

Проектирование баз данных, ч.1

ФИО преподавателя: Смирнов Михаил Вячеславович
e-mail: smirnovmgupi@gmail.com

Лекция 2

Компетенции специалистов по базам данных. Нормализация. Группы инструкций SQL.

Что должен уметь специалист по базам данных в России

УТВЕРЖДЕН
приказом Министерства
труда и социальной защиты
Российской Федерации
от «17» сентября 2014 г. №647н

ПРОФЕССИОНАЛЬНЫЙ СТАНДАРТ

Администратор баз данных

146

Регистрационный номер

I. Общие сведения

Поддержание эффективной работы баз данных, обеспечивающих
функционирование информационных систем в организации
(наименование вида профессиональной деятельности)

06.011

Код

Принципы нормализации данных в реляционной модели

Будем смотреть принципы нормализации вплоть до 3NF. Остальные формы можно подробно изучить или в учебнике Кренке (см. литературу первой лекции дисциплины) или, например, тут:

http://citforum.ru/database/osbd/glava_23.shtml (до 5 NF)

https://ru.wikipedia.org/wiki/Нормальная_форма (родная Вики, до 6NF)

Заходите первого пациента...

Во дворце спорта университета ведем базу данных о студентах, посещающих спортивные кружки. У каждой секции есть название (идентичное виду спорта) И оплата, которую студент должен внести за каждое занятие. Важно! В этом примере каждый студент может посещать только одну секцию (надорвется еще, бедняга). Студентов много, места на слайде мало, поэтому, я привел тут лишь часть сущности, которая хранится у нас в воображаемой базе данных.

→ Это весьма годный первичный ключ

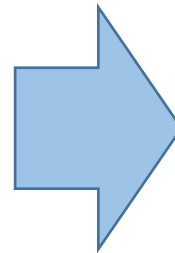
СЕКЦИЯ (НомерСтудента, Секция, Плата)

НомерСтудента	Секция	Плата за занятие
100	Лыжи	100
150	Плавание	300
175	Футбол	150
200	Плавание	300

Аномалия удаления (DA, deletion anomaly)

Посмотрим, что произойдет, когда студент с номером 100 надумает закончить свою карьеру лыжника. Администратор БД будет вынужден вычеркнуть этого студента из списка текущих спортсменов и мы получим следующий результат:

НомерСтудента	Секция	Плата за занятие
100	Лыжи	100
150	Плавание	300
175	Футбол	150
200	Плавание	300



НомерСтудента	Секция	Плата за занятие
150	Плавание	300
175	Футбол	150
200	Плавание	300

Отлично! Теперь мы никогда не узнаем, сколько у нас стоит занятие в лыжной секции.

Аномалия вставки (IA, insertion anomaly)

А теперь так – оказывается, у нас есть секции самбо, шитья и настольного тенниса, которые только-только успели открыться и на них еще никто не записался, и вот идут студенты записываться на занятия спортом и спрашивают финансовые условия занятий самбо:

НомерСтудента	Секция	Плата за занятие
100	Лыжи	100
150	Плавание	300
175	Футбол	150
200	Плавание	300

САМБО?

Информация о стоимости посещения занятий самбо не появится до тех пор, пока хотя бы один студент (его номер – первичный ключ экземпляра) не запишется на эти занятия. А он не запишется, пока не получит информацию о стоимости посещения... ой, все!

Переход 1NF -> 2NF

Вся проблема заключается в том, что от того, какой номер будет у студента, разницы в оплате занятий не будет. То есть атрибуты Плата и НомерСтудента не связаны друг с другом. А НомерСтудента то первичный ключ и определяет все свойства экземпляра сущности. Проблема будет решена, когда будут получены сущности, атрибуты которых будут зависеть от ключа.

Студент ходит на секцию.

СТУДЕНТ-СЕКЦИЯ
(НомерСтудента, Секция)

НомерСтудента	Секция
100	Лыжи
150	Плавание
175	Футбол
200	Плавание

Посещение секции стоит денег.

