



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ
КАФЕДРА

ФАКУЛЬТЕТ МЕЖДУНАРОДНЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ (ФМОП)
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)

КУРСОВАЯ РАБОТА

НА ТЕМУ:

*«Разработка базы данных для прогнозирования
вероятности задержки рейса в городе-пересадки из-
за технического обслуживания»*

Студент

ИУ7И-61Б

(группа)

(подпись, дата)

Везирова Й.Н.

(И.О. Фамилия)

Руководитель курсового проекта

(подпись, дата)

Строганов Ю.В.

(И.О. Фамилия)

2024 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой

(индекс)

Рудаков И.В.

(И.О. Фамилия)

3 марта 2024 г.

(дата)

(подпись)

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине Базы Данных

Студент группы ИУ7И-61Б Везирова Йована Недялкова

(Фамилия, имя, отчество)

Тема курсовой работы Разработка базы данных для прогнозирования вероятности задержки рейса в городе-пересадки из-за технического обслуживания

Направленность КР (учебная, исследовательская, практическая, производственная, др.)
Учебная

Источник тематики (кафедра, предприятие, НИР) Кафедра

Задание

Провести анализ предметной области на примере существующих авиакомпаний и перелетов, а также рассмотреть аналоги разрабатываемого продукта. Спроектировать сущности базы данных, функции для доступа к необходимым полям базы данных для вычисления вероятности задержки, триггеры для обновления данных в базе, а также ролевую модель на уровне базы данных. Выбрать средства реализации базы данных и приложения, а также описать интерфейс доступа к базе данных. Провести сравнительный анализ работы со строками в хранимых процедурах, в компонентах промежуточного уровня или на клиенте.

Оформление курсовой работы:

Расчетно-пояснительная записка (Отчет по КР) на 25-40 листах формата А4. Презентация на 12-18 слайдах.

Дата выдачи задания «5» марта 2024 г.

Руководитель курсовой работы

(подпись,
дата)

Строганов Ю.В.

(И.О. Фамилия)

Студент

(подпись,
дата)

Везирова Й.Н.

(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

РЕФЕРАТ

Расчетно-пояснительная записка 33 с., 15 рис., 3 лист., 14 ист., 1 прил.
БАЗА ДАННЫХ, PostgreSQL, СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ, Python, SQLAlchemy, РЕЛЯЦИОННАЯ МОДЕЛЬ.

Цель работы: разработка базы данных для прогнозирования вероятности задержки рейса в городе–пересадки из–за технического обслуживания.

Данный курсовой проект включает в себя анализ, проектирование и разработку базы данных для прогнозирования вероятности задержки рейса в городе–пересадки из–за технического обслуживания. В качестве системы управления базами данных была выбрана PostgreSQL, а для взаимодействия с базами данных в объектно–ориентированной парадигме была выбрана технология SQLAlchemy с реализацией на языке программирования Python.

На базе результатов анализа были разработаны база данных, хранимая процедура и триггер для работы с базой данных.

СОДЕРЖАНИЕ

РЕФЕРАТ	3
ВВЕДЕНИЕ	3
1 Аналитический раздел	4
1.1 Анализ предметной области	4
1.2 Анализ существующих решений	6
1.3 Формализация информации, хранимой в базе данных	7
1.4 Система ролей в базе данных	9
1.5 Выбор модели базы данных	12
1.5.1 Дореляционные СУБД	12
1.5.2 Реляционная модель данных	13
2 Конструкторский раздел	15
2.1 Проектируемая база данных	15
2.2 Описание используемого триггера	18
2.3 Описание используемой хранимой процедуры	19
2.4 Система ролей в базе данных	21
3 Технологический раздел	23
3.1 Технологии, используемые при создании ПО	23
3.2 Архитектура ПО	23
3.3 Реализация триггера	24
3.4 Реализация хранимой процедуры	24
3.5 Реализация системы ролей	26
3.6 Примеры работы	26
4 Исследовательская часть	28
4.1 Технические характеристики	28
4.2 Сравнительный анализ работы со строками в хранимой процедуре	28
Выводы	29

ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	31
ПРИЛОЖЕНИЕ А	33

ВВЕДЕНИЕ

Воздушные перевозки являются одним из наиболее предпочтительных и быстрых видов транспорта. Самолеты — это самые известные средства воздушного транспорта и могут использоваться не только для коммерческих/некоммерческих пассажирских перевозок, но и для транспортировки грузов и оборудования. Объем коммерческих авиаперевозок непрерывно растет во всем мире на протяжении десятилетий, поэтому для поддержания или улучшения качества сервиса, необходимо удовлетворять ожидания пассажиров. Поскольку пунктуальность оказывает влияние на предпочтение пассажиров при выборе авиалиний, крайне важно, чтобы рейсы четко придерживались своего расписания [1].

Заблаговременное информирование о вероятности задержки рейса пассажиров позволяет последним принимать более рациональные решения, включая выбор альтернативных маршрутов или распределение времени для учета возможных задержек. Авиакомпании, в свою очередь, могут использовать такие прогнозы для управления ожиданиями пассажиров и реализации мер по снижению потенциальных негативных воздействий на их путешествия [2].

Целью работы является разработка базы данных для прогнозирования вероятности задержки рейса в городе–пересадки из-за технического обслуживания.

Для достижения цели необходимо выполнить следующие задачи:

- 1) провести анализ предметной области;
- 2) спроектировать и разработать базу данных в соответствии с поставленной задачей;
- 3) выбрать средства реализации базы данных и приложения;
- 4) провести сравнительный анализ работы со строками в хранимой процедуре.

1 Аналитический раздел

В данном разделе будет проведен анализ предметной области, существующих решений и формулировка требований к разрабатываемой базе данных.

1.1 Анализ предметной области

Операции по техническому обслуживанию самолетов широко распространены внутри стран и между ними и выполняются как военными, так и гражданскими механиками. Механики работают в аэропортах, на ремонтных базах, военных объектах. Механиков нанимают пассажирские и грузовые перевозчики, подрядчики по техническому обслуживанию, сельскохозяйственные предприятия, а также владельцы государственного и частного флота. Небольшие аэропорты могут обеспечить работой несколько механиков, в то время как в крупных узловых аэропортах и на базах технического обслуживания могут работать тысячи. Работы по техническому обслуживанию делятся на те, которые необходимы для поддержания текущих повседневных операций (линейное техническое обслуживание), и на те, в процессе которых периодически проверяют, обслуживают и ремонтируют самолет (базовое техническое обслуживание). Линейное техническое обслуживание включает в себя техническое обслуживание в пути (между посадкой и взлетом) и техническое обслуживание в ночное время. Техническое обслуживание в пути состоит из эксплуатационных проверок и необходимого в полете ремонта для устранения несоответствий, обнаруженных во время полета. Этот ремонт, как правило, незначителен, например, замена сигнальных ламп, шин и компонентов авионики, но может быть таким же масштабным, как замена двигателя. Ночное техническое обслуживание более обширно и включает в себя любой ремонт, отложенный во время дневных полетов.

