Базы данных
Язык SQL:
Разделы DDL, DML и DQL

Виноградова М.В.

МГТУ им. Н.Э. Баумана (ИУ5)

SQL – Structured Query Lanquage

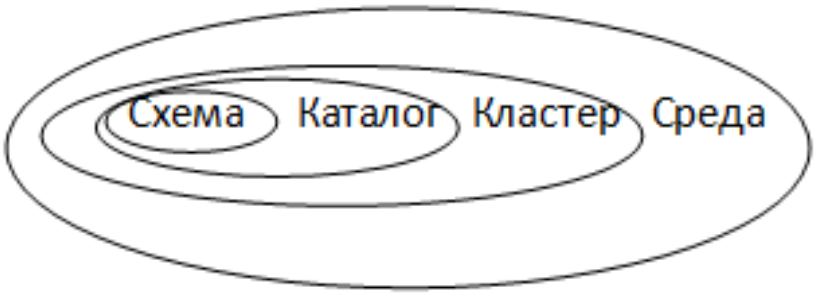
- 1986 – ANSI – SQL/86 (ISO в 1987)
- 1989 - SQL/86 – запросы и схемы
- 1992 - SQL/92 – схемы, транзакции, соединения, авторизация
- 1995 - SQL/CLI – динамический, ODBC
- 1996 - SQL/PSM – хранимые процедуры
- 1999 – SQL:1999 (SQL3) – объектное расширение, UDT
- 2003 – SQL:2003 – OLAP, XML
- 2006 – XQuery

Основные разделы SQL

- Создание/изменение схемы БД - **DDL**
- Ограничения целостности и триггеры
- Представления БД
- Структуры физического уровня
- Управление доступом
- Транзакции
- Запросы к данным (CRUD) - **DML**

Среда СУБД

- **Среда** работает под управлением **СУБД**.
- **Кластер** - аналог БД.
- **Каталог** – набор схем.
- **Схема** – набор таблиц, представлений, триггеров и т.д. (прав, методов сопоставления, доменов, наборов символов)



Основные объекты БД

- Таблицы (table)
- Представления (view)
- Триггеры (trigger)
- Ограничения целостности
- Хранимые процедуры и функции (procedure, function)
- Правила сравнения (collation)
- Домены (domain)
- Последовательности (sequance)

Базовые SQL команды DDL

- Создание структуры

CREATE тип_об назв_об параметры

- Изменение структуры

ALTER тип_об назв_об параметры

- Удаление структуры

DROP тип_об назв_об

Пример создания БД

CREATE DATABASE SalesTest;

```
CREATE DATABASE Sales
ON
( NAME = Sales_dat,
    FILENAME = 'C:\Program Files\DATA\saledat.mdf',
    SIZE = 10,
    MAXSIZE = 50,
    FILEGROWTH = 5 )
LOG ON
( NAME = Sales_log,
    FILENAME = 'C:\Program Files\DATA\salelog.ldf',
    SIZE = 5MB,
    MAXSIZE = 25MB,
    FILEGROWTH = 5MB ) ;
```

Создание схемы БД

- Обращение

  БД.каталог.схема.элемент

  myBD.public.dbo.Person

CREATE SCHEMA имя;

Create table …

Create view …

SET SCHEMA имя; - делает схему текущей.

set catalog имя; - делает каталог текущим

Создание таблицы

```
CREATE TABLE MovieStar
(
    name CHAR(30),
    address VARCHAR(255),
    gender CHAR(1) DEFAULT '?',
    birthDate DATE DEFAULT
   DATE '0000-00-00'
);
```

Изменение структуры таблицы

- Удалить таблицу

DROP TABLE имя;

- Добавить/удалить столбец

ALTER TABLE MovieStar ADD phone CHAR(16);

ALTER TABLE MovieStar ADD phone CHAR(16)
DEFAULT ‘123-45-67’;

ALTER TABLE MovieStar DROP phone;

Создание/удаление индексов

CREATE INDEX Назв. ON Табл(поля);

CREATE INDEX YearIndex1 ON Movie(year);

CREATE UNIQUE INDEX YearIndex2 ON Movie(year);

CREATE INDEX KeyIndex3 ON Movie(title, year);

DROP INDEX YearIndex1;

Ключи таблицы

```
CREATE TABLE Studio
(
    name CHAR(30) PRIMARY KEY,
    address VARCHAR(255),
    pressC# INT REFERENCES MovieExec(cert#)
        ON DELETE SET NULL
        ON UPDATE CASCADE,
   PRIMARY KEY (name),
   FOREIGN KEY (pressC#) REFERENCES
                MovieExec(cert#)
);
```

Добавление ограничений

- ALTER TABLE имя ADD | DROP CONSTRAINT….

