

Разработка и анализ базы данных учёта и контроля выполнения задач в корпоративной ИТ-среде

Саухин Сергей Иванович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматриваются вопросы проектирования и реализации реляционной базы данных, предназначенной для учёта и контроля выполнения задач в корпоративной ИТ-среде. Проведён анализ предметной области управления задачами, выделены основные сущности и их взаимосвязи, а также реализованы механизмы обеспечения целостности данных на уровне базы данных. Особое внимание уделено практической реализации структуры базы данных с использованием языка SQL и демонстрации её применения на примерах. Результаты исследования могут быть использованы при разработке корпоративных информационных систем управления задачами и в учебном процессе.

Ключевые слова: базы данных, реляционная модель, SQL, корпоративные ИТ-системы, управление задачами.

Development and analysis of a database for task accounting and control in a corporate IT environment

Saukhin Sergey Ivanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper discusses the design and implementation of a relational database intended for task accounting and control in a corporate IT environment. The task management domain is analyzed, key entities and their relationships are identified, and data integrity mechanisms at the database level are implemented. Special attention is paid to the practical implementation of the database structure using SQL and examples of its application. The results can be used in corporate task management systems and educational projects.

Keywords: databases, relational model, SQL, corporate IT systems, task management.

1 Введение

1.1 Актуальность

Актуальность данной работы обусловлена возрастающей ролью информационных систем в корпоративной деятельности и необходимостью

повышения эффективности управления рабочими процессами. Согласно исследованиям в области проектирования информационных систем, корректная организация хранения и обработки данных является фундаментом надёжной и масштабируемой ИТ-инфраструктуры [1].

Работы, посвящённые разработке автоматизированных информационных систем, показывают, что системы учёта и контроля задач позволяют снизить нагрузку на управленческий персонал, повысить дисциплину исполнения и обеспечить прозрачность процессов [2]. Кроме того, исследования в области проектирования прикладных информационных систем подчёркивают важность использования реляционных баз данных для хранения структурированной информации и обеспечения целостности данных [3–5].

Таким образом, разработка базы данных учёта и контроля выполнения задач в корпоративной ИТ-среде является актуальной задачей, имеющей практическую значимость для организаций различного профиля.

1.2 Обзор исследований

В научных и учебных публикациях вопросы проектирования информационных систем рассматриваются с точки зрения анализа предметной области, моделирования данных и реализации баз данных. В работах по методикам проектирования информационных систем подчёркивается, что корректное выделение сущностей и связей между ними является ключевым этапом разработки ИТ-решений [1].

Ряд исследований посвящён разработке автоматизированных систем учёта в различных предметных областях, включая образовательные и корпоративные среды. В них отмечается, что использование централизованных баз данных позволяет обеспечить целостность информации и упростить процессы анализа и контроля [2, 3].

Также в научных публикациях рассматриваются вопросы проектирования баз данных для прикладных информационных систем, где подчёркивается роль ограничений целостности, справочников и структурированных данных для повышения надёжности систем [4, 5]. Анализ указанных работ подтверждает целесообразность использования реляционной модели данных при разработке систем учёта и контроля задач.

1.3 Цель исследования

Целью данной работы является разработка и анализ базы данных учёта и контроля выполнения задач в корпоративной ИТ-среде.

Для достижения поставленной цели были сформулированы следующие задачи:

- проанализировать предметную область управления задачами;
- определить основные сущности и их атрибуты;
- разработать логическую структуру базы данных;
- реализовать структуру базы данных с использованием языка SQL;

– продемонстрировать практическое применение разработанных решений.

2 Материалы и методы

В ходе проведения исследования использовался комплекс теоретических и практических методов, направленных на проектирование и анализ базы данных учёта и контроля выполнения задач в корпоративной ИТ-среде. Методологическая основа работы базируется на общепринятых подходах к разработке информационных систем и реляционных баз данных, изложенных в научных и учебных публикациях [1, 6–9].

В качестве теоретической базы исследования были использованы материалы, посвящённые методикам и моделям проектирования информационных систем, вопросам анализа предметной области и логического моделирования данных [1]. В данных работах подчёркивается важность этапов предварительного анализа и формализации требований, что было учтено при разработке структуры базы данных учёта задач.

Проектирование базы данных осуществлялось с применением методов логического моделирования, включающих выделение сущностей предметной области, определение их атрибутов и установление связей между ними. В рамках предметной области управления задачами были выделены сущности «Пользователь», «Задача» и «Статус задачи», что соответствует подходам, применяемым при разработке прикладных информационных систем [2–5].

