

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 №7
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

Разработка приложения, наполнение тестовыми данными, определение необходимости использования индексов. Результат работы: приложение, отчет об обоснованности использования индексов.

2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ

Разработка приложения включает несколько ключевых этапов, начиная от проектирования структуры базы данных и заканчивая оптимизацией производительности. Важным аспектом является наполнение базы тестовыми данными, которые позволяют проверить функциональность приложения в условиях, приближенных к реальной эксплуатации.

Определение необходимости использования индексов является этапом оптимизации производительности базы данных. Индексы представляют собой специальные структуры данных, которые ускоряют выполнение запросов за счет уменьшения количества данных, подлежащих сканированию. В PostgreSQL поддерживаются различные типы индексов, включая B-деревья, хеш-индексы, GiST, GIN и BRIN, каждый из которых оптимален для определенных сценариев использования.

Обоснование создания индексов должно базироваться на анализе выполняемых запросов и характеристик данных. Ключевыми критериями для принятия решения о создании индекса являются: частота выполнения запросов, селективность столбцов, объем данных в таблице и соотношение операций чтения и записи. Индексы наиболее эффективны для столбцов, часто используемых в условиях WHERE, JOIN и ORDER BY.

Следует учитывать, что хотя индексы ускоряют операции чтения, они замедляют операции вставки, обновления и удаления данных, поскольку требуют поддержания индексных структур в актуальном состоянии. Поэтому важно находить баланс между производительностью запросов и нагрузкой на систему при модификации данных.

3 ВЫПОЛНЕНИЕ РАБОТЫ

В ходе разработки приложения для управления спортивным залом была реализована комплексная работа по наполнению базы данных тестовыми данными и оптимизации производительности через создание индексов. База данных PostgreSQL была наполнена тестовыми данными, включающими пользователей с различными ролями, администратор, тренер, клиент, абонементы, шкафчики, персональные тренировки и записи системного журнала (листинг кода 3.1).

Было создано шесть специализированных индексов для оптимизации наиболее частых запросов (листинг кода 3.2). Индекс `idx_person_role` ускоряет операции, связанные с поиском пользователей по их роли. Индекс `idx_membership_member` оптимизирует поиск абонементов по идентификатору клиента. Индексы `idx_training_member` и `idx_training_coach` улучшают производительность запросов к расписанию тренировок по клиентам и тренерам соответственно. Индекс `idx_log_person` ускоряет выборку записей журнала по пользователю, а `idx_locker_assignment_person` оптимизирует поиск бронирований шкафчиков.

Тестовые данные включают пять пользователей с хешированными паролями, использующими алгоритм BCrypt. Для каждого типа пользователей созданы соответствующие записи в таблицах деталей `member_details`, `coach_details` и `admin_details`. Добавлены три типа абонементов с различными характеристиками и стоимостью. В систему внесены шесть шкафчиков с разным гендерным назначением и статусами. Созданы три персональные тренировки с различными статусами и временными слотами. Заполнены записи системного журнала, фиксирующие ключевые события регистрацию пользователей, вход в систему, создание тренировок и приобретение абонементов.

Обоснование использования индексов базируется на анализе структуры данных и предполагаемых сценариев использования приложения. Индексы созданы для внешних ключей, которые часто участвуют в операциях JOIN и WHERE, что значительно ускоряет выполнение запросов, связанных с поиском и фильтрацией данных. Для таблиц с большим объемом данных, таких как `user_log`, индексы позволяют избежать полного сканирования таблицы при поиске записей конкретного пользователя. Наполнение базы тестовыми данными позволило проверить эффективность созданных индексов и убедиться в оптимальной производительности запросов.

