

Перед тем как дать определения СУБД, необходимо определиться с понятием базы данных.

База данных

Базой данных называется совокупность данных и метаданных. Метаданные представляют из себя правила о том, как должны соотноситься между собой разные элементы БД.

Если для текущего состояния БД все правила, описанные в метаданных, оказываются выполненными, то БД находится в состоянии **целостности**.

Пример Базы данных

Пусть наша база данных состоит из двух таблиц. Первая таблица хранит данные по каждому сотруднику. Заголовок данной таблице представлен ниже:

СЛУ_НОМЕР	СЛУ_ОТД_НОМЕР
-----------	---------------

Вторая таблица хранит информацию по отделу, заголовок которой представлены ниже:

ОТД_НОМЕР	ОТД_КОЛ_СЛУ
-----------	-------------

В каждой записи служащего столбец СЛУ_ОТД_НОМЕР - номер отдела из таблицы отделов, а в каждой записи отдела столбец ОТД_КОЛ_СЛУ - количество сотрудников в данном отделе.

В данном примере мы имеем базу данных, в которой данными являются конкретные записи в таблицах служащие и отделы, а метаданные - правила, описанные ниже:

1. Для каждой записи из таблицы отделов ОТД_КОЛ_СЛУ должен равняться сумме всех служащих, у которых СЛУ_ОТД_НОМЕР равен ОТД_НОМЕР текущей записи.
2. Для каждой записи из таблицы служащих должна существовать единственная запись таблицы отделов, у которой ОТД_НОМЕР равен СЛУ_ОТД_НОМЕР текущей записи.

СУБД

СУБД - это комплекс программных средств, посредством которых можно создавать БД и менять её состояние (добавлять, изменять и удалять записи)[1]. Также важной особенностью СУБД являются механизмы поддержки целостности (проверка первичного и вторичного ключа, транзакции, восстановление после сбоев и т.д.). Примеры СУБД: SQLite, DB2, MS SQL.

Функции СУБД

Управление данными во внешней памяти

СУБД должна поддерживать определённую структуры во внешней памяти, которая будет хранить БД и набор служебной информации (индексы, журналы и т.д.).

Существует два подхода к реализации данной функции:

1. С активным использовать возможности файловой системы.
2. Использовать собственную систему именования объектов (применяется намного чаще).

Стоит также отметить, что не все СУБД хранят данные во внешней памяти, однако данный подход является самым популярным в наше время.

Управления буферами оперативной памяти

Почти всегда СУБД поддерживает собственную систему буферизации данных и редко полагается на операционную систему. Так как СУБД известно намного больше о характере используемых данных, чем ОС, то СУБД может обеспечить более эффективную буферизацию этих данных.

Управление транзакциями

Транзакция позволяет объединить несколько операций над БД в один “атомарный” блок, который либо полностью выполняется, либо полностью отменяется. Управление транзакциями необходимо для обеспечения целостности БД, так как выполнения отдельных операций в транзакции может временно нарушать целостность данных [2].

Журнализация

СУБД должна иметь возможность восстанавливать целостность данных после сбоя. Основным способом достижения такой возможности является журнализация - ведение записей всех изменений над БД. Основным способом ведения журнала является протокол WAL (Write Ahead Log) - операция сначала записывается в журнал, а только потом выполняется.

Объекты СУБД

Объекты СУБД могут отличаться друг от друга в зависимости от того, какая модель данных используется.

Модель данных

Модель данных (МД) - это набор основным признаков и свойств, которыми должны обладать все БД, реализующие данную модель.

