Cyril Saidane

Pierre Venereau

Aleks Murauskas

Paul Seimandi

Movie Theater Management System Database

1. Attach Modified relational schema Schema from deliverable 1

Below is the deliverable 1 relational model:

Product(productId, price)

Ticket(productId, seatNum) (productId ref Product)

Food Item(productId) (productId ref Product)

Movie(movieId, name, rating, license)

Screening(movieId, screeningNum, time, date) (movieId ref Movie)

Catalog(catalogId)

Employee(eid, name, email, salary, address)

Manager(eid) (eid ref Employee)

Customer(custId, cust\_name, cust\_email)

Cinema(cId, address)

Room(cId, number, capacity) (cId ref Cinema)

We have implemented one change since deliverable one. We added a one to one relation between Cinema and Catalog, allowing simpler queueing to view films that a cinema is showing.

1. Write a Database Schema for the relational schema you have designed using the CREATE TABLE command

CREATE TABLE Product (

productId INTEGER NOT NULL,

price FLOAT NOT NULL,

PRIMARY KEY(productId)

);

CREATE TABLE Customer(

custId INTEGER NOT NULL,

custName VARCHAR(250) NOT NULL,

custEmail VARCHAR(250),

PRIMARY KEY(custId)

);

CREATE TABLE Cinema(

cid INTEGER NOT NULL,

address VARCHAR(250),

PRIMARY KEY(cid)

);

CREATE TABLE Room(

cid INTEGER NOT NULL REFERENCES Cinema(cid),

nbr INTEGER NOT NULL,

Capacity INTEGER,

FOREIGN KEY(cid) REFERENCES Cinema(cid),

PRIMARY KEY(cid,nbr)

);

CREATE TABLE Ticket (

ticketId INTEGER NOT NULL REFERENCES Product(productId),

seatNumber INTEGER,

FOREIGN KEY(ticketId) REFERENCES Product(productId)

);

CREATE TABLE Orders(

custId INTEGER NOT NULL REFERENCES Customer(custId),

productId INTEGER NOT NULL REFERENCES Product(productId),

cid INTEGER NOT NULL REFERENCES Cinema(cid),

quantity INTEGER NOT NULL,

PRIMARY KEY(custId, productId)

);

CREATE TABLE Movie (

movieId INTEGER NOT NULL,

name VARCHAR(250),

rating FLOAT,

license VARCHAR(250),

PRIMARY KEY(movieId)

);

CREATE TABLE Screening(

movieId INTEGER NOT NULL REFERENCES Movie(movieId),

cid INTEGER NOT NULL REFERENCES cinema(cid),

nbr INTEGER NOT NULL,

screeningNum INTEGER NOT NULL,

time VARCHAR(50) NOT NULL,

date DATE NOT NULL,

FOREIGN KEY(movieId) REFERENCES Movie(movieId),

FOREIGN KEY(cid,nbr) REFERENCES room(cid,nbr),

PRIMARY KEY(screeningNum)

);

CREATE TABLE Catalog(

catalogId INTEGER NOT NULL,

cid INTEGER NOT NULL REFERENCES Cinema(cid),

FOREIGN KEY (cid) REFERENCES Cinema(cid),

PRIMARY KEY(catalogID)

);

CREATE TABLE Employee(

eid INTEGER NOT NULL,

cid INTEGER NOT NULL REFERENCES Cinema(cid),

name VARCHAR(50) NOT NULL,

email VARCHAR(250),

salary FLOAT NOT NULL,

address VARCHAR(250),

PRIMARY KEY(eid),

FOREIGN KEY (cid) REFERENCES Cinema(cid)

);

CREATE TABLE Manager(

mid INTEGER NOT NULL REFERENCES Employee(eid),

PRIMARY KEY(mid),

catalogId INTEGER NOT NULL REFERENCES Catalog(catalogId),

FOREIGN KEY(catalogId) REFERENCES Catalog(catalogId)

);

CREATE TABLE FoodItem(

foodId INTEGER UNIQUE NOT NULL REFERENCES Product(productId),

FOREIGN KEY(foodId) REFERENCES Product(productId)

);

CREATE TABLE Stock(

mid INTEGER NOT NULL REFERENCES Manager(mid),

FOREIGN KEY(mid) REFERENCES Manager(mid),

foodId INTEGER NOT NULL REFERENCES FoodItem(foodId),

Foreign key(foodId) REFERENCES FoodItem(foodId),

Quantity INTEGER,

PRIMARY KEY(mid,foodId)

);

In an attached CSV file Described Tables, we have the responses of the command

SELECT \* FROM information\_schema.columns WHERE table\_name = '<tablename>';

1. Execute five INSERT commands to insert tuples into one of your relations

Example Insert statements:

INSERT INTO "ticket" (ticketid,seatnumber) VALUES

(50,118),(55,116),(60,62),(65,72),(70,73),(75,111),(80,1),(85,55),(90,64),(95,126);

INSERT INTO "customer" (custid,custname,custemail) VALUES

(5286,Shafira Guerra,[Quisque@sodales.co.uk](mailto:Quisque@sodales.co.uk)),(5317, Ina Chaney,placerat.eget@ullamcorpereueuismod.org),(5348,Reese Tanner,id.risus@duinec.ca);

INSERT INTO "product" (productid,price) VALUES (50,14),(55,8),(60,3),(65,26),(70,27),(75,28),(80,5),(85,11),(90,7),(95,21);

INSERT INTO “fooditem" (foodid) VALUES (300),(305),(310),(315),(320),(325),(330),(335),(340),(345);

INSERT INTO "manager" (mid,catalogid) VALUES (1925,4591),(1932,4612),(1939,4633),(1946,4654),(1953,4675),(1960,4696),(1967,4717),(1974,4738),(1981,4759),(1988,4780);

The results from the Select From Statements can be found in the attached CSV file Select Froms

1. Fill all tables with meaningful information so queries mean something

Please read the CSV file Select Froms to see the first 10 returns per table.

1. Write five queries on your database, using select from where construct:

The results of the Queries will be stored in CSV file Queries.

Query 1: Find the employees of a Specific Cinema, given the address

SELECT name FROM employee where cid =( Select cid FROM cinema WHERE address ='3832 Rutrum. Ave')

Query 2: Find the names of customers that placed an order at a given cinema when given the cinemaID.

SELECT custname FROM customer where custid =( Select custid FROM orders WHERE cid =1357)

Query 3: Find the price of a product that was part of an order at a given cinema when given the cinemaID.

SELECT price FROM product where productid =( Select productid FROM orders WHERE cid =1357)

Query 4: Find the time of all screenings at a cinema given the address.

SELECT time From screening where cid = (Select cid FROM cinema WHERE address ='3832 Rutrum. Ave')

Query 5: Find the seat number of all tickets on orders at a cinema, given the address.

Select seatnumber from ticket where ticketid = (Select productid from orders where cid = (Select cid FROM cinema WHERE address ='3832 Rutrum. Ave' ))

1. Write four data modification commands
2. Create two views on top of your database schema
3. Add two CHECK constraints to relations of your database schema