- [CONSTRAINT имя] CHECK()

- NOT NULL
- UNIQUE
- FOREIGN KEY
- PRIMARY KEY
- CHECK()

Добавление и удаление ограничений

ALTER TABLE **Person** ADD CONSTRAINT **myc1** CHECK (age between 1 and 120);

ALTER TABLE **Person** ADD CONSTRAINT **myc2** UNIQUE (**telefone**);

ALTER TABLE **Person** ADD CONSTRAINT **myfk1** FOREIGN KEY (**work**) REFERENCES **Org(id_org)** ON DELETE SET NULL

ON UPDATE CASCADE;

ALTER TABLE **Person** ADD CONSTRAINT **mypk** PRIMARY KEY (**idPerson**);

ALTER TABLE **Person** DROP CONSTRAINT **mypk;**

ALTER TABLE **Person** DROP CONSTRAINT **myfk1;**

ALTER TABLE **Person** DROP CONSTRAINT **myc1;**

Отложенные ограничения

- Ограничение NOT DEFERRABLE | DEFERRABLE *
- **NOT DEFFERABLE** – не откладывается
- **DEFFERABLE** – может быть отложено до конца текущей транзакции

- *: INITIALLY DEFFERED (1) | INITIALLY IMMEDIATE (2)
- (1) - Откладывает проверку ограничения до окончания транзакции
- (2) – проверка до изменений с возможной остановкой проверки

SET CONSTRAINT имя DEFFERED | IMMEDIATE

Домены

- Определение домена –тип с ограничениями

  CREATE DOMAIN назв **AS** тип

  default значение CHECK()

- В определении таблицы атрибуту указывают тип домена вместо стандартного типа.

Создание представлений

CREATE VIEW R AS Q;

R = Назв(список атрибутов)

Q = SELECT…..

CREATE VIEW Customers AS

SELECT fio, post, email FROM Persons
WHERE work is NOT NULL;

SELECT * from Customers WHERE post = ‘dir’;

DROP VIEW имя;

Базовые SQL команды DML

- **Вставка**

INSERT INTO R(A1, ..., An) VALUES (V1, ..., Vn);

INSERT INTO R(A1) SELECT A2 FROM R2;

- **Удаление**

DELETE FROM R WHERE C;

- **Обновление**

UPDATE R SET A1 = V1, A2 = V2, ..., An = Vn
WHERE C;

Базовые SQL команды DML

- **Вставка**

INSERT INTO R(A1, ..., An) VALUES (V1, ..., Vn);

INSERT INTO R(A1) SELECT A2 FROM R2;

- **Удаление**

DELETE FROM R WHERE C;

- **Обновление**

UPDATE R SET A1 = V1, A2 = V2, ..., An = Vn
WHERE C;

Примеры команды DML

Person(id, fio, age, addr, work)

```
INSERT INTO Person
    VALUES (1, ‘Иванов ИИ’, ‘Москва, Варварка,5-8’,
    ‘МГТУ’);
INSERT INTO Person( fio, age, addr, work)
    VALUES (‘Петров ПП’, ‘Москва, Ильинка,7-12’, ‘МГУ’);

DELETE FROM Person;
DELETE FROM Person WHERE id=10;

UPDATE Person SET age=age+1;
UPDATE Person
    SET addr=‘Москва, Сумской, 11-2-6’, work = ‘МГТУ’
    WHERE id=20;
```

Примеры CRUD

```
INSERT INTO Studio(name)
    SELECT DISTINCT studioName
    FROM Movie
    WHERE studioName NOT IN
        (SELECT name FROM Studio);
```

Примеры CRUD 2

```
DELETE FROM MovieExec
WHERE netWorth < 1000000;

UPDATE MovieExec
SET name = 'Pres.' || name
WHERE cert# IN
        (SELECT presC# FROM Studio);
```

Выборка данных (DQL)

