

Продвинутый SQL

Лекция 3.
Нормализация.



Николай Хащанов

Full-stack developer

Защитная зона
для интеграции
видео спикера

О спикере:

- Разрабатываю и поддерживаю crm/erp системы
- Преподаю в Нетологии
- Окончил РГТЭУ по специальности Менеджмент
- Оптимизация и автоматизация бизнес-процессов

Я в Слаке:

 @Николай Хащанов



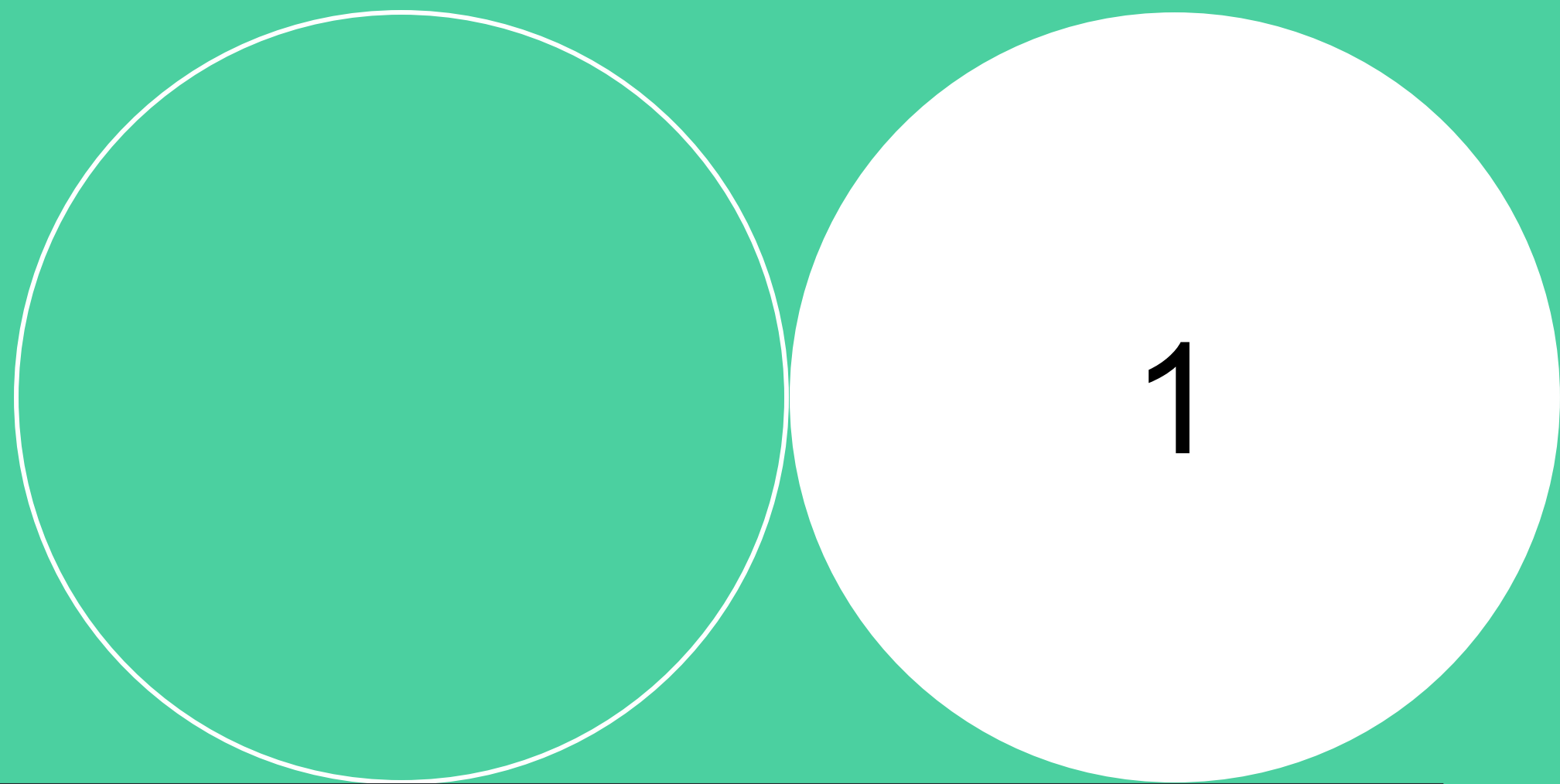
Содержание

Защитная зона
для интеграции
видео спикера

- 1 1 – 3 нормальные формы
- 2 Нормальная форма Бойса — Кодда
- 3 4 нормальная форма
- 4 5 нормальная форма
- 5 OLTP



Нормализация



Нормализация – это метод

проектирования базы данных, который используется для разработки таблицы реляционной базы данных до более высокой нормальной формы. При этом данный процесс является прогрессивным, и более высокий уровень нормализации базы данных не может быть достигнут, если не были выполнены предыдущие уровни.

Защитная зона
для интеграции
видео спикера



Исходные данные.

И снова магазин цветов!

Мы помним, что планировался запуск интернет-магазина и нужно было разработать под него базу данных.

Предположим, что кто-то создал таблицу с данными о пользователе:

```
create table customer (  
  customer_name varchar(50) not null,  
  customer_login varchar(20) not null,  
  customer_password varchar(10) not null,  
  customer_address varchar(50) not null,  
  delivery_address varchar(50) not null,  
  purchases json, --{id_продукта: количество}  
  amount decimal(10,2),  
  wish_list text[]  
)
```



Исходные данные.



