

Сроки по лабораторным работам

SQL:

- 3 лаб.работа – 3 неделя (текущая)
- 4 лаб.работа – 4 неделя
- 5 лаб.работа – 5 неделя
- 6 лаб.работа – 6 неделя
- *7 лаб.работа * – 7 неделя (+ сдача долгов)*
- РГР (индивидуально) – 8-9 недели – отчет по РГР

ОСНОВЫ ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННОЙ БД

Реляционная модель

- Основа - понятие **отношение** (relation).
- Отношение - множество элементов, называемых кортежами.
- Представляется в виде двумерной таблицы
- Записи (строки) – **кортежи**, поля (столбцы) - **атрибуты**

Характеристики реляционной базы данных

- Использование ключей
- Отсутствие избыточности данных
- Ограничение ввода
- Поддержание целостности данных

- Назначение прав
- Структурированный язык запросов (SQL)
- Переносимость
- Механизм транзакций

Проектирование БД

1. Инфологическое проектирование
2. Логическое проектирование
3. Физическое проектирование



Проектирование БД

1. Инфологическое проектирование

Проблема:

как адекватно отразить предметную область и информационные потребности пользователей в концептуальной модели?

Цель:

получение смысловых моделей, отражающих информационное содержание проблемы

Проектирование БД

1. Инфологическое проектирование

Проблема:

как адекватно отразить предметную область и информационные потребности пользователей в концептуальной модели?

анализ предметной области

Цель:

получение **смысловых моделей**, отражающих информационное содержание проблемы

Проектирование БД

2. Логическое проектирование

Проблема:

каким образом представить объекты предметной области
в абстрактных объектах так, чтобы это отражение не противоречило
семантике предметной области и было максимально удобным?

Цель:

организация данных с прошлого этапа в форму,
принятую в выбранной СУБД

Проектирование БД

2. Логическое проектирование

Проблема:

каким образом представить объекты предметной области
в **абстрактных объектах (логическая модель данных)** так, чтобы это
отражение не противоречило семантике предметной области и
было максимально удобным?

Цель:

организация данных с прошлого этапа в **форму выбранной СУБД**

Проектирование БД

3. Физическое проектирование

Проблема:

- Как обеспечить эффективность выполнения запросов к БД?
- Каким образом для конкретной СУБД расположить данные во внешней памяти?
- Создание каких доп. структур (например, индексов) необходимо?

Цель:

выбор рациональной структуры хранения данных
и методов доступа к ним

Проектирование БД

3. Физическое проектирование

Проблема:

- Как обеспечить эффективность выполнения запросов к БД?
- Каким образом для конкретной СУБД расположить данные во внешней памяти?
- Создание каких доп. структур (например, индексов) необходимо?

физическая модель данных

Цель:

выбор рациональной **структуры хранения данных**

и методов доступа к ним

Проектирование БД

1. **Инфологическое** проектирование

Анализ предметной области

2. **Логическое** проектирование

Представление в форме, принятой в СУБД

3. **Физическое** проектирование

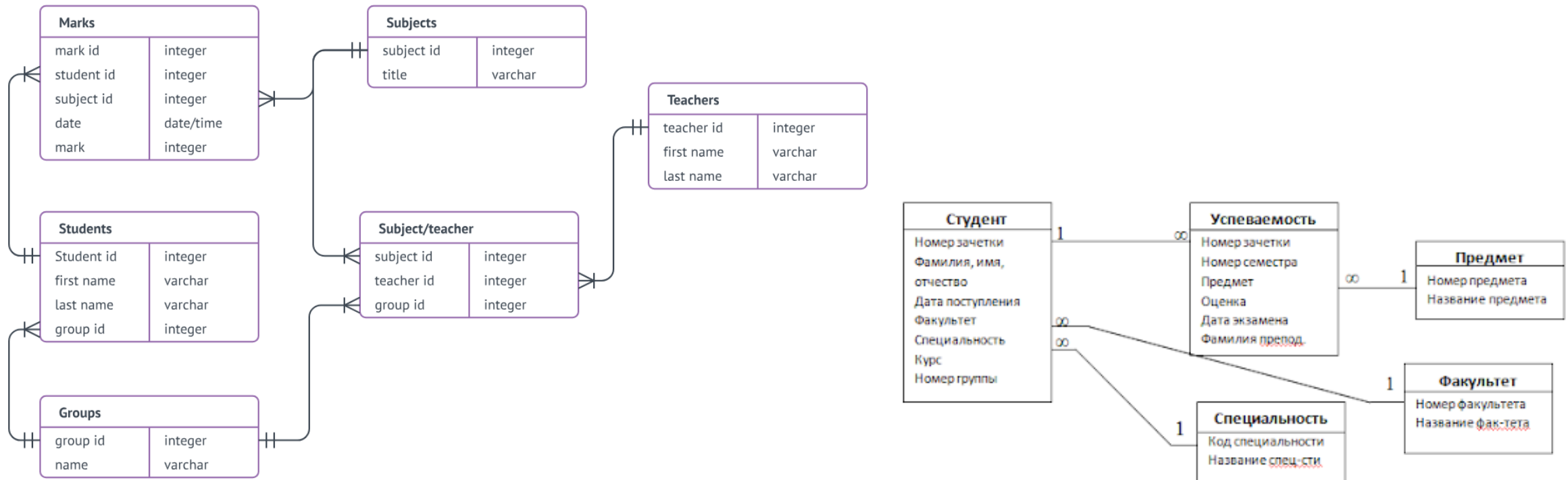
Хранение БД в памяти (структуры, методы и тд)

