



SQL Server Developer

Проектирование БД



Проверить, идет ли запись

Меня хорошо видно && слышно?



Ставим "+", если все хорошо
"-", если есть проблемы

Проектирование БД

- О выпускном проекте
- Этапы проектирования
- Нормализация



Активно
участвуем



Off-topic обсуждаем
в учебной группе
telegram



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

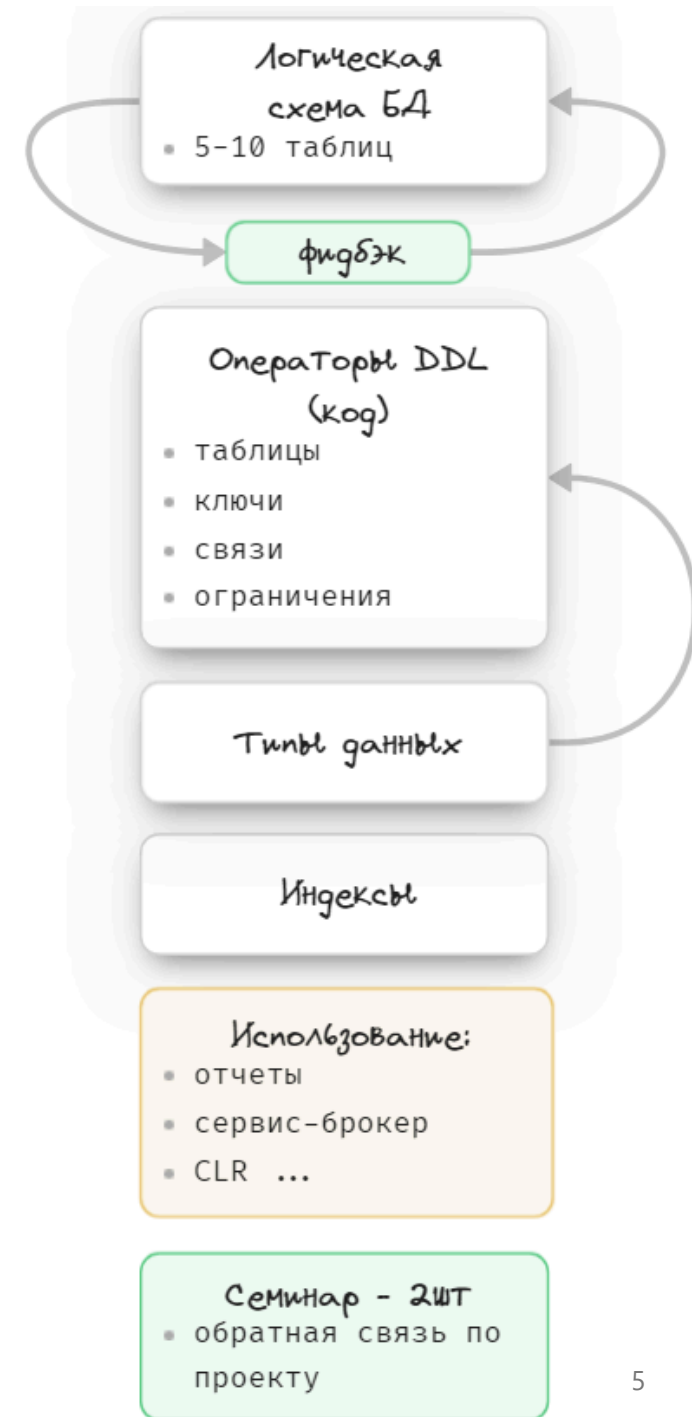
Выбор темы проекта

тема проекта пойдет в сертификат о прохождении курса

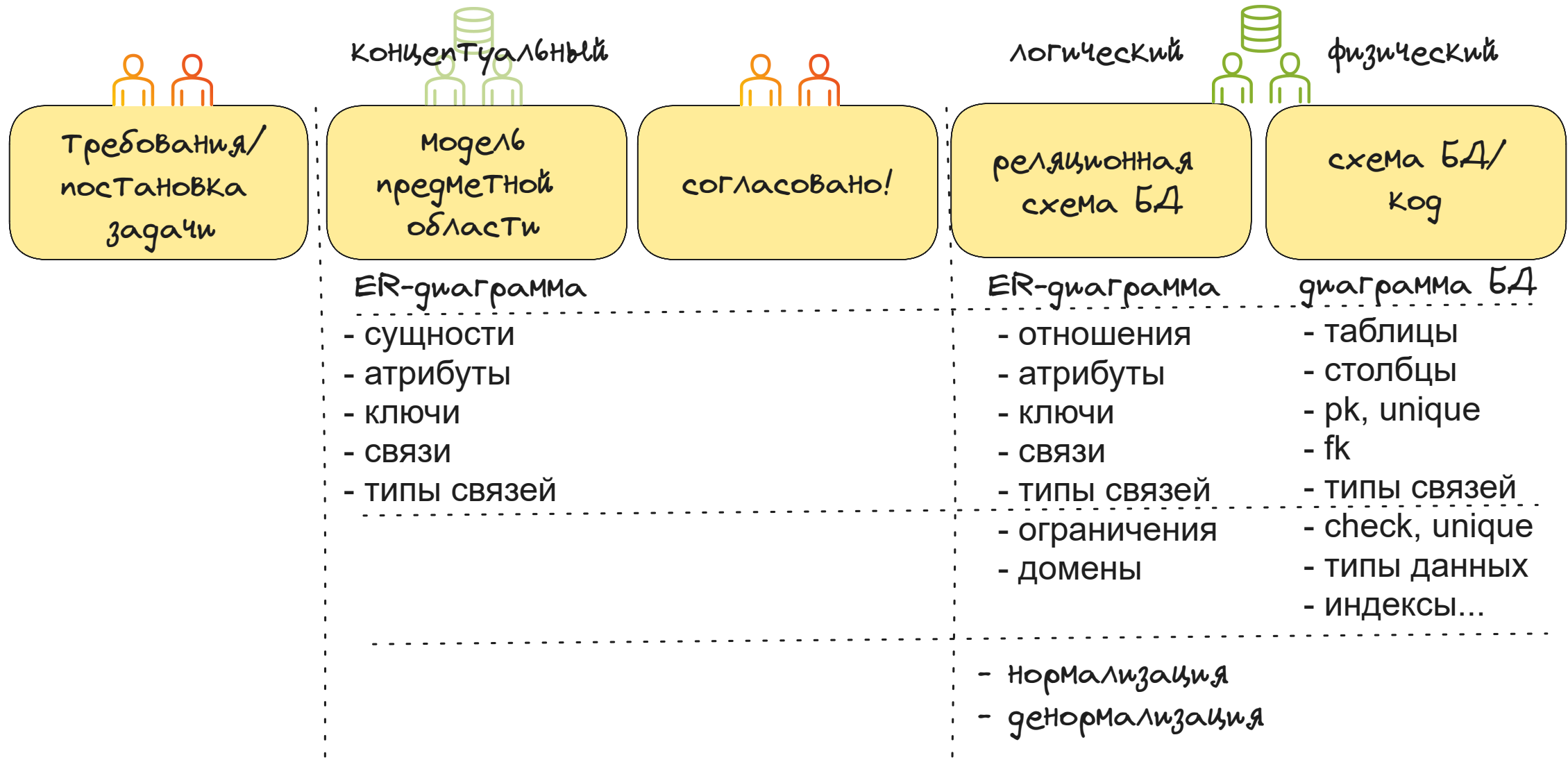
- предметная область, которую вы хорошо знаете
 - домашние финансы
 - библиотека
 - загрузка данных по API...
 - кусочек рабочего проекта (было-стало)
 - темы из материалов к вебинару
 - ...
- 5-10 таблиц

О выпускном проекте

- все ДЗ в этом модуле - работа над проектом вашей БД
- в конце курса - семинар по вашим проектам БД
 - презентация или просто схема/диаграмма БД с объяснением интересных моментов
 - если не попадаете на семинар - можно сдать проект как обычное ДЗ:
 - постановка задачи
 - схема БД (в виде диаграммы из SQL Server)
 - заполнение таблиц данными: генерация см. [ОУ - Наполнение БД тестовыми данными](#)
 - планируемое использование: запросы/отчеты/другие интересные моменты



Проектирование БД - этапы



Постановка задачи

- знакомимся с предметной областью + определяемся, что будем хранить в БД
- кратко отвечаем на вопросы:
 - что сейчас есть и что не устраивает
 - что хотим получить
 - много ли строк
 - характер нагрузки
- Университет - автоматизация учета информации "с нуля"
- на выходе хотим получить отчеты:
 - о курсах, которые слушают студенты
 - о преподавателях, читающих курсы лекций
 - о научных руководителях студентов
 - < 1 млн строк в любой из таблиц
 - ввод/редактирование данных пользователями + отчеты

