

## Table creation statements

```
create table videotape (  
  videoID int NOT NULL,  
  year int NOT NULL,  
  title varchar(100) NOT NULL,  
  FOREIGN KEY(title, year) references movie(title, year),  
  PRIMARY KEY(videoID)  
)
```

```
create table movie (  
  title varchar(100) NOT NULL,  
  year int NOT NULL,  
  cat varchar(100) NOT NULL,  
  introduction varchar(255),  
  directors varchar(255),  
  producers varchar(255),  
  PRIMARY KEY(title, year)  
);
```

### Duplication issue with orderID.

```
create table orderItem (  
  orderID varchar(10) NOT NULL,  
  videoID int NOT NULL,  
  FOREIGN KEY(videoID) references videotape(videoID),  
);
```

```
CREATE TABLE customerOrder (  
  orderID varchar(10) NOT NULL,  
  dayOrdered varchar(10) NOT NULL,  
  shippingDestination varchar(40),  
  period int,  
  status varchar(10),  
  returnTrackingInformation varchar(40),  
  email varchar(50) NOT NULL,  
  PRIMARY KEY(orderID),  
  FOREIGN KEY(email) REFERENCES Customer(email)  
);
```

- Day ordered could be a DateTime data type if that exists, timestamp
- Status could be an ENUM type?

```
CREATE TABLE Customer (  
  email varchar(50) NOT NULL,  
  password varchar(50) NOT NULL,  
  loyaltyPoints int NOT NULL,  
  creditCardNum varchar(15),  
  PRIMARY KEY (email)  
);
```

```
CREATE TABLE Manager (  
  email varchar(50) NOT NULL,  
  password varchar(50) NOT NULL,  
  PRIMARY KEY (email)  
);
```

```
CREATE TABLE Rating (  
  criticName varchar(50) NOT NULL,  
  title varchar(100) NOT NULL,  
  year int NOT NULL,  
  PRIMARY KEY(criticName),  
  FOREIGN KEY(title, year) REFERENCES movie(title, year)  
)
```

---

## Pre-inserted data

### Movies

```
insert into movie values (  
  'The Thing',  
  1982,  
  'Horror',  
  'Kurt Russel and friends fight an alien',  
  'John carpenter',
```

```
'David Foster'  
);
```

```
insert into movie values (  
'Alien',  
1979,  
'Horror',  
'Sigourney Weaver and friends fight an alien',  
'Ridley Scott',  
'Walter Hill'  
);
```

## **Videotapes**

```
insert into videotape values (  
0,  
1982,  
'The Thing'  
)
```

```
insert into videotape values (  
1,  
1979,  
'Alien'  
)
```

```
insert into videotape values (  
2,  
1979,  
'Alien'  
)
```

## **Customers**

note: @ symbol was rejected may need to tweak something here

```
insert into customer values (  
'songwangatgmail.com',
```

```
'p455w0rd',  
0,  
'1234-1234-1234'  
);
```

## customerOrders

```
insert into customerOrder values (  
0,  
'October 31st 2020',  
'200 Crescent Avenue, Toronto Ontario',  
20,  
'Unpaid',  
'Unspecified',  
'songwangatgmail.com'  
)
```

- Customer renting some movies for 20 days
- Need to also create separate orderItems for this order
- ID is 0 for the orderID

## orderItems

- adds two video tapes to song wangs order

```
insert into orderItem values (  
0,  
0  
);
```

```
insert into orderItem values (  
0,  
1  
);
```

---

## Useful queries

1. Retrieve all the movies the service has regardless of the number of tape copies.

```
select * FROM movie group by title;
```

Returns

title	year	cat	introduction	directors	producers
The Thing	1982	horror	introduction	directors	producers
Alien	1979	horror	..	..	..

2. Retrieve the number of copies of a certain film regardless of rental status.

```
select count(title) FROM videotape WHERE title='Alien' AND year='1979'
```

3. To retrieve all orderItems from an order.

```
SELECT * FROM orderItem WHERE orderID = [orderID];
```

4. Retrieve all available copies of films (everything that isn't rented)

```
select t1.title, t1.year, t1.videoID FROM videotape t1 WHERE videoID NOT IN (SELECT videoID FROM orderItem);
```

- This query is inefficient but works

5. Check if there is an available copy of a certain film. In this case, checks if there is a copy of alien.

```
select t1.title, t1.year, t1.videoID FROM videotape t1
WHERE t1.title = 'Alien' AND t1.year = '1979' AND videoID
NOT IN (SELECT videoID FROM orderItem);
```

- This query is inefficient but works
- Returns all available video IDs of alien

6. Return all available movies with the number of available copies.

```
select t1.title, t1.year, COUNT(title) FROM videotape t1
WHERE t1.videoID NOT IN (SELECT videoID from orderItem)
GROUP BY t1.title, t1.year;
```

title	year	count
Alien	1979	1