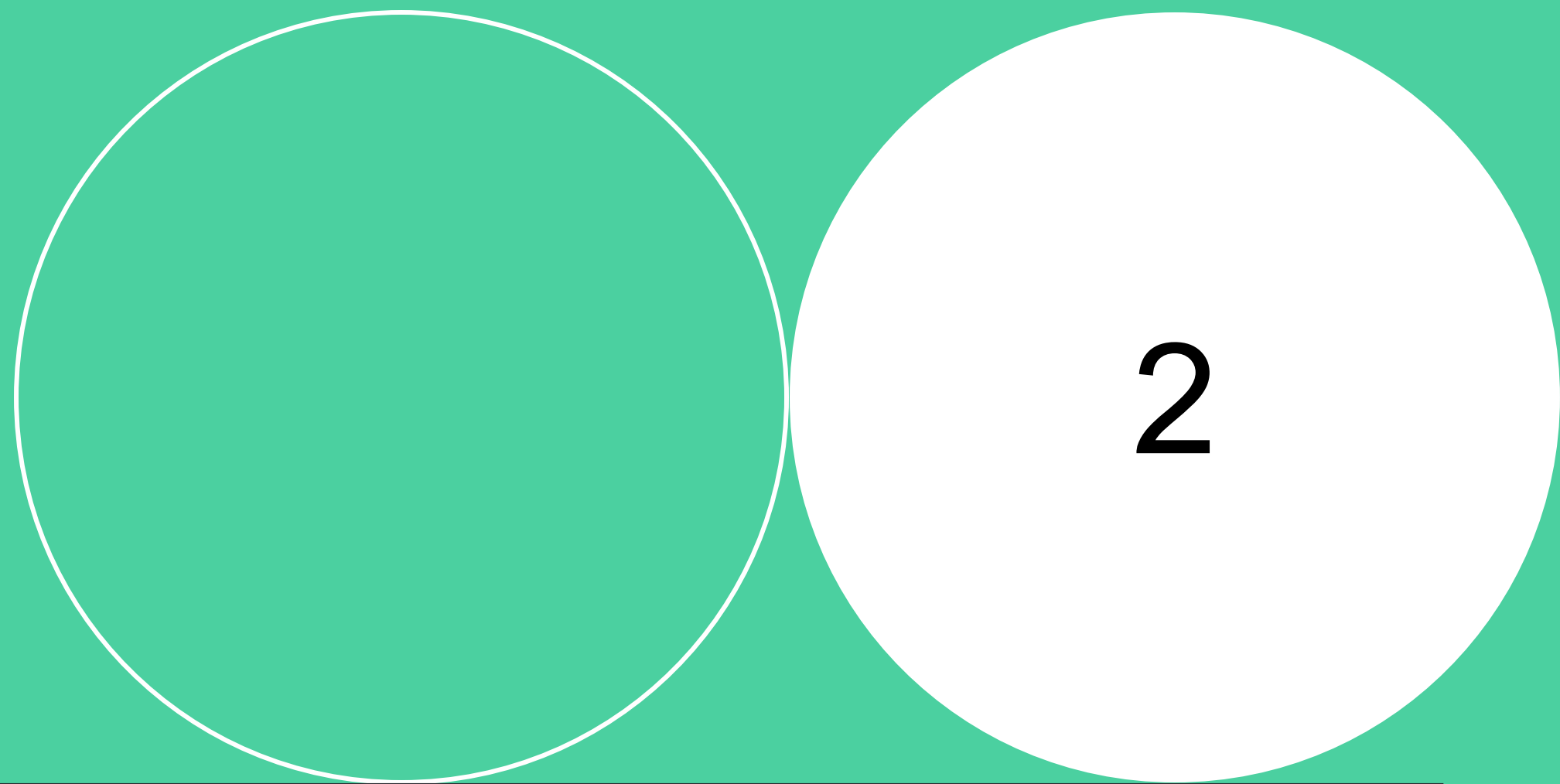
В результате будет вот такая таблица:

	customer_name	customer_login	customer_password	customer_address	delivery_address	purchases	amount	wish_list
1	Федор Иванович	fivan123	1q2w3e	London, Near BigBen	Freedom str, 56	{"23":4, "17":2, "15":3}	1 800,35	{1,14,3}
2	Федор Иванович	fivan123	1q2w3e	London, Near BigBen	Freedom str, 56	{"14":3}	150,22	{1,3}
3	Петя Галушкин	numberone	556677	Москва, Научный проезд 23	Ленинский проспект 80	{"11":1, "12":1, "73":1}	350	{5}
4	Петя Галушкин	numberone	556677	Москва, Научный проезд 23	Расплетина 33	{"51":99}	5 000	{16,28}
5	Петя Галушкин	numberone	556677	Москва, Научный проезд 23	Широкая 12/3	{"16":3, "28":6}	1 800,35	{16,28}

Скорее всего, с этим будет не совсем удобно работать...



1 нормальная форма





1 нормальная форма.

Чтобы наша таблица удовлетворяла 1NF, значения в каждом столбце должны быть атомарными. То есть, значения в домене каждого атрибута отношения не являются ни списками, ни множествами простых или сложных значений.

Столбцы `purchases` и `wish_list` хранят множества данных, а в столбце `customer_address` можно выделить город проживания.

Давайте произведем нормализацию.





Создадим несколько новых таблиц.

- 1 Города
- 2 Продажи
- 3 Список пожеланий

Мы могли бы для каждого отдельного товара или позиции из списка пожеланий делать новую запись в таблице customer, но это привело бы к аномалиям и избыточности данных (это же 3NF...)



1 нормальная форма.

Защитная зона
для интеграции
видео спикера

Города:

	123 city_id 🔒	ABC city_name 🔒
1	1	London
2	2	Москва

Продажи:

123 purchase_id 🔒	ABC customer_name 🔒	ABC customer_login 🔒	ABC customer_password 🔒	123 product_id 🔒	123 quantity 🔒
1	Федор Иванович	fivan123	1q2w3e	23	3
2	Федор Иванович	fivan123	1q2w3e	17	2
3	Федор Иванович	fivan123	1q2w3e	15	3
4	Федор Иванович	fivan123	1q2w3e	14	3
5	Петя Галушкин	numberone	556677	11	1
6	Петя Галушкин	numberone	556677	12	1
7	Петя Галушкин	numberone	556677	73	1
8	Петя Галушкин	numberone	556677	51	99
9	Петя Галушкин	numberone	556677	16	3
10	Петя Галушкин	numberone	556677	28	6

Лист пожеланий:

123 customer_id 🔒	123 product_id 🔒
1	1
1	3
2	16
2	28

Обратите внимание, на сколько лучше выглядит «Лист пожеланий», но об этом чуть попозже





1 нормальная форма.

