**Project Worksheets**

Every student must work by their own in every worksheet, then use it as a guidelines when meeting and discussing with project team members.

These worksheets are optional. There is no need to submit them to Carmen (only with the project final report, one set per team)

Rubric: Worksheets will be graded by completeness (final project report)

* All the questions must be completely answered, otherwise points will be taken off
* When asking for some examples or cases, provide at least 3-5 cases, unless something more specific is specified.
* Answers must be reasonable and be consistent with the question asked

**Worksheet HW-00 – DB Concepts** (Submit to Carmen)

1. What tables and fields might need to be included in the database?

CArtists: First Name, Last Name, gender, actor(0 for NO ,1 for Yes), singer(0 for No, 1 for Yes), Artist ID

Albums: title, genres, release date, Album ID, Artist ID, Publisher ID

Track: Track ID, Title, Album ID, Length

Video: title, Artist ID, genres, content rating, release year, lengths, Video ID, director ID

Item: Item ID ID (include Album ID (start with A) or Video ID (start with V))

Person: Person ID, First Name, Last Name, Phone Number, gender, DOB,email



Library card: Card ID, Registering Date, Lost/Damage

Transaction: Transaction ID, Check-out Date, Item ID, Location, Return Date, Expected Due Date, Card ID, status mer

Music Publisher( Publisher ID, name)

Director(Director ID, name) – type, location

* Publishers – name, location
* Authors – nam
* e

1. What informal queries and update operations might users expect?

* When we add a new album to our database
  + We need to add new album entity with title, genres, artist id( if have), publisher ID (if have), release date, and assigning a unique album ID.
* Add a new publisher if the library has albums from a new publisher
  + Add new publisher to the publisher entity, with name of publisher
  + Assigning a unique ID to this new publisher
* Add a new director if the library has movies from a new director

Add new director to the director entity with unique director ID, and name of new director

1. What kind of reports might be desired by the users?

* Customer preference summary
* The report of growth of new customer who register a library card
* The List of popular video/album
* Retrieve a list of checkouts of a specific day
* Retrieve a list of checkouts that due today
* Retrieve a list of albums that published by a specific publisher
* Retrieve a list of movies that directed by a specific director

1. Give some examples of integrity constraints that you think can apply to the database.

Domain constraints: Artists’ name can’t include number.

Entity integrity constraints: AlbumID can’t be null because it is a Primary key.

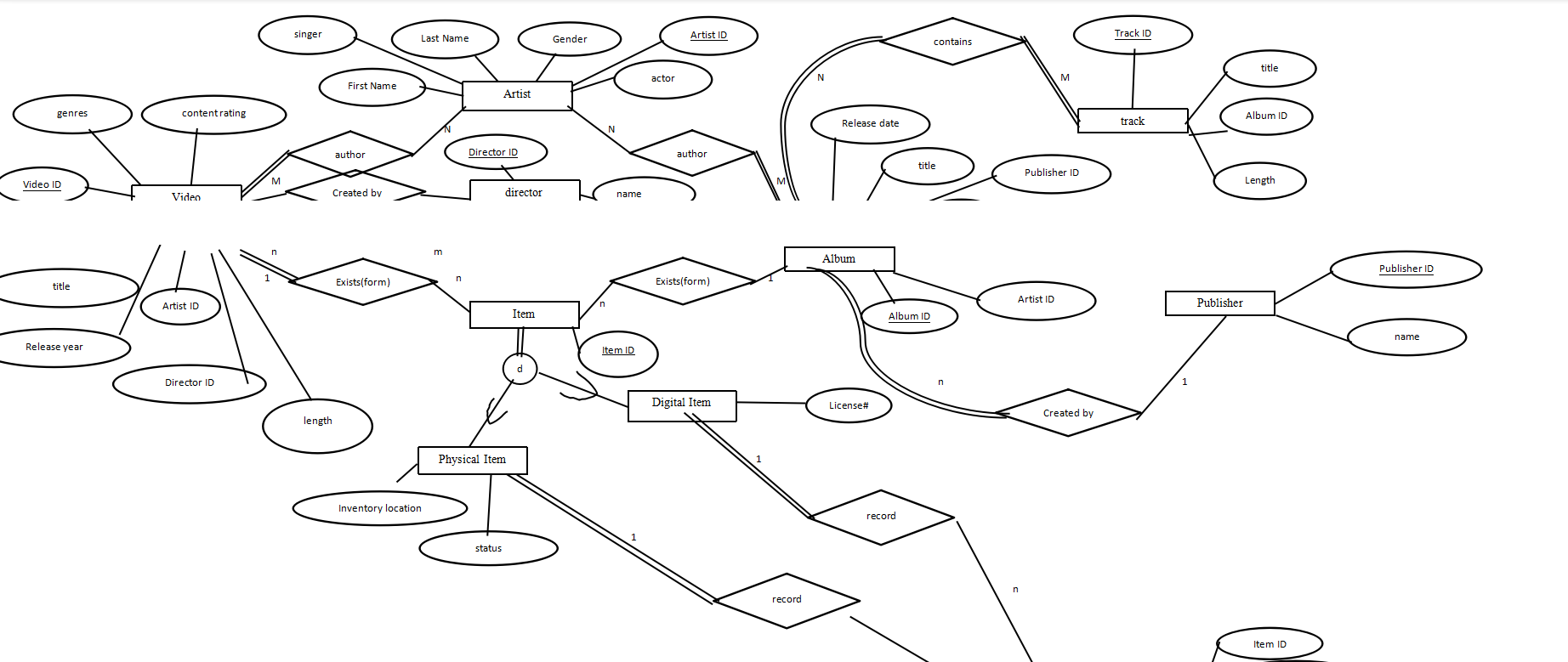
Key constraints: AlbumID can’t have two same value.

