

Оглавление

1	Лекция первая. Введение в БД.	2
1.1	Базы данных.	2
1.2	Основные требования к БД.	2
1.3	СУБД, журнализация.	3
1.4	Основные компоненты СУБД	3
1.5	Классификация СУБД	4
2	Лабораторная работа 1.	5
2.1	Задание:	5
3	Семинар 1.	6
3.1	SQL	6
3.2	Основные языки	7
3.3	Способы хранения данных	8
4	Реляционная модель	10
4.1	ER - модель	10
4.2	Реляционная модель	11

1

Лекция первая. Введение в БД.

1.1 Базы данных.

БД - это самодокументированная собрание интегрированных записей. Набор таблиц.

Самодокументированная - хранятся метаданные, т.е. данные о данных.

Интегрированные записи - Файлы данных. Целый комплекс. Имеются индексы. Метаданные

1.2 Основные требования к БД.

- Не избыточность - не храним лишнюю информацию.
- Эффективность доступа - малое время отклика на действие пользователя.
- Совместное использование.
- Безопасность. Также внутренняя безопасность - защита от дурака (пример: вместо числа ввел букву).

- Восстановление после сбоя.
- Целостность - если ссылаемся на какой-то объект, то он должен быть. Не ссылаться на несуществующие объекты.
- Независимость от сторонних приложений. Если программа отправляет ерунду БД должна обработать.

1.3 СУБД, журнализация.

СУБД - (Средства управления БД) приложение, обеспечивающее создание, хранение, обновление и поиск информации в БД. Программа.

СУБД управляет БД.

Система БД - совокупность БД.

Транзакция - набор действий, которые выполняются одновременно. (Пример: онлайн перевод, одновременно в одном месте деньги ушли, в другом появились.)

Журнализация - информация о действиях, которые происходили в системе. Помогает в откате каких-то действий. **БД** сохраняет запросы в журнале.

СУБД должна поддерживать языки.

1.4 Основные компоненты СУБД

- Ядро - управление памятью. Журнализация.
- Процессор языка БД - оптимизация. Выполнение.
- Подсистема поддержки времени исполнения.
- Сервисные программы - те утилиты, которые мы пишем, доп. возможность. (Вывод звездочек вокруг имени.)

1.5 Классификация СУБД

- По модели данных
 - Дореляционная.
 - * Инвертированный список (рис 1)
 - * Иерархия. (Дерево)
 - * Сетевые (граф)
 - Реляционная.
 - Постреляционная
- По архитектуре.
 - Локальные - на одном устройстве.
 - Распространенные - на многих устройствах.
- По способу доступа к БД
 - Файл-серверный подход - Подключились, взяли всё. Нагружаем клиента, а не сервер. **Минусы: У каждого клиента своя копия.**
 - клиент-серверные - запросы выполняются на сервере, клиент получает только нужное
 - Встраиваемые - маленькие базы, которые не нужны всем.

2

Лабораторная работа 1.

2.1 Задание:

- Выбрать тему.
- Рисуем ER-модель нашей базы.
- Создать БД. Создать таблицу. (≥ 3 объектов (таблицы связи не считаются объектами)). Создать ключи. (Все это в SQL-скрипт)
- Наполнить (csv) ≥ 1000 строк.

По итогу 2 фала: 1 SQL-скрипт и 1 модель.

3

Семинар 1.

3.1 SQL

SQL - SQL (Structured Query Language – язык структурированных запросов)

декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных.

SQL - работает в любой БД. В основах лежит реляционная модель.

В основе реляционной модели лежит теория множеств и логика предикатов.

T-SQL - нек-ое дополнение. Настройка.

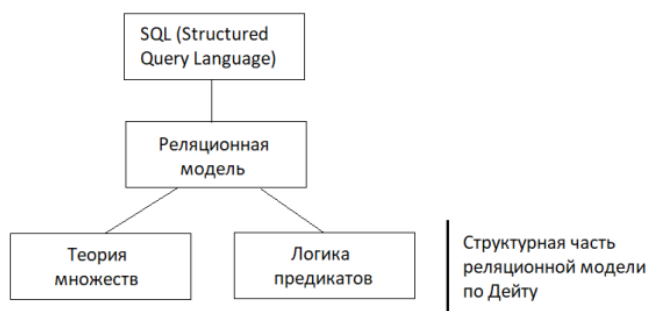


Рис. 3.1: SQL

Заголовок – набор атрибутов (В SQL - столбцы), каждый из которых имеет определенный тип.

Атрибут – совокупность имени и типа данных (Атрибут==столбец). Атрибут – название столбца, его тип + дополнительные настройки

Тело – множество картежей (В SQL – строки).

Заголовок кортежа – заголовок отношения.

Заголовок отношения				
Атрибут				
1	2	3	4	5
aaa	aba	abb	bba	bab
bbb	bcb	bcc	ccb	cbc
ccc	cdc	cdd	ddc	dcd
ddd	ded	dee	eed	ede

Кортеж

Тело
отношения

Рис. 3.2: Пример таблицы.