Концептуальное проектирование

этот этап часто пропускается

- Модель данных в терминах пользователей (модель предметной области)
- Определяем: сущности (атрибуты), связи между сущностями (типы связей), ключи
- Результат: ER-Диаграмма (Entity-Relationship , сущность - связь)
 - ПО: MS Visio, BPWin/ERWIN...
 - сайты: [ERDPlus](#) , [Lucidchart](#), [draw.io \(diagrams.net\)](#)...
 - Excel / текстовый редактор
 - листок бумаги и ручка

Сущности ("будущие таблицы")

отчеты (из постановки задачи):

- о курсах, которые слушают студенты
- о преподавателях, читающих курсы лекций
- о научных руководителях студентов

- сущности - ?

Сущности и атрибуты

отчеты (из постановки задачи):

- о курсах, которые слушают студенты
- о преподавателях, читающих курсы лекций
- о научных руководителях студентов

- сущности

- студенты (студенческий билет, ФИО, контактные данные)
- преподаватели/научные руководители (ФИО)
- курсы лекций (код курса, наименование)

Ключи

- ключ однозначно идентифицирует строку в таблице
- может состоять из 1 или нескольких столбцов
- ключи - ?

№ студенческого	ФИО	Email
1022	Пушкин Александр Сергеевич	pushkin@mail.ru
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru
4124	Некрасов Николай	nekrasov@mail.ru

Фамилия	Имя	Отчество	Дата рождения
Пушкин	Александр	Сергеевич	06.06.1799
Лермонтов	Михаил	Юрьевич	15.10.1814
Некрасов	Николай	Алексеевич	10.12.1821

Ключи

- студенты (студенческий билет, ФИО, контактные данные)
- преподаватели/научные руководители (ФИО)
- курсы лекций (код курса, наименование)

СВЯЗИ

- связь - глагол, описывающий как взаимодействуют сущности

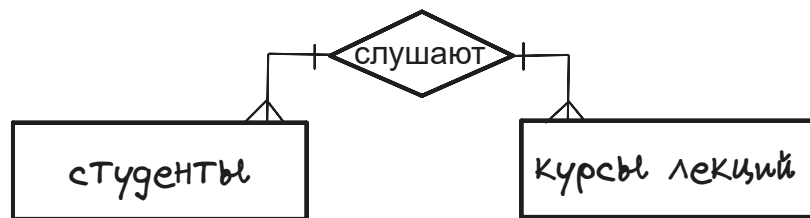
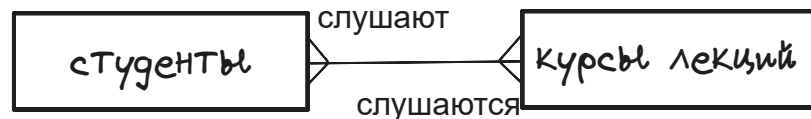
отчеты (из постановки задачи):

- о курсах, которые слушают студенты
- о преподавателях, читающих курсы лекций
- о научных руководителях студентов

- студенты **слушают** курсы
- преподаватели **читают** курсы лекций
- научные руководители **руководят работой** студентов

Типы связей

- 1 : 1 "один-к-одному" - часто оформляют как атрибут
 - 1 : N "один-ко-многим"
 - M : N "многие-ко-многим" - всегда через доп. таблицу
-
- студенты слушают курсы: многие - ко многим (M: N)
 - преподаватели читают курсы: многие - ко многим (M : N)
 - научные руководители руководят работой студентов: один - ко многим (1 : N)
 - студенты имеют студенческие билеты: один - ко одному (1 : 1)



ER-диаграмма (пример)

- разные нотации
 - crow foot (воронья лапка)



