



Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México

Irving Fuentes Aguilera	A01745759
Joaquín Ríos Corvera	A01375441
José Antonio Vázquez Gabián	A01746585
Jaime Orlando López Ramos	A01374696

ROMA Music: Documentación

Introducción

Para este Mini Proyecto se nos pidió aplicar los conceptos aprendidos en clase acerca de las bases de datos transaccionales, de modo que pudiéramos crear una base de datos indexada. Este tipo de base de datos se caracteriza por estar embebida al buscador web. Este tipo de modelo de bases de datos es orientado objetos y NOSql, lo que quiere decir que nos facilita la implementación de un sistema jerarquizado, además de que hace más simple el proceso de adición de registros.

Modelo de la Base de Datos

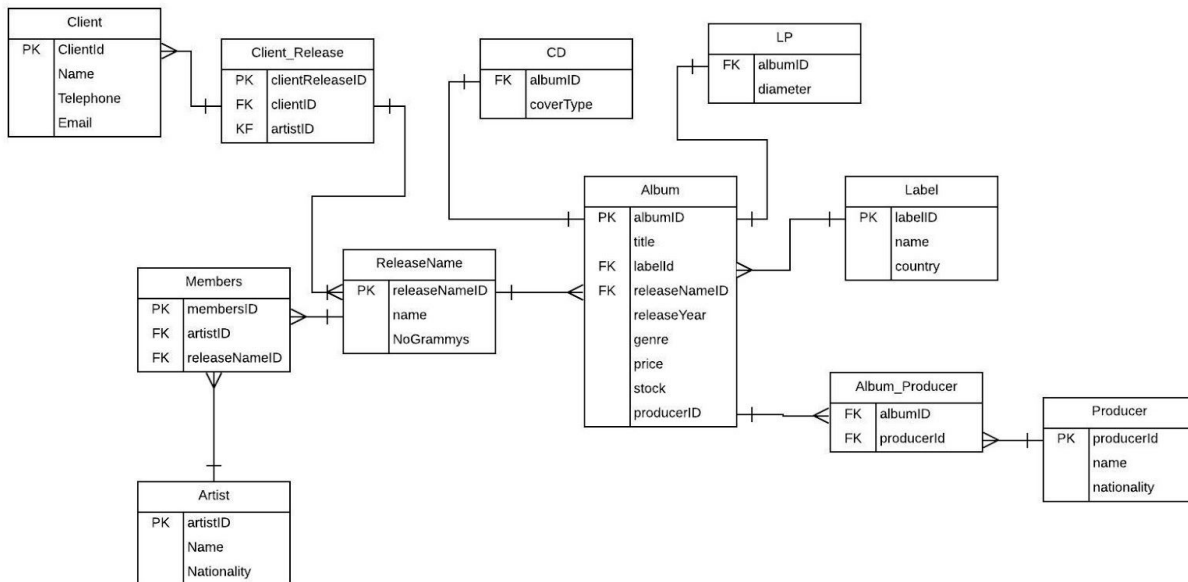
La idea del proyecto es simular una base de datos para una sucursal de una tienda de música, tomando en cuenta lo que requerían. Diseñamos una descripción de los requerimientos del sistema, de modo que estuviera claro lo que se deseaba modelar.

Dicha descripción es la siguiente:

- Se requiere de una base de datos para una tienda pequeña dedicada a la venta de música en todos sus formatos, ya sea cds o LPs. Se tiene un registro de los artistas que hay en la tienda, los discos que venden, y el género al que pertenecen. A cada álbum se le asocia un género, y un artista puede pertenecer a más de un género.
- La tienda vende 2 tipos de discos. CDs y LPs, cada uno tiene su propia unique ID. Los describe su título, artista, año de lanzamiento, género, productor principal, disquera, precio, stock, y país de origen. Tanto LPs como CDs son descritos como edición estándar o edición deluxe.
- El CD puede ser de portada de plástico u otro material.

- El disco puede ser de un artista o de una banda, una banda está conformada por muchos artistas. El artista está descrito por su nombre, nacionalidad, y su unique id.
- Se guarda la información del cliente para poder notificarles cada vez que haya un nuevo disco disponible de su artista favorito.
- El cliente registra una lista de artistas que le gustan para poder notificarle de cada uno. De igual forma el cliente junta puntos cada vez que compra un disco registrado en su lista.
- El cliente está descrito por un nombre, apellido, teléfono, correo y unique id. De igual forma cada cliente tiene su propia lista que puede estar conformada por diversos artistas o bandas.

Al analizar los requerimientos, se modelaron los datos en distintas tablas dentro de un diagrama Entidad Relación, de modo que se tuviera una representación visual del sistema completo. Dicho diagrama es el siguiente:



Diseño de la aplicación

El diseño de esta aplicación está basada en una herramienta llamada bootstrap; la cual crea prototipos rápidamente o crea toda su aplicación con variables y mixins Sass, sistema de cuadrícula receptiva, componentes precompilados extensos y complementos potentes creados en jQuery. Desde aquí se dio una definición sobre ciertas cosas como el color, tamaño, etc. El diseño fue basado en herramientas HTML con CSS las cuales ayudaron a dar definición de la página, y en parte para la funcionalidad se usó como antes mencionado Bootstrap, jQuery y Javascript. jQuery nos brindó la manera de identificar los atributos o campos que el usuario debe llenar para crear el registro, bootstrap nos dio un panorama más amplio para crear los containers y los form para las acciones de los botones y hacer un mensaje “Success” o “Please

fill all fields”, CSS se utilizó en la parte de ciertas propiedades de color, tamaño, letra, dentro del html en cierta sección del programa, Javascript fue explotado en la parte de la estructura de programación (declaración de variables, etc). Firebase almacena y sincroniza datos con una base de datos NoSQL alojada en la nube. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen disponibles cuando la app no tiene conexión. Basado en una funcionalidad Firestore que usa una base de datos NoSQL flexible, escalable y en la nube a fin de almacenar y sincronizar datos para la programación en el lado del cliente y del servidor. Por último JSON, se utiliza como la creación de archivos dentro de Firebase y es un formato de texto sencillo para el intercambio de datos.

Referencias y Librerías:

<https://www.gstatic.com/firebasejs/6.2.0/firebase-app.js>

<https://www.gstatic.com/firebasejs/6.2.0/firebase-auth.js>

<https://www.gstatic.com/firebasejs/6.2.0/firebase-firestore.js>

<https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js>

<https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js>

<http://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js>

https://www.w3schools.com/js/js_jquery_elements.asp