



# PostgreSQL: промышленная разработка баз данных

## Лекция 8

*Проектирование баз данных  
Нормализация и денормализация схемы.*

# Процесс проектирования схемы БД

- «Правильный» подход №1

анализ предметной области → ER-диаграмма → физ. диаграмма → SQL(DDL)

- «Правильный» подход №2

анализ предметной области → UML-диаграмма → физ. диаграмма → SQL(DDL)

- На практике:

физ. диаграмма → SQL(DDL)

*а иногда и сразу SQL...*

# Инструменты

- **Инструменты проектирования**
  - PgDesigner ([pgdesigner.sf.net](http://pgdesigner.sf.net))
  - Sybase Power Designer (\$)
  - ERWin (\$)
- **Инструменты администрирования/разработки**
  - pgAdmin-III
  - phpPgAdmin
  - EMS SQL Manager (\$)

# Нормализация

- Цель — уход от аномалий:
  - избыточность
  - аномалии изменения
  - аномалии удаления
- На практике достаточно 3NF:
  - таблица находится в 2NF (любой неключевой атрибут функционально полно зависит от каждого ключа-кандидата)
  - нет транзитивных зависимостей неключевых атрибутов

# Нормализация: 3NF

```
CREATE TABLE person(  
    person_id INTEGER PRIMARY KEY,  
    person_name VARCHAR(100),  
    person_position VARCHAR(32),  
    person_salary INTEGER  
);
```



```
CREATE TABLE position(  
    position_id INTEGER PRIMARY KEY,  
    position_name VARCHAR(32),  
    position_salary INTEGER  
);
```

```
CREATE TABLE person(  
    person_id INTEGER PRIMARY KEY,  
    person_name VARCHAR(100),  
    person_position_id INTEGER  
        REFERENCES position(position_id)  
);
```

# Денормализация

- Цель — повышение производительности:
  - работа с меньшим кол-вом таблиц
  - меньше соединений (JOIN)
  - (в больших проектах) горизонтальное партиционирование (sharding, user clusters)

# Задания

## 1. Спроектируйте БД для веб-системы типа «Блог»

- сущности: пользователь, блог, тема, комментарий)
- постарайтесь более полно описать сущности (больше столбцов) и использовать развитую систему ограничений целостности (типы данных, UK, FK, NOT NULL / DEFAULT, CHECK constraints, etc)
- результатом выполнения задания должна быть последовательность SQL-команд (DDL) для создания схемы БД

## 2. Заполните таблицы тестовыми данными

- используйте функции `generate_series(...)`, `random()`
- минимальное кол-во строк во всех таблицах:  $10^3$
- результатом выполнения задания должна быть последовательность SQL-команд для наполнения данными

# Задания

3. Определите, в каких ИФ находятся ваши таблицы

4. Создайте SQL-запрос для выборки 10 самых свежих тем, вместе с авторами

- результат должен включать: название темы, тело темы (сообщение), имя автора, дата создания темы

5. Создайте SQL-запрос для выборки 10 самых активных авторов

- результат должен включать: имя автора, количество тем

6. Проанализируйте запросы из пп. 4 и 5

- используйте `EXPLAIN ANALYZE`

# Задания

7. Постарайтесь улучшить производительность запросов из пп. 4 и 5 за счёт использования индексов

- создайте необходимый набор индексов и докажите, что производительность улучшилась (используя `EXPLAIN ANALYZE`)

8. Можно ли улучшить производительность, используя денормализацию?

- не увлекайтесь :-)

# Литература:

## рекомендации для данной лекции

- Гарсиа-Молина, Г., Ульман, Дж., Уидом, Дж. Системы баз данных. Полный курс.
  - Глава 2. Модель данных «сущность-связь»
  - Глава 3. Реляционная модель (**3.6 Проектирование реляционных систем**)
- Кузнецов С. Д. Основы современных баз данных  
<http://citforum.ru/database/osbd/contents.shtml>
  - **Лекция 6. Проектирование реляционных БД**
- Дейт, К.. Введение в системы баз данных
  - Часть III. Проектирование базы данных

# Контакты

- [nikolay@samokhvalov.com](mailto:nikolay@samokhvalov.com)
- Blog: <http://nikolay.samokhvalov.com>
- XMPP/GTalk: [samokhvalov@gmail.com](mailto:samokhvalov@gmail.com)
- Skype: samokhvalov & postgresmen
- +7 905 783 9804