Team: Savage and Average
Team Members: Mallory Huston and Sory Diagouraga
Date: Monday, July 11th, 2022

Project Step 2 Draft: Normalized Schema + DDL with Sample Data

Feedback by the peer reviewer for Project Step 1

Gunars Turaids:

- **Does the overview describe what problem is to be solved by a website with DB back end?**

  Rich territory to build a database. Looks like you have a fairly complex (in a good way) business with interesting data relationships.

  Two things:

  You mention promotion of artists in a particular genre (or to people who are into a particular genre). However, I didn't see any genre information stored in the database. Is this something you plan to implement? (as an attribute of artists or as a customer preference attribute?)

  Also – your draft mentions Clearwater Casino. Is the arena a venue attached to a casino? If so, are there any casino business entities you might consider including. E.g. shuttles from the casino to the venue, or casino related concert promotions (free chips with ticket!)

- **Does the overview list specific facts?**

  There are facts!

- **Are at least four entities described and does each one represent a single idea to be stored as a list?**

  There are, and they do!

- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?**

  Some observations, things to consider:

  Should customers be related to tickets? i.e. which customer purchased what.

  One concert has many customers, but can one customer attend many concerts? Or is customer attendance history not something you wish to track?

  How/why exactly are employees related to artists? Do the employees manage or coordinate with specific artists for booking?

  Just some things to think about.

- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

  I believe they are correctly formulated as presented, but see above considerations regarding some relationships.

- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

  I didn't catch any inconsistencies, but I'm still learning the proper conventions myself.

  This sounds like a fun project!

Beckett Lewis:

Good job on this, I really like the idea! Here are my answers to the prompt with some suggestions:

- **Does the overview describe what problem is to be solved by a website with DB back end?**

Yes, the DB will help Clearwater Arena track information and promote artists to groups who like their genre of music.

- **Does the overview list specific facts?**

Yes, it lists that there are 30 concerts per month, 5,000 seats and 25 employees. Good job being specific here!

- **Are at least four entities described and does each one represent a single idea to be stored as a list?**

Yes, there are 5 entities described, each as something to be representing an idea to store as a list.

- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?**

The purpose of the entities is missing in this. Also, for the artists, it should be described in more detail the constraints of the attributes, like the artistID should be noted as the PK. Also, the concert has 2 PKs, but I think an entity can only have a single PK.

- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

Yes, the relationships seem to be formulated correctly. I'm not sure if the employee needs to be related to the artist though. Unless you consider the artist to be an employee?

- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

Yes, for the most part. I would recommend changing the name of the intersection table IDs, like Concerts_concertID to just concertID to be more consistent.

<u>Brent Goldman:</u>

Overall, I like the idea of a ticket-based entertainment database. There are a few wrinkles that need to be ironed out between now and the final submission. Nice work!

- **Does the overview describe what problem is to be solved by a website with DB back end?**
    - Yes, the database will help promote certain artists to groups of people based on their music preferences.
        - The first sentence is confusing. It mentions "venues" but there is only one venue, Clearwater Arena?
    - The database will also store information on ticket sales, promotions, and ticket pricing. What will this information be used for?
- **Does the overview list specific facts?**
    - Yes:

