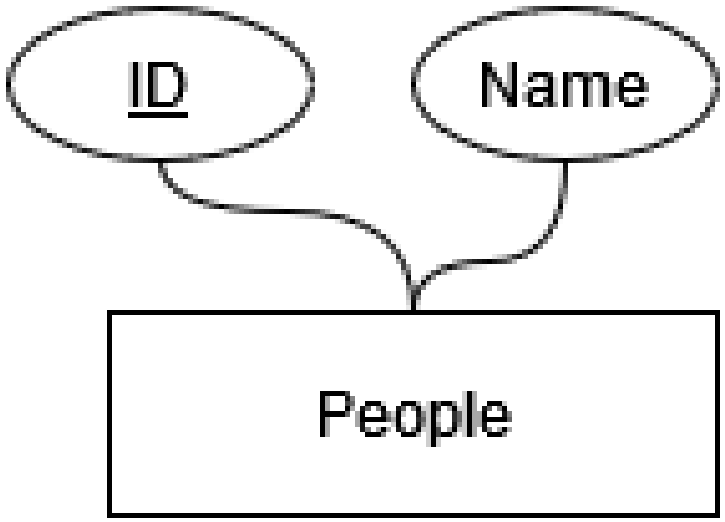
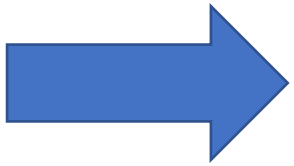
Handbook on ER-diagram to DDL translations.

Entities

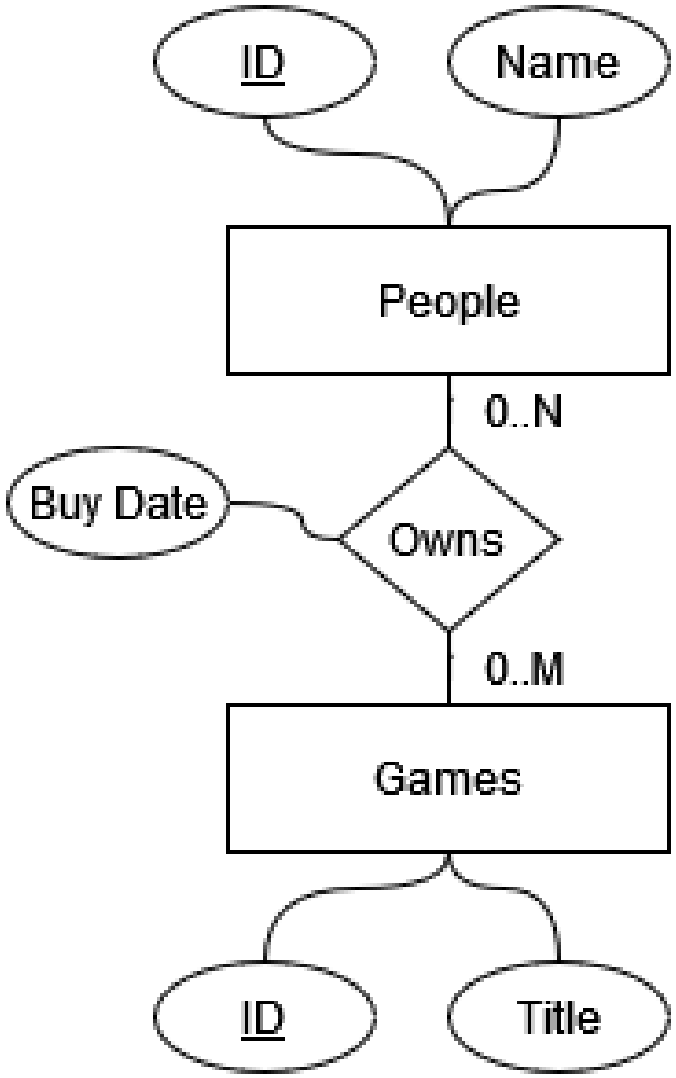




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);
```

Relations

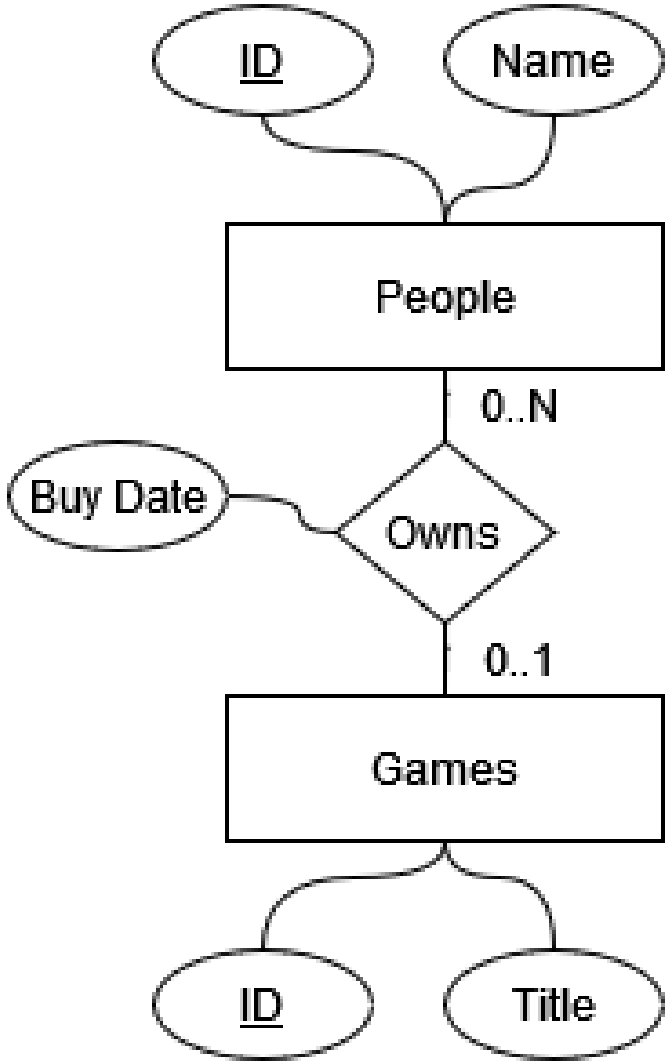
Basic




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    PRIMARY KEY (PeopleID, GameID)
);
```