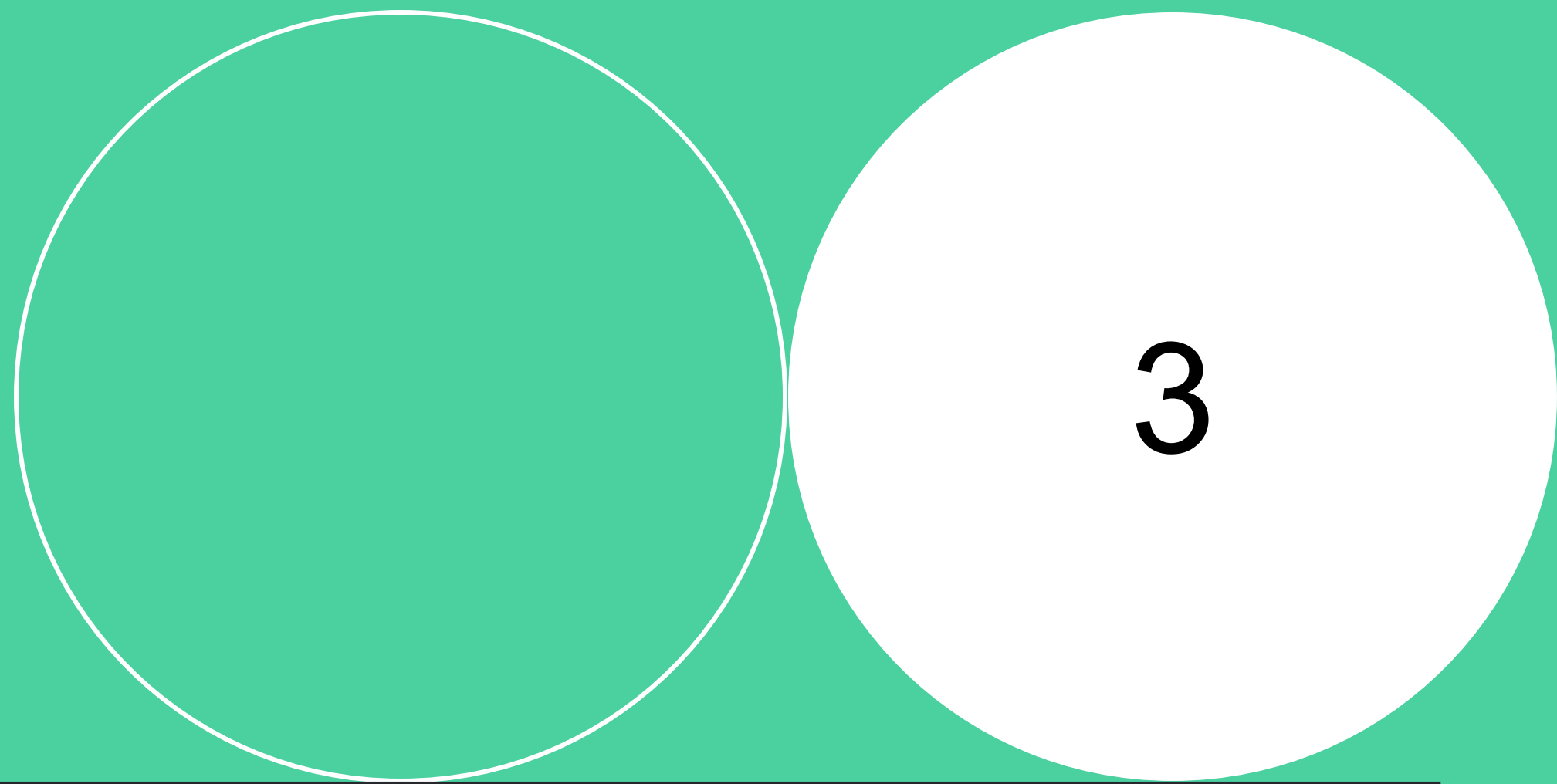
И таблица customer будет выглядеть следующим образом:

ABC customer_name T↑	ABC customer_login T↑	ABC customer_password T↑	123 city_id T↑	ABC customer_address T↑	ABC delivery_address T↑
Федор Иванович	fivan123	1q2w3e	1	Near BigBen	Freedom str, 56
Федор Иванович	fivan123	1q2w3e	1	Near BigBen	Freedom str, 56
Петя Галушкин	numberone	556677	2	Научный проезд 23	Ленинский проспект 80
Петя Галушкин	numberone	556677	2	Научный проезд 23	Расплетина 33
Петя Галушкин	numberone	556677	2	Научный проезд 23	Широкая 12/3

Теперь мы можем сказать, что наша таблица удовлетворяет условиям 1NF. Так же как и созданные дополнительные три таблицы.



2 нормальная форма





2 нормальная форма.

Таблица обязана соответствовать первой нормальной форме.

Все столбцы, которые не являются частью ключа, зависят от этого ключа.

Чтобы соответствовать 2NF и удалить дубликаты, каждый неключевой атрибут должен зависеть от всего ключа, а не только от его части.

Вы наверно обратили внимание, как не хорошо выглядит таблица с продажами, надо вносить изменения в структуру продаж и менять первичный ключ.



2 нормальная форма.



В таблице customer создадим идентификатор, который сделаем первичным ключом.

