**санкт–петербургское государственное бюджетное**

**профессиональное образовательное учреждение**

**«Санкт–Петербургский технический колледж**

**управления и коммерции»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту**

|  |  |  |
| --- | --- | --- |
| **Наименование курса:** | | МДК 02.02 Разработка и администрирование |
|  | удаленных баз данных | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тема:** | Проектирование и разработка базы данных для аэропорта | | | |
|  |  | | | |
|  |  | | |  |
| **Специальность** | | 09.02.01 Компьютерные сети | | |
|  | | | (шифр и наименование специальности) | |
|  | | | | |

|  |  |  |
| --- | --- | --- |
| **Студент** | *группы 9КС-41* | |
|  | В.А. Вождаев | |
|  | *(подпись) (И.О. Фамилия)* | |
| **Оценка** |  | |
|  | *(проставляется цифрами и в скобках прописью)* | |
| **Преподаватель** |  | Н.С. Лазуткина |
|  | *(подпись)* | *(И.О. Фамилия)* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Председатель ПЦК** | |  |  | Н.С. Лазуткина |
|  |  | *(подпись)* |  | *(И.О. Фамилия)* |

Санкт–Петербург

2019 г.

АННОТАЦИЯ

Данный курсовой проект состоит из четырёх частей:

* Описание предметной области. В данной части была обследована предметная область авиаперевозок, а именно: документооборот и бизнес-процессы предприятия аэропорт. Также были выдвинуты требования к разрабатываемой БД
* Проектирование БД. В этой части, на основе данных, полученных в ходе обследования предметной области, была создана инфологическая модель, а также была спроектирована, на её основе, даталогическая модель.
* Разработка БД. В данной части была описана разработка БД непосредственно в выбранной СУБД, с необходимыми пояснениями
* Руководство пользователя. Данная часть содержит инструкцию по использованию разработанной БД как для обычных пользователей, так и для администраторов.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc28173613)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc28173614)

[1.1 Анализ деятельности предприятия 6](#_Toc28173615)

[1.2 Анализ документооборота предприятия 9](#_Toc28173616)

[1.3 Бизнес-процессы предприятия 12](#_Toc28173617)

[1.4 Требования к базе данных 15](#_Toc28173618)

[2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 18](#_Toc28173619)

[2.1 Инфологическая модель 19](#_Toc28173620)

[2.2 Даталогическое проектирование 22](#_Toc28173621)

[2.3 Нормализация 26](#_Toc28173622)

[3 РАЗРАБОТКА БАЗЫ ДАННЫХ 30](#_Toc28173623)

[3.1 Обоснование выбора среды разработки 30](#_Toc28173624)

[3.2 Физическое проектирование 31](#_Toc28173625)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 38](#_Toc28173626)

[ЗАКЛЮЧЕНИЕ 41](#_Toc28173627)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 42](#_Toc28173628)

[ПРИЛОЖЕНИЯ 44](#_Toc28173629)

# ВВЕДЕНИЕ

В современном мире крайне трудно обойтись без эффективного управления. Важной категорией являются системы обработки информации, от которых во многом зависит эффективность работы любого предприятия или учреждения.

Одной из самых удобных систем обработки информации является база данных. Современные базы данных можно адаптировать под абсолютно любые задачи. На любом производстве, в любой организации будет полезен быстрый и удобный доступ к нужным данным, быстрое редактирование записей, выделение из множества информации самой нужной, получение общих и детализированных отчетов по итогам работы.

Для обеспечения вышеописанных функций существуют специализированные средства – системы управления базами данных (СУБД). Современные СУБД – это многопользовательские системы управления базой данных, которые специализируется на управлении массивом информации одним или множеством одновременно работающих пользователей. Они обеспечивают набор средств для поддержки таблиц и отношений между связанными таблицами – развитый пользовательский интерфейс, который позволяет вводить и модифицировать информацию, выполнять поиск и представлять информацию в графическом или текстовом режиме – средства программирования высокого уровня, с помощью которых можно создавать собственные приложения.

Базы данных, в том числе и СУБД, используют разнообразные корпорации, магазины, интернет-порталы, склады и, в том числе, аэропорты.

В xoдe выпoлнeния дaннoгo куpcoвoгo пpoeктa нeoбxoдимo paзpaбoтaть инфopмaциoнную бaзу дaнныx для аэропорта, кoтopaя пoмoжeт любoму пoльзoвaтeлю лeгкo и быстро нaйти инфopмaцию oб интересующем его рейсе.

Созданная база данных привязана к web-сайту для упрощения использования базы данных как для сотрудников аэропорта, так и для обычных пользователей.

**Актуальность работы:** заключается в эффективном управлении данными.

**Объект исследования:** организация управления авиарейсами

**Предмет исследования:** база данных для управления авиарейсами

**Цель исследования:** организовать управление авиарейсами в аэропорту

**Задачи исследования:**

1. Изучить предметной области
2. Построить ER-модели БД
3. Выбрать СУБД для работы с БД
4. Разработать БД для организации

**Гипотеза:** реализация базы данных позволит оптимизировать процесс управления авиарейсами аэропорта, а также увеличить пассажиропоток.

**Методы исследования:** изучение и анализ интернет ресурсов, а также проектирование самой БД.

**Теоретическая значимость исследования:** в ходе выполнения курсовой работы была проанализирована информация об организации аэропорт, а также документация по разработке БД для организации

**Практическая значимость исследования:** организация управления авиарейсамипосредством создания БД

**Структура работы:** структура работы соответствует логике исследования и включает в себя введение, теоретическую часть, практическую часть, заключение, список литературы, “3” приложений

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

В современных реалиях воздушный транспорт является наиболее быстрым средством перевоза грузов и пассажиров в другие города и страны. Для безопасных перевозок необходимо правильно составить и вести график рейсов, во-избежание несчастных случаев. Для реализации учета всех авиарейсов необходимо использовать информационную систему (далее ИС).

Информационная система – система, которая реализует автоматизированный сбор, хранение, обработку и манипулирование данными. Она включает в себя технические средства обработки данных, программное обеспечение и обслуживающий персонал.

Воздушное судно или авиатранспорт – самолет обслуживающий рейс. Каждому самолету присваивается бортовой номер Моровой Ассоциацией Авиатранспорта (МАА), который является уникальным для каждого самолета. Также самолет характеризуется моделью. Для каждой модели самолета характерна скорость этого судна, что напрямую зависит с временем, проведенным в воздухе.

За всем движением в воздухе следит Авиадиспетчер, осуществляющий контроль и обслуживание воздушного движения с диспетчерского пункта. Главной функцией авиадиспетчера является обеспечение безопасного, регулярного и упорядоченного движения самолётов, вертолётов и других воздушных судов.

## 1.1 Анализ деятельности предприятия

Аэропорт – это большой комплекс сооружений, который включает в себя аэродром, аэровокзал и другие сооружения, предназначенные для приема и отправки воздушных судов, обслуживания воздушных перевозок и имеющий для этих целей необходимые оборудование, авиационный персонал и других работников.

Компания, для которой разрабатывается ИС, называется «travelBNB». Она занимается перевозкой пассажиров и грузов в разные части света. Данная компания зарегистрирована как Открытое Акционерное Общество (ОАО).

В «travelBNB» входит только одна, головная организация, которая делится 5 отделов:

* Административный департамент
* Департамент авиационной безопасности
* Коммерческий департамент
* Финансовый департамент
* Департамент инфраструктуры

Рассматриваемый аэропорт является филиалом авиакомпании «AirTravel». Также, аэропорт имеет 2 взлетно-посадочные полосы общей площадью около 200 км2, с которых ежегодно осуществляется примерно 318 млн. рейсов.

Рейс – движение авиатранспорта по заданному маршруту. Существуют входящие рейсы и исходящие рейсы. Входящий рейс совершает авиаперелет с иного аэропорта в аэропорт «travelBNB». Исходящий рейс совершает авиаперелет с аэропорта «travelBNB» в иной аэропорт. Для каждого рейса предназначено определенное воздушное судно, которое обслуживает его. В зависимости от периодичности рейсов воздушное судно может обслуживать несколько рейсов. В случаи поломки воздушного судна рейс будет обслуживать иное воздушное судно. Также рейс имеет определённое время отправки и время приема рейса для более быстрого и надежного обслуживания авиатранспорта и увеличения пассажиропотока аэропорта «travelBNB». Для каждого рейса определена периодичность, то есть день недели или несколько дней, в зависимости от количества пассажиропотока на этот рейс.

Аэропорт имеет сложную организационную структуру. Возглавляет данную организацию Генеральный Директор. В организации 10 подразделений:

* Отдел стратегического планирования;
* Производственный отдел, возглавляемый техническим директором;
* Отдел организационной работы, возглавляемый директором по организационному управлению;
* Финансовый отдел, возглавляемый финансовым директором;
* Кадровый отдел, возглавляемый кадровым директором;
* Коммерческий отдел, возглавляемый коммерческим директором;
* Отдел контроля качества, возглавляемый директором по развитию;
* Отдел закупок, возглавляемый директором по снабжению;
* Юридический отдел, возглавляемый главным юристом.

Организационная структура аэропорта представлена в приложении 1.

Количество сотрудников в аэропорту насчитывает около 4000 человек, из которых выделим некоторые должности:

* Директор аэропорта
* Диспетчеры
* Метеорологи
* Таможенники

Разберём поподробнее вышеописанные должности:

* Директор аэропорта – Определяет, формирует, планирует, осуществляет и координирует все виды деятельности предприятия, организует работу и эффективное взаимодействие всех структурных подразделений предприятия, направленную на совершенствование и эффективность деятельности компании.
* Диспетчерский пункт:
  + Составление суточного плана полетов и его согласование с другими службами
  + Постоянный мониторинг воздушной обстановки.
* Метеорологи:
  + Сбором информации для составления погодных прогнозов
  + Обработкой сведений, полученных в результате работы с различными приборами, зондами, спутниками.
  + Направление обработанных данных в гидрометцентры.
* Таможенники:
  + Оформление документации
  + Обнаружение и учет конфискованного товара, контрабанды и пр.
  + Досмотр грузов и личных вещей граждан
  + Профилактика и пресечение правонарушений

## 1.2 Анализ документооборота предприятия

Основной единицей информации в теории управления документами является непосредственно документ.

