CSS475: Group Project Report

**Team PB&J:**

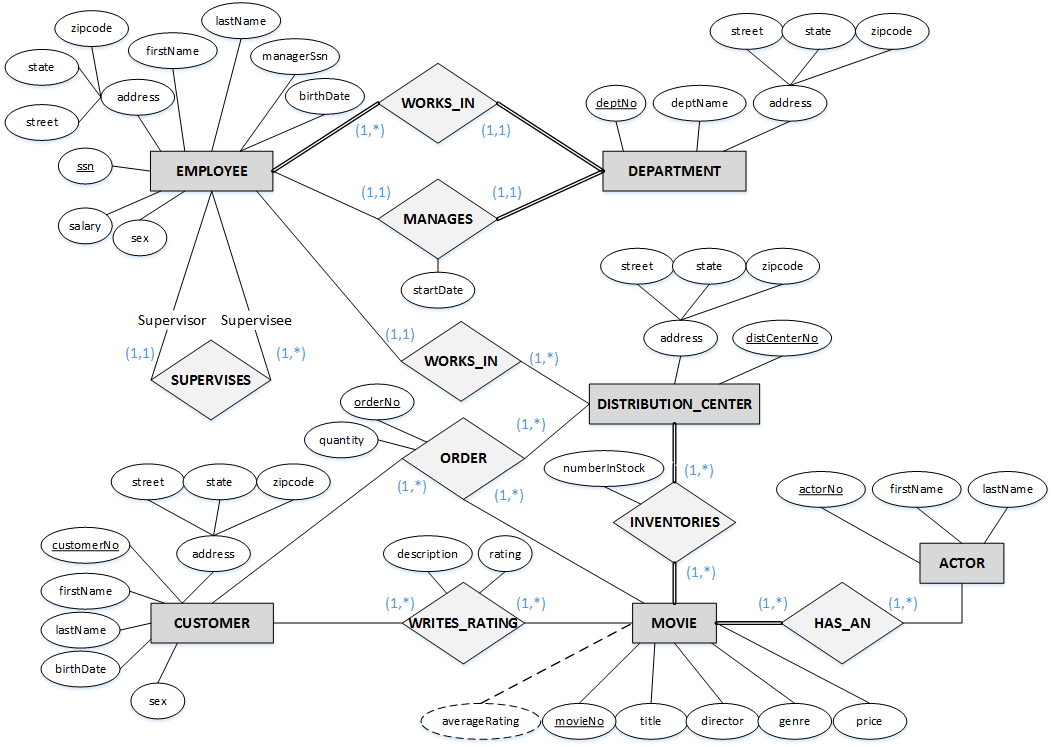
Bartosz Dabkowski

Paul Kessler

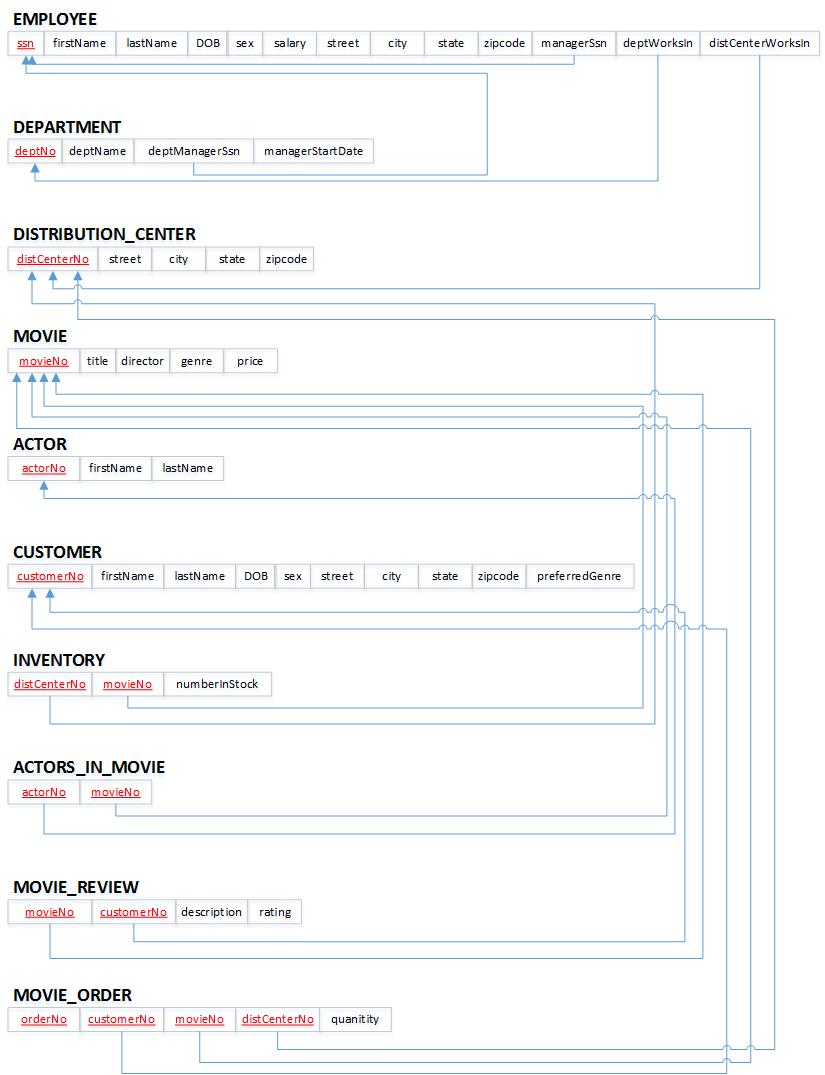
Joscelyn Kim

1. Design Documentation

ER Diagram:



Rational Data Model:



2. Populated Database

SQL Statements:

create table EMPLOYEE

(

ssn CHAR(9) NOT NULL,

firstName VARCHAR(15) NOT NULL,

lastName VARCHAR (15) NOT NULL,

DOB DATE NOT NULL.

sex CHAR(1) NOT NULL CHECK(sex = ‘M’ or sex = ’F’),

salary DECIMAL(10,2) NOT NULL,

street VARCHAR(20) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20) NOT NULL,

zipcode CHAR(5) NOT NULL,

managerSsn CHAR(9),

deptWorksIn INT,

distCenterWorksIn INT,

PRIMARY KEY(ssn),

Constraint EMPFK1 FOREIGN KEY(managerSsn) references EMPLOYEE(ssn) on update cascade on delete set null,

Constraint EMPFK2 FOREIGN KEY(deptWorksIn) references DEPARTMENT(deptNo) on update cascade on delete cascade,

Constraint EMPFK3 FOREIGN KEY(distCenterWorksIn) references DISTRIBUTION\_CENTER(distCenterNo) on update cascade on delete cascade

);

create table DEPARTMENT

(

deptNo INT NOT NULL,

deptName VARCHAR(20) NOT NULL,

deptManagerSsn CHAR(9) NOT NULL,

managerStartDate DATE,

PRIMARY KEY(deptNo),