Инфологическая модель

Одна из наиболее популярных семантических моделей

на этапе инфологического проектирования –

ER-модель (entity-relation model)



Инфологическая модель

ER-модель (entity-relation model): сущность, атрибут и связь

Сущность – реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступной.

Пример сущности: животное, книга, продавец, заказ, деталь, студент

Атрибут – поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества. Каждому атрибуту дается имя, уникальное в пределах сущности.

Пример атрибутов сущностей:

кошка.вес, город.название, счёт.номер_клиента, заказ.стоимость

Инфологическая модель

ER-модель (entity-relation model): сущность, атрибут и связь

Атрибут может выступать как отдельная сущность, если это необходимо для корректного представления предметной области.

Например :

для автомобильного завода: цвет – атрибут сущности машина,

для лакокрасочной фабрики: цвет – сущность

Инфологическая модель

Связь между сущностями:

✓ *Один-к-одному (1:1)*

- с обязательной связью
- с необязательной связью

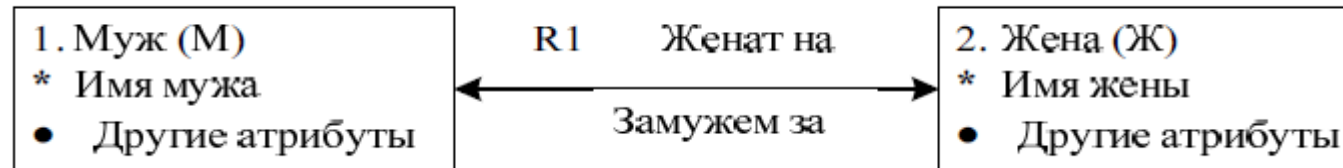
✓ *Один-ко-многим (1:M)*

- с обязательной связью
- с необязательной связью

✓ *Многие-ко-многим (M:N)*

Связь один-к-одному (1:1)

Один экземпляр одной сущности связан с **единственным** экземпляром другой сущности



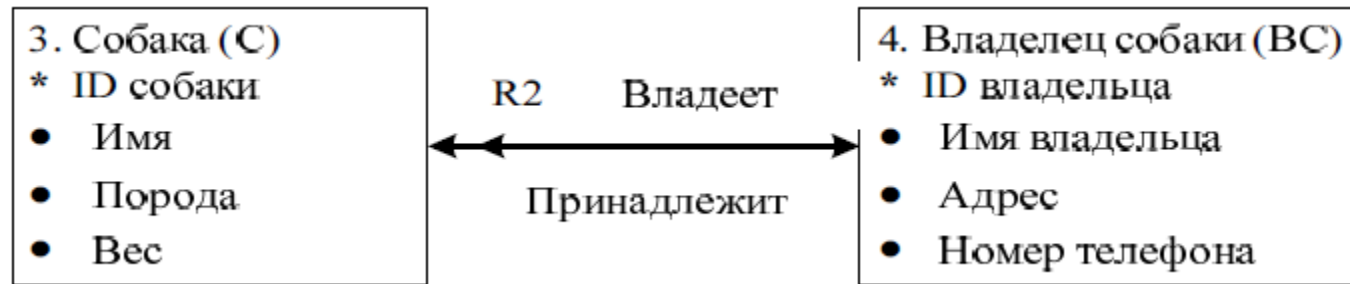
- Список сотрудников с ограниченными возможностями здоровья

Disabled_Person_ID	Employee_ID
1	159
2	722
3	937

- Каждый сотрудник может быть вписан *не более одного раза* →
столбец Employee_ID должен быть *уникальным*

Связь один-ко-многим (1:M)

Один экземпляр одной сущности связан с **одним или более** экземпляром другой сущности, **НО** каждый экземпляр другой сущности связан **только с одним** экземпляром первой сущности.