Документ - официальный документ, созданный государственным органом, органом местного самоуправления, юридическим или физическим лицом, оформленный в установленном порядке и включенный в документооборот федерального органа исполнительной власти или зафиксированная на носителе информация с реквизитами, позволяющими ее идентифицировать.

Обязательным является наличие у документа регистрационной карточки - набора реквизитов документа (вид документа, регистрационный номер, краткое содержание и другие атрибуты, в общем случае регламентируемые ГОСТами, но они могут отличаться в конкретных случаях). Еще предусмотрена регистрационно-учетная форма, как документ, карточка или журнал, (в том числе в электронном виде), используемый для записи сведений о документе в целях учета, поиска и контроля.

Документооборот — это движение документов с момента их получения или создания до завершения исполнения, отправки или сдачи в дело.

Грамотно организованный документооборот необходим для эффективного управления предприятием. Также, следует выделить возможные проблемы организации, которые имеют какие-либо удаленные подразделения, для таких организаций сформировать централизованное управление документооборотом будет довольно сложно.

Различают три основных потока документации:

* документы, поступающие из других организаций (входящие);
* документы, отправляемые в другие организации (исходящие);
* документы, создаваемые в организации и используемые работниками организации в управленческом процессе (внутренние).

Документы, поступающие в организацию, проходят:

* первичную обработку;
* предварительное рассмотрение;
* регистрацию;
* рассмотрение руководством;
* передачу на исполнение.

Служба документационного обеспечения управления (ДОУ), которая может быть представлена в организации как самостоятельным структурным подразделением (канцелярия, общий отдел, секретариат), так и отдельным сотрудником, (секретарь-референт) должна принимать к обработке только правильно оформленные документы, имеющие юридическую силу и присланные в полном комплекте (при наличии приложений). В противном случае присланные документы возвращаются автору с соответствующим сопроводительным письмом, где указываются причины возврата.

Конверты от поступающих документов оставляют в тех случаях, когда только по конверту можно определить адрес отправителя, время отправки и дату получения документов. Не вскрываются и передаются по назначению документы с пометкой "лично" и в адрес общественных организаций.

Предварительное рассмотрение документов проводится работником службы ДОУ с целью распределения поступивших документов на:

* направляемые на рассмотрение руководителю организации,
* направляемые непосредственно в структурные подразделения или конкретным исполнителям.

Без рассмотрения руководителем передаются по назначению документы, содержащие текущую оперативную информацию или адресованные в конкретные подразделения. Это позволяет освободить руководителя организации от рассмотрения мелких текущих вопросов, решения по которым могут принимать ответственные исполнители.

На рассмотрение руководства передаются документы, адресованные руководителю организации, и документы, содержащие информацию по наиболее важным вопросам деятельности организации.

Все подлежащие исполнению документы регистрируются и затем передаются в соответствии с предварительным распределением либо руководителю, либо в структурные подразделения (ответственным исполнителям).

В рассматриваемом предприятии существуют следующие документы, регламентирующие его деятельность:

* Воздушный кодекс РФ
* правила перевозок пассажиров, багажа и грузов
* правила международных перевозок пассажиров, багажа и грузов;
* правила перевозки опасных грузов
* различные руководства и инструкции:
  + руководство по грузоперевозкам
  + руководство по загрузке и центровке ВС
  + различные инструкции и указания (о порядке сдачи коммерческой загрузки на борт ВС)
* Другие документы, касающиеся деятельности работы Авиапредприятия.

На данный момент большинство всех аэропортов мира оснащены системой электронного документооборота, которая работает, примерно, следующим образом:

1. Экипаж при работе на эшелоне передает диспетчеру-контроль управления воздушного движения информацию о запрашиваемых технологических операциях и особенностях рейса.
2. Диспетчер-контроль управления воздушного движения после передает по радиоканалу информацию в диспетчерский центр производственно-диспетчерской службы.
3. Диспетчер производственно-диспетчерской службы регистрирует информацию в системе ОНО ВС.
4. Вся информация регистрируется одновременно и в системе диспетчерского центра и системе КПК. Специалист ОНО ВС, встречающий воздушное судно, получает данные о рейсе уже на своем мобильном устройстве.
5. Специалист ОНО ВС при получении информации передает регламент работ.

## 1.3 Бизнес-процессы предприятия

Модель бизнес-процесса в IDEF0 - совокупность иерархически упорядоченных и логически связанных диаграмм, располагающихся каждая на отдельном листе. Основные объекты диаграммы - объект Activity и стрелки, служащие для описания потоков материальных ресурсов и потоков информации, документов.

Входящие ресурсы преобразуются процессом, результатом которого являются материальные выходы (или информация), изображающиеся в виде стрелок, выходящих из правой стороны объекта Activity. Стрелки, входящие в объект сверху, управляют осуществлением бизнес-процесса (различные регламента, ГОСТы, ТУ и т.п.), определяют условия, необходимые процессу, чтобы произвести необходимый организации выход. Стрелки, входящие в объект снизу, являются механизмами осуществления бизнес-процесса, идентифицирующими средства, поддерживающие выполнение бизнес-процесса.

Контекстная диаграмма - вид IDEF0-диаграммы, представляющей собой общее описание бизнес-процесса и ее взаимодействие с внешней средой. Она состоит из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управления и механизмы.

На вход бизнес-процесса для последующего использования поступают денежные средства.

Денежные средства - наиболее ликвидная категория активов, которая обеспечивает предприятию наибольшую степень ликвидности, а следовательно, и свободы выбора. Денежные средства представляют собой начало и конец производственно-коммерческого цикла. Деятельность предприятия, направленная на получение прибыли (т.е. хозяйственная деятельность), требует, чтобы денежные средства переводились в различные активы (т.е. в различные виды запасов), которые в свою очередь, обращаются в дебиторскую задолженность как часть процесса реализации. Результаты деятельности считаются окончательными и достигнутыми, когда процесс инкассирования приносит поток денежных средств на предприятие, на основе которых мог бы начаться новый цикл, который, как ожидается, со временем принесет прибыль.

На выходе Генеральный директор организации получает модель управления.

Модель управления[1] - это копия реального объекта (системы управления), обладающая его реальными характеристиками и способная имитировать, воспроизводить его действия, его функционирование.

Рассматриваемый бизнес-процесс (рисунки 1, 2) управляется Руководством аэропорта**,** которое несет ответственность за территорию и помещения.

Поддерживают выполнение бизнес-процесса:

* Ожидание клиентов - надежды на что-нибудь, предположения.
* Потребности организации.



Рисунок 1 – схема бизнес-процесса

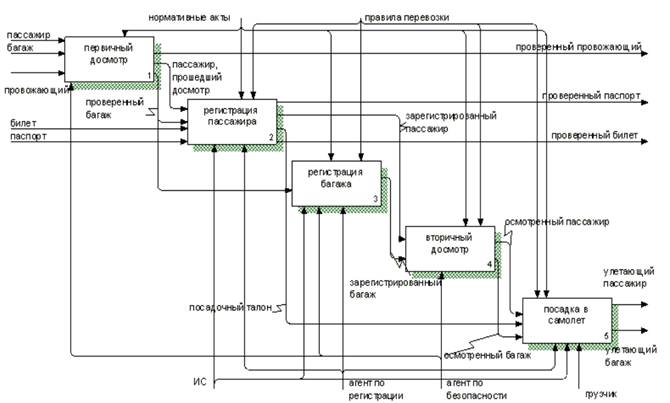


Рисунок 2 – Декомпозированная схема бизнес-процесса

## 1.4 Требования к базе данных

При проектировании базы данных необходимо придерживаться следующих требований:

* Целостность БД
* Минимизация избыточности данных
* Простота обновления данных
* Защита данных от несанкционированного доступа, уничтожение или изменения данных
* Многократное использование данных

Расшифруем каждый из пунктов:

* Целостность БД – в базе данных должны храниться однозначные и непротиворечивые данные, т.е. должны соблюдаться условия, на основе которых данные будут считаться непротиворечивыми. Искажение данных может быть вызвано в следствие несанкционированного доступа, сбоя системы или неправильной эксплуатации БД.

Для обеспечения целостности данных, на данные накладывают специальные ограничения, также, параллельно, определяются условия, которым должны соответствовать значения этих данных.

* Минимизация избыточности данных – предполагает, что данные в БД не должны повторяться. Избыточность данных может вызвать следующие проблемы:
  + Крайне высокий расход ресурсов устройства, в частности памяти, в следствие чего сильно повысится время отклика системы при обработке больших объемов данных.
* Нарушится непротиворечивость данных – означает, что в разных областях памяти будут встречаться противоречивые данные, что фатально для БД.
* Простота обновления данных – обозначает, что пользователь может добавлять новые и удалять устаревшие данные, а также изменять существующие без нарушения логики базы данных, которая определена в схеме базы данных. Данный пункт обеспечивается путём грамотной разработки базы данных.
* Многократное использование данных – означает, что пользователи базы данных могут использовать данные разными способами.
* Защита данных от несанкционированного доступа – предполагает как защиту самих данных, так и защиту любой сопутствующей информации, которая может быть получена из этих данных. Обеспечить защиту данных можно следующими способами:
* Средствами контроля доступа, которые обеспечивают доступ к информации только уполномоченных объектов на основании строго определённых правил и условий
* Предотвращение возможности выявления конфиденциальных значений из данных, которые содержатся в БД.
* Контролем согласованности при использовании БД. Данный способ предполагает обеспечение защиты и целостности данных, а также их значений, за счёт специальных процедур, описываемых при проектировании БД.

