

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 №1-2
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

1 Выбрать предметную область, определить и обосновать перечень информационных сущностей. Определить перечень функций выполняемых функций. Среди обязательных числятся:

- а) Авторизация пользователя;
- б) Управление пользователями (создание, удаление, наличие как минимум двухгрупп администраторы и пользователи);
- в) Журналирование действий пользователей.

2 Разработать ER-структуру данных с описанием всех сущностей и связей. Результат работы: ER-схема, описание всех информационных сущностей.

2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ

Концептуальная модель базы данных представляет собой абстрактное описание предметной области, которое включает в себя основные сущности, их атрибуты и связи между ними. Основными абстракциями при построении такой модели являются сущности (объекты предметной области), атрибуты (свойства сущностей) и связи (отношения между сущностями). Ключевым этапом является анализ текстового описания предметной области для выделения этих элементов. На основе анализа формируются возможные запросы к базе данных, которые определяют требования к системе. Концептуальная модель часто визуализируется в виде ER-диаграммы, которая наглядно отображает сущности, атрибуты и связи. Далее модель представляется в терминах реляционной модели в виде таблиц, где каждый атрибут имеет свой домен – допустимое множество значений с указанием типа данных и характеристик. Для каждой таблицы определяются ключи (первичные и внешние), которые обеспечивают целостность данных и связи между таблицами. Важным этапом является анализ функциональных зависимостей между атрибутами, который позволяет провести нормализацию отношений.

3 ВЫПОЛНЕНИЕ РАБОТЫ

3.1 Обоснование выбора предметной и определение перечня сущностей

В качестве предметной области выбрана система управления спортивного зала, так как она предполагает сложную структуру данных, включающую пользователей с различными ролями, абонементы, персональные тренировки, шкафчики и систему логирования действий, что идеально соответствует заданию. Основные информационные сущности включают:

Person: Необходима для хранения основных данных о людях (имя, фамилия, email, пароль, телефон, дата рождения), так как система требует единого хранилища информации о всех участниках спортивного зала.

- id: Уникальный идентификатор человека (SERIAL);
- role_id: Идентификатор роли (INTEGER);
- first_name: Имя (VARCHAR(100));
- last_name: Фамилия (VARCHAR(100));
- phone: Номер телефона (VARCHAR(30));
- email: Электронная почта (VARCHAR(255));
- password: Пароль (VARCHAR(255));
- birth_date: Дата рождения (DATE).

Role: Определяет роли пользователей в системе (администратор, клиент, тренер), что обеспечивает разграничение прав доступа.

- id: Уникальный идентификатор роли (SERIAL);
- name: Название роли (role_name_enum).

MemberDetails: Содержит дополнительную информацию о клиентах спортивного зала (дата присоединения), так как клиенты имеют специфические атрибуты.

- person_id: Идентификатор человека (INTEGER);
- join_date: Дата присоединения к залу (DATE).

CoachDetails: Хранит профессиональную информацию о тренерах (опыт, образование, описание), что важно для выбора тренера клиентами.

- person_id: Идентификатор человека (INTEGER);
- description: Описание тренера (VARCHAR(255));
- experience_years: Опыт работы в годах (INTEGER);
- education: Образование (VARCHAR(100)).

AdminDetails: Содержит данные об администраторах (должность, права доступа), необходимые для управления залом.

- person_id: Идентификатор человека (INTEGER);
- position: Должность (VARCHAR(100));
- permissions: Права доступа (TEXT).

Membership: Представляет абонементы клиентов (название, описание, цена, период действия), так как это основная бизнес-модель спортивного зала.

- id: Уникальный идентификатор абонемента (SERIAL);

- member_id: Идентификатор клиента (INTEGER);
- name: Название абонемента (VARCHAR(50));
- description: Описание абонемента (VARCHAR(255));
- price: Цена (DECIMAL(10,2));
- start_date: Дата начала действия (DATE);
- end_date: Дата окончания действия (DATE).

PersonalTraining: Фиксирует данные о персональных тренировках (дата, время, статус, цена), что является ключевой услугой клуба.

- id: Уникальный идентификатор тренировки (SERIAL);
- coach_id: Идентификатор тренера (INTEGER);
- member_id: Идентификатор клиента (INTEGER);
- date: Дата тренировки (DATE);
- start_time: Время начала (TIME);
- end_time: Время окончания (TIME);
- status: Статус тренировки (training_status_enum);
- price: Цена тренировки (DECIMAL(10,2));
- notes: Примечания (VARCHAR(255)).

Locker: Описывает шкафчики в раздевалках (номер, гендерная принадлежность, статус), необходимые для организации хранения вещей.