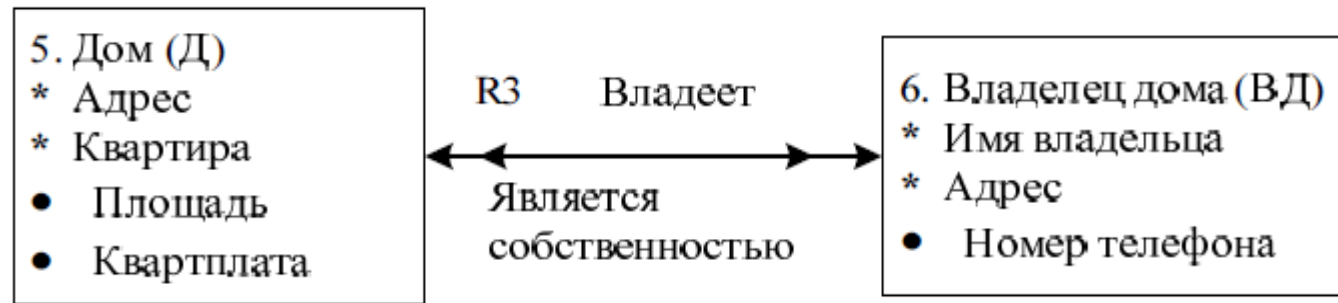
- Телефонная книга

Phone_ID	Person_ID	Phone_Number
1	5	333-22-11
2	5	444-55-66
3	17	777-88-99

- У одного человека (5) два телефона, но каждый телефон принадлежит одному человеку

Связь многие-ко-многим (M:N)

Один экземпляр одной сущности связан с **одним или более** экземпляром другой сущности и каждый экземпляр другой сущности связан **с одним или более** экземпляром первой сущности.



- Сотрудники и должности

Employee_ID	Position_ID
1	1
1	2
2	2
3	3

- У одного сотрудника (1) две должности, и на одной должности (2) два сотрудника

Инфологическая модель

Связь между сущностями:

✓ *Один-к-одному (1:1)*

- с обязательной связью
- с необязательной связью

✓ *Один-ко-многим (1:M)*

- с обязательной связью
- с необязательной связью

✓ *Многие-ко-многим (M:N)*

Обязательные и необязательные связи

✓ Один-к-одному с обязательной связью:

- У одного **гражданина** определенной страны обязательно есть только один паспорт этой страны.
- У одного **паспорта** есть только один владелец.

✓ Один-к-одному с необязательной связью:

- У одной **страны** может быть только одна конституция.
- Одна **конституция** принадлежит только одной стране.
- Но конституция не является обязательной. У страны она может быть, а может и не быть, как например, у Израиля и Великобритании.

Обязательные и необязательные связи

✓ Одну и ту же связь можно рассматривать **как обязательную и как необязательную:**

- У одного **человека** может быть только один загранпаспорт.
- У одного **загранпаспорта** есть только один владелец.

➤ А) Наличие загранпаспорта необязательно – его может и не быть у гражданина. Это необязательная связь.

➤ Б) У загранпаспорта обязательно есть только один владелец.
В этом случае, это уже обязательная связь.

Обязательные и необязательные связи

✓ Один-ко-многим с обязательной связью:

- К одному **полку** относятся многие бойцы.
- Один **боец** относится только к одному полку.
- Любой солдат обязательно принадлежит к одному полку, а полк не может существовать без солдат.

✓ Один-ко-многим с необязательной связью:

- На планете **Земля** живут все люди.
- Каждый **человек** живет только на Земле.
- При этом планета может существовать и без человечества.

Соответственно, нахождение нас на Земле не является обязательным

Обязательные и необязательные связи

✓ Одну и ту же связь можно рассматривать **как обязательную и как необязательную**:

- У одной биологической **матери** может быть много детей.
- У **ребенка** есть только одна биологическая мать.

