

# SQL. Вводный курс

Лекция №1: Базы данных

# Что такое БД?

В литературе предлагается множество определений понятия «база данных», отражающих скорее субъективное мнение тех или иных авторов, однако общепризнанная единая формулировка отсутствует.

Определения из международных стандартов:

**База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

**База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения.

**База данных** — организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

**База данных** — некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия.

**База данных** — совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

# Виды баз данных

- Иерархические базы данных
- Сетевые базы данных
- Реляционные базы данных
- NoSQL

# Иерархические базы данных

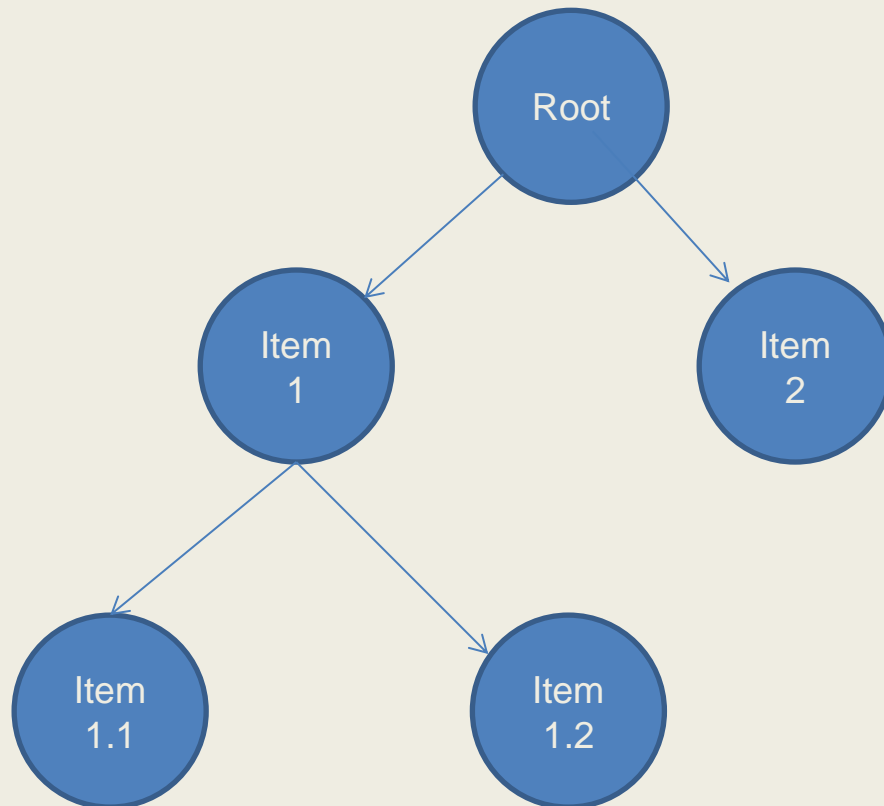
**Иерархические базы данных** поддерживают древовидную организацию информации. Связи между записями выражаются в виде отношений предок/потомок, а у каждой записи есть ровно одна родительская запись. Это помогает поддерживать ссылочную целостность. Когда запись удаляется из дерева, все ее потомки также должны быть удалены. Примером иерархической базы данных может быть, документ в формате XML или файловая система компьютера.

**Базы данных подобного вида оптимизированы для чтения информации**, то есть, базы данных, имеющие иерархическую структуру умеют очень быстро выбирать запрашиваемую информацию и отдавать ее пользователям.

Пример – реестр Windows.