+	ОДИН
—	МНОГО
+=	ОДИН (И ТОЛЬКО ОДИН)
○+	НОЛЬ ИЛИ ОДИН
— K	ОДИН ИЛИ МНОГО
○— K	НОЛЬ ИЛИ МНОГО

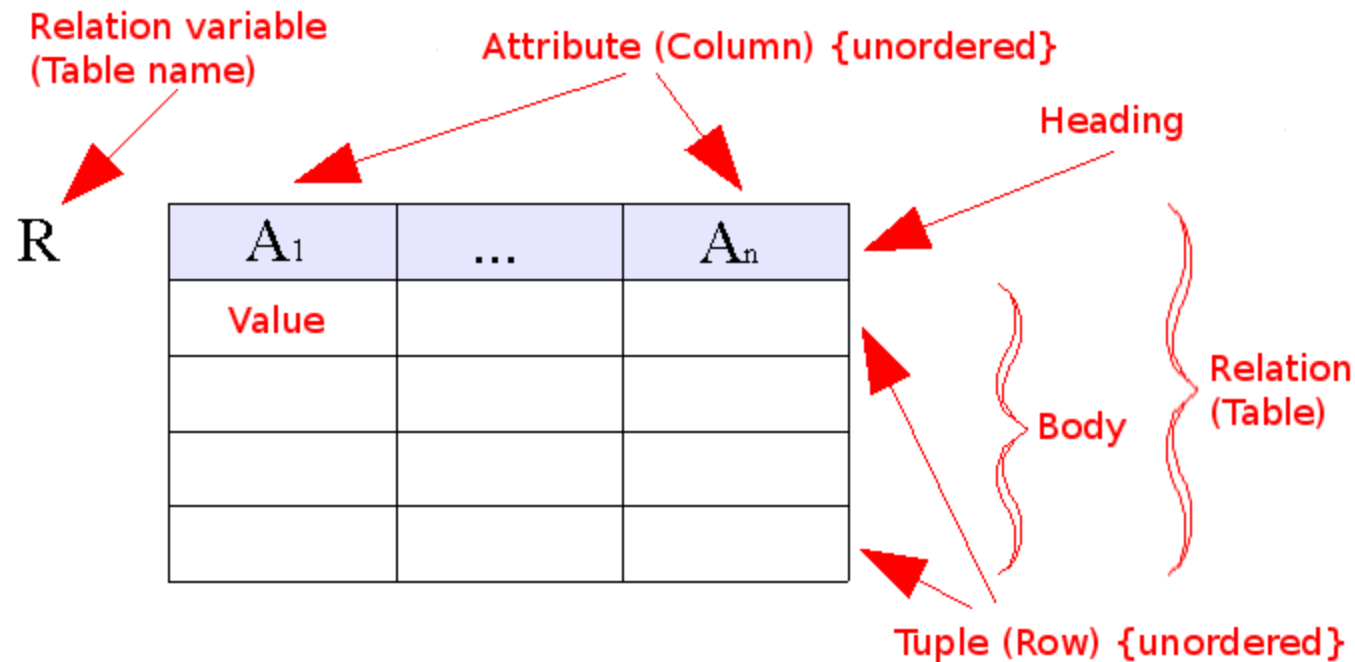
Логическое проектирование

обычно начинаем с этого этапа

- Описание модели данных в терминах **реляционной теории** (но без кода)
- Доработка концептуальной модели (типы данных, ключи, связи, ограничения)
- Нормализация
- Результат: **Реляционная диаграмма** (**Реляционная схема БД**)
 - ПО (есть trial): [DBeaver PRO](#) , [dbForge Studio](#)
 - VS Code (free): [bigER Modeling Tool](#)
 - VS (free): [ERD Editor](#)
 - сайты: [quickdatabasediagrams](#), [dbdiagram.io](#)

Реляционная БД / Реляционная теория

- Эдгар Кодд (1970 - первая работа по реляционной теории данных)
- Реляционная БД - реляционная (табличная) модель хранения данных
- Relation - отношение атрибутов (столбцов)
- Relationship - связи между таблицами



Ключи и связи

что должно быть в "хорошо" спроектированной БД

- Primary Key (PK) - для однозначной идентификации строки в течение всего времени жизни таблицы => Surrogate Key
- Unique Key (Natural Key) - там где возможно

StudentId	Фамилия	Имя	Отчество	Дата рождения
1	Пушкин	Александр	Сергеевич	06.06.1799
2	Лермонтов	Михаил	Юрьевич	15.10.1814
3	Некрасов	Николай	Алексеевич	10.12.1821

- Foreign Key (FK) - поле-ссылка на Ключ строки из связанной таблицы.
 - желательно определить для связанных таблиц
 - желательно использовать одинаковые имена для референсных полей в обеих таблицах

