Базы Данных

(нормализация базы данных, первичный и внешний ключи, SQL CREATE)

Андрей Голяков

Нормализация Базы Данных

Указания для правильного проектирования реляционных баз данных изложены в реляционной модели данных. Они собраны в пять групп, которые называются пятью нормальными формами.

1-ая нормальная форма представляет самый низкий уровень нормализации баз данных. 5-ый уровень представляет высший уровень нормализации.

Нормальные формы – это рекомендации по проектированию баз данных. Не всегда конкретная задача решается оптимально, если данные приведены к пятой нормальной форме. Тем не менее, рекомендуется нормализовать базу данных в некоторой степени потому, что этот процесс имеет ряд существенных преимуществ с точки зрения эффективности и удобства обращения с вашей базой данных.

Чем же полезна нормализация?

- Обеспечивает целостность данных (т.е., консистентное состояние)
- Предотвращает появление избыточности в хранимых данных
- Обеспечивает масштабируемость (до определённых пределов)
- Позволяет выполнять запросы на выборку и расчёт данных, используя однотипные подходы

Нормализация как процесс

- Упорядочивание данных в логические группы или наборы.
- Нахождение связей между наборами данных.
- Минимизация избыточности данных.

На практике к четвёртой и пятой форме базы данных почти никогда не приводят. Обычно базы данных нормализуются до второй или третьей нормальной формы.

Мы рассмотрим первые три нормальные формы на примере детально.

* Для более глубокого изучения данной темы рекомендуется к прочтению труд Кристофера Дейта “SQL и реляционная теория. Как грамотно писать код на SQL”, ISBN 978-5-93286-173-8.

Совместная работа (формулировка задачи)

Спроектировать БД информационной системы управления процессом отправки и получения корреспонденции небольшой организации.

1. Описываются отправители и получатели корреспонденции
    - Это люди с именами, должностями и адресами для доставки корреспонденции
2. Сама корреспонденция
    - Это бумажные документы с названиями, количеством страниц
3. Регистрируется момент времени изменения статуса процесса
    - Возможные статусы
        - ожидает отправки
        - отослано
        - вручено

Выделяем сущности

Отправитель
- ФИО
- Должность
- Адрес

Получатель
- ФИО
- Должность
- Адрес

Корреспонденция
- Наименование документа
- Количество страниц документа

Статусы отправлений и время
- Дата и время изменения статуса
- Статус

Снова термины (посложнее)

Атрибут — свойство некоторой сущности. Часто называется полем таблицы.

Домен атрибута — множество допустимых значений, которые может принимать атрибут.

Кортеж — конечное множество взаимосвязанных допустимых значений атрибутов, которые вместе описывают некоторую сущность (строка таблицы).

Отношение — конечное множество кортежей (таблица).

Схема отношения — конечное множество атрибутов, определяющих некоторую сущность. Иными словами, это структура таблицы, состоящей из конкретного набора полей.

1 НФ: Первая нормальная форма

Таблица является отношением в 1НФ, если

1. Отсутствует упорядочивание строк сверху вниз (другими словами, порядок строк не несет в себе никакой информации)
2. Отсутствует упорядочивание столбцов слева направо (другими словами, порядок столбцов не несет в себе никакой информации)
3. Отсутствуют повторяющиеся строки
4. Каждое пересечение строки и столбца содержит ровно одно значение из соответствующего домена (и больше ничего)

Про атомарность (из Википедии)

Многие авторы дополняют определение первой нормальной формы требованием *атомарности* (*неделимости*) значений.

Однако концепция “атомарности” является слишком неясной.

Например, многие типы данных (строки, даты, числа с фиксированной точкой и т. д.) при необходимости легко могут быть декомпозированы на составляющие элементы с помощью стандартных операций, предоставляемых СУБД.

Кристофер Дейт заключает, что “понятие атомарности не имеет абсолютно никакого смысла”.

И ещё немного терминов (level:nightmare)

Проекция — отношение, полученное из заданного путём удаления и (или) перестановки некоторых атрибутов.

Например, отношение (таблица) с только двумя полями “ФИО отправителя” и “Должность отправителя” будет проекцией нашего исходного отношения.

