

УДК 004.021

**ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ
ДЛЯ МУЗЫКАЛЬНОГО СТРИМИНГОВОГО ВЕБ-СЕРВИСА «BLZMUSIC»****Д.С. ТАТАРИН***(Представлено: канд. тех. наук И.Б. БУРАЧЁНОК)*

Рассматриваются цели и принципы моделирования базы данных на примере музыкального стримингового веб-сервиса «BlzMusic». В спроектированной базе данных обосновывается выбор основных сущностей и связей между ними.

Трудно представить хороший сайт с большой аудиторией без разработанной базы данных. Они позволяют удобно хранить большие объемы информации не на устройстве пользователя, а на удаленном сервере, который может эти данные предоставить по запросу от приложения.

В представленной статье приведем описание базы данных для разработанного веб-сервиса «BlzMusic» и особенностей ее проектирования.

Цель моделирования данных состоит в обеспечении разработчика концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных. Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» или Entity-Relationship Model (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных. Нотация ERD была впервые введена П. Ченом и получила дальнейшее развитие в работах Баркера. Методология Integration Definition For Information Modeling (IDEF1), разработанная Т. Рэмей, основана на подходе П. Чена и позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме [1].

Сущность – это реальный или представляемый тип объекта, информация о котором должна сохраняться и быть доступной. В диаграммах сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности – это имя типа, а не некоторого конкретного экземпляра этого типа. Каждый экземпляр сущности (объект) должен быть отличим от любого другого экземпляра той же сущности [2].

Для реализации музыкального стримингового веб-сервиса «BlzMusic» в качестве базы данных был выбран MS SQL Server 2017, это обусловлено следующим.

- Масштабируемость. Алгоритмы использования дискового пространства SQL Server 7.0 масштабируются в диапазоне от портативных компьютеров начиная с Windows версии 3.11.
- Превосходная производительность. Увеличенный до размер страниц способствует быстрому извлечению данных, позволяет использовать строки и столбцы большего размера, что открывает возможность эффективного хранения сложных, подробных данных.
- Простота использования. Динамическое самоадминистрирование. Выполнение многих рутинных задач администрирования теперь автоматизировано. Алгоритмы управления памятью и блокированием адаптируются динамически, размер файлов автоматически увеличивается и сокращается. Кроме того, средства автоматической настройки динамически настраивают алгоритмы использования ресурсов в зависимости от рабочей нагрузки.

Для построения самого приложения был выбран Java – объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно компилируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине (JVM) вне зависимости от компьютерной архитектуры. Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Достоинство подобного способа выполнения программ – в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина [3]. Java является достаточно мощным инструментом разработки приложений, к тому же он имеет огромное комьюнити и большим количеством учебных курсов.

Для разработки интерфейса было решено использовать: html, css, js, что позволит создать приемлемый пользовательский интерфейс, который облегчит работу пользователю, который будет работать с приложением. Более того на данный момент сайты набирают все большую и большую популярность, потому что намного проще «зайти» на сайт, чем скачивать и ставить какое-либо приложение.

На первоначальном этапе разработки, с целью построения базы данных, был проведен анализ предметной области и выявление необходимого набора сущностей. База данных предназначена для хранения данных о музыке (исполняемых музыкальных произведениях), группах, артистах, концертах, жанрах, альбомах, пользователях. Для каждого трека и альбома должна присутствовать таблица с принадлежностью к какой-либо группе или артисту. У каждой группы должен присутствовать один или несколько жанров. Для каждого концерта должно быть указано к какой группе он относится. Также должна присутствовать таблица, где хранится логин и пароль пользователя и связь с его ролью, логином и корзиной. Следовательно, определяем роль, корзину и логин как отдельные сущности.

В результате краткого анализа предметной области выявлен необходимый набор сущностей:

- Music (Музыка);
- Band (Группа);
- Concert (Концерты группы);
- Genre (Жанры групп);
- Album (Альбом групп);
- Cart (Корзина пользователя);
- User (Пользователь);
- Login (Информация о логине пользователя);
- Role (Роль пользователя).

Связь – это графически изображаемая ассоциация, устанавливаемая между двумя сущностями. Связь может существовать между двумя разными сущностями или между сущностью и ей же самой (рекурсивная связь) [2].

