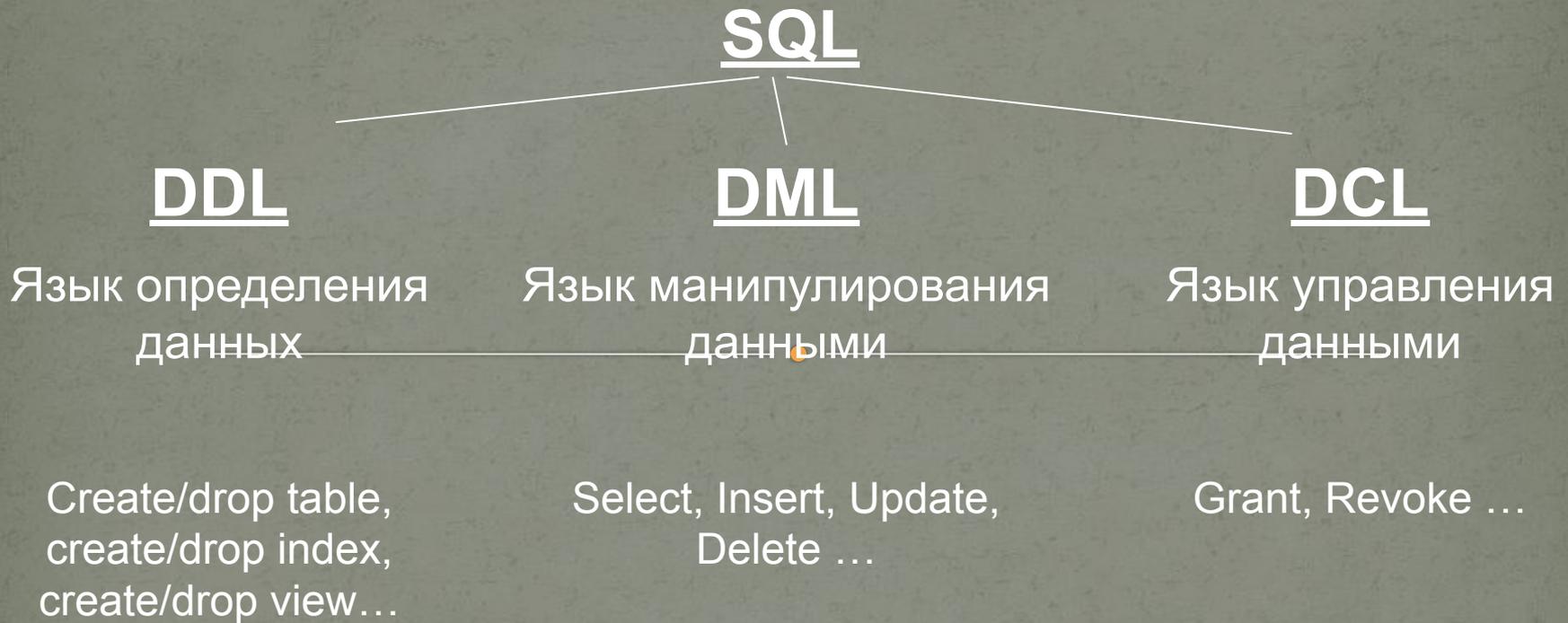


БАЗЫ ДАННЫХ

**Использование SQL
для построения
запросов**

Разделы языка SQL

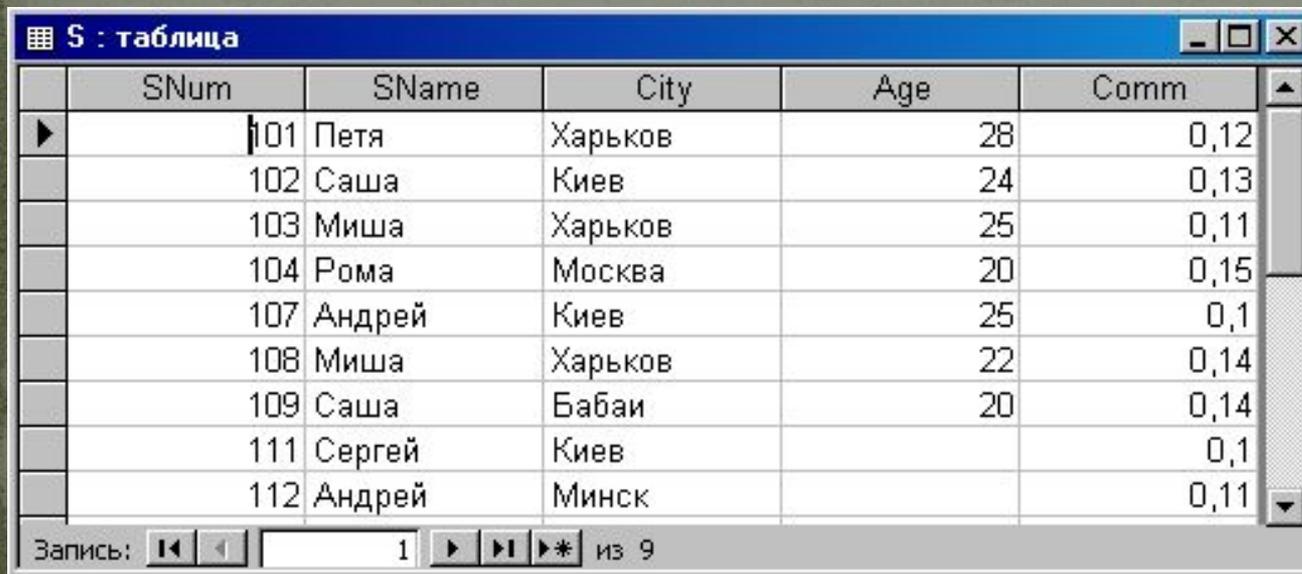


Команда SELECT

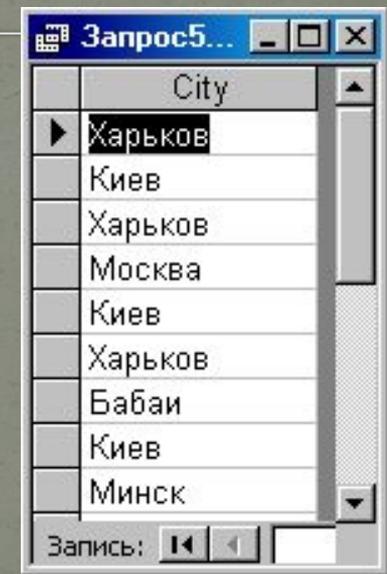
ЗАПРОС – команда, которая адресуется к Базе Данных для вывода требуемой информации из таблиц

SELECT перечень требуемой информации **FROM** имя таблицы ;

SELECT City FROM S;



SNum	SName	City	Age	Comm
101	Петя	Харьков	28	0,12
102	Саша	Киев	24	0,13
103	Миша	Харьков	25	0,11
104	Рома	Москва	20	0,15
107	Андрей	Киев	25	0,1
108	Миша	Харьков	22	0,14
109	Саша	Бабаи	20	0,14
111	Сергей	Киев		0,1
112	Андрей	Минск		0,11



City
Харьков
Киев
Харьков
Москва
Киев
Харьков
Бабаи
Киев
Минск

Синтаксис команды SELECT

```
SELECT * |  
    [DISTINCT | ALL] <поля, скалярные выражения,  
константы>.,..  
    FROM {<имя таблицы> [ < псевдоним > ] ....  
    [ WHERE < условие для строк>]}  
    [ GROUP BY { <имя поля, по которому  
производится группировка> | <номер поля> }.,..]  
    [ HAVING <условие для группы>]  
    [ ORDER BY { <имя поля> | <номер поля> }.,..  
[ASC | DESC]]  
    [ UNION [ALL] SELECT...]
```

Вывод полей таблицы в запросе

Запрос № 1: вывести выборочно поля таблицы.

```
SELECT SNum, SName, City, Comm FROM S;
```

Результат:

SNum	SName	City	Comm
101	Петя	Харьков	0,12
102	Саша	Киев	0,13
103	Миша	Харьков	0,11
104	Рома	Москва	0,15
107	Андрей	Киев	0,1
108	Миша	Харьков	0,14
109	Саша	Бабаи	0,14
111	Сергей	Киев	0,1
112	Андрей	Минск	0,11

Запрос № 2: вывести всю информацию о служащих.

```
SELECT SNum, SName, City, Age, Comm FROM S;
```

```
SELECT * FROM S;
```

Оператор * заменяет список всех полей таблицы.