- SELECT L
  FROM R
  WHERE C

- L – список выражений проекции (столбцы после SELECT)

- R – схема отношения (таблица, из которой берутся записи – после FROM)

- C – условие / селекции (условия после WHERE)

Часть SELECT

- SELECT * | список имен столбцов через запятую или выражения (Таблица.поле)

- Атрибут или Таблица.атрибут
- Выражение(атрибуты)
- SELECT DISTINCT поля – удалить дубликаты
- SELECT функция(поля) или агрегат(поля); примеры: count, avg, sum, min, max

Примеры SELECT

- SELECT * FROM Movie;
- SELECT Movie.title, avg(length) FROM Movie;
- SELECT title, length*0.01 as len, ‘час’ as inHours, now() as curtime

  FROM Movie
  WHERE studioName = ‘Назв’ AND year = 1990;

Часть FROM

- Таблица
- Представление
- Подзапрос (SELECT…)
- Имя/псевдоним – для ссылки на атрибуты

Пример FROM

```
SELECT DISTINCT old.org
FROM
(select * from Pesron where age>65) as old,
(select * from Pesron where age<30) as kid
WHERE
old.org = kid.org;
```

Часть WHERE

= <> < > >= <=

() AND () () OR () NOT ()

S LIKE P → S – строка, P – образец;

строка равна образцу или его части;

[‘_’ – 1 символ, ‘%’ - * символов]

CONTAINTS(поле, ‘текст’) – ‘текст’ содержится в поле

A IS NULL A IS NOT NULL

Примеры WHERE

```
SELECT * FROM Person
WHERE ( age>65 ) AND
   (city like ‘Mos%’ OR city = ‘SPb’) ;

SELECT * FROM Person
WHERE (email is not NULL) and
   (year between 20 and 30) and
   salary>=100000;
```

Условия в WHERE

- EXISTS R = true, если R != ∅
- S IN R = true, если S = хотя бы одному значению в R
- S NOT IN R = true, если в R нет ни одного значения из S
- S > ALL R = true, если S > всех в R
- S > ANY R = true, если есть такой r ∈ R, для которого S > r

Примеры условий

- Жители столиц

```
SELECT * FROM Person
WHERE city IN (select capital from Country);
```

- Альтернатива

```
SELECT * FROM Person
WHERE exists (select icd from Country
   WHERE Person.city = Country.capital);
```

Примеры условий 2

- Те, кто моложе некого босса

```
SELECT * FROM Person
WHERE age < ANY (select age from Boss);
```

- Те, кто моложе всех боссов

```
SELECT * FROM Person
WHERE age < ALL (select age from Boss);
```

Сортировка

- ORDER BY список полей asc | desc
- Только поля из SELECT

```
SELECT title, year, length
FROM Movie
WHERE studio > ‘MF’
ORDER BY year desc, title;
```

Запросы к нескольким таблицам

- Декартово произведение
- Соединения
- Объединение
- Пересечение
- Разность

Примеры отношений

Subject(Курс, Семестр, Дисц) - дисциплины

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД

Student(ФИО, Возраст, Дсц(FK), город) - студенты

ФИО	Возраст	Дсц(FK)	город
Ваня	25	1	Москва
Коля	17	2	Уфа

Декартово произведение

SELECT * FROM Subject, Student;

Subject

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД

Student

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
1	2	ПКШ	Коля	17	2	Уфа
2	4	БД	Ваня	25	1	Москва
2	4	БД	Коля	17	2	Уфа

Соединение по условию

SELECT * FROM Subject, Student

WHERE Курс = Дсц;

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
1 (нет)	2 (нет)	ПКШ	Коля	17 (нет)	2 (нет)	Уфа
2 (нет)	4 (нет)	БД	Ваня	25 (нет)	1 (нет)	Москва
2	4	БД	Коля	17	2	Уфа

Соединение по условию - 2

SELECT *

FROM Subject **JOIN** Student **ON** Курс = Дсц;

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
2	4	БД	Коля	17	2	Уфа

Внутреннее соединение

SELECT * FROM Subject **INNER JOIN** Student ON Курс = Дсц;

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД
3	1	ООП

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа
Аня	20	1	Москва
Том	21	null	Лондон

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
1	2	ПКШ	Аня	20	1	Москва
2	4	БД	Коля	17	2	Уфа

