



# УПРАВЛЕНИЕ ДАННЫМИ 2023



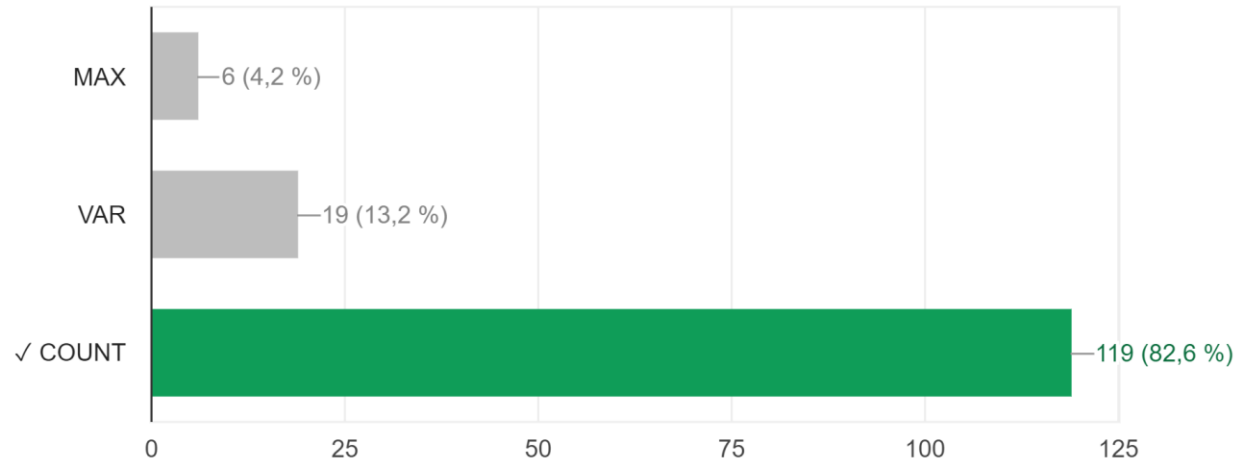
# ЛЕКЦИЯ 5

Физическое моделирование реляционных баз данных. Базовая семантика языка SQL.



Какая из агрегатных функций языка SQL учитывает неопределенное значение?