Аргумент DISTINCT

Аргумент DISTINCT - удаляет избыточные данные.

Запрос № 3: вывести города, в которых проживают служащие.

```
SELECT DISTINCT City FROM S;
```

Результат:

City
Бабаи
Киев
Минск
Москва
Харьков

Предложение WHERE - накладывает ограничение на выборку.

Запрос № 4: вывести всю информацию о служащих, проживающих в городе Харьков.

```
SELECT * FROM S WHERE City='Харьков';
```

Результат:

SNum	SName	City	Age	Comm
101	Петя	Харьков	28	0,12
103	Миша	Харьков	25	0,11
108	Миша	Харьков	22	0,14

SELECT: WHERE

- Сравнение: сравниваются результаты вычисления одного выражения с результатами вычисления другого.
- Диапазон: проверяется, попадает ли результат вычисления выражения в заданный диапазон значений.
- Принадлежность множеству: проверяется, принадлежит ли результат вычислений выражения заданному множеству значений.
- Соответствие шаблону: проверяется, отвечает ли некоторое строковое значение заданному шаблону.
- Значение NULL: проверяется, содержит ли данный столбец определитель NULL (неизвестное значение).

SELECT: WHERE (Сравнение)

Операторы сравнения

- =
- <
- >
- <=
- >=
- <>
- AND
- OR
- NOT

SELECT: WHERE (Сравнение)

- Выражение вычисляется слева направо.
- Первыми вычисляются подвыражения в скобках.
- Операторы NOT выполняются до выполнения операторов AND и OR.
- Операторы AND выполняются до выполнения операторов OR.

SELECT: WHERE (Сравнение)

Использование в предложении WHERE:

- реляционных операторов (=, >, <, >=, <=, <>);

Запрос №5: вывести имена служащих, комиссионные которых выше 12%.

```
SELECT SName FROM S WHERE Comm>0.12;
```

Результат:

SName
Саша
Рома
Миша
Саша

- булевых операторов (AND, OR, NOT).

Запрос № 6: вывести информацию о служащих, которые не живут в городе «Москва», но имеют комиссионные не ниже 14%.

```
SELECT * FROM S WHERE NOT City='Москва' AND  
Comm>=0.14;
```

Результат:

SNum	SName	City	Age	Comm
108	Миша	Харьков	22	0,14
109	Саша	Бабаи	20	0,14

SELECT: WHERE (Диапазон)

- оператор BETWEEN (определяет диапазон для значений поля);

Запрос № 7: вывести информацию о служащих с комиссионными больше 10%, но меньше 14%.

```
SELECT * FROM S WHERE
```

```
Comm BETWEEN 0.12 AND 0.14;
```

Результат:

SNum	SName	City	Age	Comm
101	Петя	Харьков	28	0,12
102	Саша	Киев	24	0,13
108	Миша	Харьков	22	0,14
109	Саша	Бабаи	20	0,14

```
SELECT * FROM S WHERE Comm >= 0.12 AND Comm <= 0.14;
```

Для исключения концов диапазона:

```
SELECT * FROM S WHERE (Comm BETWEEN 0.12 AND 0.14) AND NOT Comm IN (0.12, 0.14);
```

SELECT: WHERE (Принадлежность множеству)

- оператор IN (определяет набор значений для поля, которые могут попасть в результат вывода);

Запрос №8: вывести имена служащих, проживающих в Киеве или в Москве.

```
SELECT SName FROM S WHERE City='Киев' OR City='Москва';
```

```
SELECT SName FROM S WHERE City IN ('Киев',  
'Москва');
```

Результат:

SName
Раша
Рома
Андрей
Сергей

Запрос № 9: вывести всю информацию о служащих с номерами 102, 107, 111 и 112.

```
SELECT * FROM S WHERE SNum IN (102, 107, 111, 112);
```

SELECT: WHERE (Соответствие шаблону)

- % - вместо этого символа может быть подставлено любое количество произвольных символов.
- _ - заменяет один символ строки.
- [] - вместо символа строки будет подставлен один из возможных символов, указанный в этих ограничителях.
- [^] - вместо соответствующего символа строки будут подставлены все символы, кроме указанных в ограничителях.