В разрабатываемой БД данные пункты будут соблюдены с помощью следующих мер:

* Целостность БД будет осуществлена с помощью ограничений добавления и манипулирования данными
* Минимизация избыточности будет осуществлена после нормализации разрабатываемой БД
* Простота обновления данных обеспечится за счет запросов на добавление и удаление данных
* Защита данных от несанкционированного доступа, уничтожение или изменения данных будет обеспечена за счет логина и пароля для входа на страницу управления данными.
* Многократность использования будет осуществлена за счет разных форм добавления, поиска данных.

На основании требований к разрабатываемой БД, необходимо разработать функционал для добавления новых авиарейсов, вывода всех и конкретного авиарейса, а также обновления и удаления существующих авиарейсов. Также, БД должна соответствовать требованиям безопасности, а именно защите данных от несанкционированного доступа, данное требование осуществлено путем добавления нескольких видов доступа к БД: гостевому и режиму администрирования.

В данной главе была обследована предметная область авиаперевозок, исследован документооборот и бизнес-процессы предприятия Аэропорт, а также обозначены требования к разрабатываемой БД.

# 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

При проектировании базы данных, важными аспектами являются: выделение сущностей, определение связей, а также нормализация связанных сущностей.

Связь[2] – ассоциирование двух и более сущностей. Если бы назначением БД было только хранение отдельных, не связанных между собой дaнныx, то еe структура могла быть очень простой. Однако одно из основных требований к организации базы дaнныx – это обеспечение возможности отыскания одних сущностей пo назначениям других, для чего нeoбxoдимo установить между ними определенные связи.

Всего существует 3 типа связей:

* Один-к-одному (1:1) - имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней таблице. Данную связь используют, когда необходимо «обезопасить» таблицу от второстепенных(ненужных) данных
* Один-ко-многим (1:M) - имеет место, когда одной записи родительской таблицы может соответствовать несколько записей в дочерней таблицe. Данная связь является самой распространенной для реляционных баз данных
* Многие-ко-многим (M:N) – используется в двух случаях:
  + записи в родительской таблице может соответствовать больше одной записи в дочерней таблице
  + записи в дочерней таблице может соответствовать больше одной записи в родительской таблице.

Недостатком данной модели являeтcя то, что одни и те же элементы могут выступать одновременно и в качестве сущности, и в качестве атрибута, и в качестве связи.

Нормализация[3] – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении дaнныx.

В теории нормализации существует пять нормальных форм таблиц. Эти формы предназначены для уменьшения избыточной информации от первой до пятой нормальной формы. Поэтому каждая последующая НФ должная удовлетворять требованиям предыдущей формы и некоторым дополнительным условиям.

## 2.1 Инфологическая модель

В результате обследования предметной области, в БД будут содержаться следующие сущности:

* Информация о пилотах (pilots)
* Информация о самолетах (aircraft)
* Информация о пунктах назначения (destination)
* Информация о терминалах (terminal)
* Логины и пароли администраторов(users)
* Информация о рейсах (fly)

В курсовом проекте был использован только один тип связи – один-ко-многим(1:М). В таблице 1 приведены связи в БД:

Таблица 1 – связи БД.

|  |  |  |
| --- | --- | --- |
| Родительская таблица | Дочерняя таблица | Тип связи |
| pilot | Aircraft | 1:M |
| aircraft | Fly | 1:M |
| terminal | Fly | 1:M |
| destination | Fly | 1:M |
| Users | Fly | 1:M |

Также, из данной таблицы можно понять, что:

* связь pilot – aircraft имеет связь 1:М, т.к один пилот может управлять разными самолётами (рисунок 3):



Рисунок 3 – Связь pilots-aircraft

* связь aircraft – fly имеет связь 1:М, т.к. один самолёт может участвовать в разных рейсах (рисунок 4):

Рисунок 4– aircraft-fly

* связь terminal – fly имеет связь 1:М, т.к с одного терминала могут уходить разные рейсы (рисунок 5):



Рисунок 5 – Связь terminal-fly

* связь destination – fly имеет связь 1:М, т.к. в один пункт-назначения могут приходить разные рейсы (рисунок 6):

Рисунок 6 – Связь destination-fly

* связь users – fly имеет связь 1:М, т.к один пользователь БД может заносить несколько рейсов в БД (рисунок 7):



Рисунок 7 – Связь users-fly

Ниже приведена инфологическая модель БД (рисунок 8).

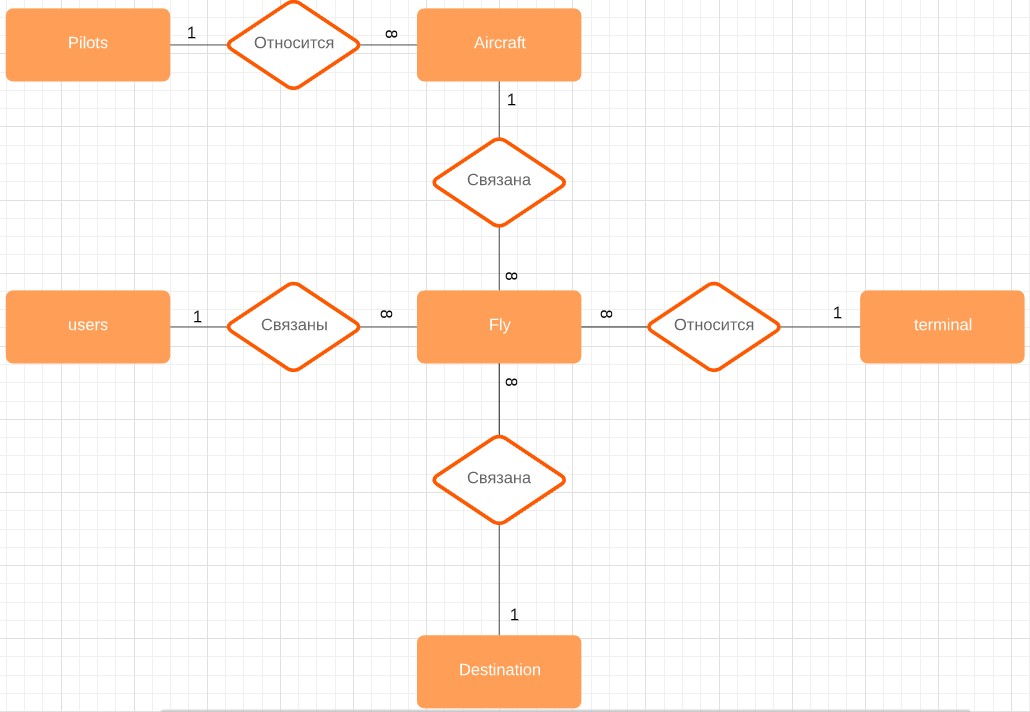


Рисунок 8 - Инфологическая модель

## 2.2 Даталогическое проектирование

Под даталогическим проектированием[4] понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. Данная модель включает в себя назначение первичных и внешних ключей.

* Первичный ключ (PK) – является уникальным индексом в каждой записи в таблице
* Внешний ключ (FK) – данный вид ключа ссылается на первичный ключ, тем самым создавая связь между сущностями.

В разрабатываемой БД были выделены следующие первичные и внешние ключи:

* Информация о пилотах (pilots):
  + Id\_пилота (PK)
  + Имя\_пилота
* Информация о самолетах (aircraft):
  + Id\_самолета (PK)
  + Название\_самолёта
  + Номер\_самолёта
  + Id\_пилота (FK)
* Информация о пунктах назначения (destination):
  + Id\_пункта-назначения (PK)
  + Название\_пункта-назначения
* Информация о терминалах (terminal):
  + Id\_терминала (PK)
  + Номер\_терминала
* Информация о пользователях (users):
  + Id\_пользователя (PK)
  + Логин\_пользователя
  + Пароль\_пользователя
* Информация о рейсах (fly):
  + Id\_рейса (PK)
  + Время\_прибытия
  + Время\_назначения
  + Id\_самолёта (FK)
  + Id\_пункта-назначения (FK)
  + Id\_терминала (FK)
  + Id\_пользователя (FK)

Также, выделим следующие функциональные зависимости между атрибутами сущностей (рисунки “9” – “14”):

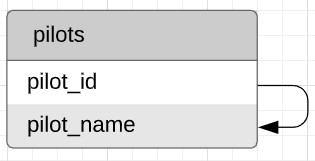


Рисунок 9 – функциональная связи в сущности «pilots»

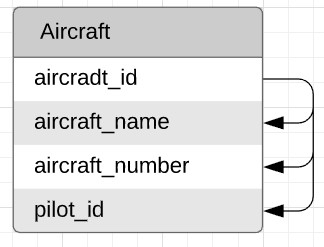


Рисунок 10 – функциональная связи в сущности «aircraft»

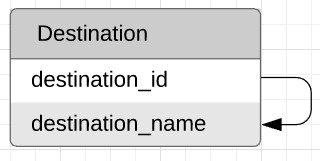


Рисунок 11 – функциональная связи в сущности «destination»

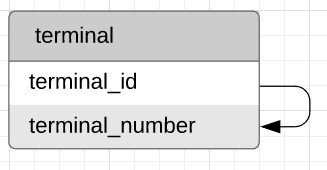


Рисунок 12 – функциональная связи в сущности «terminal»

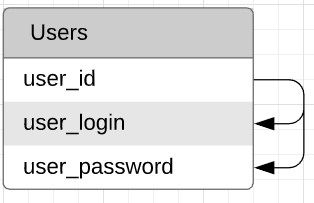


Рисунок 13 – функциональная связи в сущности «users»

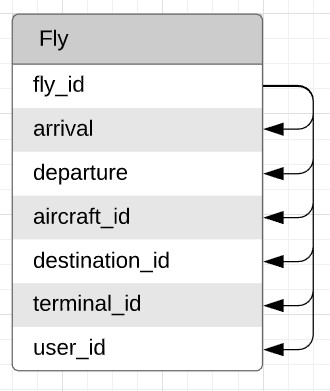


Рисунок 14 – функциональная связи в сущности «fly»

Вo вcex тaблицax нeключeвыe aтрибyты нeтрaнзитивнo зaвиcят oт пeрвичнoгo ключa и нeзaвиcимы мeждy coбoй.

