

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)» (МАИ)**

Институт № 8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

КУРСОВОЙ ПРОЕКТ

по дисциплине «Базы данных»

Тема: «Информационная система для трекинга привычек»

Студент:	Чесноков В.Д.
Группа:	М8О-309Б-23
Преподаватель:	Грубенко М.Д.
Оценка:	_____
Дата:	_____
Подпись:	_____

Москва 2025

Содержание

ВВЕДЕНИЕ	3
1 АНАЛИТИЧЕСКАЯ ЧАСТЬ	4
1.1 Обзор предметной области	4
1.2 Постановка задачи	4
2 ПРОЕКТНАЯ ЧАСТЬ	5
2.1 Архитектура системы	5
2.2 Проектирование структуры базы данных	5
2.3 Описание таблиц базы данных	5
2.3.1 Таблица users	6
2.3.2 Таблица user_profiles	7
2.3.3 Таблица habits	7
2.3.4 Таблица habit_schedules	8
2.3.5 Таблица habit_checkins	8
2.3.6 Таблица tags	9
2.3.7 Таблица habit_tags	9
2.3.8 Таблица reminders	9
2.3.9 Таблица habit_stats	10
2.3.10 Таблица audit_log	10
2.3.11 Таблица _manual_migrations	11
2.3.12 Таблица batch_import_jobs	11
2.3.13 Таблица batch_import_errors	12
2.4 Ограничения целостности данных	12
2.5 Реализация SQL-функциональности	13
2.6 Описание API и взаимодействия с БД	13
2.7 Примеры SQL-запросов и анализ производительности	14
3 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ	15
3.1 Контейнеризация и развертывание	15
3.2 Тестирование системы	15
ЗАКЛЮЧЕНИЕ	16

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17

ПРИЛОЖЕНИЕ А Логическая схема базы данных 18

ВВЕДЕНИЕ

Сервисы трекинга привычек требуют надежного хранения данных, аналитики и контроля изменений. Поэтому важно спроектировать базу данных, которая поддерживает целостность, аудит и быструю выборку. Требования к структуре пояснительной записки определены ГОСТ 7.32-2017 [?] и техническим заданием [?].

Объект исследования — информационная система для трекинга привычек пользователей. Предмет исследования — проектирование и реализация реляционной базы данных и API, обеспечивающих хранение, обработку и анализ данных о привычках.

Цель работы — разработать информационную систему с реляционной БД и серверным API, поддерживающими учет привычек, отметок выполнения и аналитические отчеты.

Для достижения цели нужно было:

- 1) провести анализ предметной области и требований к системе;
- 2) спроектировать структуру БД с необходимыми связями и типами данных;
- 3) реализовать ограничения целостности (PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, NOT NULL);
- 4) разработать SQL-объекты: триггеры, функции и представления;
- 5) реализовать REST API и механизм batch import;
- 6) оптимизировать запросы индексами и выполнить анализ производительности;
- 7) описать контейнеризацию, развертывание и тестирование.

Практическая ценность работы — готовая БД и API для сервиса трекинга привычек. Отчет включает введение, аналитическую, проектную и технологическую части, заключение и приложения.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Обзор предметной области

Сервис трекинга привычек предназначен для ведения списка привычек, контроля регулярности выполнения и получения аналитических отчетов. Для корректной работы системы требуется хранить учетные записи пользователей, перечни привычек, расписания, отметки выполнения, теги и напоминания, а также агрегированную статистику. Для контроля корректности и отслеживания изменений необходим журнал аудита.

Ключевые сущности предметной области:

- пользователи и их профили;
- привычки и расписания выполнения;
- отметки выполнения и связанные параметры (настроение, длительность);
- теги и напоминания;
- агрегированная статистика и аудит изменений.

Для предметной области характерны частые транзакционные операции (ежедневные отметки) и аналитические выборки. Это обуславливает необходимость индексирования и оптимизации запросов.

1.2 Постановка задачи

В соответствии с техническим заданием [?] система должна содержать не менее 9–10 таблиц со связями 1:1, 1:N и N:M, при этом каждая таблица включает не менее пяти атрибутов различных типов. Требуется реализовать ограничения целостности, каскадные правила, триггеры аудита и агрегаторов, SQL-функции (скалярные и табличные), а также не менее трех представлений. Для ключевых запросов необходимо создать индексы и показать ускорение с помощью EXPLAIN ANALYZE. API должно предоставлять CRUD-операции, отчеты, batch import и быть документировано через Swagger.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Архитектура системы

Система реализована как монорепозиторий с разделением на backend, frontend и пакет базы данных. Backend построен на NestJS [2] и взаимодействует с PostgreSQL 16 [1] через ORM Prisma [3]. Frontend реализован на Next.js [4]. Контейнеризация выполнена с помощью Docker и Docker Compose [5]. Архитектура взаимодействия компонентов представлена на рисунке 1.