Relations

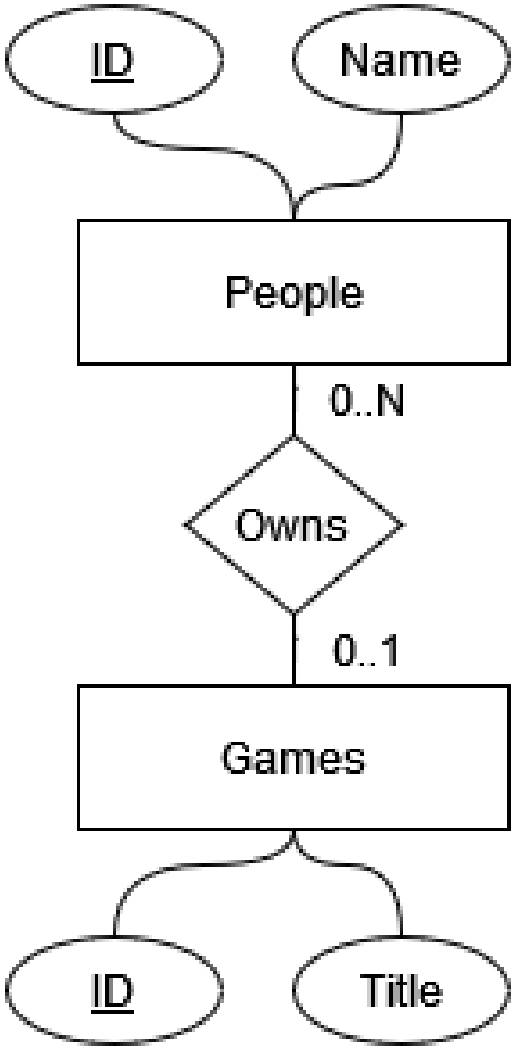
Maximum 1 with attributes




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    PRIMARY KEY (PeopleID)
);
```

Relations

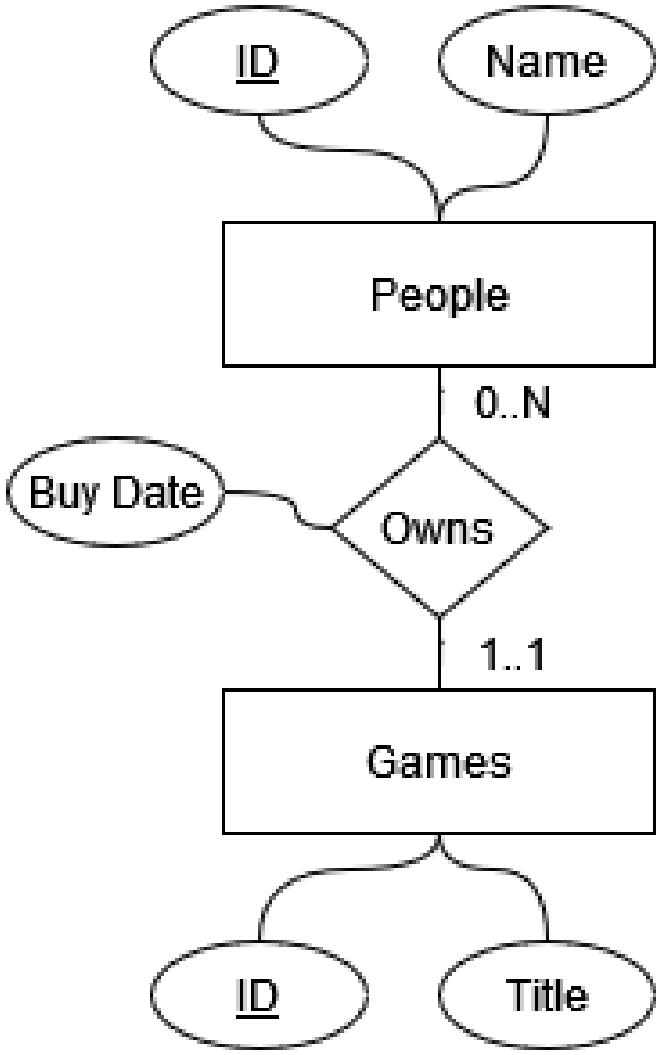
Maximum 1 without attributes




```sql
CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE People (
    ID INT PRIMARY KEY,
    GameID INT REFERENCES Games(ID),
    Name VARCHAR NOT NULL
);
```

Relations

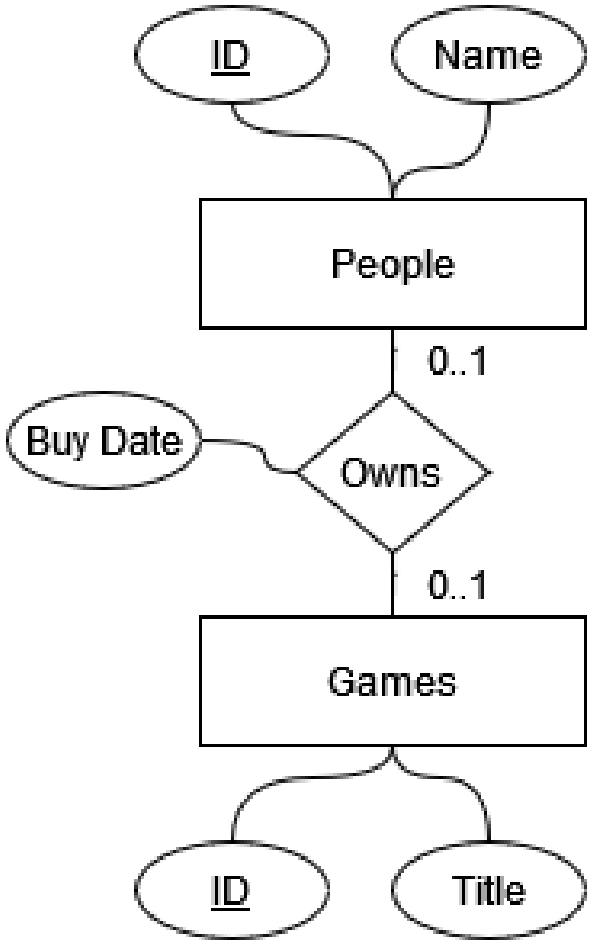
Exactly 1




