Megan Shea
SWDV 691

Design: Database
Event Planner

Database Technology

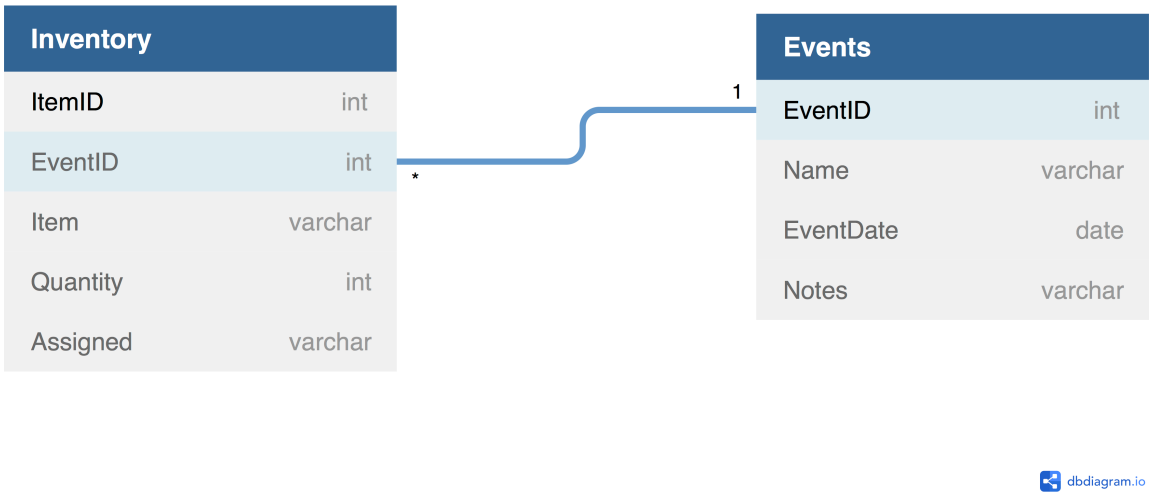
For my project, I chose and am sticking with a relational database design and will use MySQL. This is due to all the data associated with the events. To help track and plan future events, using a relational database provides an option for data analysis. Also, another deciding factor for a relational database is the use of cascade update/delete. For example, if an event is deleted from the Events table, all matching rows in the child tables(Inventory and Receipts) will be deleted.

Database Design

My previous database design took into account my stretch goal and not just the MVP. So for this re-design, I provided both the database design with the MVP and Stretch Goal (labeled the same below).

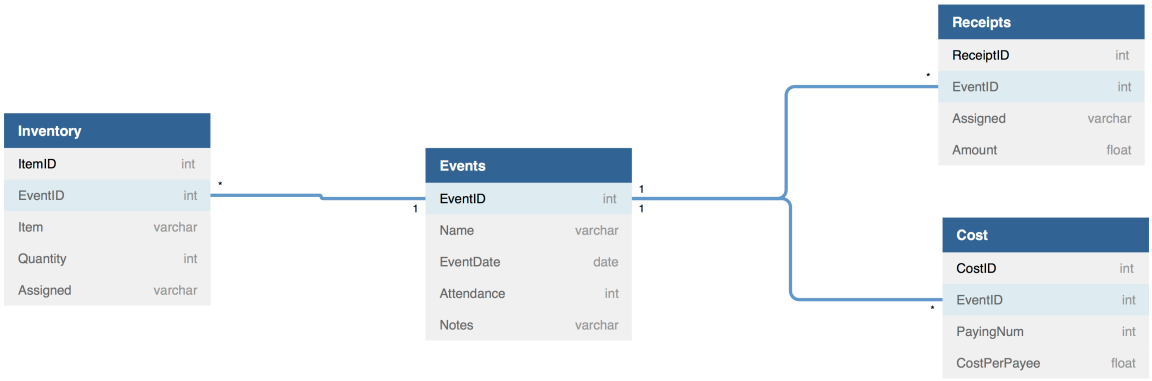
MVP – ER Diagram

I made a few changes to the database in the re-design. The ER Diagram for my project MVP consists of 2 tables: Events and Inventory. The Events table is the main table with a primary key of EventID. The remaining columns are Name(Unique ID), EventDate, and Notes. EventID is also used as a foreign key in the Inventory. The Events table has a one to many relationship with the Inventory table. Out of the two tables, only the Notes(Events) and Assigned(Inventory) can be NULL; all other fields will be NOT NULL.
The Inventory table has a primary key of ItemID and will keep track of inventory for all events. The EventID will be used to distinguish which specific event the inventory item is for. The Inventory table will also consist of Item, Quantity, and Assigned columns.



Stretch Goal – ER Diagram

The ER Diagram for my stretch goal consists of 4 tables: Events, Inventory, Costs, and Receipts. The Events table is the main table with a primary key of EventID. The remaining columns are Name(Unique ID), EventDate, Attendance, Cost, and Notes. EventID is also used as a foreign key in both the Inventory and Receipts tables. The Events table is the main table with a primary key of EventID. The remaining columns are Name, EventDate, and Notes. EventID is also used as a foreign key in the Inventory. The Events table has a one to many relationship with the Inventory table. Out of the two tables, only the Notes(Events) and Assigned(Inventory) can be NULL; all other fields will be NOT NULL.
The Inventory table has a primary key of ItemID and will keep track of inventory for all events. The EventID will be used to distinguish which specific event the inventory item is for. The Inventory table will also consist of Item, Quantity, and Assigned columns.
The Receipts table has a primary key of ReceiptID and will keep track of the receipts for all events. Again, the EventID will be used to distinguish which specific event the receipt is for. The Receipts table will also consist of Assigned and Amount columns.
The Costs table has a primary key of CostID and will keep track of the paying individuals and how much they owe for the event. Again, the EventID will be used to distinguish which specific event this is for. The Cost table will also consist of PayingNum and CostPerPayee columns.


Inventory
ItemID int
EventID int
Item varchar
Quantity int
Assigned varchar
Events
EventID int
Name varchar
EventDate date
Attendance int
Notes varchar
Receipts
ReceiptID int
EventID int
Assigned varchar
Amount float
Cost
CostID int
EventID int
PayingNum int
CostPerPayee float
*
1
1
1
1
*
*
dbdiagram.io