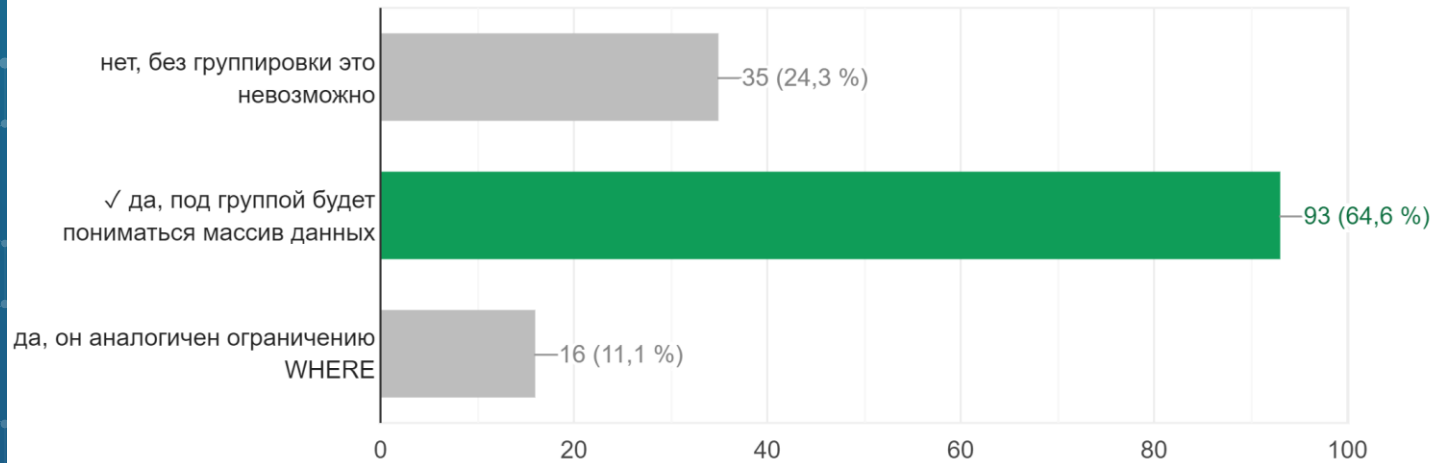
Верных ответов: 119 из 144





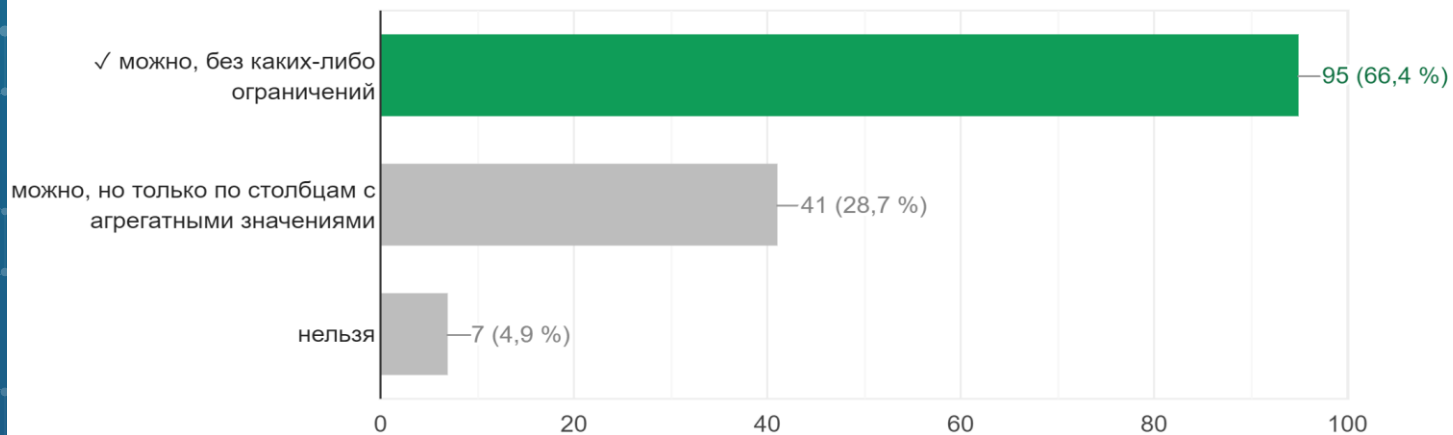
## Актуально ли использование ограничения HAVING без группировки?

Верных ответов: 93 из 144



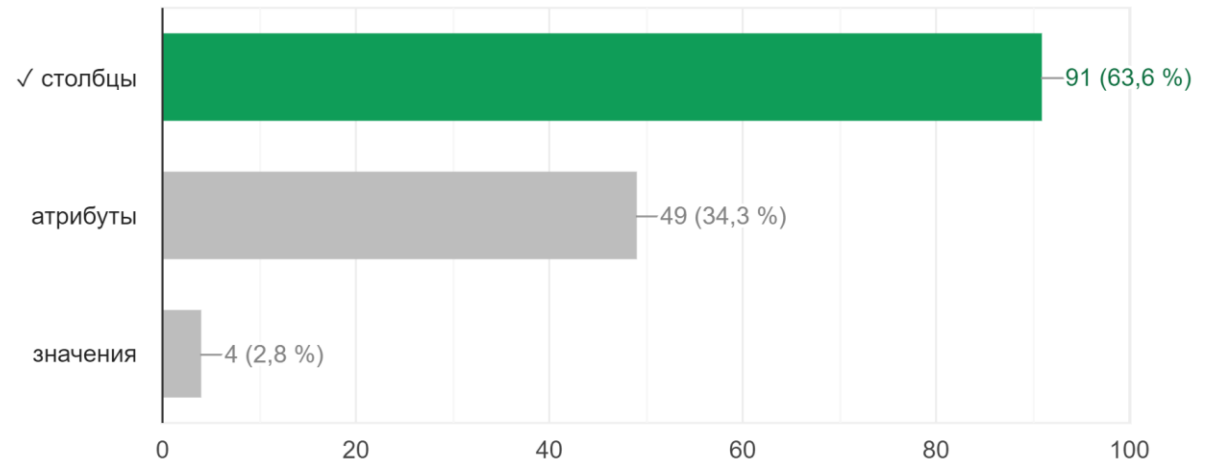
## Можно ли сгруппировать результат запроса по сформированным в этом запросе временным столбцам?