```sql
CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE People (
    ID INT PRIMARY KEY,
    GameID INT NOT NULL REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    Name VARCHAR NOT NULL
);
```

Relations

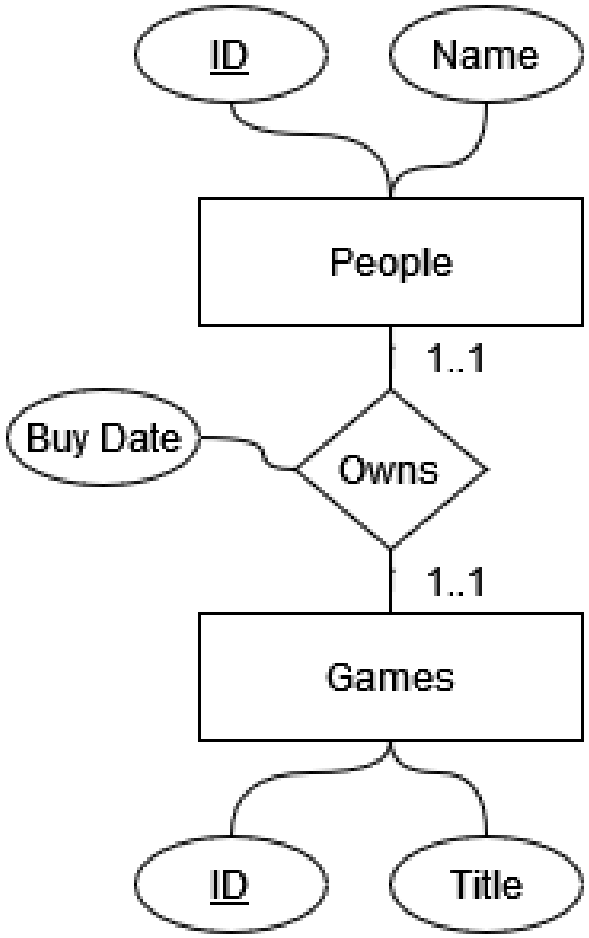
Maximum 1 in both directions




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    PRIMARY KEY (PeopleID),
    UNIQUE (GameID)
);
```

Relations

Exactly 1 in both directions



Alternative 2:
Merge tables

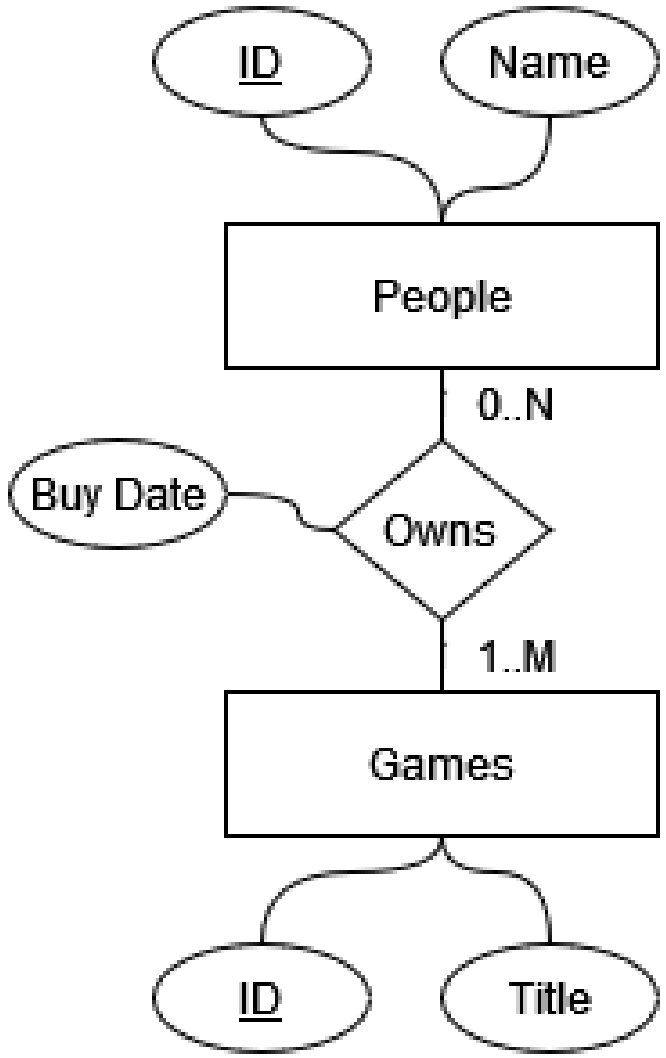
```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL,
    GameTitle VARCHAR NOT NULL,
    BuyDate DATE NOT NULL
);
```

Alternative 3:
Pick one FK

```sql
CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL,
    GameID INT NOT NULL REFERENCES Games(ID),
    BuyDate DATE NOT NULL
);
```

