

Лекция #5

# Базы данных

Михаил Моисеев

SQL-DML

# Определения

В SQL определены два подмножества языка:

- **SQL-DDL (Data Definition Language)** - язык определения структур и ограничений целостности баз данных.  
Сюда относятся команды создания и удаления баз данных; создания, изменения и удаления таблиц; управления пользователями и т.д.
- **SQL-DML (Data Manipulation Language)** - язык манипулирования данными: добавление, изменение, удаление и извлечение данных, управления транзакциями

# Типы данных

Целые числа:

- INTEGER - обычное целое 4 байта;
- SMALLINT - короткое целое 2 байта.

Строки:

- CHAR(n) - символьная строка длины n;
- VARCHAR(n) - символьная строка переменной длины, но не более n;

Вещественные типы данных:

- FLOAT - число с плавающей точкой 4 байта.
- DECIMAL(p,n) - p - число знаков до запятой, n - после запятой;
- DOUBLE PRECISION - число с плавающей точкой 8 байт.

Другие типы данных:

- BLOB – двоичные типы данных большого размера;
- DATE – дата с точностью до дня, единица измерения день;
- TIME - время с точностью до мс, единица измерения секунда;
- TIMESTAMP – дата и время вместе.

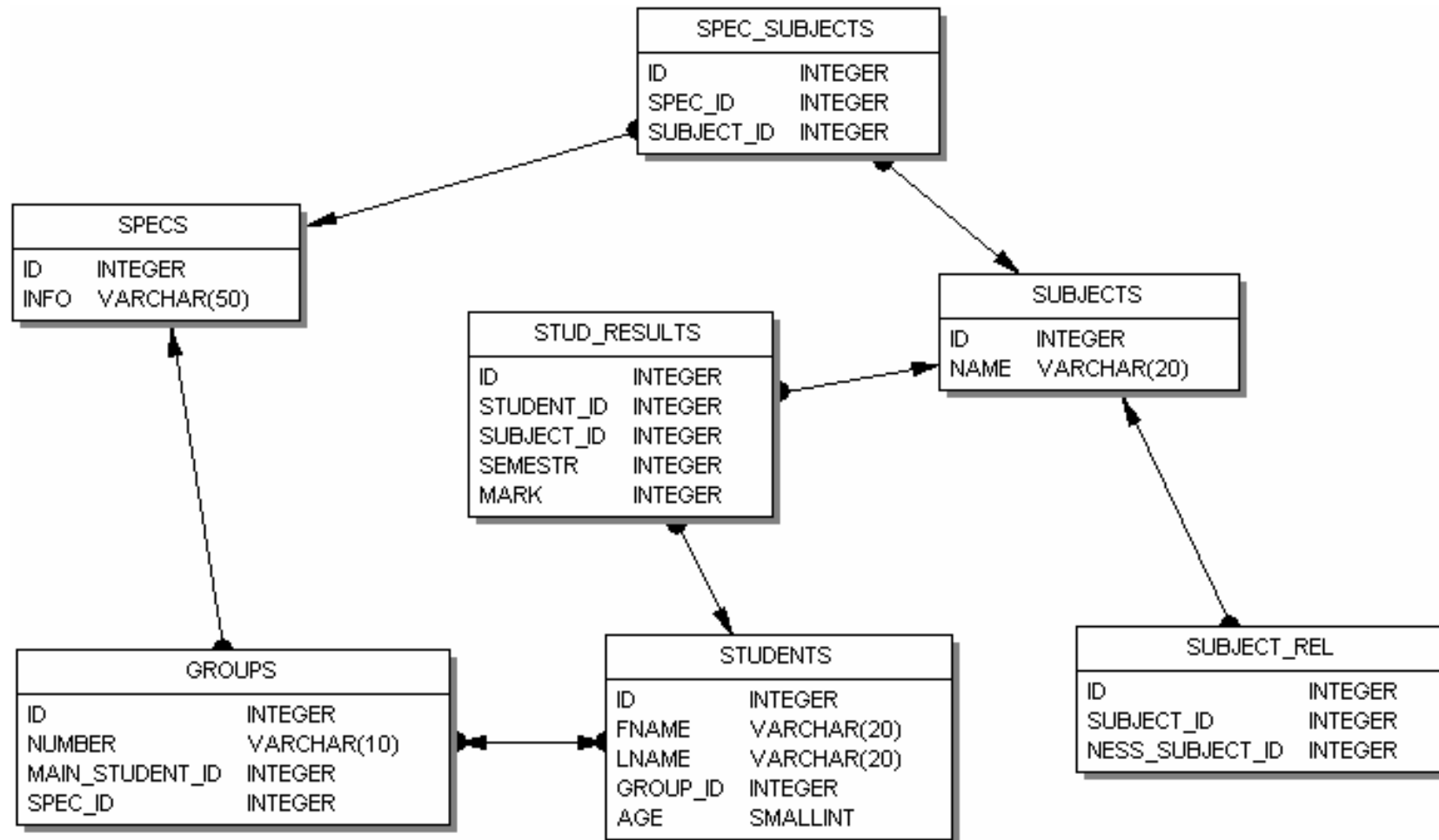
# Оператор SELECT

**SELECT** – оператор извлечения записей из таблиц в SQL.  
Операции РА: ПРОЕКЦИЯ, ВЫБОРКА, СОЕДИНЕНИЕ.

```
SELECT [ALL | DISTINCT] <список_выбора>  
  FROM <имя_таблицы>, ...  
  [WHERE <условие> ]  
  [GROUP BY <имя_столбца>,... ]  
  [HAVING <условие> ]  
  [ORDER BY <имя_столбца> [ASC | DESC],... ]
```

Порядок предложений должен строго соблюдаться.

# Схема БД



# Оператор SELECT #2

- SELECT \* FROM students
- SELECT id, fname, lname as “Фамилия”, age FROM students
- SELECT fname, lname FROM students  
WHERE (fname like ‘%A’) and (age > 20)
- SELECT fname, lname FROM students  