Концептуальная модель -> Реляционная схема БД

сущности (будущие таблицы), атрибуты и ключи:

- студенты `Students` (`StudentId` , студенческий билет, ФИО, контактные данные)
- преподаватели/научные руководители `Lecturers` (`LecturerId` , ФИО)
- курсы лекций `Courses` (`CourseId` , код курса, наименование)

связи (глаголы), их типы:

- студенты **слушают** курсы: **M: N**
 - `b_Students_Courses` (`StudentId` FK `Students.StudentId`, `CourseId` FK `Courses.CourseId`)
- преподаватели **читают** курсы: **M: N**
 - `b_Lecturers_Courses` (`LecturerId` FK `Lecturers.LecturerId`, `CourseId` FK `Courses.CourseId`)
- науч.руководители **руководят** работой студентов: **1 : N**
 - `Students` (`LecturId` FK `Lecturers.LecturId`)

Другая постановка задачи

- университет - автоматизация учета информации по студентам, курсам лекций и преподавателям
- сейчас учет ведется в Excel, данных много, бывают ошибки
- требуется получить отчет "как в Excel"
- можно поместить все в 1 таблицу, но будет неудобно работать
 - *см. дальше*

No	StudentId	ФИО	Contacts	Adviser	AdvRoom	Course1	Course2	Lecturer1	Lecturer2
1	1022	Пушкин Александр Сергеевич	89001234567; pushkin@mail.ru	Толстой	101	101-07	143-01	Есенин	Блок
2	4123	Лермонтов Михаил Юрьевич	89007654321; lermontov@mail.ru	Блок	102	101-07	143-01	Есенин	Ахматова
3	4124	Некрасов Николай	nekrasov@mail.ru			179-04		Фет	

Аномалии (проблемы)

невозможно сделать что-то с одним объектом, не затронув другой

№	StudentId	ФИО	Contacts	Adviser	AdvRoom	Course1	Course2	Lecturer1	Lecturer2
1	1022	Пушкин Александр Сергеевич	89001234567; pushkin@mail.ru	Толстой	101	101-07	143-01	Есенин	Блок
2	4123	Лермонтов Михаил Юрьевич	89007654321; lermontov@mail.ru	Блок	102	101-07	143-01	Есенин	Ахматова
3	4124	Некрасов Николай	nekrasov@mail.ru			179-04		Фет	

Аномалии:

- вставки - сложность добавления информации (insert Course 3)
- удаления - риск удаления лишней информации
 - удаление Некрасова удалит и инфу по курсу
- обновления – риск неполного обновления
 - заменена преподавателя
 - обновить e-mail, начинающийся на p (like 'p%' не работает)
- Причины:
 - избыточность/дублирование данных
 - сложные зависимости между данными

Нормализация

последовательность шагов, с помощью которых улучшается модель реляционной БД

- Таблица делится на несколько таблиц меньшего размера по определенным правилам (нормальным формам)
- Нормальные формы применяются последовательно
- Каждая следующая НФ включает все требования предыдущих и добавляет новые
- 1НФ, 2НФ, 3НФ, НФ Бойса-Кодда, 4НФ, 5НФ
 - устраняют избыточность данных
 - устраняет сложные зависимости
 - не приводит к потере производительности
- чем больше нормализована БД тем больше джойнов
 - важно вовремя остановиться

Реляционная теория (ОНФ)

- порядок строк не имеет значения (нет нумерации строк)
- порядок столбцов не имеет значения (доступ к столбцам - по названию)
- нет дублей строк

№	StudentId	ФИО	Contacts	Adviser	AdvRoom	Course1	Course2	Lecturer1	Lecturer2
1	1022	Пушкин Александр Сергеевич	89001234567; pushkin@mail.ru	Толстой	101	101-07	143-01	Есенин	Блок
2	4123	Лермонтов Михаил Юрьевич	89007654321; lermontov@mail.ru	Блок	102	101-07	143-01	Есенин	Ахматова
3	4124	Некрасов Николай	nekrasov@mail.ru			179-04		Фет	

StudentId	ФИО	Contacts	Adviser	AdvRoom	Course1	Course2	Lecturer1	Lecturer2
1022	Пушкин Александр Сергеевич	89001234567; pushkin@mail.ru	Толстой	101	101-07	143-01	Есенин	Блок
4123	Лермонтов Михаил Юрьевич	89007654321; lermontov@mail.ru	Блок	102	101-07	143-01	Есенин	Ахматова
4124	Некрасов Николай	nekrasov@mail.ru			179-04		Фет	