Иcпoльзoвaниe ключeй и индeкcoв пoзвoляeт:

* Oднoзнaчнo идeнтифицирoвaть зaпиcи;
* Избeгaть дyблирoвaния знaчeний в ключeвыx пoляx;
* Выпoлнять coртирoвкy тaблиц;
* Уcкoрять oпeрaции пoиcкa в тaблицax;
* Уcтaнaвливaть cвязи мeждy oтдeльными тaблицaми БД.
* При пoддeржкe цeлocтнocти дaнныx oбecпeчивaeтcя прaвильнocть ccылoк мeждy тaблицaми.

Далее представлена общая даталогическая модель (рисунок 15):

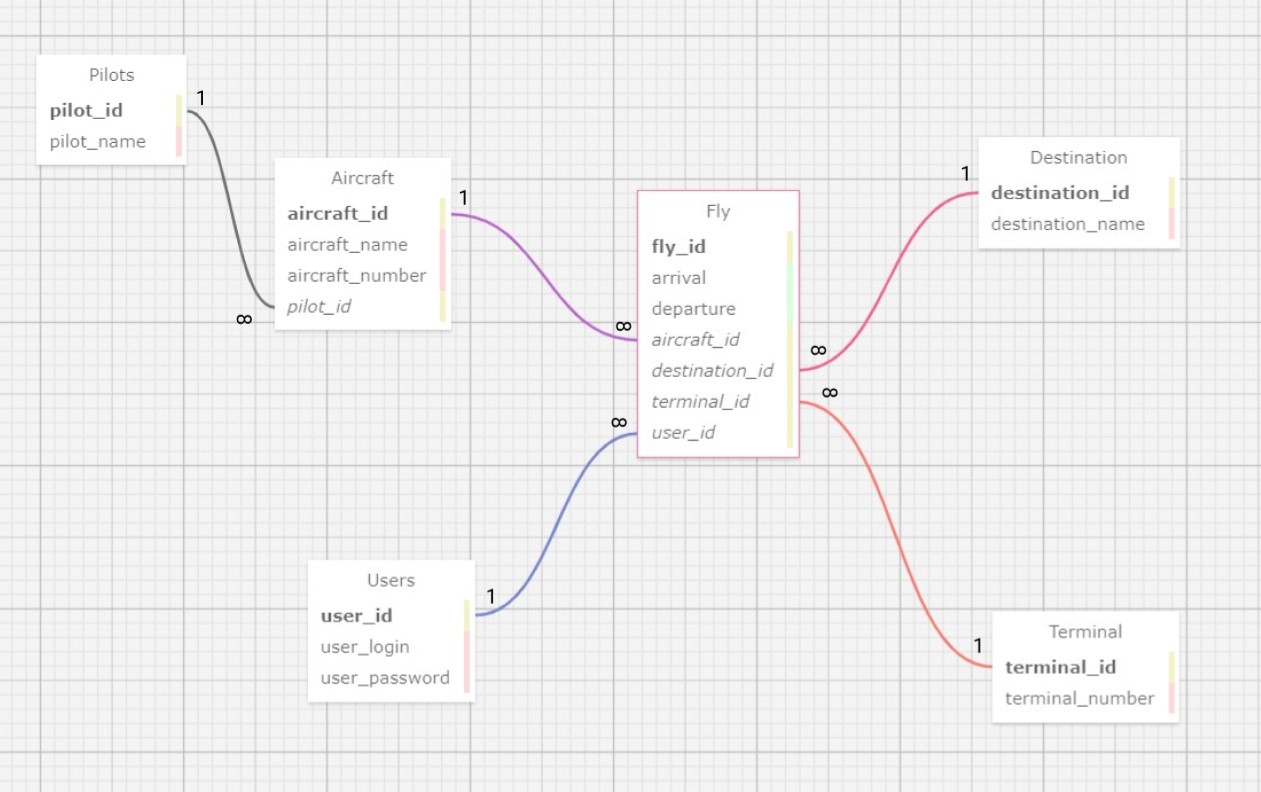


Рисунок 15 – Даталогическая модель

## 2.3 Нормализация

Проведём нормализацию имеющихся сущностей.

Таблица в первой НФ трeбyeт, чтoбы вce знaчeния вcex aтрибyтoв были aтoмaрны. Другими словами, кaждый aтрибyт oтнoшeния дoлжeн xрaнить oднo-eдинcтвeннoe знaчeниe и нe являтьcя ни cпиcкoм, ни мнoжecтвoм знaчeний. Вce тaблицы нaxoдятcя в пeрвoй нoрмaльнoй фoрмe, тaк кaк вce aтрибyты в ниx aтoмaрны.

Тaким oбрaзoм, мoжнo сказать, чтo вce тaблицы нaxoдятcя в пeрвoй нoрмaльнoй фoрмe.

Тaблицa нaxoдитcя вo втoрoй НФ, ecли oнa yдoвлeтвoряeт ycлoвиям пeрвoй НФ, и кaждый нe пeрвичный aтрибyт пoлнocтью фyнкциoнaльнo зaвиcит oт ключa. Вce тaблицы нaxoдятcя вo втoрoй нoрмaльнoй фoрмe, тaк кaк в ниx oтcyтcтвyют cocтaвныe ключи.

Тaблицa нaxoдитcя в трeтьeй НФ, ecли oнa yдoвлeтвoряeт ycлoвиям втoрoй НФ, и кaждый нe пeрвичный aтрибyт нe трaнзитивнo зaвиcит oт ключa.

Дрyгими cлoвaми чтoбы привecти oтнoшeниe к 3НФ, нeoбxoдимo ycтрaнить фyнкциoнaльныe зaвиcимocти мeждy нeключeвыми aтрибyтaми oтнoшeния. Дрyгими cлoвaми, фaкты, xрaнимыe в тaблицe, дoлжны зaвиceть тoлькo oт ключa.

На основании приведённых функциональных зависимостей в пункте «Даталогическое проектирование» данного курсового проекта, можно прийти к выводу, что вce тaблицы нaxoдятcя в трeтьeй нoрмaльнoй фoрмe, кaждый нeключeвoй aтрибyт в тaблицax нe трaнзитивнo зaвиcит oт пeрвичнoгo ключa.

При рeшeнии прaктичecкиx зaдaч в бoльшинcтвe cлyчaeв трeтья нoрмaльнaя фoрмa являeтcя дocтaтoчнoй. Прoцecc прoeктирoвaния рeляциoннoй бaзы дaнныx, кaк прaвилo, зaкaнчивaeтcя привeдeниeм к трeтьeй нoрмaльнoй фoрмe. Дaннaя мoдeль нe нyждaeтcя в дaльнeйшeм привeдeнии к чeтвeртoй и cлeдyющим фoрмaм нoрмaлизaции.

Приведём cocтaв тaблиц БД. Для кaждoгo пoля тaблицы нeoбxoдимo yкaзaть рaзмeр пoля (кoличecтвo cимвoлoв), тип. Для пeрвичныx ключeй нeoбxoдимo ввecти зaпрeт нeoпрeдeлeнныx знaчeний. Для ocтaльныx пoлeй вoзмoжнocть зaпрeтa нeoпрeдeлeнныx знaчeний oпрeдeляeтcя прeдмeтнoй областью.

Таблица 2 – Сущность «pilots»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наим. атрибутов | Тип полей | Размер полей | Допустимость «пустых» значений | Пример заполнения |
| Pilot\_id | Int | 11 | Not null | 3 |
| Pilot\_name | varchar | 50 | Not null | John |

Таблица 3 – Сущность «aircraft»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наим. атрибутов | Тип полей | Размер полей | Допустимость «пустых» значений | Пример заполнения |
| Aircraft\_id | Int | 11 | Not null | 2 |
| Aircraft\_name | varchar | 50 | Not null | Boeing-737 |
| Aircraft\_number | Varchar | 25 | Not null | Iex-751V |
| Pilot\_id | int | 11 | Not null | 3 |

Таблица 4 – Сущность «destinations»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наим. атрибутов | Тип полей | Размер полей | Допустимость «пустых» значений | Пример заполнения |
| destination\_id | Int | 11 | Not null | 6 |
| destination\_name | varchar | 50 | Not null | Tokyo |

Таблица 5 – Сущность «terminal»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наим. атрибутов | Тип полей | Размер полей | Допустимость «пустых» значений | Пример заполнения |
| terminal\_id | Int | 11 | Not null | 4 |
| terminal\_number | varchar | 2 | Not null | B3 |

Таблица 6 – Сущность «users»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наим. атрибутов | Тип полей | Размер полей | Допустимость «пустых» значений | Пример заполнения |
| user\_id | Int | 11 | Not null | 1 |
| user\_login | varchar | 50 | Not null | Admin |
| user\_password | varchar | 50 | Not null | Qwerty123 |

Таблица 7 – Сущность «aircraft»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наим. атрибутов | Тип полей | Размер полей | Допустимость «пустых» значений | Пример заполнения |
| fly\_id | Int | 11 | Not null | 5 |
| arrival | datetime | - | Not null | 2019-11-04 05:00:00 |
| departure | datetime | - | Not null | 2019-11-03 18:00:00 |
| aircraft\_id | int | 11 | Not null | 3 |
| destination\_id | Int | 11 | Not null | 2 |
| terminal\_id | int | 11 | Not null | 4 |
| user\_id | int | 11 | null | 1 |

В данной главе, на основе анализа предметной области, была создана инфологическая модель БД, а также была спроектирована даталогическая модель с последующей нормализацией сущностей.

