TO: Ms. Shana Delroy, Membership Director of NTJWC

FROM: Derek Herr, Trevor Leitzel, and Roger Baumbach

RE: NTJWC System: Database Schema

DATE: November 20, 2013

A database schema is a physical level model of the database designed to be independent of database management system (DBMS) choice. It contains additional information about data store such as access and content controls, development, relationships, and details of physical store (data element descriptions, locations, indexes, sorts). The database schema takes place during the design phase of the SDLC. A design schema uses much of the previous information given in the entity-relationship diagram (ERD) and also takes into account some information from the technical requirements.

To develop the initial table design, there are four steps you must complete. They are: create a table for each entity in the ERD, choose or invent primary key for each table, add foreign keys to represent one-to-many relationships. For the fundraising events table the primary key is event ID and there are no foreign keys. For the member table the primary key is member ID and the foreign key is committee name because the one-to-many relationship between the member table and the committee table has the “many” going towards the member table.

For the donor table the primary key is donor ID and there are no foreign keys. For the cost table the primary key is cost ID and the foreign key is vendor ID. Vendor ID is the foreign key because of the one-to-many relationship with the cost table and the vendor table (the many going towards the cost table). For the vendor table the primary key is vendor ID and the foreign key is event ID for the fact that the one-to-many relationship between the vendor table and the fundraising events table has the “many” going towards the vendor table. For the committee fundraising events table it consists of two primary keys, committee name and event ID, the same two keys are also the foreign keys because the table is an associative entity.

For the committee table primary key is committee name and there are no foreign keys. For the donation table the primary key is donation ID and the two foreign keys are donor ID and event ID. There are two foreign keys to this table because the donation table is involved with two one-to-many relationships with the fundraising event table and the donor table in which the “many” is going towards the donation table in both.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Primary Key | Attributes | | | | | | | |  |
| Cost | CostID | **\*VendorID\*** | Category | Description | Dollar Aount | Amount Paid | Associated Vendor |  |  |  |
| Vendor | VendorID | **\*EventID\*** | Name | Address | City | State | Zip Code |  |  |  |
| Fundraising Events | EventID | Name | Location | Date | Starting Time | Ending Time | Breif Description |  |  |  |
| Committee Fundraising Events | **\*Committee Name\*** |  |  |  |  |  |  |  |  |  |
| **\*EventID\*** |  |  |  |  |  |  |  |  |  |
| Committee | Committee Name | Mission | # of Members | Chair Person |  |  |  |  |  |  |
| Member | MemberID | **\*Committee Name\*** | First Name | Last Name | Address | City | State | Zip Code | Phone Number |  |
| Fax Number | Email Address | Membership Status | Community | Mail or Email | Spouse First Name | Spouse Last Name | Child First Name |  |
| Child Last Name | Child DoB | Member Skills |  |  |  |  |  |  |
| Donation | DonationID | **\*DonorID\*** | **\*EventID\*** | Donation purpose | Associated event | Monetary value | Member responsible | Type of donation | Percent received |  |
| Donor | DonorID | Business/Individual | Business Name | Donor First Name | Donor Last Name | Address | City | State | Zip Code |  |
| Phone Number | Email Address |  |  |  |  |  |  |  |
| \*Bold Faced and Asterisked Attributes are Foreign Keys\* | | | | | | | | | |  |

Normalization is a process that is based on functional dependency and ensures schema quality by minimizing, but not eliminating, data redundancy. The cost table is a relation, which it first normal form (1NF). The table also has a single subject, therefore it is in second normal form (2NF). Finally the cost table does not have any zip code problems, so it is in third normal form (3NF) with no issues at all. There are also no changes to the primary and foreign keys, with the primary key still being cost ID and the only foreign key still being vendor ID.

The second table, vendor, is a relation, which also makes it in 1NF. It also only has a single subject which puts it in 2NF. Finally, vendor is not in 3NF because it has a zip code problem. There are three steps that must be followed to move the vendor table to 3NF. First you must create a zip code table and make the primary key zip code, and then you must add the fields of city and state as well. This allows you to remove the fields of city and state from the vendor table and that eliminates the redundancies. However you must change the zip code field to a foreign key in the vendor table so the user can refer back to the zip code table to see the city and state that the vendor is from. After these steps are completed, the table vendor is in 3NF. The primary key is still vendor ID and now there are two foreign keys, which are event ID and zip code.