Рис. 1: Архитектура информационной системы

2.2 Проектирование структуры базы данных

Схема БД включает 13 таблиц и удовлетворяет требованиям по числу сущностей, разнообразию типов данных и наличию связей разных типов. Используются типы данных UUID, TEXT, VARCHAR, CHAR, INTEGER, BIGINT, SMALLINT, NUMERIC, BOOLEAN, DATE, TIME, TIMESTAMPTZ, JSONB и INET. Основные связи:

- users — user_profiles (1:1);
- users — habits (1:N);
- habits — habit_checkins (1:N);
- habits — tags (N:M через habit_tags);
- habits — habit_stats (1:1);
- batch_import_jobs — batch_import_errors (1:N).

Логическая схема базы данных с основными сущностями и атрибутами представлена в приложении А (рисунок А.1).

2.3 Описание таблиц базы данных

Ключевые таблицы и их назначение приведены в таблице 1. Полный перечень содержит 13 таблиц: users, user_profiles, habits,

habit_schedules, habit_checkins, tags, habit_tags, reminders, habit_stats, audit_log, _manual_migrations, batch_import_jobs, batch_import_errors.

Таблица 1: Основные таблицы базы данных

Таблица	Назначение	Ключевые поля
users	учетные записи пользователей	id, email
user_profiles	профиль пользователя	user_id, timezone
habits	перечень привычек	user_id, name, type
habit_schedules	расписания привычек	habit_id, frequency_type
habit_checkins	отметки выполнения	habit_id, checkin_date
tags	справочник тегов	name, slug
habit_tags	связь привычек и тегов	habit_id, tag_id
reminders	напоминания	habit_id, reminder_time
habit_stats	агрегированная статистика	habit_id, total_checkins
audit_log	журнал аудита	table_name, operation
batch_import_jobs	задания импорта	entity_type, status
batch_import_errors	ошибки импорта	job_id, error_message
_manual_migrations	учет SQL-миграций	name, status

2.3.1 Таблица users

Назначение: основная таблица пользователей системы.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY, DEFAULT gen_random_uuid()	Уникальный идентификатор
email	TEXT	UNIQUE, NOT NULL	Email пользователя
password_hash	TEXT	NOT NULL	Хеш пароля
full_name	TEXT	NOT NULL	Полное имя

Колонка	Тип	Ограничения	Описание
role	VARCHAR(50)	DEFAULT 'user'	Роль пользователя
is_active	BOOLEAN	DEFAULT true	Активен ли аккаунт
created_at	TIMESTAMPTZ	DEFAULT NOW()	Дата создания
login_count	INTEGER	DEFAULT 0	Количество входов

Уникальные типы: UUID, TEXT, VARCHAR, BOOLEAN, TIMESTAMPTZ, INTEGER.

2.3.2 Таблица user_profiles

Назначение: дополнительная информация о пользователе (связь 1:1 с users).

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY, DEFAULT gen_random_uuid()	Уникальный идентификатор
user_id	UUID	UNIQUE, NOT NULL, FK -> users	Ссылка на пользователя
bio	TEXT	NULL	Биография
avatar_url	TEXT	NULL	URL аватара
timezone	VARCHAR(100)	DEFAULT 'UTC'	Часовой пояс
date_of_birth	DATE	NULL	Дата рождения
notification_enabled	BOOLEAN	DEFAULT true	Уведомления включены
theme_preference	SMALLINT	DEFAULT 0, CHECK (0–2)	Тема оформления

Уникальные типы: UUID, TEXT, VARCHAR, DATE, BOOLEAN, SMALLINT.

2.3.3 Таблица habits

Назначение: привычки пользователей (хорошие и плохие).