Для реализации приложения, необходимо установить все связи между сущностями: необходимо рассмотреть всю информационную систему в совокупности и определить взаимное влияние сущностей.

Полученная схема базы данных представлена на рисунке.

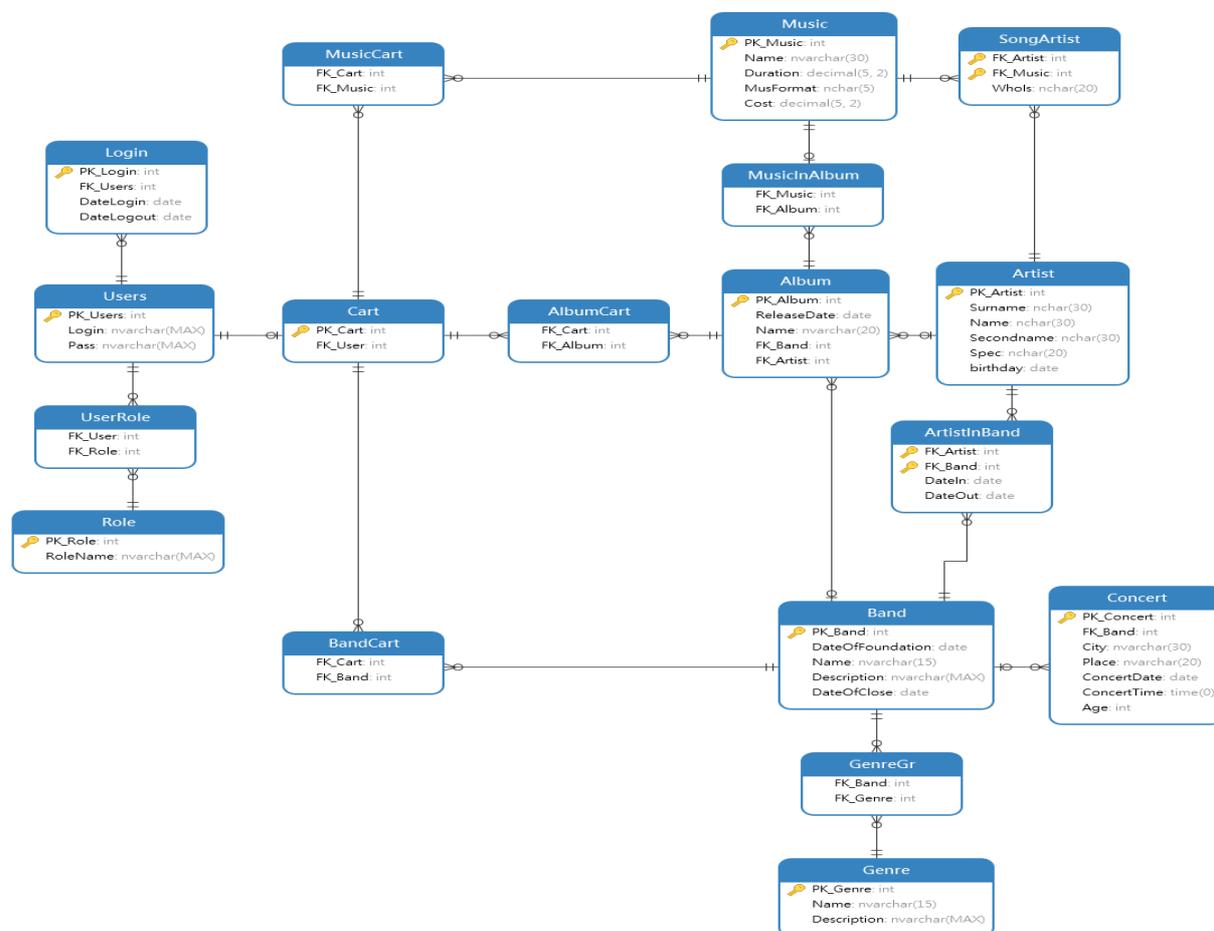


Рисунок. – Схема базы данных

Реляционная модель представляет базу данных в виде множества взаимосвязанных отношений. Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы – атрибутам отношений, доменам, полям. В соответствии выделенными выше сущностями, определен набор необходимых таблиц базы данных, представленный в таблице 1.

