# Семинар 1

Реляционные базы данных

## На чём писать? 3 Путя:

1. mssql + c# (стандартный путь)
2. postgreSQL + python (C++) (современный путь)
3. Oracle + Java (путь мазохиста)

Установить можно в “Докер”

На MySql лучше не писать

## Базы данных

База Данных – это само документируемое собрание интегрированных записей.

Самодокументируемое – журналирование (записывает данные о том что делают пользователи или какие процессы происходят)

**Мета**данные (данные о данных)

Интегрированные – (собранные вместе) файлы данных + мета + индексы

Требования:

1. Не избыточность (Не дублируем данные)
2. Целостность

|  |  |  |
| --- | --- | --- |
| **Id** | FIO | Age |
| 1 | И.И.И. | 18 |
| 2 | П.П.П | 19 |

Id – Primary Key внутренний ключ (простой/составной)

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Sub | **Student** | date |
| 1 | БД | 1 | 01.09.2021 |
| 2 | БД | 1 | 02.09.2021 |
| 3 | БД | 2 | 02.09.2021 |
| 4 | БД | *3 (ошибка)* | 02.09.2021 |

Student - Foreign key (Внешний ключ)

1. Безопасность (Защита от утечки данных) (и Защита от дурака)
2. Восстановление после сбоя (Отказоустойчивость)
3. Совместный доступ
4. Эффективность
5. Независимость (База данных не должна зависеть от сторонних приложений)

## Система управления базой данных

**СУБД** – система управления базой данных.

Чем занимается СУБД:

1. Работа с внешней памятью
2. Работа с внутренней памятью (оперативной памятью)
3. Журнализация
4. Управление транзакциями (выполнить всё или ничего)
5. Поддержка языка SQL
6. Служебные программы (загружаемые модули)

Первые 4 пункта это ядро СУБД

5ый пункт – языковой пакет

6ой пункт – блок служебных программ

## SQL

SQL (structed query language – структурированный язык запросов)

SQLi2011

ANSI

Mssql – Е-SQL; Oracle - PLSQL; postgres - PLpgSQL;

Сам SQL базируется на реляционной модели, которая основана на:

* Теория множеств
* Логика предикатов

SQL состоит из 3 языков:

* DML (data manipulation language) (работаем с таблицами)
  1. Добавить (insert)
  2. Удалить (delete)
  3. Обновить (update)
  4. Выбрать (select)
  5. \*удалить всё (truncate)
* DDL (data definition language) (работаем с объектами БД)
  1. Создать (create)
  2. Удалить (drop)
  3. Изменить (alter)
* DCL (data control language) (права на данные)
  1. Выдать права (grant)
  2. Забрать права (revoke)

## Как хранить данные?

1. Таблица (table)
2. Временная таблица (temp table, #table) – существуют одну сессию
3. Табличные переменные – нельзя изменить (создаётся с помощью declare)
   * Локальные (@)
   * Глобальные (@@)
4. Представление (view)

|  |  |  |
| --- | --- | --- |
| Id | FIO | Salary |
| 1 | И.И.И. | 100 000,00 р |

|  |  |
| --- | --- |
| Id | FIO1 |
| 1 | И.И.И. |

Любой insert, delete, update в двух таблицах – не приятно - синхронизация

1. Бухгалтерия
2. Охрана (не должна иметь доступ к Salary (зарплате))

Create view employer as

Select FIO from employer

1. Индексированное представление

## Объекты БД

1. Сама база данных
2. Таблица
3. Представление (view)
4. Ограничения
5. Схемы

## Создание таблицы

create table Empl (

id int [not null],

FIO text/varchar(100)/nvarchar(100),

Salary numeric(15,2)

);

nvarchar – mssql

varchar(10) vs nvarchar(10)

varchar – всегда выделит 10

nvarchar – выделит только столько символов сколько заполнено

numeric(15,2) => 13 цифр до запятой, 2 после запятой

## Ограничения

1. [not] null
2. Ключи (PK, FK)
3. Проверка (check)
4. Уникальность
5. По умолчанию