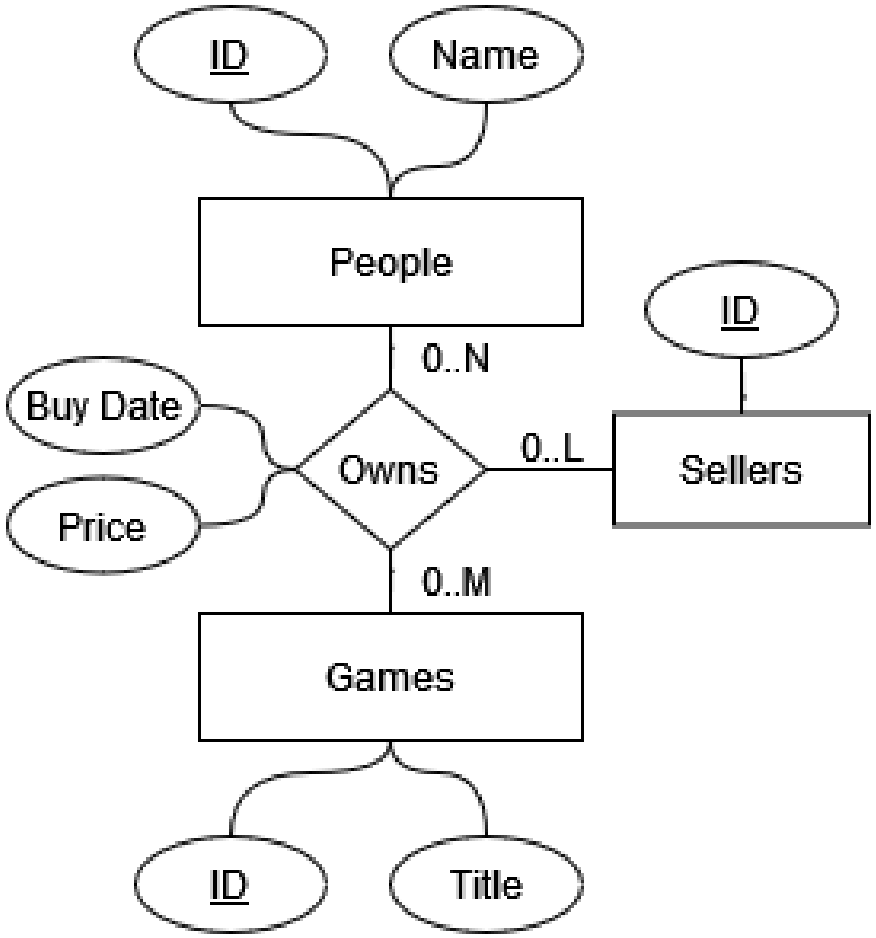
Relations

Minimum 1 - Not Supported in DDL!



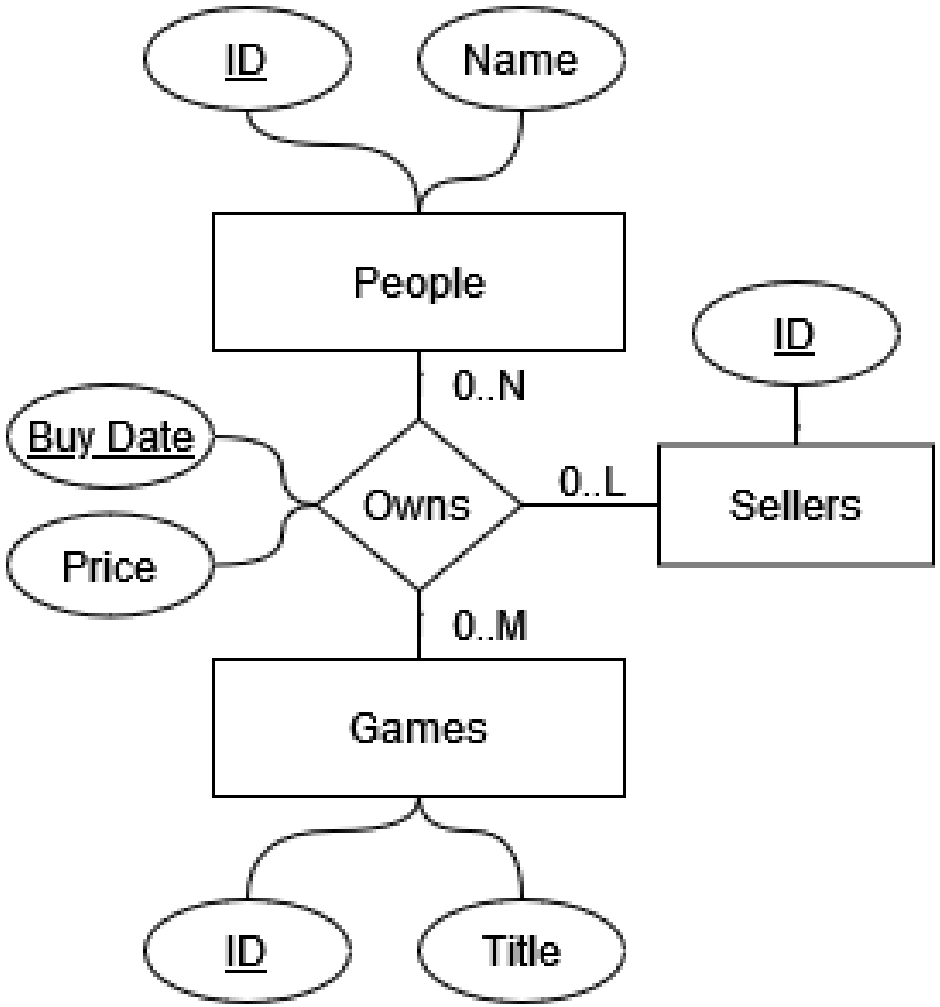

```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    PRIMARY KEY (PeopleID, GameID)
);
```