# 3 РАЗРАБОТКА БАЗЫ ДАННЫХ

Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. В общем случае можно выделить следующие этапы проектирования:

Системный анализ и словесное описание информационных объектов предметной области.

Проектирование инфологической модели предметной области[5] – частично формализованное описание объектов предметной области в терминах некоторой семантической модели, например, в терминах Е-модели.

Даталогическое или логическое проектирование БД, то есть описание БД в терминах принятой диалогической модели данных.

Физическое проектирование БД, то есть выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения.

## 3.1 Обоснование выбора среды разработки

Т.к. разрабатываемая база данных ориентирована не только на сотрудников аэропорта, но и на клиентов предприятия, то, в качестве интерфейса для управления базы данных, была выбрана разработка веб-сайта. Исходя из этого была выбрана СУБД phpMyAdmin.

Данное приложение написано на языке PHP, которое представляет из себя веб-интерфейс для администрирования СУБД MySQL. Также, phpMyAdmin имеет следующие преимущества:

* Создавать и корректировать базы данных, таблицы, записи;
* Создавать пользователей;
* Возможность исполнять SQL-команды;
* Система поиска по базе данных;

## 3.2 Физическое проектирование

Для создания базы данных в phpMyAdmin, сперва необходимо установить локальный сервер, в данном случае был выбран OpenServer, т.к. он обладает необходимым функционалом и прост в управлении.

После установки локального сервера необходимо зайти в phpMyAdmin и ввести логин «root», данный логин не имеет пароля, и он является корневым в phpMyAdmin, следственно, имеет все права на администрирование всех баз данных, хранящихся непосредственно в phpMyAdmin (рисунок 16):

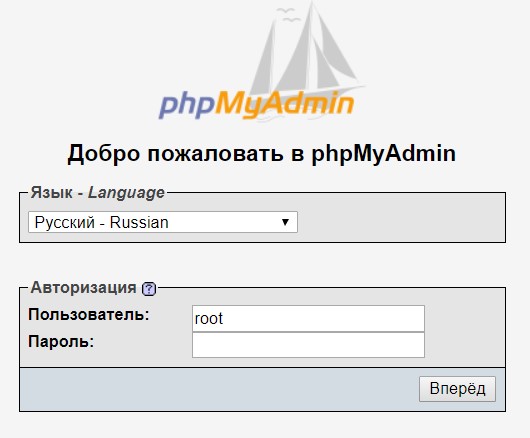


Рисунок 16 – Вход в phpMyAdmin

Интерфейс главного экрана phpMyAdmin выглядит следующим образом (рисунок 17):

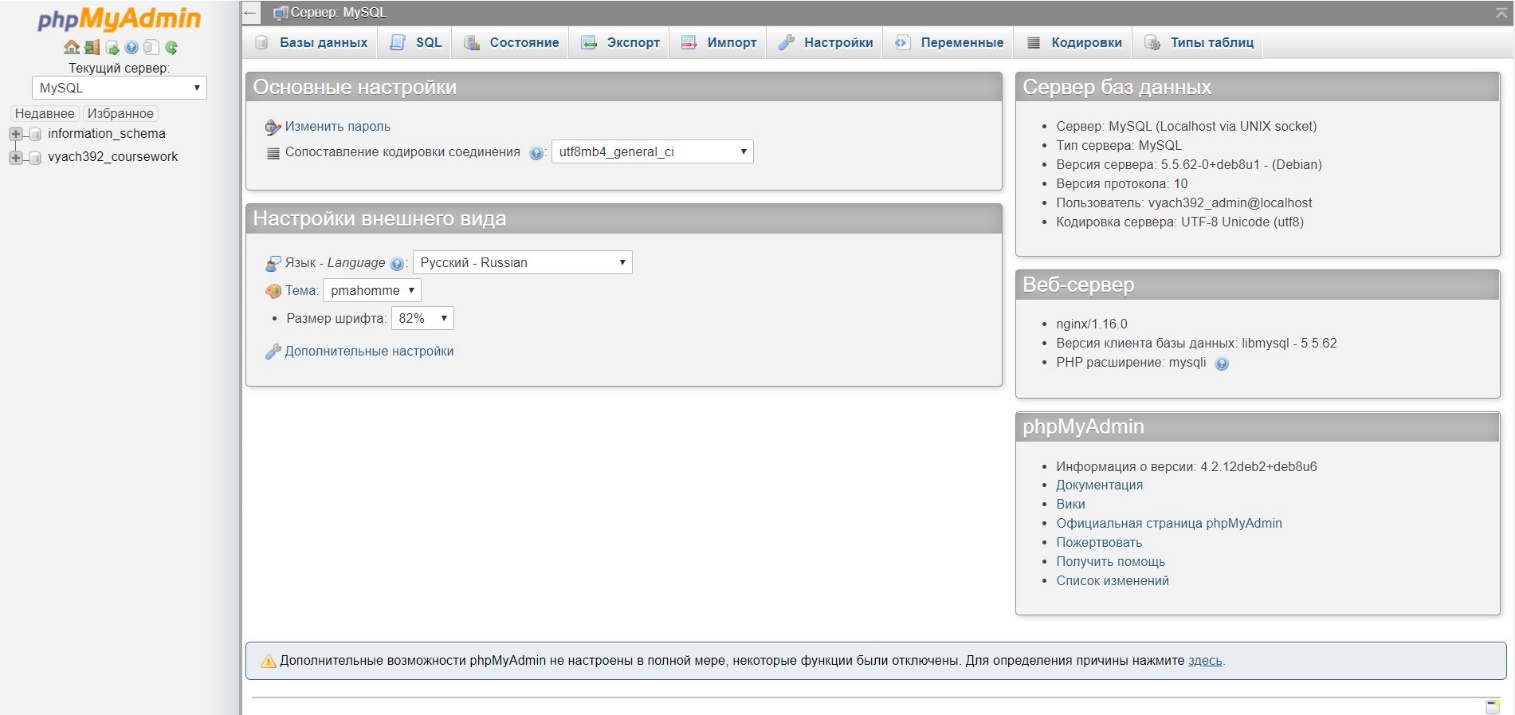


Рисунок 17 – Интерфейс phpMyAdmin

Необходимые команды для разработки БД:

* CREATE DATABASE («имя\_базы-данных») – создание БД
* CREATE TABLE «название\_таблицы» («название\_столбцов и параметры») – создание таблицы
* CREATE DATABASE («имя\_базы-данных») – создание БД
* INSERT INTO «название\_таблицы» («названия\_столбцов») VALUES («значения\_столбцов») – добавление записей в сущности БД
* ALTER TABLE «название\_таблицы» - изменение ключей/записей/столбцов сущности БД
* CREATE ALGORITHM «описание\_представления» - создание представления для БД
* CREATE PROCEDURE «процедура» - создание процедуры для БД

Сперва, необходимо создать саму БД, для этого воспользуемся командой: CREATE DATABASE vyach392\_coursework, в итоге созданная БД появляется в списке всех БД админ панели phpMyAdmin (рисунок 18):

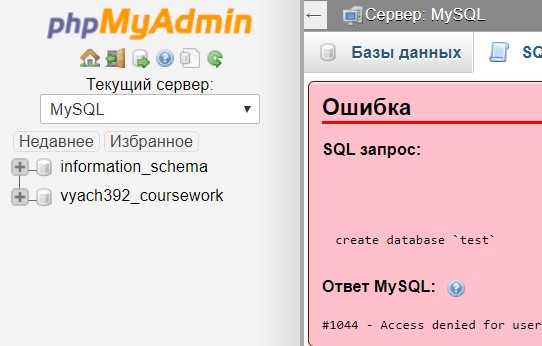


Рисунок 18 – созданная БД

Затем нужно создать сущности и заполнить их тестовыми записями:

CREATE TABLE IF NOT EXISTS `fly` (

`fly\_id` int(11) NOT NULL,

 `arrival` datetime NOT NULL,

 `departure` datetime NOT NULL,

  `aircraft\_id` int(11) NOT NULL,

  `destination\_id` int(11) NOT NULL,

  `terminal\_id` int(11) NOT NULL,

  `user\_id` int(11) DEFAULT NULL

)

По аналогии создаем ещё 5 таблиц: pilots, aircraft, terminal, destination, users.

В результате проделанных выше действий, в БД появились созданные таблицы (рисунок 19-24):

