

CSE 3241 Project Checkpoint 01

Entities and Relationships

Names (team members): Sruti Chigurupati, Elizabeth Gilbert, Trey Hakanson, Maggie Baumgardner

Date: 2/1/18

In a **NEATLY TYPED** document (Upload it to Canvas), provide the following:

- Based on the requirements given in the project overview, list the entities to be modeled in this database. For each entity, provide a list of associated attributes.
 - Album: title, release date, copies
 - Media: ID, type, status
 - Artist: name, ID number
 - Track: title, track number, length
 - Genre: title
 - Person: first name, last name
 - Patron: card number, card activation date
 - Employee: employee ID, salary, start date
- Based on the requirements given in the project overview, what are the various relationships between entities? (For example, "ARTIST entities author ALBUM entities", "ALBUM entities contain TRACK entities", etc.)
 - 1 ALBUM has n ARTIST
 - 1 ALBUM contains n TRACK
 - 1 ALBUM has n MEDIA
 - 1 TRACK has n ARTIST
 - 1 GENRE has n ALBUM
 - 1 PATRON checks out n ALBUM
- Propose at least two additional entities that it would be useful for this database to model beyond the scope of the project requirements. Provide a list of possible attributes for the additional entities and possible relationships they may have with each other and the rest of the entities in the database. Give a brief, one sentence rationale for why adding these entities would be interesting/useful to the stakeholders for this database project.
 - Reviews**
 - Attributes*: number of stars, title, description
 - Relationship*: album has, patron reviews
 - Why*: This helps future patrons to better understand the music available from their library.
 - Feedback**
 - Attributes*: description, date, category
 - Relationship*: given by patron
 - Why*: This helps employees better the library as a whole.
- Give at least four examples of some informal queries/reports that it might be useful for this database might be used to generate. Include one example for each of the additional entities you proposed in question 3 above.
 - List of music by artist/genre
 - List of available music
 - Currently checked out by a patron

- Feedback given in one category
 - Reviews for a given album/artist/genre
5. Suppose we want to add a new artist to the database. How would we do that given the entities and relationships you've outlined above? Given your above description, is it possible to add a new artist to your database without knowing the title of any albums they have released? If not, revise your model to allow for artists to be added as separate entities.
- An artist has its own id, and thus is not a weak entity. Total participation is not required in any of the connected relationships, so an artist with no content available yet is possible.
6. Determine at least three other informal update operations and describe what entities would need to have attributes altered and how they would need to be changed given your above descriptions. Include one example for each of the additional entities you proposed in question 3 above.
- Update checkout status of a album
 - Adding a new album
 - Adding a new patron
 - Adding a new review
 - Adding a new feedback
7. Provide an ER diagram for your database. Make sure you include all of the entities and relationships you determined in the questions above ***INCLUDING the entities for question 3 above***, and remember that ***EVERY*** entity in your model needs to connect to another entity in the model via some kind of relationship.

Grader Feedback:

If an entity only has one attribute and one relationship, consider making that entity into a simple attribute. The orientation of the diagram should have the most important entities in the top left corner, and generally, should flow top-bottom, left-right. Overall, this diagram looks good and seems reasonable to implement.

CSE 3241 Project Checkpoint 02

Relational Model and Relational Algebra

Names: Sruti Chigurupati, Elizabeth Gilbert, Trey Hakanson, Maggie Baumgardner

Date: 2/20/18

In a **NEATLY TYPED** document, provide the following:

1. Provide a current version of your ER Model as per Project Checkpoint 01. If you were instructed to change the model for Project Checkpoint 01, make sure you use the revised version of your ER Model.
2. Map your ER model to a relational schema. Indicate all primary and foreign keys.
3. Given your relational schema, provide the relational algebra to perform the following queries. If your schema cannot provide answers to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries: $\sigma \ \pi \ \rho \ \bowtie$

- a. Find the titles of all songs by ARTIST released before YEAR

$$\pi_{\text{track_title}}(\sigma_{\text{release_date} < \text{YEAR}}(\sigma_{\text{name} = \text{ARTIST}}(\text{ARTIST} * \text{ARTIST_ALBUMS} * \text{ALBUM} * \text{TRACK})))$$

- b. Give all the albums and their date of their checkout from a single patron (you choose how to designate the patron)

$$\pi_{\text{album_title}, \text{checkout_date}}(\sigma_{\text{card_number} = \text{CARD_NUMBER}}(\text{PERSON} * \text{MEDIA} * \text{ALBUM}))$$

- c. List all the albums and their unique identifiers with less than 5 copies held by the library.

$$\pi_{\text{album_id}, \text{title}}(\sigma_{\text{count} < 5}(\sigma_{\text{album_id}}(\text{F_COUNT } \text{album_id}(\text{ALBUM} * \text{MEDIA}))))$$

- d. Give all the patrons who checked out an album by ARTIST and the albums they checked out.

$$\pi_{\text{first_name}, \text{last_name}, \text{album_title}}(\sigma_{\text{checkout_date} \neq \text{NULL}}(\sigma_{\text{name} = \text{ARTIST}}(\text{ARTIST} * \text{ARTIST_ALBUMS} * \text{ALBUM} * \text{PERSON} * \text{MEDIA})))$$