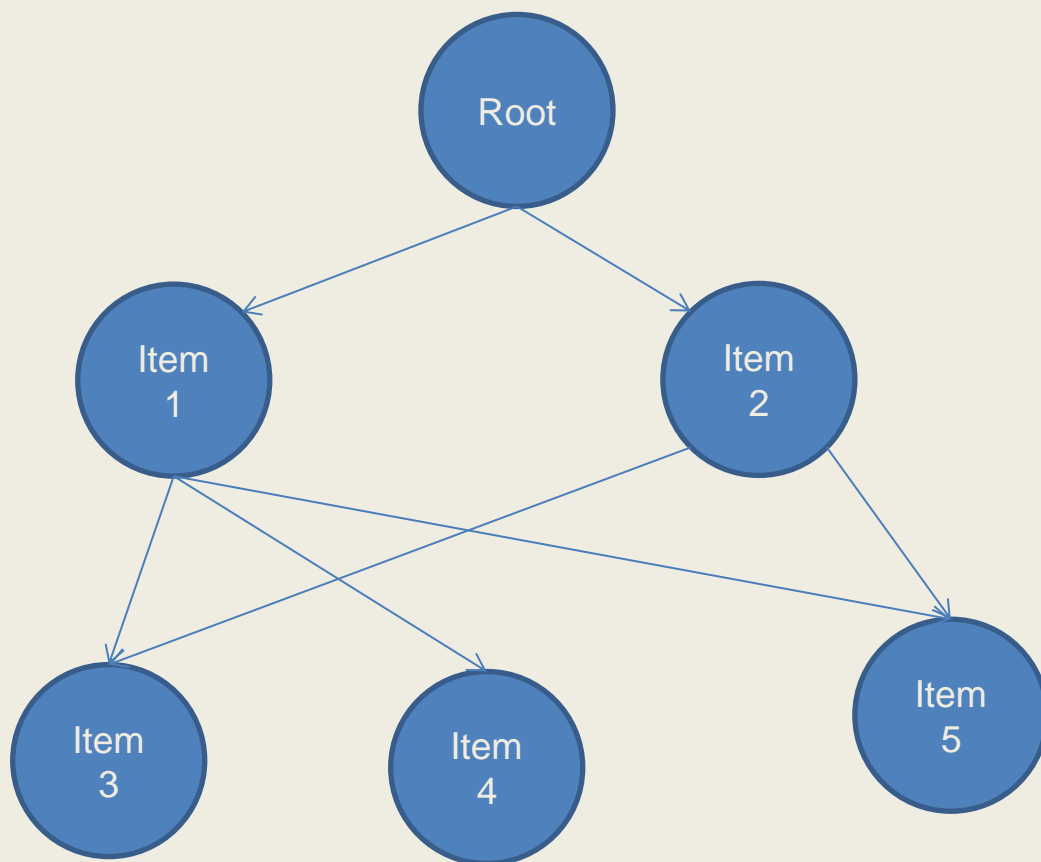
# Иерархические базы данных (продолжение)

Пример



# Сетевые базы данных

Являются расширением иерархической модели. Допускают множественность родителей узла.



# Реляционные базы данных

**Реляционная база данных** — база данных, основанная на реляционной модели данных. Понятие «реляционный» происходит от английского слова *relation* («отношение», «зависимость», «связь»).

Математическим фундаментом реляционных баз данных является раздел математики **Реляционная алгебра**.

Принципы реляционной модели были сформулированы в 1969—1970 годах Э. Ф. Коддом (E. F. Codd). Идеи Кодда были впервые публично изложены в статье «A Relational Model of Data for Large Shared Data Banks». Строгое изложение теории реляционных баз данных (реляционной модели данных) в современном понимании можно найти в книге К. Дж. Дейта. «C. J. Date. An Introduction to Database Systems» («Дейт, К. Дж. Введение в системы баз данных»).

# Реляционные базы данных

Термин «реляционный» означает, что реляционная модель данных основана на математическом понятии отношения (*relation*). В качестве неформального синонима термину «отношение» часто встречается слово таблица. Необходимо помнить, что «таблица» есть понятие нестрогое и неформальное и часто означает не «отношение» как абстрактное понятие, а визуальное представление отношения на бумаге или экране. Некорректное и нестрогое использование термина «таблица» вместо термина «отношение» нередко приводит к недопониманию. Наиболее частая ошибка состоит в рассуждениях о том, что РМД имеет дело с «плоскими», или «двумерными» таблицами, тогда как таковыми могут быть только визуальные представления таблиц. Отношения же являются абстракциями и не могут быть ни «плоскими», ни «неплоскими».