Сроки, распределение работников и характер технического обслуживания самолетов контролируются каждой авиакомпанией и документируются в ее руководстве по техническому обслуживанию. Эти плановые мероприятия по техническому обслуживанию обеспечивают проверку, техни-

ческое обслуживание и ремонт всего воздушного судна через соответствующие промежутки времени. Проверки технического обслуживания более низкого уровня могут быть включены в работы по техническому обслуживанию линии, но более обширные работы выполняются на базе технического обслуживания. Повреждения самолета и отказы компонентов устраняются по мере необходимости [3].

Время оборота самолета (*англ.* TAT — turnaround time) — это общее время, которое требуется для выполнения всех необходимых действий для подготовки самолета к следующему рейсу. Из-за ограниченного пространства вокруг самолета, операции оборота должны выполняться в точной хронологической последовательности: некоторые из них обязательно должны выполняться последовательно, в то время как другие могут выполняться одновременно. Эффективность оборота самолета можно определить как способность выполнения необходимых операций в доступное время для обеспечения пунктуального вылета.

Наземное обслуживание самолета — это сервисный процесс с жесткими требованиями, включающий более 10 взаимосвязанных подпроцессов и вовлекающий более 30 различных сотрудников.

На рисунке 1.1 представлена схема процесса наземного обслуживания самолета, включающая три основные категории обслуживания: пассажиров, багажа и груза, а также самого самолета.

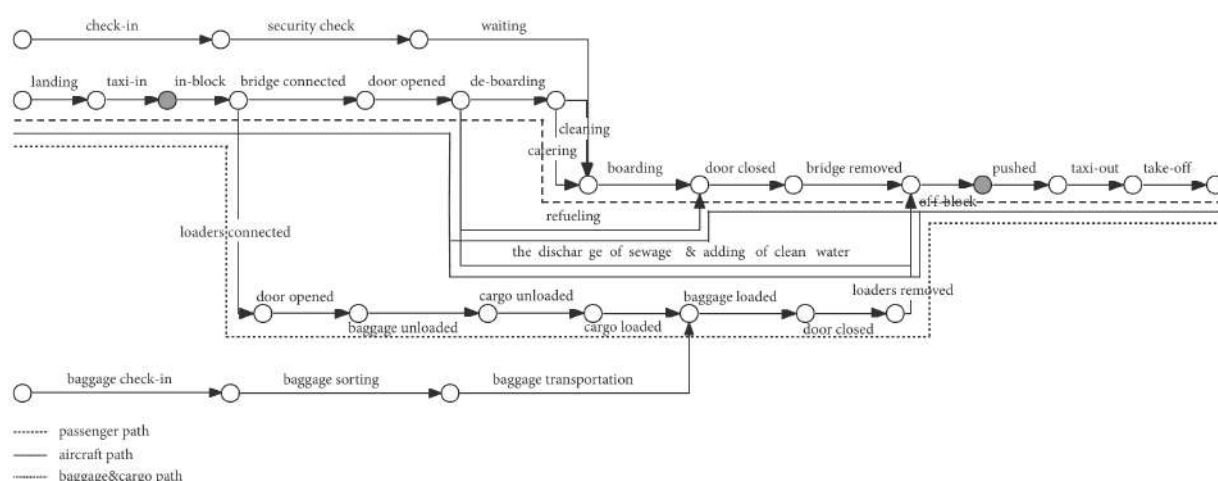


Рисунок 1.1 – Схема процессов наземного обслуживания самолета

Пассажиры проходят регистрацию, проверку безопасности и ожидают посадки. После посадки самолета пассажиры выходят через подклю-

ченный мост и проходят высадку. После уборки и обслуживания питания начинается посадка пассажиров на следующий рейс.

Багаж регистрируется, сортируется и транспортируется к самолету. После посадки самолета багаж выгружается и новый багаж загружается. После завершения загрузки багажных отсеков двери закрываются.

Самолет проходит этапы от посадки до взлета, включая такси, подключение моста, открытие дверей, разгрузку и загрузку багажа и груза, заправку, уборку, обслуживание питания и техническое обслуживание. После завершения всех операций, мост убирается, самолет выталкивается, проходит такси и взлетает.

Эта схема иллюстрирует важность координации и последовательности действий для обеспечения эффективного и своевременного оборота самолета [4].

1.2 Анализ существующих решений

Существует несколько решений в сфере мониторинга полетов, которые обеспечивают отслеживание воздушного движения в реальном времени:

- 1) Flightradar24 — это глобальный сервис отслеживания рейсов, который предоставляет информацию о 1000 воздушных судов по всему миру в режиме реального времени [5];



Рисунок 1.2 – Пример отслеживания полетов на Flightradar24

- 2) FlightAware — это цифровая авиационная компания, которая управляет крупнейшей в мире платформой отслеживания полетов и сбора данных. Благодаря глобальной связи с каждым сегментом авиации, FlightAware предоставляет более 10000 эксплуатантам воздушных судов и поставщикам услуг, а также более 13000 000 пассажиров глобальные решения для отслеживания полетов, технологии прогнозирования, аналитики и инструменты принятия решений [6].