- e. Find the total number of albums checked out by a single patron (you choose how to designate the patron)

$$\text{F_COUNT } \text{album_id}(\sigma_{\text{card_number} = \text{CARD_NUMBER}}(\text{MEDIA}))$$

- f. Find the patron who has checked out the most albums and the total number of albums they have checked out.

$$\text{F_MAX count}(\text{card_number } \text{F_COUNT } \text{album_id}(\sigma_{\text{checkout_date} \neq \text{NULL}}(\text{MEDIA})))$$

4. Come up with three additional interesting queries that your database can provide. Give what the queries are supposed to retrieve in plain English and then as relational algebra. Your queries should include joins and at least one should include an aggregate function. At least one of your queries should use “extra” entities you added to your model in Checkpoint 01.

Each patron has given how many feedbacks?

$$\text{card_number } \text{F_COUNT } \text{feedback_id}(\text{PERSON} * \text{FEEDBACK})$$

What is the average star rating for each album?

album F AVERAGE stars (REVIEW*ALBUM)

How many copies does each album have (physical and digital)?

album_id F COUNT media_id (ALBUM*MEDIA)

Grader Feedback:

Looks great, keep up the good work!

CSE 3241 Project Checkpoint 03

SQL and More SQL

Names: Sruti Chigurupati, Elizabeth Gilbert, Trey Hakanson, Maggie Baumgardner

Date: 3/6/18

Submitted to the Carmen Dropbox

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 02. **If you were instructed to change the model for Project Checkpoint 02, make sure you use the revised versions of your models**
2. Given your relational schema, create a text file containing the SQL code to create your database schema. Use this SQL to create a database in SQLite. Populate this database with the data provided for the project as well as 20 sample records for each table that does not contain data provided in the original project documents.
3. Given your relational schema, provide the SQL to perform the following queries. If your schema cannot provide answers to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries. These queries should be provided in a plain text file named "WorksheetTwoSimpleQueries.txt":
 - a. Find the titles of all tracks by ARTIST released before YEAR
 - b. Give all the albums and their date of their checkout from a single patron (you choose how to designate the patron)
 - c. List all the albums and their unique identifiers with less than 5 copies held by the library.
 - d. Give all the patrons who checked out an album by ARTIST and the albums they checked out.
 - e. Find the total number of albums checked out by a single patron (you choose how to designate the patron)
 - f. Find the patron who has checked out the most albums and the total number of albums they have checked out.
4. For Project Checkpoint 02, you were asked to come up with three additional interesting queries that your database can provide. Give what those queries are supposed to retrieve in plain English, as relational algebra and then as SQL. Your queries should include joins and at least one should include an aggregate function, and they should be the same as the queries you outlined for Worksheet 02. If you were instructed to fix the queries in Checkpoint 02, make sure you use the fixed queries here. These queries should be provided in a plain text file named "WorksheetTwoExtraQueries.txt".

Each patron has given how many feedbacks?

$\text{card_number} \bowtie_{\text{COUNT feedback_id}} (\text{PERSON} * \text{FEEDBACK})$

What is the average star rating for each album?

$\text{album} \bowtie_{\text{AVERAGE stars}} (\text{REVIEW} * \text{ALBUM})$

How many copies does each album have (physical and digital)?

$\text{album_id} \bowtie_{\text{COUNT media_id}} (\text{ALBUM} * \text{MEDIA})$

Once you have completed all of the questions, create a ZIP archive containing the binary SQLite file and the three text files and submit this to the Carmen Dropbox. **Make sure your queries work against your database and provide your expected output before you submit them!**

Grader Feedback:

Grader:

1. Missing ER diagram
2. Missing a text file to create the database and insert data – I was expecting a text file with CREATE TABLE and INSERT INTO statements that can be copied and pasted into SQLite.
3. & 4. Can't read/verify the queries since they were submitted as a database instead of a text file.

Team:

All of the data is contained in the *.sql files; these are not databases, they are SQL scripts, containing the queries. They can be opened by any text editor/IDE; sorry for the confusion, I am just used to writing my queries in files with the .sql extension

Grader:

Ok I will look at them again.

