High Level Database Design

dmg

May 31, 2018

Student and Course

```
DROP TABLE IF EXISTS Student;
CREATE TABLE Student (
       sid CHAR(10),
       sname VARCHAR,
       PRIMARY KEY (sid) -- constraint
       );


DROP TABLE IF EXISTS Course;
CREATE TABLE Course (
       cid CHAR(10),
       cname VARCHAR,
       term CHAR(3),
       PRIMARY KEY (cid, term)
       );
```

DROP TABLE

CREATE TABLE

Enrolled

```
CREATE TABLE Enrolled
       sid CHAR(10),
       cid CHAR(10),
       term CHAR(3),

       PRIMARY KEY (sid, cid, term),
       FOREIGN KEY(sida) REFERENCES Students,
       FOREIGN KEY(cid,term) REFERENCES Course
       );
```

0 or 1 relationship

```
CREATE TABLE R (
        a1 INTEGER, -- not null, implied by being part of key
        b1 INTEGER NOT NULL,
        r1 INTEGER,
        PRIMARY KEY (a1),
        FOREIGN KEY(a1) REFERENCES A,
        FOREIGN KEY(b1) REFERENCES B,
)
```

```
CREATE TABLE R (
        a1 INTEGER, -- not null, implied by being part of key
        a2 INTEGER, -- not null, implied by being part of key
        b1 INTEGER NOT NULL,
        r1 INTEGER,
        PRIMARY KEY (a1,a2),
        UNIQUE (b1),
        UNIQUE (a1,a2), -- implicit
        FOREIGN KEY(a1,a2) REFERENCES A,
        FOREIGN KEY(b1) REFERENCES B,
)
```

Arrows in both directions

```
CREATE TABLE R (
        a1 INTEGER, -- a1 -> b1, r1
        b1 INTEGER NOT NULL,
        r1 INTEGER,
        PRIMARY KEY (a1),
        UNIQUE (b1) --- b1 -> a1,r1
        FOREIGN KEY(a1) REFERENCES A,
        FOREIGN KEY(b1) REFERENCES B,
);
```

or n

```
CREATE TABLE R (
        a1 INTEGER  NOT NULL,
        b1 INTEGER, -- not null, implied by being part of key
        r1 INTEGER,
        PRIMARY KEY (b1),
        UNIQUE (a1)
        FOREIGN KEY(a1) REFERENCES A,
        FOREIGN KEY(b1) REFERENCES B,
);
```

Exactly one relationship

```
CREATE TABLE AR (
        a1 INTEGER,
        a2 INTEGER,
        b1 INTEGER NOT NULL, -- always related to one tuple in B
        r1 INTEGER, -- can be null
        PRIMARY KEY (a1),
        FOREIGN KEY(b1) REFERENCES B,
)
```

Roles

```
CREATE TABLE Sequel (
        originaltitle VARCHAR,
        originalyear INT,
        sequeltitle VARCHAR,
        sequelyear INT,
        PRIMARY KEY (sequaltitle, sequelyear),
        FOREIGN KEY(originaltitle, originalyear)
             REFERENCES Movie,
        FOREIGN KEY(sequeltitle, sequelyear)
            REFERENCES Movie
);
```

Inheritance

- create Movies as usual

```
CREATE TABLE MurderMysteries (
        title CHAR(30),
        year INTEGER,
        weapon VARCHAR,
        PRIMARY KEY(title, year)
        FOREIGN KEY(title, year) REFERENCES Movies
);
```

- create table voices...
  - ... as usual... but references Cartoons

Weak entity

```
CREATE TABLE  Dependents (
    eid CHAR(10),
    dname CHAR(30),
    age  INTEGER,
    PRIMARY KEY(eid, dname),
    FOREIGN KEY(eid) REFERENCES Employees
        ON DELETE CASCADE
);
```