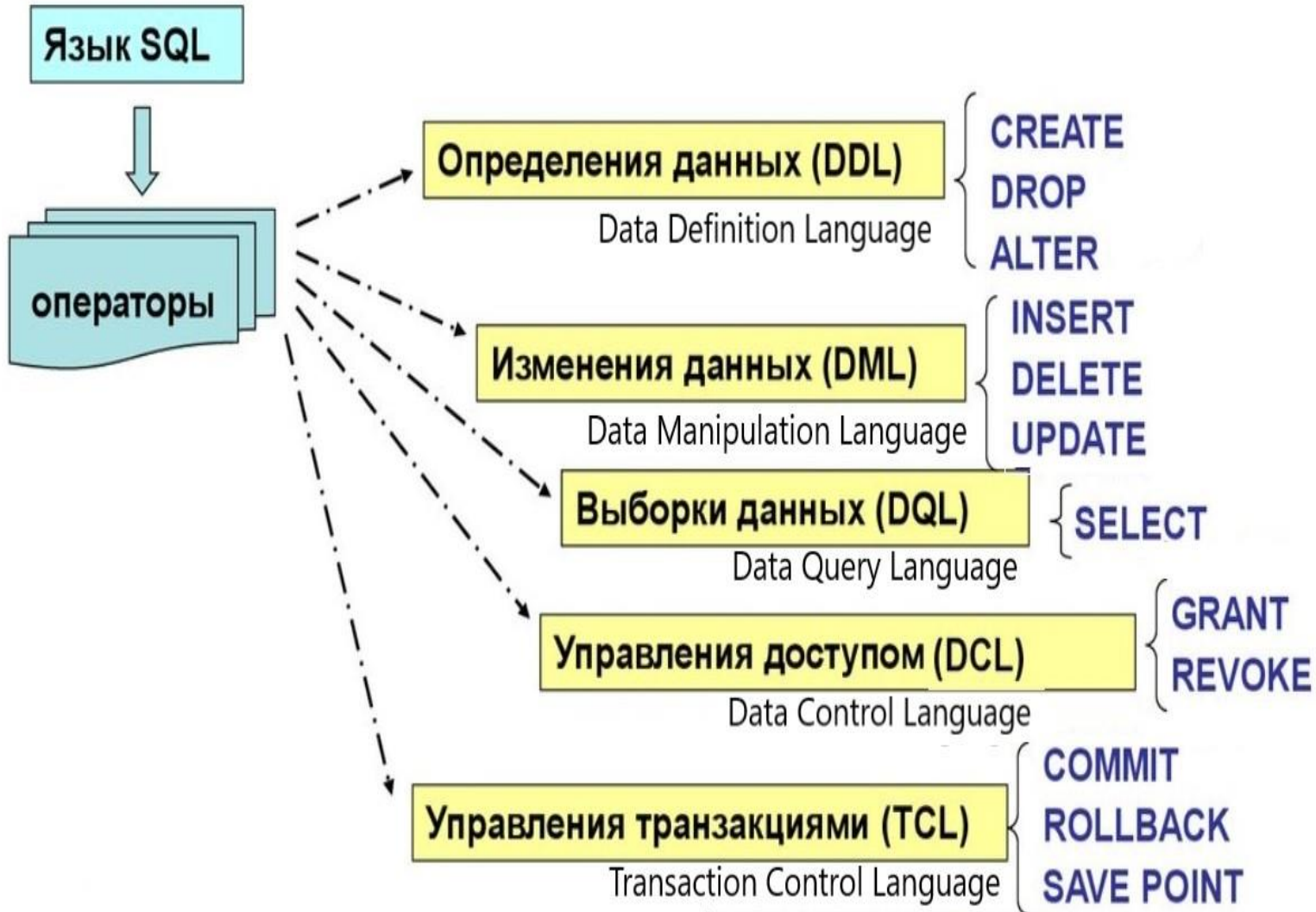


S Q L

SQL – Structured Query Language

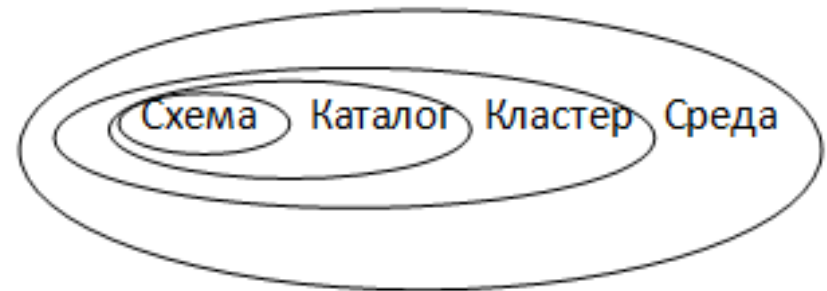
- 1986 – ANSI – SQL/86 (ISO в 1987)
- 1989 - SQL/86 – запросы и схемы
- 1992 - SQL/92 – схемы, транзакции,
соединения, авторизация
- 1995 - SQL/CLI – динамический, ODBC
- 1996 - SQL/PSM – хранимые процедуры
- 1999 – SQL:1999 (SQL3) – объектное
расширение, UDT
- 2003 – SQL:2003 – OLAP, XML
- 2006 – XQuery

Подмножества языка SQL



Среда СУБД

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



Основные объекты БД

Таблицы (table)

Представления (view)

Триггеры (trigger)

Ограничения целостности

Хранимые процедуры и функции (procedure)

Правила сравнения (collation)

Домены (domain)

Последовательности (sequence)

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

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;

CRUD – Create, Retrieve, Update, Delete

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 (столбцы после SELECT)
- R – схема отношения (таблица, из которой берутся записи – после FROM)
- C – условие / селекции (условия после WHERE)

Часть SELECT

SELECT * *или* список имен столбцов
через запятую (или *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 ='MF' AND year =1999;`

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]

CONTAINS(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 $R \neq \emptyset$
- *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 $r \in R$, 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 какого-нибудь 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, Дисцпл) -- дисциплины

<u>Курс</u>	Семестр	Дисцпл
1	2	ПКШ
2	4	БД

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

<u>ФИО</u>	Возраст	Дисц(FK)	город
Ваня	25	1	Москва
Коля	17	2	Уфа

Декартово произведение

SELECT * FROM Subject, Student;

Subject

Курс	Семестр	Дисцпл
1	2	ПКШ
2	4	БД

Student

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



Результат

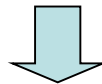
Курс	Семестр	Дисцпл	ФИО	Возраст	Дисц	город
1	2	ПКШ	Ваня	25	1	Москва
1	2	ПКШ	Коля	17	2	Уфа
2	4	БД	Ваня	25	1	Москва
2	4	БД	Коля	17	2	Уфа