➤ А) У женщины необязательно есть свои дети.

Соответственно, связь необязательна.

➤ Б) У ребенка обязательно есть только одна биологическая мать – в таком случае, связь обязательна.

Обязательные и необязательные связи

✓ Любая связь многие-ко-многим является необязательной.

- **Человек** может инвестировать в акции разных компаний (многих).
 - Инвесторами **компании** являются определенные люди (многие).
-
- А) Человек может вообще не инвестировать свои деньги в акции.
 - Б) Акции компании мог никто не купить.

Проектирование БД

1. **Инфологическое** проектирование

Анализ предметной области

2. **Логическое** проектирование

Представление в форме, принятой в СУБД

3. **Физическое** проектирование

Хранение БД в памяти (структуры, методы и тд)

Логическое проектирование

В ходе логического проектирования необходимо принять решения:

- из каких отношений должна состоять база данных;
- какие атрибуты должны быть у этих отношений.

Метод нормализации

(Недостаточно ли здравого смысла??)

Две основные цели:

- избежать **избыточности** данных;
- устранить **аномалии обновления** сущностей.

Нормализация

Нормализация заключается в приведении БД
в определенную **нормальную форму (НФ)**.

- Каждая следующая НФ является более ограниченной, чем предыдущая, но более удобной для практической работы.
- При переходе к следующей НФ свойства предыдущих НФ сохраняются

Чаще всего приведение в нормальную форму осуществляется с помощью **декомпозиции** – разбиения таблицы на две или более таблиц *без потери данных*.

Нормальные формы БД

- Ненормализованная форма или нулевая нормальная форма (UNF)
- Первая нормальная форма (1NF)
- Вторая нормальная форма (2NF)
- **Третья** нормальная форма (3NF)
- Нормальная форма **Бойса-Кодда** (BCNF)
- Четвертая нормальная форма (4NF)
- Пятая нормальная форма (5NF)
- Доменно-ключевая нормальная форма (DKNF)
- Шестая нормальная форма (6NF)

Нормальные формы БД

Ненормализованная форма или нулевая нормальная форма (UNF):

- Отсутствие каких-либо ограничений на данные таблицы

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Нормальные формы БД

Первая нормальная форма (1NF):

Отношение находится в первой нормальной форме тогда и только тогда, когда

в любом допустимом значении отношения каждый его кортеж содержит **только одно значение** для каждого из атрибутов

- В каждой ячейке таблицы только одно значение
 - Нет повторяющихся строк

Нормальные формы БД

Нулевая нормальная форма (UNF):

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Первая нормальная форма (1NF):

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

Нормальные формы БД

Вторая нормальная форма (2NF):

Отношение находится во второй нормальной форме тогда и только тогда, когда

когда оно находится в 1НФ

и каждый неключевой атрибут **неприводимо** (функционально полно) **зависит** от её потенциального ключа

- Есть первичный ключ
- Все атрибуты зависят от первичного ключа целиком, а не от какой-то его части

Нормальные формы БД

Первая нормальная форма (1NF):

Неключевые поля

<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%

Составной первичный ключ

Нормальные формы БД

Первая нормальная форма (1NF):

<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%

Составной первичный ключ



Нормальные формы БД

Первая нормальная форма (1NF):



<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%

Составной первичный ключ

ДЕКОМПОЗИЦИЯ

Нормальные формы БД

Вторая нормальная форма (2NF):

<u>Модель</u>	<u>Фирма</u>	Цена
M5	BMW	5500000
X5M	BMW	6000000
M1	BMW	2500000
GT-R	Nissan	5000000

<u>Фирма</u>	Скидка
BMW	5%
Nissan	10%

Нормальные формы БД

Третья нормальная форма (3NF):

