Video Rental Database

John Wensink

MIS417 Database Concepts

Colorado State University–Global Campus

Professor Nathan Braun

August 11, 2019

Video Rental Database

We at Ace Software Inc are honored to have been selected to implement a modern information solution to your growing business. We thank you for the privilege. Our team has elicited the requirements specific to your business model and we believe that we can design a relational database that addresses all of your current requirements and is adaptable to inevitable changes that will serve your company into the foreseeable future. The following synopsis should act as a roadmap to ensure we are on the same page as we follow the process to bring our creation to life. Revisions are to be expected as we use agile methodologies to break down the project as a whole into smaller deliverable pieces. Your input is a critical part of this process so please do not hesitate to speak up about anything you believe may benefit from a closer look.

**ER Diagram**

When developing a relational database, it is useful to have the details laid out ahead of time. To this end we will begin work by creating an “Entity-Relationship Diagram” (ERD) which is a logical database model depicting entities, attributes, and the unique relationships that tie everything together. (Al-Btoush, 2015) First we will identify the entities involved and the attributes which define them. The staff working at a branch will be an entity. If you haven’t done so already, it is important that each member of your staff be assigned a unique employee identification number. A five digit numeric is sufficient. This will be the employee entity’s primary key that will be used later as a cross-reference for other functions. For the employee entity, the attributes that are important to keep track of include the employee’s first name, middle initial, last name, username, hashed password, a URL to the employee’s photograph, which branch they work, and what their role is within the enterprise. If you haven’t done so already, it is important also to give each branch location a unique identifier that will serve as the branch’s primary key. This will be important for the employee table as it will serve as a foreign key that will yield highlity query results.

Branches are another example of an entity. As your business grows you may add or remove branches from your enterprise and setting up the database with branches as their own entity will simplify the process. Attributes of the branch entity should contain the branch’s identifier as the primary key, a five-digit alphabetic identifier should be sufficient for our purposes. Other attributes of a branch should be it’s address (including number, street, city, state, and zip), phone number, hours of operation (including opening time and closing time), as well as the branch manager’s name (first name, middle initial, last name), and employees (first name, middle initial, last name) that is borrowed from the employee table.

Customers will be another entity we describe in detail. There is predictive value in having high-quality customer data. Upon starting a membership the new customer should fill out a form that elicits many of the attributes that we will use in the database. For existing customers, staff should attempt to collect this data as a condition of their next rental.We need to develop a primary key to uniquely identify customers. This will be called membership number. One might think that a phone number would be a good primary key, however this could potentially cause problems with recycled phone numbers, more than one person having the same phone number, and people who do not wish to share that information. Instead we will generate a random ten digit membership number, which will allow for ten billion unique identities to be accounted for. We may be getting a bit ahead of ourselves here with attributes that have yet to be discussed, but they will be addressed later in the roadmap. A customer entity will have very many attributes as this is the substance that will make our database a valuable business tool. Required attributes should include the customer’s name (first name, middle initial, and last name), billing account number, phone number, email address, physical address (number, street, city, state, postal code, zip+4, country), mailing address (number, street, city, state, postal code, zip+4, country), signup date, latest account activity, account balance, movies rented, movies returned, agreement violations (late, missing, lost, damaged, not rewound), violation fees, birthday, gender, and marital status. This seems like a lot to keep track of, but herein lies the beauty of a relational database. These attributes will be important in enforcing contract rules such as maximum number of rentals out simultaneously, as well as tracking rental abusers who consistently damage or lose inventory. From this data you will be able to charge a rewind fee for rentals returned in person, dropped in the after-hours box un-rewound. We can use many of the attributes to identify patterns for which we can gain business intelligence and insight into predicting future rental trends.

When considering customer accounts, it is important to recognize that some members will share an account, or post charges to a business account that is not their own personal account. Therefore we will need to create another entity called account. The primary key will be called account number and should be a ten digit alphanumeric identifier that will be different than the membership number. Using alphanumeric inputs instead of simply numeric serves to prevent a mix-up of membership and account numbers, and also adds more unique identifier possibilities (3610.) This primary key should not be case sensitive to reduce input time and errors. Other attributes of the account entity should be account balance, last payment date, payment type, member who paid, type of payment, payment amount, receivable age, and blacklisted accounts. Having this data will help us enforce rental agreement requirements such as at least zero but no more than one personal account as well as maximum balance and receivable time until the account is blacklisted from additional rentals until payment is received.