# Реляционные базы данных

Для лучшего понимания **Реляционной Модели Данных** следует отметить три важных обстоятельства:

- модель является логической, то есть отношения являются логическими (абстрактными), а не физическими (хранимыми) структурами;
- для реляционных баз данных верен информационный принцип: всё информационное наполнение базы данных представлено одним и только одним способом, а именно — явным заданием значений атрибутов в кортежах отношений; в частности, нет никаких указателей (адресов), связывающих одно значение с другим;
- наличие реляционной алгебры позволяет реализовать декларативное программирование и декларативное описание ограничений целостности, в дополнение к навигационному (процедурному) программированию и процедурной проверке условий.

# Реляционные базы данных

Пример: Отношение (Relation) **Stutents**

FirstName	LastName	University	Course
John	Smith	Stanford	2
Peter	Storozhenko	KNU	5

Синонимы: Таблица (table), Entity, Table

# Реляционные базы данных

## Атрибуты (attributes)

FirstName	LastName	University	Course
John	Smith	Stanford	2
Peter	Storozhenko	KNU	5

Атрибуты выполняют функцию именования столбцов отношения и содержательно описывают назначение данных содержащихся в соответствующих ячейках.

С каждым атрибутом сопоставлен тип данных

Синонимы: Столбец, Колонка, Field

# Реляционные базы данных

## Кортежи (tuples)

FirstName	LastName	University	Course
John	Smith	Stanford	2
Peter	Storozhenko	KNU	5

Строки отношений, отличные от той, которая представляет наименование атрибутов называют кортежами. Кортеж содержит по одному значению для каждого атрибута входящего в кортеж.

Синонимы: строка, запись, row, record

# Связи между таблицами

Между таблицами могут быть связи.

Существование связей между таблицами обеспечивается наличие так называемых первичных и вторичных ключей.

Первичный ключ (**Primary key**) свойство атрибута (колонки) - в таблице только один, может быть составным

Что касается вторичных (**Foreign keys**), то их в таблице может присутствовать несколько.

Основная таблица

PK	FK1	...	FK2
7	1		C
13	3		C
19	1		A
22	2		B
27	1		A
56	1		B

PK	Название
A	Шкаф
B	Диван
C	Стол

Таблицы  
деталей

PK	Название
1	Лампа
2	Светильник
3	Торшер

# Связи между таблицами (Пример)

Студенты

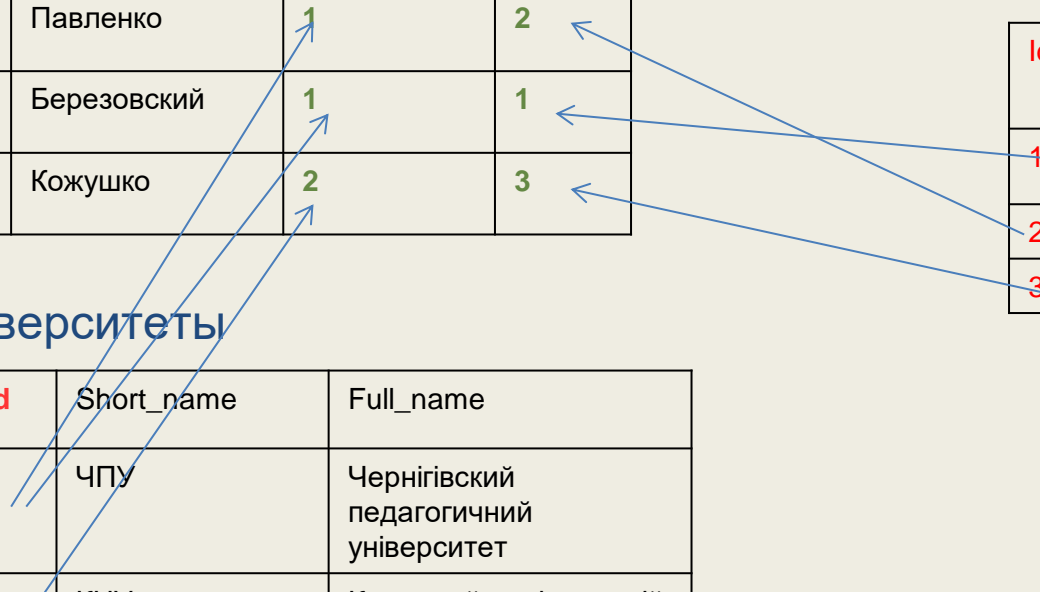
Id	Name	University	City
1	Павленко	1	2
2	Березовский	1	1
3	Кожушко	2	3