        - 30 concerts per month from various artists
        - Venue has 5,000 seats maximum
            - I don't see any attributes for seats filled
        - 25 employees
            - You mention Clearwater Casino, did you mean Clearwater Arena?
- **Are at least four entities described and does each one represent a single idea to be stored as a list?**
    - Yes, there are 5 entities plus 2 bridging entities for M:M relationships.
    - Tickets is confusing since the ERD shows a relationship between Tickets and Concerts, but the outline shows a relationship between Tickets and Customers. A customer would have a relationship with a ticket, but it doesn't make much sense for a concert to have a relationship with a ticket. Some things to think about: Can a customer purchase multiple tickets, how does pricing work? Are tickets really even necessary, since the main goal of the project is to promote certain genres. If you remove tickets, should Customers become something like "Fans" instead?
        - The overview talks about price per ticket, but this isn't an attribute.
    - I don't see any Entities or attributes for promotions but this is mentioned in the overview.
- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?**
    - No - there is no description provided for each entity. There are attributes with datatypes. I don't see any constraints provided.
    - For the "role" attribute in Employees, it may be beneficial to create a separate entity that contains the various roles. Can an employee have multiple roles?
    - concertDate is listed as an int, but should be date type. Also it's auto_increment, but that doesn't make sense. And it's listed as unique, but can there be multiple concerts on the same date?
    - Several relationships do not mention the FK being used in the other entity.
- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**
    - No, there are a few issues that need to be resolved. There are 2 M:M relationships. The ERD has some issues.
        - Under Artists, I see Artistscol and Artistcol1, what is this? It wasn't listed in the outline.
        - In Customers, Concerts_concertID is listed twice, as a PK and FK.
        - The outline relationship says that Customers is related to Tickets, but the ERD shows a relationship between Customers and Concerts.
        - The ERD shows a relationship between Employees and Artists, this isn't mentioned in the Employees relationship section, only the Artists. Also, do you really need a relationship between Employees and Artists? I don't know if this adds value.
        - ERD shows M:M between Concerts and Employees, but the Concerts in the outline don't list this relationship.

- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**
    - Yes, for the most part. The entities are all capitalized and plural. The FK attributes were capitalized, but I think this was done automatically by MySQL when generating the ERD. If it wasn't automatic, then you should remove the FK entity name, i.e. "Concerts_concertID" should become "concertID".

Actions based on the feedback

Actions Taken

Overview: We immediately changed the typo of “Clearwater Arena” to “Clearwater Casino” in the overview to avoid any further confusion. We then took out “promotions” and “genres” in the overview since we did not cover that whatsoever since it would make this project way too complicated.
Employees and Artists: We also took out any relationship between the Employees and Artists because it was not necessary and would be far too complicated. Plus, it is not how real life works and whichever employees would be helping out the artists are the ones who are on staff that night. There cannot be more than one show per night either.
numberOfTickets: We also put in the entity “numberOfTickets” under Concerts so Employees can keep track of how many people show up and that the venue never goes over max capacity.
artistID: We then made the artistID entity the same as the other entities that are unique, not NULL, auto_increment, etc.
Fans: We decided to rename the entity “Customers” to “Fans” because that type of demographic makes more sense for a concert arena anyway and it’s less confusing.

Actions Not Taken

Genres and Promotions: We did not follow the other suggestions of putting in genres and promotions since it would over complicate our database.
Employee role: We also want to make it clear that all Employees can only have one role each, not multiple.
Artistscol and Artistscol1: We did delete “Artistscol” and “Artistscol 1” since they did not make much sense anyway.
concertID: Additionally, we want to make it clear that the relationship between Concerts and Fans is implemented with “concertID” while Artists and Concerts has an implementation table.
Concerts_concertID: Lastly, we did not remove “Concerts_concertID” from Concerts_has_Employees and Concerts_has_Artists because the Foreign Key entity names were done automatically by MySQL.

Overview

Clearwater Arena is a stadium near multiple small towns in the state of Wisconsin that provide world-class entertainment through their venues. They provide 30 concerts per month from different artists around the country. It requires a database to store their ticket sales, artists, fan information, and price per ticket. The venue has 5,000 seats and the Employees have to make sure that it never goes over maximum capacity. There are currently 25 employees at Clearwater Arena who provide different services to fans such as bartenders, customer service reps, cashiers, and security. A database driven website will record all of the sales of Tickets for Fans, Artists, and Employees.