Для обеспечения корректности структуры базы данных применялись принципы нормализации данных, направленные на снижение избыточности информации и предотвращение аномалий обновления. Реализация справочников и использование внешних ключей позволили обеспечить ссылочную целостность данных и логическую согласованность информации, что соответствует рекомендациям по проектированию реляционных баз данных [6–9].

Практическая часть исследования включала реализацию разработанной логической модели с использованием языка структурированных запросов SQL. В ходе работы были применены стандартные средства СУБД для создания таблиц, задания первичных и внешних ключей, а также реализации ограничений целостности. Для автоматизации отдельных процессов использовались механизмы триггеров, позволяющие минимизировать влияние человеческого фактора при работе с данными.

Для анализа функционирования базы данных применялись методы моделирования типовых сценариев эксплуатации системы, включая добавление пользователей, постановку задач, изменение их статусов и анализ сроков выполнения. Также использовались методы выборки и агрегирования данных, реализованные с помощью SQL-запросов и представлений, что позволило оценить возможности системы по контролю и анализу выполнения задач.

Использование совокупности указанных материалов и методов позволило обоснованно реализовать структуру базы данных, проверить её

работоспособность и сформировать практико-ориентированные выводы о целесообразности применения разработанных решений в корпоративной ИТ-среде.

3 Результаты и обсуждение

В рамках проведённого исследования была реализована реляционная база данных, предназначенная для учёта и контроля выполнения задач в информационной системе. Практическая реализация осуществлялась с использованием языка структурированных запросов SQL и включала создание таблиц, определение связей между сущностями, задание ограничений целостности и выполнение типовых операций работы с данными.

В данном разделе представлены результаты проектирования и реализации базы данных, а также приведены примеры SQL-кода, иллюстрирующие применение разработанных решений на практике. Каждый этап реализации сопровождается описанием его назначения и области применения, что позволяет наглядно продемонстрировать функционирование базы данных и оценить эффективность принятых проектных решений.

3.1 Реализация сущности «Пользователь»

Для корректной работы системы учёта задач необходимо однозначно идентифицировать пользователей, ответственных за выполнение задач. Хранение информации о пользователях в отдельной таблице позволяет централизовать данные и избежать их дублирования (рис. 1).

```
CREATE TABLE users (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    full_name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) UNIQUE NOT NULL,  
    role VARCHAR(100) NOT NULL  
);
```

Рис 1. Создание таблицы «users»

Использование первичного ключа id обеспечивает уникальность каждой записи, а ограничение UNIQUE для поля email предотвращает создание дублирующих учётных записей. Данное решение упрощает назначение задач и дальнейший анализ активности пользователей.

3.2 Формализация ролей пользователей

В информационных системах управления задачами пользователи могут выполнять различные функции. Для формализации ролей используется

отдельный справочник, позволяющий расширять систему без изменения основной структуры (рис. 2,3).

```
CREATE TABLE user_roles (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    role_name VARCHAR(100) NOT NULL  
);
```

Рис 2. Создание таблицы «user_roles»

```
INSERT INTO user_roles (role_name)  
VALUES ('исполнитель'), ('руководитель'), ('администратор');
```

Рис. 3. Добавление ролей в таблицу «user_roles»

Выделение ролей в отдельную таблицу позволяет в дальнейшем реализовать разграничение прав доступа и повысить гибкость системы.

3.3 Реализация справочника статусов задач

Для корректного отслеживания жизненного цикла задач необходимо использовать ограниченный набор допустимых состояний (рис. 4,5).

```
CREATE TABLE task_statuses (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    status_name VARCHAR(100) NOT NULL  
);
```

Рис. 4 Создание таблицы «task_statuses»

```
INSERT INTO task_statuses (status_name)  
VALUES ('Новая'), ('В работе'), ('Выполнена'), ('Отменена');
```

Рис. 5. Внесение в таблицу «task_statuses» статус работы

Использование справочника статусов исключает ошибки, связанные с произвольным вводом текстовых значений, и облегчает анализ состояния задач.

3.4 Реализация сущности «Задача»

Таблица задач является центральным элементом базы данных и предназначена для хранения всей информации, необходимой для контроля выполнения задач (рис. 6).

```
CREATE TABLE tasks (  
    id INT PRIMARY KEY AUTO_INCREMENT  
    title VARCHAR(255) NOT NULL,  
    description TEXT,  
    created_at DATE NOT NULL,  
    deadline DATE NOT NULL,  
    completed_at DATE,  
    user_id INT NOT NULL,  
    status_id INT NOT NULL  
);
```

Рис. 6. Создание таблицы «tasks»

Наличие отдельных полей для дат создания, завершения и срока выполнения позволяет анализировать временные характеристики выполнения задач.