Таблица 1. – Перечень таблиц базы данных

Название таблицы	Описание
Music	Хранение данных о всех треках
MusicCart	Хранение данных о всех треках в корзине пользователя
MusicInAlbum	Хранение данных о треках состоящий в альбоме
Artist	Хранение данных о всех артистах
SongArtist	Хранение данных о треках артистов
ArtistInBand	Хранение данных о артистах в группе
Album	Хранение данных о всех альбомах
AlbumCart	Хранение данных о альбомах в корзине пользователя
Band	Хранение данных о всех группах
BandCart	Хранение данных о группах в корзине пользователя
Concert	Хранение данных о концертах группы
Genre	Хранение данных о жанрах
GenreGr	Хранение данных о жанрах группы
Cart	Хранение данных о корзине пользователя
Users	Хранение данных о пользователях
Login	Хранение данных о логине пользователя
Role	Хранение данных о роли
UserRole	Хранение данных о роли пользователя

Важное достоинство баз данных – возможность хранить сгруппированные данные в разных таблицах и устанавливать связи между этими таблицами посредством ключей и последующее объединение этих данных в единую базу. Это позволяет снизить избыточность данных, которые хранятся в базе данных, упростить организацию запросов для их извлечения.

Ключ сущности – первичный ключ – это атрибут (или множество атрибутов) уникальным образом идентифицирующих экземпляр сущности (объект). Например, ключ сущности Band – уникальный номер, ключ Music – уникальный код песни. Если ключ состоит из одного атрибута, его называют простым ключом. Если ключ сущности состоит из нескольких атрибутов, его называют составным ключом.

Для организации связей между двумя и более таблицами, используются вторичные ключи. Они служат ограничителями целостности связей нескольких таблиц, т.к. подчиненная таблица не может ссылаться на несуществующие записи главной таблицы (что позволяет строить целостные модели данных). Существует три вида связей между таблицами:

- один-к-одному;
- один-ко-многим;
- многие-ко-многим.

Отношение «один-ко-многим» имеет место, когда одной записи родительской таблицы может соответствовать несколько записей в дочерней таблице [4]. Связь «один-ко-многим» является самой распространенной для реляционных баз данных. Отношение «один-к-одному» имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней таблице [4]. Отношение «многие-ко-многим» имеет место, когда [4]:

- записи в родительской таблице может соответствовать больше одной записи в дочерней таблице;
- записи в дочерней таблице может соответствовать больше одной записи в родительской таблице.

Например, каждая группа имеет несколько жанров. К каждому жанру относятся несколько групп (табл. 2).

Таблица 2. – Первичные и внешние ключи отношений

Название таблицы	Первичный ключ	Внешние ключи
Album	PK_Album	FK_Band
Music	PK_Music	
Artist	PK_Artist	
Band	PK_Band	
Concert	PK_Concert	FK_Band
Genre	PK_Genre	FK_Subtype
Cart	PK_Cart	FK_User
User	PK_User	
Login	PK_Login	FK_User
Role	PK_Role	

Таким образом была спроектирована база данных музыкального стримингового веб-сервиса «BlzMusic». Были выделены 18 таблиц, которые находятся в ЗНФ. Написаны скрипты создания и наполнения базы данных, поэтому базу можно легко развернуть на любой СУБД.

ЛИТЕРАТУРА

1. Case-метод Баркера [Электронный ресурс]. – Режим доступа: http://citforum.ru/database/case/glava2_4_1.shtml. – Дата доступа: 14.09.2018.
2. Основные понятия баз данных [Электронный ресурс]. – Режим доступа: http://inf.susu.ac.ru/Klinachev/lc_sga_26.htm. – Дата доступа: 14.09.2018.
3. Основные понятия БД [Электронный ресурс]. – Режим доступа: <http://informatic.ugatu.ac.ru/lib/office/proekt.htm>. – Дата доступа: 14.09.2018.
4. Подробнее о технологии Java [Электронный ресурс]. – Режим доступа: <https://java.com/ru/about>. – Дата доступа: 14.09.2018.