За историю развития баз данных было разработано множество различных МД. Ниже приведены некоторые из них:

Модель инвертированных таблиц

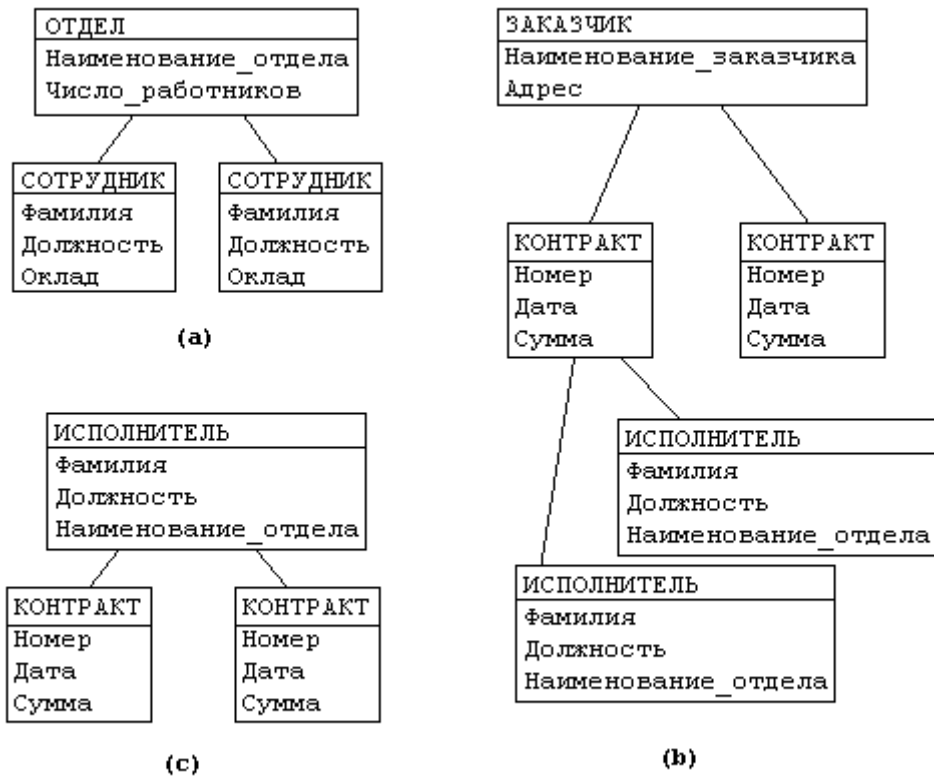
БД данной модели состоит из набора **упорядоченных** таблиц. Перед тем как изменить базу данных, необходимо выполнить операцию позиционирования - переместиться на необходимую строку в таблице.

Индекс «Клиент»		
№	Клиент	Ссылки
1	Иванов	1, 4
2	Белов	2
3	Петров	3

Иерархическая модель данных

БД данной модели представляет собой набор переменных типа дерева. Каждый тип дерева состоит из корневого типа записи и набора типа поддеревьев, каждое из которых

является некоторым типом дерева. Перед изменением БД также необходимо выполнить операцию позиционирования (переместиться от родителя к конкретному потомку, переместиться от потомка к родителю и т.д.).



Реляционная модель данных

Реляционная модель данных является самой распространённой на данный момент. БД реляционной модели данных состоит из набора **неупорядоченных** таблиц (без порядка строк и столбцов), которую принято называть отношением [3].

Пример таблицы реляционной БД

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i>Год рождения</i>	<i>Отдел</i>
023	Волкова Елена Павловна	секретарь	26000	1985	2
113	Белов Сергей Юрьевич	инженер	39800	1980	1
101	Рогов Сергей Михайлович	директор	62000	1972	2
056	Панина Анна Алексеевна	инженер-программист	41800	1978	1
...
098	Фролов Юрий Вадимович	начальник отдела	49200	1971	9

Мощность отношения. Арность отношения.

Такой подход имеет массу преимуществ в сравнение с другими моделями данных:

1. Отсутствие порядка в таблицах приводит к отсутствию операции позиционирования. Это позволяет выполнять запросы в декларативном виде, что значительно упрощает использование СУБД.
2. Реляционный подход основан на использование фундаментальных математических конструкций - кортежей и отношений. Такое математическое определение упрощает процесс формальных рассуждений и помогает в решение задач оптимизации запросов.

Источники

1. DBMS Introduction: <https://www.w3schools.in/dbms/intro>
2. DBMS Transaction: <https://www.w3schools.in/dbms/transaction>
3. Relational Data Model: <https://www.w3schools.in/dbms/relation-data-model>