Constraint DEPTFK1 FOREIGN KEY(deptManagerSsn) references EMPLOYEE(ssn) on update cascade on delete cascade

);

create table DISTRIBUTION\_CENTER

(

distCenterNo INT NOT NULL,

street VARCHAR(20) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20) NOT NULL,

zipcode CHAR(5) NOT NULL,

PRIMARY KEY(distCenterNo)

);

create table MOVIE

(

movieNo INT NOT NULL,

title VARCHAR(20) NOT NULL,

director VARCHAR(15) NOT NULL,

genre VARCHAR(20) NOT NULL,

year CHAR(4) NOT NULL,

price DECIMAL(4,2) NOT NULL,

PRIMARY KEY(movieNo)

);

create table ACTOR

(

actorNo INT NOT NULL,

firstName VARCHAR(15) NOT NULL,

lastName VARCHAR(15) NOT NULL,

PRIMARY KEY(actorNo)

);

create table CUSTOMER

(

customerNo INT NOT NULL,

firstName VARCHAR(15) NOT NULL,

lastName VARCHAR (15) NOT NULL,

DOB DATE NOT NULL.

sex CHAR(1) NOT NULL CHECK(sex = ‘M’ or sex = ’F’),

street VARCHAR(20) NOT NULL,

city VARCHAR(20) NOT NULL,

state VARCHAR(20) NOT NULL,

zipcode CHAR(5) NOT NULL,

preferredGenre VARCHAR(20),

PRIMARY KEY(customerNo)

);

create table INVENTORY

(

distCenterNo INT NOT NULL,

movieNo INT NOT NULL,

numberInStock INT DEFAULT 1,

PRIMARY KEY(distCenterNo,movieNo),

Constraint INVENTFK1 FOREIGN KEY(distCenterNo) references DISTRIBUTION\_CENTER(distCenterNo) on update cascade on delete cascade,