Соединение по условию

```
SELECT * FROM Subject, Student  
WHERE Курс = Дисц;
```

Курс	Семестр	Дисцпл
1	2	ПКШ
2	4	БД

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



Результат

Курс	Семестр	Дисцпл	ФИО	Возраст	Дисц	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 = Дисц;

<u>Курс</u>	Семестр	Дисцпл
1	2	ПКШ
2	4	БД

<u>ФИО</u>	Возраст	Дисц	город
Ваня	25	1	Москва
Коля	17	2	Уфа



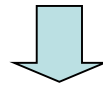
Результат

Курс	Семестр	Дисцпл	ФИО	Возраст	Дисц	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 = Дисц;**

Курс	Семестр	Дисцпл
1	2	ПКШ
2	4	БД
3	1	ООП



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

Результат

Курс	Семестр	Дисцпл	ФИО	Возраст	Дисц	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 = Дисц;

Курс	Семестр	Дисцпл
1	2	ПКШ
2	4	БД
3	1	ООП

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



Курс	Семестр	Дисцпл	ФИО	Возраст	Дисц	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	Петр	21	Null	Лондон

Результат

Внешнее левое соединение

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

Курс	Семестр	Дисцпл
1	2	ПКШ
2	4	БД
3	1	ООП

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



Курс	Семестр	Дисц.	ФИО	Возраст	Дисц	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

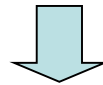
Результат

Внешнее правое соединение

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

Курс	Семестр	Дисцпл
1	2	ПКШ
2	4	БД
3	1	ООП

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



Курс	Семестр	Дисцпл	ФИО	Возраст	Дисц	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	Петр	21	Null	Лондон

Результат

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	ПАО	Москва
Коля	17	ПАО	Уфа
Аня	20	ООО	Москва
Саша	70	ООО	Москва
Петр	21	null	Лондон



work	cnt
ПАО	2
ООО	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	ПАО	Москва
Коля	17	ПАО	Уфа
Аня	20	ООО	Москва
Саша	70	ООО	Москва
Петр	21	null	Лондон



work	cnt
ПАО	2

Количество сотрудников в большой 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 title = 'Avatar');
```

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'
     )
  );
```

