**Реляционная база данных** – это совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Однако пользователи могут воспринимать такую базу данных как совокупность таблиц.

1. Каждая таблица состоит из однотипных строк и имеет уникальное имя.

2. Строки имеют фиксированное число полей (столбцов) и значений (множественные поля и повторяющиеся группы недопустимы). Иначе говоря, в каждой позиции таблицы на пересечении строки и столбца всегда имеется в точности одно значение или ничего.

3. Строки таблицы обязательно отличаются друг от друга хотя бы единственным значением, что позволяет однозначно идентифицировать любую строку такой таблицы.

4. Столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (даты, фамилии, целые числа или денежные суммы).

5. Полное информационное содержание базы данных представляется в виде явных значений данных и такой метод представления является единственным. В частности, не существует каких-либо специальных "связей" или указателей, соединяющих одну таблицу с другой. Так, связи между строкой с БЛ = 2 таблицы "Блюда" на рис. 3.2 и строкой с ПР = 7 таблицы продукты (для приготовления Харчо нужен Рис), представляется не с помощью указателей, а благодаря существованию в таблице "Состав" строки, в которой номер блюда равен 2, а номер продукта – 7.

6. При выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке безотносительно к их информационному содержанию. Этому способствует наличие имен таблиц и их столбцов, а также возможность выделения любой их строки или любого набора строк с указанными признаками (например, рейсов с пунктом назначения "Париж" и временем прибытия до 12 часов).

**SQL**

Все языки манипулирования данными (ЯМД), созданные до появления реляционных баз данных и разработанные для многих систем управления базами данных (СУБД) персональных компьютеров, были ориентированы на операции с данными, представленными в виде логических записей файлов. Это требовало от пользователей детального знания организации хранения данных и достаточных усилий для указания не только того, какие данные нужны, но и того, где они размещены и как шаг за шагом получить их.

Рассматриваемый же ниже непроцедурный язык SQL (Structured Query Language - структуризованный язык запросов) ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц. Особенность предложений этого языка состоит в том, что они ориентированы в большей степени на конечный результат обработки данных, чем на процедуру этой обработки. SQL сам определяет, где находятся данные, какие индексы и даже наиболее эффективные последовательности операций следует использовать для их получения: не надо указывать эти детали в запросе к базе данных.

Реализация в SQL концепции операций, ориентированных на табличное представление данных, позволило создать компактный язык с небольшим (менее 30) набором предложений. SQL может использоваться как интерактивный (для выполнения запросов) и как встроенный (для построения прикладных программ). В нем существуют:

* предложения определения данных (определение баз данных, а также определение и уничтожение таблиц и индексов);
* запросы на выбор данных (предложение SELECT);
* предложения модификации данных (добавление, удаление и изменение данных);
* предложения управления данными (предоставление и отмена привилегий на доступ к данным, управление транзакциями и другие). Кроме того, он предоставляет возможность выполнять в этих предложениях:
  + арифметические вычисления (включая разнообразные функциональные преобразования), обработку текстовых строк и выполнение операций сравнения значений арифметических выражений и текстов;
  + упорядочение строк и (или) столбцов при выводе содержимого таблиц на печать или экран дисплея;
  + создание представлений (виртуальных таблиц), позволяющих пользователям иметь свой взгляд на данные без увеличения их объема в базе данных;
  + запоминание выводимого по запросу содержимого таблицы, нескольких таблиц или представления в другой таблице (реляционная операция присваивания).
* агрегатирование данных: группирование данных и применение к этим группам таких операций, как среднее, сумма, максимум, минимум, число элементов и т.п.

В SQL используются следующие основные типы данных, форматы которых могут несколько различаться для разных СУБД:

* **INTEGER** - целое число (обычно до 10 значащих цифр и знак);
* **SMALLINT** - "короткое целое" (обычно до 5 значащих цифр и знак);
* **DECIMAL(p,q)** - десятичное число, имеющее p цифр (0 < p < 16) и знак; с помощью q задается число цифр справа от десятичной точки (q < p, если q = 0, оно может быть опущено);
* **FLOAT** - вещественное число с 15 значащими цифрами и целочисленным порядком, определяемым типом СУБД;
* **CHAR(n)** - символьная строка фиксированной длины из n символов (0 < n < 256);
* **VARCHAR(n)** - символьная строка переменной длины, не превышающей n символов (n > 0 и разное в разных СУБД, но не меньше 4096);
* **DATE** - дата в формате, определяемом специальной командой (по умолчанию mm/dd/yy); поля даты могут содержать только реальные даты, начинающиеся за несколько тысячелетий до н.э. и ограниченные пятым-десятым тысячелетием н.э.;
* **TIME** - время в формате, определяемом специальной командой, (по умолчанию hh.mm.ss);
* **DATETIME** - комбинация даты и времени;
* **MONEY** - деньги в формате, определяющем символ денежной единицы ($, руб, ...) и его расположение (суффикс или префикс), точность дробной части и условие для показа денежного значения.

До сих пор понятие "таблица", как правило, связывалось с реальной или базовой таблицей, т.е. c таблицей, для каждой строки которой в действительности имеется некоторый двойник, хранящийся в физической памяти машины (рис.1.2). Однако SQL использует и создает ряд виртуальных (как будто существующих) таблиц: представлений, курсоров и неименованных рабочих таблиц, в которых формируются результаты запросов на получение данных из базовых таблиц и, возможно, представлений. Это таблицы, которые не существуют в базе данных, но как бы существуют с точки зрения пользователя. Базовые таблицы создаются с помощью предложения CREATE TABLE (создать таблицу).