3.2 Основные языки

Логику работы с данными можно разделить на три основных языка:

- DDL (*Data Definition Language*) - (Создаем объекты для хранения данных). Служит для описания структуры БД:
 - Создать (Create)
 - Удалить (Drop)
 - Изменить (Alter)
- DML (*Data Manipulation Language*) - Язык для работы с данными
 - Обновить (update)
 - Загружать (insert)
 - Удалять (delete/truncate)

- Читать (select)
- DCL (*Data Control Language*) - Служит для управления доступа к объектам.
 - Выдача прав доступа к объекту (grant)
 - Удаление прав доступа на объект (revoke)

Обращение к таблице. Схема обращения к таблице: [название БД].[название схемы].Название таблицы. Рис. 3.3

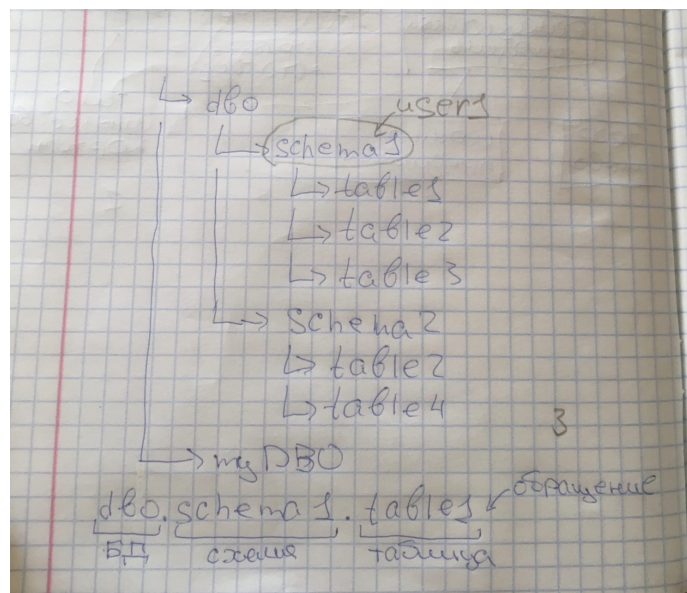


Рис. 3.3: Структура БД.

3.3 Способы хранения данных

- Таблица (table).
- Временные таблица (temp table). По завершению сессии таблица удаляется.
- Представление (View)

- Производные таблицы. (Временная)
- Индексированное представление.

Пример создания таблицы

```
CREATE TABLE dbo.EmployeePhoto
(
  Id int IDENTITY(1, 1),
  EmployeeId int NOT NULL PRIMARY KEY,
  Photo varbinary(max) FILESTREAM NULL,
  MyRowGuidColumn uniqueidentifier NOT NULL ROWGUIDCOL
  UNIQUE DEFAULT NEWID()
);
```

id - АТРИБУТ типа int счетчик шагаем начиная с единицы с шагом 1.

EmployeeId - поле, которое используем в кач-ве идентификатора.

Photo - По умолчанию NULL - пустой.

MyRowGuidColumn - Уникальное, DEFAULT - по умолчанию поле задается newID.

nvarchar - Выделяет столько памяти, какова длина строки

varchar - Физически занята вся строка. Занято пробелами.

Salary - numeric(15, 2) - Сколько всего цифр выделено в нашем числе, сколько знаков после запятой.

4

Реляционная модель

4.1 ER - модель

- Сущность
- Связь

Объекты обозначаются прямоугольниками. Внутри пишем название.

Виды сущностей:

- Сильные - Обозначаются просто в рамке.
- Слабые - не могут существовать друг без друга. Факультет и предметы. Обозначается вложенным квадратом (рамочка).

Атрибуты отображаются овалами. Внутри пишем название атрибута.

Виды связей:

- Один к одному. Студент-зачетка.
- Один ко многим. Статья-рецензия. Добавляем внешний ключ со стороны многих. Из многих в сторону одного.
- Многие ко многим. Студент-преподаватель. Добавляем связочную таблицу.

4.2 Реляционная модель

Реляционная модель

- Структурная часть - отвечает за то, какие объекты есть.
- Целостная - отвечает за ссылки. DDL.
 - Ссылочная целостность (FK)
 - Целостность сущности (PK) - говорит о том, что есть первичный ключ. Нет повторения. Всегда знаем на что ссылаемся.
- Манипуляционная - за механизм работы с данными. DML.

Домен = (примерно равно) тип данных.

Атрибут (отношения) = (примерно равно) столбец. Упорядоченная пара вида:

имя-атрибута, имя-домена

Схема отношений = (примерно равно) Заголовок. имя-отношение, имя-домена

Кортеж = (примерно равно) Строка. Имя-атрибута, значение-атрибута

Отношение = (примерно равно) таблица.

Непустое подмножество множества атрибутов схемы 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)
- неизбыточности (никакое из собственных подмножеств множества потенциального ключа не обладает свойством уникальности)

Внешний ключ в отношении R2 – это непустое подмножество множества атрибутов FK этого отношения, такое, что:

- Существует отношение $R1$ (причем отношения $R1$ и $R2$ обязательно различны) с потенциальным ключом $СК$;
- Каждое значение внешнего ключа FK в текущем значении отношения $R2$ обязательно совпадает со значением ключа $СК$ некоторого кортежа в текущем значении отношения $R1$.