Отношение находится в третьей нормальной форме тогда и только тогда, когда

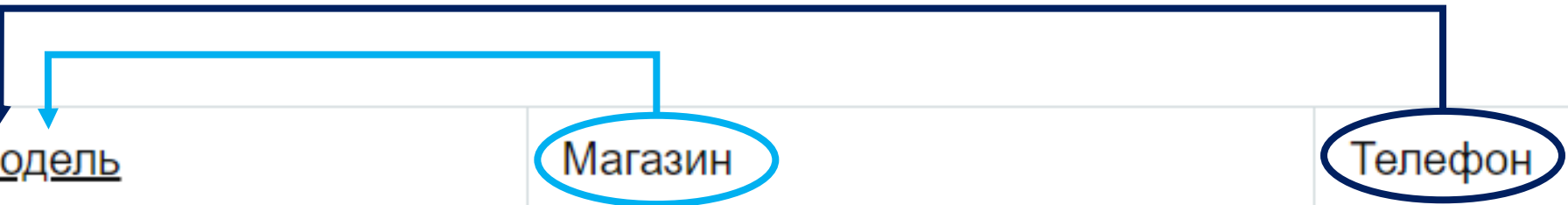
когда оно находится в 2НФ

**и отсутствуют транзитивные функциональные зависимости
неключевых атрибутов от ключевых.**

- Все атрибуты зависят только от первичного ключа,
но не от других атрибутов

Нормальные формы БД

Вторая нормальная форма (2NF):




The diagram illustrates a transitive dependency in a database table. A primary key 'Модель' (underlined) is connected by a blue line to a foreign key 'Магазин' (circled in blue). A dark blue line then connects 'Магазин' to another attribute 'Телефон' (circled in dark blue), indicating that 'Телефон' is functionally dependent on 'Магазин'.

<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12

Первичный ключ

Нормальные формы БД

Вторая нормальная форма (2NF):



<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12

Первичный ключ

Неключевое поле!

ДЕКОМПОЗИЦИЯ

Нормальные формы БД

Третья нормальная форма (3NF):

<u>Модель</u>	Магазин
BMW	Риал-авто
Audi	Риал-авто
Nissan	Некст-Авто

<u>Магазин</u>	Телефон
Риал-авто	87-33-98
Некст-Авто	94-54-12

Нормальные формы БД

Нормальная форма Бойса-Кодда (усиленная третья) (BCNF):

Отношение находится в норм.форме Бойса-Кодда тогда и только тогда, когда

когда оно находится в 3НФ

и каждая её нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего **детерминанта** некоторый **потенциальный ключ**

➤ Ключевые атрибуты не должны зависеть от неключевых

Нормальные формы БД

Третья нормальная форма (3NF):

Номер стоянки	Время начала	Время окончания	Тариф
1	09:30	10:30	Бережливый
1	11:00	12:00	Бережливый
1	14:00	15:30	Стандарт
2	10:00	12:00	Премиум-В
2	12:00	14:00	Премиум-В
2	15:00	18:00	Премиум-А

Возможны составные первичные ключи: {Номер стоянки, Время начала},

{Номер стоянки, Время окончания}, {Тариф, Время начала}, {Тариф, Время окончания}

- Неключевых полей нет, транзитивных зависимостей нет

Нормальные формы БД

Третья нормальная форма (3NF):

Номер стоянки	Время начала	Время окончания	Тариф
1	09:30	10:30	Бережливый
1	11:00	12:00	Бережливый
1	14:00	15:30	Стандарт
2	10:00	12:00	Премиум-В
2	12:00	14:00	Премиум-В
2	15:00	18:00	Премиум-А

Пусть ключом является {Тариф, Время начала}

ДЕКОМПОЗИЦИЯ

Тариф зависит от Номера стоянки, который не является ключевым

Нормальные формы БД

Нормальная форма Бойса-Кодда (BCNF):

<u>Тариф</u> Перв. ключ	Номер стоянки	Имеет льготы Доп. атрибут
Бережливый	1	Да
Стандарт	1	Нет
Премиум-А	2	Да
Премиум-В	2	Нет

Составной ключ

<u>Тариф</u>	<u>Время начала</u>	Время окончания
Бережливый	09:30	10:30
Бережливый	11:00	12:00
Стандарт	14:00	15:30
Премиум-В	10:00	12:00
Премиум-В	12:00	14:00
Премиум-А	15:00	18:00

Нормальные формы БД

Четвертая нормальная форма (4NF):

Отношение находится в четвертой норм.форме тогда и только тогда, когда
когда оно находится в НФБК

и не содержит нетривиальных многозначных зависимостей

➤ Устраняются многозначные зависимости

Нормальные формы БД

Многозначная зависимость в таблице:

Таблица должна иметь как минимум три столбца, например, A, B и C, при этом B и C между собой никак не связаны и *не зависят друг от друга*, **НО** по отдельности *зависят от A*, и для каждого значения A есть множество значений B, а также множество значений C.