- в каждой ячейке содержится неделимое значение
- нет разных столбцов с одинаковым смыслом
- каждый столбец содержит один тип данных

StudentId	ФИО	Contacts	Adviser	AdvRoom	Course1	Course2	Lecturer1	Lecturer2
1022	Пушкин Александр Сергеевич	89001234567; pushkin@mail.ru	Толстой	101	101-07	143-01	Есенин	Блок
4123	Лермонтов Михаил Юрьевич	89007654321; lermontov@mail.ru	Блок	102	101-07	143-01	Есенин	Ахматова
4124	Некрасов Николай	nekrasov@mail.ru			179-04		Фет	

StudentId	ФИО	Email	Phone	Adviser	AdvRoom	Course	Lecturer
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101	101-07	Есенин
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101	143-01	Блок
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102	101-07	Есенин
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102	143-01	Ахматова
4124	Некрасов Николай	nekrasov@mail.ru				179-04	Фет

Адрес - делить или нет?

руководствуемся здравым смыслом и не усложняем себе жизнь

- делим на составные части только если планируем работать с частями адреса

690001, Приморский край, город Владивосток, Светланская ул., д. 104а

- В какой БД будем разбивать адрес на составные части?
 - (1) БД ресторан с доставкой
 - (2) БД университет - паспортные данные студентов

2НФ

- таблица в 1НФ
 - **определен ключ** по которому можно идентифицировать строку (*может состоять из нескольких колонок*)
 - все неключевые столбцы функционально (**напрямую**) зависят от **всего** ключа, а не от его части
- актуально для таблиц с **составным** ключом

StudentId	ФИО	Email	Phone	Adviser	AdvRoom	Course	Lecturer
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101	101-07	Есенин
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101	143-01	Блок
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102	101-07	Есенин
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102	143-01	Ахматова
4124	Некрасов Николай	nekrasov@mail.ru				179-04	Фет

преподаватель зависит от курса, но не зависит от студента (сложность обновления)

2НФ

- таблица в 1НФ
- **определен ключ** по которому можно идентифицировать строку (*может состоять из нескольких колонок*)
- все неключевые столбцы функционально (**напрямую**) зависят от **всего** ключа, а не от его части

StudentId	ФИО	Email	Phone	Adviser	AdvRoom	Course	Lecturer
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101	101-07	Есенин
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101	143-01	Блок
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102	101-07	Есенин
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102	143-01	Ахматова
4124	Некрасов Николай	nekrasov@mail.ru				179-04	Фет

StudentId	ФИО	Email	Phone	Adviser	AdvRoom
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102
4124	Некрасов Николай	nekrasov@mail.ru			

StudentId	Course	Lecturer
1022	101-07	Есенин
1022	143-01	Блок
4123	101-07	Есенин
4123	143-01	Ахматова
4124	179-04	Фет

StudentId	ФИО	Email	Phone	Adviser	AdvRoom
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102
4124	Некрасов Николай	nekrasov@mail.ru			

3НФ

- таблица во 2НФ
- все **неключевые столбцы** зависят только от ключа и не зависят от других неключевых столбцов (транзитивная/промежуточная зависимость)

<u>StudentId</u>	ФИО	Email	Phone	Adviser	AdvRoom
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102
4124	Некрасов Николай	nekrasov@mail.ru			

<u>StudentId</u>	<u>Course</u>	Lecturer
1022	101-07	Есенин
1022	143-01	Блок
4123	101-07	Есенин
4123	143-01	Ахматова
4124	179-04	Фет

в 1й таблице - 2 сущности (студенты и научные руководители)

ЗНФ

- таблица во 2НФ
- все **неключевые столбцы** зависят только от ключа и не зависят от других неключевых столбцов (транзитивная/промежуточная зависимость)

<u>StudentId</u>	ФИО	Email	Phone	Adviser	AdvRoom
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой	101
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок	102
4124	Некрасов Николай	nekrasov@mail.ru			

<u>StudentId</u>	ФИО	Email	Phone	Adviser
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок
4124	Некрасов Николай	nekrasov@mail.ru		

<u>StudentId</u>	<u>Course</u>	Lecturer
1022	101-07	Есенин
1022	143-01	Блок
4123	101-07	Есенин
4123	143-01	Ахматова
4124	179-04	Фет