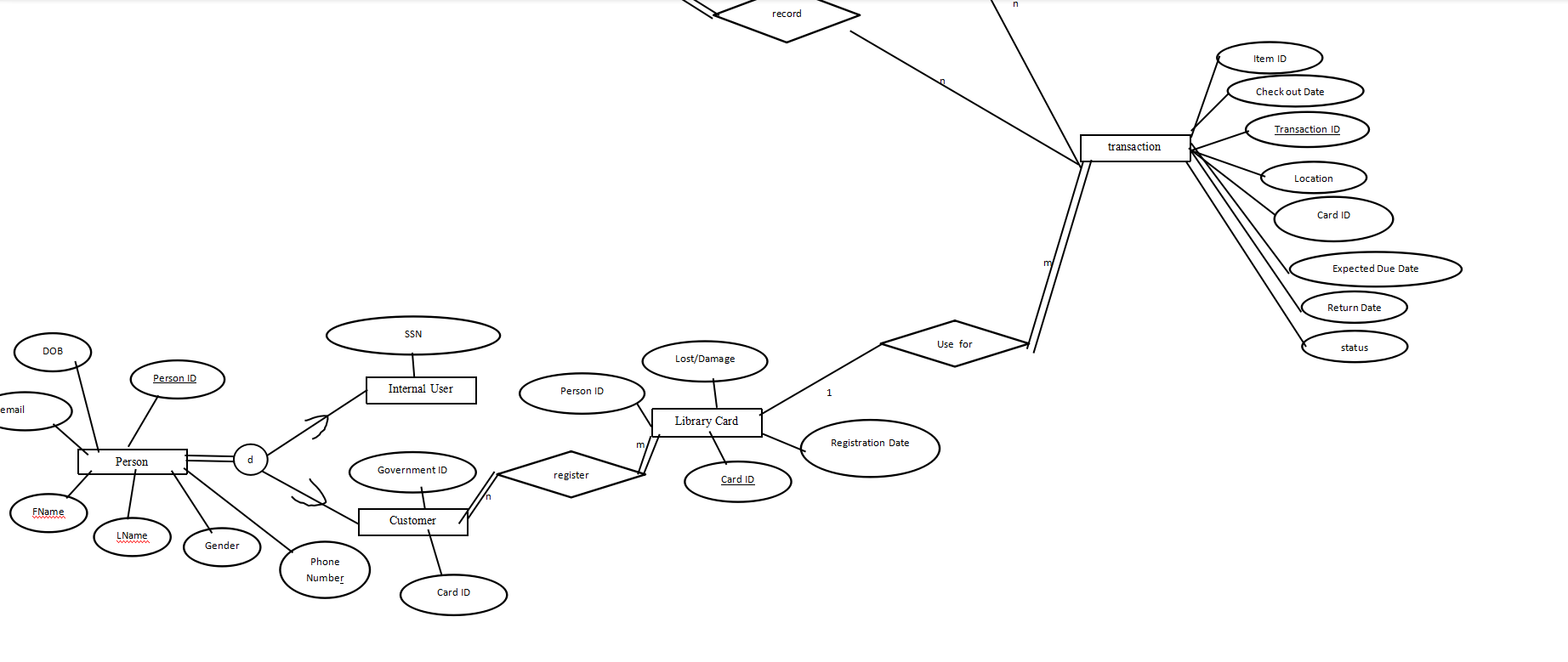
1. If a patron changes their address, what would need to be updated? What other changes could affect the data and relationships between the tables?