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY, DEFAULT gen_random_uuid()	Уникальный идентификатор
user_id	UUID	NOT NULL, FK -> users	Владелец привычки
name	TEXT	NOT NULL	Название привычки
description	TEXT	NULL	Описание
type	VARCHAR(10)	NOT NULL, CHECK IN ('good', 'bad')	Тип привычки

Колонка	Тип	Ограничения	Описание
priority	SMALLINT	DEFAULT 0, CHECK (0–10)	Приоритет
is_archived	BOOLEAN	DEFAULT false	Архивирована
display_order	INTEGER	DEFAULT 0	Порядок отображения
created_at	TIMESTAMPTZ	DEFAULT NOW()	Дата создания

Уникальные типы: UUID, TEXT, VARCHAR, SMALLINT, BOOLEAN, INTEGER, TIMESTAMPTZ.

2.3.4 Таблица habit_schedules

Назначение: расписания для привычек (ежедневные, еженедельные и т.д.).

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
habit_id	UUID	NOT NULL, FK -> habits	Ссылка на привычку
frequency_type	VARCHAR(20)	CHECK IN (...)	Тип частоты
frequency_value	INTEGER	DEFAULT 1, CHECK > 0	Значение частоты
weekdays_mask	SMALLINT	DEFAULT 127	Битовая маска дней недели
start_date	DATE	NOT NULL	Дата начала
end_date	DATE	NULL	Дата окончания
is_active	BOOLEAN	DEFAULT true	Активно

Уникальные типы: UUID, VARCHAR, INTEGER, SMALLINT, DATE, BOOLEAN.

2.3.5 Таблица habit_checkins

Назначение: отметки выполнения привычек (транзакционная таблица).

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
habit_id	UUID	NOT NULL, FK -> habits	Ссылка на привычку
user_id	UUID	NOT NULL, FK -> users	Ссылка на пользователя
checkin_date	DATE	NOT NULL	Дата отметки
checkin_time	TIME	DEFAULT CURRENT_TIME	Время отметки
notes	TEXT	NULL	Заметки

Колонка	Тип	Ограничения	Описание
mood_rating	SMALLINT	CHECK (1–5)	Оценка настроения
duration_minute	INTEGER	NULL	Длительность в минутах
created_at	TIMESTAMPTZ	DEFAULT NOW()	Дата создания

Уникальные типы: UUID, DATE, TIME, TEXT, SMALLINT, INTEGER, TIMESTAMPTZ. UNIQUE: (habit_id, checkin_date).

2.3.6 Таблица tags

Назначение: теги для категоризации привычек.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
name	VARCHAR(100)	UNIQUE, NOT NULL	Название тега
slug	TEXT	UNIQUE, NOT NULL	URL-friendly имя
color	VARCHAR(7)	DEFAULT '#gray'	Цвет
usage_count	INTEGER	DEFAULT 0	Счётчик использований
is_system	BOOLEAN	DEFAULT false	Системный тег
created_at	TIMESTAMPTZ	DEFAULT NOW()	Дата создания

Уникальные типы: UUID, VARCHAR, TEXT, INTEGER, BOOLEAN, TIMESTAMPTZ.

2.3.7 Таблица habit_tags

Назначение: связь N:M между привычками и тегами.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
habit_id	UUID	NOT NULL, FK -> habits	Ссылка на привычку
tag_id	UUID	NOT NULL, FK -> tags	Ссылка на тег
priority	SMALLINT	DEFAULT 0	Приоритет
is_primary	BOOLEAN	DEFAULT false	Основной тег
assigned_by	VARCHAR(50)	DEFAULT 'user'	Кто назначил
assigned_at	TIMESTAMPTZ	DEFAULT NOW()	Дата назначения

Уникальные типы: UUID, SMALLINT, BOOLEAN, VARCHAR, TIMESTAMPTZ. UNIQUE: (habit_id, tag_id).

2.3.8 Таблица reminders

Назначение: напоминания для привычек.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
habit_id	UUID	NOT NULL, FK -> habits	Ссылка на привычку
reminder_time	TIME	NOT NULL	Время напоминания
days_of_week	SMALLINT	DEFAULT 127	Дни недели (битовая маска)
notification_text	TEXT	NULL	Текст уведомления
delivery_method	VARCHAR(20)	DEFAULT 'push', CHECK IN (...)	Метод доставки
is_active	BOOLEAN	DEFAULT true	Активно
created_at	TIMESTAMPTZ	DEFAULT NOW()	Дата создания

Уникальные типы: UUID, TIME, SMALLINT, TEXT, VARCHAR, BOOLEAN, TIMESTAMPTZ.