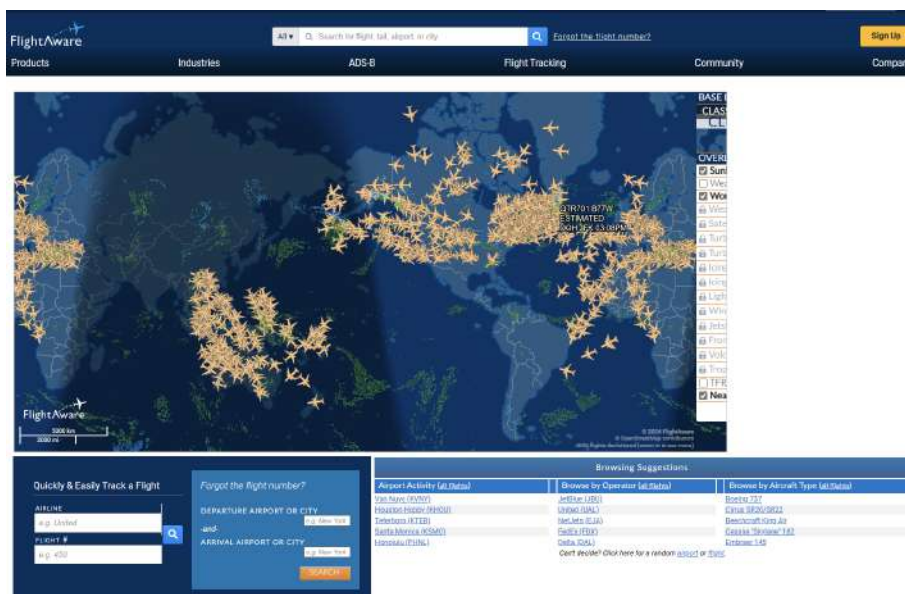


Рисунок 1.3 – Пример отслеживания полетов на FlightAware

1.3 Формализация информации, хранимой в базе данных

Для наполнения данными базы данных были использованы датасеты с платформы Kaggle, что позволило провести более глубокий анализ полученных результатов. Это платформа для машинного обучения и аналитики данных, которая предоставляет доступ к множеству датасетов, подготовленных сообществом [7]. На платформе Kaggle были найдены датасеты, содержащие информацию о задержках рейсов, аэропортах, авиакомпаниях и полетах.

После анализа датасетов были выделены следующие сущности:

- 1) полет;

- 2) задержка;
- 3) аэропорт;
- 4) авиакомпания;
- 5) самолет;
- 6) экипаж;
- 7) отчет;
- 8) система.

На рисунке 1.4 представлена ER–диаграмма сущностей системы в нотации Чена.

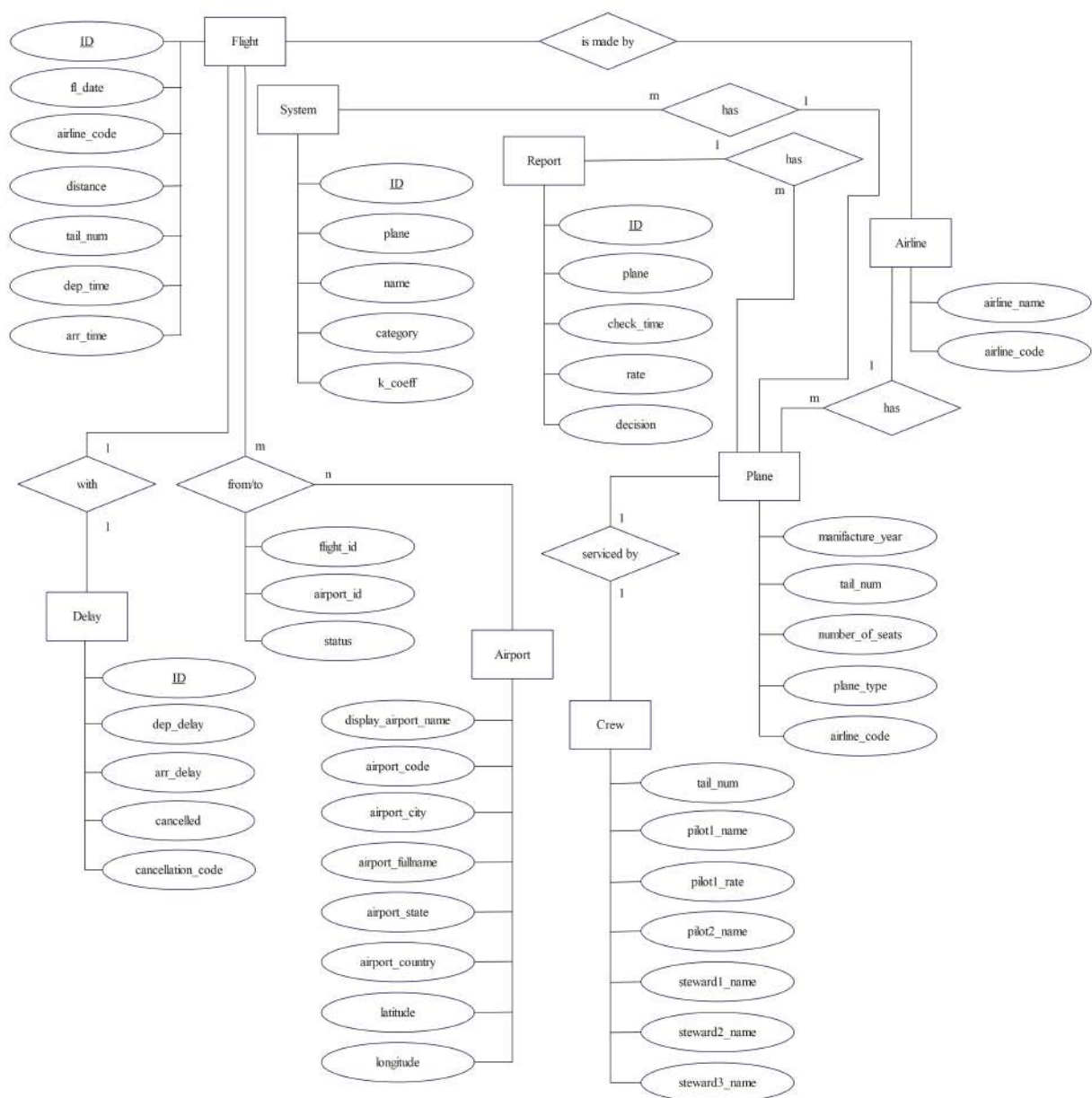


Рисунок 1.4 – ER – диаграмма для базы данных

1.4 Система ролей в базе данных

Ролевая модель в базе данных состоит из трех ролей:

- 1) гость — пользователь, у которого есть доступ к чтению сущностей полета, аэропорта, авиакомпании, задержки, а также возможность получить информацию о вероятности задержки конкретного перелета;



Рисунок 1.5 – Use-case диаграмма для роли "гость"

- 2) сотрудник — пользователь, который имеет доступ к чтению и записи сущностей отчета и системы;



Рисунок 1.6 – Use-case диаграмма для роли "сотрудник"

- 3) администратор — пользователь, имеющий полный контроль над всеми сущностями, а также возможность получить информацию о вероятности задержки конкретного перелета.