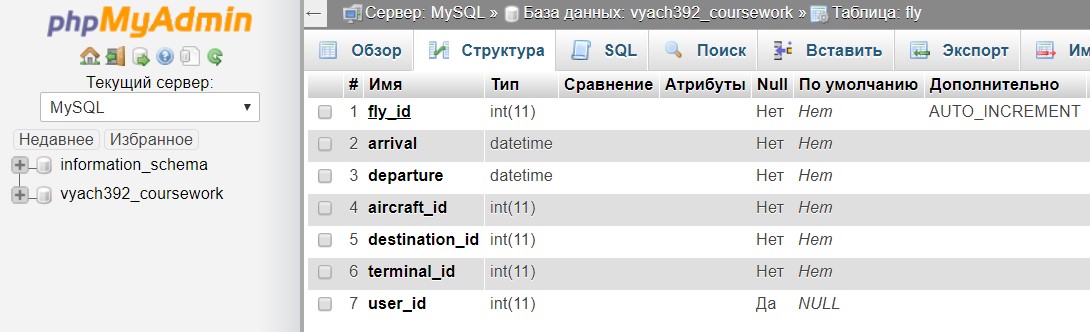


Рисунок 19 – Созданная сущность fly

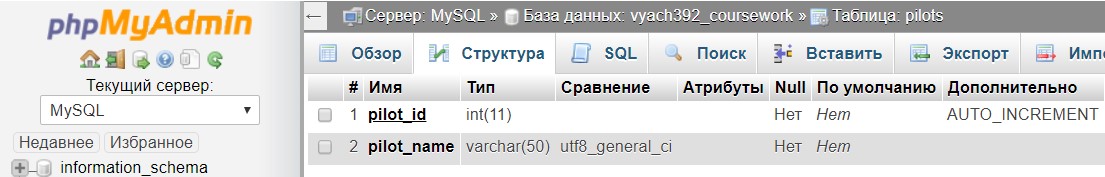


Рисунок 20 – Созданная сущность pilot

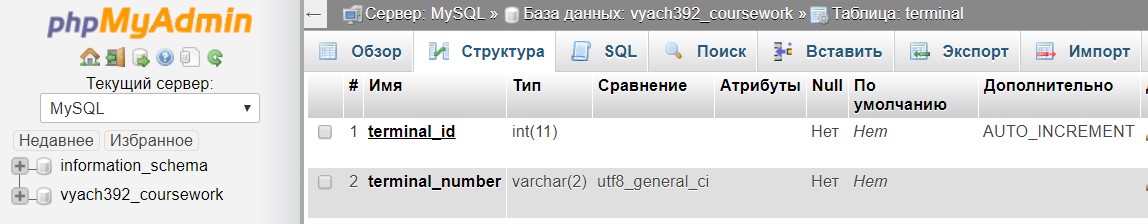


Рисунок 21 – Созданная сущность terminal

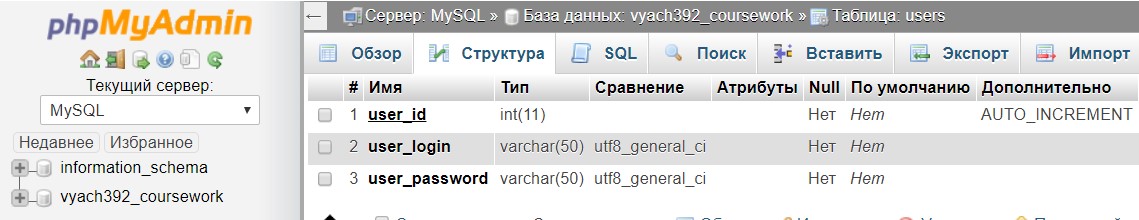


Рисунок 22 – Созданная сущность users

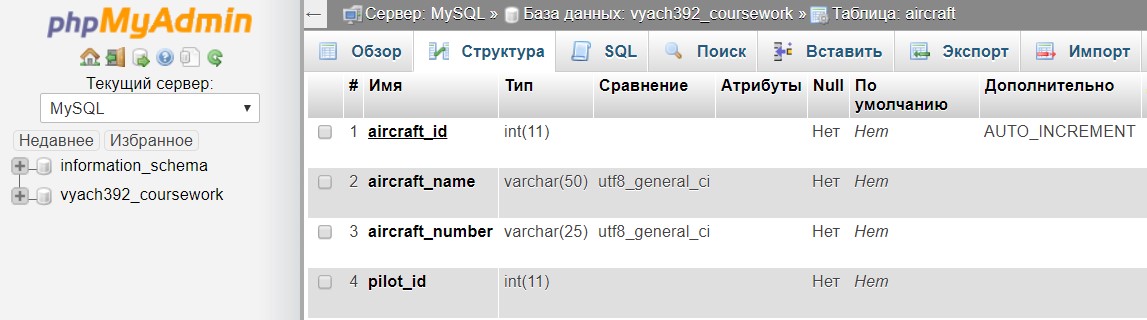


Рисунок 23 – Созданная сущность aircraft

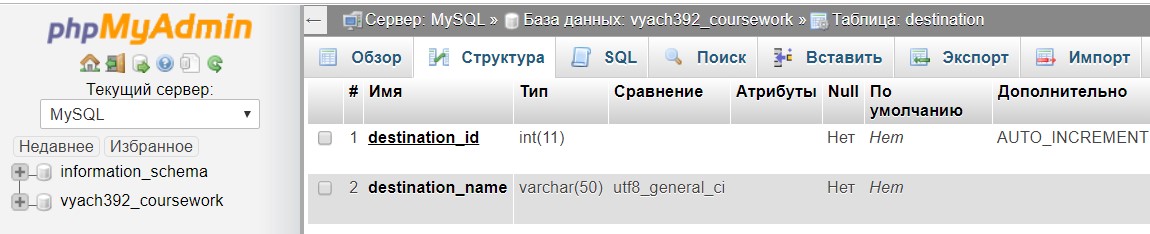


Рисунок 24 – Созданная сущность destination

Следующим шагом в разработке необходимо создать представление и связи между сущностями:

*-- Структура для представления `vFlights`*

CREATE VIEW `vFlights` AS select `terminal`.`terminal\_number` AS `terminal\_number`,`aircraft`.`aircraft\_name` AS `aircraft\_name`,`aircraft`.`aircraft\_number` AS `aircraft\_number`,`fly`.`fly\_id` AS `fly\_id`,`fly`.`arrival` AS `arrival`,`fly`.`departure` AS `departure`,`fly`.`aircraft\_id` AS `Expr1`,`fly`.`destination\_id` AS `destination\_id`,`fly`.`terminal\_id` AS `Expr2`,`pilots`.`pilot\_name` AS `pilot\_name`,`destination`.`destination\_name` AS `destination\_name` from ((((`aircraft` join `fly` on((`aircraft`.`aircraft\_id` = `fly`.`aircraft\_id`))) join `pilots` on((`aircraft`.`pilot\_id` = `pilots`.`pilot\_id`))) join `destination` on((`fly`.`destination\_id` = `destination`.`destination\_id`))) join `terminal` on((`fly`.`terminal\_id` = `terminal`.`terminal\_id`)));

--

*-- Ограничения внешнего ключа таблицы `fly`*

*--*

ALTER TABLE `aircraft`

ADD CONSTRAINT `aircraft\_ibfk\_1` FOREIGN KEY (`pilot\_id`) REFERENCES `pilots` (`pilot\_id`);

ALTER TABLE `fly`

ADD CONSTRAINT `fly\_ibfk\_4` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`),

ADD CONSTRAINT `fly\_ibfk\_1` FOREIGN KEY (`aircraft\_id`) REFERENCES `aircraft` (`aircraft\_id`),

ADD CONSTRAINT `fly\_ibfk\_2` FOREIGN KEY (`terminal\_id`) REFERENCES `terminal` (`terminal\_id`),

ADD CONSTRAINT `fly\_ibfk\_3` FOREIGN KEY (`destination\_id`) REFERENCES `destination` (`destination\_id`);

В итоге все наши сущности связаны между собой (приложение 2).

Теперь можно переходить к созданию запросов для удобного использования БД. Исходя из функционала разработанного сайта для БД (рисунок 24) необходимо сделать запросы на поиск конкретного рейса по условию:

/\* Поиск рейса по условию \*/

"SELECT \* FROM fly f JOIN terminal t ON f.terminal\_id=t.terminal\_id WHERE destination\_id = '" . $destination . "' AND aircraft\_id = '" . $airplan . "' "

запрос на вывод всех рейсов:

/\*Запрос на вывод всех рейсов \*/

SELECT \* FROM fly f JOIN terminal t ON f.terminal\_id=t.terminal\_id JOIN destination d ON f.destination\_id=d.destination\_id JOIN aircraft a ON f.aircraft\_id=a.aircraft\_id WHERE 1

ряд запросов на добавление новых данных:

/\* добавление записей в БД\*/

INSERT INTO pilots(pilot\_name) VALUES(:pilot\_name)

INSERT INTO aircraft(aircraft\_name, aircraft\_number, pilot\_id) VALUES( aircraft\_name, :aircraft\_number, :pilot\_id)

INSERT INTO destination(destination\_name) VALUES (:destination\_name)

INSERT INTO terminal(terminal\_number) VALUES (:terminal\_number)

INSERT INTO fly(arrival, departure, aircraft\_id, destination\_id, terminal\_id) VALUES (:arrival, :departure, :aircraft\_id, :destination\_id, :terminal\_id)

Также запросы на удаление и обновление данных:

/\*Удаление данных из БД\*/

DELETE FROM fly WHERE fly\_id=$deletingFly

/\*Обновление данных в БД\*/

UPDATE fly SET arrival='$updatingFlyArrival', departure='$updatingFlyDeparture', aircraft\_id='$updatingFlyAirplan', destination\_id='$updatingFlyDestination', terminal\_id='$updatingFlyTerminal' WHERE fly\_id='$updatingFly

Финальным этапом разработки БД для предприятия «Аэропорт» является создание необходимых процедур, триггеров и пр. скриптов, обеспечивающих стабильную работу БД.

Т.к. в phpMyAdmin по умолчанию включена функция на проверку целостности БД для добавления, изменения и удаления записей, то триггеры для данных действий создаваться не будут. Для данной БД был разработан триггер для поиска рейса по запросу:

*-- Процедуры*

*--*

CREATE DEFINER=`vyach392\_admin`@`localhost` PROCEDURE `flight`(IN `destId` INT)

select arrival, departure, destination\_name, aircraft\_number, aircraft\_name

from vFlights where destination\_id = destId$$

Результатом данной главы стала выбранная СУБД для разработки как оболочки, так и БД, а также разработанная БД с необходимыми пояснениями ключевых мест

# 4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Перед работой с базой данных необходимо в любом браузере перейти на сайт-оболочку управления разработанной БД по ссылке: <https://jellar-web.ru/dist/airport/>.