The third table is fundraising events, and it too is a relation, which makes it in 1NF. It also only has a single subject which puts it in 2NF. The fundraising events table also does not have any zip code, therefore it is in 3NF with no issues. The primary key is also still event ID and there are no foreign keys.

The next table is committee and it is a relation, putting it in 1NF. There is only a single subject as well for the table, which puts it in 2NF. Finally the committee table doesn’t have any zip code, therefore it is already in 3NF. The primary key remains as committee name and there are no foreign keys.

Next is the member table, which is a relation and it is in 1NF. There is not a single subject yet which means it is not in 2NF. To make the table have a single subject, spouse and children must be changed. After completing this, the spouse and children problems must still be addressed. To fix this problem, a new table needs to be created called family and the primary key becomes member ID. The spouse’s first name and last name must also be two fields in the newly created table along with the children’s first and last names and the child’s date of birth. This allows you to remove the fields of spouse’s first name and spouse’s last name from the member table along with the child’s first name, last name, and date of birth. Also the member ID field must be made a foreign key as well so users can refer to the family table to find the rest of the information. Now the member table is in 2NF. There is also a zip code problem so it is not in 3NF yet. Because the zip code problem was already addressed, it is not necessary to create another table, but it is still required that you remove the city and state fields from the member table. Then it is also mandatory to make the zip code field a foreign key so the users can refer back to the zip code table to see what city and state the members are from. After completing all these steps, the member table is in 3NF. The primary key remains as member ID and there are now three foreign keys, which are committee name, zip code, and member ID.

After that, the next table is donations, which is a relation and that makes it in 1NF. There is only a single subject, and that puts it in 2NF. Finally there are no zip code, therefore it is in 3NF already. The primary key is still donation ID and the foreign keys are still donor ID and event ID.

The final table is donors, and it is a relation, therefore it is in 1NF. This table has a single subject, therefore it is in 2NF. Finally there is a zip code problem, so the problem must be addressed before donors can be in 3NF. There is already a zip code table, so one does not need to be created, but the fields city and state still need to be removed from the donors table. However in the donor table the zip code field must be made a foreign key so the user can refer back to the zip code table to see the city and state that the donors are from. After completing this, the donor table is in 3NF. The primary key is still donor ID and now there is a foreign key which is zip code.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Primary Key | Attributes | | | | | | | |
| Cost | CostID | **\*VendorID\*** | Category | Description | Dollar Aount | Amount Paid | Associated Vendor |  |  |
| Vendor | VendorID | **\*EventID\*** | **\*Zip Code\*** | Name | Address |  |  |  |  |
| FundraisingEvents | EventID | Name | Location | Date | Starting Time | Ending Time | Breif Description |  |  |
| CommitteeFundraisingEvents | **\*Committee Name\*** |  |  |  |  |  |  |  |  |
| **\*EventID\*** |  |  |  |  |  |  |  |  |
| Committee | Committee Name | Mission | # of Members | Chair Person |  |  |  |  |  |
| Member | MemberID | **\*Committee Name\*** | **\*Zip Code\*** | **\*Spouse ID\*** | **\*Child ID\*** | First Name | Last Name | Address | Phone Number |
| Fax Number | Email Address | Membership Status | Community | Mail or Email | Marrital Status | # of Children | Member Skills |
| Donation | DonationID | **\*DonorID\*** | **\*EventID\*** | Donation Purpose | Associated Event | Monetary Value | Member Responsible | Type of Donation | Percent Received |
| Donor | DonorID | **\*Zip Code\*** | Business/Individual | Business Name | Donor First Name | Donor Last Name | Address | Phone Number | Email Address |
| Zip Code | Zip Code | City | State |  |  |  |  |  |  |
| Family | **\*MemberID\*** | Spouse First Name | Spouse Last Name | Child First Name | Child Last Name | Child DoB |  |  |  |
| \*Bold Faced and Asterisked Attributes are Foreign Keys\* | | | | | | | | | |