Constraint INVENTFK2 FOREIGN KEY(movieNo) references MOVIE(movieNo) on update cascade on delete cascade

);

create table ACTORS\_IN\_MOVIE

(

actorNo INT NOT NULL,

movieNo INT NOT NULL,

PRIMARY KEY(actorNo, movieNo),

Constraint ACTORFK1 FOREIGN KEY(actorNo) references ACTOR(actorNo) on update cascade on delete cascade,

Constraint ACTORFK2 FOREIGN KEY(movieNo) references MOVIE(movieNo) on update cascade on delete cascade

);

create table MOVIE\_REVIEW

(

movieNo INT NOT NULL,

customerNo INT NOT NULL,

description VARCHAR(200) NOT NULL,

rating INT NOT NULL CHECK(rating <= 5 and rating >= 0),

PRIMARY KEY(movieNo, customerNo),

Constraint MOVIE\_REVFK1 FOREIGN KEY(customerNo) references CUSTOMER(customerNo) on update cascade on delete cascade

);

Create table MOVIE\_ORDER

(

customerNo INT NOT NULL,

movieNo INT NOT NULL,

distCenterNo INT NOT NULL,

orderNo INT NOT NULL,

quantity INT DEFAULT 1,

PRIMARY KEY(customerNo, movieNo, distCenterNo),

Constraint M\_ORDERFK1 FOREIGN KEY(customerNo) references CUSTOMER(customerNo) on update cascade on delete cascade,

Constraint M\_ORDERFK2 FOREIGN KEY(movieNo) references MOVIE(movieNo) on update cascade on delete cascade,

Constraint M\_ORDERFK3 FOREIGN KEY(distCenterNo) references DISTRIBUTION\_CENTER(distCenterNo) on update cascade on delete cascade

);

Table Names:

ACTOR

ACTORS\_IN\_MOVIE

CUSTOMER

DEPARTMENT

DISTRIBUTION\_CENTER

EMPLOYEE

INVENTORY

MOVIE

MOVIE\_ORDER

MOVIE\_REVIEW

3. SQL Query Statements

|  |  |
| --- | --- |
| SQL Statement | Purpose |
| Select \* from MOVIE where genre = ‘Drama’; | Searches movies with a genre of Drama. This function supports searching movies by genre. |
| Select \* from MOVIE where director = ‘Steven Spielberg’ | Searches for all movies with the director ‘Steven Spielberg’. This function supports searching all movies directed by a specific director |
| Select title from MOVIE, MOVIE\_REVIEW  Where MOVIE\_REVIEW.movieNo = MOVIE.movieNo and rating = 5; | Searches for all movie titles with a 5-star rating. This function supports searching all movies with a specific star rating. |
| Select title from MOVIE,ACTOR,ACTORS\_IN\_MOVIE  Where ACTOR.actorNo = ACTORS\_IN\_MOVIE.actorNo and firstName = ‘Brad’ AND lastName = ‘Pitt’ and MOVIE.movieNo = ACTORS\_IN\_MOVIE.movieNo; | Searches for all movie titles starring the actor ‘Brad Pitt’. This function supports searching all movies that a certain actor/actress has played in |
| Select \* from EMPLOYEE where distCenterWorksIn = 555 | Searches for employees who work in distribution center 555. This function supports searching for employees in a specific distribution center |
| Select orderNo from MOVIE\_ORDER, MOVIE where MOVIE.movieNo = MOVIE\_ORDER.movieNo and title = ‘Pulp Fiction’; | Searches order numbers of movie orders of ‘Pulp Fiction’. This function supports searching orders that deal with a specific movie |
| Select firstName, lastName from CUSTOMER where preferredGenre = ‘horror’; | Searches for first and last names of customers who prefer horror movies. This function supports searching for customers who prefer a specific movie genre |
| Select \* from MOVIE where price >= ‘7.99’ AND price <= ’10.00’  Order by price | Searches all movies with a price between $7.99 and $10.00 and displays in ascending order. This function supports searching for movies within a specified price range |
| Select firstName ,lastName ,ssn from EMPLOYEE where state = ‘CA’ | Searches first names, last names, and SSN of employees who live in California. This function supports searching for employees that live in a specific state. |