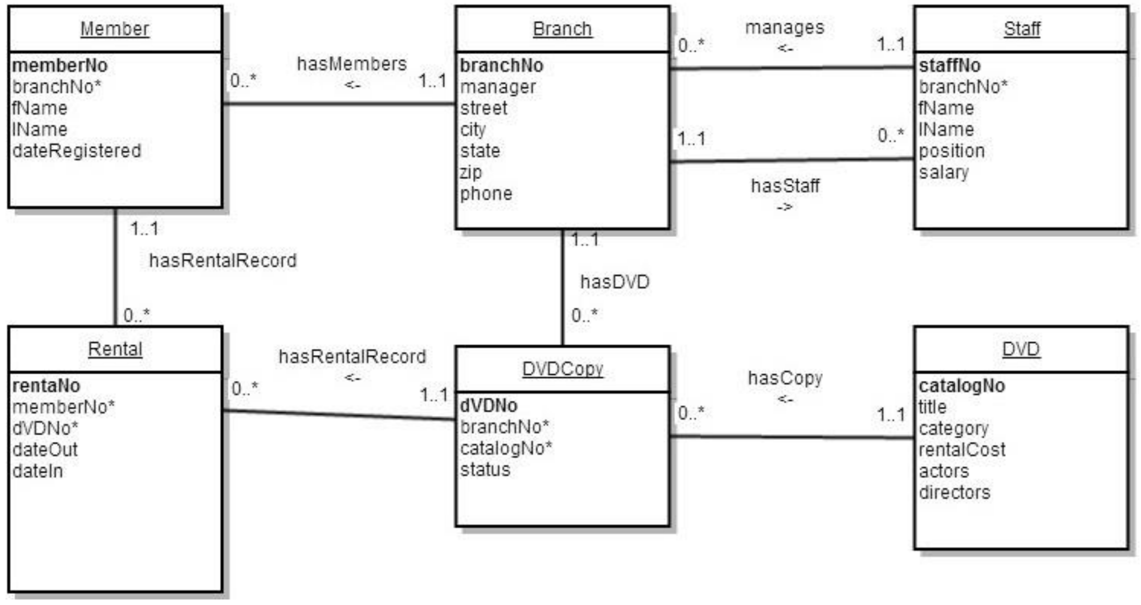
Database Design, CSCI 340, Spring 2016
Data Definition Language (DDL), INSERT, DELETE and UPDATE, Chapter 7, Feb. 12



Member:
memberNo – CHAR(7)
fName – CHAR(30)
lName – CHAR(30)
dateRegistered – DATE

Branch:
branchNo – CHAR(4)
street – VARCHAR(100)
city – CHAR(30)
state – CHAR(2)
zip – CHAR(5)
phone – CHAR(10)

Staff:
staffNo – CHAR(7)
fName – CHAR(30)
lName – CHAR(30)
position – CHAR(30)
salary - SMALLMONEY

Rental:
rentalNo – INTEGER (auto increment)
dateOut - DATE
dateIn – DATE

DVD:
catalogNo – CHAR(8)
title – VARCHAR(200)
category – CHAR(10)
rentalCost – SMALLMONEY
actors – VARCHAR(1000)
directors – VARCHAR(1000)

DVDCopy:
dVDNo – INTEGER (auto increment)
status –CHAR(10)

1. Create the tables above, using Microsoft's SQL Server, within your personal database.

   Here is a script to create the Branch and Staff tables:

```
CREATE TABLE Branch
(branchNo CHAR(4) PRIMARY KEY,
manager CHAR(7),
street VARCHAR(100),
city CHAR(30),
state CHAR(2),
zip CHAR(5),
phone CHAR(10))

CREATE TABLE Staff
(staffNo CHAR(7) PRIMARY KEY,
 branchNo CHAR(4) CONSTRAINT staff_branch_FK
      FOREIGN KEY REFERENCES Branch(branchNo),
fName CHAR(30),
lName CHAR(30),
position CHAR(30),
salary SMALLMONEY)


/* Add the manager reference */
ALTER TABLE Branch
ADD CONSTRAINT branch_manager_FK
FOREIGN KEY (manager) REFERENCES Staff(staffNo)
```

If you make a mistake and need to drop the tables:

```
/*
ALTER TABLE Branch
DROP CONSTRAINT branch_manager_FK

DROP TABLE Staff

DROP TABLE Member

DROP TABLE Branch
*/
```

I typically use scripts to create databases. If you would like, you can also use the SQLServer graphical interface. To use the graphical interface, right click "Tables" in the Object Explorer.

2. Place at least three records into each table.

   Here is a script to insert records into the Branch and Staff tables:

```
/* Insert some records into Branch */
INSERT INTO Branch (branchNo, street, city, state, zip, phone)
     VALUES ('B299','17 West Park Street', 'Butte',
          'MT','59701','4067234432')

INSERT INTO Branch (branchNo, street, city, state, zip, phone)
     VALUES ('B100','500 Montana Street', 'Butte',
          'MT','59701','4067829855')

INSERT INTO Branch (branchNo, street, city, state, zip, phone)
     VALUES ('B516','170 Higgins', 'Missoula',
          'MT','59834','4064984422')


/* Insert some records into Staff */
INSERT INTO Staff (staffNo, branchNo, fName, lName,
     position, salary) VALUES ('S333333','B299',
          'Eli', 'Rose','cashier','12.00')

INSERT INTO Staff (staffNo, branchNo, fName, lName,
     position, salary) VALUES ('S654321','B299',
          'Rio', 'Riley','stocker','12.00')


INSERT INTO Staff (staffNo, branchNo, fName, lName,
     position, salary) VALUES ('S123456','B299',
          'Jill', 'Kon','manager','22.00')


/* Jill manages branch B299 */
UPDATE Branch
SET manager='S123456' WHERE branchNo='B299'
```

3. The current design allows any text to be entered as a category. Fix the design so that only the categories: action, comedy, drama, and foreign are allowed.

Using a Category tables:

```
CREATE TABLE Category
(categoryId INTEGER IDENTITY(1,1) PRIMARY KEY,
 category CHAR(30))


CREATE TABLE DVD
(catalogNo CHAR(8) PRIMARY KEY,
 title VARCHAR(200),
 category INTEGER CONSTRAINT category_FK
     FOREIGN KEY REFERENCES Category(categoryId),
 rentalCost SMALLMONEY,
 actors VARCHAR(1000),
 directors VARCHAR(100))
```

Using a CHECK constraint:

```
CREATE TABLE DVD
(catalogNo CHAR(8) PRIMARY KEY,
 title VARCHAR(200),
 category CHAR(10) CONSTRAINT category_CK
     CHECK(category IN {‘Action’, ‘comedy’,
                           ‘drama’, ‘foreign’}),
 rentalCost SMALLMONEY,
 actors VARCHAR(1000),
 directors VARCHAR(100))
```

4. Write a query which gives, for each branch, the name of the branch manager (first and last), along with the title and category of each movie that has been rented from that branch, and the name of the person (first and last) who rented the movie.

Extra Credit: The current design allows any text to be entered as actors and directors. If an actor plays in multiple movies, their name will be repeated in several different DVD tables. If the actor changes their name, it will need to be changed in several tables.

Avoid this repletion by creating an Actor table and an AppearsIn table. Actors are listed once in the Actor table. The AppearsIn table allows several actors to appear in each DVD, and each DVD to have several actors which appear in it.