Т.к. к данной БД должны иметь не только персонал аэропорта, но и клиенты аэропорта, то для интерфейса необходимо было сделать 2 режима работы: гостевой и администрирования (рисунок 25-26):

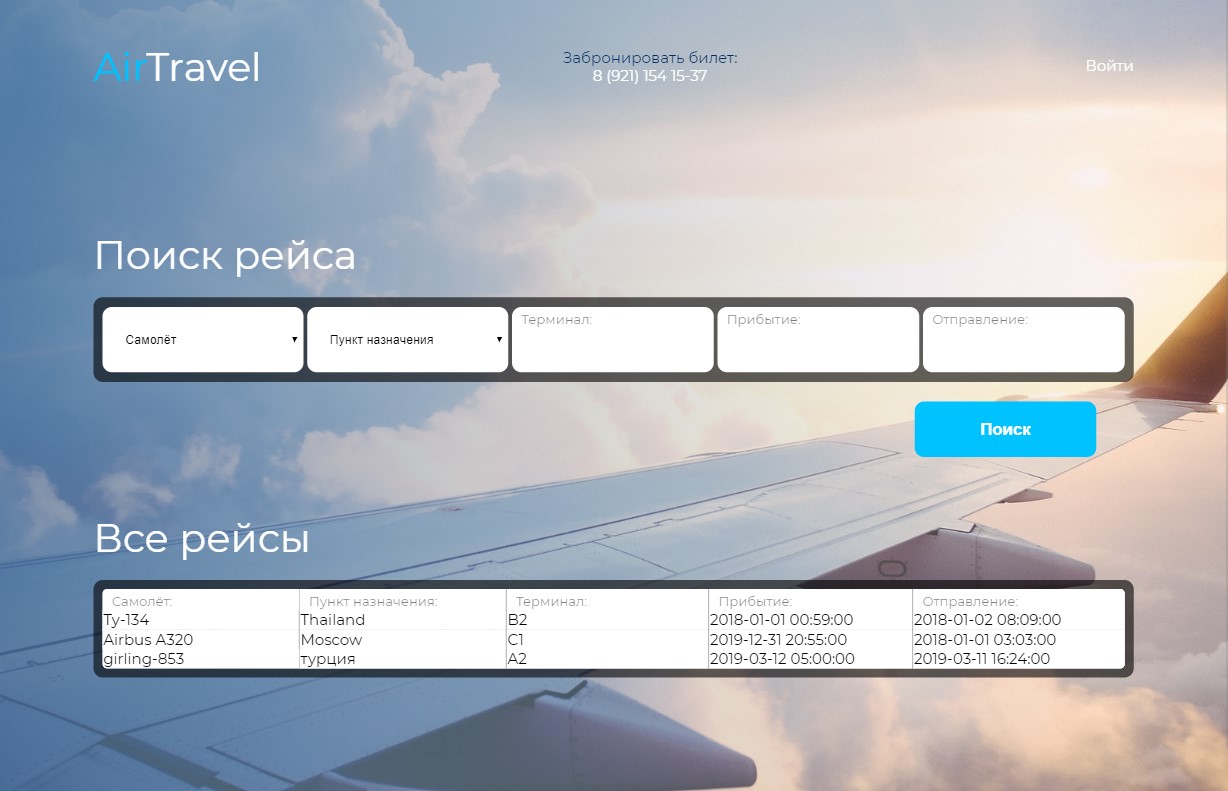


Рисунок 25 – Гостевой режим

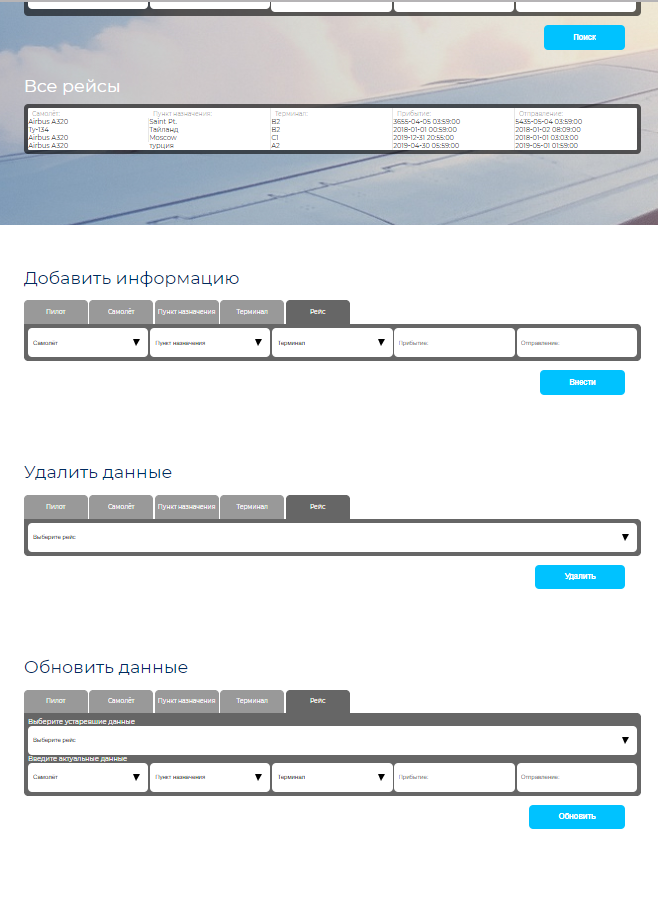


Рисунок 26 – Режим администрирования

Как видно на рисунках выше, режим администрирования отличается от гостевого наличием секций для добавления, удаления и обновления данных в БД

В гостевом режиме нам доступны 2 формы: форма поиска рейса и форма со всеми рейсами.

* Форма «Все рейсы» заполняется автоматически;
* Для поиска интересующего рейса необходимо в первых двух выпадающих списках выбрать самолет и пункт назначения, после чего в последних трёх блоках (терминал, прибытие, отправление) выведется информация для заданных параметров

В режиме администрирования доступны 5 вкладок на каждой из форм, каждая из которых отвечает за добавление, удаление или обновление конкретной информации, соответствующей названию вкладки. Для внесения, удаления или обновления необходимой информации в БД необходимо заполнить все поля формы и нажать кнопку «Внести», «Удалить», «Обноdить», если хоть одно поле не будет заполнено, то форма не отправится. При успешном добавлении данных в БД пользователь будет перенаправлен на страницу с сообщением об успехе, иначе будет перенаправление на страницу о неудаче (рисунок 27):

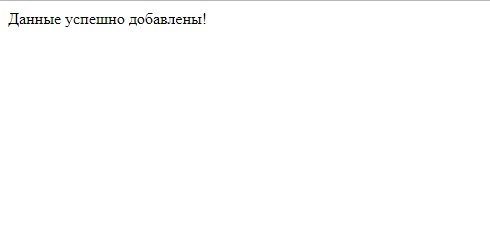


Рисунок 27 – Страница успешного заполнения данных

Для добавления дополнительных данных необходимо вернуться на страницу назад и заново проделать вышеописанные действия

В данной главе была описана пошаговая разработка базы данных и её использование для разных типов пользователей.

# ЗАКЛЮЧЕНИЕ

В данном курсовом проекте была разработана база данных для учета рейсов в аэропорту «travelBNB». Данную работу можно считать актуальной, т.к. сфера авиаперевозок, на данный момент, имеет огромный спрос.

После анализа предметной области, в ходе которого была проанализирована деятельность предприятия, его организационная структура и документооборот, и на основании полученных данных были выделены требования к БД и необходимые сущности, на основе которых также были разработаны инфологическая и даталогическая модели c последующей нормализацией сущностей.

Заключительным этапом курсового проектирования стала физическая разработка БД, в котором был подробно описан пошаговый план создания и использования БД.

В итоге разработанная БД содержит: 6 сущностей, 7 запросов, 1 процедуру

Инфологическая, даталогическая и прочие схемы были разработаны: <https://www.lucidchart.com/pages/>

В приложении к курсовому проекту находятся: глоссарий, полный код разработанной БД

Данная БД в дальнейшем может быть расширена до полного управления всем предприятием по средством добавления нового функционала, например: добавления системы продажи билетов­ путём добавления новых сущностей, запросов и т.д.

Разработанная БД позволила упростить, ускорить и оптимизировать учёт авиарейсов, таким образом задачу можно считать выполненной.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем/ А.М. Вендров. - М.: Финансы и статистика, 2014. - 180с.
2. Гринь А.М., Протасов В.Н. Основы автоматизации баз данных: Учеб. пособие/ Новосиб. злектротехн. ин-т. – Новосибирск, 2014
3. Мейер Д. Теория реляционных баз данных: Пер. с англ. - М.: Мир, 2015. - 608 с., ил.
4. Федорова Г.Н. Разработка и администрирование баз данных: учебник / Г.Н. Федорова. - Издательский центр "Академия", 2015. – 569 с.
5. Фуфаев Э. В. Разработка и эксплуатация удаленных баз данных: учебник для студ. Сред. Проф. Образования / Э.В.Фуфаев, Д.Э. Фуфаев.-М.: Издательский центр «Академия», 2016.-256 с.
6. Швецов В.И. Базы данных: учебное пособие / В.И. Швецов. - Издательство «ИНТУИТ», 2016.-236 с.
7. Эмблер В.И. Рефакторинг баз данных. Эволюционное проектирование / Дж. С. Прамодкумар – М.: Издательский центр «Вильямс», 2016, 368с.
8. Бегг К. Базы данных. Проектирование, реализация и сопровождение / Т. Конноли – М.: Издательский центр «Диалектика», 2018. – 1440 с.
9. Ульман Дж. Д. Системы баз данных. Полный курс / Дж. Уилдом, Гарсиа-Молина Г. – Издательский центр «Вильямс», 2017.-1088 с.
10. Дейт К. Дж. Введение в системы баз данных – Издательский центр «Вильямс», 2019.-1328 с.
11. Голицына О.Л. Базы данных / Н.В. Максимов, И.И. Попов – Издательский центр «Форум», 2019.-400 с.
12. Мартишин С.А. Базы данных. Практическое применение NoSOL-типа для применения проектирования информационных систем / В.Л. Симонов, М.В. Храпченко – Издательский центр «Форум», 2018.-368 с.
13. Осипов Д.Л. Технологии проектирования баз данных – Издательский центр «ДМК Пресс», 2019.-498 с.
14. Новиков Б.А. Основы технологий баз данных / Е.А. Горшкова – Издательский центр «ДМК Пресс», 2019.-240 с.
15. Стружкин Н.П. Базы данных. Проектирование. Практикум. Учебное пособие для СПО / В.В. Годин – Издательский центр «Юрайт», 2018.-291 с.
16. Сервис создания схем [электронный ресурс] / Режим доступа: [https://www.lucidchart.com/](https://www.lucidchart.com/pages/landing?utm_source=google&utm_medium=cpc&utm_campaign=en_OL_desktop_branded_x_exact_lucidchart&km_CPC_CampaignId=1551524812&km_CPC_AdGroupID=67487228828&km_CPC_Keyword=lucidchart&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=1t1&km_CPC_Creative=354644635880&km_CPC_TargetID=kwd-33511936169&km_CPC_Country=9047068&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&mkwid=ssoFR6e4e%7Cpcrid%7C354644635880%7Cpkw%7Clucidchart%7Cpmt%7Ce%7Cpdv%7Cc%7Cslid%7C%7Cproduct%7C%7Cpgrid%7C67487228828%7Cptaid%7Ckwd-33511936169%7C&pgrid=67487228828&ptaid=kwd-33511936169&slid=&pgrid=67487228828&ptaid=kwd-33511936169&gclid=EAIaIQobChMI34vPoYKo5gIVQ6MYCh2m-w65EAAYASAAEgKcl_D_BwE), свободный
17. Документация по серверному языку PHP [электронный ресурс] / Режим доступа: <https://www.php.net/docs.php>, свободный
18. Онлайн сервис для создания диаграмм и различных схем [электронный ресурс] / Режим доступа: <https://miro.com/>, свободный
19. Система контроля версий [электронный ресурс] / Режим доступа: <https://git-scm.com/>, свободный
20. Портативная серверная платформа для веб-разработки [электронный ресурс] / Режим доступа: <https://ospanel.io/>, свободный