Такая зависимость обозначается:

$$A \twoheadrightarrow B$$

$$A \twoheadrightarrow C$$

Если подобная многозначная зависимость есть в таблице, то она не в 4НФ

Нормальные формы БД

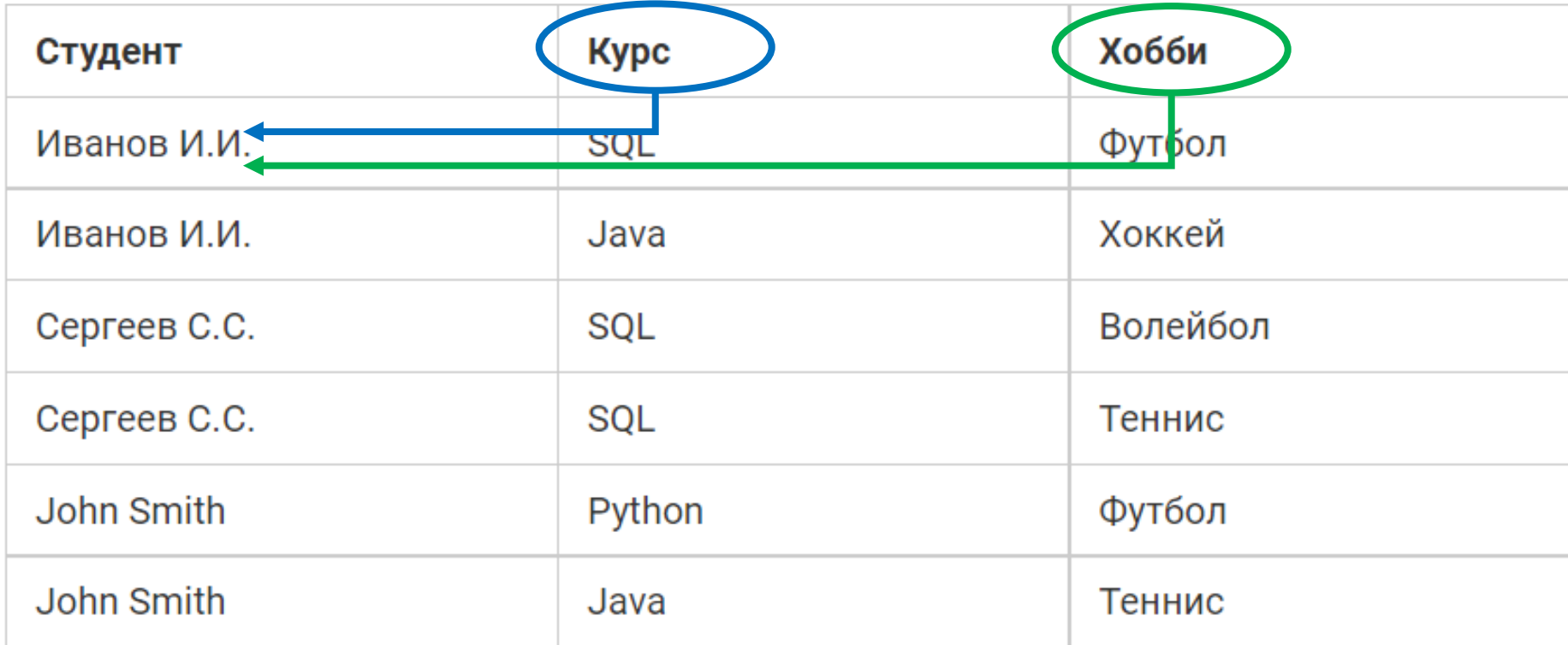
Нормальная форма Бойса-Кодда (BCNF):

Студент	Курс	Хобби
Иванов И.И.	SQL	Футбол
Иванов И.И.	Java	Хоккей
Сергеев С.С.	SQL	Волейбол
Сергеев С.С.	SQL	Теннис
John Smith	Python	Футбол
John Smith	Java	Теннис

Нормальные формы БД

Нормальная форма Бойса-Кодда (BCNF):

Студент	Курс	Хобби
Иванов И.И.	SQL	Футбол
Иванов И.И.	Java	Хоккей
Сергеев С.С.	SQL	Волейбол
Сергеев С.С.	SQL	Теннис
John Smith	Python	Футбол
John Smith	Java	Теннис



Первичный ключ здесь также составной и состоит он из всех трех столбцов. Курс и хобби никак не связаны и не зависят друг от друга, но по отдельности зависят от студента.