Address column in person table need to be updated

**Worksheet HW-01 Entity-Relationship (ER)** (Submit to Carmen)

Provide an ER diagram for your database

****

****

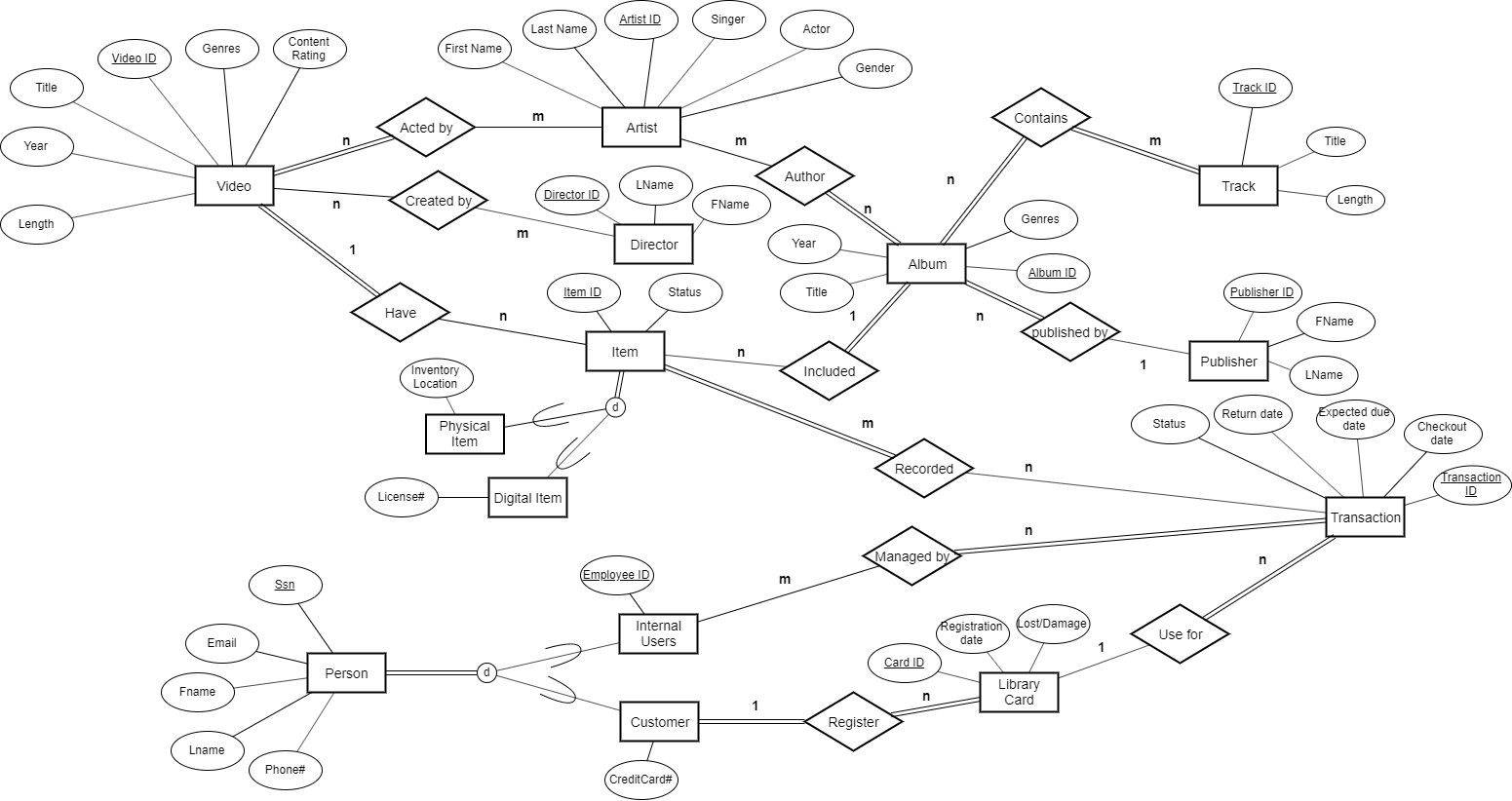
**Worksheet HW–02 Enhanced ER (EER)** (Submit to Carmen)

Are there any hierarchical relationships in the database? If so, what are they? Think of at least two hierarchical relationships you can add to the database.

