

## Лабораторная работа № 3. Создание БД для аналитики

РСУБД часто используются в аналитических целях. В этом случае объем данных постоянно увеличивается. При этом их изменение или удаление затруднено из-за большого объема. Хранение данных в денормализованном виде с помощью массивов и json-формата упрощает их обработку. В этом случае с СУБД взаимодействует аналитик, который имеет доступ не ко всем данным. Это может быть неудобно работать с большими объемами данных и запросами. Для сложно заранее предугадать все задачи, которые перед ним будут стоять. При этом разработчику такой базы данных необходимо уметь проектировать высокопроизводительную схему данных ориентированную на большие объемы и регулярно появляющиеся и устаревающие данные, использовать механизмы индексации и построения индексов, анализировать планы запросов и оптимизировать их, упрощать интерфейс базы данных с помощью процедур и представлений, ограничивать доступ с помощью ролей и прав.

Данная лабораторная работа призвана сформировать у студента понимание особенностей создания аналитических баз данных и умение их настраивать и поддерживать.

### Задание 1. Проектирование схемы базы данных

#### Постановка задачи

Задание связано с проектированием схемы базы данных для аналитики. Будем исходить из того, что приложение, для которого была сделана база данных в лабораторной работе стало очень популярным и по нему каждый день можно собирать большой объем статистической информации. Что это будет за статистика? Почему именно ее необходимо собирать, обрабатывать и анализировать? Задачей студента является ответить на эти вопросы, и, исходя из этого, разработать базу данных и заполнить ее данными. Результатом данного задания является схема базы данных, скрипты создания базы данных и ее наполнения, обладающие следующими

- Как минимум одна таблица должна содержать не меньше 100 млн. записей, которые временем теряют актуальность.
- Другая таблица, связанная с первой, должна содержать не меньше 1 млн. записей.
- В одной из таблиц с количеством записей больше 1 млн. должна быть колонка с текстом, по которой будет необходимо настроить полнотекстовый поиск.
- В одной из таблиц с количеством записей больше 1 млн. должна быть колонка с данными в формате json.
- В одной из таблиц с количеством записей больше 1 млн. должна быть колонка с данными в формате массива.

При выполнении задания важно учитывать плюсы и минусы денормализации схемы данных и использования массивов и json-формата. При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Для проектирования схемы и построения диаграммы можно использовать любые средства, один из вариантов использовать сайт:

<https://www.lucidchart.com/pages/examples/er-diagram-tool>

## Темы для проработки

- Денормализация  
<https://habr.com/ru/company/laterna/blog/281262/>  
<https://habr.com/ru/post/64524/>
- Массивы  
<https://postgrespro.ru/docs/postgrespro/11/arrays>  
<https://postgrespro.ru/docs/postgrespro/11/functions-array>
- Json  
<https://postgrespro.ru/docs/postgrespro/11/datatype-json>  
<https://postgrespro.ru/docs/postgrespro/11/functions-json>
- Наполнение базы данных  
<https://postgrespro.ru/docs/postgrespro/11/populate>

## Задание 2. Управление доступом

### Постановка задачи

Целью седьмого практического задания является освоение работы с представлениями и другими способами управления доступом. При выполнении задания необходимо:

- Создать пользователя test и выдать ему доступ к базе данных.
- Составить и выполнить скрипты присвоения новому пользователю прав доступа к таблицам, созданным в практическом задании 1. При этом права доступа к различным таблицам должны быть различными, а именно:
  - По крайней мере, для одной таблицы новому пользователю присваиваются права SELECT, INSERT, UPDATE в полном объеме.
  - По крайней мере, для одной таблицы новому пользователю присваиваются права SELECT и UPDATE только избранных столбцов.
  - По крайней мере, для одной таблицы новому пользователю присваивается только право SELECT.
- Создать стандартную роль уровня базы данных, присвоить ей право доступа (UPDATE на некоторые столбцы) к представлению, созданному в практическом задании №3.3, назначить новому пользователю созданную роль.
- Выполнить от имени нового пользователя некоторые выборки из таблиц и представления. Убедиться в правильности контроля прав доступа.
- Выполнить от имени нового пользователя операторы изменения таблиц с ограниченными правами доступа. Убедиться в правильности контроля прав доступа.
- Составить SQL-скрипты для создания нескольких представлений, которые позволяли бы упростить манипуляции с данными или позволяли бы ограничить доступ к данным, предоставляя только необходимую информацию.

## Темы для проработки

- Роли и пользователи.  
<https://postgrespro.ru/docs/postgresql/11/user-manag>  
<https://postgrespro.ru/docs/postgrespro/11/app-createuser>
- Директивы GRANT и REVOKE.  
<https://postgrespro.ru/docs/postgrespro/11/ddl-priv>  
<https://postgrespro.ru/docs/postgrespro/11/ddl-schemas#DDL-SCHEMAS-PRIV>
- Представления.  
<https://postgrespro.ru/docs/postgrespro/11/sql-createview>  
<https://postgrespro.ru/docs/postgrespro/11/rules-views>  
<https://postgrespro.ru/docs/postgrespro/11/rules-materializedviews>
- Полное описание синтаксиса встретившихся команд  
<https://postgrespro.ru/docs/postgrespro/11/sql-commands>

## Примеры вопросов

- Для чего нужны роли?
- Что такое схема?
- Рассказать про директивы GRANT и REVOKE.
- Для чего нужна роль PUBLIC?
- Как добавить нового пользователя в текущую базу данных?
- Как позволить пользователю заходить на сервер?
- Какие существуют права?
- Исправить ошибки в обязательной части.
- Сменить владельца базы данных.
- Сменить пароль для пользователя.
- Определить роль с заданными правами.
- Объяснить, как работают написанные запросы.
- Рассказать о CHECK OPTION.
- Рассказать о модификации данных через представления.
- Рассказать о вставке данных через представления.
- Примеры вопросов по оператору SELECT см. в задании №1.
- Исправить неверно работающий запрос (запросы).
- Упростить один или несколько запросов.
- Продемонстрировать изменение и вставку данных через представления.
- Написать или модифицировать запрос по сформулированному заданию.
- Продемонстрировать полезность материализованного представления.