Города

Id	Name
1	Львів
2	Чернівці
3	Київ

Університети

Id	Short_name	Full_name
1	ЧПУ	Чернігівський педагогічний університет
2	КНУ	Київський національний університет
3	КПІ	Київський політехнічний університет



# Свойства реляционной модели данных

Свойства таблиц реляционной базы данных:

- в таблицах не должно быть записей-дубликатов, следовательно:
  - в таблице всегда имеется поле или их набор, позволяющих однозначно идентифицировать запись, иначе говоря, в таблице всегда есть первичный ключ;
- записи таблицы не упорядочены, следовательно, в таблице не существует:
  - иного способа адресации записей, кроме адресации по ключу;
  - первой записи, последней, предыдущей, следующей и т.д.
- столбцы таблицы также не упорядочены, следовательно:
  - единственным способом их адресации в запросах является использование наименования поля
  - их «порядок», задается в момент создания таблицы

# Что такое SQL и для чего он нужен?

**S** STRUCTURED

**Q** QUERY

**L** LANGUAGE

← "язык структурированных запросов"

SQL - это универсальное средство – **формальный язык** - для общения с различными видами Баз Данных



ХОЧУ...

Создать

Узнать

Изменить



# SQL

Был создан в начале 1970-х годов в фирме IBM в рамках проекта System R. Первоначально назывался **SEQUEL** (расшифровывалась как *Structured English QUery Language* — «структурированный английский язык запросов»). Позже был переименован в **SQL**.