СЕКЦИЯ-ПЛАТА
(Секция, Плата)

Секция	Плата
Лыжи	100
Плавание	300
Футбол	150

Следующий пациент. Проблема 2NF. Транзитивная зависимость (TD, transitive dependency)

Отдел ВУЗа по работе с иногородними распределяет студентов по трем общежитиям. В отличие от того, где они располагаются и какие в них условия, студент платит за размещение разную сумму денег. Таблица для сохранения данных выглядит так:

НомерСтудента	Общежитие	Плата
100	Первое	6200
150	Второе	6000
200	Первое	6200
250	Третье	6100
300	Первое	6200

У этой таблицы нет проблем с точки зрения 2NF. Ключ НомерСтудента связан и с атрибутом общежитие (он там проживает) и с атрибутом плата (он эти деньги выплачивает за проживание). Но проблемы с аномалиями DA и IA все-таки есть.

Следующий пациент. DA и IA во второй нормальной форме

DA все также появляется, когда мы выпишем экземпляр студента, который Единственный числится в каком-либо общежитии. IA появится тогда, когда в наш фонд жилья для студентов поступит еще одно или несколько общежитий.

НомерСтудента	Общежитие	Плата
100	Первое	6200
150	Второе	6000
200	Первое	6200
250	Третье	6100
300	Первое	6200

НомерСтудента	Общежитие	Плата
100	Первое	6200
150	Второе	6000
200	Первое	6200
250	Третье	6100
300	Первое	6200
???	Четвертое	5800

Проблема тут заключается в том, что атрибут плата зависит не только от ключа НомерСтудента, но и от неключевого атрибута Общежитие. Такая зависимость называется транзитивной и приводит к возникновению проблем DA и IA.

Переход 2NF -> 3NF

Лечим проблему отловом всех неключевых атрибутов, которые успели в нашей таблице подружиться за спиной у первичного ключа. И да, нам придется их разлучить по разным таблицам.

СТУДЕНТ-ПРОЖИВАНИЕ

НомерСтудента	Общежитие
100	Первое
150	Второе
200	Первое
250	Третье
300	Первое

ОБЩЕЖИТИЕ-ПЛАТА

Общежитие	Плата
Первое	6200
Второе	6000
Третье	6100

Следующий пациент для закрепления. Транзитивная зависимость, раунд два.

Сейчас будет немного посложнее. Эта таблица тоже не имеет проблем с 2NF, но имеет транзитивную зависимость. Предлагаю самим определить, в чем она заключается.

id	f_name	s_name	province	city	postal_code
23	Khalid	Boulahrouz	Noord	Alkmaar	1825HH
24	Zinedine	Zidane	Noord	Langedijk	1834DK
25	Ruud	Nistelrooy	Noord	Schermer	1844JJ
26	Philip	Cocu	Noord	Heilo	1850WI

На следующем слайде покажу как я вижу решение этой аномалии.

Лечим таблицу с футболистами. 2NF -> 3NF

Суть проблемы заключается в том, что атрибуты city и province привязаны не только к первичному ключу, но и к атрибуту postal_code.

id	f_name	s_name	postal_code
23	Khalid	Boulahrouz	1825HH
24	Zinedine	Zidane	1834DK
25	Ruud	Nistelrooy	1844JJ
26	Philip	Cocu	1850WI

postal_code	province	city
1825HH	Noord	Alkmaar
1834DK	Noord	Langedijk
1844JJ	Noord	Schermer
1850WI	Noord	Heilo

Язык SEQUEL (SQL)

Формальный не процедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД). SQL основывается на исчислении кортежей. *

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций: создание в базе данных новой таблицы, добавление в таблицу новых записей, изменение записей, удаление записей, выборка записей из одной или нескольких таблиц (в соответствии с заданным условием), изменение структур таблиц.

