Project Step 3 Draft: Design HTML Interface + DML SQL - Datasphere Devs

Members: Madonna Matta and Haedon Kaufman

URL: http://classwork.engr.oregonstate.edu:9263/

Overview:

BandTogether sells $50 million worth of concerts, sports, and other special event tickets online a year. This website will record Ticket sales of Events to Customers, as well as Venues and Performers. This platform seamlessly connects users to their favorite performers and events, while also facilitating event discovery as well.

Database Outline in Words:

Tables would be Events, Customers, TicketSales, Performers, and Venues. Detailed entities are listed below.

- **Events -** This table would document the specific details of events occurring in connection with our venues and performers.
    - event_id: int, auto_increment, unique, not NULL, PK
        - Purpose: Serves as a unique identifier for the Event
    - name: varchar (200) , not NULL
        - Purpose: Stores the name of the event
    - date: datetime, not NULL
        - Purpose: Lists the date and time of the event taking place
    - price: decimal (10, 2), not NULL
        - Purpose: Lists the price of the event.
    - venue_id: int, auto_increment, unique, not NULL, FK (links to Venues table)
        - Purpose: Serves as a unique identifier for the venue. Links to Venue_ID to specify where the event is held.
    - total_attendees: int, not NULL
        - Purpose: Records the number of attendees that can attend the event
    - relationships:
        - Events to Venues - (M:1) Each event held to one venue at a time, however, Venues can have multiple events. FK is Event_Location linking to Venue_id
        - Events to TicketSales - (1:M) Each event has multiple ticket sales records. FK is in TicketSales as Event_id.
        - Events to Performers - (M:N) Reflects which performers will perform at which events.
- **Customers -** This table serves as a catalog of information of our platform's customers.
    - customer_id: int, auto_increment, unique, not NULL, PK
        - Purpose: Serves as a unique identifier for the customer
    - first_name: varchar (200) , not NULL
        - Purpose: Stores the first name of customer

- last_name: varchar (200), not NULL
  - Purpose: Stores the last name of customer
- email: varchar (200), unique, not NULL
  - Purpose: Stores the email of the customer, used as a unique identifier for logging in.
- password: varchar (200), not NULL
  - Purpose: Stores the password of the customer

- Relationships: 1:M relationship between Customers and TicketSales. Each customer can buy multiple tickets. The relationship is key to tracking every purchase made per customer on the site. Foreign key would be CustomerID in TicketSales

- **Venues -** This table serves as a catalog of information for our platform's venues in our network.
  - venue_id: int, auto_increment, unique, not NULL, PK
    - Purpose: Serves as a unique identifier for the venue
  - name: varchar (200), not NULL
    - Purpose: Stores the name of the Venue
  - venue_location: varchar(299), notNULL
    - Purpose: Stores the specific physical address of the venue
  - capacity: int, not NULL
    - Purpose: Indicates the max amount of attendees the venue can hold.
  - Relationships: Venues to Events - (1:M) A venue can host multiple events.

- **Ticket_Sales -** This table serves as a catalog of all the tickets the platform has sold for our events to our customers.
  - ticket_sales_id: int, auto_increment, unique, not NULL, PK
    - Purpose: Serves as a unique identifier for each ticket sale.
  - customer_id: int, not NULL, FK (links to Customer table)
    - Purpose: Links ticket sales to a customer
  - event_id: int, not NULL, FK (links to Event table)
    - Purpose: Links the ticket to an event
  - number_of_tickets: int, not NULL,
    - Purpose: Records the number of tickets that were sold per transaction
  - date_of_purchase: datetime, not NULL
    - Purpose: Records when tickets were purchased
  - total_cost: decimal (10,2), not NULL,
    - Purpose: Calculates and lists the total cost of the tickets sold.
  - relationships:
    - TicketSales to Customers (M:1) Many tickets can be sold to a single customer.
    - TicketSales to Events (M:1) Multiple ticket sales can be associated with a single event