WHERE not (id between 100 and 200)
- SELECT fname, lname FROM students  
WHERE lname in (‘Ivanov’, ‘Petrov’, ‘Sidorov’)

Операторы:

=, <>, >, <, AND, OR, NOT, LIKE, BETWEEN... AND ..., IN (...)

# Оператор SELECT #3

Соединение нескольких таблиц:

- ❑ SELECT fname, lname, number FROM students, groups
- ❑ SELECT fname, lname, number FROM students, groups  
WHERE students.group\_id = groups.id
- ❑ SELECT fname, lname FROM students, groups  
WHERE (students.group\_id = groups.id) and (groups.number = '4081/2')
- ❑ SELECT fname, lname, specs.name FROM students, groups, specs  
WHERE (students.group\_id = groups.id) and (groups.spec\_id = specs.id)

Вычисляемые поля:

- ❑ SELECT fname, lname, (2008 - age) AS YoB FROM students

# Оператор SELECT #4

Вычисление совокупных характеристик:

- SELECT MAX(age) FROM students
- SELECT AVG(mark) FROM students, stud\_results  
WHERE (students.id = stud\_results.student\_id) and (lname = 'Ivanov')

Агрегатные функции: MAX(<поле>), MIN (<поле>),  
COUNT (<поле>), AVG (<поле>), SUM (<поле>).



# Оператор SELECT #5

Группировка данных:

- SELECT groups.id from students, groups  
WHERE students.group\_id = groups.id
- SELECT groups.id, COUNT(students.id) from students, groups  
WHERE students.group\_id = groups.id GROUP BY groups.id
- SELECT fname, lname, AVG(mark) from students, stud\_results  
WHERE students.id = stud\_results.student\_id GROUP BY lname, fname

При группировке в выбираемом списке могут быть только поля, по которым делается группировка и агрегатные функции

# Оператор SELECT #6

Ограничения на результаты группировки:

- SELECT groups.id, COUNT(students.id) as s\_count from students, groups  
WHERE (students.group\_id = groups.id) GROUP BY groups.id  
HAVING COUNT(students.id) > 10

Условия могут содержать только выбираемые поля.