# Синтаксис

Create table [db.[schema.]]table\_name (

{<column\_defintion> |

<computed\_column\_defention> |

[<table\_constraint>]

[<table\_index>] [,…n]

}

)

[ON {partition}]

[with (<table\_option>)]

## Имя таблицы

db.[schema.]]table\_name – имя таблицы

* БД1
  + Schema1 (user1)
    - *Table1*
    - Table2
  + Schema2 (user2) grant … table1 (для oracle)
    - Table1
    - Table3
* БД2
  + …

Postgres – schema это public

Mssql – schema это master

*БД1.Schema1.Table1*

## Имя столбца

<column\_definition>::=

column\_name

<data\_type>

[NOT NULL|NULL]|

IDentity(seed, increment)

[<column\_constraint>[… n]]

## Data\_type

<data\_type>::= typename

[(precision [,scale]|

max)]

## Column\_constraint (столбцовое ограничение)

<column\_cinstraint>::= [CONSTRAINT constraint\_name]

{ {PRIMARY KEY | UNIQUE} |

[FOREIGN KEY] REFRENCES [Schema\_name.]ref\_table\_name([ref\_column]) |

CHECK (logical\_expression) |

DEFAULT <value>

}

## Column\_computed\_definition

<Column\_computed\_definition>::= column\_name AS <computed\_expression>

## Пример создания

Create table dbo.{employerPhoto(

Id int IDENTITY(1,1),

{eomplyerId int not null primary key,

photo varbinary(max) null,

MyRowGuidColumn int not null unique default newID()

);

Создаём таблицу в схеме dbo с атрибутами:

1. Id (IDENTITY(1,1) 1->2->3->4->5)

Id (IDENTITY(4,10) 4->14->24->24->34)

1. employerId целочисленное, не может не иметь значение, явл. внутренним (primary) ключём
2. photo бинарный тип (нули и единицы 10110), может принять любое значение до максимального, может быть null (можно не указывать)
3. MyRowGuidColumn целочисленное, не может = null, уникальные значения, по умолчанию принимает значение newID()

## Primary Key бывают

1. По кол-ву атрибутов
   1. Составные
   2. Простые
2. По природе
   1. Натуральные (создан из тех данных что мы получили, “природой”)
   2. Суррогатные (создали сами, своими ручками)

## Удаление

drop table <table\_name> - таблицы

drop constraint <constraint\_name> - ограничения

drop database myBase – базы данных

Удаляем сначала ссылки на нашу таблицу, а потом нашу таблицу (alter->…)

Еще можно удалить объект и его зависимости:

drop table <table\_name> [cascade];

## Изменение (alter)

1. alter table table\_name ADD (column\_name1 column\_type1,

column\_name2 column\_type2)

1. alter table table\_name drop column column\_name

alter table table\_name drop constraint constraint\_name

1. alter table table\_name alter column column\_name new\_data\_type

alter table table\_name modify(for Oracle) column\_name new\_data\_type

1. alter table table\_name rename column old\_name to new\_name
2. alter table table\_name ADD constraint constraint\_name PRIMARY KEY(attribute1, attribute2, attribute3)
3. alter table table\_name rename to new\_table\_name

## Переходим к языку DML

Insert, update, delete, select:

1. insert

insert into table\_name values(…) -> 5 значений, иначе ошибка

*insert into table\_name(attr2, attr1) values(1; 2)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attr1a | Attr2 | Attr3 | Attr4 | Attr5 |
| *2* | *1* | *null* | *null* | *null* |
| 2 | 4 | 5 | 6 | 1 |

1. update

update table\_name

set atr5=0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attr1a | Attr2 | Attr3 | Attr4 | Attr5 |
| 2 | 1 | null | null | 0 |
| 2 | 4 | 5 | 6 | 0 |

update table\_name

set atr5=0

where attr5 = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attr1 | Attr2 | Attr3 | Attr4 | Attr5 |
| 2 | 1 | null | null | null |
| 2 | 4 | 5 | 6 | 0 |

1. delete

delete from table\_name

where attr1=2

1. truncate table table\_name ⬄delete from table\_name