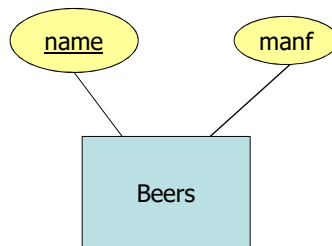
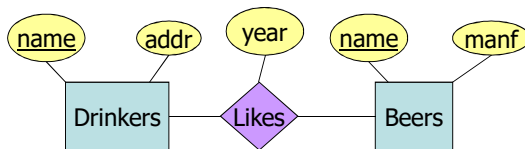


## Entity Set -> Relation



Relation: **Beers**(name, manf,  
PRIMARY KEY (name))

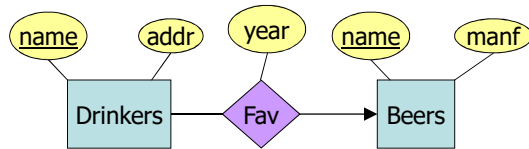


```
CREATE TABLE Drinkers (dName VARCHAR(10) PRIMARY KEY, dAddr VARCHAR(50));
```

```
CREATE Beers (bName VARCHAR(10) PRIMARY KEY, bManf VARCHAR(10));
```

```
CREATE TABLE Likes (    drinker VARCHAR(10), beer VARCHAR(10),  
                        year INT,  
                        PRIMARY KEY (drinker, beer),  
                        FOREIGN KEY (drinker) REFERENCES Drinkers,  
                        FOREIGN KEY (beer) REFERENCES Beers  
);
```





```

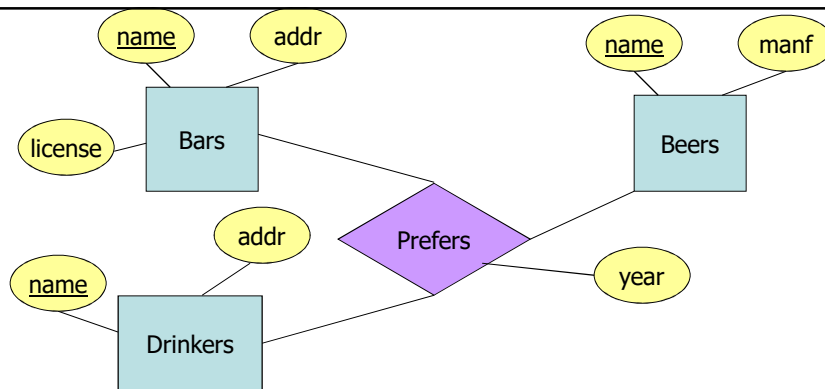
CREATE TABLE Drinkers (dName VARCHAR(10) PRIMARY KEY, dAddr VARCHAR(50));

CREATE Beers (bName VARCHAR(10) PRIMARY KEY, bManf VARCHAR(10));

CREATE TABLE Fav (
    drinker VARCHAR(10), beer VARCHAR(10) NOT NULL,
    year INT,
    PRIMARY KEY (drinker),
    FOREIGN KEY (drinker) REFERENCES Drinkers,
    FOREIGN KEY (beer) REFERENCES Beers
);

CREATE Beers (bName VARCHAR(10) PRIMARY KEY, bManf VARCHAR(10));

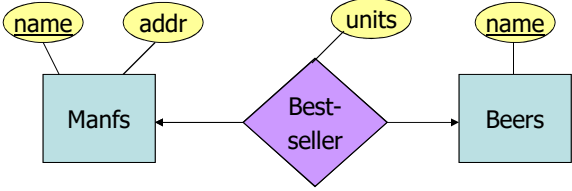
CREATE TABLE Drinkers (dName VARCHAR(10) PRIMARY KEY, dAddr VARCHAR(50),
    favBeer VARCHAR(10) REFERENCES Beers, year INT
);
  
```



```

CREATE TABLE Prefers (
    drinker ... REFERENCES Drinkers,
    bar ... REFERENCES Bars,
    beer ... REFERENCES Beers, year INT,
    PRIMARY KEY (drinker, bar, beer)
);
  
```





```

CREATE TABLE Manfs (    manfName ... PRIMARY KEY, manfAddr ...);
CREATE TABLE Beers (    beerName ... PRIMARY KEY);
CREATE TABLE BestSeller (manf ... REFERENCES Manfs,
                           beer ... REFERENCES Beers NOT NULL,
                           units ..., PRIMARY KEY (manf), UNIQUE (beer));
CREATE TABLE BestSeller (manf ... REFERENCES Manfs NOT NULL,
                           beer ... REFERENCES Beers,
                           units ..., PRIMARY KEY (beer), UNIQUE (manf));

CREATE TABLE Beers (    beerName ... PRIMARY KEY);
CREATE TABLE Manfs (    manfName ... PRIMARY KEY, manf Addr ...,
                           bestSellerBeer ... UNIQUE REFERENCES Beers, units ...);

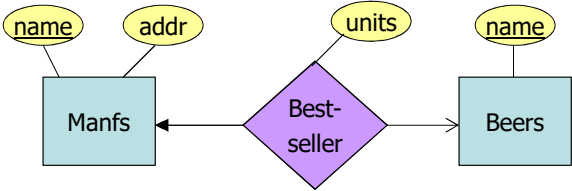
CREATE TABLE Manfs (    manfName ... PRIMARY KEY, manfAddr ...);
CREATE TABLE Beers (    beerName ... PRIMARY KEY,
                           manfBestSeller ... UNIQUE REFERENCES Manfs, units ...);

