

# Базы данных

Лекция 7

Язык SQL

# Язык SQL

- История и стандарты языка SQL
  - Первый стандарт – 1986
  - Наиболее используемый – SQL'92
  - Последний – SQL:2008 (6 версия)

# Язык SQL

- Классификация операторов:
  - выборка (SELECT)
  - DML: модификация (INSERT, DELETE, UPDATE, ...)
  - DDL: изменение структуры (CREATE, DROP, ALTER, ...)
  - TCL: управление транзакциями (COMMIT, ROLLBACK, ...)
  - DCL: управление доступом (GRANT, REVOKE, ...)
- Структура операторов SQL
  - <имя><аргументы>[<фраза>[,...]]
- Структура фразы:
  - <имя><аргументы>

# Оператор SELECT

- SELECT { \* | [DISTINCT] <столбец> | <выражение> [alias], ... }
- FROM <источники>
- [WHERE <условие>]
- [GROUP BY <столбцы>]
- [HAVING <условие>]
- [UNION <оператор SELECT>]
- [ORDER BY <столбцы>]

# Однотабличные запросы

- простейший вариант – отображение всей таблицы  
`SELECT * FROM Student;`
- выборка столбцов  
`SELECT FirstName, LastName FROM Student;`

# Выражения

- Базовые элементы: имена столбцов, константы (числовые и строковые), встроенные функции (например, DATE())
- Константы: 5,  $-3.5e-4$ , 'строка', '26.09.2014' ,  
''' строка'''
- Арифметические операторы: + - \* /
- Строковый оператор конкатенации: +
- Скобки в выражениях – в неограниченном количестве

# Выражения

- `SELECT LastName, Salary + 300  
FROM Employee;`
- `SELECT LastName, Salary, 12 * Salary + 100  
FROM Employee;`
- `SELECT LastName, Salary, 12 * (Salary + 100)  
FROM Employee;`
- `SELECT LastName + ' is a ' + JobID  
FROM employees;`

# NULL-значения

- NULL – значение, которое недоступно, неопределено, неизвестно или неприменимо. Не то же самое, что 0 или пробел.
- Если это значение встретится в выражении, все выражение будет равно NULL.

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

R	LAST_NAME	R	JOB_ID	R	SALARY	R	COMMISSION_PCT
1	King		AD_PRES		24000		(null)
2	Kochhar		AD_VP		17000		(null)
3	De Haan		AD_VP		17000		(null)

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

R	LAST_NAME	R	12*SALARY*COMMISSION_PCT
1	King		(null)
2	Kochhar		(null)
3	De Haan		(null)
4	Hunold		(null)



# Псевдонимы столбцов (alias)

- Переименовывают заголовок столбца в выборке
- Удобны использовать в вычислениях
- Следуют за названием столбца или выражением (возможно использование ключевого слова AS)
- Для использования пробелов следует использовать двойные кавычки

```
SELECT last_name AS name, commission_pct comm  
FROM employees;
```

```
SELECT last_name Name , salary * 12 "Annual Salary"  
FROM employees;
```

# Дубликаты столбцов (DISTINCT)

- По умолчанию дубликаты допустимы
- Для исключения дубликатов – ключевое слово DISTINCT

```
SELECT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	90
2	90
3	90
4	60
5	60
6	60
7	50
8	50

```
SELECT DISTINCT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	(null)
2	90
3	20
4	110
5	50
6	80
7	60
8	10

# Ограничения (WHERE)

- Операции сравнения: = <> < > <= >=

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees
WHERE  department_id = 90;
```

```
SELECT last_name, job_id, department_id
FROM   employees
WHERE  last_name = 'Whalen';
```

```
SELECT last_name
FROM   employees
WHERE  hire_date = '17-FEB-96';
```

- Нельзя использовать псевдонимы столбцов в WHERE

# Ограничения (WHERE)

- **BETWEEN <нач\_значение> AND <конеч\_значение>**

```
SELECT last_name, salary
FROM   employees
WHERE  salary BETWEEN 2500 AND 3500;
```

- **IN (<набор значений>)**

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201);
```

# Ограничения (WHERE). LIKE

- LIKE <шаблон> [ESCAPE '<символ>']
  - % - произвольное количество символов
  - \_ - один произвольный символ

```
SELECT first_name  
FROM employees  
WHERE first_name LIKE 'S%';
```

```
SELECT employee_id, last_name, job_id  
FROM employees  
WHERE job_id LIKE '%SA\_%' ESCAPE '\\';
```

# Ограничения (WHERE). Проверка на NULL

- Операторы IS NULL, IS NOT NULL

```
SELECT last_name, manager_id  
FROM   employees  
WHERE  manager_id IS NULL;
```

- Сравнивать с NULL с помощью операторов сравнения некорректно

# Ограничения (WHERE). Логические операторы

- Операторы AND, OR, NOT

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >= 10000
AND    job_id LIKE '%MAN%';
```

```
SELECT last_name, job_id
FROM   employees
WHERE  job_id
      NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

# Сортировка (ORDER BY)

- Предложение ORDER BY

```
SELECT last_name, job_id, department_id, hire_date  
FROM employees  
ORDER BY hire_date DESC;
```

```
SELECT employee_id, last_name, salary*12 annsal  
FROM employees  
ORDER BY annsal;
```

```
SELECT last_name, job_id, department_id, hire_date  
FROM employees  
ORDER BY 3;
```

```
SELECT last_name, department_id, salary  
FROM employees  
ORDER BY department_id, salary DESC;
```