Оператор LIKE

▪ оператор LIKE (накладывает маску на значения поля);

/ '_' – заменяет один символ; '%' – 0 или несколько символов/

Запрос № 10: вывести информацию о служащих, проживающих в городах, начинающихся на букву М.

```
SELECT * FROM S WHERE City LIKE 'M%';
```

Результат:

SNum	SName	City	Age	Comm
104	Рома	Москва	20	0,15
112	Андрей	Минск		0,11

Запрос № 11: вывести информацию о служащих, в именах которых встречается буква «р».

```
SELECT * FROM S WHERE SName LIKE 'P_%' OR  
SName LIKE '%p%';
```

Запрос № 12: Вывести все номера телефонов, в которых вторая цифра равна 3 или 5.

```
SELECT PHONE  
FROM PHONENOTE.PHONE  
WHERE PHONE LIKE '_[35]%'
```

Оператор NULL

Оператор NULL (указывает на отсутствие значения)

- *Запрос № 12: вывести информацию о служащих, для которых не известен возраст.*
- SELECT * FROM S WHERE Age IS NULL;
- Пример: Вывести все записи из таблицы PHONE, в которых не указан номер телефона
- SELECT PHONE
- FROM PHONENOTE.PHONE
- WHERE PHONE IS NULL;
- Пример: Вывести все записи из таблицы PHONE, в которых указан номер телефона
- SELECT PHONE
- FROM PHONENOTE.PHONE
- WHERE PHONE IS NOT NULL;

Формирование вывода запроса

использование скалярных выражений

Запрос № 13: вывести имена служащих, города их проживания и КОМИССИОННЫЕ.

```
SELECT Sname, City, Comm*100 FROM S;
```

Результат:

SName	City	Expr1002
Петя	Харьков	12
Саша	Киев	13
Рома	Москва	15
Андрей	Киев	10
Миша	Харьков	11
Миша	Харьков	14
Саша	Бабаи	14
Сергей	Киев	10
Андрей	Минск	11

использование текста в выводе

Запрос № 14: вывести информацию о служащих с пояснениями.

```
SELECT 'Служащий ', Sname, ' проживает в городе ',  
City FROM S;
```

Результат:

Expr1000	Sname	Expr1002	City
Служащий	Петя	проживает в городе	Харьков
Служащий	Саша	проживает в городе	Киев
Служащий	Миша	проживает в городе	Харьков
Служащий	Рома	проживает в городе	Москва

Вычисляемые поля и псевдонимы

Псевдонимы применяются с целью дать тому или иному столбцу более подходящее название, а также в конструкциях GROUP BY, ORDER BY или HAVING для ссылки на столбец.

Вывести Фамилию и имя в одном поле

```
SELECT LASTNAME || ' ' || FIRSTNAME AS NAME,  
ADDRESS
```

```
FROM PHONENOTE;
```

**Вывести все известные нам года рождения из
таблицы BIRTHDAY**

```
SELECT YEAR(BIRTHDAY) AS YEAR FROM PHONENOTE;
```

Упорядочивание результатов вывода /ORDER BY/

- ASC – по возрастанию (используется по умолчанию);
- DESC – по убыванию.

Запрос № 15: вывести информацию о служащих, упорядочив ее по возрастанию комиссионных.

```
SELECT * FROM S ORDER BY Comm ASC;
```

Результат:

SNum	SName	City	Age	Comm
111	Сергей	Киев		0,1
107	Андрей	Киев	25	0,1
112	Андрей	Минск		0,11
103	Миша	Харьков	25	0,11
101	Петя	Харьков	28	0,12
102	Саша	Киев	24	0,13
109	Саша	Бабаи	20	0,14
108	Миша	Харьков	22	0,14
104	Рома	Москва	20	0,15

```
SELECT * FROM S  
ORDER BY Comm;
```

Запрос № 16: вывести информацию о служащих города Киев в порядке убывания возраста.

```
SELECT * FROM S WHERE City='Киев' ORDER BY Age DESC;  
SELECT * FROM S WHERE City='Киев' ORDER BY 4 DESC;
```

Агрегатные функции

Агрегатная функция (групповая операция) выдает одиночное значение для группы записей таблицы.

- COUNT – считает количество строк или не-NULL значений поля;
- SUM – выдает арифметическую сумму всех выбранных значений поля;
- AVG – производит усреднение всех выбранных значений поля;
- MAX – выводит наибольшее из значений поля;
- MIN – выводит наименьшее значение из поля.