Функциональная зависимость между атрибутами (множествами атрибутов) X и Y означает, что для любого допустимого набора кортежей в данном отношении:

- если два кортежа совпадают по значению X, то они совпадают по значению Y.

Например, если мы посмотрим на нашу таблицу, то должность у получателя в данном примере в каждом кортеже атрибут “Должность отправителя” всегда находится в зависимости от атрибута “ФИО отправителя”.

Совместная работа

Приводим таблицу к 1НФ

2 НФ: Вторая нормальная форма

Отношение находится во 2НФ, если оно находится в 1НФ и

1. Каждый не ключевой атрибут неприводимо зависит от Первичного Ключа (ПК).
2. Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Первичный ключ — это один атрибут или наименьший набор атрибутов, однозначно определяющих один и только один кортеж.

Т.е. таблица должна обладать первичным ключом и все неключевые записи должны однозначно от него зависеть.

Совместная работа

Приводим таблицу к 2НФ

3 НФ: Тертья нормальная форма

Отношение находится в 3НФ, когда находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Проще говоря, правило требует выносить все неключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Совместная работа

Приводим таблицу к 3НФ

SQL-запросы для добавления ПК

Для существующей таблицы:

```
ALTER TABLE dbo.PostalItem
ADD CONSTRAINT PK_PostalItem PRIMARY KEY CLUSTERED (Id);
GO
```

При создании новой таблицы:

```
CREATE TABLE dbo.Position (
    Id INT NOT NULL,
    [Name] VARCHAR(250) NOT NULL,
    CONSTRAINT PK_Position PRIMARY KEY CLUSTERED (Id)
);
GO
```

SQL-запросы для манипуляций ПК

Для удаления ПК:

```
ALTER TABLE dbo.Position
DROP CONSTRAINT PK_Position;
```

Для изменения ПК:

```
ALTER TABLE dbo.Position
DROP CONSTRAINT PK_Position;
GO
ALTER TABLE dbo.PostalItem
ADD CONSTRAINT PK_PostalItem PRIMARY KEY CLUSTERED (Id);
GO
```

Внешний ключ (FOREIGN KEY)

Внешний ключ представляет собой одно из возможных ограничений. Оно ограничивает значения атрибута одного отношения набором значений потенциального ПК другого отношения.

Адрес

Идентификатор, ПК	Идентификатор Города	Адрес
1	1	ул. Большая Садовая, д. 10
2	2	ул. Садовая, д. 22

Город

Идентификатор, ПК	Название Города
1	1
2	2

SQL-запрос для создания ВК

Для существующей таблицы:

```
ALTER TABLE dbo.Address
ADD CONSTRAINT FK_Address_CityId FOREIGN KEY (CityId)
     REFERENCES dbo.City(Id)
     ON DELETE CASCADE
     ON UPDATE CASCADE
```

При создании новой таблицы:

```
CREATE TABLE dbo.Address (
     Id INT NOT NULL,
     CityId INT NOT NULL,
     [Address] VARCHAR(250) NOT NULL,
     CONSTRAINT PK_Address PRIMARY KEY CLUSTERED (Id),
     CONSTRAINT FK_Address_CityId FOREIGN KEY (CityId)
          REFERENCES dbo.City(Id));
```