The last entity we need to consider is the inventory itself. The best primary key for inventory is the Universal Price Code (UPC) which has already been uniquely generated and will not vary from store to store. (Nicasio, 2018) This will enable us to track inventory across the enterprise and allow inventory sharing between stores. UPC’s are not limited to videos, and are used for all inventory including accessories, hardware, food products, and gift cards. Attributes for inventory should include inventory type, supplier, cost per supplier unit, rent price, sell price, expiration date, and if the inventory is a video. The videos themselves will also need to contain attributes that don’t apply to other types of inventory. If the inventory is a video, we will need to have attributes such as release date, title, director, studio, category (new release, existing release, and classic), genre, subgenre, description, rating, medium, cast, length, and IMDB rating. From the database we are able to infer attributes such as popularity, average rental length, and demographic information regarding who may be interested in the film. An interesting inference can be made by comparing member’s rental histories. If a customer is looking for a recommendation, the database could query other members rental histories and suggest movies that other members, with similar rental histories, have rented but the inquiring member has not. We imagine a self-service kiosk with filterable search criteria in plain language that can narrow down the selection from many thousands, down to tens of recommended titles.

**Relationships**

Now that we have discussed the entities involved in the database we are ready to discuss the relationship pairs between them. A customer applies for a membership and a membership is assigned to a customer. An employee either works in or manages a branch that employs or is managed by an employee. A branch contains inventory and inventory is stocked by a branch. A customer registers for an account, and then rents or buys inventory and inventory is rented or purchased on an account. A customer charges or pays an account and the account is credited or debited by the customer. An employee purchases inventory and inventory is ordered by an employee. After all of the complexity in identifying the many attributes in play, the relationships may seem simple, but this simplicity is where the power of the relational database comes from. Once we are comfortable with the logical data model on paper, we can then proceed to the physical database design where we can incorporate all of the entities, attributes, primary keys, foreign keys, and relationships into tables based logical model. These tables will serve as the backbone of our database.

Ace Software needs your feedback to ensure a successful deployment of the database. What are some features of your stores that you would like to see represented as entities or attributes? What other kinds of business intelligence would you like to build using this data? To be sure, implementation of this new system will be time consuming to input the data, but once it’s in we we will be sure it is not lost or stolen using cryptography to secure sensitive data, and RAID storage to ensure that the data is not lost or corrupted. We recommend a phased implementation of this database, while continuing to use the legacy system until implementation is gradually completed. The choice of database hardware is ultimately up to you, but we recommend taking advantage of the recent advances of cloud database architecture as a service to ensure we are spending capital resources wisely and efficiently. Physical data servers are costly, quickly obsolete, and expensive to scale. With cloud services such as Amazon’s AWS or Microsoft’s Azure, you receive on demand storage that will easily scale as your business grows, and you need not worry about your server racks becoming obsolete in the future. Having a powerful relational database that is easy to query in plain language is a necessity as very few employees and customers will be literate in SQL language. A Graphical User Interface will need to be designed to ensure timely results. Once complete, the database can be tuned to ensure that it performs efficiently based on the complexity, frequency, and scope of the queries it may be called upon to answer. Implementing a relational database at your video rental chain will provide unparalleled insight into your business and lead you to make the next right decision now and into the future.

References

Al-Btoush, A. A. (2015). Extracting Entity Relationship Diagram (ERD) From English Sentences. International Journal of Database Theory and Application, 8(2), 235-244. doi:10.14257/ijdta.2015.8.2.22

Mannino, M. (2019). Database design, development, and application administration (7th ed). Retrieved from <https://platform.virdocs.com/r/s/0/doc/592088/sp/46211979/mi/190880309?cfi=%2F4%2F8&menu=table-of-contents>

Nicasio, F. (2018, December 13). Everything You Ever Wanted to Know About SKU Numbers. Retrieved August 11, 2019, from <https://www.vendhq.com/blog/sku-numbers/>