Tertiary Relations



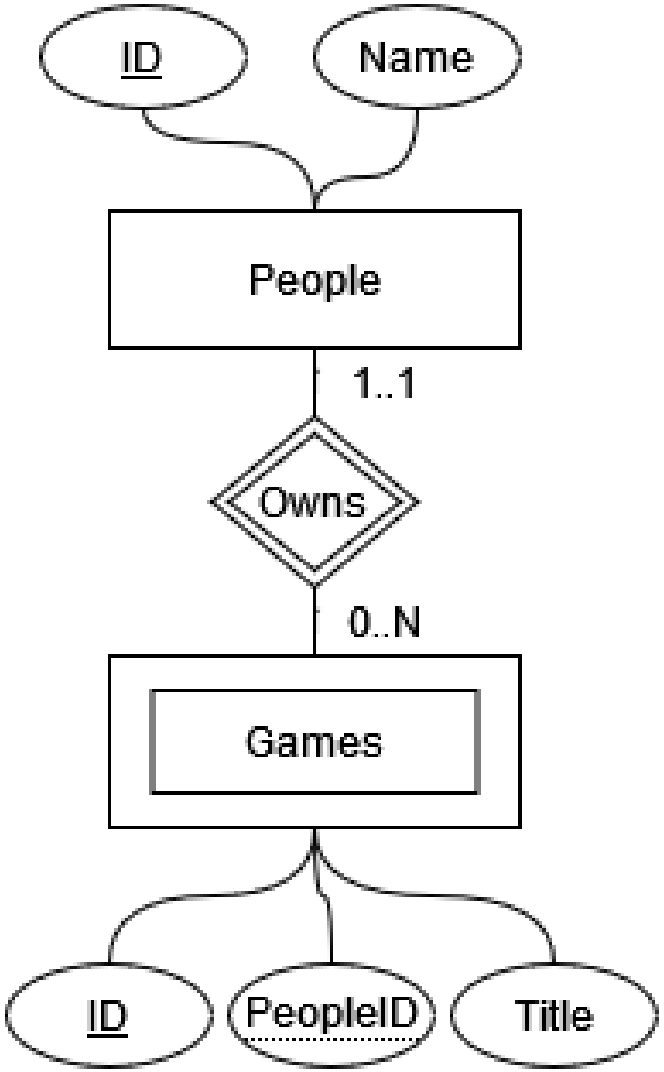

```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Sellers (
    ID INT PRIMARY KEY
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    SellerID INT REFERENCES Sellers(ID),
    BuyDate DATE NOT NULL,
    Price FLOAT NOT NULL,
    PRIMARY KEY (PeopleID, GameID, SellerID)
);
```

Tertiary Relations / Partial Keys




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Sellers (
    ID INT PRIMARY KEY
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    SellerID INT REFERENCES Sellers(ID),
    BuyDate DATE,
    Price FLOAT NOT NULL,
    PRIMARY KEY (PeopleID, GameID, SellerID, BuyDate)
);
```

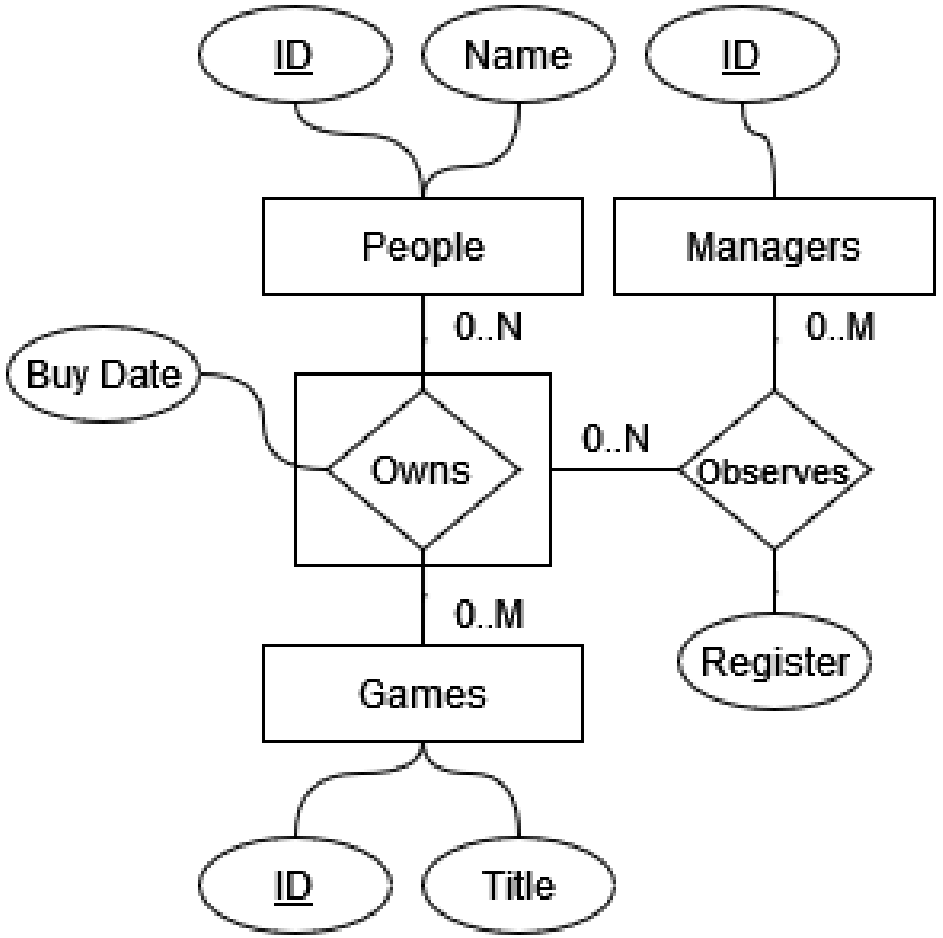
Weak Entities




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT,
    PeopleID INT REFERENCES People(ID),
    Title VARCHAR NOT NULL,
    PRIMARY KEY(ID, PeopleID)
);
```