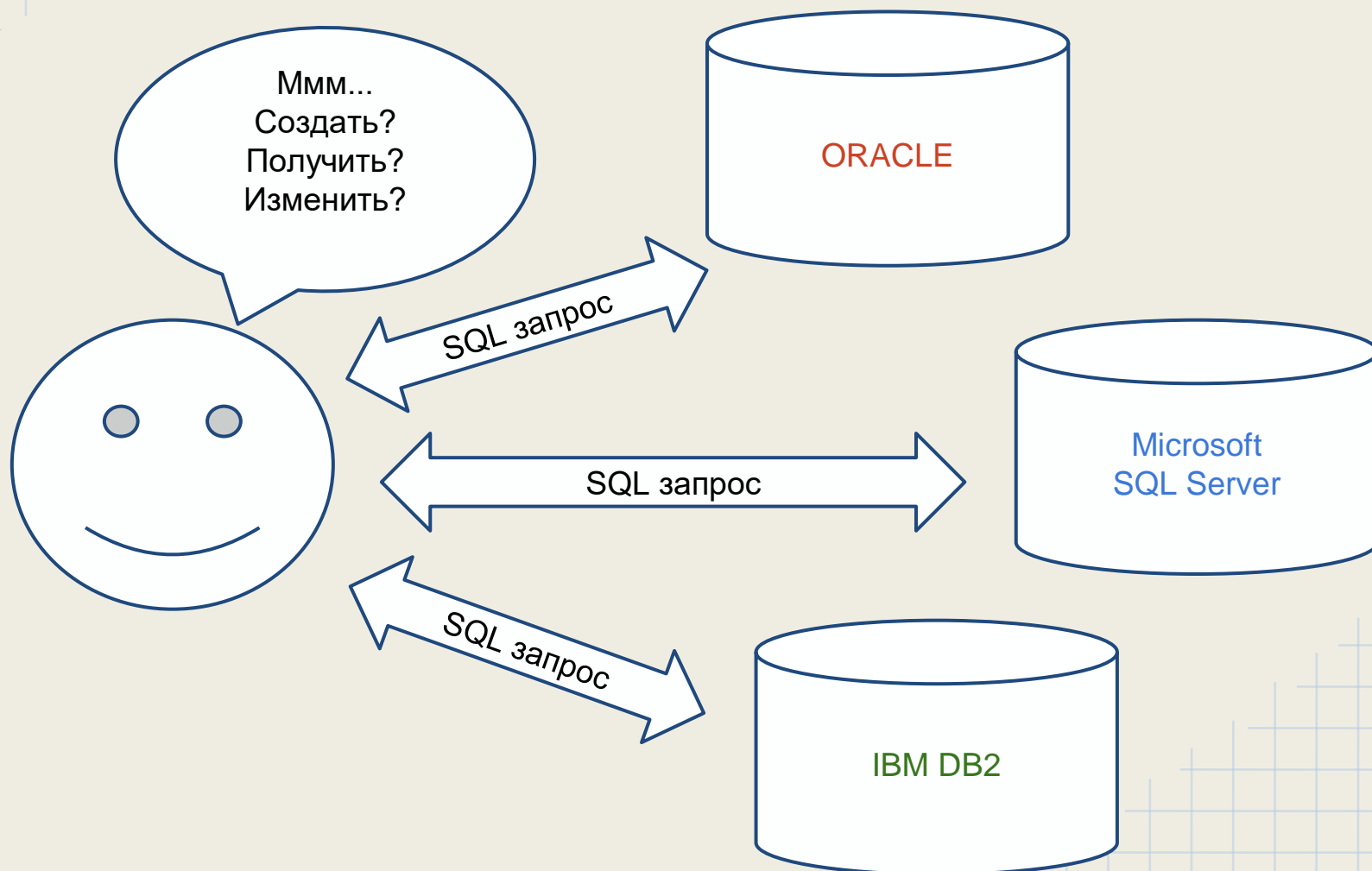
**SQL** является декларативным языком программирования. При помощи SQL задается **спецификация** решения, то есть описывается, **что** представляет собой проблема и ожидаемый результат.

Противоположностью декларативных языков программирования являются императивные языки программирования, описывающие на том или ином уровне детализации, **как** решить задачу и представить результат. В общем и целом, декларативное программирование идёт от человека к машине, тогда как императивное — от машины к человеку. Как следствие, декларативные программы не используют понятия состояния, то есть не содержат переменных и операторов присваивания.

## Стандарты

Год	Название	Иное название	Изменения
1986	SQL-86	SQL-87	Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987 году.
1989	SQL-89	<a href="#">FIPS 127-1</a>	Немного доработанный вариант предыдущего стандарта.
<a href="#">1992</a>	<a href="#">SQL-92</a>	SQL2, FIPS 127-2	Значительные изменения (ISO 9075)
<a href="#">1999</a>	<a href="#">SQL:1999</a>	SQL3	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, нескаллярные типы данных и некоторые объектно-ориентированные возможности.
<a href="#">2003</a>	<a href="#">SQL:2003</a>		Введены расширения для работы с XML-данными, оконные функции (применяемые для работы с OLAP-базами данных), генераторы последовательностей и основанные на них типы данных.
<a href="#">2006</a>	<a href="#">SQL:2006</a>		Функциональность работы с XML-данными значительно расширена. Появилась возможность совместно использовать в запросах SQL и XQuery.
<a href="#">2008</a>	<a href="#">SQL:2008</a>		Улучшены возможности оконных функций, устранены некоторые неоднозначности стандарта SQL:2003

