

Лабораторная работа № 10

«Реляционная база данных»

Теоретическая часть

Теория реляционных баз данных подробно рассматривается в курсе «Базы данных». В данной лабораторной работе решается задача имитации реляционной базы данных с помощью файловой системы языка C и библиотеки CRT. Рассматривается одна (общая для всех вариантов) учебная база данных «Поставщики и Продукты».

Замечание.

Сторонники реляционных баз данных ввели свою собственную терминологию и называют простые вещи специальными словами. Таблицы данных называются *отношениями*, строки таблиц – *кортежами*, столбцы таблиц – *атрибутами*. Количество кортежей в отношении называется его *кардинальностью*, а количество атрибутов – *степенью*. Совокупность значений определенного атрибута называется *доменом*. Домен – это не что иное, как *тип данных* с ограничением. Основное назначение домена – ограничить сравнение. Каждый кортеж имеет один или несколько *потенциальных ключей*. Непустое подмножество множества атрибутов отношения будет *потенциальным ключом* тогда и только тогда, когда оно будет обладать свойствами *уникальности* (в отношении нет двух различных кортежей с одинаковыми значениями потенциального ключа) и *неизбыточности* (никакое из собственных подмножеств множества потенциального ключа не обладает свойством уникальности). По традиции один из потенциальных ключей должен быть выбран в качестве *первичного* ключа, а все остальные потенциальные ключи будут называться *альтернативными*. Если ключ состоит из одного атрибута, то он называется *простым*. Часто простой ключ бывает суррогатным. *Суррогатный ключ* – это дополнительное служебное поле, добавленное к уже имеющимся информационным полям отношения (таблицы), единственное предназначение которого – служить первичным ключом. Значение этого поля не образуется на основе каких-либо других данных из базы данных, а генерируется искусственно.

Отношение, определенное на множестве из n доменов (не обязательно различных), содержит две части: заголовок (схему отношения) и тело (множество из m кортежей). *Схема отношения* – это именованное множество упорядоченных пар <имя_атрибута, имя_домена>. Понятие схемы отношения близко к понятию структурного типа в языках программирования (например, **record** в языке Pascal или **struct** в языке C). Кортеж, соответствующий данной схеме отношения, – это множество упорядоченных пар <имя_атрибута, значение_атрибута>, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. Отношение обладает следующими свойствами.

- В отношении нет одинаковых кортежей.
- Кортежи не упорядочены (сверху вниз).
- Атрибуты не упорядочены (слева направо).
- Каждый кортеж содержит ровно одно значение для каждого атрибута. Отношение, удовлетворяющее этому свойству, называется нормализованным или представленным в первой нормальной форме (1NF).
- Все значения атрибутов *атомарны*, т. е. не обладают структурой.

Схема базы данных в реляционной модели – это множество именованных схем отношений. *Реляционная база данных* – это набор нормализованных отношений, имена которых совпадают с именами схем отношений в схеме базы данных.

В реляционной модели фиксируются два базовых требования *целостности*, которые должны выполняться для любых отношений в любых реляционных базах данных. Это *целостность сущностей* и *ссылочная целостность* (или *целостность внешних ключей*). Поддержание целостности сущностей осуществляется с помощью двух ограничений:

1. при добавлении кортежей в отношение проверяется уникальность их первичных ключей,
2. не допускается изменение значений атрибутов, входящих в первичный ключ.

Ссылочная целостность означает следующее. Отношения в реляционной базе данных связаны между собой. При этом связи между атрибутами внутри отношения описываются в терминах *функциональных*

зависимостей, а для отражения зависимостей между кортежами разных отношений используется дублирование первичного ключа одного отношения (родительского) в другое (дочернее). Атрибуты, представляющие собой копии ключей родительских отношений, называются *внешними ключами*. Внешний ключ в дочернем отношении – это непустое подмножество множества атрибутов этого отношения, такое, что: а) существует родительское отношение (причем эти отношения необязательно различны) с потенциальным ключом; б) каждое значение внешнего ключа в текущем дочернем отношении обязательно совпадает со значением потенциального ключа некоторого кортежа родительского отношения. Требование ссылочной целостности состоит в следующем: для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском отношении должен найтись кортеж с таким же значением первичного ключа.

Конец замечания.

Описание предметной области

Учебная база данных «Поставщики и Продукты», состоит из трех отношений (далее называемых таблицами): Suppliers, Products и Shipments. **В программе требуется таблицы реализовать в виде файлов, а не в виде массивов!!!**

Таблица Suppliers представляет поставщиков. Каждый поставщик имеет
обязательный уникальный номер (SupplierID) – первичный суррогатный ключ;
обязательное уникальное имя (SupplierName) – альтернативный ключ;
необязательный рейтинг или статус (Status);
обязательное место расположения (City).