SELECT агрегатная функция(наименование поля)

FROM имя таблицы;

Функция COUNT

Запрос № 17: посчитать количество служащих в таблице.

```
SELECT COUNT (SNum) FROM S;
```

Результат:

Expr1000
9

Запрос № 18: посчитать количество городов, в которых проживают служащие.

```
SELECT COUNT (DISTINCT City) FROM S;
```

Результат:

Expr1000
5

Оператор * позволяет посчитать количество строк в таблице.

Запрос № 19: посчитать количество клиентов в таблице.

```
SELECT COUNT (*) FROM C;
```

Функции AVG, MAX и MIN

Запрос № 20: посчитать средний размер комиссионных служащих.

```
SELECT AVG (Comm) FROM S;
```

Результат:

Expr1000
0,12222222

Запрос № 21: найти максимальный возраст служащих.

```
SELECT MAX (Age) FROM S;
```

Результат:

Expr1000
28

Запрос № 22: вывести имя первого по алфавиту служащего.

```
SELECT MIN (SName) FROM S;
```

Результат:

Expr1000
Андрей

Предложение GROUP BY

Предложение GROUP BY (позволяет выделить группу записей для проведения групповой операции):

Запрос № 23: по каждому городу посчитать средний размер комиссионных служащих.

```
SELECT City, AVG (Comm) FROM S GROUP BY City;
```

SNum	SName	City	Age	Comm
101	Петя	Харьков	28	0,12
102	Саша	Киев	24	0,13
103	Миша	Харьков	25	0,11
104	Рома	Москва	20	0,15
107	Андрей	Киев	25	0,1
108	Миша	Харьков	22	0,14
109	Саша	Бабаи	20	0,14
111	Сергей	Киев		0,1
112	Андрей	Минск		0,11

Результат:

City	Expr1001
Бабаи	0,1400000006
Киев	0,1099999994
Минск	0,1099999994
Москва	0,1500000006
Харьков	0,1233333324

Запросы с GROUP BY

Запрос № 24: по каждой возрастной группе посчитать количество служащих, относящейся к ней.

```
SELECT Age, COUNT (SNum)
FROM S GROUP BY Age;
```

Результат:

Age	Expr1001
	2
20	2
22	1
24	1
25	2
28	1

Запрос № 25: вывести статистику о группах служащих с одинаковыми размерами комиссионных по городам.

```
SELECT City, Comm, COUNT(SNum) FROM S
GROUP BY City, Comm;
```

Результат:

City	Comm	Expr1002
Бабаи	0,14	1
Киев	0,1	2
Киев	0,13	1
Минск	0,11	1
Москва	0,15	1
Харьков	0,11	1
Харьков	0,12	1
Харьков	0,14	1

Предложение HAVING

(накладывает ограничения на вывод групповых записей)

Запрос № 28: вывести информацию по городам и количеству служащих в них, если это количество больше одного.

```
SELECT City, COUNT(SNum) FROM S GROUP BY City  
HAVING COUNT(SNum)>1 ;
```

Результат:

City	Expr1001
Киев	3
Харьков	3

Запрос № 29: вывести информацию о городах, средние комиссионные в которых не ниже 13%.

```
SELECT City, AVG(Comm) FROM S GROUP BY City  
HAVING AVG(Comm)>0.13 ;
```

Результат:

City	Expr1001
Бабаи	0,140000000
Москва	0,150000000

Добавление строк (INSERT)

INSERT INTO название_таблицы VALUES
(значение1[, значение2]);

- INSERT INTO PHONENOTE VALUES (1, 'IVANOV', 'IVAN', 'PR. POBEDI, 40');
- INSERT INTO PHONENOTE (NOTE_ID, LASTNAME, FIRSTNAME) VALUES (1, 'Смирнов', 'Андрей');

Модификация строк (UPDATE)

UPDATE название_таблицы SET название_поля1= значение[, название_поля2= значение] WHERE условие;

- UPDATE PHONENOTE SET ADDRESS='г. Челябинск';
- UPDATE PHONENOTE SET ADDRESS='г. Челябинск' WHERE ADDRESS IS NULL;
- UPDATE PHONENOTE SET ADDRESS=NULL.

Удаление строк (DELETE)

Удалить все записи из таблицы NOTE

```
DELETE FROM PHONENOTE;
```

Удалить первую запись из таблицы NOTE

```
DELETE FROM PHONENOTE WHERE NOTE_ID=1;
```