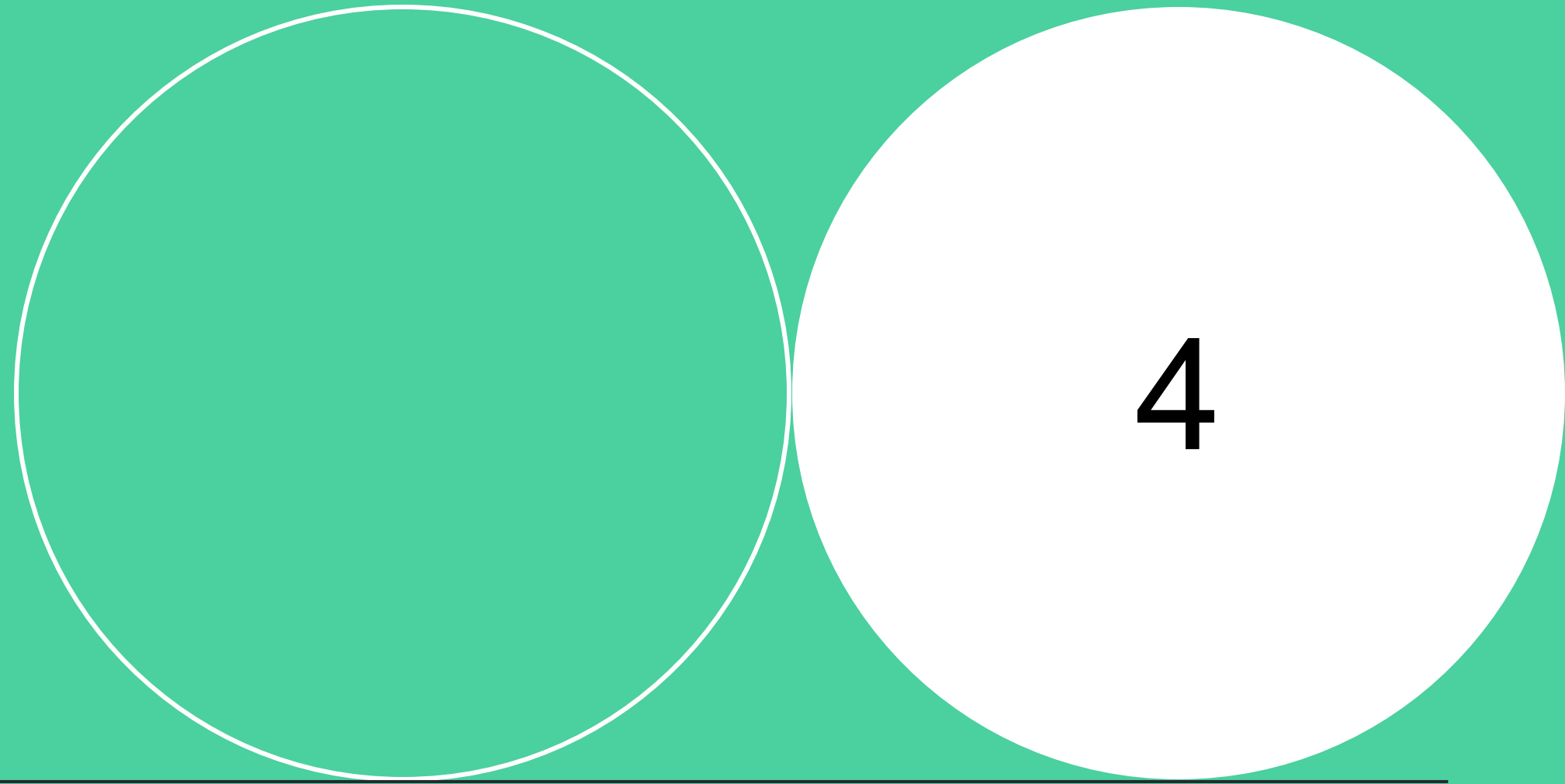
customer_id	customer_name	customer_login	customer_password	city_id	customer_address	delivery_address
1	Федор Иванович	fivan123	1q2w3e	1	Near BigBen	Freedom str, 56
2	Федор Иванович	fivan123	1q2w3e	1	Near BigBen	Freedom str, 56
3	Петя Галушкин	numberone	556677	2	Научный проезд 23	Ленинский проспект 80
4	Петя Галушкин	numberone	556677	2	Научный проезд 23	Расплетина 33
5	Петя Галушкин	numberone	556677	2	Научный проезд 23	Широкая 12/3

А что стало с продажами?
Явно лучше!

purchase_id	customer_id	product_id	quantity
1	1	23	3
2	1	17	2
3	1	15	3
4	1	14	3
5	2	11	1
6	2	12	1
7	2	73	1
8	2	51	99
9	2	16	3
10	2	28	6



3 нормальная форма





3 нормальная форма.

Таблица обязана соответствовать второй нормальной форме.

Значения, входящие в запись и не являющиеся частью ключа этой записи, не принадлежат таблице.

В таблице customer осталась информация по адресу доставки, которая относится к продажам.

Создадим таблицу, в которую будем записывать адреса для доставки и уберем информация из customer.



3 нормальная форма.



Таблица с пользователями:

customer_id	customer_name	customer_login	customer_password	city_id	customer_address
1	Федор Иванович	fivan123	1q2w3e	1	Near BigBen
2	Петя Галушкин	numberone	556677	2	Научный проезд 23

Таблица по доставке. Обратите внимание, что добавили информацию по городу

delivery_id	customer_id	city_id	address
1	1	1	Freedom str, 56
2	1	1	Freedom str, 56
3	2	2	Ленинский проспект 80
4	2	2	Расплетина 33
5	2	2	Широкая 12/3



3 нормальная форма.

Защитная зона
для интеграции
видео спикера

Но тогда мы должны привязать идентификатор доставки к продажам:

purchase_id	customer_id	product_id	quantity	delivery_id
1	1	23	3	1
2	1	17	2	1
3	1	15	3	1
4	1	14	3	2
5	2	11	1	3
6	2	12	1	3
7	2	73	1	3
8	2	51	99	4
9	2	16	3	5
10	2	28	6	5



Нормальная форма Бойса — Кодда



4

Реляционная схема считается в нормальной форме Бойса-Кодда (BCNF), если для каждой из ее зависимостей $A \rightarrow B$ выполняется одно из следующих условий:

- $A \rightarrow B$ является тривиальной функциональной зависимостью (то есть B является подмножеством A)
- A – первичный ключ для схемы реляционной схемы





Как правило 3НФ является желаемым результатом и дальнейшая нормализация может приводить к ненужному результату, из-за которого усложняется выборка данных.

