

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина «Модели данных и системы управления базами данных»

**ОТЧЕТ**  
к лабораторной работе №4  
на тему:  
**«РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С  
ИСПОЛЬЗОВАНИЕМ БАЗЫ ДАННЫХ»**  
БГУИР 6-05-0612-02

Выполнил студент группы 353503  
КОХАН Артём Игоревич  
РЯЗАНЦЕВ Алексей Владимирович

---

(дата, подпись студента)

Проверил ассистент каф. информатики  
КОЖЕМЯКО Евгения Александровна

---

(дата, подпись преподавателя)

Минск 2025

## **1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

Реализация запросов по манипуляции данными (insert, update, delete).  
Результат работы: набор SQL-запросов с описанием.

## 2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ

Манипуляция данными в реляционных базах данных осуществляется с помощью операторов DML (Data Manipulation Language), которые позволяют работать с содержимым таблиц. Основные операции включают добавление, изменение и удаление записей.

Оператор INSERT используется для добавления новых записей в таблицу. Существует несколько способов вставки данных: добавление одной записи с указанием значений для всех столбцов, добавление записи с указанием конкретных столбцов, а также массовая вставка нескольких записей одновременно. При вставке данных важно соблюдать типы данных и ограничения, определенные в структуре таблицы.

Оператор UPDATE предназначен для изменения существующих записей в таблице. Он позволяет модифицировать значения одного или нескольких столбцов для выбранных строк. Обязательным элементом UPDATE является условие WHERE, которое определяет, какие именно записи должны быть обновлены. Без условия WHERE будут обновлены все строки таблицы, что может привести к серьезным последствиям.

Оператор DELETE используется для удаления записей из таблицы. Как и в случае с UPDATE, условие WHERE является критически важным для точного указания удаляемых записей. При отсутствии условия WHERE будут удалены все данные из таблицы.

Все операции манипуляции данными поддерживают транзакции, что позволяет объединять несколько операций в одну атомарную единицу работы. Это обеспечивает целостность данных – либо выполняются все операции транзакции, либо не выполняется ни одна из них. В PostgreSQL также доступен оператор TRUNCATE для быстрого удаления всех записей из таблицы с автоматическим сбросом последовательностей.

При работе с операциями изменения данных важно учитывать ограничения целостности, установленные между таблицами, особенно внешние ключи (FOREIGN KEY), которые могут ограничивать или каскадно распространять операции удаления и обновления на связанные таблицы.

### 3 ВЫПОЛНЕНИЕ РАБОТЫ

Представленный код с использованием фреймворка Spring Boot демонстрирует реализацию операций манипуляции данными для системы управления спортивным залом. Работа с базой данных PostgreSQL организована через JdbcTemplate, что обеспечивает прямое выполнение SQL-запросов.

Операции вставки данных реализованы с помощью INSERT-запросов для всех основных сущностей системы (листинг кода 3.1). Для таблицы locker\_assignment выполняется добавление назначений шкафчиков с указанием идентификатора шкафчика, пользователя и даты бронирования. Вставка записей в таблицу locker предусматривает указание номера, гендерной принадлежности и статуса шкафчика с явным приведением типов к пользовательским ENUM-типам PostgreSQL. Добавление абонементов в таблицу membership включает заполнение информации о клиенте, названии, описании, цене и периоде действия.

Операции обновления данных реализованы через UPDATE-запросы для таблиц membership, locker и personal\_training (листинг кода 3.2). Каждый запрос включает условие WHERE для идентификации изменяемой записи по идентификатору. Для таблиц locker и personal\_training сохраняется явное приведение к пользовательским ENUM-типам при обновлении соответствующих полей.

Операции удаления данных организованы через DELETE-запросы для всех основных таблиц системы: locker\_assignment, locker, membership, personal\_training и person (листинг кода 3.3). Все запросы используют параметризованный идентификатор для точного указания удаляемой записи, что предотвращает случайное удаление всех данных из таблицы.

Все операции выполняются в рамках транзакционного управления Spring Framework, что обеспечивает атомарность и целостность данных.

#### Листинг 3.1 – Вставка данных с использованием INSERT-запросов

```
INSERT INTO locker_assignment (locker_id, person_id, booked_from) VALUES (?, ?, ?)

INSERT INTO locker (number, gender, status) VALUES (?, ?::gender_enum, ?::locker_status_enum)

INSERT INTO membership (member_id, name, description, price, start_date, end_date)
VALUES (?, ?, ?, ?, ?, ?)

INSERT INTO personal_training
(coach_id, member_id, date, start_time, end_time, status, price, notes)
VALUES (?, ?, ?, ?, ?, CAST(? AS training_status_enum), ?, ?)

INSERT INTO role (name) VALUES (?) RETURNING id
```

```

INSERT INTO user_log (person_id, action_type, action_description, action_time,
ip_address, severity)
VALUES (?, ?, ?, ?, ?, ?, ?::log_severity_enum)

INSERT INTO person (role_id, first_name, last_name, phone, email, password,
birth_date)
VALUES (?, ?, ?, ?, ?, ?, ?)

```

### Листинг 3.2 – Обновление данных с использованием UPDATE-запросов

```

UPDATE membership
SET member_id=?, name=?, description=?, price=?, start_date=?,
end_date=?
WHERE id=?

UPDATE locker SET number=?, gender=?::gender_enum, status=?::locker_status_enum
WHERE id=?

UPDATE personal_training
SET coach_id=?, member_id=?, date=?, start_time=?, end_time=?,
status=CAST(? AS training_status_enum), price=?, notes=?
WHERE id=?

```

### Листинг 3.3 – Удаление данных с использованием DELETE-запросов

```

DELETE FROM locker_assignment WHERE id = ?

DELETE FROM locker WHERE id = ?

DELETE FROM membership WHERE id = ?

DELETE FROM personal_training WHERE id = ?

DELETE FROM person WHERE id = ?

```

## **ВЫВОД**

В ходе выполнения индивидуального задания была успешно разработана и реализована система управления данными для спортивного зала с использованием Java Spring Boot и PostgreSQL. В рамках работы были созданы сервисы, обеспечивающие полный цикл операций манипуляции данными: добавление, изменение и удаление записей во всех основных сущностях системы.

Ключевым достижением стала реализация безопасного и эффективного взаимодействия с базой данных через JdbcTemplate с использованием параметризованных запросов, что обеспечивает надежное выполнение операций.

Разработанные операции охватывают все бизнес-процессы системы: управление пользователями и их ролями, бронирование шкафчиков, оформление абонементов, планирование персональных тренировок и ведение системного журнала событий.

Реализованная система предоставляет безопасный инструмент для управления данными спортивного зала. Архитектура решения позволяет легко расширять функциональность и адаптировать систему под изменяющиеся бизнес-требования.