Aggregation

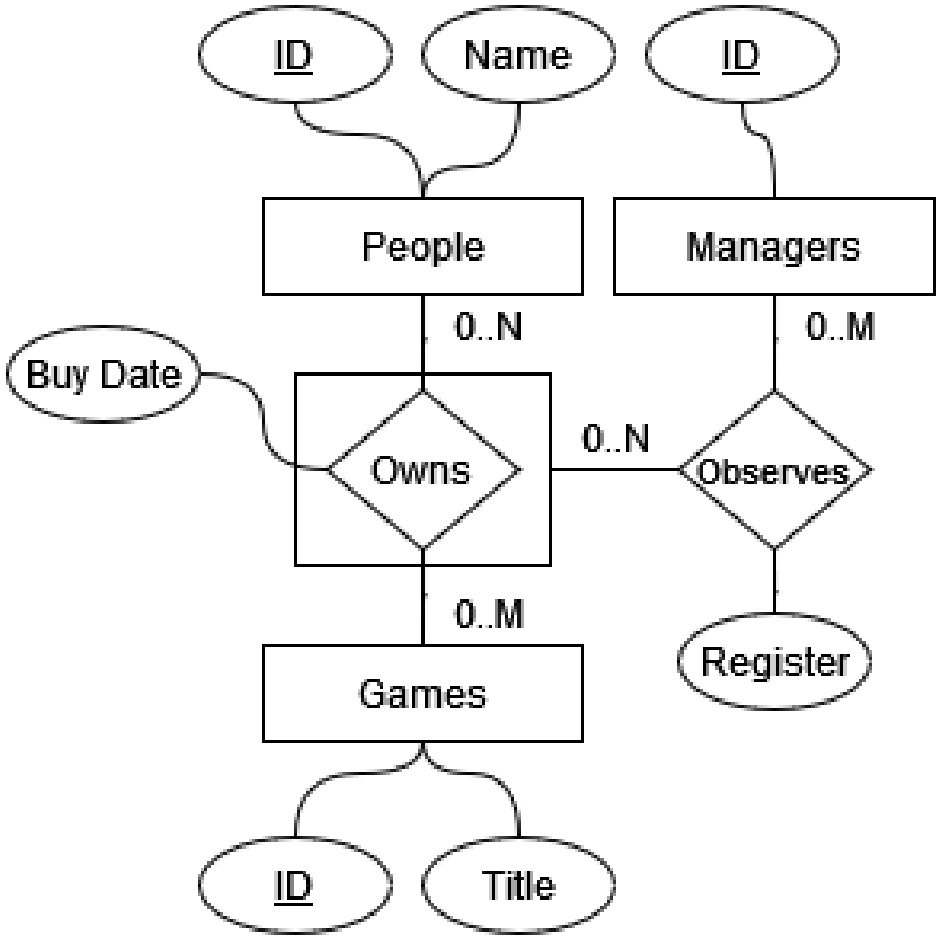
Option 1: Relationship Key




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    PRIMARY KEY (PeopleID, GameID)
);

CREATE TABLE Managers (
    ID INT PRIMARY KEY
);

CREATE TABLE Observes (
    ManagerID INT REFERENCES Managers(ID),
    PeopleID INT,
    GameID INT,
    Register INT NOT NULL,
    FOREIGN KEY (PeopleID, GameID)
      REFERENCES Owns(PeopleID, GameID),
    PRIMARY KEY (ManagerID, PeopleID, GameID)
);
```

Aggregation

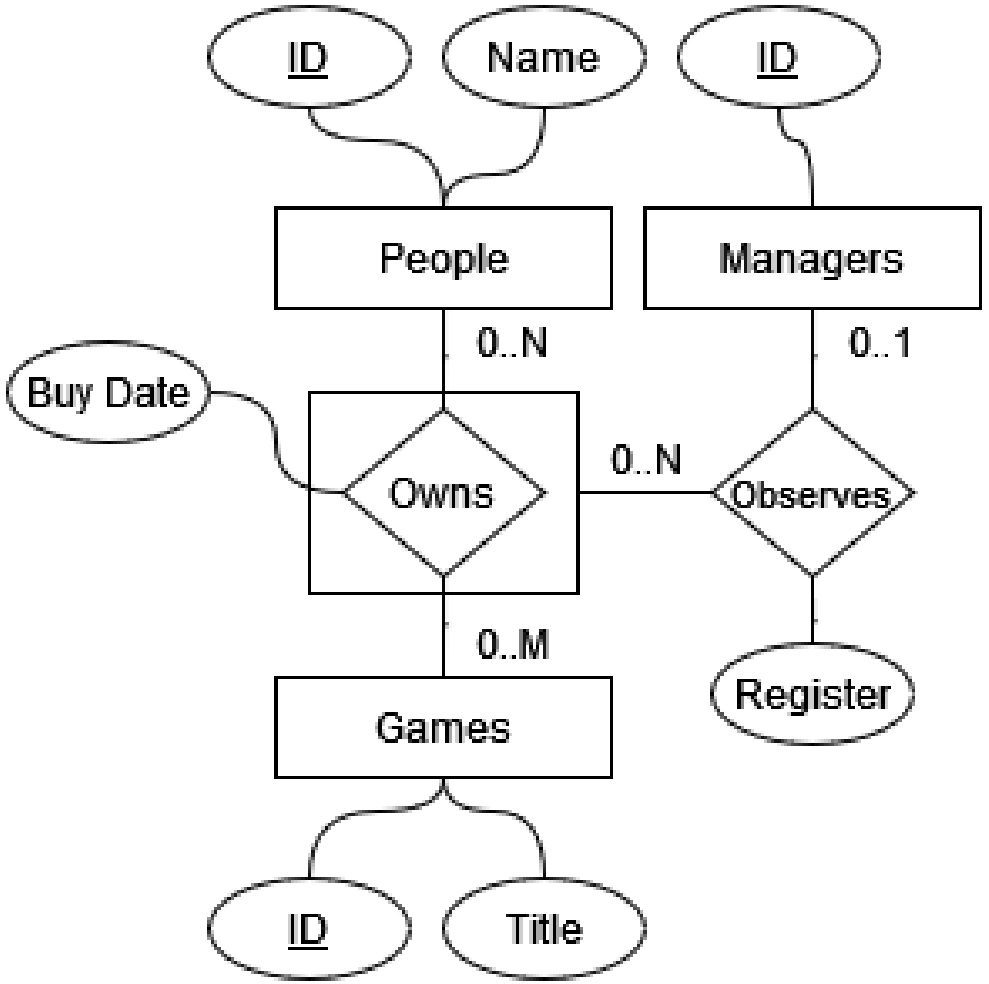
Option 2: New Relationship Key




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    ID INT PRIMARY KEY,
    PeopleID INT NOT NULL REFERENCES People(ID),
    GameID INT NOT NULL REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    UNIQUE (PeopleID, GameID)
);

CREATE TABLE Managers (
    ID INT PRIMARY KEY
);

CREATE TABLE Observes (
    ManagerID INT REFERENCES Managers(ID),
    OID INT REFERENCES Owns(ID),
    Register INT NOT NULL,
    PRIMARY KEY (ManagerID, OID)
);
```