- id: Уникальный идентификатор шкафчика (SERIAL);
- number: Номер шкафчика (INTEGER);
- gender: Гендерная принадлежность (gender_enum);
- status: Статус шкафчика (locker_status_enum).

LockerAssignment: Хранит информацию о бронировании шкафчиков, обеспечивая управление доступом к ним.

- id: Уникальный идентификатор бронирования (SERIAL);
- locker_id: Идентификатор шкафчика (INTEGER);
- person_id: Идентификатор человека (INTEGER);
- booked_from: Время бронирования (TIMESTAMP).

UserLog: Фиксирует действия пользователей в системе (тип действия, описание, время, IP-адрес), что важно для безопасности.

- id: Уникальный идентификатор записи лога (SERIAL);
- person_id: Идентификатор человека (INTEGER);
- action_type: Тип действия (VARCHAR(100));
- action_description: Описание действия (VARCHAR(255));
- action_time: Время действия (TIMESTAMP);
- ip_address: IP-адрес (VARCHAR(45));
- severity: Уровень важности (log_severity_enum).

Связи:

PersonRole: Каждый человек (Person) имеет одну роль (Role) через role_id. Одна роль может быть у многих людей.

MemberDetailsPerson: Детали клиента (MemberDetails) относятся к одному человеку (Person) через person_id. Один человек может быть клиентом.

CoachDetailsPerson: Детали тренера (CoachDetails) относятся к одному человеку (Person) через person_id. Один человек может быть тренером.

AdminDetailsPerson: Детали администратора (AdminDetails) относятся к одному человеку (Person) через person_id. Один человек может быть администратором.

MembershipMember: Абонементы (Membership) относятся к одному клиенту (MemberDetails) через member_id. Один клиент может иметь несколько абонементов.

PersonalTrainingCoach: Тренировки (PersonalTraining) относятся к одному тренеру (CoachDetails) через coach_id. Один тренер может проводить много тренировок.

PersonalTrainingMember: Тренировки (PersonalTraining) относятся к одному клиенту (MemberDetails) через member_id. Один клиент может иметь много тренировок.

LockerAssignmentLocker: Бронирования шкафчиков (LockerAssignment) относятся к одному шкафчику (Locker) через locker_id. Один шкафчик может иметь много бронирований в разное время.

LockerAssignmentPerson: Бронирования шкафчиков (LockerAssignment) относятся к одному человеку (Person) через person_id. Один человек может бронировать много шкафчиков.

UserLogPerson: Записи логов (UserLog) относятся к одному человеку (Person) через person_id. Один человек может иметь много записей в логах.

3.2 Перечень выполняемых функций

Управление пользователями:

- Создание новых пользователей с указанием роли (администратор, клиент, тренер);
- Авторизация пользователей по email и паролю;
- Удаление пользователей из системы;
- Поиск пользователей по email для проверки уникальности;
- Получение информации о пользователе по идентификатору;
- Обновление роли пользователя (например, назначение администратором);
- Поиск пользователей по email.

Журналирование действий пользователей:

- Регистрация событий авторизации, регистрации, удаления пользователей;
- Запись информации о времени действия, IP-адресе пользователя;
- Классификация событий по уровням важности (информация, предупреждение, ошибка).

Управление абонементами:

- Создание и оформление абонементов для клиентов;
- Просмотр списка всех абонементов;
- Редактирование и обновление информации об абонементах;
- Удаление абонементов из системы.

Управление персональными тренировками:

- Создание расписания персональных тренировок
- Назначение тренеров и клиентов на тренировки;
- Просмотр тренировок;
- Редактирование тренировок;
- Удаление тренировок.

Управление шкафчиками:

- Добавление шкафчиков в раздевалки;
- Управление статусами шкафчиков (доступен, занят, неисправен), то есть редактирование данных о шкафчике;
- Регистрация бронирований шкафчиков пользователями;
- Просмотр истории бронирований;
- Удаление шкафчиков из базы.

Все функции системы реализованы с учетом требований безопасности, целостности данных и журналирования действий, что обеспечивает надежную работу спортивного зала и возможность контроля всех бизнес-процессов.

3.3 ER-диаграмма базы данных

Построим концептуальную модель ER-диаграммы с помощью PlantUML, полученный результат можно увидеть на рисунке 1.