Студент --> Курс

Студент --> Хобби

Нормальные формы БД

Нормальная форма Бойса-Кодда (BCNF):

Студент	Курс	Хобби
Иванов И.И.	SQL	Футбол
Иванов И.И.	Java	Хоккей
Сергеев С.С.	SQL	Волейбол
Сергеев С.С.	SQL	Теннис
John Smith	Python	Футбол
John Smith	Java	Теннис

ДЕКОМПОЗИЦИЯ

Пусть необходимо получить информацию о хобби студентов, которые посещают курс по SQL. Очевидным действием станет выборка с условием **Курс = SQL** с результатом «волейбол», «теннис», «футбол». Хобби Иванова И.И. «хоккей» потеряно.

Нормальные формы БД

Четвертая нормальная форма (4NF):

Студент	Курс
Иванов И.И.	SQL
Иванов И.И.	Java
Сергеев С.С.	SQL
John Smith	Python
John Smith	Java

Нормализация до 4НФ, как и до последующих (5 и 6), в современном мире на реальных данных практически не встречается.

Студент	Хобби
Иванов И.И.	Футбол
Иванов И.И.	Хоккей
Сергеев С.С.	Волейбол
Сергеев С.С.	Теннис
John Smith	Футбол
John Smith	Теннис

Нормальные формы БД

Пятая нормальная форма (5NF):

Отношение находится в пятой нормальной форме тогда и только тогда, когда

когда оно находится в 4НФ

каждая нетривиальная зависимость соединения
в нем определяется потенциальным ключом
(ключами) этого отношения.

- Устраняются нетривиальные зависимости
- Декомпозиция без потерь

Нормальные формы БД

Без знания предметной области (беседы с экспертом и т.д.) **очень сложно** определить, находится ли таблица в 5НФ, т.е. есть ли в ней многозначные нетривиальные зависимости.

Сотрудник	Проект	Направление
Иванов И.И.	Интернет магазин	Разработка
Сергеев С.С.	Интернет магазин	Бухгалтерия
Сергеев С.С.	Новый офис	Реализация
John Smith	Личный кабинет	Бухгалтерия
Иванов И.И.	Личный кабинет	Разработка
Иванов И.И.	Информационная система	Разработка

Нормальные формы БД

Декомпозиция без потерь – процесс разбиения одной таблицы на несколько, при условии, что в случае соединения таблиц, которые были получены в результате декомпозиции, будет формироваться ровно та же самая информация, что и в исходной таблице до декомпозиции.

Нормальные формы БД

Пятая нормальная форма (5NF)

Существует таблица Т (С1, С2, С3) где С1, С2, С3 – составной первичным ключом. Таблица в 4НФ.

В соответствии с требованиями предметной области у нас проявляется зависимость соединения:

{С1, С2}, {С1, С3}, {С2, С3}

Чтобы привести данную таблицу к **5НФ**, необходимо декомпозировать ее на три таблицы:

Т1 (С1, С2),

Т2 (С1, С3),

Т3 (С2, С3)

При этом, если соединить (JOIN) три новые таблицы (Т1, Т2, Т3) и получить исходную таблицу (Т), то это будет означать, что декомпозицию выполнена без потерь.

Нормальные формы БД

Сотрудник	Проект	Направление
Иванов И.И.	Интернет магазин	Разработка
Сергеев С.С.	Интернет магазин	Бухгалтерия
Сергеев С.С.	Новый офис	Реализация
John Smith	Личный кабинет	Бухгалтерия
Иванов И.И.	Личный кабинет	Разработка
Иванов И.И.	Информационная система	Разработка

- Иванов И.И. может работать **только** в направлении «Разработка»
- Сергеев С.С. может работать в любом направлении, **за исключением** «Разработка»
- Иванов И.И. может участвовать в **большом** количестве проектов
- John Smith может участвовать только в **одном** проекте
- Чтобы выполнить декомпозицию без потерь, нам нужно разбить данную таблицу на три проекции: {Сотрудник, Проект}, {Сотрудник, Направление}, {Проект, Направление}

Нормальные формы БД

Пятая нормальная форма (5NF)

Проект	Направление
Интернет магазин	Разработка
Интернет магазин	Бухгалтерия
Новый офис	Реализация
Личный кабинет	Бухгалтерия
Личный кабинет	Разработка
Информационная система	Разработка