Aggregation

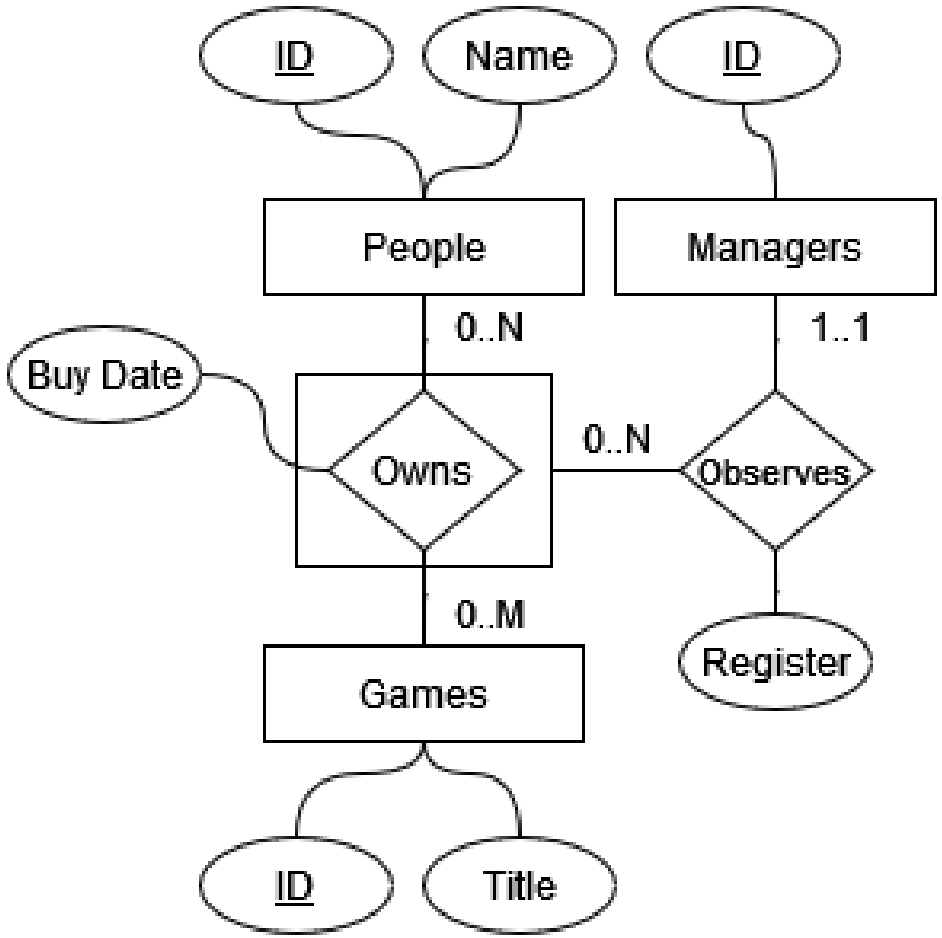
Maximum 1




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    BuyDate DATE NOT NULL,
    PRIMARY KEY (PeopleID, GameID)
);

CREATE TABLE Managers (
    ID INT PRIMARY KEY
);

CREATE TABLE Observes (
    ManagerID INT NOT NULL REFERENCES Managers(ID),
    PeopleID INT,
    GameID INT,
    Register INT NOT NULL,
    FOREIGN KEY (PeopleID, GameID)
      REFERENCES Owns(PeopleID, GameID),
    PRIMARY KEY (PeopleID, GameID)
);
```

Aggregation

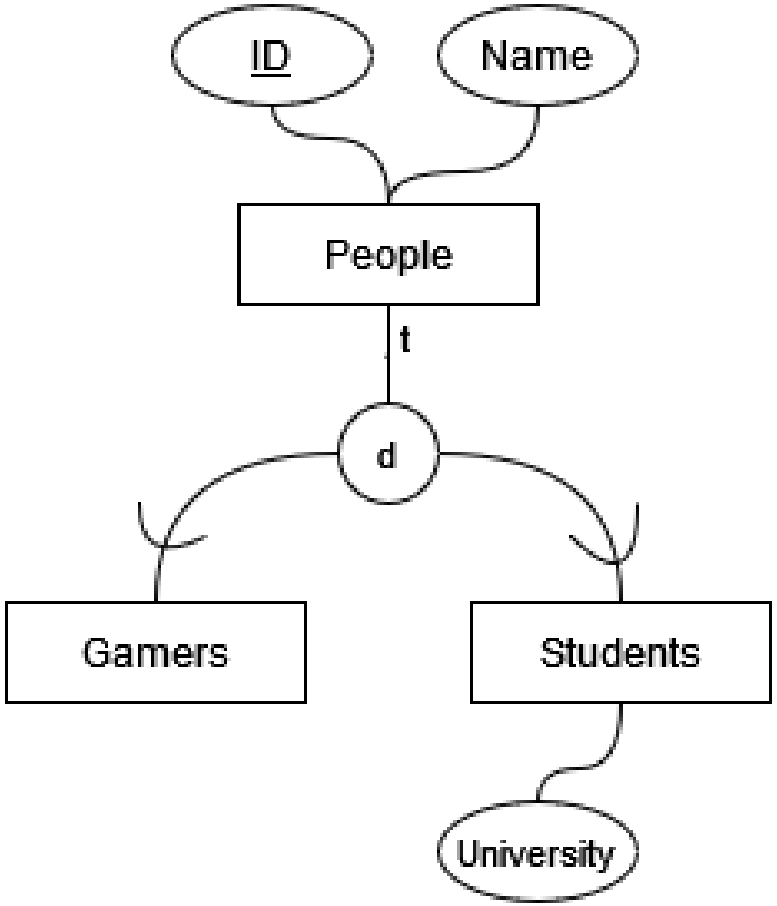
Exactly 1




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Games (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL
);

CREATE TABLE Managers (
    ID INT PRIMARY KEY
);

CREATE TABLE Owns (
    PeopleID INT REFERENCES People(ID),
    GameID INT REFERENCES Games(ID),
    ManagerID INT NOT NULL REFERENCES Managers(ID),
    BuyDate DATE NOT NULL,
    Register INT NOT NULL,
    PRIMARY KEY (PeopleID, GameID)
);
```