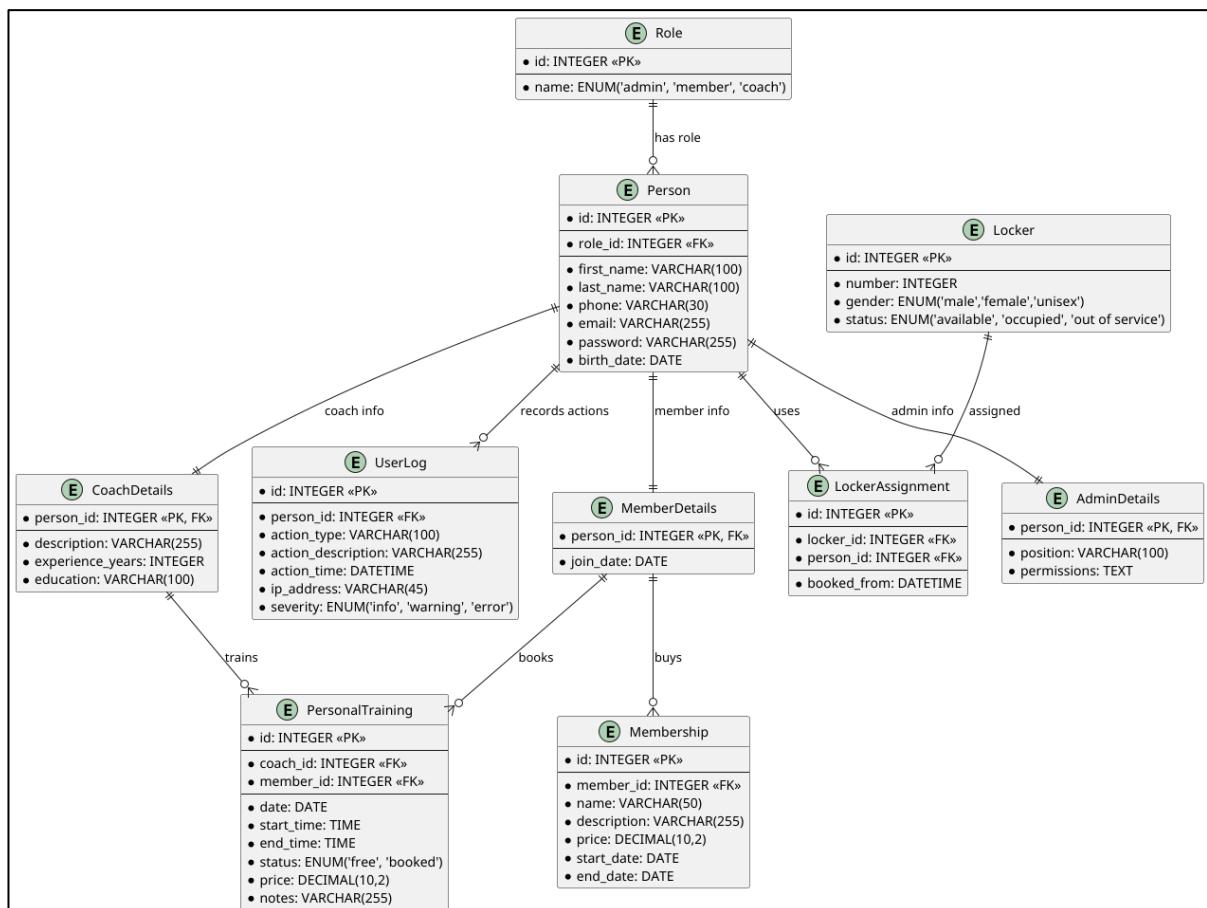


Рисунок 1 – Концептуальная модель в виде ER-диаграммы

ВЫВОД

В ходе выполнения лабораторной работы была успешно разработана и реализована система управления спортивного зала, соответствующая всем поставленным требованиям. Была выбрана и обоснована предметная область - система управления спортивного зала, которая обладает достаточной сложностью для демонстрации работы с реляционной базой данных и включает все необходимые функциональные требования. Разработана полноценная ER-структура данных, включающая 10 взаимосвязанных сущностей с четко определенными атрибутами, типами данных и связями. Структура обеспечивает хранение всей необходимой информации для функционирования спортивного зала.

Реализованы все обязательные функции системы: авторизация пользователей с проверкой учетных данных и определением ролей, управление пользователями с поддержкой трех групп (администраторы, клиенты и тренеры), журналирование действий с фиксацией всех значимых событий в системе. Дополнительно реализованы бизнес-функции: управление абонементами клиентов, организация персональных тренировок, бронирование шкафчиков и ведение логов действий пользователей.

Спроектированная база данных использует преимущества PostgreSQL, включая ENUM-типы для стандартизации допустимых значений, внешние ключи для обеспечения ссылочной целостности и индексы для оптимизации производительности. Система демонстрирует правильное применение принципов нормализации, что обеспечивает эффективное хранение данных и минимизацию избыточности.

Разработанное решение полностью удовлетворяет исходным требованиям лабораторной работы и может быть использовано как основа для реальной системы управления спортивным залом. Все компоненты системы интегрированы между собой и обеспечивают надежную работу в соответствии с бизнес-процессами спортивного учреждения.