

## Учебно-тематический план программы повышения квалификации

«Информационная система на PostgreSQL»

**Программа соответствует квалификационным требованиям:**

- Профессионального стандарта 06.001 «Программист», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н, зарегистрирован в Министерстве юстиции Российской Федерации 22 августа 2022 г. N 69720 (уровень квалификации 5 – 6).

**Форма обучения** – очная, очно-заочная (в т.ч. с использованием дистанционных технологий).

**Срок обучения** (количество часов) – 72 академических часа

**Режим занятий** – не менее 30 академических часов в неделю

**Категория слушателей** – студенты бакалавриата технических и естественнонаучных специальностей, преподаватели дисциплин, связанных с применением современных цифровых технологий, научные работники и аспиранты профильных кафедр, а также сотрудники соответствующих ИТ служб компаний.

Наименование темы	Общее количество часов	Контактная работа		Самостоятельная работа, ч	Формы текущего контроля
		Лекции, ч	Практические занятия, ч		
Тема 1. Основы логического моделирования баз данных	9	2	4	3	Отчет по ПР
Тема 2. Нормализация баз данных	10	2	4	4	Отчет по ПР
Тема 3. Физическая реализация базы данных PostgreSQL	8	-	4	4	Отчет по ПР
Тема 4. Обеспечение целостности базы данных	9	2	3	4	Отчет по ПР
Тема 5. Основы выполнения SQL запросов	9	1	4	4	Отчет по ПР
Тема 6. Использование расширенных методов SQL запросов для доступа к данным	9	1	4	4	Отчет по ПР
Тема 7. Создание функций, хранимых процедур и триггеров	9	1	4	4	Отчет по ПР

Тема 8. Администрирование PostgreSQL	7	1	2	4	Отчет по ПР
Итоговая аттестация	2	-	2	-	Зачет
ИТОГО:	72	18	18	36	

## Тема 1. Основы логического моделирования баз данных.

Лекционное занятие: В данном разделе представлен обзор четырех основных фаз проектирования реляционных баз данных: концептуальной, логической, реализации и физической. По соображениям экономии времени и бюджета часто возникает соблазн пропустить предыдущие фазы проектирования базы данных и сразу перейти к фазе реализации. Я объясняю, почему пропуск любого или всех этих этапов может привести к неполному и/или неправильному проектированию, а также к тому, что оно не будет поддерживать высокопроизводительные запросы и отчеты. Я дам краткое описание некоторых фундаментальных объектов базы данных, включая саму базу данных, а также таблицы, столбцы и ключи. Эти объекты, скорее всего, знакомы большинству, но есть несколько распространенных ошибок в их использовании, которые могут сделать разницу между посредственным дизайном и высококлассным, профессиональным дизайном. В частности, непонимание важной роли ключей в базе данных может привести к серьезным проблемам с целостностью данных, а также к ошибочному мнению, что такие ключи и ограничения могут быть эффективно реализованы вне базы данных. Я кратко рассмотрю различные типы бинарных и небинарных отношений, которые могут существовать между реляционными таблицами. Я расскажу о понятиях зависимостей между значениями и о том, как они определяют процесс проектирования баз данных.

Практическое занятие: Создание логической модели базы данных. Определение объектов предметной области, их атрибуты, ограничения, типы связи между объектами. Для моделирования используется он-лайн инструмент draw.io  
Решение задач и практических упражнений на основе изученных методов.

## Тема 2. Нормализация баз данных.

Лекционное занятие: Я подробно рассмотрю многочисленные причины, по которым необходимо нормализовать данные. В конечном итоге нормализация необходима для повышения эффективности и защиты целостности реляционных данных. Нормализация оптимизирует базу данных для эффективного хранения и обновления, а не для выполнения запросов. Это значительно снижает вероятность появления аномалий обновления (разные записи отображают разные значения для одного и того же фрагмента данных), но увеличивает сложность запросов, поскольку приходится собирать данные из множества различных таблиц. Далее я рассмотрю поочередно каждую из нормальных форм, объясню на наглядных примерах требования каждой из них, программные аномалии, которых они помогают избежать, и признаки того, что реляционные данные нарушают ту или иную нормальную форму. В заключение я расскажу о лучших практиках нормализации. Нормализация может быть суммирована в единственном предложении: "Каждая сущность в вашей БД должна описывать единственный объект или процесс".

Практическое занятие: устранение аномалий в логической схеме базы данных. Приведение модели к нормальной форме Бойса-Кодда. Решение задач и практических упражнений на основе изученных методов.