- **Performers -** This table serves as a catalog of all the artists, athletes, and overall performers our venues have hosted.
    - performer_id: int, auto_increment, unique, not NULL, PK
        - Purpose: Serves as a unique identifier for each performer.
    - name: varchar (200), not NULL
        - Purpose: Stores the name of the Performer
    - genre: varchar(100), not NULL
        - Purpose: Specifies the genre of the performer (ex. concert, theater, sport)
    - biography: text, NULL
        - Purpose: Lists a brief history of the performer.
    - website: varchar (200), NULL
        - Purpose: Links the performer's official website
    - relationships:
        - Performers to Events (M:N) Many performers can perform multiple events, just not at the same time.

- **Events_Performers -** This would be a junction table that connects both events and performers.
    - event_id: int, auto_increment, unique, not NULL, FK
    - performer_id: int, auto_increment, unique, not NULL, FK
    - relationships: Intersection table that facilitates the M:N relationship between events and performers.
        - Events_Performers to Events (M:1) Many performers can have one event associated with it but can only refer to one event at a time establishing which performers are at which event.
        - Events_Performers to Performers (M:1) One performer can perform at many events but can only refer to one performer and one event at a time.

Project Step 3 Draft: Design HTML Interface + DML SQL - Datasphere Devs

Feedback From Reviewers and Graders:
No feedback for Graders.
Feedback from reviewers:
Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
We did not receive any negative feedback here.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
We received feedback regarding Venues entity. It was recommended to make this plural. We have made this correction in our outline and diagrams.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
We did not receive any negative feedback here.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
We received feedback regarding our intersection table. They recommended using an additional primary key for this table. We do have a primary key, which is the conjunction of the primary keys from Performers and Events. After further research, this is normal and is done in professional environments. We will not be making this change.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
We did not receive any negative feedback here.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
We received feedback regarding varchar(255) for address. When looking further into this, this is recommended for addresses. Unless you expect exceedingly long addresses then varchar(255) is adequate. So, we are not making changes based on this recommendation.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
We received feedback regarding our CASCADE operations. After reviewing how the reviewers made their CASCADE declarations, I understand where they are coming from. The way we constructed our tables was more for readability. I noticed in their tables they chose to add additional queries altering their tables and adding the CASCADE. This does the same thing and we will not be making any changes based on their recommendation.

In the SQL, are relationship tables present when compared to the ERD/Schema?
We did not receive any negative feedback here.