2.3.9 Таблица habit_stats

Назначение: агрегированная статистика по привычкам (обновляется триггером).

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
habit_id	UUID	UNIQUE, NOT NULL, FK -> habits	Ссылка на привычку
total_checkins	INTEGER	DEFAULT 0	Всего отметок
current_streak	INTEGER	DEFAULT 0	Текущая серия
longest_streak	INTEGER	DEFAULT 0	Лучшая серия
completion_rate	NUMERIC(5,2)	DEFAULT 0.00	Процент выполнения
average_mood	NUMERIC(3,2)	NULL	Средняя оценка настроения
last_checkin_at	DATE	NULL	Последняя отметка
updated_at	TIMESTAMPTZ	DEFAULT NOW()	Дата обновления

Уникальные типы: UUID, INTEGER, NUMERIC, DATE, TIMESTAMPTZ.

2.3.10 Таблица audit_log

Назначение: журнал аудита изменений.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
table_name	VARCHAR(100)	NOT NULL	Название таблицы
operation	VARCHAR(10)	NOT NULL, CHECK IN (...)	Тип операции

Колонка	Тип	Ограничения	Описание
record_id	UUID	NULL	ID изменённой записи
user_id	UUID	NULL	ID пользователя
old_data	JSONB	NULL	Старые данные
new_data	JSONB	NULL	Новые данные
ip_address	INET	NULL	IP-адрес
changed_at	TIMESTAMPTZ	DEFAULT NOW()	Дата изменения

Уникальные типы: UUID, VARCHAR, JSONB, INET, TIMESTAMPTZ.

2.3.11 Таблица _manual_migrations

Назначение: контроль применения SQL-миграций (триггеров, функций, представлений и индексов).

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
name	VARCHAR(255)	UNIQUE, NOT NULL	Название миграции
checksum	CHAR(64)	NULL	SHA-256 hash
applied_at	TIMESTAMPTZ	DEFAULT NOW()	Дата применения
execution_time_ms	INTEGER	NULL	Время выполнения
status	VARCHAR(20)	DEFAULT 'success', CHECK IN (...)	Статус
applied_by	TEXT	NULL	Кто применил

Уникальные типы: UUID, VARCHAR, CHAR, TIMESTAMPTZ, INTEGER, TEXT.

2.3.12 Таблица batch_import_jobs

Назначение: задачи батчевого импорта данных.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
user_id	UUID	FK -> users, ON DELETE SET NULL	Пользователь
entity_type	VARCHAR(50)	NOT NULL	Тип сущности
status	VARCHAR(20)	CHECK IN (...)	Статус
total_records	INTEGER	DEFAULT 0	Всего записей
success_count	INTEGER	DEFAULT 0	Успешных
error_count	INTEGER	DEFAULT 0	Ошибка
progress_percent	NUMERIC(5,2)	DEFAULT 0.00	Прогресс
file_size_bytes	BIGINT	NULL	Размер файла

Колонка	Тип	Ограничения	Описание
started_at	TIMESTAMPTZ	DEFAULT NOW()	Время начала
completed_at	TIMESTAMPTZ	NULL	Время завершения

Уникальные типы: UUID, VARCHAR, INTEGER, NUMERIC, BIGINT, TIMESTAMPTZ.

2.3.13 Таблица batch_import_errors

Назначение: ошибки батчевого импорта.

Колонка	Тип	Ограничения	Описание
id	UUID	PRIMARY KEY	Уникальный идентификатор
job_id	UUID	NOT NULL, FK -> batch_import_jobs	Ссылка на задачу
row_number	INTEGER	NULL	Номер строки
record_data	JSONB	NOT NULL	Данные записи
error_message	TEXT	NOT NULL	Сообщение об ошибке
error_code	VARCHAR(50)	NULL	Код ошибки
created_at	TIMESTAMPTZ	DEFAULT NOW()	Дата создания

Уникальные типы: UUID, INTEGER, JSONB, TEXT, VARCHAR, TIMESTAMPTZ.

2.4 Ограничения целостности данных

Для обеспечения целостности реализованы следующие типы ограничений: PRIMARY KEY на поле id во всех таблицах, FOREIGN KEY с каскадным удалением или установкой NULL, UNIQUE для уникальных атрибутов, CHECK для доменных ограничений и NOT NULL для обязательных полей. Примеры:

- PRIMARY KEY: id UUID PRIMARY KEY DEFAULT gen_random_uuid();
- FOREIGN KEY: habits.user_id -> users.id ON DELETE CASCADE;
- UNIQUE: habit_checkins(habit_id, checkin_date);
- CHECK: habits.type IN ('good', 'bad'),
habit_checkins.mood_rating BETWEEN 1 AND 5;
- NOT NULL: users.email, habits.name, habits.type.