Верных ответов: 95 из 143



## Свойства сущности в физической модели называются...

Верных ответов: 90 из 143





<b>Концепт</b>	<b>Логика</b>	<b>Физика</b>
Сущность	Отношение	Таблица
Свойство	Атрибут	Столбец
Экземпляр	Кортеж	Строка
UID	Ключ	Ограничение ключа
Связь	Связь	Связь

Бизнес-  
требования  
заказчика

Исследование  
предметной  
области

Концептуальные  
модели

Логическая  
модель

Физическая  
модель

Постреляционная БД

Реляционная БД  
Хранилище данных

MS SQL Server  
IBM DB/2  
ORACLE 12c

## СОЗДАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ



# ФИЗИЧЕСКАЯ МОДЕЛЬ БД

- ◆ это база данных, развернутая на сервере;
- ◆ у нее своя собственная терминология;
- ◆ ее можно увидеть в результате процедуры обратного инжиниринга в виде схемы в одной из общепринятых нотаций;
- ◆ это последний этап создания реляционной базы данных (да и впрочем, любой другой).

# ЭЛЕМЕНТЫ ФИЗИЧЕСКОЙ МОДЕЛИ

- ◆ **Представление** - объект базы данных, который хранит в себе запрос SELECT и в случае обращения к данному объекту будет возвращен результирующий набор данных, который формирует запрос, указанный в определении представления.
- ◆ **Хранимая процедура** это организованная группа из одного или нескольких операторов SQL. Содержит программные инструкции, выполняющие операции в БД.
- ◆ **Триггер** - особая разновидность хранимой процедуры, которая автоматически выполняется при возникновении события на сервере базы данных.

# ЭЛЕМЕНТЫ ФИЗИЧЕСКОЙ МОДЕЛИ

- ◆ **Функция** - подпрограмма SQL, которая принимает параметры, выполняет действия, такие как сложные вычисления, а затем возвращает результат этих действий в виде значения.
- ◆ **Индекс** – инструмент управления производительностью запросов.
- ◆ **Временная таблица** – свойство базы данных передавать промежуточную информацию о данных, хранимых в таблице в любой момент времени (отслеживание изменений), # или ##.
- ◆ **Локальная и глобальная переменная**, @ или @@.

# SQL, ИНСТРУМЕНТ СОЗДАНИЯ ФИЗИЧЕСКОЙ МОДЕЛИ ДАННЫХ

- ◆ Инструкции определения данных (Data Definition Language, DDL).
- ◆ Инструкции манипуляции данными (Data Manipulation Language, DML).
- ◆ Инструкции определения доступа к данным (Data Control Language, DCL).
- ◆ Инструкции управления транзакциями (Transaction Control Language, TCL).

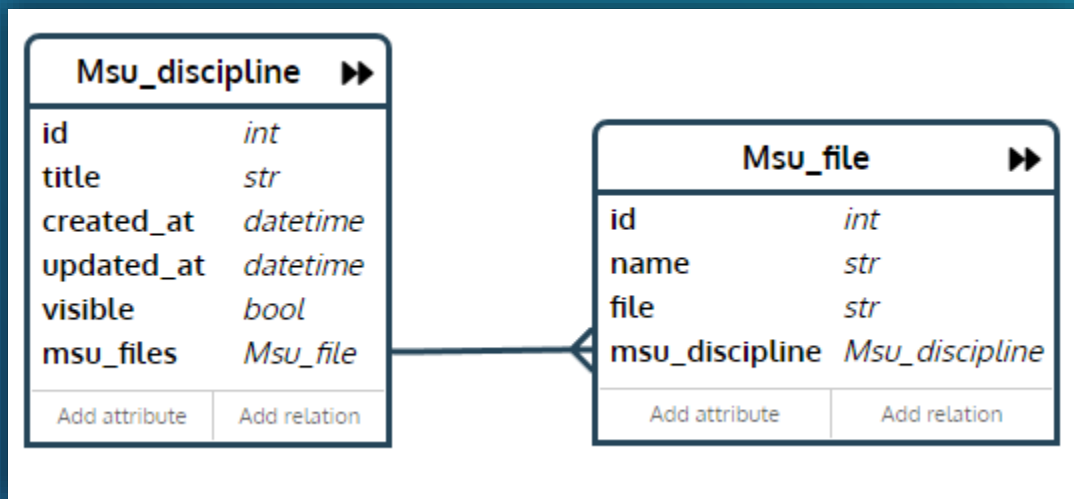
# SQL, ИНСТРУКЦИИ ОПРЕДЕЛЕНИЯ ДАННЫХ

Группа CREATE  
CREATE database  
CREATE table  
CREATE view  
CREATE index  
CREATE trigger  
CREATE procedure  
CREATE function

Группа ALTER  
ALTER database  
ALTER table  
ALTER view  
ALTER index  
ALTER trigger  
ALTER procedure  
ALTER function

Группа DROP  
DROP database  
DROP table  
DROP view  
DROP index  
DROP trigger  
DROP procedure  
DROP function