Рисунок 1.7 – Use-case диаграмма для роли "администратор"

1.5 Выбор модели базы данных

По модели хранения данных выделяют следующие СУБД (*англ.* DBMS — database management system):

- 1) дореляционные;
- 2) реляционные;
- 3) постреляционные.

Ниже рассмотрены основные модели баз данных.

1.5.1 Дореляционные СУБД

1. Инвертированные списки — БД на основе инвертированных списков представляет собой совокупность файлов, содержащих записи (таблиц). Для записей в файле определен некоторый порядок, диктуемый физической организацией данных. Для каждого файла может быть определено произвольное число других упорядочений на основании значений некоторых полей записей (инвертированных списков). Обычно для этого используются индексы. В такой модели данных отсутствуют ограничения целостности как таковые. Все ограничения на возможные экземпляры БД задаются теми программами, которые работают с БД. Одно из немногих ограничений, которое все-таки может присутствовать - это ограничение, задаваемое уникальным индексом [8].
2. Иерархическая модель данных — иерархическая модель БД состоит из объектов с указателями от родительских объектов к потомкам, соединяя вместе связанную информацию. Иерархические БД могут быть представлены в виде дерева [8].
3. Сетевая модель данных — к основным понятиям сетевой модели БД относятся: элемент (узел), связь. Узел — это совокупность атрибутов данных, описывающих некоторый объект. Сетевые БД могут быть

представлены в виде графа. В сетевой БД логика процедуры выборки данных зависит от физической организации этих данных. Поэтому эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение [8].

На рисунке 1.8 представлена сетевая модель данных.

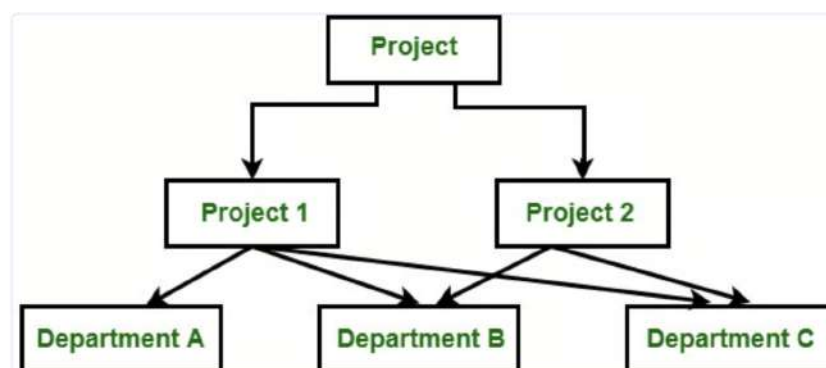


Рисунок 1.8 – Сетевая модель данных

1.5.2 Реляционная модель данных

Реляционная модель данных включает в себя следующие аспекты:

- 1) структурный аспект — данные в базе данных представлены в виде набора отношений, также известных как таблицы.
- 2) целостностный аспект — отношения (таблицы) должны соответствовать определенным условиям целостности, гарантирующим правильное и надежное хранение данных.
- 3) манипуляционный аспект — манипулирование данными в отношениях осуществляется с помощью средств реляционной алгебры и/или реляционного исчисления. Это позволяет выполнять различные операции над данными, такие как выборка, вставка, обновление и удаление. Кроме того, реляционная модель данных включает в себя теорию нормализации, которая помогает улучшить организацию данных в базе данных и избежать избыточности и аномалий. В 1985 году доктор

Кодд сформулировал двенадцать правил, которым должна соответствовать настоящая реляционная база данных. Эти правила являются полуофициальным определением понятия реляционной базы данных и предоставляют стандарт для оценки соответствия реляционной базы данных этой модели [8].

Выводы

В данном разделе проанализированы техническое обслуживание самолетов, процесс оборота самолета и наземное обслуживание. Определены требования к разрабатываемой базе данных, выделены основные сущности для хранения данных. Разработана система ролей для пользователей на уровне базы данных. Выбрана реляционная модель данных для реализации базы данных.

2 Конструкторский раздел

В данном разделе будут описаны проектируемая база данных, используемый триггер и хранимая процедура, а также система ролей в базе данных.

2.1 Проектируемая база данных

На рисунке 2.1 представлена диаграмма проектируемой базы данных.

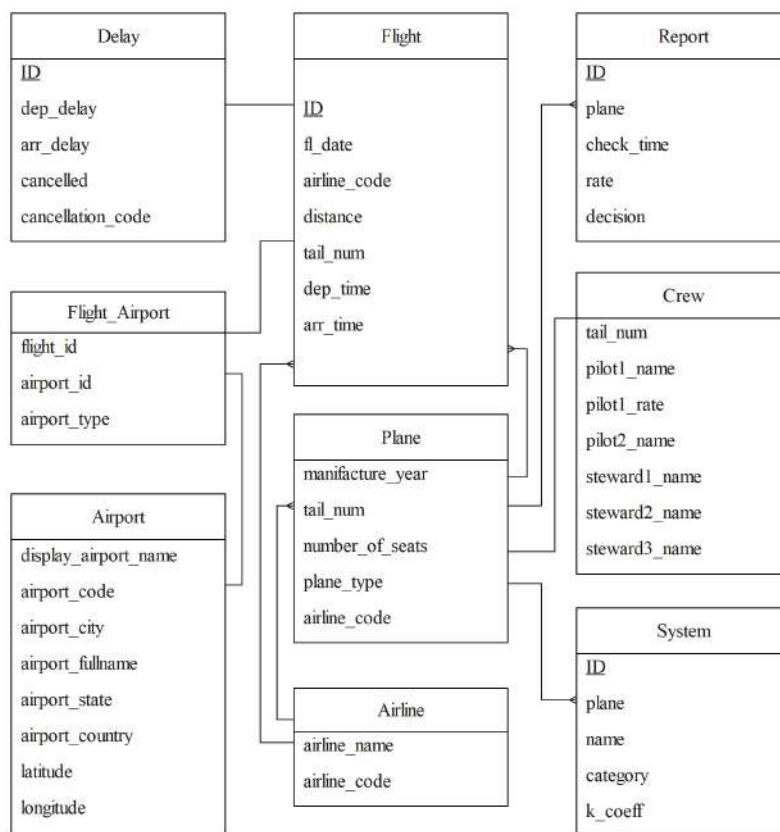


Рисунок 2.1 – Диаграмма проектируемой базы данных

В таблицах 2.1– 2.9 представлены описания полей соответствующих таблиц базы данных:

Таблица 2.1 – Описание таблицы flight

Атрибут	Тип данных
Идентификатор	Целое число
Дата вылета	Дата
Код авиакомпании	Строка
Время вылета	Время
Время прилета	Время
Номер борта	Строка