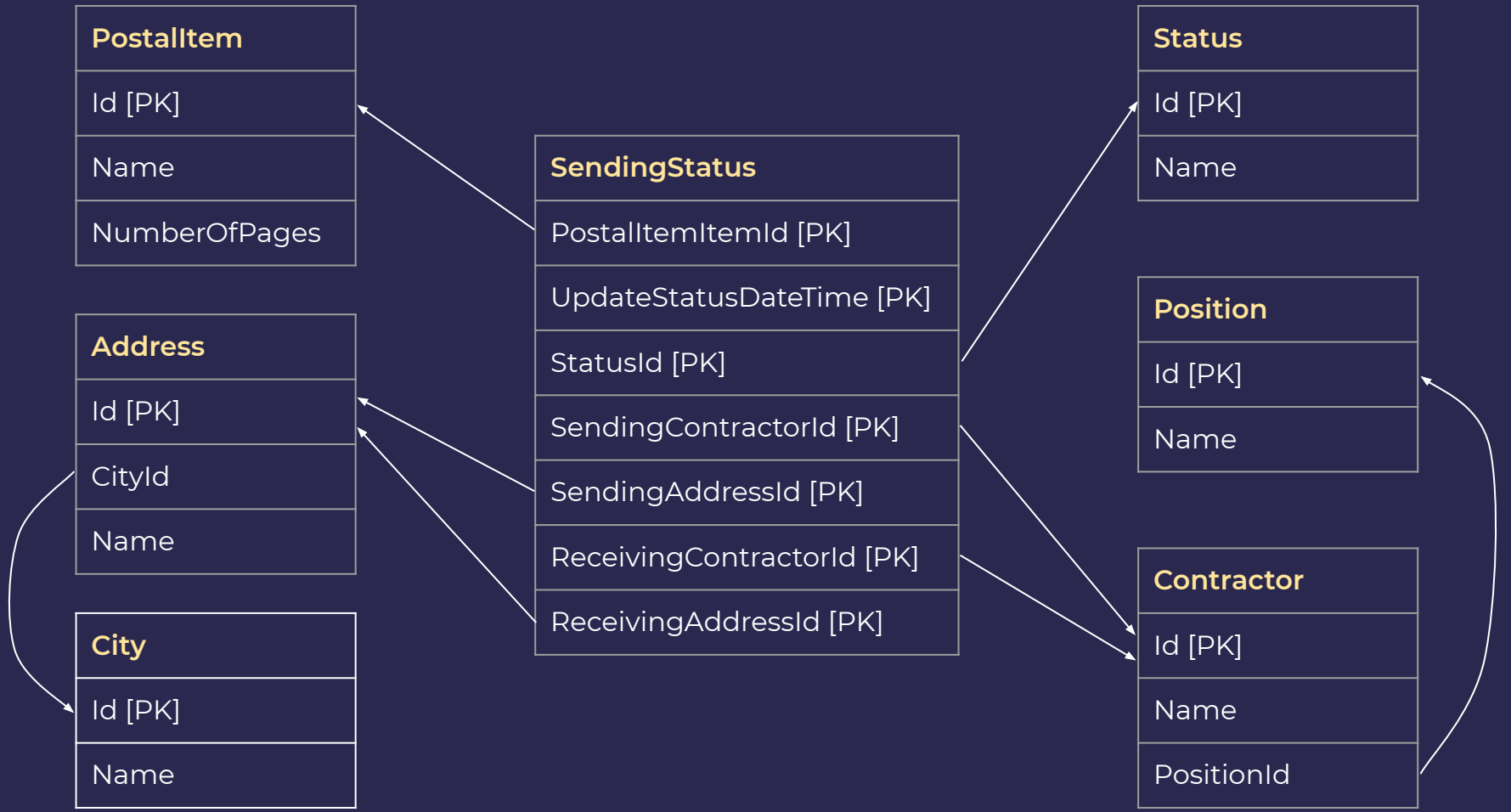
SQL-запросы для манипуляций ВК

Для удаления ПК:

```
ALTER TABLE dbo.Address
DROP CONSTRAINT FK_Address_CityId ;
```

Для изменения ПК:

```
-- DROP CONSTRAINT FK_Address_CityId...
-- ADD CONSTRAINT FK_Address_CityId...
```


PostalItem
Id [PK]
Name
NumberOfPages
Address
Id [PK]
CityId
Name
City
Id [PK]
Name
SendingStatus
PostalItemItemId [PK]
UpdateStatusDateTime [PK]
StatusId [PK]
SendingContractorId [PK]
SendingAddressId [PK]
ReceivingContractorId [PK]
ReceivingAddressId [PK]
Status
Id [PK]
Name
Position
Id [PK]
Name
Contractor
Id [PK]
Name
PositionId

Полезные ссылки

- [Habr] Как работает реляционная БД
- [Habr] Руководство по проектированию реляционных баз данных
  - (1-3, 4-6, 7-9, 10-13, 14-15)
- [Habr] Нормализация отношений. Шесть нормальных форм
- [MS] Создание связей по внешнему ключу

Домашняя работа

Спроектировать и написать SQL-запросы для создания схемы отношений хранилища нашего чат-бота.

Спасибо за внимание.