

1 Логическое проектирование

Процесс проектирования представляет собой процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая [2].

Нормализация – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных [2].

Цели нормализации:

1. упорядочить данные в структуру таблиц, которые не имеют повторяющихся записей
2. минимизировать избыточность информации
3. гарантировать, что элементы данных в различных таблицах связаны правильными ключами [2]

Для создания базы данных необходимо определить структуру таблиц, которые будут содержать данные, и нормализовать эти таблицы.

Нормализация – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации.

База данных для выполнения задачи должна содержать следующую информацию: код вида дерева, вид дерева, код дерева, год посадки, привито несколько сортов, код сорта, название сорта.

Первая нормальная форма требует удаления повторяющихся групп, то есть создания отдельной записи для каждого элемента группы (таблицы 1.1).

Таблица «Дерево»

Таблица 1.1

Код дерев а	Код вида дерев а	Вид дерев а	Год посадки	Привито несколько сортов	Код сорта	Название сорта
12	545	яблоко	11.04.2005	да	112	абориген
14	536	груша	06.09.2006	нет	115	аллегро
8	538	вишня	01.08.2008	да	113	алтайская
3	535	яблоко	10.07.2007	да	109	авангард
1	544	слива	15.06.2010	нет	107	аскер
5	537	вишня	23.04.2009	да	103	болотовская
10	533	груша	30.07.2010	да	110	аннушка
15	540	яблоко	25.08.2006	нет	104	августа
7	543	слива	18.09.2005	да	105	балада
11	534	вишня	05.07.2007	нет	101	полевка
6	531	слива	06.09.2008	нет	111	волошка
2	542	вишня	03.06.2010	да	102	преемница
4	541	яблоко	02.06.2009	нет	106	антовка
9	539	груша	01.04.2008	нет	108	внучка
13	532	слива	13.03.2007	да	114	жигули

Вторая нормальная форма требует удаления функциональных зависимостей. Таблица находится во второй нормальной форме, если она удовлетворяет определению первой нормальной формы и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом

Так как поля Автор, Стиль, Страна и Вид диска могут повторяться, то логично присвоить каждому полю ключ и для избавления от функциональных зависимостей вынести в отдельные таблицы (табл. 1.3,

табл. 1.4, табл. 1.5, табл. 1.6).

Таблица **autors**

Таблица 1.3

Код автора	Автор
1	Red Hot Chill Peppers
2	Da Octopusss
3	#####
4	Brooklyn Bounce
5	Kazantip

Таблица **mus_styles**

Таблица 1.4

Код стиля	Стиль
1	Club
2	Rock
3	Soundtrack
4	Hardrock
5	House

Таблица **countries**

Таблица 1.5

Код страны	Страна
1	США
2	Россия
3	Франция
4	Казантип

Таблица **type_dsk**

Таблица 1.6

Код вида диска	Вид диска
1	Сольный
2	Концерт
3	Сборник

Так как созданы новые таблицы, необходимо в таблице первой нормальной формы (табл. 1.1) удалить поля: Автор, Стил, Страна и Вид диска. После удаления добавляются новые: Код автор, Код стиль, Код страны, Код вида диска (табл. 1.7).

Таблица **disks**

Таблица 1.7

Название альбома	Код автора	Код стиля	Количество композиций	Продолжительность	Год	Код страны	Код вида диска
Californication	1	2	15	73	1999	1	1
Banlieue	2	3	16	78	2004	3	3
Камикадзе	3	4	13	64	2006	2	2
Club Rotation	4	5	18	84	2013	3	3
Kazantip 2014 Euro session	5	1	20	96	2014	4	3

Таблица находится в третьей нормальной форме, если она удовлетворяет определению 2NF и ни одно из ее не ключевых полей не зависит функционально от любого другого не ключевого поля.

Следовательно, необходимо проверить таблицы на транзитивные зависимости.

В таблице **disks** зависимостей нет. Остальные таблицы имеют следующие зависимости:

- **Код автора** → **Автор** (Таблица **authors**).
- **Код стиля** → **Стиль** (Таблица **mus_styles**).
- **Код страны** → **Страна** (Таблица **countries**).
- **Код вида диска** → **Вид диска** (Таблица **type_dsk**).

Так как транзитивных зависимостей не обнаружено, можно сделать вывод, что таблицы находятся в третьей нормальной форме. На этом процесс нормализации заканчивается.

Спроектирована реляционная база данных, состоящая из пяти таблиц: таблица 1.7, 1.3, 1.4, 1.5, 1.6.

После определения структуры таблиц, связей между ними и ключей, которые будут использованы для связывания отдельных таблиц, можно создавать реляционную базу данных в СУБД.

На рис. 1.1 показана схема данных разработанной базы данных.

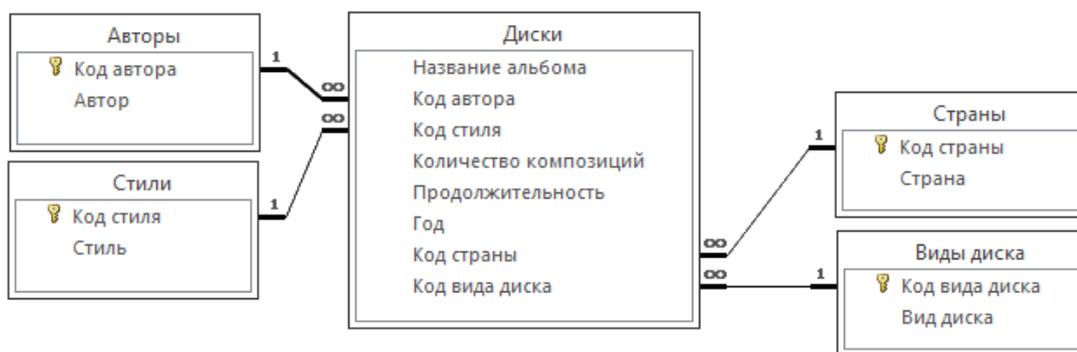


Рис. 1.1 Схема данных разработанной базы данных