3.5 Обеспечение ссылочной целостности данных

Связи между задачами, пользователями и статусами должны быть логически корректными (рис. 7).

```
ALTER TABLE tasks  
ADD CONSTRAINT fk_tasks_users  
FOREIGN KEY (user_id) REFERENCES users(id);  
  
ALTER TABLE tasks  
ADD CONSTRAINT fk_tasks_statuses  
FOREIGN KEY (status_id) REFERENCES task_statuses(id);
```

Рис. 7. Создание внешних ключей

Внешние ключи предотвращают появление задач без назначенного исполнителя или корректного статуса.

3.6 Контроль корректности сроков выполнения

Необходимо исключить ввод задач с некорректными временными параметрами (рис. 8).

```
ALTER TABLE tasks  
ADD CONSTRAINT chk_task_dates  
CHECK (deadline >= created_at);
```

Рис. 8. Добавление ограничения

Реализация данного ограничения на уровне базы данных обеспечивает корректность данных независимо от прикладного интерфейса.

3.7 Отсутствие контроля времени жизни сессии

Добавление пользователей моделирует реальный процесс эксплуатации системы (рис. 9).

```
INSERT INTO users (full_name, email, role)  
VALUES  
( 'Иванов И.И.', 'ivanov@mail.ru', 'исполнитель' ),  
( 'Петров П.П.', 'petrov@mail.ru', 'руководитель' );
```

Рис. 9. Добавление тестовых данных в таблицу «users»

Данный код используется при первичном наполнении базы данных и расширении состава пользователей,

3.8 Постановка задач исполнителям

Реализация процесса создания новой задачи (рис. 10).

```
INSERT INTO tasks (  
    title,  
    description,  
    created_at,  
    deadline,  
    user_id,  
    status_id  
)  
VALUES (  
    'Подготовка отчёта',  
    'Подготовить отчёт по практике',  
    CURRENT_DATE,  
    DATE_ADD(CURRENT_DATE, INTERVAL 10 DAY),  
    1,  
    1  
);
```

Рис. 13. Добавление в таблицу «tasks» тестовый запрос

Запрос демонстрирует корректное использование связей и справочников.

3.9 Управление жизненным циклом задачи

Изменение статуса отражает продвижение задачи по этапам выполнения (рис. 14).

```
UPDATE tasks  
SET status_id = 2  
WHERE id = 1;
```

Рис. 14. Добавление в таблицу «tasks» тестовый запрос

Такая операция используется для мониторинга хода выполнения задач.

3.10 Автоматизация фиксации завершения задач

Исключение ручного ввода даты завершения (рис. 15).

```
CREATE TRIGGER trg_task_complete
BEFORE UPDATE ON tasks
FOR EACH ROW
BEGIN
    IF NEW.status_id = 3 AND OLD.status_id <> 3 THEN
        SET NEW.completed_at = CURRENT_DATE;
    END IF;
END;
```

Рис. 15. Добавление триггера

Использование триггеров повышает надёжность и согласованность данных.

3.11 Контроль активных задач

Получение списка задач, находящихся в работе (рис. 15).

```
SELECT
    t.title,
    u.full_name,
    s.status_name,
    t.deadline
FROM tasks t
JOIN users u ON t.user_id = u.id
JOIN task_statuses s ON t.status_id = s.id
WHERE s.status_name IN ('Новая', 'В работе');
```

Рис. 15. Запрос вывода задач находящийся «В работе», «Новая»

Запрос используется для оперативного контроля состояния системы.

3.12 Выявление просроченных задач

Контроль соблюдения сроков выполнения (рис. 16).

```
SELECT
    title,
    deadline
FROM tasks
WHERE deadline < CURRENT_DATE
AND status_id <> 3;
```

Рис. 16. Запрос на вывод проблемных задач

Позволяет выявлять проблемные задачи и принимать управленческие решения.

3.12 Выявление просроченных задач

Оценка количества задач на пользователя (рис. 17).

```
CREATE VIEW user_task_load AS
SELECT
    u.full_name,
    COUNT(t.id) AS task_count
FROM users u
LEFT JOIN tasks t ON u.id = t.user_id
GROUP BY u.full_name;
```

Рис. 17. Запрос на вывод проблемных задач

Представление используется для анализа эффективности распределения задач.