Листинг 3.1 – Наполнение базы данных тестовыми данными

```
-- =====
-- Добавление пользователей
-- =====

-- Пароль: 123 (BCrypt хеш)
INSERT INTO person (role_id, first_name, last_name, email, password,
birth_date, phone)
```

```

VALUES
    (2, 'Иван', 'Иванов', 'user@gym.com',
    '$2a$10$PVfhZ0rF/Dz6dXl/ZNZPdesUgBb0RAaa4V0gEtNof98B2HqjKG0tu', '2000-12-12',
    '+375251234567'),
    (1, 'Алексей', 'Админов', 'admin@gym.com',
    '$2a$10$PVfhZ0rF/Dz6dXl/ZNZPdesUgBb0RAaa4V0gEtNof98B2HqjKG0tu', '1990-05-15',
    '+375251234568'),
    (3, 'Петр', 'Тренеров', 'coach@gym.com',
    '$2a$10$PVfhZ0rF/Dz6dXl/ZNZPdesUgBb0RAaa4V0gEtNof98B2HqjKG0tu', '1985-08-20',
    '+375251234569'),
    (2, 'Мария', 'Сидорова', 'maria@gym.com',
    '$2a$10$PVfhZ0rF/Dz6dXl/ZNZPdesUgBb0RAaa4V0gEtNof98B2HqjKG0tu', '1995-03-10',
    '+375291234567'),
    (2, 'Сергей', 'Петров', 'sergey@gym.com',
    '$2a$10$PVfhZ0rF/Dz6dXl/ZNZPdesUgBb0RAaa4V0gEtNof98B2HqjKG0tu', '1998-07-25',
    '+375331234567');

-- =====
-- Добавление деталей пользователей
-- =====

INSERT INTO member_details (person_id, join_date)
VALUES
    (1, '2024-01-15'),
    (4, '2024-02-01'),
    (5, '2024-02-10');

INSERT INTO coach_details (person_id, description, experience_years, education)
VALUES
    (3, 'Профессиональный тренер по фитнесу, специализация: силовые
    тренировки', 8, 'Высшее спортивное образование');

INSERT INTO admin_details (person_id, position, permissions)
VALUES
    (2, 'Главный администратор', 'full_access');

-- =====
-- Добавление абонементов
-- =====

INSERT INTO membership (member_id, name, description, price, start_date,
end_date)
VALUES
    (1, 'Стандарт', 'Базовый абонемент на 1 месяц', 3000.00, '2024-01-15',
    '2024-02-15'),
    (4, 'Премиум', 'Премиум абонемент с персональными тренировками', 8000.00,
    '2024-02-01', '2024-05-01'),
    (5, 'Базовый', 'Базовый абонемент на 2 недели', 1500.00, '2024-02-10',
    '2024-02-24');

-- =====
-- Добавление шкафчиков
-- =====

INSERT INTO locker (number, gender, status)
VALUES
    (1, 'male', 'available'),
    (2, 'male', 'occupied'),
    (3, 'female', 'available'),
    (4, 'female', 'out of service'),
    (5, 'unisex', 'available'),
    (6, 'unisex', 'available');

-- =====

```

```

-- Добавление бронирований шкафчиков
-- =====

INSERT INTO locker_assignment (locker_id, person_id, booked_from)
VALUES
    (2, 1, '2024-01-15 10:00:00'),
    (5, 4, '2024-02-01 09:30:00');

-- =====
-- Добавление персональных тренировок
-- =====

INSERT INTO personal_training (coach_id, member_id, date, start_time, end_time,
status, price, notes)
VALUES
    (3, 1, '2024-01-20', '10:00', '11:00', 'booked', 1500.00, 'Первая
тренировка, ознакомительная'),
    (3, 4, '2024-02-05', '18:00', '19:00', 'booked', 2000.00, 'Силовая
тренировка'),
    (3, 1, '2024-01-25', '11:00', '12:00', 'free', 1500.00, 'Свободный слот');

-- =====
-- Добавление логов пользователей
-- =====

INSERT INTO user_log (person_id, action_type, action_description, ip_address,
severity)
VALUES
    (1, 'REGISTER', 'Пользователь зарегистрирован в системе', '192.168.1.100',
'info'),
    (1, 'LOGIN', 'Успешный вход в систему', '192.168.1.100', 'info'),
    (2, 'USER_CREATED', 'Создан новый пользователь', '192.168.1.1', 'info'),
    (3, 'TRAINING_CREATED', 'Создана новая тренировка', '192.168.1.50',
'info'),
    (4, 'MEMBERSHIP_PURCHASED', 'Приобретен абонемент Премиум', '192.168.1.75',
'info');

```

Листинг 3.2 – Создание индексов

```

CREATE INDEX idx_person_role ON person(role_id);

CREATE INDEX idx_membership_member ON membership(member_id);

CREATE INDEX idx_training_member ON personal_training(member_id);

CREATE INDEX idx_training_coach ON personal_training(coach_id);

CREATE INDEX idx_log_person ON user_log(person_id);

CREATE INDEX idx_locker_assignment_person ON locker_assignment(person_id);

```

ВЫВОД

В рамках работы проведено комплексное наполнение базы данных тестовыми данными и выполнена оптимизация производительности через создание специализированных индексов. Было добавлено наполнение базы данных, включающее пользователей всех основных ролей администратора, тренеров и клиентов с корректно хешированными паролями. Сформированы связанные данные по абонеентам, шкафчикам, персональным тренировкам и системному журналу, что позволило создать полноценную тестовую среду, отражающую реальные бизнес-процессы спортивного зала.

Также было создано шесть индексов для оптимизации производительности запросов. Индексы были созданы для столбцов, участвующих в условиях соединения таблиц и фильтрации данных, что значительно ускоряет выполнение часто используемых запросов. Особое внимание уделено индексам для внешних ключей, которые активно используются в операциях JOIN между таблицами.