

УДК 004.658

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ДЛЯ МУЗЫКАЛЬНОГО СТРИМИНГОВОГО ВЕБ-СЕРВИСА

М.Р. ГРАДЮШКО

(Представлено: О.Н. ПЕТРОВИЧ)

В данной статье рассмотрены основные цели и принципы моделирования базы данных, на основе которых спроектировано хранилище информации музыкального стримингового веб-сервиса «Bro-Music».

Введение. На данный момент для большинства людей трудно представить свою жизнь без музыки – она окружает нас повсюду, её разнообразие огромно, а доступность легка. Музыкальный рынок как никогда актуален и широк, благодаря интернету распространение музыки стало весьма лёгкой задачей. Поэтому необходимы стриминговые веб-сервисы, которые дают возможность пользователям в полной мере насладиться музыкой, именно для одного из таких приложений и необходимо спроектировать базу данных.

Данная статья направлена на описание базы данных для разрабатываемого веб-сервиса и методов его проектирования.

Основной раздел. Разработка базы данных осуществляется с помощью моделирования данных. Цель моделирования данных состоит в обеспечении разработчика ИС 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" (ERD). С помощью ERD осуществляется детализация накопителей данных DFD – диаграммы, а также документируются информационные аспекты бизнес-системы, включая идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их связей с другими объектами (отношений).

Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности. Каждая сущность должна обладать некоторыми свойствами:

– иметь уникальное имя; к одному и тому же имени должна всегда применяться одна и та же интерпретация; одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами;

– иметь один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;

– иметь один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности.

Каждая сущность может обладать любым количеством связей с другими сущностями модели. [1]

В ходе анализа необходимых возможностей приложения и основ проектирования базы данных были выявлены следующие сущности.

Сущность *Пользователь* описывает пользователя, имеющего доступ к сайту. Характеризуется логином, паролем и ролью. Для него необходимо выделить следующие подсущности: *Роль* (определяет права пользователя), *Посещение сайта* (необходимо для ведения статистики посещения сайта), *Голосование* (предназначено для хранения понравившихся пользователю песен), *Комментарий* (предназначен для хранения комментариев пользователя к песням).

Сущность *Песня* хранит описывает музыкальную композицию. Для нее можно выделить следующие подсущности: *Жанр песни* и *Музыкант*, принимающий участие в создании песни. Для связи музыкантов и песни была добавлена сущность *Музыканты песни*.

Сущность *Группа* является основной и хранит всю информацию о музыкальной группе. Для нее можно выделить следующие подсущности: *Репертуар* (песни, принадлежащие группе), *Состав группы* с указанием *Амплуа музыкантов*, *Тип группы*, *Страна происхождения*.

Отдельной сущностью можно выделить *Тур*, который хранит информацию о концертах группы. Для нее можно выделить следующие подсущности: *Город проведения концерта*, *Концертный зал*, *Концерт*.

Атрибуты сущности служат для уточнения, идентификации, классификации, числовой характеристики или выражение состояния сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различных типов сущности.

Теперь для каждой сущности необходимо определить атрибуты:

1. Сущность «Пользователь»: *Логин*, *Пароль*, *Роль*, *Email*.
2. Сущность «Роль»: *Название роли*.
3. Сущность «Посещение сайта»: *Пользователь*, *Дата входа*, *Дата выхода*.

4. Сущность «Голосование»: *Пользователь, Дата голоса, Песня.*
5. Сущность «Комментарий»: *Пользователь, Дата создания, Песня, Содержимое комментария.*
6. Сущность «Песня»: *Название песни, Дата выпуска, Изображение, Трек, Жанр.*
7. Сущность «Жанр песни»: *Название жанра.*
8. Сущность «Музыкант»: *Имя, Отчество, Фамилия, Дата рождения.*
9. Сущность «Музыканты песни»: *Песня, Музыкант, Статус музыканта.*
10. Сущность «Группа»: *Название группы, Дата начала карьеры, Дата окончания карьеры, Изображение, Тип группы, Страна происхождения.*
11. Сущность «Репертуар»: *Группа, Песня, Дата начала, Дата окончания.*
12. Сущность «Состав группы»: *Группа, Музыкант, Дата начала, Дата окончания, Амплуа музыканта.*
13. Сущность «Амплуа музыканта»: *Название амплуа.*
14. Сущность «Тип группы»: *Название типа.*
15. Сущность «Страна происхождения»: *Название страны.*
16. Сущность «Тур»: *Название тура, Дата начала, Дата окончания, Группа.*
17. Сущность «Город»: *Название тура, Численность населения, Страна.*
18. Сущность «Концертный зал»: *Название зала, Город.*
19. Сущность «Концерт»: *Тур группы, Концертный зал, Дата начала, Дата окончания, Средняя цена билетов.*

Для каждой сущности выделим свой идентифицирующий атрибут, что представлено в таблице 1.