Внешнее полное соединение

SELECT * FROM Subject **FULL JOIN** Student
ON Курс = Дсц;

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД
3	1	ООП

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа
Аня	20	1	Москва
Том	21	null	Лондон

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
1	2	ПКШ	Аня	20	1	Москва
2	4	БД	Коля	17	2	Уфа
3	1	ООП	Null	Null	Null	Null
Null	Null	Null	Том	21	Null	Лондон

Внешнее левое соединение

SELECT * FROM Subject **LEFT JOIN** Student ON Курс = Дсц;

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД
3	1	ООП

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа
Аня	20	1	Москва
Том	21	null	Лондон

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
1	2	ПКШ	Аня	20	1	Москва
2	4	БД	Коля	17	2	Уфа
3	1	ООП	Null	Null	Null	Null

Внешнее правое соединение

SELECT * FROM Subject **RIGHT JOIN** Student ON Курс = Дсц;

Курс	Семестр	Дисц.
1	2	ПКШ
2	4	БД
3	1	ООП

ФИО	Возраст	Дсц	город
Ваня	25	1	Москва
Коля	17	2	Уфа
Аня	20	1	Москва
Том	21	null	Лондон

Результат

Курс	Семестр	Дисц.	ФИО	Возраст	Дсц	город
1	2	ПКШ	Ваня	25	1	Москва
1	2	ПКШ	Аня	20	1	Москва
2	4	БД	Коля	17	2	Уфа
Null	Null	Null	Том	21	Null	Лондон

Операции над множествами

- Количество, последовательность и типы столбцов множеств должны совпадать!

- Пересечение

(SELECT…) **INTERSECT** (SELECT…)

- Разность

(SELECT…) **EXCEPT** (SELECT…)

- Объединение множеств

(SELECT…) **UNION** (SELECT…)

- Объединение мультимножеств

(SELECT…) **UNION ALL** (SELECT…)

Группировка и агрегирование

- GROUP BY список полей
- В SELECT только группирующие атрибуты и результат агрегации прочих

SELECT work, count(*) as cnt
FROM Persons ...
WHERE age<65 ...
GROUP BY work

Пример группировки

fio	age	work	city
Ваня	25	MMM	Москва
Коля	17	MMM	Уфа
Аня	20	OOO	Москва
Саша	70	OOO	Москва
Том	21	null	Лондон

work	cnt
MMM	2
OOO	1
null	1

Количество сотрудников в организации

```
SELECT work, count(*) as cnt
FROM Persons
WHERE age<65
GROUP BY work
```

Агрегирование

- При группировке null учитывается как отдельное значение

- **null** – игнорируется при вычислении агрегатов sum() и пр;
- **count(DISTINCT A)** – считает различные значения
- **count(*)** – считает все записи
- **count(A)** – считает только где A != null

Группировка с условием

fio	age	work	city
Ваня	25	MMM	Москва
Коля	17	MMM	Уфа
Аня	20	OOO	Москва
Саша	70	OOO	Москва
Том	21	null	Лондон

work	cnt
MMM	2

Количество сотрудников в больших организации

```
SELECT work, count(*) as cnt
FROM Persons WHERE age<65
GROUP BY work
HAVING count(*) > 1
```

Порядок действий

(5) **SELECT** work, count(*) as cnt

(1) **FROM** Persons

(2) **WHERE** age<65

(3) **GROUP BY** work

(4) **HAVING** count(*) > 1

(6) **ORDER BY** cnt

Пример подзапроса

```
SELECT name
FROM MovieExec
WHERE cert# = (SELECT productC#
                FROM Movie
                WHERE titile = ‘Назв’);
```

Пример подзапроса 2

```
SELECT name
FROM MovieExec
WHERE cert# IN
    (SELECT producerC#
    FROM Movie
    WHERE (title, year) IN
        (SELECT movieTitle, movieYear
        FROM Stars2n
        WHERE starName = 'Имя'
        )
    );
```

Пример подзапроса 3

```
SELECT name
FROM MovieExec, (SELECT producerC#
                 FROM Movie, Stars2n
                 WHERE title = movieTitle
                 AND year = movieYear
                 AND starName = 'Имя'
                 ) Prod
WHERE cert# = Prod.producerC#;
```