Сортировка выбранных записей:

- SELECT fname, lname, age FROM students ORDER BY age
- SELECT fname, lname, age FROM students  
ORDER BY fname DESC, lname

Выбор записей с исключением повторений:

- SELECT DISTINCT fname, lname FROM students

# Оператор SELECT #7

Вложенные запросы в конструкции WHERE:

- SELECT fname, lname FROM students WHERE age = (SELECT MAX(age) FROM students)
- SELECT fname, lname FROM students WHERE (SELECT AVG(mark) FROM stud\_results WHERE stud\_results.student\_id = students.id ) > 3.5

Вложенные запросы в конструкции FROM (виртуальные таблицы):

- SELECT fname, lname, s\_mark FROM students, (SELECT students.id as s\_id, AVG(mark) as s\_mark FROM students, stud\_results WHERE stud\_results.student.id = students.id ) as vt\_marks, WHERE (students.id = vt\_marks.s\_id) and (vt\_marks.s\_mark > 3.5)

# Оператор SELECT #8

Выбор заданного числа записей:

- SELECT **FIRST 1** fname, lname FROM students WHERE age = 21
- SELECT **FIRST 2 SKIP 1** fname, lname FROM students WHERE age = 21
- SELECT fname, lname FROM students WHERE age = 21  
**ROWS 2 TO 3**

Операторы: FIRST<число>, SKIP <число>,  
ROWS <число> [TO<число>]

# Оператор SELECT #9

Преобразование типов:

- SELECT fname, lname, AVG(CAST(mark AS float))  
FROM students, stud\_results  
WHERE students.id = stud\_results.student\_id GROUP BY fname, lname

Оператор: CAST (<поле> AS <требуемый\_тип>)

Работа с датой и временем:

- SELECT fname, lname, CURRENT\_DATE - CAST(birthdate AS DATE)  
FROM students  
WHERE EXTRACT(month FROM birthdate) = 3
- CAST('22.03.2010' AS DATE)

Операторы: CURRENT\_DATE, CURRENT\_TIME,  
EXTRACT (year | month | day FROM <поле>),  
CAST('NOW' AS DATE | TIME | TIMESTAMP)

# Значение NULL

Трехзначная логика: TRUE, FALSE, NULL.

Значение NULL – существует для переменных любого типа  
Значение NULL – недопустимо для PK и допустимо для FK.

Любые операции с полем содержащим NULL дает NULL.

Пусть  $A = 1$ ,  $B = \text{NULL}$ , тогда:

- $(A + B) = \text{NULL};$
- $(A = B) = \text{NULL};$
- $(A \parallel B) = \text{NULL};$
- $(A <> B) = \text{NULL};$
- $(B = \text{NULL}) = \text{NULL};$
- $\text{not} ( B ) = \text{NULL};$

# Вычисление условий с NULL

Логические операции AND и OR с NULL:

- NULL or false = NULL
- NULL or true = true
- NULL or NULL = NULL
- NULL and false = false
- NULL and true = NULL
- NULL and NULL = NULL

Запросы должны учитывать возможность появления NULL для полей у которых явно не указано NOT NULL. NULL может появляться при операциях и при вычислении условий в конструкциях WHERE, HAVING, IF, ... .

В агрегатных функциях NULL игнорируется, за исключением COUNT(\*).

# Вычисление условий с NULL#2

Примеры запросов не учитывающих появление NULL

- SELECT fname || lname FROM students
- IF (A <> B) THEN  
    result = 'NOT EQUAL'  
ELSE  
    result = 'EQUAL';

Операторы обрабатывающие NULL:

**DISTINCT (<>), NOT DISTINCT (=)** – результат только TRUE или FALSE

```
IF (A IS DISTINCT B) THEN  
    result = 'NOT EQUAL'  
ELSE  
    result = 'EQUAL';
```

IS NULL, IS NOT NULL – явные проверки на NULL;