Physical item and digital item are subclass of Item table, disjointedness

Internal user and external user are subclass of Person table, disjointedness

Refine your ERD to show these new relationships.



**Worksheet HW–03 EER to Relational Mapping** (Submit to Carmen)

Map your EER to a relational schema and show all primary and foreign keys.

Artists: Artist ID, FName, LName, gender, actor, singer

Recorded: Item ID(FK), TransactionID(FK)

Managed by: SSN(FK), TransactionID(FK)

Acted by: Artist ID(FK), Video ID(FK)

Created by: Director ID(FK), Video ID(FK)

Author: Album ID(FK), Artist ID(FK)

Contains: Track ID(FK), Album ID(FK)

Albums: Album ID, title, genres, release year, Artist ID(FK), Publisher ID(FK)

Track: Track ID, Title, Length

Video: Video ID, title, Artist ID(FK), genres, content rating, release year, lengths

Item: Item ID, status, product ID (FK), Video ID (FK)

Physical Item: Item ID (FK), Inventory location

Digital Item: Item ID (FK), licenses#

Person: SSN, First Name, Last Name, Phone Number, email

Internal user: SSN (FK), employeeID

Customer: SSN (Fk), credit card

Library card: Card ID, Registering Date, Lost/Damage, SSN (FK)

Transaction: Transaction ID, Check-out Date, Item ID(FK), Return Date, Expected Due Date, Card ID(FK), status

Publisher: Publisher ID, FName, LName

Director: Director ID, FName, LName

**Worksheet HW– 04 Relational Data Model** (Submit to Carmen)

Propose ten Update operations that could be applied directly to the database. For each discuss *all* integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints:

|  |  |  |
| --- | --- | --- |
|  | Operation | Integrity constraints |
|  | Add a user into database | Key constraints: SSN of the user should be unique.  Domain constraints: SSN can’t include letter.  Entity constraints: SSN can’t be null |
|  | Delete an artist from database | Referential integrity: some albums will not have artists and their foreigner key will be null. |
|  | update a transaction | Domain constraints: checkout date can’t before the check-in date. |
|  | Add an artist | Key constraints: ArtistID of the user should be unique.  Entity constraints: ArtistID can’t be null |
|  | Delete an Item | Referential integrity: some transaction will not have item and their foreigner key will be null. |
|  | Update a track | Key constraints: TrackID of the user should be unique.  Entity constraints: TrackID can’t be null |
|  | Add a video | Key constraints: VideoID of the user should be unique.  Entity constraints: VideoID can’t be null |
|  | Add a library card | Key constraints: CardID of the user should be unique.  Entity constraints: CardID can’t be null |
|  | Add a publisher | Key constraints: publisherID of the user should be unique.  Entity constraints: PublisherID can’t be null |
|  | Add a director | Key constraints: DirectorID of the user should be unique.  Entity constraints: DirectorID can’t be null |

**Worksheet HW– 05 Relational Algebra** (Submit to Carmen)

Show schema for your database that you have so far. Indicate relationships, and foreign keys using the “arrow” notation from the book

Artists: Artist ID, FName, LName, gender, actor, singer

↑

Acted by: Artist ID(FK), Video ID(FK)

Created by: Director ID(FK), Video ID(FK)

Author: Album ID(FK), Artist ID(FK)

Contains: Track ID(FK), Album ID(FK)

Albums: Album ID, title, genres, release year, Artist ID(FK), Publisher ID(FK)

Track: Track ID, Title, Album ID(FK), Length

Video: Video ID, title, Artist ID(FK), genres, content rating, release year, lengths, director ID(FK)

Item: Item ID, product ID (FK)

Physical Item: Inventory location, status

Digital Item: licenses#

Person: SSN, First Name, Last Name, Phone Number, gender, DOB, email

Internal user: SSN

Customer: SSN

Library card: Card ID, Registering Date, Lost/Damage, SSN

Transaction: Transaction ID, Check-out Date, Item ID(FK), Location, Return Date, Expected Due Date, Card ID(FK), status

Publisher: Publisher ID, FName, LName

Director: Director ID, FName, LName

Specify the following queries in relational algebra.

