

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Базы данных

Отчет по лабораторной работе №2
Разработка структур и нормализация БД

Работу выполнила:

Темнова А.С.

Группа: 43501/3

Преподаватель:

Мяснов А.В.

Санкт-Петербург
2016

Содержание

1	Цель работы	2
2	Программа работы	2
3	Ход выполнения работы	2
3.1	Содержание таблиц	2
3.2	SQL-схема БД	3
3.3	Приведение к 3НФ	3
4	Выводы	4

1 Цель работы

Знакомство с основами проектирования схемы БД, способами нормализации отношений в БД.

2 Программа работы

1. Представить SQL-схему БД, соответствующую заданию (должно получиться не менее 7 таблиц)
2. Привести схему БД к 3НФ
3. Согласовать с преподавателем схему БД. Обосновать соответствие схемы 3НФ.
4. Продемонстрировать результаты преподавателю

3 Ход выполнения работы

3.1 Содержание таблиц

Разработана схема базы данных в соответствии с выбранным заданием. Выбранная тема - винотека (WINE_SHOP.FDB).

Создание базы данных в firebird 2.5:

```
1 create database 'D:/database/wine_shop.fdb' user 'SYSDBA' password 'masterkey';  
2 commit;
```

Рассмотрим сущности и их атрибуты (см.табл. 1).

Таблица 1: Содержание базы данных

Основные таблицы			
Сущность	Атрибуты	Тип атрибутов	Комментарии
WINE (вино)	WINE_ID	integer	primary key
	NAME	varchar	название вина
	YEAR	integer	год вина (для вычисления выдержки вина)
	COLOR_ID	integer	foreign key
SHOP (магазин)	TYPE_ID	integer	foreign key
	SHOP_ID	integer	primary key
	NAME	varchar	название магазина
	PHONE	integer	номер телефона
SUPPLIER (поставщик вина)	ADDRESS	varchar	адрес
	SUPPLIER_ID	integer	primary key
	PHONE	integer	номер телефона
	ADDRESS	varchar	адрес
PRODUCER (производитель)	PRODUCER_ID	integer	primary key
	NAME	varchar	название фирмы
	PHONE	integer	номер телефона
	ADDRESS	varchar	адрес
COUNTRY (страна)	COUNTRY_ID	integer	primary key
TYPE	NAME	varchar	название страны
	TYPE_ID	integer	primary key
COLOR	NAME	varchar	название типа вина (сладкое, сухое и тд)
	COLOR_ID	integer	primary key
	NAME	varchar	название цвета

Таблицы связей "многие-ко-многим"			
Сущность	Атрибуты	Тип атрибутов	Комментарии
COUNTRY-PRODUCER	ID_CO_PR COUNTRY_ID PRODUCER_ID	integer integer integer	primary key foreign key foreign key
PRODUCER-WINE	ID_PR_WI PRODUCER_ID WINE_ID	integer integer integer	primary key foreign key foreign key
STORAGE (склад магазина)	STORAGE_ID SHOP_ID ID_SU_PR_WI NUMBER	integer integer integer integer	primary key foreign key foreign key кол-во вина
SUPPLIER-PRO-WINE	ID_SU_PR_WI SUPPLIER_ID ID_PR_WI	integer integer integer	primary key foreign key foreign key

3.2 SQL-схема БД

Схемы БД представлена на рисунке 1.

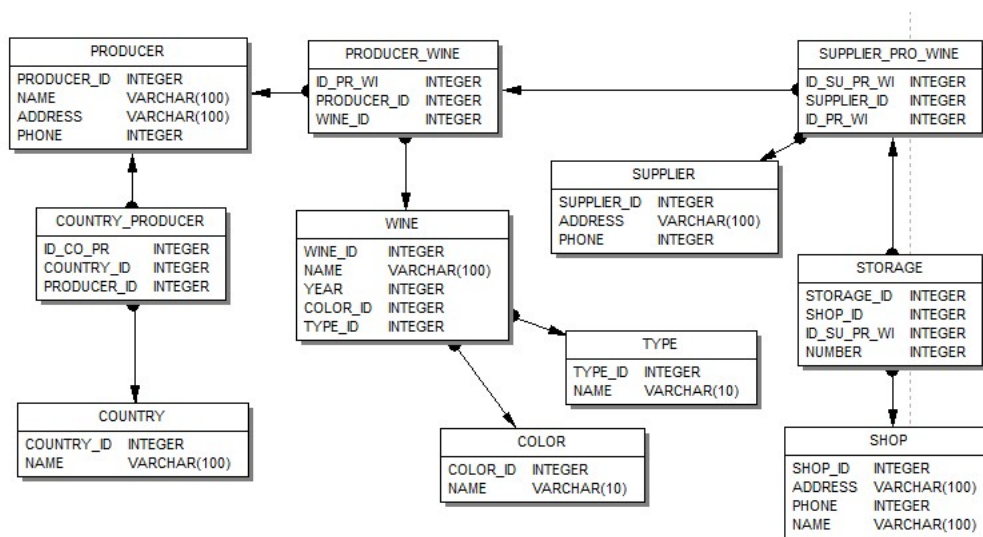


Рис. 1: Структура БД

В итоговой БД 11 таблиц и 10 связей по внешним ключам.

3.3 Приведение к 3НФ

1НФ - первая нормальная форма.

Отношение находится в 1НФ если значения всех его атрибутов атомарны. То есть 1НФ запрещает записывать в одну ячейку несколько значений, что в принципе уже ограничено средствами создания БД.

2НФ - вторая нормальная форма.

Неключевой атрибут функционально полно зависит от составного ключа если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

То есть должно быть соблюдено условие — любой столбец, который не является ключом (в том числе внешним), должен зависеть от первичного ключа. Обычно такие столбцы, имеющие значения, который не зависят от ключа, легко определить. Если данные, содержащиеся в столбце, не имеют отношения к ключу, который описывает строку, то их следует отделять в свою отдельную таблицу. В старую таблицу

надо возвращать первичный ключ.

ЗНФ - третья нормальная форма.

Отношение находится в ЗНФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

То есть требуется отсутствие «транзитивности» - это ситуация когда данные из второго столбца зависят от данных первого, а данные третьего, в свою очередь зависят от данных второго. Чтобы избежать такой ситуации таблица делится на две новых таблицы.

Схема приведена к ЗНФ. Все таблицы находятся во второй нормальной форме, в данной схеме нет транзитивных функциональных зависимостей.

4 Выводы

В результате проделанной работы я познакомилась с основами проектирования схемы БД и способами нормализации отношений в БД.

В отчете представлена БД в ЗНФ. В ходе ознакомления с нормализацией были выявлены ее плюсы и минусы.

Из плюсов: нормализация БД гарантирует надежность и жизнеспособность базы.

Из минусов: нормализация БД не всегда является лучшим решением и может иметь долгосрочные разрушающие последствия. Две основных причины, чтобы нарушить правила нормализации — удобство и быстрое действие. Меньшим числом таблиц проще управлять, чем большим. Кроме того, из-за более сложного характера, нормализованные таблицы более медленные для обновления, изменения и выдачи данных.

В противовес последнему, можно отметить, что есть достаточно способов улучшения производительности БД, но не так много способов, которые могут исправить поврежденные данные, возникшие из-за плохого дизайна структуры.