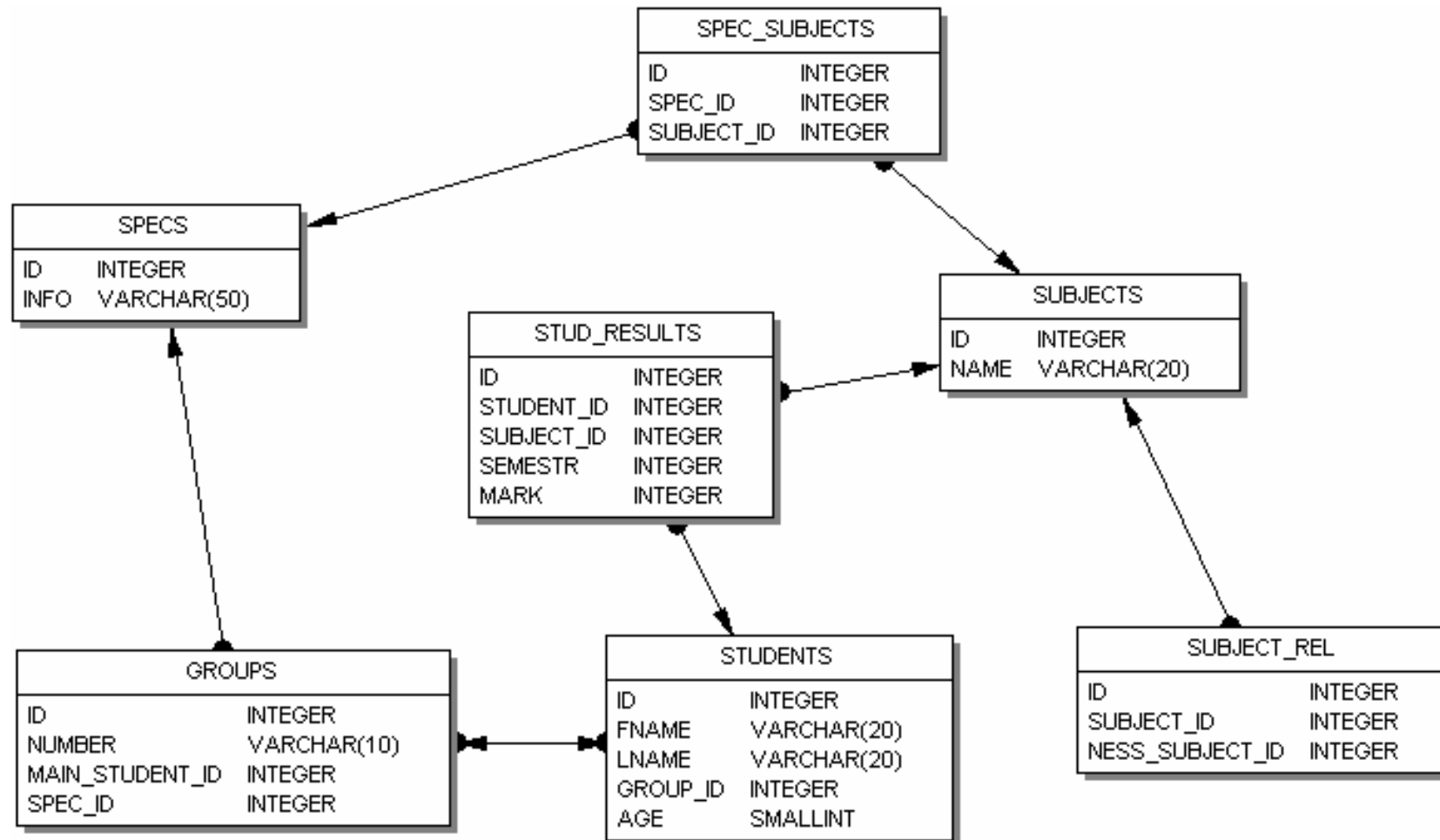
**NULLIF(<выражение 1>, <выражение2>)** – если <выражение1> равно NULL, то выдаст <выражение2>;



# Вопросы

- Какие операции РД можно выполнить с помощью оператора SELECT ?
- Перечислите конструкции оператора SELECT.
- Требуется ли соблюдения порядка следования конструкций ?
- Как осуществить выборку записей в соответствии с заданным условием ?
- Какое число записей может выдать оператор SELECT ?
- Можно ли отбросить повторяющиеся в запросе записи ?
- Для чего используются агрегатные функции ?
- Придумайте пример использования группировки.
- Каким образом осуществляется соединение нескольких таблиц ?
- Чем отличается условие WHERE от условия HAVING ?
- Какие варианты сортировки обеспечивает ORDER BY ?
- Для чего нужны вложенные запросы ?
- В каких конструкциях оператора SELECT они могут использоваться ?
- Можно ли из подзапроса обращаться к полям внешнего запроса ?
- Возможно ли отбор заданного количества записей в запросе ?
- В чем особенности обработки значений NULL ?

# Схема БД



# Контрольная работа #1

- Выбрать все предметы изучаемые группой '4081/1', кроме тех которые зависят от изучения предмета 'Математика';
- Найти группу, в которой староста имеет наилучшую успеваемость в 3-м семестре;
- Найти предмет по которому наилучшая успеваемость у группы '4081/1';
- Найти специальность, для которой наибольшее число студентов имеет средний балл больше 3.5;

Время выполнения 15 минут