1. Find all the albums by ARTIST that were released before YEAR.
2. Give all the movies and dates for check outs made by a particular patron.

1. List all the albums with less than 5 copies available to check out.

**Worksheet HW–06 Normalization** (Submit to Carmen)

Is your database fully normalized? Why or why not?

No, because Transaction table is not fully normalized

If not, restate your schema in 3NF

Artists: Artist ID, FName, LName, gender, actor, singer

Acted by: Artist ID(FK), Video ID(FK)

Created by: Director ID(FK), Video ID(FK)

Author: Album ID(FK), Artist ID(FK)

Contains: Track ID(FK), Album ID(FK)

Albums: Album ID, title, genres, release year, Artist ID(FK), Publisher ID(FK)

Track: Track ID, Title, Album ID(FK), Length

Video: Video ID, title, Artist ID(FK), genres, content rating, release year, lengths, director ID(FK)

Item: Item ID, product ID (FK)

Physical Item: Inventory location, status

Digital Item: licenses#

Person: SSN, First Name, Last Name, Phone Number, gender, DOB, email

Internal user: SSN

Customer: SSN

Library card: Card ID, Registering Date, Lost/Damage, SSN

Transaction: TransactionID, ItemID, returned date, CardID, Status, Check-out date

due\_date: Check-out date, expected return date

Publisher: Publisher ID, FName, LName

Director: Director ID, FName, LName

**Worksheet HW– 07 SQL** (Submit to Carmen)

1. What are the referential integrity constraints that should hold on the schema?

Artist ID reference Artist

VIdeoID reference Video

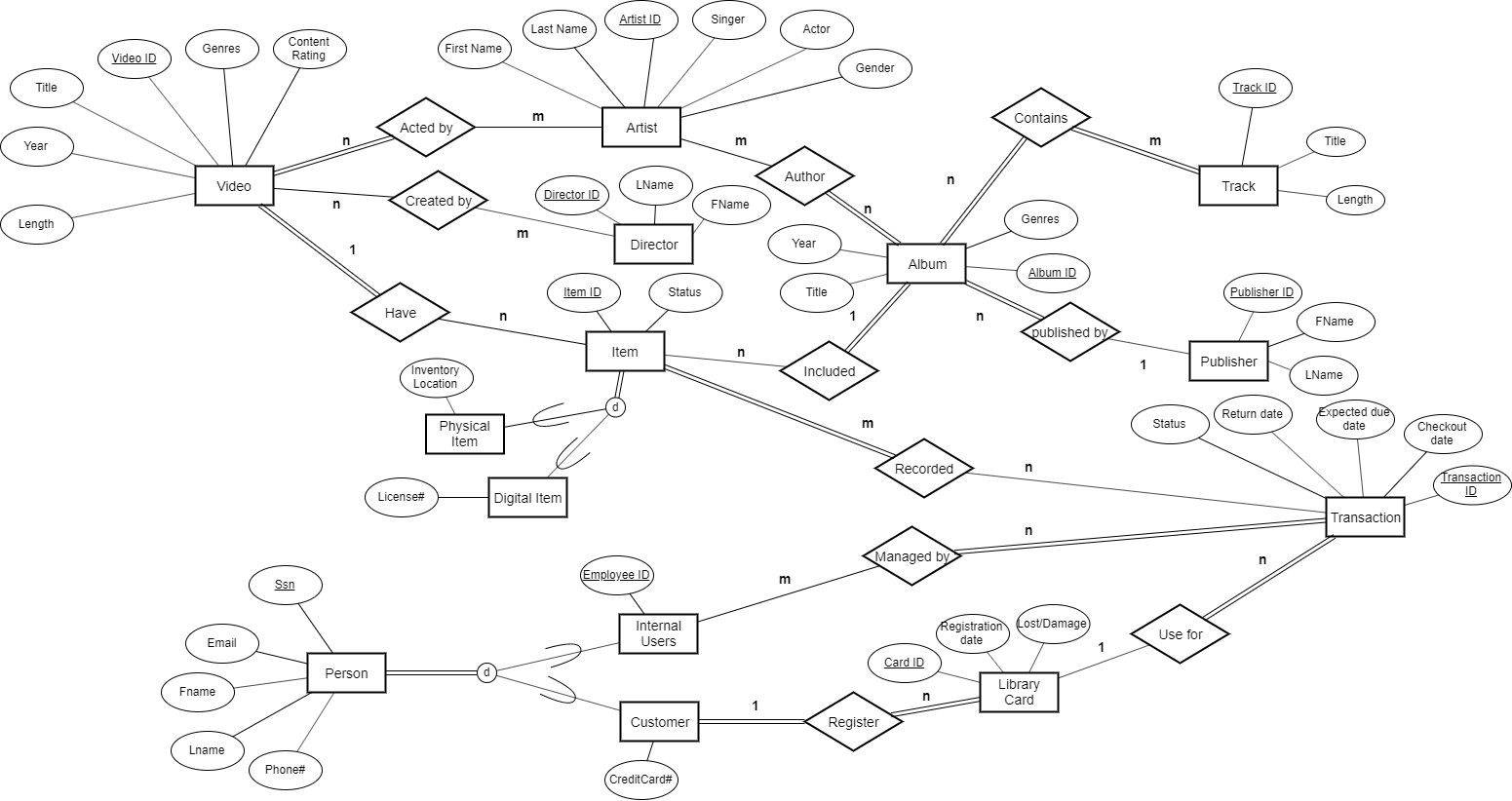
DirectorId reference Director

AlbumID reference Album

PublisherID reference publisher

1. What tables do you think should be included? What fields? Provide a copy of your relational diagram from Worksheet 04 (or an updated version if you have changed it since then).

Artist, Video, Director, Album, publisher, director, Acted by, created by, author contains.



1. Write appropriate SQL DDL statements to define the database.

create table Video(

VideoID int not null,

Title varchar(50) not null,

Genre varchar(50) not null,

Rating double,

Year YYYY,

Length int not null,

primary key(VideoID)

)

;

create table Artists(

ArtistID int not null,

name varchar(50) not null,

Actor boolean not null,

Singer boolean not null,

Group\_ boolean not null,

primary key(ArtistID)

)

;

create table ActedBy(

VideoID int not null,

ArtistID int not null,