Database Outline

Object Entities:

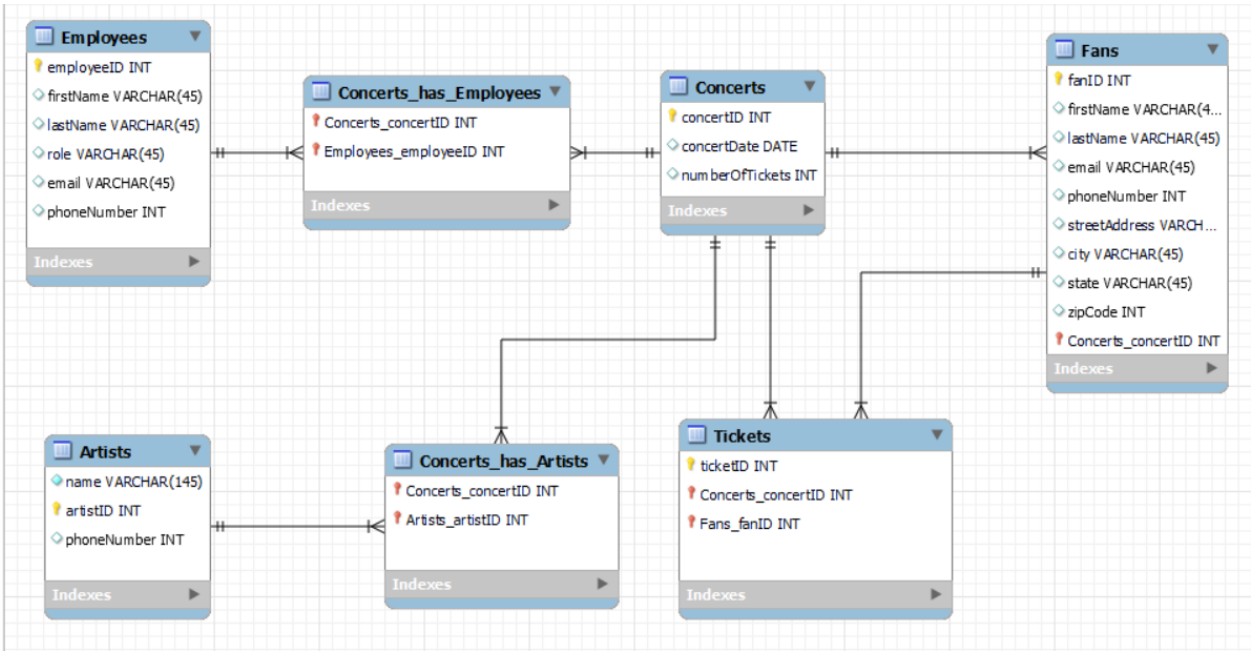
- Fans
    - fanID: int, auto_increment, unique, not NULL, PK
    - email: varchar
    - firstName: varchar
    - lastName: varchar
    - phone: varchar
    - address: varchar
    - city: varchar
    - state: varchar
    - zipCode: varchar
    - Relationships:
        - A 1:M relationship between Fans and Tickets is implemented with fan_ID as a foreign key inside of Tickets. One fan may buy a ticket and a ticket can be purchased by multiple fans.

- Employees
    - employeeID: int, auto_increment, unique, not NULL, PK
    - role: varchar
    - firstName: varchar
    - lastName: varchar
    - email: varchar
    - phoneNumber: int
    - Relationships:
        - A M:M relationship between Employees and Concerts is implemented with an intersection table, since one Employee may be responsible for multiple Concerts, and one Concert may be served by multiple Employees.

- Concerts
    - concertID: int, auto_increment, unique, not NULL, PK
    - concertDate: date
    - numberOfTickets: int
    - Relationships:
        - A 1:M relationship exists between Concerts and Fans. It's implemented with concertID as a foreign key inside of Fans. One Fan may only attend one Concert and a Concert may have many Fans.
        - A M:M relationship exists between Concerts and Artists. A concert can have many artists and an artist can have multiple concerts.
- Artists
    - artistID: int, auto_increment, unique, not NULL, PK
    - name: varchar, not NULL
    - phoneNumber: int
    - Relationship:
        - A M:M relationship between Artists and Fans is implemented with an intersection table, since one Artist may perform at multiple concerts and a Concert may have multiple Artists performing.