Предполагается, что каждый поставщик находится только в одном городе.

Таблица Products представляет продукты (точнее, товары или изделия). Каждый продукт имеет
обязательный уникальный номер (ProductID) – первичный суррогатный ключ;
обязательное название продукта (ProductName);
необязательный цвет продукта (Color);
необязательный вес продукта (Weight);
необязательную цену продукта (Price);
обязательное место хранения или производства продукта (City).

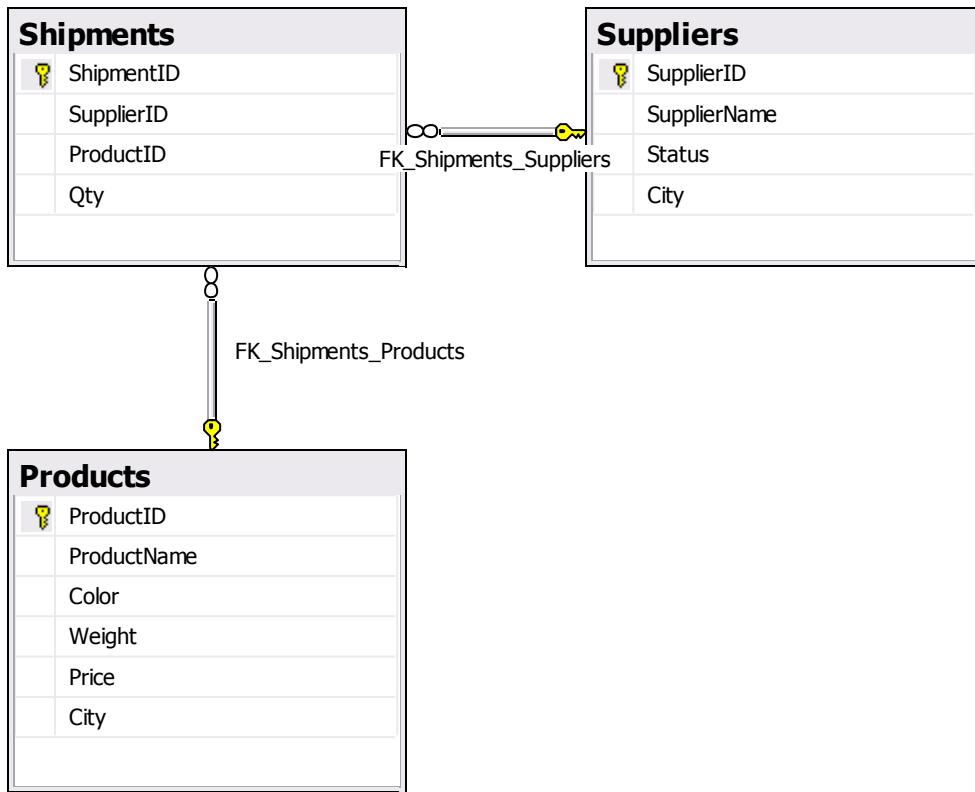
Предполагается (где это имеет значение), что вес продукта и цена продукта указана в условных единицах. Также предполагается, что каждый продукт имеет только один цвет и хранится или производится только в одном городе.

Таблица Shipments представляет поставки. Она в известном смысле служит для организации логической связи двух других таблиц. Например, в приводимом ниже примере заполнения таблиц данными первая строка таблицы Shipments связывает поставщика с номером 502 в таблице Suppliers с соответствующим продуктом, имеющим номер 908 в таблице Products, т.е. представляет факт поставки продуктов типа 908 поставщиком с номером 502 (а также указывает количество продуктов – 30 штук). Таким образом, каждая поставка характеризуется

обязательным уникальным номером (ShipmentID) – первичный суррогатный ключ;
обязательным, но не уникальным номером поставщика (SupplierID);
обязательным, но не уникальным номером продукта (ProductID);
обязательным количеством (Qty).

Предполагается, что в одно и то же время может быть не более одной поставки для одного поставщика и одного продукта, поэтому для каждой поставки комбинация значений столбцов SupplierID и ProductID уникальна с точки зрения набора текущих поставок, представленных в таблице Shipments. Номер поставщика (SupplierID) и номер детали (ProductID) будут внешними ключами.