Сотрудник	Направление
Иванов И.И.	Разработка
Сергеев С.С.	Бухгалтерия
Сергеев С.С.	Реализация
John Smith	Бухгалтерия

Сотрудник	Проект
Иванов И.И.	Интернет магазин
Сергеев С.С.	Интернет магазин
Сергеев С.С.	Новый офис
John Smith	Личный кабинет
Иванов И.И.	Личный кабинет
Иванов И.И.	Информационная система

Нормальные формы БД

Пятая нормальная форма (5NF)

Если выполнить следующий запрос, который соединяет эти три таблицы, и он вернет точно такие же данные, что и в исходной таблице, то зависимости соединения нет, и наши таблицы находятся в 5NF.

```
SELECT СП.Сотрудник, ПН.Проект, СН.Направление
FROM СотрудникПроект СП
JOIN ПроектНаправление ПН ON СП.Проект = ПН.Проект
JOIN СотрудникНаправление СН ON СП.Сотрудник = СН.Сотрудник
AND ПН.Направление = СН.Направление
```

Данные совпадают
Таблицы находятся в 5НФ.

	Сотрудник	Проект	Направление
1	Иванов И.И.	Интернет магазин	Разработка
2	Сергеев С.С.	Интернет магазин	Бухгалтерия
3	Сергеев С.С.	Новый офис	Реализация
4	John Smith	Личный кабинет	Бухгалтерия
5	Иванов И.И.	Личный кабинет	Разработка
6	Иванов И.И.	Информационная система	Разработка

	Сотрудник	Проект	Направление
1	Иванов И.И.	Интернет магазин	Разработка
2	Сергеев С.С.	Интернет магазин	Бухгалтерия
3	Сергеев С.С.	Новый офис	Реализация
4	John Smith	Личный кабинет	Бухгалтерия
5	Иванов И.И.	Личный кабинет	Разработка
6	Иванов И.И.	Информационная система	Разработка

Нормальные формы БД

- Ненормализованная форма или нулевая нормальная форма (UNF)
- Первая нормальная форма (1NF)
- Вторая нормальная форма (2NF)
- **Третья нормальная форма (3NF)**
- **Нормальная форма Бойса-Кодда (BCNF)**
- Четвертая нормальная форма (4NF)
- Пятая нормальная форма (5NF)
- Доменно-ключевая нормальная форма (DKNF)
- Шестая нормальная форма (6NF)

Проектирование БД

1. Инфологическое проектирование
2. Логическое проектирование
3. Физическое проектирование



Процедура проектирования

Процесс проектирования информационных систем начинается с идентификации сущностей. Затем необходимо выполнить следующие шаги:

1. Представить каждый **стержень** (независимую сущность) таблицей базы данных (базовой таблицей) и специфицировать *первичный ключ* этой базовой **таблицы**.
2. Представить каждую **ассоциацию** (связь вида «многие-ко-многим» или «многие-ко-многим-ко-многим» и т.д. между сущностями) как базовую **таблицу**.

Использовать в этой таблице *внешние ключи* для идентификации участников ассоциации и специфицировать ограничения, связанные с каждым из этих внешних ключей.

Процедура проектирования

3. Представить каждую **характеристику** как базовую **таблицу с внешним ключом**, идентифицирующим сущность, описываемую этой характеристикой.
Специфицировать ограничения на внешний ключ этой таблицы и ее первичный ключ - по всей вероятности, комбинации этого внешнего ключа и свойства, которое гарантирует «уникальность в рамках описываемой сущности».
4. Представить каждое **обозначение**, которое не рассматривалось в предыдущем пункте, как базовую **таблицу с внешним ключом**, идентифицирующим обозначаемую сущность. Специфицировать связанные с каждым таким внешним ключом ограничения.
5. Представить каждое **свойство** как **поле** в базовой таблице, представляющей сущность, которая непосредственно описывается этим свойством.

Процедура проектирования

6. Для того, чтобы исключить в проекте непреднамеренные нарушения каких-либо принципов нормализации, выполнить **процедуру нормализации**.
7. Если в процессе нормализации было произведено разделение каких-либо таблиц, то следует модифицировать инфологическую модель базы данных и повторить перечисленные шаги.
8. Указать **ограничения целостности** проектируемой базы данных и дать (если это необходимо) краткое описание полученных таблиц и их полей.