<u>Adviser</u>	AdvRoom
Толстой	101
Блок	102

можно остановиться: избыточность устранена

НФ Бойса-Кодда (усиленная ЗНФ)

редкий случай, обычно ЗНФ = ЗНФБК

- таблица в ЗНФ
- **неключевые столбцы** зависят **от всего ключа**, а не от его части
актуально: ∃ составные (многостолбцевые) пересекающиеся альтернативные ключи
- лектора можно определить только зная весь ключ: курс и студента
- изменим данные в таблице для нарушения ЗНФБК
 - преподаватель напрямую зависит от курса, но не зависит от студента

<u>StudentId</u>	<u>ФИО</u>	<u>Email</u>	<u>Phone</u>	<u>Adviser</u>
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок
4124	Некрасов Николай	nekrasov@mail.ru		

<u>Adviser</u>	<u>AdvRoom</u>
Толстой	101
Блок	102

<u>StudentId</u>	<u>Course</u>	<u>Lecturer</u>
1022	101-07	Есенин
1022	143-01	Блок
4123	101-07	Есенин
4123	143-01	Ахматова
4124	179-04	Фет

<u>StudentId</u>	<u>Course</u>	<u>Lecturer</u>
1022	101-07	Есенин
1022	143-01	Блок
4123	101-07	Есенин
4123	143-01	Блок
4124	179-04	Фет

НФ Бойса-Кодда (усиленная 3НФ)

- таблица в 3НФ
- **неключевые столбцы** зависят **от всего ключа**, а не от его части
актуально: ∃ составные (многостолбцевые) пересекающиеся альтернативные ключи

<u>StudentId</u>	ФИО	Email	Phone	Adviser
1022	Пушкин Александр Сергеевич	pushkin@mail.ru	89001234567	Толстой
4123	Лермонтов Михаил Юрьевич	lermontov@mail.ru	89007654321	Блок
4124	Некрасов Николай	nekrasov@mail.ru		

<u>Adviser</u>	AdvRoom
Толстой	101
Блок	102

<u>StudentId</u>	<u>Course</u>	<u>Lecturer</u>
1022	101-07	Есенин
1022	143-01	Блок
4123	101-07	Есенин
4123	143-01	Блок
4124	179-04	Фет

<u>StudentId</u>	<u>Lecturer</u>
1022	Есенин
1022	Блок
4123	Есенин
4123	Блок
4124	Фет

<u>Course</u>	<u>Lecturer</u>
101-07	Есенин
143-01	Блок
101-07	Есенин
143-01	Блок
179-04	Фет

<u>Course</u>	<u>Lecturer</u>
101-07	Есенин
143-01	Блок
179-04	Фет

4НФ

- таблица в 3НФБК
- отсутствуют множественные зависимости столбцов друг от друга

множественная зависимость: \exists таблица со столбцами A, B, C:

B и C не связаны между собой, но по отдельности зависят от A:

A \twoheadrightarrow B (одному значению A соответствует несколько B)

A \twoheadrightarrow C (одному значению A соответствует несколько C)

A	B	C
<u>StudentId</u>	<u>Hobby</u>	<u>Course</u>
1022	стрельба	101-07
1022	поэзия	143-01
4123	конный спорт	101-07
4123	поэзия	143-01
4124	охота	179-04
4124	поэзия	

<u>StudentId</u>	<u>Hobby</u>
1022	стрельба
1022	поэзия
4123	конный спорт
4123	поэзия
4124	охота
4124	поэзия

<u>StudentId</u>	<u>Course</u>
1022	101-07
1022	143-01
4123	101-07
4123	143-01
4124	179-04
4124	

<u>StudentId</u>	<u>Course</u>
1022	101-07
1022	143-01
4123	101-07
4123	143-01
4124	179-04

5НФ

- таблица в 4НФ
- отсутствуют зависимости соединения

Э таблицы, для которых нельзя корректно провести декомпозицию на 2 таблицы, но можно выполнить корректную декомпозицию на 3 таблицы
можно доказать, что таблица, находящаяся в 5НФ удовлетворяет требованиям 4НФ

Проекты, в которых задействованы преподаватели

<u>Lecturer</u>	<u>Проект</u>	<u>Направление</u>
Есенин	Сайт	Разработка
Блок	Сайт	Маркетинг
Блок	Научная работа	Филиалы
Ахматова	SEO	Маркетинг
Есенин	SEO	Разработка
Есенин	Информационная система	Разработка