4 Выводы

В результате проведённого исследования была разработана и проанализирована реляционная база данных, предназначенная для учёта и контроля выполнения задач в корпоративной ИТ-среде. В ходе работы выполнен анализ предметной области управления задачами, что позволило выявить основные информационные сущности, необходимые для реализации системы, а также определить их атрибуты и взаимосвязи.

На этапе проектирования была сформирована логическая структура базы данных, основанная на принципах реляционной модели и нормализации

данных. Выделение отдельных таблиц для пользователей, задач и справочников состояний обеспечило снижение избыточности информации и повышение целостности данных. Применение внешних ключей и ограничений целостности позволило предотвратить возникновение логических ошибок и обеспечить согласованность информации независимо от способа доступа к базе данных.

Практическая реализация базы данных с использованием языка SQL продемонстрировала возможность эффективного применения стандартных механизмов СУБД для автоматизации процессов учёта и контроля выполнения задач. Использование триггеров позволило автоматизировать фиксацию даты завершения задач и минимизировать влияние человеческого фактора при вводе данных. Реализация аналитических запросов и представлений обеспечила возможность мониторинга текущего состояния задач, выявления просроченных заданий и анализа распределения нагрузки между пользователями.

Полученные результаты подтверждают, что разработанная база данных может служить надёжной основой для создания корпоративных информационных систем управления задачами. Предложенное решение является масштабируемым и может быть адаптировано под различные предметные области и условия эксплуатации. В дальнейшем структура базы данных может быть расширена за счёт внедрения механизмов разграничения прав доступа, журналирования изменений и интеграции с прикладными веб-интерфейсами, что повысит функциональные возможности системы и её практическую ценность.

Библиографический список

1. Крамаренко Т. А., Лукьяненко Т. В. Методики и модели проектирования и разработки информационных систем: монография. Краснодар: КубГАУ, 2018. 175 с. URL: https://elibrary.ru/download/elibrary_36805515_60814187.pdf (дата обращения: 10.01.2026).
2. Минцаев М. Ш., Алисултанова Э. Д., Мидаева А. А. Исследование и разработка автоматизированной информационной системы для учета самозанятых студентов // Вестник ГГНТУ. Технические науки. 2023. Т. 19. № 2. С. 25–30. URL: <https://scholar.archive.org/work/uewpjzlmejfhdcgfa13f4rfzum/access/wayback/lib.gstou.ru/articles/bx0LC3rvu4IomtYgWkgOt983UXkmfXxKVHVkDdEl.pdf> (дата обращения: 10.01.2026).
3. Ромашкова О. Н., Ермакова Т. Н. Алгоритм работы с модулем «Учебная деятельность» управленческой информационной системы для образовательного комплекса // Исследование различных направлений современной науки. Материалы VIII Международной научно-практической конференции. 2016. С. 917–920. URL:

- https://elibrary.ru/download/elibrary_25658429_46797006.pdf#page=917
(дата обращения: 10.01.2026).
4. Емельянова Ю. А., Одинцева А. В. Разработка базы данных информационной системы управления рисками для предметных областей // Молодежь и современные информационные технологии. Сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. 2016. Т. 2. URL: https://earchive.tpu.ru/bitstream/11683/16994/1/conference_tpu-2015-C04-v2-043.pdf (дата обращения: 10.01.2026).
 5. Круликовский А. П., Агеева К. Г. Проектирование информационной системы учёта товаров // Теория и практика экономики и предпринимательства. Труды XX Международной научно-практической конференции. 2023. С. 20–22. URL: https://elibrary.ru/download/elibrary_51623346_48975045.pdf#page=246 (дата обращения: 10.01.2026).
 6. MySQL Documentation. MySQL 8.0 Reference Manual // MySQL. 2026. URL: <https://dev.mysql.com/doc/refman/8.0/en/> (дата обращения: 10.01.2026).
 7. PostgreSQL Documentation. PostgreSQL Official Documentation // PostgreSQL. 2026. URL: <https://www.postgresql.org/docs/> (дата обращения: 10.01.2026).
 8. Oracle Corporation. Oracle Database SQL Language Reference // Oracle. 2026. URL: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/> (дата обращения: 10.01.2026).
 9. Microsoft SQL Server Documentation. SQL Server Documentation // Microsoft Learn. 2026. URL: <https://learn.microsoft.com/en-us/sql/> (дата обращения: 10.01.2026).
 10. SQLite Documentation. SQLite Official Documentation // SQLite. 2026. URL: <https://www.sqlite.org/docs.html> (дата обращения: 10.01.2026).