Диаграмма базы данных



Пример заполнения таблиц базы данных

ТАБЛИЦА Suppliers // Поставщики

SupplierID	SupplierName	Status	City
11	Sharp Bikes	89	Bothell
25	Coalition Bike Company	40	Dallas
28	Commuter Bicycle Store	78	Phoenix
185	Weekend Bike Tours	98	Bellevue
448	Action Bicycle Specialists	97	Toronto

ТАБЛИЦА Products // Продукты

ProductID	ProductName	Color	Weight	Price	City
756	Road-450 Red, 44	Red	7606.70	1457.99	Zeeland
774	Mountain-100 Silver, 48	Silver	9715.90	3399.99	Pontiac
775	Mountain-100 Black, 38	Black	9230.56	3374.99	Novi
776	Mountain-100 Black, 42	Black	9421.06	3374.99	Southfield
785	Mountain-300 Black, 38	Black	11498.51	1079.99	Madison Heights
785	Mountain-300 Black, 38	Black	11498.51	1079.99	Madison Heights

ТАБЛИЦА Shipments // Поставки

ShipmentID	SupplierID	ProductID	Qty
487	502	908	30
488	502	918	10
489	502	743	40
490	502	739	50
491	502	987	20

Семантика таблиц

ТАБЛИЦА Suppliers // Поставщики

SupplierID	идентификатор поставщика
SupplierName	имя (название компании) поставщика
Status	статус (рейтинг) поставщика
City	город, в котором проживает (зарегистрирован) поставщик

ТАБЛИЦА Products // Продукты

ProductID	идентификатор продукта
ProductName	название продукта
Color	цвет продукта
Weight	вес продукта
Price	цена продукта
City	город, в котором производится или хранится продукт

ТАБЛИЦА Shipments // Поставки

ShipmentID	идентификатор поставки
SupplierID	идентификатор поставщика
ProductID	идентификатор продукта
Qty	количество поставляемого продукта в данной поставке

Синтаксис таблиц

ТАБЛИЦА Suppliers // Поставщики

SupplierID	целое число
SupplierName	строка
Status	целое число
City	строка

ТАБЛИЦА Products // Продукты

ProductID	целое число
ProductName	строка
Color	строка
Weight	вещественное число с точностью до второго знака после запятой
Price	вещественное число с точностью до второго знака после запятой
City	строка

ТАБЛИЦА Shipments // Поставки

ShipmentID	целое число
SupplierID	целое число
ProductID	целое число
Qty	целое число

Ограничения таблиц

ТАБЛИЦА Suppliers // Поставщики

SupplierID	NOT NULL PRIMARY KEY
SupplierName	NOT NULL UNIQUE
Status	NULL Status BETWEEN 0 AND 100
City	NOT NULL

ТАБЛИЦА Products // Продукты

ProductID	NOT NULL PRIMARY KEY
ProductName	NOT NULL
Color	NULL
Weight	NULL Weight > 0.00

Price	NULL Price >= 0.00
City	NOT NULL

ТАБЛИЦА Shipments // Поставки

ShipmentID	NOT NULL PRIMARY KEY
SupplierID	NOT NULL FOREIGN KEY(SupplierID) REFERENCES Suppliers (SupplierID)
ProductID	NOT NULL FOREIGN KEY(ProductID) REFERENCES Suppliers (ProductID)
Qty	NOT NULL Qty > 0

Примечания.

1. NULL допускаются отсутствующие значения
2. NOT NULL не допускаются отсутствующие значения
3. PRIMARY KEY первичный ключ (в таблице всегда один)
4. UNIQUE ключ уникальности, но не первичный ключ (таких в таблице может быть несколько)
5. FOREIGN KEY внешний ключ (в дочерней таблице), который ссылается на первичный ключ (в родительской таблице). Внешний ключ не обладает свойством уникальности.

Операции обновления над таблицами базы данных «Поставщики и Продукты» (общие для всех вариантов)

1. Вставить строку в любую таблицу.
2. Удалить строку из любой таблицы по значению первичного ключа.
3. Изменить строку любой таблицы по значению первичного ключа.

Замечания.

1. В случае удаления объекта ссылки внешнего ключа операцию удаления ссылочных строк не выполнять. Прежде надо удалить ссылающиеся строки дочерних таблиц.
2. В случае добавления строки значение первичного ключа генерировать автоматически, например, путем инкремента (+1) к значению последнего первичного ключа. Значения первичных ключей удаленных строк не могут повторяться.
3. Удаление строки выполнять на логическом уровне, т.е. удаленная строка только помечается как удаленная, а физическое удаление выполнять с помощью утилиты сжатия таблицы.