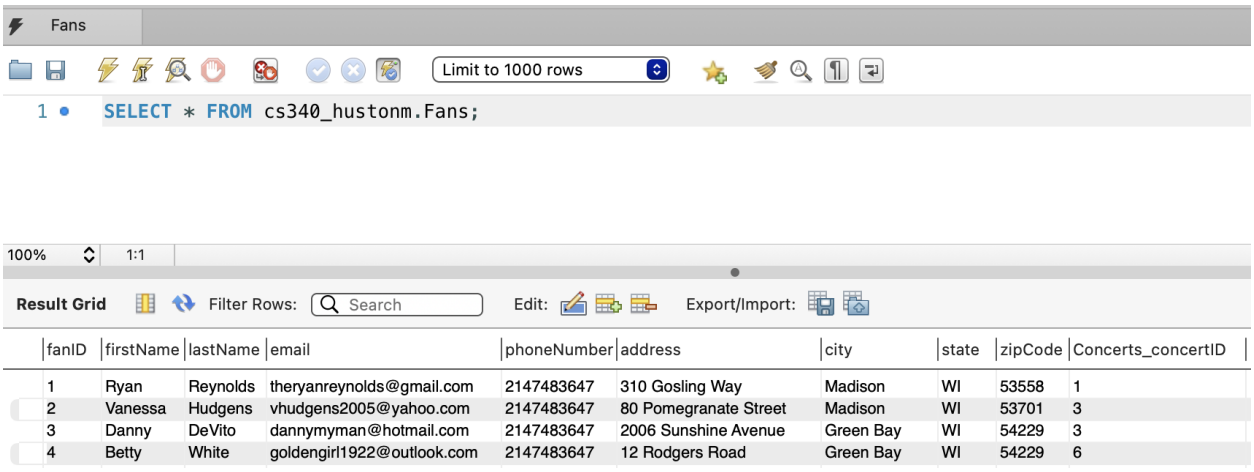
- Tickets
    - ticketID: int, auto_increment, unique, not NULL, PK
    - Relationships:
        - A 1:M relationship exists between Tickets and Fans. A ticket may only be bought by one fan, but a fan can buy multiple tickets.
        - A 1:M relationship exists between Tickets and Concerts. A ticket may only be for one concert, while a concert may have many tickets.

Entity Relationship Diagram



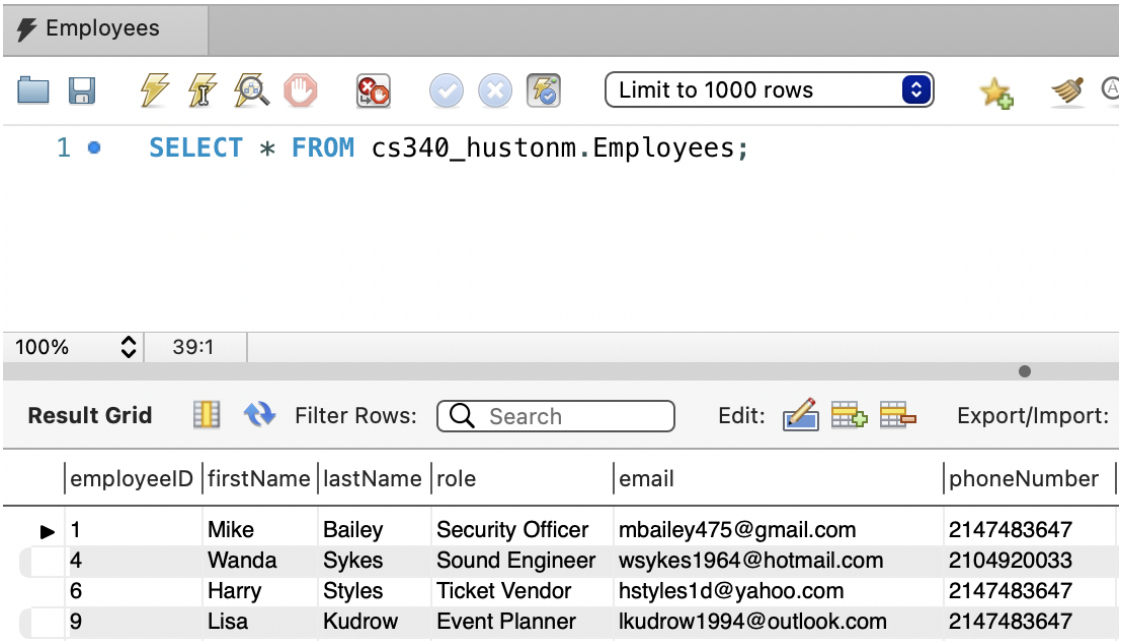
Data Used in Databases

Fans:



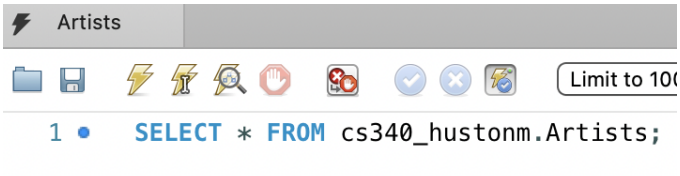
fanID	firstName	lastName	email	phoneNumber	address	city	state	zipCode	Concerts_concertID
1	Ryan	Reynolds	theryanreynolds@gmail.com	2147483647	310 Gosling Way	Madison	WI	53558	1
2	Vanessa	Hudgens	vhudgens2005@yahoo.com	2147483647	80 Pomegranate Street	Madison	WI	53701	3
3	Danny	DeVito	dannymyman@hotmail.com	2147483647	2006 Sunshine Avenue	Green Bay	WI	54229	3
4	Betty	White	goldengirl1922@outlook.com	2147483647	12 Rodgers Road	Green Bay	WI	54229	6

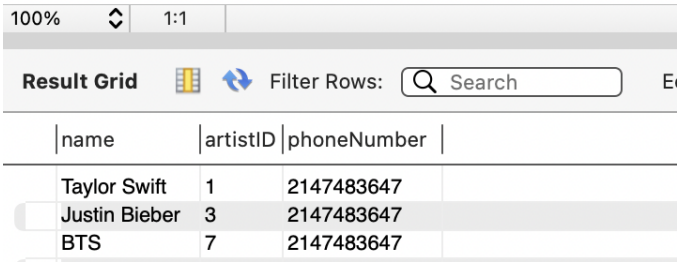
Employees:



employeeID	firstName	lastName	role	email	phoneNumber
1	Mike	Bailey	Security Officer	mbailey475@gmail.com	2147483647
4	Wanda	Sykes	Sound Engineer	wsykes1964@hotmail.com	2104920033
6	Harry	Styles	Ticket Vendor	hstyles1d@yahoo.com	2147483647
9	Lisa	Kudrow	Event Planner	lkudrow1994@outlook.com	2147483647

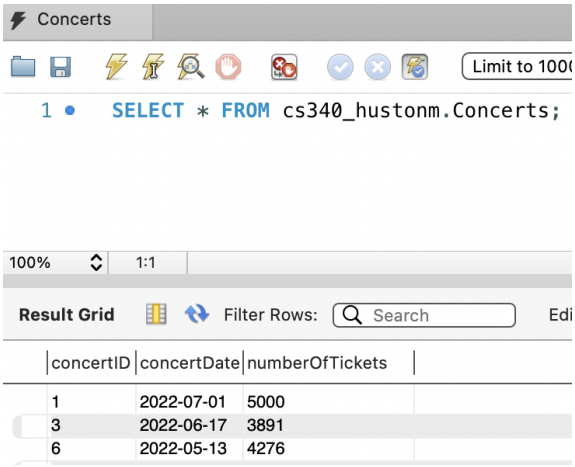
Artists:





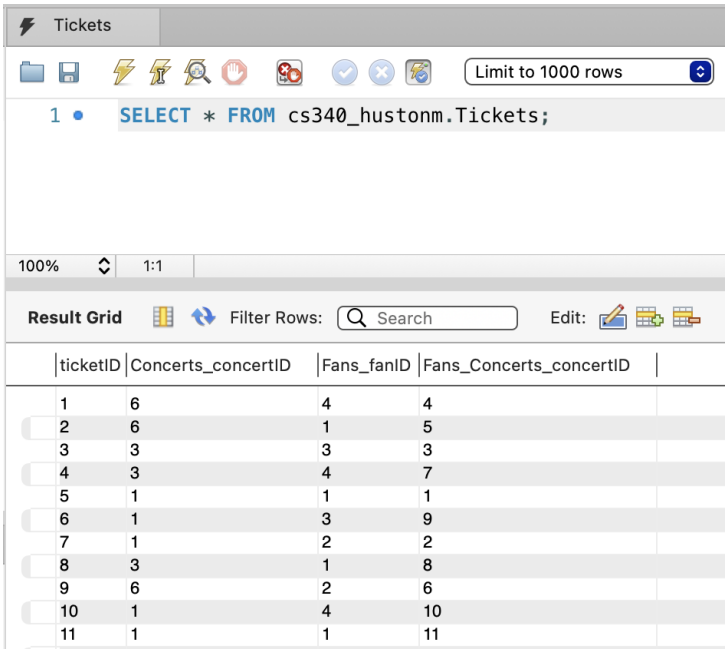
name	artistID	phoneNumber
Taylor Swift	1	2147483647
Justin Bieber	3	2147483647
BTS	7	2147483647

Concerts:



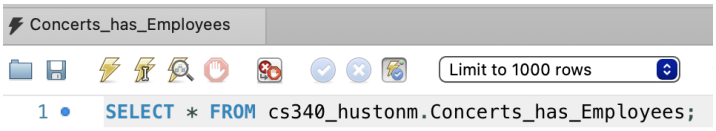
concertID	concertDate	numberOfTickets
1	2022-07-01	5000
3	2022-06-17	3891
6	2022-05-13	4276

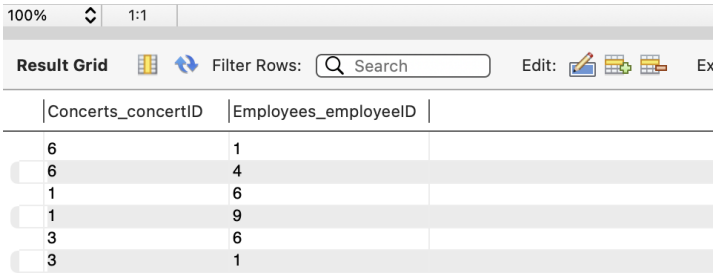
Tickets:



ticketID	Concerts_concertID	Fans_fanID	Fans_Concerts_concertID
1	6	4	4
2	6	1	5
3	3	3	3
4	3	4	7
5	1	1	1
6	1	3	9
7	1	2	2
8	3	1	8
9	6	2	6
10	1	4	10
11	1	1	11

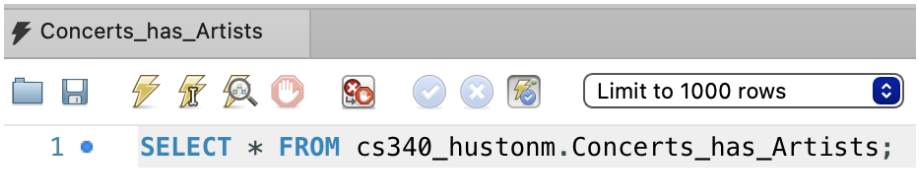
Concerts_has_Employees (Intersection Table):

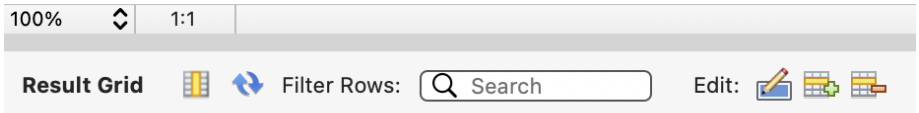




Concerts_concertID	Employees_employeeID
6	1
6	4
1	6
1	9
3	6
3	1

Concerts_has_Artists (Intersection Table):





Concerts_concertID	Artists_artistID
6	1
1	3
3	7