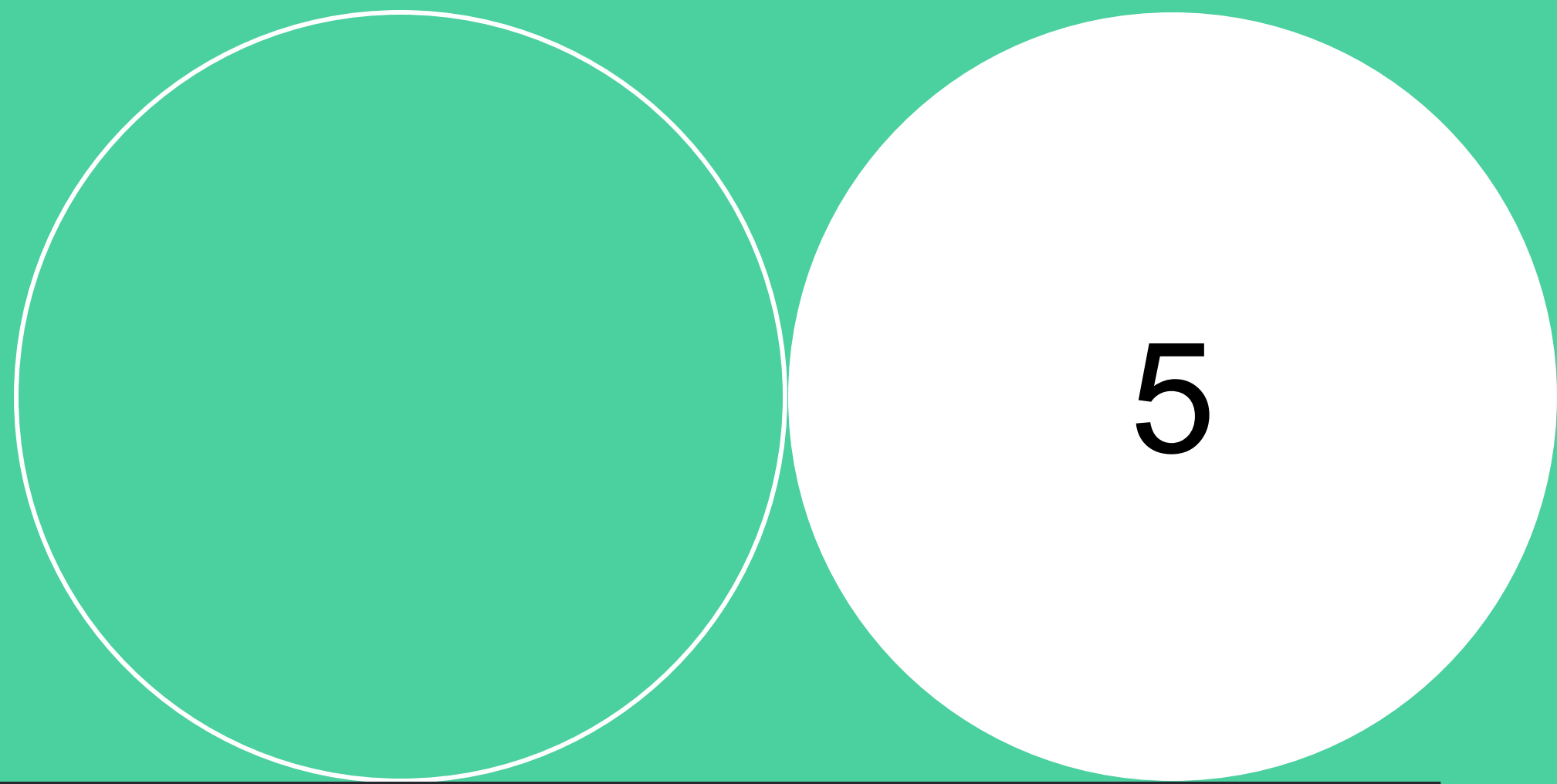
Давайте разделим таблицу по продажам на две:

123 purchase_id	123 product_id	123 quantity
1	23	3
2	17	2
3	15	3
4	14	3
5	11	1
6	12	1
7	73	1
8	51	99
9	16	3
10	28	6

123 purchase_id	123 delivery_id
1	1
2	1
3	1
4	2
5	3
6	3
7	3
8	4
9	5
10	5



4 нормальная форма





4 нормальная форма.

4 нормальная форма применяется для устранения многозначных зависимостей - таких зависимостей, где столбец с первичным ключом имеет связь один-ко-многим со столбцом, который не является ключом. Эта нормальная форма устраняет некорректные отношения многие-ко-многим.





4 нормальная форма.

Предположим, что из-за большого количества заказов пришлось открыть несколько складов в разных городах и мы создали таблицу, которая хранит количество товара на каждом складе:

123 store_id	123 product_id	123 city_id	123 amount
1	2	1	100
1	3	1	200
1	15	1	300
2	2	2	1 230
2	17	2	80
2	89	2	171

Так как в этой таблице составной первичный ключ (store_id, product_id, city_id), она находится в НФБК.





4 нормальная форма.

Для того, что бы удовлетворить требования 4 нормальной формы, разделим таблицу по складам:

123 store_id	123 city_id
1	1
2	2

123 store_id	123 product_id	123 amount
1	2	100
1	3	200
1	15	300
2	2	1 230
2	17	80
2	89	171




5 нормальная форма



6

5 нормальная форма.



Защитная зона
для интеграции
видео спикера

5 нормальная форма разделяет таблицы на более малые таблицы для устранения избыточности данных. Разбиение идет до тех пор, пока нельзя будет воссоздать оригинальную таблицу путем объединения малых таблиц.





5 нормальная форма.

Таблица с продажами в НФБК выглядела следующим образом:

123store_id	123product_id	123city_id	123amount
1	2	1	100
1	3	1	200
1	15	1	300
2	2	2	1 230
2	17	2	80
2	89	2	171

Мы ее разбили на две таблицы и пусть каждый склад работает на несколько городов:

123store_id	123city_id
1	1
2	2
1	3
2	4

123store_id	123product_id	123amount
1	2	100
1	3	200
1	15	300
2	2	1 230
2	17	80
2	89	171





5 нормальная форма.

Если мы выполним join двух малых таблиц, то получим следующий результат:

store_id	city_id	product_id	amount
1	1	2	100
1	3	2	100
1	1	3	200
1	3	3	200
1	1	15	300
1	3	15	300
2	2	2	1 230
2	4	2	1 230
2	2	17	80
2	4	17	80
2	2	89	171
2	4	89	171

Где темно-синим выделен ложный результат.





5 нормальная форма.

Давайте разобьем исходную таблицу на три:

123 store_id	123 city_id
1	1
2	2
1	3
2	4

123 store_id	123 product_id	123 amount
1	2	100
1	3	200
1	15	300
2	2	1 230
2	17	80
2	89	171

123 city_id	123 product_id
1	2
1	3
1	15
2	2
2	17
2	89

Что теперь будет возвращать join и как его придется писать?

Необходимо помнить, что при извлечении информации (например, о городе и товарах) необходимо в запросе соединить все три отношения. Любая комбинация соединения двух отношений из трех неминуемо приведет к извлечению ложной информации.





5 нормальная форма.