```

U. of Michigan-Flint, Dept. of CSEP

CIS/CSC 384 Database Design

[M. Mani] 50



```

CREATE TABLE Manfs (    manfName ... PRIMARY KEY, manfAddr ...);
CREATE TABLE Beers (    beerName ... PRIMARY KEY);
CREATE TABLE BestSeller (manf ... REFERENCES Manfs,
                           beer ... REFERENCES Beers NOT NULL,
                           units ..., PRIMARY KEY (manf), UNIQUE (beer));
CREATE TABLE BestSeller (manf ... REFERENCES Manfs NOT NULL,
                           beer ... REFERENCES Beers,
                           units ..., PRIMARY KEY (beer), UNIQUE (manf));

CREATE TABLE Beers (    beerName ... PRIMARY KEY);
CREATE TABLE Manfs (    manfName ... PRIMARY KEY, manf Addr ...,
                           bestSellerBeer ... UNIQUE NOT NULL REFERENCES Beers,
                           units ...);

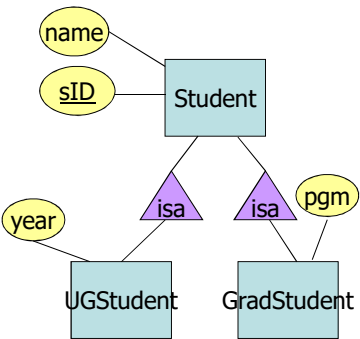
CREATE TABLE Manfs (    manfName ... PRIMARY KEY, manfAddr ...);
CREATE TABLE Beers (    beerName ... PRIMARY KEY,
                           manfBestSeller ... UNIQUE REFERENCES Manfs, units ...);

```

U. of Michigan-Flint, Dept. of CSEP

CIS/CSC 384 Database Design

[M. Mani] 51



Solution 1:

```
CREATE TABLE Student (
    sID ... PRIMARY KEY, sName ...);

CREATE TABLE UGStudent (
    sID ... PRIMARY KEY REFERENCES Student,
    year ...);

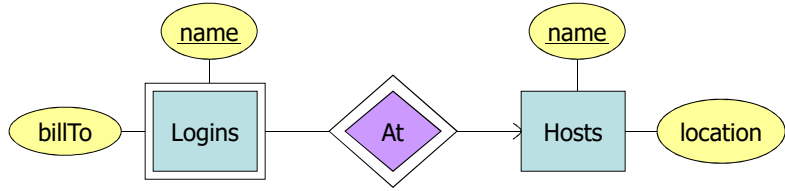
CREATE TABLE GradStudent (
    sID ... PRIMARY KEY REFERENCES Student,
    pgm ...);
```

```
CREATE TABLE Student (sID ... PRIMARY KEY, sName ..., year ..., pgm ...);

NULL value for year – could mean, it is not an UG student; or it is a UGStudent
and we do not know the value for year at this point in time..

CREATE TABLE NeitherUGNorG (sID ... PRIMARY KEY, sName ...);
CREATE TABLE UGOnly (sID ... PRIMARY KEY, sName ..., year ...);
CREATE TABLE GradOnly (sID ... PRIMARY KEY, sName ..., pgm ...);
CREATE TABLE BothUGAndGrad (sID ... PRIMARY KEY, sName ..., year ..., pgm ...);
```

U. of Michigan-Flint, Dept. of CSEP      CIS/CSC 384 Database Design      [M. Mani]      52



```
CREATE TABLE Hosts (hostname ... PRIMARY KEY, location ...);
CREATE TABLE Logins (loginName ..., hostname ... REFERENCES Hosts, billTo ...,
PRIMARY KEY(loginName, hostName));
```

```
INSERT INTO <tableName> VALUES (...);

DELETE FROM <tableName>; -- delete all the rows in a table
```

U. of Michigan-Flint, Dept. of CSEP      CIS/CSC 384 Database Design      [M. Mani]      53

```
CREATE TABLE Dept (deptNumber ... PRIMARY KEY, dName ...);
CREATE TABLE Emp (eNum ... PRIMARY KEY, dept ... REFERENCES Dept , ...);
CREATE TABLE Emp (eNum ... PRIMARY KEY, dept ... REFERENCES Dept ON
DELETE CASCADE, ...);
CREATE TABLE Emp (eNum ... PRIMARY KEY, dept ... REFERENCES Dept ON
DELETE SET NULL, ...);
```

Dept:  
1, 2, 3, 4

Emp:  
101    1  
102    1

Can I delete the row for Emp 101??? – yes  
Can I delete the row for Dept 2??? – yes  
Can I delete the row for Dept 1??? – no

Some times, it will be easier if Dept 1 can be deleted even if Emps point to that..

-- cascade delete ... Dept1 is deleted, and the Emp that point to Dept1 are also deleted  
-- set null... Dept 1 is deleted, the references to Dept1 in Emp are set to NULL