Таблица 2.2 – Описание таблицы delay

Атрибут	Тип данных
Идентификатор	Целое число
Задержка вылета	Вещественное число
Задержка прилета	Вещественное число
Статус	Целое число
Причина отмены	Строка

Таблица 2.3 – Описание таблицы airport

Атрибут	Тип данных
Название	Строка
Код аэропорта	Строка
Город	Строка
Полное имя	Строка
Страна	Строка
Географская ширина	Вещественное число
Географская высота	Вещественное число

Таблица 2.4 – Описание таблицы airline

Атрибут	Тип данных
Название	Строка
Код авиакомпании	Строка

Таблица 2.5 – Описание таблицы plane

Атрибут	Тип данных
Год выпуска	Целое число
Номер борта	Строка
Количество мест	Целое число
Модель	Строка
Код авиакомпании	Строка

Таблица 2.6 – Описание таблицы stew

Атрибут	Тип данных
Номер борта	Строка
ФИО первого пилота	Строка
ФИО второго пилота	Строка
ФИО первого стюарда	Строка
ФИО второго стюарда	Строка
ФИО третьего стюарда	Строка

Таблица 2.7 – Описание таблицы report

Атрибут	Тип данных
Идентификатор	Целое число
Номер борта	Строка
Время создания отчета	Время
Оценка	Вещественное число
Решение	Логический тип

Таблица 2.8 – Описание таблицы system

Атрибут	Тип данных
Идентификатор	Целое число
Номер борта	Строка
Имя	Строка
Категория	Строка
Коэффициент износа	Вещественное число

Таблица 2.9 – Описание таблицы flight_airport

Атрибут	Тип данных
Идентификатор полета	Целое число
Идентификатор аэропорта	Целое число
Тип аэропорта	Строка

2.2 Описание используемого триггера

Для автоматического обновления данных в таблице system при добавлении нового отчета о самолете в таблицу report был создан триггер, который автоматически обновляет поле k_coeff всех систем таблицы system, которые относятся к самолету.

На рисунке 2.2 изображена схема алгоритма функции триггера.

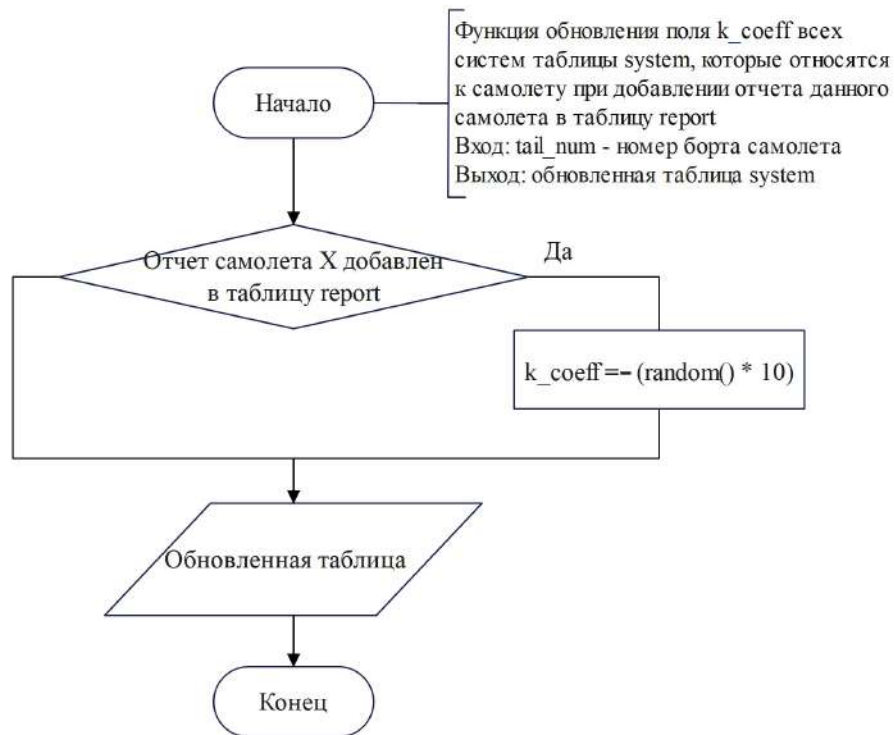


Рисунок 2.2 – Схема алгоритма функции триггера

2.3 Описание используемой хранимой процедуры

Для получения информации о вероятности задержки рейса была создана хранимая процедура, которая принимает на вход идентификаторы полета, аэропортов взлета и посадки, а также дату, до которой производится перебор всех полетов для вычисления вероятности задержки.

На рисунке 2.3 изображена схема алгоритма хранимой процедуры.