# Универсальный язык (стандарт)



# SQL

SQL можно разделить на две части:

- **DDL – data definition language** (язык определения данных).  
Используется для определения, создания, редактирования и удаления объектов баз данных (таблиц, ограничений, представлений ....)

```
create <object>...
```

```
alter <object>...
```

```
drop <object>...
```

- **DML – data manipulation language** (язык манипулирования данными).  
Используется для добавления, удаления, модификации данных

```
select
```

```
insert
```

```
update
```

```
delete
```

# Типы данных

**Тип данных** (*тип*) — множество значений и операций на этих значениях.

Другие определения:

- Тип данных — класс данных, характеризуемый членами класса и операциями, которые могут быть к ним применены.
- Тип данных — категоризация абстрактного множества возможных значений, характеристик и набор операций для некоторого атрибута.
- Тип данных — категоризация аргументов операций над значениями, как правило, охватывающая как поведение, так и представление.
- Тип данных — допустимое множество значений.

Тип определяет возможные значения и их смысл, операции, а также способы хранения значений типа. Неотъемлемой частью большинства языков программирования являются системы типов, использующие типы для обеспечения той или иной степени типобезопасности.

[https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)

# Типы данных SQL

## Текстовые

- **CHARACTER(длина)** - последовательность символов, *длина* описывает неизменное количество символов строки
- **CHAR(длина)** - сокращенная форма
- **VARCHAR(длина)** - отступление от стандарта, строка переменной длины, но не более указанной

Все текстовые данные заключаются в одинарные кавычки.

Пример:

```
'QA Light!!! Курс введение в SQL.' as CHARACTER(32)
'QA Light!!!' as CHARACTER(32)
'QA Light!!!' as VARCHAR(32)
'QA Light!!! Общение с БД на ее языке!' as CHAR(37)
'QA Light!!! Общение с БД на ее я' as VARCHAR(32)
```

# Типы данных SQL

## Числовые

- **INTEGER (INT)** - целочисленный, диапазон значений от  $-2^{31}$  до  $+2^{31}$
- **SMALLINT** - целочисленный, диапазон значений от  $-2^{15}$  до  $+2^{15}$
- **DECIMAL**(точность[,масштаб]) - десятичное число с фиксированной точкой, где точность - общее количество значащих цифр, масштаб - максимальное количество цифр после запятой
- **NUMERIC**(точность[,масштаб])
- **FLOAT**(точность) - число с плавающей точкой и указанной минимальной точностью
- **REAL** - число с плавающей точкой, зависящее от конкретной реализации SQL

# Типы данных SQL

## Дата-время

Чаще всего записывается как **DATE** или **DATETIME**.  
Также отдельный тип для времени **TIME**.

Значения записываются, как строковые, в одинарных кавычках в соответствии с форматом, определенным в системе или клиентском приложении:

- '2012-sep-14'
- '2012.09.14'
- '2012/09/14'
- '2012/09/14 12:05:44.123'
- '2012/09/14 00:00:00.000'



# Типы данных SQL

Неопределенные или пропущенные данные определяются как *NULL*

Такое поле нельзя проверить на равенство

***~~X = NULL~~***

в таких случаях необходимо использовать

***X IS NULL***

***X IS NOT NULL***

В агрегирующих функциях значение такого поля игнорируется

В случае с NULL условные операторы, результатом которых может быть true/false, расширяются еще одним значением unknown

<https://sql-language.ru/sqldatatype.html>

# Литература:

<http://www.sqlservercentral.com/>

<http://www.sql.ru/>

<https://www.codeproject.com>

Подборка материалов для самостоятельного изучения SQL

<https://proglib.io/p/sql-digest/>

Хорошая книга для самостоятельного обучения

*Майкл Дж. Хернандес*

*Джон Л. Вьюкас*

**SQL-запросы для простых смертных**