Запросы к базе данных «Поставщики и Продукты» (по вариантам)

ВАРИАНТ 01

1. Для каждого продукта указать номер и общее количество поставщиков, поставляющих этот продукт.
2. Получить имена поставщиков продукта под номером ?.
3. Определить поставщиков, которые ничего не поставляют.
4. Получить имена поставщиков, по крайней мере, одного продукта цвета ?.
5. Найти все товары, поставляемые поставщиками с статусом больше среднего.

ВАРИАНТ 02

1. Вычислить суммарную стоимость всех продуктов, поставляемых поставщиками из города ?.
2. Получить имена поставщиков всех типов продуктов.
3. Получить имена поставщиков, которые не поставляют продукт с номером ?.
4. Получить имена поставщиков, по крайней мере, тех типов продуктов, которые поставляет поставщик под номером ?.
5. Вычислить суммарную стоимость всех поставляемых продуктов, находящихся в городе ?.

ВАРИАНТ 03

1. Получить все пары имен поставщиков, находящихся в одном городе.
2. Определить общее количество поставщиков, участвующих в поставках.
3. Вычислить суммарную стоимость всех продуктов, поставляемых поставщиком под номером ?.
4. Для каждого поставщика указать номер и общий объем поставки в штуках.

5. Определить цвета продуктов, поставляемых поставщиком под номером ?.

ВАРИАНТ 04

1. Найти всех поставщиков, поставляющих продукты, цена которых отличается не более чем на 10% от текущей средней цены продуктов.
2. Выбрать информацию обо всех поставках, таких, что поставщик и поставляемый им продукт находятся в одном городе.
3. Указать номера продуктов, поставляемых более чем одним поставщиком.
4. Указать имена поставщиков, статус которых меньше текущего среднего статуса в таблице поставщиков.
5. Определить в поставках максимальное и минимальное количество продуктов с номером ?.

ВАРИАНТ 05

1. Определить общее количество продуктов с номером ?, поставляемых поставщиком под номером ?.
2. Определить имена поставщиков из города ? со статусом больше ?.
3. Найти всех поставщиков, поставляющих очень дорогие продукты. Цена таких продуктов отличается от цены самого дорогого продукта не более чем в 4 раза.
4. Определить номера и вес всех типов продуктов, вес которых превышает ?.
5. Для каждого продукта выбрать номер и общий объем поставки в штуках.

ВАРИАНТ 06

1. Указать названия продуктов, цена которых больше текущей средней цены в таблице продуктов.
2. Определить общее количество поставляемых продуктов.
3. Найти всех поставщиков, поставляющих продукты, находящиеся в ценовом диапазоне от ? до ? включительно.
4. Определить продукты, которые никем не поставляются.
5. Найти все пары названий городов, таких, что поставщик из первого города поставляет продукт, находящийся в другом городе.

Замечание. Символ ? обозначает мишень для подстановки параметра. Таким образом можно реализовать параметрический запрос.

Задание на лабораторную работу

1. Написать программу, которая моделирует создание и использование реляционной базы данных «Поставщики и Продукты» средствами языка C и библиотеки CRT.
2. Исходные данные для массового копирования данных в таблицы базы данных брать из папок «Входные данные в формате XML» или «Входные данные в формате TXT» по выбору.
3. В работе использовать функций обработки файлов последовательного и произвольного доступа, определенные в <stdio.h>.

/* Прототипы рекомендованных к использованию функций */

```
FILE * fopen(const char *, const char *);
FILE * freopen(const char *, const char *, FILE *);
int    fclose(FILE *);
size_t fread(void *, size_t, size_t, FILE *);
size_t fwrite(const void *, size_t, size_t, FILE *);
int    fgetpos(FILE *, fpos_t *);
int    fsetpos(FILE *, const fpos_t *);
int    fseek(FILE *, long, int);
long   ftell(FILE *);
void   rewind(FILE *);
```

4. Допускается реализация ввода/вывода бинарных файлов и использование связанных с ними функций неформатируемого ввода/вывода, определенных в <io.h>.

5. Для любознательных в файле СЦЕНАРИИ.pdf представлен сценарий создания базы данных «Поставщики и Продукты» на языке Transact-SQL. Там же представлены запросы для всех вариантов, обернутые в хранимые процедуры. В файле SQLQuery1.sql тексты хранимых процедур повторены еще раз.
6. Для очень любознательных в папке Решения собраны файлы с ответами для всех запросов.

Рекомендуемая литература

1. Шилдт, Герберт. Полный справочник по С, 4-е издание. – М.: Издательский дом «Вильямс», 2004. (Глава 9. Файловый ввод/вывод. Стр. 215-240)
2. Харви Дейтел, Пол Дейтел. Как программировать на С. 3-е изд. – М.: Бином-Пресс, 2002. (Глава 11. Работа с файлами в С. Стр. 425-456.)