Со временем SQL усложнился — обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры)...

* [Материал из Википедии - свободной энциклопедии](#)

Группы инструкций SQL

Инструкции определения данных
(*Data Definition Language*, [DDL](#))

Инструкции манипуляции данными
(*Data Manipulation Language*, [DML](#))

Инструкции определения доступа к
данным (*Data Control Language*, [DCL](#))

Инструкции управления транзакциями
(*Transaction Control Language*, [TCL](#))

Инструкции определения данных (DDL)

Используются в СУБД для создания и изменения структуры базы данных и ее составных частей - таблиц, индексов, представлений (виртуальных таблиц), а также триггеров и хранимых процедур.

Группа CREATE

CREATE database

CREATE table

CREATE view

CREATE index

CREATE trigger

CREATE procedure

Группа ALTER

ALTER database

ALTER table

ALTER view

ALTER index

ALTER trigger

ALTER procedure

Группа DROP

DROP database

DROP table

DROP view

DROP index

DROP trigger

DROP procedure

Инструкции определения данных (DDL). Типовой скрипт создания БД.

```
CREATE DATABASE test_createdb
ON
( NAME = crdb_dat,
  FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL13.FORLAB\MSSQL\DATA\crdb_dat.mdf',
  SIZE = 30MB, MAXSIZE = 50MB,
  FILEGROWTH = 20%
)
LOG ON
( NAME = crdb_log,
  FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL13.FORLAB\MSSQL\DATA\crdb_log.ldf',
  SIZE = 5MB,
  MAXSIZE = UNLIMITED,
  FILEGROWTH = 2MB
);
```

Инструкции манипулирования данными (DML)

Используются, для манипулирования данными в таблицах баз данных.

SELECT считывает данные, удовлетворяющие заданным условиям.

INSERT добавляет новые данные.

UPDATE изменяет существующие данные.

DELETE удаляет данные.

Инструкции определения доступа к данным (DCL)

Используется для управления правами доступа к данным и выполнением процедур в многопользовательской среде. Более точно его можно назвать "язык управления доступом".

GRANT предоставляет пользователю (группе) разрешения на определенные операции с объектом.

REVOKE отзывает ранее выданные разрешения.

DENY задает запрет, имеющий приоритет над разрешением.

Инструкции определения доступа к данным (DCL).
Типовые скрипты группы GRANT/REVOKE.

```
GRANT INSERT ON dbo.paintings TO test;
```

```
GRANT SELECT ON dbo.paintings TO test;
```

```
REVOKE INSERT ON dbo.paintings TO test AS dbo;
```

```
REVOKE SELECT ON dbo.paintings TO test AS dbo;
```

Инструкции определения доступа к данным (DCL). Типовой скрипт группы DENY.

USE SAMPLE;

GRANT SELECT ON project TO public;

DENY SELECT ON project TO user_1, user_2;

Инструкции управления транзакциями (TCL)

Компьютерный язык и часть SQL, используемый для обработки транзакций.

COMMIT применяет транзакцию.

ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции.

SAVEPOINT делит транзакцию на более мелкие участки.

Инструкции управления транзакциями (TCL). Типовой скрипт транзакции.

BEGIN TRANSACTION

INSERT INTO work VALUES

(001, 'Севастополь', 'На картине изображена бухта города в утренней дымке', 001),

(002, 'Шхуна под парусами', 'На картине изображена шхуна с флагом', 001),

(003, 'Ялта. Горы ночью', 'Группа людей прогуливаются вдоль берега Крыма', 001);

IF (@@error <> 0)

ROLLBACK

ELSE

COMMIT;

Чтение на дом

- Книга «Системы баз данных», стр. 123-147 (нормализация), 249-290 (повтор SQL), 393-417 (транзакции и управление доступом к БД).
- Русский Кренке, стр. 468-499.
- Английский Кренке, 334-388.

Спасибо за внимание!