# ФРАГМЕНТ ФИЗИЧЕСКОЙ РЕЛЯЦИОННОЙ МОДЕЛИ





# СКРИПТ СОЗДАНИЯ МОДЕЛИ ДЛЯ SQLITE

```
CREATE TABLE "Msu_discipline" (  
  "id" INTEGER PRIMARY KEY AUTOINCREMENT,  
  "title" VARCHAR(40) NOT NULL,  
  "created_at" DATETIME NOT NULL,  
  "updated_at" DATETIME,  
  "visible" BOOLEAN NOT NULL  
);  
  
CREATE TABLE "Msu_file" (  
  "id" INTEGER PRIMARY KEY AUTOINCREMENT,  
  "name" VARCHAR(30) NOT NULL,  
  "file" VARCHAR(60) NOT NULL,  
  "msu_discipline" INTEGER NOT NULL REFERENCES "Msu_discipline" ("id") ON DELETE CASCADE  
);  
  
CREATE INDEX "idx_msu_file_msu_discipline" ON "Msu_file" ("msu_discipline")
```

# СКРИПТ СОЗДАНИЯ МОДЕЛИ ДЛЯ POSTGRESQL

```
CREATE TABLE "msu_discipline" (  
  "id" SERIAL PRIMARY KEY,  
  "title" VARCHAR(40) NOT NULL,  
  "created_at" TIMESTAMP NOT NULL,  
  "updated_at" TIMESTAMP,  
  "visible" BOOLEAN NOT NULL  
);  
  
CREATE TABLE "msu_file" (  
  "id" SERIAL PRIMARY KEY,  
  "name" VARCHAR(30) NOT NULL,  
  "file" VARCHAR(60) NOT NULL,  
  "msu_discipline" INTEGER NOT NULL  
);  
  
CREATE INDEX "idx_msu_file__msu_discipline" ON "msu_file" ("msu_discipline");  
  
ALTER TABLE "msu_file" ADD CONSTRAINT "fk_msu_file__msu_discipline" FOREIGN KEY ("msu_discipline") REFERENCES "msu_discipline" ("id") ON DELETE CASCADE
```



# SQL, ИНСТРУКЦИИ МАНИПУЛЯЦИИ ДАННЫМИ

```
SELECT [DISTINCT] target-list  
FROM relation-list  
WHERE qualification
```

relation-list - список имен таблиц.

target-list - список столбцов таблиц, которые указаны в relation-list.

Qualification - условие сравнения (Таблица-Значение или Таблица1-Таблица2, по условию  $<$ ,  $>$ ,  $=$ ,  $\leq$ ,  $\geq$ , комбинированные с AND, OR, NOT).

# СЕМАНТИКА SQL ЗАПРОСА

- ◆ Рассчитать перечисленные таблицы.
- ◆ Исключить из результата строки, которые не попадают в условия Qualification.
- ◆ Исключить столбцы, отсутствующие в target-list.
- ◆ Если имеется группировка или иная сортировка данных, провести ее на последнем этапе.

# ФИЗИЧЕСКИЙ СЛОВАРЬ ДАННЫХ

AdventureWorks

Data Dictionary

2017-05-30

# САМОСТОЯТЕЛЬНОЕ ЗАДАНИЕ (ЛЕКЦИЯ НА САМОСТОЯТЕЛЬНОЕ ИЗУЧЕНИЕ)

- ◆ **Видеозапись лекции по адресу:**  
[https://youtube.com/live/9JgbQ\\_u4rrw](https://youtube.com/live/9JgbQ_u4rrw).
- ◆ Системы баз данных (полный курс), стр. 203-218, 317-344.

# САМОСТОЯТЕЛЬНОЕ ЗАДАНИЕ (ТЕСТЫ)

- ◆ Выполнить тест 5, предварительно самостоятельно изучив материал лекции (см. предыдущий слайд). Дедлайн – 8 ноября.
- ◆ Google Classroom для групп БФБО-01,02,03,04,05: jgc7j5b;
- ◆ Google Classroom для групп БСБО-04,05,06,07,08,09: хакуутр;
- ◆ Google Classroom для групп БСБО-10,11,12: qafm35i;
- ◆ Google Classroom для групп БСБО-01,02,03,13,14: 3idkjkxk.
- ◆ Обратите внимание на то, чтобы в Google Classroom ваш псевдоним содержал вашу фамилию кириллицей (в крайнем случае - латиницей).

# СПАСИБО!

ВАШИ ВОПРОСЫ,  
ПОЖАЛУЙСТА?