Проверим на практике:

```
select sc.store_id, sc.city_id, sp.product_id, sp.amount
from store_city sc
join store_product sp on sp.store_id = sc.store_id
join city_product cp on cp.city_id = sc.city_id and cp.product_id = sp.product_id
```



























































































123store_id	123city_id	123product_id	123amount
1	1	2	100
1	1	3	200
1	1	15	300
2	2	2	1 230
2	2	17	80
2	2	89	171

Соответственно наши таблицы удовлетворяют пятой нормальной форме



Результат:

Защитная зона
для интеграции
видео спикера

<table><tr><td> flowers.city</td></tr><tr><td>123  city_id int4</td></tr><tr><td>ABC city_name varchar(50)</td></tr></table>	 flowers.city	123  city_id int4	ABC city_name varchar(50)	<table><tr><td> flowers.city_product</td></tr><tr><td>123  city_id int4</td></tr><tr><td>123  product_id int4</td></tr></table>	 flowers.city_product	123  city_id int4	123  product_id int4	<table><tr><td> flowers.purchases_info</td></tr><tr><td>123  purchase_id int4</td></tr><tr><td>123  delivery_id int4</td></tr></table>	 flowers.purchases_info	123  purchase_id int4	123  delivery_id int4	<table><tr><td> flowers.store_city</td></tr><tr><td>123  store_id int4</td></tr><tr><td>123  city_id int4</td></tr></table>	 flowers.store_city	123  store_id int4	123  city_id int4									
 flowers.city																								
123  city_id int4																								
ABC city_name varchar(50)																								
 flowers.city_product																								
123  city_id int4																								
123  product_id int4																								
 flowers.purchases_info																								
123  purchase_id int4																								
123  delivery_id int4																								
 flowers.store_city																								
123  store_id int4																								
123  city_id int4																								
<table><tr><td> flowers.store_product</td></tr><tr><td>123  store_id int4</td></tr><tr><td>123  product_id int4</td></tr><tr><td>123 amount int4</td></tr></table>	 flowers.store_product	123  store_id int4	123  product_id int4	123 amount int4	<table><tr><td> flowers.delivery</td></tr><tr><td>123  delivery_id int4</td></tr><tr><td>123 customer_id int4</td></tr><tr><td>123 city_id int4</td></tr><tr><td>ABC address varchar(50)</td></tr></table>	 flowers.delivery	123  delivery_id int4	123 customer_id int4	123 city_id int4	ABC address varchar(50)	<table><tr><td> flowers.store</td></tr><tr><td>123  store_id int4</td></tr><tr><td>123  product_id int4</td></tr><tr><td>123  city_id int4</td></tr><tr><td>123 amount int4</td></tr></table>	 flowers.store	123  store_id int4	123  product_id int4	123  city_id int4	123 amount int4	<table><tr><td> flowers.products</td></tr><tr><td>123  product_id int4</td></tr><tr><td>ABC product_name varchar(50)</td></tr><tr><td>123 product_amount numeric(10,2)</td></tr><tr><td>ABC product_color varchar(30)</td></tr><tr><td>123 product_count int4</td></tr></table>	 flowers.products	123  product_id int4	ABC product_name varchar(50)	123 product_amount numeric(10,2)	ABC product_color varchar(30)	123 product_count int4	
 flowers.store_product																								
123  store_id int4																								
123  product_id int4																								
123 amount int4																								
 flowers.delivery																								
123  delivery_id int4																								
123 customer_id int4																								
123 city_id int4																								
ABC address varchar(50)																								
 flowers.store																								
123  store_id int4																								
123  product_id int4																								
123  city_id int4																								
123 amount int4																								
 flowers.products																								
123  product_id int4																								
ABC product_name varchar(50)																								
123 product_amount numeric(10,2)																								
ABC product_color varchar(30)																								
123 product_count int4																								
<table><tr><td> flowers.customer_info</td></tr><tr><td>123  customer_id int4</td></tr><tr><td>ABC customer_name varchar(50)</td></tr><tr><td>ABC customer_login varchar(20)</td></tr><tr><td>ABC customer_password varchar(10)</td></tr><tr><td>123 city_id int4</td></tr><tr><td>ABC customer_address varchar(50)</td></tr></table>	 flowers.customer_info	123  customer_id int4	ABC customer_name varchar(50)	ABC customer_login varchar(20)	ABC customer_password varchar(10)	123 city_id int4	ABC customer_address varchar(50)	<table><tr><td> flowers.purchases</td></tr><tr><td>123  purchase_id int4</td></tr><tr><td>ABC purchase_name varchar(50)</td></tr><tr><td>123 purchase_amount numeric(10,2)</td></tr><tr><td>ABC purchase_color varchar(30)</td></tr><tr><td>123 purchase_count int4</td></tr><tr><td> payment_date timestamp(6)</td></tr></table>	 flowers.purchases	123  purchase_id int4	ABC purchase_name varchar(50)	123 purchase_amount numeric(10,2)	ABC purchase_color varchar(30)	123 purchase_count int4	 payment_date timestamp(6)	<table><tr><td> flowers.purchases_shop</td></tr><tr><td>123  purchase_id int4</td></tr><tr><td>123 product_id int4</td></tr><tr><td>123 quantity int4</td></tr></table>	 flowers.purchases_shop	123  purchase_id int4	123 product_id int4	123 quantity int4	<table><tr><td> flowers.wish_list</td></tr><tr><td>123 customer_id int4</td></tr><tr><td>123 product_id int4</td></tr></table>	 flowers.wish_list	123 customer_id int4	123 product_id int4
 flowers.customer_info																								
123  customer_id int4																								
ABC customer_name varchar(50)																								
ABC customer_login varchar(20)																								
ABC customer_password varchar(10)																								
123 city_id int4																								
ABC customer_address varchar(50)																								
 flowers.purchases																								
123  purchase_id int4																								
ABC purchase_name varchar(50)																								
123 purchase_amount numeric(10,2)																								
ABC purchase_color varchar(30)																								
123 purchase_count int4																								
 payment_date timestamp(6)																								
 flowers.purchases_shop																								
123  purchase_id int4																								
123 product_id int4																								
123 quantity int4																								
 flowers.wish_list																								
123 customer_id int4																								
123 product_id int4																								



Нужно ли начинать с 1 формы?