primary key(VideoID,ArtistID),

foreign key(VideoID) references Video(VideoID) on update cascade on delete cascade,

foreign key(ArtistID) references Artists(ArtistID) on update cascade on delete cascade

)

;

create table Publisher(

PublisherID int not null,

FirstName varchar(50) not null,

LastName varchar(50) not null,

primary key(PublisherID)

)

;

create table Albums(

AlbumID int not null,

Title varchar(50) not null,

Genre varchar(50) not null,

Year YYYY not null,

PublisherID int not null,

primary key(AlbumID)

foreign key(PublisherID) references Publisher(PublisherID) on update cascade on delete cascade

)

;

create table Author(

AlbumID int not null,

ArtistID int not null,

primary key(AlbumID,ArtistID)

foreign key(AlbumID) references Albums(AlbumID) on update cascade on delete cascade,

foreign key(ArtistID) references Artists(ArtistID) on update cascade on delete cascade

)

;

create table Director(

DirectorID int not null,

Name varchar(50) not null,

primary key(DirectorID)

)

;

create table CreateBy(

VideoID int not null,

DirectorID int not null,

primary key(VideoID,DirectorID),

foreign key(VideoID) references Video(VideoID) on update cascade on delete cascade,

foreign key(DirectorID) references Director(DirectorID) on update cascade on delete cascade

)

;

create table Status(

StatusID int not null,

Description varchar(50) not null,

primary key(StatusID)

)

;

create table Item(

ItemID char(10) not null,

statusID int not null,

AlbumID int,

VideoID int,

primary key(ItemID)

foreign key(AlbumID) references Albums(AlbumID) on update cascade on delete cascade

foreign key(VideoID) references Video(VideoID) on update cascade on delete cascade

foreign key(statusID) references Status(StatusID) on update cascade on delete cascade

)

;

create table PlysicalItem(

ItemID char(10) not null,

InventoryLocation varchar(50) not null,

primary key(ItemID),

foreign key(ItemID) references Item(ItemID) on update cascade on delete cascade

)

;

create table digitalItem(

ItemID char(10) not null,

LicenseNum varchar(50) not null,

primary key(ItemID),

foreign key(ItemID) references Item(ItemID) on update cascade on delete cascade

)

;

create table Person(

SSN char(9) not null,

FirstName varchar(50) not null,

LastName varchar(50) not null,

PhoneNumber char(10) not null,

Email varchar(60) not null,

primary key(SSN)

)