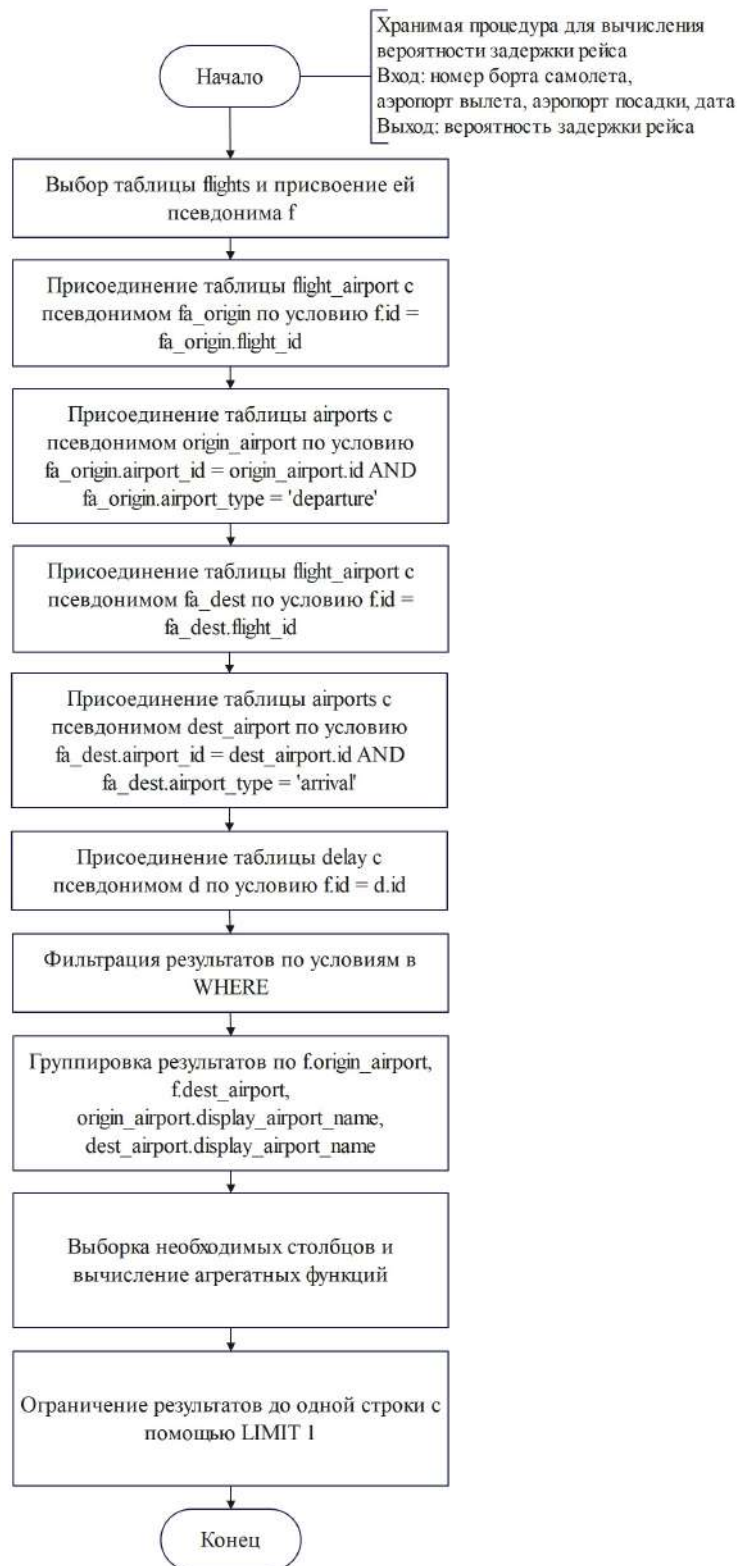


Рисунок 2.3 – Схема алгоритма хранимой процедуры

Для вычисления вероятности задержки рейса была использована фор-

мула:

$$P = \frac{N_{\text{задержанных}}}{N_{\text{всех}}}, \quad (2.1)$$

где $N_{\text{задержанных}}$ — количество задержанных рейсов, $N_{\text{всех}}$ — общее количество рейсов.

Вероятность задержки рассчитывается как отношение количества задержанных рейсов к общему числу рейсов для конкретного самолета, места взлета и посадки до указанной даты.

2.4 Система ролей в базе данных

В контексте разработанной базы данных были определены три основные роли: гость, сотрудник и администратор. Эти роли были созданы для обеспечения разграничения доступа к данным и функциональности базы данных.

- Роль “Гость” предоставляет возможность просмотра информации о полетах, аэропортах, авиакомпаниях и задержках, а также может получить информацию о вероятности задержки конкретного перелета. Это обеспечивает доступ к общедоступной информации без возможности ее изменения.
- Роль “Сотрудник” предоставляет возможность просмотра, добавления или изменения информации о системах самолетов и отчетах. Это позволяет сотрудникам обновлять и поддерживать актуальность данных, необходимых для их работы, при этом ограничивая их доступ только к определенным частям базы данных.
- Роль “Администратор” предоставляет полный доступ к функциональности базы данных, включая просмотр, добавление или изменение информации о полетах, аэропортах, авиакомпаниях, самолетах, задержках, экипаже, системах самолетов и отчетах. Администратор также может получить информацию о вероятности задержки конкретного перелета.

Система ролей является важным элементом обеспечения безопасности и целостности данных. Она позволяет контролировать доступ к данным и предотвращает несанкционированные изменения или удаление информации.

3 Технологический раздел

В этом разделе осуществляется выбор инструментов для реализации, предоставляется описание разрабатываемой базы данных и ее структуры, а также приводятся примеры работы программы.

3.1 Технологии, используемые при создании ПО

Для выполнения данной курсовой работы был выбран язык *Python* [9].

Для взаимодействия с базами данных в объектно-ориентированной парадигме была выбрана технология *SQLAlchemy* [10].

Для работы с базой данных была выбрана СУБД *PostgreSQL* [11].

3.2 Архитектура ПО

В процессе разработки были реализованы следующие структуры:

- 1) Flight — описывает информацию о полете;
- 2) Delay — описывает информацию о задержке;
- 3) Airport — описывает информацию об аэропорте;
- 4) Airline — описывает информацию об авиакомпании;
- 5) Plane — описывает информацию о самолете;
- 6) Crew — описывает информацию о членах экипажа;
- 7) Report — описывает информацию о отчетах;
- 8) System — описывает информацию о системах;

Также была создана функция, проверяющая наличие необходимой роли у пользователя для выполнения отправленного запроса.

3.3 Реализация триггера

В листинге 3.1 представлен триггер для обновления данных в таблице system при добавлении нового отчета о самолете в таблицу report.

Листинг 3.1 – Триггер для обновления данных в таблице system

```
1 def create_trigger(db: Session):
2     sql = """
3     CREATE OR REPLACE FUNCTION decrease_k_coeff() RETURNS
4     TRIGGER AS $$
5     BEGIN
6     UPDATE systems
7     SET k_coeff = k_coeff - (random() * 10)
8     WHERE plane = NEW.plane;
9     RETURN NEW;
10    $$ LANGUAGE plpgsql;
11
12    CREATE TRIGGER decrease_k_coeff_trigger
13    AFTER INSERT ON reports
14    FOR EACH ROW EXECUTE FUNCTION decrease_k_coeff();
15    """
16    db.execute(text(sql))
```

3.4 Реализация хранимой процедуры

В листинге 3.2 представлена хранимая процедура для получения вероятности задержки рейса.