Защитная зона
для интеграции
видео спикера

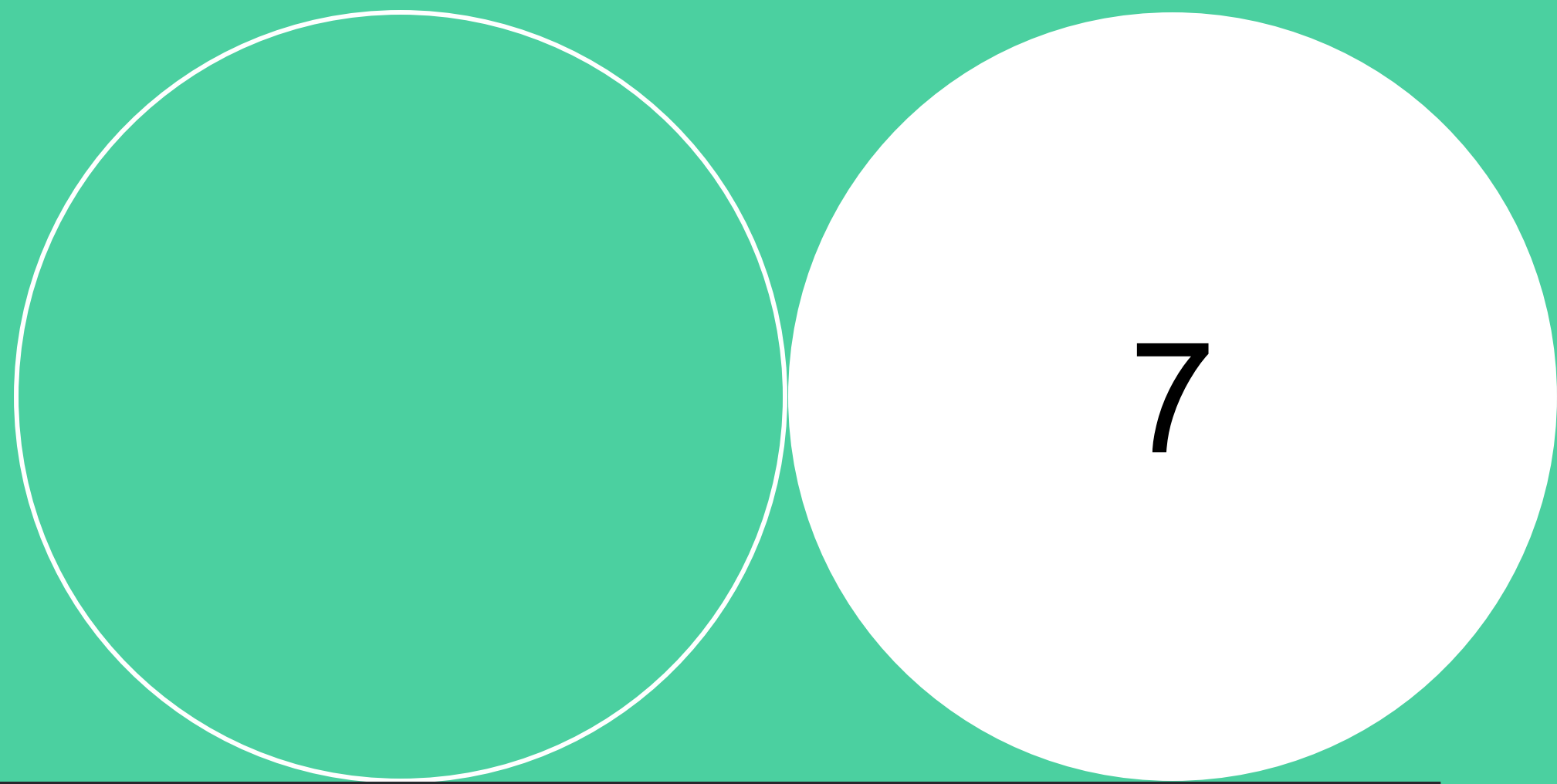
При проектировании базы можно избегать какие либо из форм нормализации. Если изначально есть понимание разделения сущностей и связей по своим местам, то можно пропустить начальные формы нормализации.

Так же при исправлении нарушений одной нормальной формы можно заранее учесть нарушения более высокой формы.

Мы не будем рассматривать ETNF (основного домена), DKNF (ключа домена), 6 нормальную форму так как они носят больше научный характер и на практике не применяются.



OLTP



OLTP – обработка транзакций в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика.

Защитная зона
для интеграции
видео спикера



OLTP.

Защитная зона
для интеграции
видео спикера

Большинство запросов к базе данных в OLTP-приложениях состоят из команд insert, update, delete. Select в OLTP - системах, как правило, предназначены для предоставления данных из справочников.

База данных OLTP часто обновляется. Транзакция в OLTP системе во время выполнения может завершиться ошибкой, что может привести к нарушению целостности данных . Таким образом, необходимо отслеживать целостность данных и мы должны понимать, что таблицы базы данных при работе с OLTP системой должны быть нормализованы (3НФ).

При возникновении ошибок транзакция должна откатиться и вернуть систему в состояние, в котором та была до начала транзакции.



Транзакция – это

последовательность операций,
выполняемых в логическом порядке
пользователем, либо программой, которая
работает с базой данных.

Защитная зона
для интеграции
видео спикера





Свойства OLTP для стабильной работы.

1. Атомарность. Транзакция должна выполняться как единая операция доступа к базе данных и может быть выполнена полностью либо не выполнена совсем.
2. Согласованность. Свойство согласованности гарантирует выполнение ограничений целостности базы данных после окончания обработки транзакции.
3. Изолированность. Транзакции должны выполняться независимо друг от друга, и доступ к данным, изменяемым с помощью одной транзакции, для других транзакций должен быть запрещен, пока изменения не будут завершены.
4. Долговечность. Если транзакция выполнена успешно, то произведенные ею изменения в данных не должны быть потеряны ни при каких обстоятельствах.



Неужели это все?



Конечно нет.

Про OLTP-системы мы еще немного поговорим, но в рамках сравнения с OLAP-системами, что бы понять преимущества, недостатки и какие задачи решает в сравнении с другой системой.