Specialization

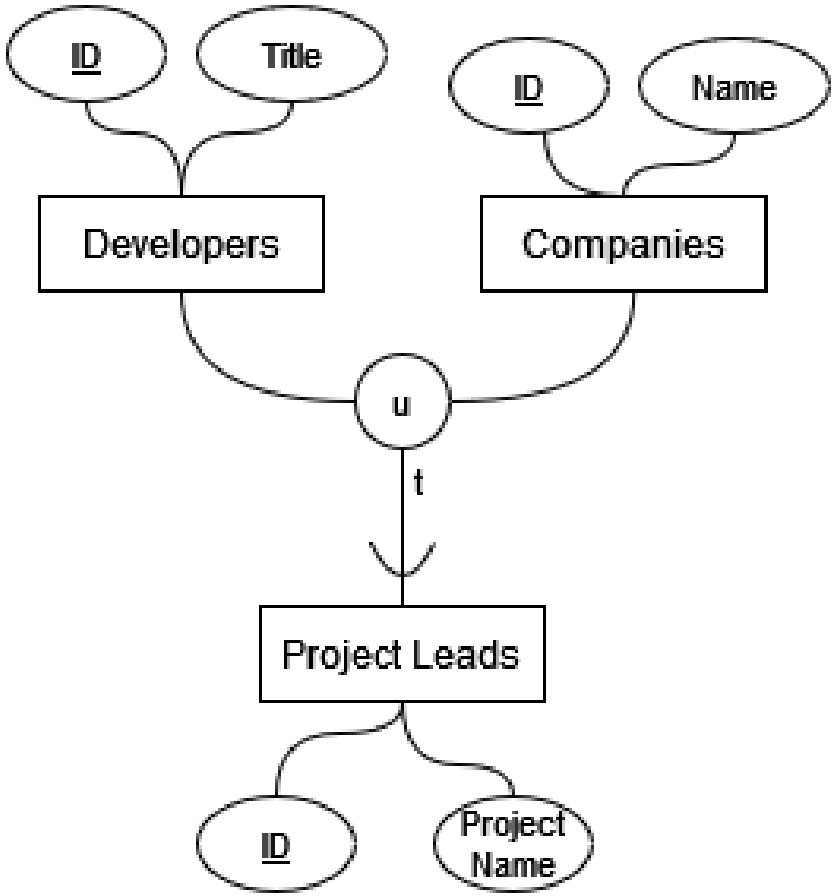
This is the same for partial and overlapping specialization




```sql
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL
);

CREATE TABLE Gamers (
    ID INT PRIMARY KEY REFERENCES People(ID)
);

CREATE TABLE Students (
    ID INT PRIMARY KEY REFERENCES People(ID),
    University VARCHAR NOT NULL
);
```

Categorization

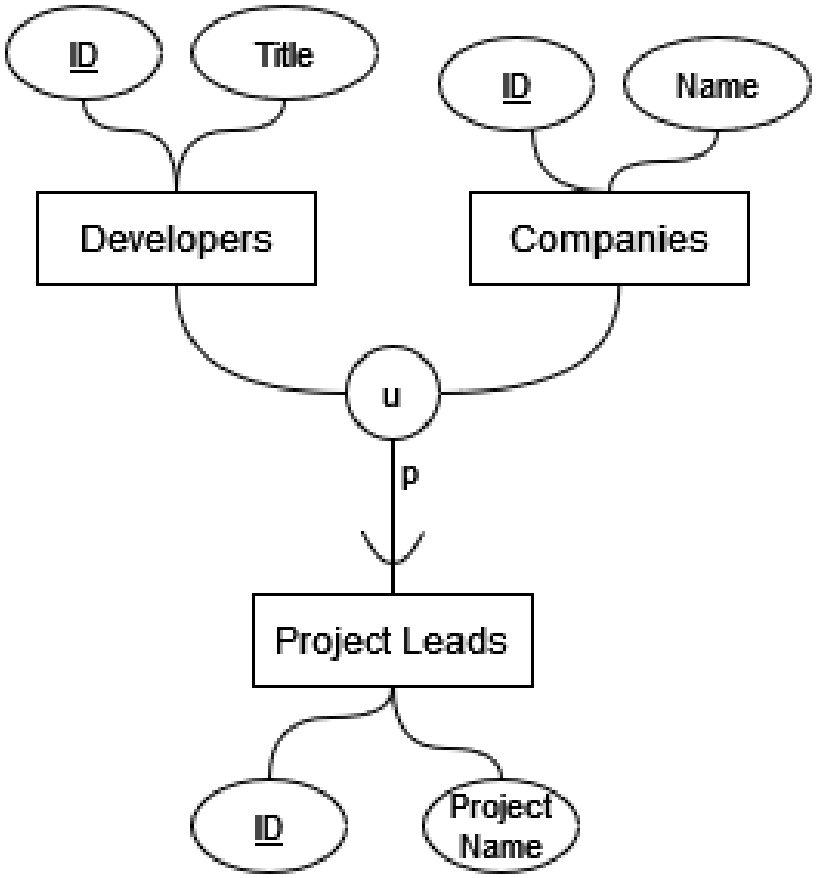
Total




```sql
CREATE TABLE ProjectLeads (
    ID INT PRIMARY KEY,
    ProjectName VARCHAR NOT NULL
);

CREATE TABLE Developers (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL,
    PLID INT NOT NULL REFERENCES ProjectLeads(ID)
);

CREATE TABLE Companies (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL,
    PLID INT NOT NULL REFERENCES ProjectLeads(ID)
);
```

Categorization

Partial




```sql
CREATE TABLE ProjectLeads (
    ID INT PRIMARY KEY,
    ProjectName VARCHAR NOT NULL
);

CREATE TABLE Developers (
    ID INT PRIMARY KEY,
    Title VARCHAR NOT NULL,
    PLID INT REFERENCES ProjectLeads(ID)
);

CREATE TABLE Companies (
    ID INT PRIMARY KEY,
    Name VARCHAR NOT NULL,
    PLID INT REFERENCES ProjectLeads(ID)
);
```