Листинг 3.2 – Хранимая процедура для получения вероятности задержки рейса

```
1 CREATE OR REPLACE PROCEDURE probability(
2 plane VARCHAR,
3 origin VARCHAR,
4 destination VARCHAR,
5 date_limit VARCHAR
6 )
7 LANGUAGE plpgsql
```

```

8 AS $$
9 BEGIN
10     CREATE TEMP TABLE IF NOT EXISTS temp_result (
11         airport VARCHAR,
12         average_delay FLOAT,
13         total_delays INT,
14         delay_percentage FLOAT
15     ) ON COMMIT DROP;
16
17     INSERT INTO temp_result
18     SELECT
19         dest_airport.display_airport_name AS airport ,
20         AVG(d.dep_delay) as average_delay ,
21         COUNT(d.dep_delay) as total_delays ,
22         ((CAST(COUNT(d.dep_delay) AS FLOAT) /
23         (SELECT COUNT(*) FROM flights WHERE tail_num = plane
24         AND fl_date::DATE < date_limit::DATE)) * 100.0) as
25         delay_percentage
26     FROM
27         flights f
28     JOIN
29         flight_airport fa_origin ON f.id = fa_origin.flight_id
30     JOIN
31         airports origin_airport ON fa_origin.airport_id =
32         origin_airport.id AND fa_origin.airport_type = 'departure'
33     JOIN
34         flight_airport fa_dest ON f.id = fa_dest.flight_id
35     JOIN
36         airports dest_airport ON fa_dest.airport_id =
37         dest_airport.id AND fa_dest.airport_type = 'arrival'
38     JOIN
39         delay d ON f.id = d.id
40     WHERE
41         f.tail_num = plane AND
42         origin_airport.airport_code = origin AND
43         dest_airport.airport_code = destination AND
44         f.fl_date::DATE < date_limit::DATE
45     GROUP BY
46         dest_airport.display_airport_name;
47 END;
48 $$;

```

3.5 Реализация системы ролей

В листинге 3.3 представлен код создания ролей в системе, в котором выдаются права на определенные таблицы в базе данных.

Листинг 3.3 – Создание ролей в системе

```
1 CREATE ROLE guest ;
2 CREATE ROLE employee ;
3 CREATE ROLE admin ;
4
5 GRANT SELECT ON flights , airports , airlines , delay TO guest ;
6
7 GRANT ALL PRIVILEGES ON reports , systems , checklist TO employee ;
8
9 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO admin ;
```

3.6 Примеры работы

На рисунках 3.1, 3.2, 3.3 представлены примеры работы приложения.

— авторизация сотрудника;

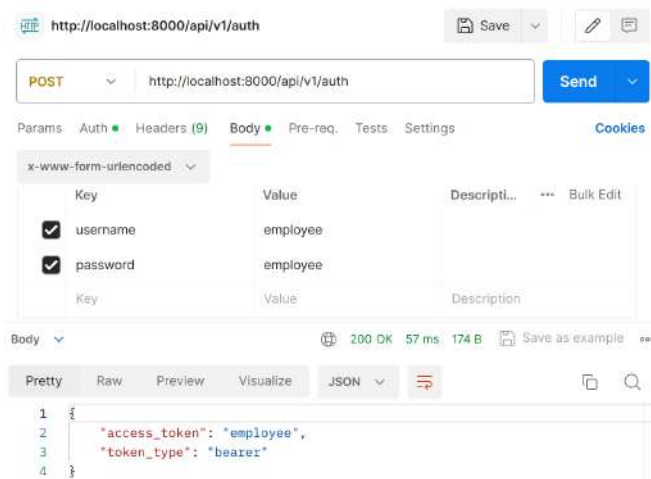


Рисунок 3.1 – Пример работы программы

— получение информации о полете;

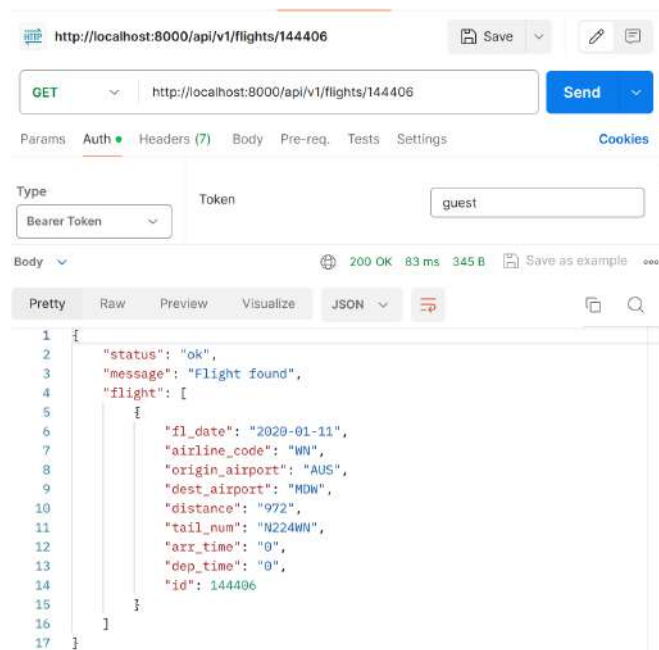


Рисунок 3.2 – Пример работы программы

— получение информации о вероятности задержки полета;

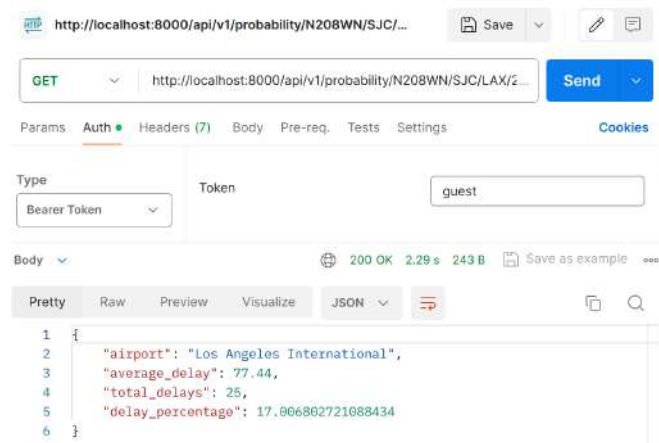


Рисунок 3.3 – Пример работы программы

4 Исследовательская часть

В данном разделе будет проведен анализ зависимости количества строк на время выполнения хранимой процедуры в базе данных.

4.1 Технические характеристики

Технические характеристики устройства, использовавшегося для выполнения замеров, представлены ниже.

- Процессор: AMD Ryzen 7–5800H with Radeon Graphics [12].
- Оперативная память: 16 ГБайт.
- Операционная система: Windows 11 Home [13].

При замерах времени ноутбук был включен в сеть электропитания и был нагружен только системными приложениями.

4.2 Сравнительный анализ работы со строками в хранимой процедуре

Время выполнения процедуры представляет собой критический параметр, определяющий производительность базы данных. Более длительное время выполнения может привести к замедлению работы приложения и негативному влиянию на пользовательский опыт [14].

В процессе написания курсового проекта была найдена связь между количеством строк, обрабатываемых хранимой процедурой, и временем ее выполнения при работе со строками. Для подтверждения этой гипотезы был проведен сравнительный анализ работы с разным количеством строк. В качестве хранимой процедуры была выбрана процедура, возвращающая вероятность задержки рейса.