CSE 3241 Project Checkpoint 04

SQL and More SQL

Names: Sruti Chigurupati, Elizabeth Gilbert, Trey Hakanson, Maggie Baumgardner

Date: 3/20/18

Submitted to the Carmen Dropbox

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 03. **If you were instructed to change the model for Project Checkpoint 02, make sure you use the revised versions of your models**
2. Use the database populated with the data provided for Checkpoint 3 (with at least 20 sample records for each table and including data provided in the original project documents).
3. For Project Checkpoint 03, you were asked to come up with a set of queries and three additional interesting queries that your database can provide. outlined in Worksheet 02. If you were instructed to fix the queries in Checkpoint 03, make sure you include the fixed queries here. These queries should be provided in a plain text file named "WorksheetTwoExtraQueries.txt".
4. Given your relational schema, provide the SQL for the following more advanced queries. These queries may require you to use techniques such as nesting, aggregation using having clauses, and other techniques. If your database schema does not contain the information to answer to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries. **Note that if your database does contain the information but in non-aggregated form, you should NOT revise your model but instead figure out how to aggregate it for the query!** These queries should be provided in a plain text file named "WorksheetTwoAdvancedQueries.txt".
 - a. Provide a list of patron names, along with the total combined running time of all the albums they have checked out.
 - b. Provide a list of patron names and email addresses for patrons who have checked out more albums than the average patron.
add email address
 - c. Provide a list of the albums in the database and associated total copies lent to patrons, sorted from the album that has been lent the most to the album that has been lent the least.
 - d. Provide a list of the titles in the database and associated totals for copies checked out to customers, sorted from the title that has been checked out the highest amount to the title checked out the smallest.
 - e. Find the most popular artist in the database (i.e. the one who has had the most lent albums)
 - f. Find the most listened to artist in the database (use the running time of the album and number of times the album has been lent out to calculate)
 - g. Provide a list of customer information for patrons who have checked out anything by the most listened to artist in the database.

- h. Provide a list of artists who authored the albums checked out by customers who have checked out more albums than the average customer.

Once you have completed all of the questions for Part Two, create a ZIP archive containing the binary SQLite file and the three text files and submit this to the Carmen Dropbox. **Make sure your queries work against your database and provide your expected output before you submit them!**

Grader Feedback:

Feedback for queries from Checkpoint #3:

3b, d, e, f: Queries do not work since the media table does not have a card_number column and thus that cannot be used to join the media and person table.

The three extra queries work, good job.

Feedback for the advanced queries:

4a: Same issue as the previous queries where there is no card_number column in the media table.

4g: Need to specify which table the card_number information is being selected from or else SQL does not know where to pull the data from.

Missing file/spreadsheet of the expected outcomes from the queries.

Functional Dependencies and Normal Forms

Names: Sruti Chigurupati, Elizabeth Gilbert, Trey Hakanson, Maggie Baumgardner

Date: 4/3/18

In a **NEATLY TYPED** document, provide the following:

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 03. **If you were instructed to change the model for Project Checkpoint 03, make sure you use the revised versions of your models.**
2. For each relation schema in your model, indicate the functional dependencies. Think carefully about what you are modeling here - make sure you consider all the possible dependencies in each relation and not just the ones from your primary keys. For example, a customer's credit card number is unique, and so will uniquely identify a customer even if you have another key in the same table (in fact, if the customer can have multiple credit card numbers, the dependencies can get even more involved).

ALBUM

{ album_id } -> { album_title, release_date, genre }

ARTIST_ALBUMS

{ album_id, artist_id } -> { album_id, artist_id }

ARTIST

{ artist_id } -> { name }

TRACK

{ album_id, title } -> { number, length, artist_id }

MEDIA

{ media_id } -> { type, album_id }

CHECKOUT

{ checkout_id } -> { checkout_date, media_id, return_date, card_number }

PERSON

{ card_number } -> { activation_date, first_name, last_name }

EMPLOYEE

{ card_number } -> { position, salary, start_date }

FEEDBACK

{ feedback_id } -> { description, date, category, card_number }

REVIEW

{ review_id } -> { stars, title, card_number, album_id, description }

3. For each relation schema in your model, determine the highest normal form of the relation. If the relation is not in 3NF, rewrite your relation schema so that it is in at least 3NF.

ALBUM: BCNF

ARTIST_ALBUMS: BCNF

ARTIST: BCNF

TRACK: BCNF

MEDIA: BCNF

CHECKOUT: BCNF

PERSON: BCNF

EMPLOYEE: BCNF

FEEDBACK: BCNF

REVIEW: BCNF

4. For each relation schema in your model that is in 3NF but not in BCNF, either rewrite the relation schema to BCNF or provide a short justification for why this relation should be an exception to the rule of putting relations into BCNF.

Already in BCNF

5. For your database, propose at least two interesting views that can be built from your relations. These views must involve joining at least two tables together each and must include some kind of aggregation in the view. Each view must also be able to be described by a one or two sentence description in plain English. Provide the code for constructing your views along with the English language description of what the view is supposed to be providing.

View 1: What albums are checked out. Checkout -> media_id -> album_id -> album_title

```
CREATE VIEW IF NOT EXISTS album_checkouts(  
    checkout_date,  
    return_date,  
    card_number,  
    album_title  
) AS  
  
SELECT checkout_date, return_date, card_number, album_title  
    FROM checkout c  
        JOIN media m ON c.media_id = m.media_id  
        JOIN album a ON m.album_id = a.album_id;
```

View 2: What albums are reviewed. review_id -> album_id -> album_title

```
CREATE VIEW IF NOT EXISTS album_reviews(  
    title,  
    description,  
    card_number,  
    album_title
```

) AS

```
SELECT title, description, card_number, album_title
```

```
FROM review r
```

```
JOIN album a ON r.album_id = a.album_id;
```

Grader Feedback:

1. Can't have disjoint with only one entity. You also have an N to M relationships between PERSON and CHECKOUT, and CHECKOUT and MEDIA but no new tables for those foreign keys.
5. Need aggregate functions in views