### **Тема 3. Физическая реализация базы данных PostgreSQL.**

Практическое занятие: создание базы данных, таблиц в PostgreSQL. На этом этапе происходит адаптация логической модели для реализации в принимающей реляционной системе управления базами данных (РСУБД; в нашем случае - PostgreSQL). Создается модель, в которой данные отображаются на физические дисковые структуры. На этапе реализации логический проект адаптируется к используемому инструменту (в нашем случае это PostgreSQL). Это включает в себя выбор типов хранения, построение таблиц, наложение ограничений, написание триггеров и т.д. для наиболее эффективной реализации логической модели.

### **Тема 4. Обеспечение целостности базы данных.**

Лекционное занятие: Целостность данных относится к состоянию, в котором все значения данных, хранящихся в базе данных являются правильными. Целостность объектов определяет строку конкретной таблицы как уникальный экземпляр объекта. Целостность домена это условие правильности данных для столбца. Ссылочная целостность сохраняет определенные связи между таблицами, при вставке и удалении строк в одну из таблиц.

Практическое занятие: Реализация целостности объектов, целостности домена и ссылочной целостности. Решение задач и практических упражнений на основе изученных методов.

### **Тема 5. Основы выполнения SQL запросов.**

Лекционное занятие: В этом разделе представлены этапы логической обработки запроса. Сначала я кратко опишу каждый этап CRUD. Понимание фаз обработки SQL-запросов и уникальных аспектов языка SQL важно для того, чтобы сформировать особое мышление, необходимое для программирования на SQL. Зная эти аспекты языка, вы сможете создавать эффективные решения и объяснять свой выбор. Идея заключается в том, чтобы овладеть основами SQL-запроса.

Практическое занятие: практика выполнения оператора SELECT. Воплотите свой замысел в жизнь с помощью SQL - стандартного языка для работы с базами данных. Узнайте, как создавать, манипулировать и запрашивать базы данных с помощью SQL. Решение задач и практических упражнений на основе изученных методов.

### **Тема 6. Использование расширенных методов SQL запросов для доступа к данным**

Лекционное занятие: Лекция и практика нацелены на то, чтобы познакомить слушателей с расширенными методами SQL запросов, такими как объединение таблиц и результатов запросов, а также использование подзапросов для более гибкого доступа к данным. Расширенные методы SQL запросов для работы с данными: Объединение таблиц: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN. Объединение результатов запросов: UNION, INTERSECT, EXCEPT. Комбинирование операторов: использование скобок и операторов

AND, OR. Подзапросы в SQL: скалярные, возвращающие одно значение, и множественные, возвращающие наборы данных. Применение подзапросов в различных частях SQL запроса: SELECT, FROM, WHERE, HAVING, EXISTS. Оптимизация подзапросов и использование алиасов.

Практическое занятие: Работа с операторами объединения таблиц: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN. Использование операторов объединения результатов запросов: UNION, INTERSECT, EXCEPT. Практические примеры работы с подзапросами в различных частях SQL запроса. Оптимизация запросов с использованием подзапросов и алиасов. Решение задач и практических упражнений на основе изученных методов.

## **Тема 7. Создание функций, хранимых процедур и триггеров**

Лекционное занятие: Введение в функции, хранимые процедуры и триггеры. Различия между функциями, хранимыми процедурами и триггерами. Синтаксис создания функций. Возвращаемые значения функций. Параметры функций. Примеры использования функций. Синтаксис создания хранимых процедур. Параметры хранимых процедур. Выполнение хранимых процедур. Примеры использования хранимых процедур. Синтаксис создания триггеров. Типы триггеров: BEFORE и AFTER. Триггеры на INSERT, UPDATE и DELETE операции. Примеры использования триггеров.

Практическое занятие: Создание функции для вычисления среднего значения. Создание функции для конвертации валюты. Тестирование функций на примере данных. Создание процедуры для добавления новой записи. Создание процедуры для обновления записи. Вызов и тестирование хранимых процедур. Создание триггера для автоматического обновления данных. Создание триггера для проверки ограничений. Тестирование триггеров на примере операций INSERT, UPDATE и DELETE.

## **Тема 8. Администрирование PostgreSQL**

Лекционное занятие: В заключительной главе рассматриваются некоторые вопросы, связанные с управлением базами данных. Вы узнаете, как предоставлять доступ к базе данных на основе ролей пользователей, как разбивать таблицы на более мелкие части, о чем следует помнить при интеграции данных и какая СУБД лучше всего соответствует потребностям вашего бизнеса.

Практическое занятие: Работа с пользователями и правами доступа: Создание и удаление пользователей. Назначение прав доступа для пользователей. Управление ролями и привилегиями. 4. Резервное копирование и восстановление данных: Создание резервных копий базы данных. Восстановление данных из резервной копии. Автоматизация процесса резервного копирования. 5. Мониторинг и оптимизация производительности: Использование инструментов мониторинга производительности. Оптимизация запросов и индексов. Настройка параметров для повышения производительности.

## **Итоговая аттестация. Зачет.**