Итоги

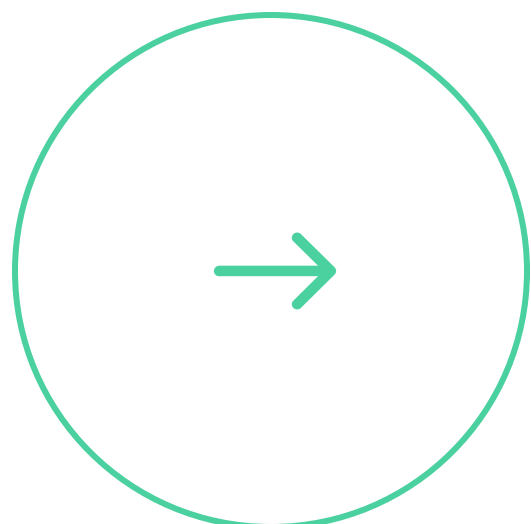


8

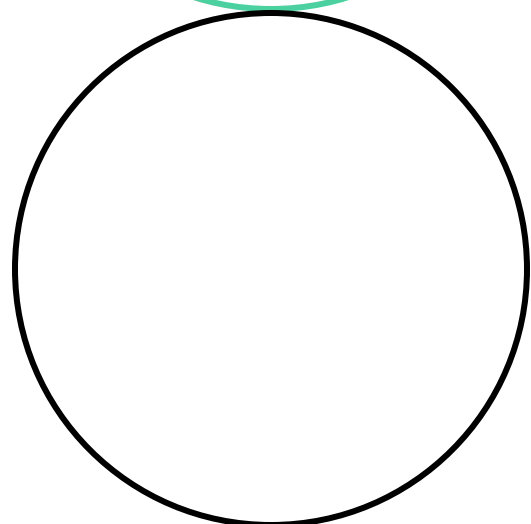
Николай Хащанов

Продвинутый SQL

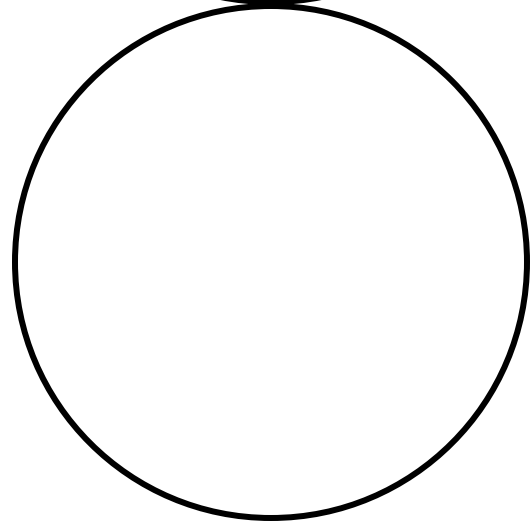
Подведем итог:



Рассмотрели различные формы
нормализации



Поговорили про OLTP-системы



Домашнее задание



9

Задание 1.

Спроектируйте базу данных, которая будет использоваться на предприятии в Системе Контроля Удаленного Доступа (СКУД), учете рабочего времени и начислении заработной платы.

База должна содержать следующие таблицы, которые должны иметь НФБК:

- Сотрудники
- Города
- Адреса (филиалов предприятия, проживания сотрудников)
- Timesheet (Учет рабочего времени)
- Начисление заработной платы

Задание 2 (на основе базы из задания 1).

СКУД система использует OLTP-систему, напишите основные запросы, которые будет выполнять OLTP-приложение (Сотрудник находится на рабочем месте – учитывается рабочее время). Так же будет необходимость каждый час формировать отчет по наличию сотрудников в каждом в филиале.

Дополнительное задание:

Реализуйте первое задание начиная с одной таблицы без нормализации и используйте каждую форму нормализации до НФБК



Полезные ссылки



10

Нормализация и формы:

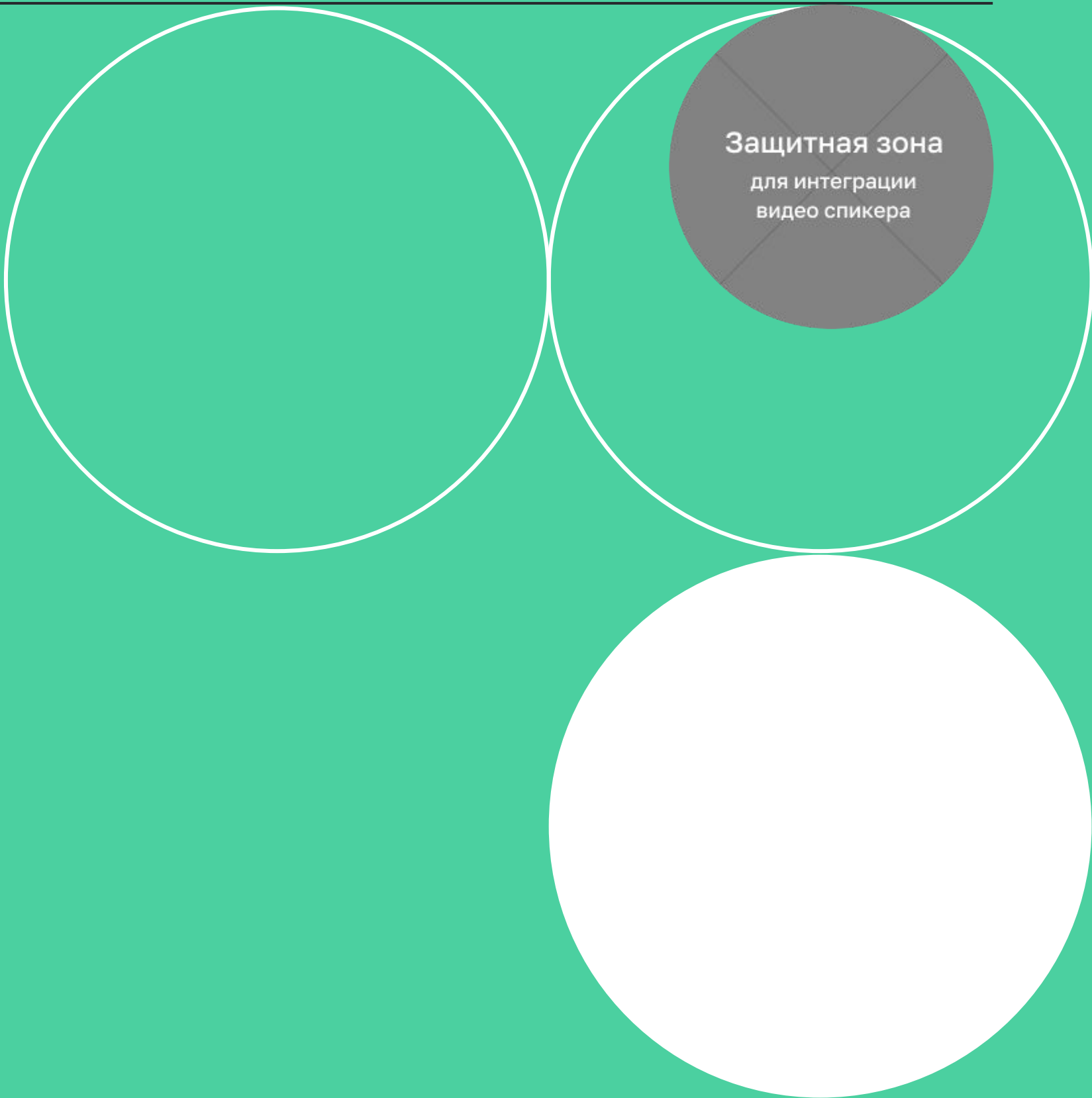
<https://habr.com/ru/post/254773/>





Спасибо за внимание!

Николай Хащанов



Защитная зона
для интеграции
видео спикера