;

create table internalUser(

SSN char(9) not null,

EmployeeID int not null,

primary key(EmployeeID),

foreign key(SSN) references Person(SSN) on update cascade on delete cascade

)

;

create table Customer(

SSN char(9) not null,

creditCardNum char(16) not null,

primary key(SSN),

foreign key(SSN) references Person(SSN) on update cascade on delete cascade

)

;

create table LibararyCard(

CardID int not null,

RegisteringDate DATE not null,

Activation Boolean not null,

SSN char(9) not null,

primary key(CardID),

foreign key(SSN) references Person(SSN) on update cascade on delete cascade

)

;

create table TransactionRecord(

TransactionID int not null,

CheckOutDate DATE not null,

ReturnDate DATE not null,

ExpectedDueDate Date not null,

cardID int not null,

primary key(TransactionID)

foreign key(cardID) references LibararyCard(CardID) on update cascade on delete cascade

)

;

create table ManagedBy(

EmployeeID int not null,

TransactionID int not null,

primary key(EmployeeID,TransactionID),

foreign key(TransactionID) references TransactionRecord(TransactionID) on update cascade on delete cascade,

foreign key(EmployeeID) references internalUser(EmployeeID) on update cascade on delete cascade

)

**Worksheet HW– 08 INSTALL SQLite** (Submit to Carmen)

1. Install a SQLite console manager. Download and execute SQLite-Studio to use graphical interfaces. Alternatively, you could download sqlite3 and execute from a text (terminal) window if you like cli interfaces.
2. Go over the tutorial and get familiar with the interface and menu/commands. Create a simple database with at least three tables from with few attributes using DDL statements. You should come up with your own database (you are free to imagine a small database for any purpose).

create table Publisher(

PublisherID int not null,

FirstName varchar(50) not null,

LastName varchar(50) not null,

primary key(PublisherID)

)

;

create table Albums(

AlbumID int not null,

Title varchar(50) not null,

Genre varchar(50) not null,

Year YYYY not null,

PublisherID int not null,

primary key(AlbumID)

foreign key(PublisherID) references Publisher(PublisherID) on update cascade on delete cascade

)

;

create table Video(

VideoID int not null,

Title varchar(50) not null,

Genre varchar(50) not null,

Rating double,

Year YYYY,

Length int not null,

primary key(VideoID)

)

;

1. Using SQL queries, insert different rows with values into each table and practice some basic queries to select some of the rows. Use joins.

Insert Publisher (PublisherID, FirstName, LastName)

Values (123, A,B)

Insert Album (AlbumID, Title, Genre, Year)

Values ( 123, AA, pop, 2000)

Insert Video (VideoID, Title, Genre, Rating, Year, Length)

Values (123, AA, documentary, 4.0m 2002, 327813)

Select PublisherID

From Publisher

Where FirstName = F

Select AlbumID

From Album

Where Year = 2019

Select VideoID

From Video

Where Rating = 5.0

Submit the database file that you created and screen copies of the SQL commands you use.

**Worksheet HW– 09 More SQL** (Submit to Carmen)

Specify the following queries and show a sample results for the database.

1. Find all the albums by an arbitrary artist that were released before an arbitrary year.
2. SQL

Select T.Title

From Artists AS art, Author AS aut, Albums As alb, Track As T

WHERE art.ArtistID = aut.ArtistID AND aut.AlbumID = alb.AlbumID

AND T.AlbumID = alb.AlbumID AND art.name = "Accept"

AND alb.Year < 2010

1. Sample Result

Restless and Wild

Princess of the Dawn

Fast As a Shark

Balls to the Wall

1. Give all the albums and checkout dates for check outs made by a particular patron.
2. SQL

Select a.Title, d.CheckoutDate

From Albums AS a, Item AS b, recorded AS c, Transactionrecord AS d, LibraryCard AS e, Person AS f

Where a.albumID = b.AlbumID AND b.ItemID = c.ItemID AND c.TransactionID = d.TransactionID

