Database Design

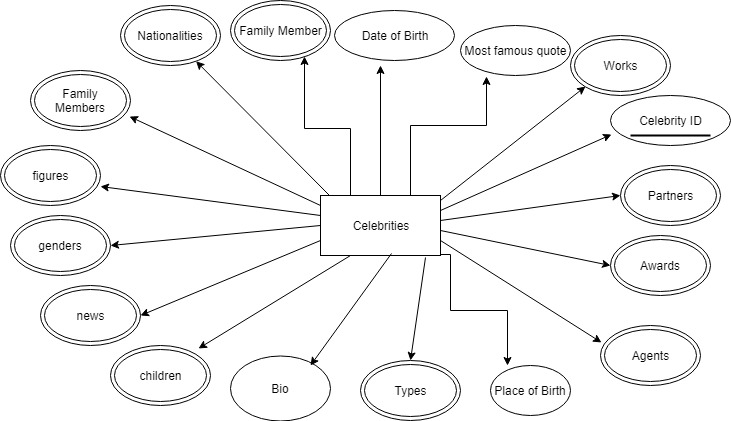
**Business requirements**

* I want to hire someone to create a database that is able to store the information of the celebrities for a celebrity info website.
* The database needs to include the information from basic information like bio to advanced info like partners, rumors, figures and so on.
* People can search the whole file, which includes the works, news, and etc., of a certain celebrity just by typing in the name.
* Meanwhile, the database needs to store the number of visits for each celebrity, the last commercial advertisement paid amount, and the agent’s info.

**Translate as technical specifications**

* It’s a celebrity information website so everything should be tied to each celebrity. There should be a celebrity’s info table which connects to every other table in the database.
* The basic information like bio or date of birth should be included in the celebrity’s info table because the information is closely related to each specific celebrity.
* The advanced information like partners, news should each be stored in separate tables because each celebrity can have multiple partners or news.
* All the tables should be joined together in one to one, one to many, or many-to-many relationship so that people can search the whole file of a certain celebrity just by typing in the name.
* The amount of visit and last commercial paid should be able to be updated and can’t be null.

**Assumptions**

* Varchar (100) should be enough for all name, news title, celebrity’s nationality, and his or her place of birth. Varchar (50) should be enough for a celebrity’s gender.
* The date of a celebrity’s info added in the system is the date a person become a celebrity. Thus, it can’t be null.
* Smallint should be enough for all body measurements for a celebrity.
* Bigint should be enough for amount of visit, amount of views, amount of read.
* Varchar (max) should be enough for bio, most famous quote, news content, and work description.
* Decimal (2,1) should be the format of the reviews.
* The amount of read for the latest news of a celebrity means the current heat of a celebrity.
* A celebrity can’t have more than one gender, body measurements, place of birth, nationality, bio, and name at the same time.
* The database only stores the current agent of a celebrity. The names of the agents are all different.

**ER Diagram**

First

The reason why my diagram is in 3NF is as follow. First, my ER diagram is in first and second normal form. Second, all the tables contain only columns that are non-transitively dependent on the primary key.

**Explanation of entities and attributes**

* Celebritiesinfos

The purpose of this entity is to store those basic information which is closely dependent on each celebrity. Every attribute in this entity is dependent on the primary key, celebrity ID. Every attribute is unique to each celebrity in this entity. This entity can have as much records as the number of celebrities. I put 14 records in this table.

* + Attributes in this entity
    - Cname means a celebrity’s name, include only first and last name.
    - CID is a celebrity’s unique ID that differentiate them.
    - Amount of visit means how many times the celebrity’s information has been viewed.
    - Last commercial ads paid means how much a celebrity got paid for his or her last commercial advertisement. It provides a business value or price for a celebrity.
    - Most famous quote could be something a celebrity has said in public or movie lines. If a company wants to find a relationship between its brand and a celebrity, it can brute force search those quotes and see if there’s any related result.
    - Date add in the system is the date when a person officially become a celebrity. It tells us how long a person has been a celebrity.
* Agents

It stores the name of agents for celebrities. There’re 12 records in this table. The only one attribute under this table is the agent’s name, which is the primary key of this table.

* CelebritiesAgents

It’s a junction table which connect celebrities info and agents table. The primary key of this entity is a composite key which consists of agent’s name and celebrity ID. Agent’s name and celebrity ID are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship. The start date under this table is the date a celebrity starts to work with that agent.

* Awards

The purpose of this table is to store all the awards that celebrities could have won. By creating this table, a celebrity can different awards and multiple celebrities can share a same award. I have 17 records in this table. There’re two attributes under this entity, award name and award ID, which makes every award unique.

* CelebritiesAwards

It’s a junction table which connect celebrities info and awards table. The only two attributes under this table are award ID and celebrity ID, which are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship.

* Figures

The purpose of this entity is to store some body measurements of a celebrity. The primary key of the table is the same as the celebrities info table, so the relationship between two table is one to one. It has the same number of records as celebrities info table.

The unit for the height, bust, waist, and hip is centimeter. The unit for the weight is pound.

* News

The purpose of this table is to store all the news that related to the celebrities in the database. By creating this table, a celebrity can be in different news and multiple celebrities can share a same news. I have 16 records in this table.

* + Attributes in this entity
* Title and content

Companies can brute force search them for any product related words.

* Amount of read

It shows the current heat of a celebrity.

* CelebritiesNews

It’s a junction table which connect celebrities info and news table. The only two attributes under this table are news ID and celebrity ID, which are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship.

* Partners

The purpose of this table is to store all the partners that all celebrity can have separately. By creating this table, a celebrity can have different partners and multiple celebrities can share a same partner. I have 9 records in this table. The Bio in this table is the biography for a partner.

* PartnersWhoAreCelebrities

It’s a junction table which connect celebrities info and partners table. The primary key of this entity is a composite key which consists of partner ID and celebrity ID. Partner ID and celebrity ID are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship. The relationship under this table is the type of relationship between a celebrity and his or her partner.

* Typesinfo

The purpose of this table is to store all the types that all celebrity can have. By creating this table, a celebrity can have different types and multiple celebrities can share a same type. I have 10 records in this table.

* CelebritiesTypes

It’s a junction table which connect celebrities info and Typesinfo table. The primary key of this entity is a composite key which consists of type ID and celebrity ID. Type ID and celebrity ID are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship.

* Works

The purpose of this table is to store all the works that all celebrity can have. By creating this table, a celebrity can have different works and multiple celebrities can share a same work. I have 9 records in this table.

* + Attributes in this table
* Review is the rate of a work from viewers. It provides a judgement for companies and users.
* Amount of views is the times that a work has been viewed.
* Formats

The purpose of this table is to store all the formats that all works in this database can have. By creating this table, a work can have different formats and multiple works can share a same format. I have 6 records in this table. The only attribute under this table, Ftype, which is the primary key, is the name of a format.

* Publishers

The purpose of this table is to store all the publishers that all works in this database can have. By creating this table, a work can have different publishers and multiple works can share a same publisher. I have 8 records in this table. The only attribute under this table, Pname, which is the primary key, is the name of a publisher.

* WorkFormats

It’s a junction table which connect Formats and Works table. The primary key of this entity is a composite key which consists of work ID and Ftype. Work ID and Ftype are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship.

* WorkPublishers

It’s a junction table which connect Publishers and Works table. The primary key of this entity is a composite key which consists of work ID and Pname. Work ID and Pname are foreign keys reference the primary keys in that two tables. It creates the many-to-many relationship.