2.5 Реализация SQL-функциональности

Базовые CRUD-операции реализованы через Prisma ORM, а сложные выборки и отчеты — через параметризованные SQL-запросы. Реализованы триггеры аудита и автоматического обновления агрегированной статистики, скалярная функция calc_completion_rate и табличная функция report_user_habits. Основные SQL-скрипты:

- packages/db/sql/001_triggers.sql — триггеры аудита и агрегаторов;
- packages/db/sql/002_functions.sql — функции;
- packages/db/sql/003_views.sql — представления;
- packages/db/sql/004_indexes.sql — индексы.

2.6 Описание API и взаимодействия с БД

Backend предоставляет REST API с документацией в Swagger. Основные группы эндпойнтов:

- аутентификация: /auth/register, /auth/login, /auth/me;
- управление привычками: /habits (GET, POST), /habits/:id (GET, PUT, DELETE);
- отметки выполнения: /checkins (GET, POST), /checkins/:id (DELETE);
- отчеты: /reports/user/:userId, /reports/completion-rate/:userId;
- массовая загрузка: /batch-import и /batch-import/:jobId.

Безопасность обеспечивается JWT-аутентификацией, хешированием паролей (bcrypt) и параметризованными запросами к БД. SQL-инъекции исключаются благодаря использованию ORM и параметризации.

2.7 Примеры SQL-запросов и анализ производительности

Примеры типовых запросов приведены в листинге 1.

Листинг 1: Примеры SQL-запросов

```
1 SELECT * FROM habits
2 WHERE user_id = $1 AND NOT is_archived
3 ORDER BY display_order;
4
5 SELECT * FROM report_user_habits($1, $2, $3);
6
7 SELECT * FROM v_user_habit_summary
8 WHERE user_id = $1;
```

Для повышения производительности созданы индексы на полях, используемых в фильтрации и сортировке. В таблице 15 приведено сравнение времени выполнения запросов до и после индексирования. Полные планы EXPLAIN ANALYZE приведены в файле docs/perf.md. Основные результаты: 52x быстрее для истории отметок и 69x быстрее для журнала аудита, среднее ускорение около 60x.

Таблица 15: Сравнение производительности запросов

Запрос	До, мс	После, мс	Ускорение
История отметок пользователя за 30 дней	145.456	2.789	52x
Журнал аудита за 24 часа	89.456	1.289	69x

Ключевые индексы: idx_habits_user_id, idx_habit_checkins_user_date, idx_audit_log_user_time и др.

3 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1 Контейнеризация и развертывание

Развертывание системы выполняется с помощью Docker Compose. Используются три сервиса: база данных PostgreSQL, backend и frontend. Конфигурация размещена в файле `docker-compose.yml` и описывает порты, переменные окружения и зависимости сервисов.

После запуска контейнеров выполняются миграции и применение SQL-скриптов: `pnpm db:migrate`, `pnpm db:sql`, `pnpm db:seed`. Swagger доступен по адресу `http://localhost:3001/api/docs`.

3.2 Тестирование системы

Для проверки работоспособности используются тестовые данные и контрольные запросы. Скрипт заполнения БД формирует реалистичный набор данных, что позволяет проверить триггеры, функции и производительность.

Таблица 16: Объем тестовых данных после заполнения

Таблица	Количество записей
users	100
habits	800
habit_checkins	10000
tags	50
habit_tags	1500
reminders	600

Примеры проверочных запросов включают выборку последних записей аудита, расчет `calc_completion_rate` за период и получение ежедневной статистики из представления `v_daily_completion`.

ЗАКЛЮЧЕНИЕ

В проекте разработана информационная система для трекинга привычек. Реализована реляционная БД из 13 таблиц со связями 1:1, 1:N и N:M, ограничениями целостности, триггерами аудита и агрегатов, функциями и представлениями. Реализован REST API и batch import. Оптимизация запросов индексами дала ускорение в десятки раз.