And e.CardID = d.CardID AND e.Ssn = f.Ssn And f.SSN = 614411001

1. Sample Result

Big Ones Jan/2/2012

Let There Be Rock Jan/3/2012

Jagged Little Pill Jan/1/2013

For Those About To Rock We Salute You Jan/2/2013

Restless and Wild Jan/3/2013

Big Ones Jan/1/2014

Jagged Little Pill Jan/2/2014

Let There Be Rock Jan/2/2016

1. List all the albums with less than 5 copies available to check out.
2. SQL

Select \*

from

(select Albums.Title, Item.ItemID, Count(Item.ItemID) AS Quantity

From Item, Albums

Where Item.AlbumID = Albums.AlbumID

Group By Item.AlbumID)

where quantity < 5

1. Sample Result

For Those About To Rock We Salute You a15 2

Let There Be Rock a16 2

Balls to the Wall a17 1

Restless and Wild a18 1

Big Ones a19 2

Jagged Little Pill a20 2

Back to Black a24 1

Warner 25 Anos a25 1

Plays Metallica By Four Cellos a26 1

1. List all the patrons with an overdue album checked out, ordered by most overdue.
   1. SQL

Select Person.FirstName, Person.LastName

From person, TransactionRecord, Albums, Item, Recorded

Where Item.AlbumID = Albums.AlbumID and Item.ItemID = Recorded.ItemID and TransactionRecord.TransactionID = Recorded.TransactionID

Having returndate < expectedDuedate

* 1. Sample Result

1. List all tracks that have a file size larger than the average file size.
   1. SQL

Select Track.Title

From Track

Having Track.Length > avg(Track.Length)

* 1. Sample Result

Breaking The Rules

C.O.D.

Evil Walks

For Those About To Rock (We Salute You)

Inject The Venom

Let's Get It Up

Night Of The Long Knives

Put The Finger On You

Snowballed

Spellbound

Bad Boy Boogie

Dog Eat Dog

Go Down

Hell Ain't A Bad Place To Be

Let There Be Rock

Overdose

Problem Child

Whole Lotta Rosie

Balls to the Wall

Fast As a Shark

Princess of the Dawn

Restless and Wild

Amazing

Angel

Blind Man

Crazy

Cryin'

Deuces Are Wild

Dude (Looks Like A Lady)

Eat The Rich

Janie's Got A Gun

Livin' On The Edge

Love In An Elevator

Rag Doll

The Other Side

Walk On Water

What It Takes

All I Really Want

Forgiven

Hand In My Pocket

Head Over Feet

Ironic

Mary Jane

Not The Doctor

Perfect

1. List all patrons who have been issued multiple library cards.
   1. SQL

Select distinct person.FirstName

From Person , LibraryCard as L

Where person.ssn = l.ssn

having count(l.ssn) > 2

* 1. Sample Result

Madison

Brandon

Mia

Isaiah

Elizabeth

Owen

Natalie

Connor

Alexis

Jose

Grace

Julian

Alyssa

1. List all patrons with one or more albums checked out with a library card that is now inactive.
   1. SQL

Select Person.FirstName, Person.LastName

From Person, LibraryCard, Item, Recorded, TransactionRecord, Albums

Where Person.SSN = LibraryCard.SSN And Item.AlbumID = Albums.AlbumID and Item.ItemID = Recorded.ItemID and TransactionRecord.TransactionID = Recorded.TransactionID and LibraryCard.Activation = "False"

Having count(Albums.ALbumID) > 0;

* 1. Sample Result

1. List the top 10 albums (physical or digital), ordered by number of check outs.
   1. SQL

Select Albums.Title,count(t.transactionID) as c

From Albums, Item, TransactionRecord as t, Recorded as r

where albums.AlbumID = item.AlbumID and t.transactionID = r.transactionID and item.ItemID = r.itemID

group by Albums.AlbumID

Order by c desc

limit 10;

* 1. Sample Result

For Those About To Rock We Salute You 3

Let There Be Rock 3

Big Ones 3

Jagged Little Pill 3

Restless and Wild 2

Balls to the Wall 1

**Worksheet HW– 10 Indexing and Transactions**

1. What indexes are supported in SQlite?

Hash-based indexes

Tree-based indexes

1. How do you create a transaction in SQLite?
2. Begin Transaction
3. Add read/write operations
4. If error then rollback
5. Commit
6. End transaction
7. Define a useful transaction for your database and create that transaction in your database

BEGIN TRANSACTION

NEW\_Track INSERT INTO Track VALUES ('123', 'AA',’BBB’, 1000);

IF error THEN GO TO UNDO;

END IF;

UPDATE Track SET Length = Length + 1000

WHERE TrackID='123’;

IF error THEN GO TO UNDO;

END IF;

COMMIT;

GO TO FINISH;

UNDO:

ROLLBACK;

FINISH:

END TRANSACTION;

**For the final project submission (What to turn in.)**

1. For the final project submission, each team need to compile one set of these worksheets to include in that final report.