Замеры производились на следующих параметрах:

- 1) количество строк, обрабатываемых хранимой процедурой;

- 2) время выполнения — рассчитывается как среднее значение из 10 измерений.

На рисунке 4.1 представлен график зависимости времени выполнения хранимой процедуры от количества строк.

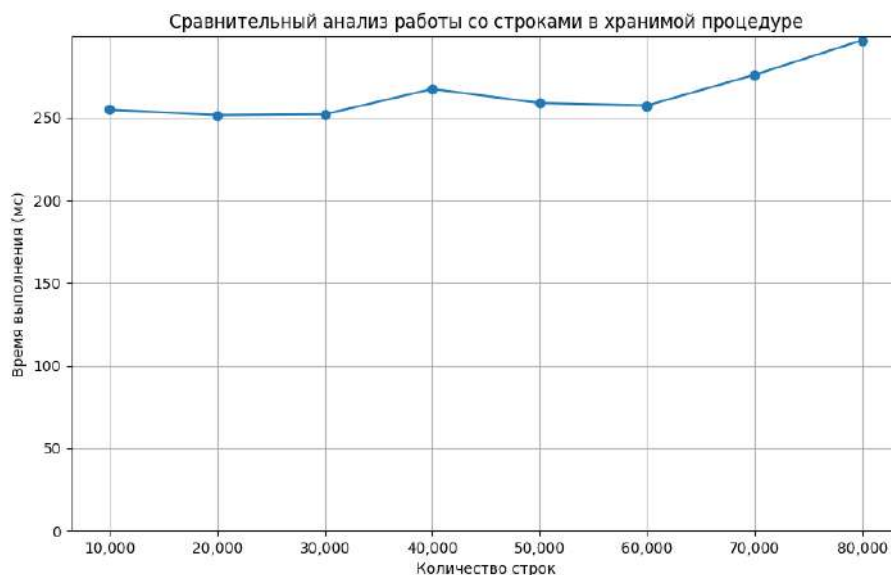


Рисунок 4.1 – График сравнительного анализа работы со строками в хранимой процедуре

Выводы

В данном разделе был проведен сравнительный анализ работы с разным количеством строк в хранимой процедуре. По результатам анализа было установлено, что время выполнения хранимой процедуры зависит от количества обрабатываемых строк — с увеличением количества строк время выполнения хранимой процедуры линейно увеличивается. Результаты анализа позволяют сделать выводы о том, что для оптимизации работы с базой данных необходимо уменьшать количество строк, обрабатываемых хранимой процедурой.

ЗАКЛЮЧЕНИЕ

Курсовой проект включал в себя анализ, проектирование и разработку базы данных для прогнозирования вероятности задержки рейса в городе–пересадки из–за технического обслуживания.

На базе результатов анализа были разработаны база данных, хранимая процедура и триггер для работы с базой данных.

Сравнительный анализ в области работы с строками в хранимой процедуре был проведен на основе замеров времени выполнения. Было установлено, что время выполнения хранимой процедуры зависит от количества обрабатываемых строк — с увеличением количества строк время выполнения хранимой процедуры линейно увеличивается. Результаты анализа позволяют сделать выводы о том, что для оптимизации работы с базой данных необходимо уменьшать количество строк, обрабатываемых хранимой процедурой и использовать другие методы для работы с данными.

Цель курсовой работы была достигнута. В ходе выполнения работы были выполнены следующие задачи:

- 1) проведен анализ предметной области;
- 2) спроектирована и разработана база данных в соответствии с поставленной задачей;
- 3) выбраны средства реализации базы данных и приложения;
- 4) проведен сравнительный анализ работы со строками в хранимой процедуре.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. A turnaround control system [Электронный ресурс]. — Режим доступа: (дата обращения: 02.03.2024) <https://www.sciencedirect.com/science/article/pii/S0952197622002056>.
2. Turnaround Time: Why it's important [Электронный ресурс]. — Режим доступа: (дата обращения: 02.03.2024) Свободный <https://simpleflying.com/turnaround-time-importance/>.
3. Операции по техническому обслуживанию самолетов [Электронный ресурс]. — Режим доступа: (дата обращения: 25.05.2024) <https://iloencyclopaedia.org/ru/part-xvii-65263/transport-industry-and-warehousing/item/910-aircraft-maintenance-operations>.
4. Mechanism Analysis of the Impact of COVID-19 on the Whole Process of Aircraft Turnaround Operations [Электронный ресурс]. — Режим доступа: (дата обращения: 02.05.2024) https://www.researchgate.net/publication/379569382_Mechanism_Analysis_of_the_Impact_of_COVID-19_on_the_Whole_Process_of_Aircraft_Turnaround_Operations.
5. Live Flight Tracker - Real-Time Flight Tracker Map | Flightradar24 [Электронный ресурс]. — Режим доступа: (дата обращения: 02.05.2024) <https://www.flightradar24.com/>.
6. FlightAware - отслеживание полетов / информация о статусе рейсов [Электронный ресурс]. — Режим доступа: (дата обращения: 02.05.2024) <https://ru.flightaware.com/>.
7. Kaggle [Электронный ресурс]. — Режим доступа: (дата обращения: 05.03.2024) <https://www.kaggle.com/>.
8. Гаврилова Ю. М. Лекции по курсу "Базы данных". — Москва: МГТУ им. Н. Э. Баумана, 2023.
9. Welcome to Python [Электронный ресурс]. — Режим доступа: (дата обращения: 10.05.2024) <https://www.python.org>.

10. SQLAlchemy [Электронный ресурс]. — Режим доступа: (дата обращения: 05.05.2024) <https://www.sqlalchemy.org/>.
11. PostgreSQL [Электронный ресурс]. — Режим доступа: (дата обращения: 05.05.2024) <https://www.postgresql.org/>.
12. AMD Ryzen [Электронный ресурс]. — Режим доступа: (дата обращения: 10.05.2024) <https://www.amd.com/en.html>.
13. Windows 11 Home 64-bit [Электронный ресурс]. — Режим доступа: (дата обращения: 10.05.2024) <https://www.microsoft.com/ru-ru/software-download/windows11>.
14. Хранимые процедуры (ядро СУБД) [Электронный ресурс]. — Режим доступа: (дата обращения: 02.06.2024) <https://learn.microsoft.com/ru-ru/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver16>.

ПРИЛОЖЕНИЕ А

Презентация к курсовой работе