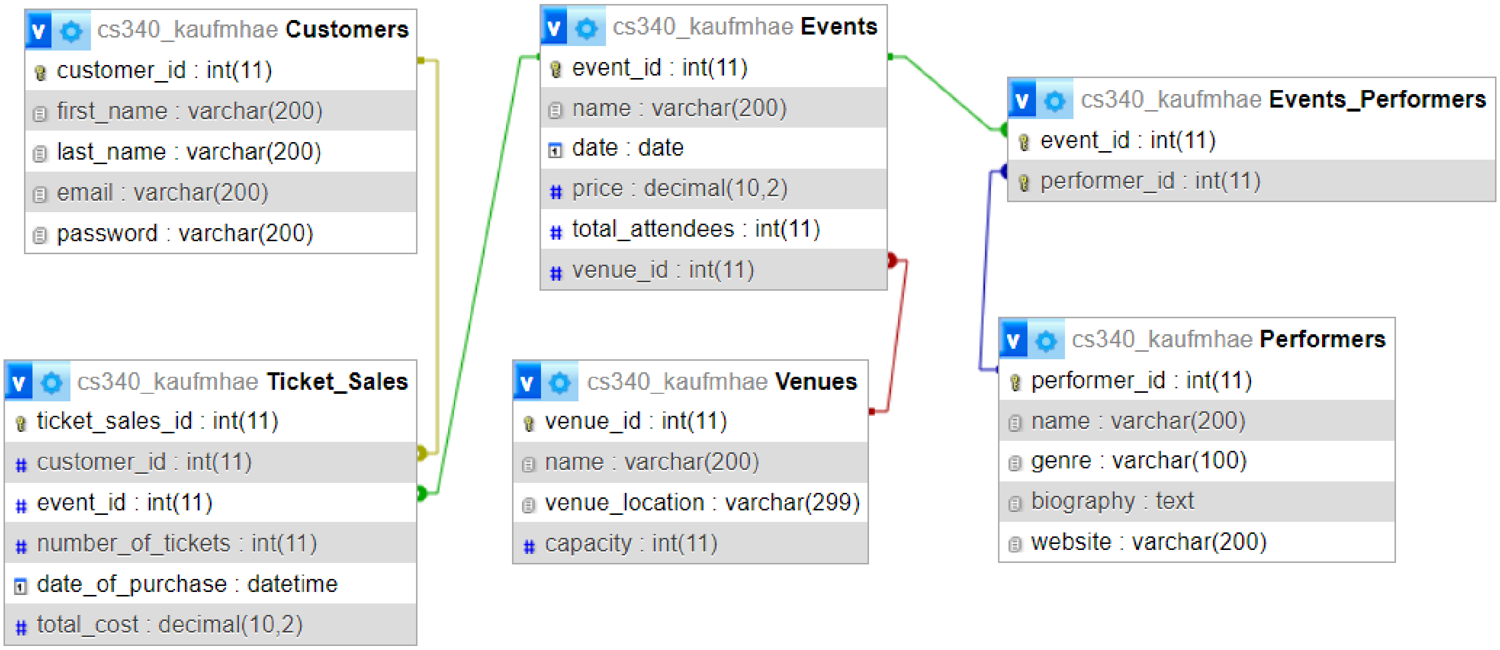
Project Step 3 Draft: Design HTML Interface + DML SQL - Datasphere Devs

In the SQL, is all example data shown in the PDF INSERTED?
We did not receive any negative feedback here.

Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
We received feedback regarding our comments. We did not add enough comments to establish readability for our SQL document. We will be taking this recommendation and applying more comments to our document.

Project Step 3 Draft: Design HTML Interface + DML SQL - Datasphere Devs

ERD Schema:



Example Data:

Event's Table					
event_id	name	date	price	venue_id	total_attendees
1	Rock Concert	2024-05-09	75	1	3400
2	Showcase	2024-07-09	125	2	9200
3	Pop Concert	2024-08-05	50	3	1200

Customer Table				
customer_id	first_name	last_name	email	password
1	Adam	Smith	adamsmith@dummyemail.com	password123
2	Jane	Doe	j.doe@dummyemail.com	123password
3	Bob	Williams	bobwilliams@dummyemail.com	12345

Venues Table			
venue_id	name	venue_location	capacity
1	Live Oak Music Hall	123 Main St.	5000
2	Verizons Stadium	234 Crocket Ln.	12500
3	Banana Bar Music Hall	345 2nd St.	3500

Ticket Sales Table					
ticket_sales_id	customer_id	event_id	number_of_tickets	date_of_purchase	total_cost
1	1	1	2	2024-05-01	150
2	2	2	1	2024-05-25	125
3	3	3	2	2024-06-13	100

Project Step 3 Draft: Design HTML Interface + DML SQL - Datasphere Devs

Performer Table				
performer_id	name	genre	biography	website
1	Hozier	Rock	Andrew John Hozier-Byrne (born 17 March 1990), known professionally as Hozier is an Irish musician, singer and songwriter.	https://hozier.com/
2	UCLA Showcase	Student Showcase	The UCLA music departs welcomes the city of Los Angeles to a unique production of our finest musicians tonight!	UCLAmusic.com
3	Selena Gomez	Pop	Selena Gomez is an American singer, actress, and producer, known for her musical hits and roles in TV shows and films, as well as her work as a mental health advocate.	https://www.selenagomez.com/

Event_Performers Table	
event_id	performer_id
1	1
2	2
3	3