Таблица 1. – Первичные и внешние ключи отношений.

№	Название таблицы	Первичный ключ	Внешние ключи
1	Users	id	roleId
2	Logging	id	userId
3	UserRoles	id	отсутствуют
4	Voices	id	loggingId songId
5	Comments	id	loggingId songId
6	Repertoires	id	groupId songId
7	Songs	id	genId
8	Genres	id	отсутствуют
9	Musicians	id	отсутствуют
10	MapSongMusician	id	songId musicianId
11	Groups	id	groupTypeId countryId
12	GroupTypes	id	отсутствуют
13	CompositionGroup	id	musicianId groupId professionId
14	Professions	id	отсутствуют
15	Tours	id	groupId
16	Countries	id	отсутствуют

Одно из важнейших достоинств реляционных баз данных состоит в том, что можно хранить логически сгруппированные данные в разных таблицах и задавать связи между ними, объединяя их в единую базу. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов.

Существует три разновидности связей между таблицами базы данных:

1. «один-ко-многим». В типе связей один ко многим одной записи первой таблицы соответствует несколько записей в другой таблице.

2. «один-к-одному». Связь один к одному образуется, когда ключевой столбец (идентификатор) присутствует в другой таблице, в которой тоже является ключом либо свойствами столбца задана его уникальность (одно и тоже значение не может повторяться в разных строках).

3. «многие-ко-многим». Если нескольким записям из одной таблицы соответствует несколько записей из другой таблицы, то такая связь называется «многие-ко-многим» и организовывается посредством связывающей таблицы [2].

Связи выполняют более важную роль, чем просто информация размещения данных по таблицам. Прежде всего они требуются для поддержания целостности баз данных.

Правильно настроив связи, можно быть уверенным, что ничего не потеряется, поэтому необходимо ответственно отнестись к этому вопросу. Для реализации информационной системы музыкального сайта необходимо установить связи между таблицами ее базы данных, что представлено на рисунке 1.

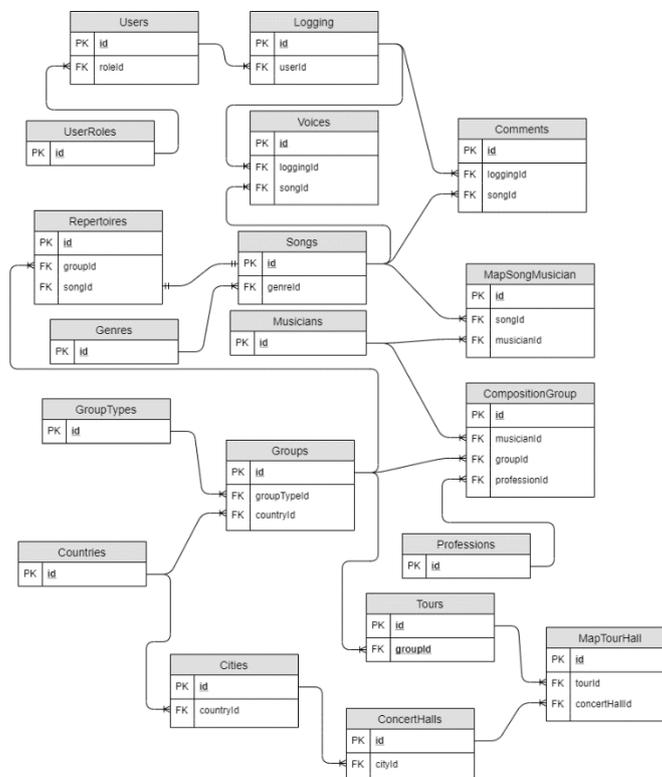


Рисунок 1 – Связи между сущностями базы данных

Заключение. В данной статье были рассмотрены основные цели и принципы моделирования базы данных для стримингового веб-сервиса «Bro-Music». Также были рассмотрены основные сущности спроектированной базы данных.

ЛИТЕРАТУРА

1. НОУ ИНТУИТ [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/2195/55/lecture/1636>. – Дата доступа: 20.09.2020.
2. Базы данных [Электронный ресурс]. – Режим доступа: <http://informatic.ugatu.ac.ru/lib/office/Proekt.htm>. – Дата доступа: 21.09.2020.