5НФ: деление на 2 таблицы

<u>Lecturer</u>	<u>Проект</u>	<u>Направление</u>
Есенин	Сайт	Разработка
Блок	Сайт	Маркетинг
Блок	Наука	Филиалы
Ахматова	SEO	Маркетинг
Есенин	SEO	Разработка
Есенин	БД	Разработка

t1

<u>Lecturer</u>	<u>Проект</u>
Есенин	Сайт
Блок	Сайт
Блок	Наука
Ахматова	SEO
Есенин	SEO
Есенин	БД

t2

<u>Проект</u>	<u>Направление</u>
Сайт	Разработка
Сайт	Маркетинг
Наука	Филиалы
SEO	Маркетинг
SEO	Разработка
БД	Разработка

t1

<u>Lecturer</u>	<u>Проект</u>	<u>Проект</u>	<u>Направление</u>
Есенин	Сайт	Сайт	Разработка
		Сайт	Маркетинг
Блок	Сайт	Сайт	Разработка
		Сайт	Маркетинг
Блок	Наука	Наука	Филиалы
Ахматова	SEO	SEO	Маркетинг
		SEO	Разработка
Есенин	SEO	SEO	Маркетинг
		SEO	Разработка
Есенин	БД	БД	Разработка

t2

t1

<u>Lecturer</u>	<u>Направление</u>
Есенин	Разработка
Блок	Маркетинг
Блок	Филиалы
Ахматова	Маркетинг
Есенин	Разработка
Есенин	Разработка

t2

<u>Направление</u>	<u>Проект</u>
Разработка	Сайт
Маркетинг	Сайт
Филиалы	Наука
Маркетинг	SEO
Разработка	SEO
Разработка	БД

t1

<u>Проект</u>	<u>Lecturer</u>
Сайт	Есенин
Сайт	Блок
Наука	Блок
SEO	Ахматова
SEO	Есенин
БД	Есенин

t2

<u>Lecturer</u>	<u>Направление</u>
Есенин	Разработка
Блок	Маркетинг
Блок	Филиалы
Ахматова	Маркетинг
Есенин	Разработка
Есенин	Разработка

5НФ: деление на 3 таблицы

<u>Lecturer</u>	<u>Проект</u>	<u>Направление</u>
Есенин	Сайт	Разработка
Блок	Сайт	Маркетинг
Блок	Научная работа	Филиалы
Ахматова	SEO	Маркетинг
Есенин	SEO	Разработка
Есенин	Информационная система	Разработка

t1

<u>Lecturer</u>	<u>Проект</u>
Есенин	Сайт
Блок	Сайт
Блок	Научная работа
Ахматова	SEO
Есенин	SEO
Есенин	Информационная система

t2

<u>Lecturer</u>	<u>Направление</u>
Есенин	Разработка
Блок	Маркетинг
Блок	Филиалы
Ахматова	Маркетинг

t3

<u>Проект</u>	<u>Направление</u>
Сайт	Разработка
Сайт	Маркетинг
Научная работа	Филиалы
SEO	Маркетинг
SEO	Разработка
Информационная система	Разработка

<u>Lecturer</u>	<u>Проект</u>	<u>Lecturer</u>	<u>Направление</u>	<u>Проект</u>	<u>Направление</u>
Есенин	Сайт	Есенин	Разработка	Сайт	Разработка
Блок	Сайт	Блок	Маркетинг	Сайт	Маркетинг
Блок	Научная работа	Блок	Филиалы	Научная работа	Филиалы
Ахматова	SEO	Ахматова	Маркетинг	SEO	Маркетинг
Есенин	SEO	Есенин	Разработка	SEO	Разработка
Есенин	Информационная система	Есенин	Разработка	Информационная система	Разработка

```
select *  
from t1  
inner join t2 on t2.Lecturer = t1.Lecturer  
inner join t3 on t3.Направление = t2.Направление and t3.Проект = t1.Проект
```

После нормализации

- денормализация (если слишком много джоинов/не устраивает скорость запросов)
- не забываем указать связи и их типы (1:1, 1:N, M:N)

Рефлексия

- как в реляционной схеме/БД реализуется связь M:N?
- зачем нужна нормализация?
- до какой формы обычно нормализуют данные?

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!