
SELECT DISTINCT C1.sid
FROM Catalog C1, Parts P1
WHERE C1.pid = P1.pid
AND P1.color = 'Green'
```

(6) SELECT S.sname, MAX(C.cost) as MaxCost

```
FROM Suppliers S, Parts P, Catalog C
WHERE P.pid = C.pid AND C.sid = S.sid
GROUP BY S.sname, S.sid
HAVING ANY ( P.color='green' ) AND ANY ( P.color = 'red' )
```

3.

Answer:

(1) SELECT M.MovieName

```
FROM Movie M ,MovieSuppiler MS ,Suppliers S
WHERE M.MovieID =MS.MovieID
AND MS.SupplierID =S.SupplierID
AND S.SupplierName='Benis Video'
UNION
```

```
SELECT M.MovieName
FROM Movie M ,MovieSuppiler MS ,Suppliers S
WHERE M.MovieID =MS.MovieID
AND MS.SuppljerID =S.SupplierID
AND S.SupplierName='Benis Video' ;
S.  SupplierName='VideoClubhouse';

(2)  SELECT M.MovieName
FROM Movie M,Inventory I,Rentals R
WHERE M.MovieID=I.MovieID
AND I.TapeID =R.TapeID
AND R.duration =(SELECT MAX(R2.duration)
FROM Rentals R2);

(3)  SELECT S.SupplierName
FROM Suppliers S
WHERE NOT EXISTS
  ((SELECT I.MovieID FROM Inventory I)
EXCEPT
(SELECT MS.MovieID
FROM MovieSupplier.MS
WHERE MS.SupplierID=SupplierID));

(4)  SELECT COUNT (DISTINCT MS.MovieID),S.SupplierName
FROM Suppliers, MovieSupplier MS. Movies M
WHERE S.SulplierID =MS.SupplierID
AND MS.MovieID=M.MovieID
GROUP BY S.SupplierName;

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

(6)  SELECT C.LastName,C.FirstName
FROM Customer C,Inventory I,Rentals R, Movies M
WHERE C.CustomerID=R.CustomerID
```

```
AND R.TapeID =I.TapeID
AND I.MovieID=M.MovieID AND M.MovieName='KungFu Panda'
UNION
SELECT C LastName , C FirstName
FROM Customer C,Inventory I,Rentals R,MovieSupplier MS,Suppliers S
WHERE C.CustomerID=R.CustomerID
AND R.TapeID=I.TapeID
AND I.MovieID=MS.MovieID
AND MS.SupplierID=S.Supplier ID AND S.SupplierName ='Palm Video';
```

(7)
```
SELECT Movies.MovieName
FROM Movies M, Inventory I, Inventory I2
WHERE I.MovieID = I2.MovieID
AND I.TapeID !=I2.TapeID
AND M.MovieID=I.MovieID;
```

(8)
```
SELECT DISTINCT C.LastName ,C.FirstName
FROM Customer C ,Rentals R
WHERE R.CustomerID =C.CustomerID
AND (COUNT(R.Duration)>=5);
```

(9)
```
SELECT S.SupplierName
FROM MovieSupplier MS,Movies M,Suppliers S
WHERE S.SupplierID=MS.SupplierID
AND MS.MovieID=M.MovieID
AND MS.price=
(SELECT MIN(MS2.price)
FROM MovieSupplier MS2, Movie M2
WHERE M2.MovieName='Cinderalla 2015'
AND M2.MovieID=MS2.MovieID);
```

(10)
```
SELECT M.MovieName
FROM Movie M
WHERE EXISTS
(SELECT M2.MovieID
FROM Movie M2)
EXCEPT
(SELECT I.MovieID
```

FROM Inventory I
WHERE M.MovieID=I.MovieID);

4.

Answer:

(1) (111,3) first judge the trigger condition in the clause 'when' , 4(OldTuple.price)>3(NewTuple.price) and 3(NewTuple.price)>1, so trigger the clauses from BEGIN to END to update purchase with new price set to be 1.5 where purchaseID=111, then update (111,3)
(2) (111,1.5) first update (111,3) then judge the trigger condition, 4(OldTuple.price)>3(NewTuple.price) and 3(NewTuple.price)>1, so trigger the clauses from BEGIN to END to update the changed price=1.5 where purchaseID=111
(3) (111,1.5) judge the trigger condition in 'when' clause, 4(OldTuple.price)>3(NewTuple.price) and 3(NewTuple.price)>1, so trigger the clauses from BEGIN to END to to update purchase with new price set to be 1.5 where purchaseID=111 and not update (111,3) because of the instead of trigger