Задачи выполнены. Система корректно работает и может развиваться дальше за счет расширения аналитики и функциональности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Список литературы

- [1] PostgreSQL 16 Documentation [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/16/>.
- [2] NestJS Documentation [Электронный ресурс]. — URL: <https://docs.nestjs.com/>.
- [3] Prisma Documentation [Электронный ресурс]. — URL: <https://www.prisma.io/docs>.
- [4] Next.js Documentation [Электронный ресурс]. — URL: <https://nextjs.org/docs>.
- [5] Docker Documentation [Электронный ресурс]. — URL: <https://docs.docker.com/>.

ПРИЛОЖЕНИЕ А Логическая схема базы данных

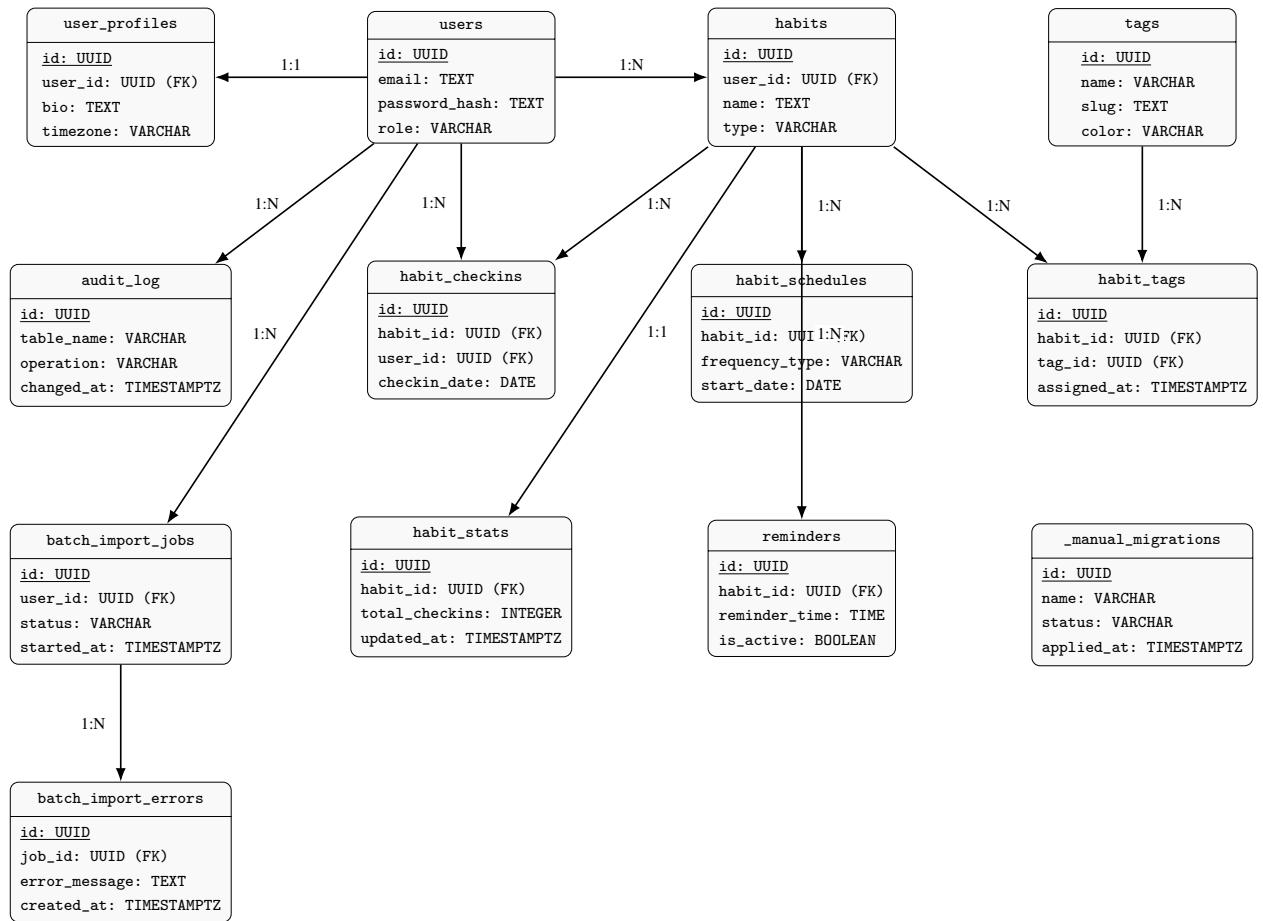


Рис. А.1: Логическая схема базы данных