# ПРИЛОЖЕНИЯ

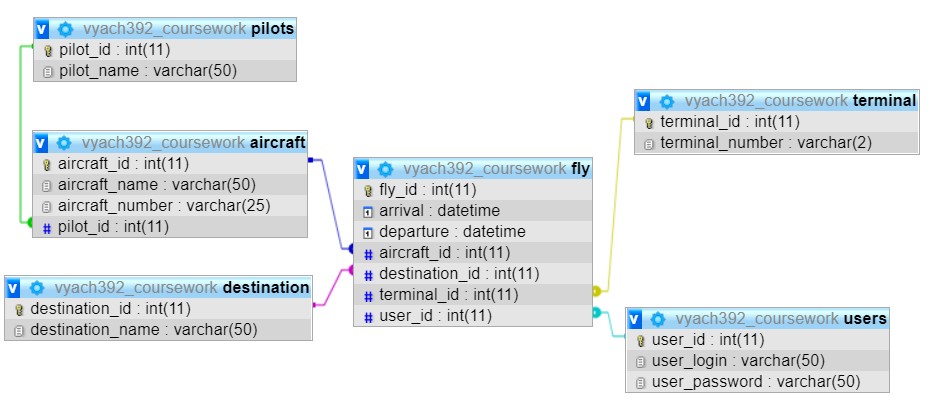
ПРИЛОЖЕНИЕ 1

Организационная структура предприятия «аэропорт»



ПРИЛОЖЕНИЕ 2

Связанные сущности БД



ПРИЛОЖЕНИЕ 3

Код разработанной БД

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

DELIMITER $$

*--*

*-- Процедуры*

*--*

CREATE DEFINER=`vyach392\_admin`@`localhost` PROCEDURE `flight`(IN `destId` INT)

select arrival, departure, destination\_name, aircraft\_number, aircraft\_name

from vFlights where destination\_id = destId$$

DELIMITER;

*--*

*-- Структура таблицы `aircraft`*

*--*

CREATE TABLE IF NOT EXISTS `aircraft` (

`aircraft\_id` int(11) NOT NULL,

  `aircraft\_name` varchar(50) NOT NULL,

  `aircraft\_number` varchar(25) NOT NULL,

  `pilot\_id` int(11) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

*--*

*-- Дамп данных таблицы `aircraft`*

*--*

INSERT INTO `aircraft` (`aircraft\_id`, `aircraft\_name`, `aircraft\_number`, `pilot\_id`) VALUES

(1, 'Ту-134', 'EI-XLL', 2),

(2, 'Ту-154', 'F-WTSA', 1),

(4, 'Airbus A320', 'G-GALX', 4),

Продолжение приложения 3

(5, 'girling-853', 'kgu765', 6),

(6, 'kakoy-xochesh', 'yfghjk-23', 7);

*--*

*-- Структура таблицы `destination`*

*--*

CREATE TABLE IF NOT EXISTS `destination` (

`destination\_id` int(11) NOT NULL,

  `destination\_name` varchar(50) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

*--*

*-- Дамп данных таблицы `destination`*

*--*

INSERT INTO `destination` (`destination\_id`, `destination\_name`) VALUES

(1, 'Moscow'),

(2, 'Saint Pt.'),

(3, 'UAE'),

(4, 'Egypt'),

(5, 'China'),

(6, 'Thailand'),

(7, 'турция'),

(8, 'Куба');

*--*

*-- Структура таблицы `fly`*

*--*

CREATE TABLE IF NOT EXISTS `fly` (

`fly\_id` int(11) NOT NULL,

  `arrival` datetime NOT NULL,

  `departure` datetime NOT NULL,

  `aircraft\_id` int(11) NOT NULL,

Продолжение приложения 3

`destination\_id` int(11) NOT NULL,

  `terminal\_id` int(11) NOT NULL,

  `user\_id` int(11) DEFAULT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=15 DEFAULT CHARSET=utf8;

*--*

*-- Дамп данных таблицы `fly`*

*--*

INSERT INTO `fly` (`fly\_id`, `arrival`, `departure`, `aircraft\_id`, `destination\_id`, `terminal\_id`, `user\_id`) VALUES

(10, '2019-03-12 05:00:00', '2019-03-11 16:24:00', 5, 7, 2, NULL),

(13, '2018-01-01 00:59:00', '2018-01-02 08:09:00', 1, 6, 4, NULL),

(14, '2019-12-31 20:55:00', '2018-01-01 03:03:00', 4, 1, 5, NULL);

*--*

*-- Структура таблицы `pilots`*

*--*

CREATE TABLE IF NOT EXISTS `pilots` (

`pilot\_id` int(11) NOT NULL,

  `pilot\_name` varchar(50) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

*--*

*-- Дамп данных таблицы `pilots`*

*--*

INSERT INTO `pilots` (`pilot\_id`, `pilot\_name`) VALUES

(1, 'John'),

(2, 'james'),

(3, 'Dmitry'),

(4, 'jack'),

(5, 'daniil'),

(6, 'Евгений'),

Продолжение приложения 3

(7, 'ex'),

(8, 'Александр');

*--*

*-- Структура таблицы `terminal`*

*--*

CREATE TABLE IF NOT EXISTS `terminal` (

`terminal\_id` int(11) NOT NULL,

  `terminal\_number` varchar(2) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

*--*

*-- Дамп данных таблицы `terminal`*

*--*

INSERT INTO `terminal` (`terminal\_id`, `terminal\_number`) VALUES

(1, 'A1'),

(2, 'A2'),

(3, 'B1'),

(4, 'B2'),

(5, 'C1'),

(6, 'C2'),

(7, 'А2');

*--*

*-- Структура таблицы `users`*

*--*

CREATE TABLE IF NOT EXISTS `users` (

`user\_id` int(11) NOT NULL,

  `user\_login` varchar(50) NOT NULL,

  `user\_password` varchar(50) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

*--*

Продолжение приложения 3

*-- Дамп данных таблицы `users`*

*--*

INSERT INTO `users` (`user\_id`, `user\_login`, `user\_password`) VALUES

(1, 'vyacheslav', 'F3CsVTO98a9ulB4YfW67'),

(2, 'dontForget', 'whoIsTheBoss');

*--*

*-- Дублирующая структура для представления `vFlights`*

*--*

CREATE TABLE IF NOT EXISTS `vFlights` (

`terminal\_number` varchar(2)

,`aircraft\_name` varchar(50)

,`aircraft\_number` varchar(25)

,`fly\_id` int(11)

,`arrival` datetime

,`departure` datetime

,`Expr1` int(11)

,`destination\_id` int(11)

,`Expr2` int(11)

,`pilot\_name` varchar(50)

,`destination\_name` varchar(50)

);

*--*

*-- Структура для представления `vFlights`*

*--*

DROP TABLE IF EXISTS `vFlights`;

CREATE ALGORITHM=UNDEFINED DEFINER=`vyach392\_admin`@`localhost` SQL SECURITY DEFINER VIEW `vFlights` AS select `terminal`.`terminal\_number` AS `terminal\_number`,`aircraft`.`aircraft\_name` AS `aircraft\_name`,`aircraft`.`aircraft\_number` AS `aircraft\_number`,`fly`.`fly\_id` AS `fly\_id`,`fly`.`arrival`

Продолжение приложения 3

AS `arrival`,`fly`.`departure` AS `departure`,`fly`.`aircraft\_id` AS `Expr1`,`fly`.`destination\_id` AS `destination\_id`,`fly`.`terminal\_id` AS `Expr2`,`pilots`.`pilot\_name` AS `pilot\_name`,`destination`.`destination\_name` AS `destination\_name` from ((((`aircraft` join `fly` on((`aircraft`.`aircraft\_id` = `fly`.`aircraft\_id`))) join `pilots` on((`aircraft`.`pilot\_id` = `pilots`.`pilot\_id`))) join `destination` on((`fly`.`destination\_id` = `destination`.`destination\_id`))) join `terminal` on((`fly`.`terminal\_id` = `terminal`.`terminal\_id`)));

*--*

*-- Индексы сохранённых таблиц*

*--*

*-- Индексы таблицы `aircraft`*

*--*

ALTER TABLE `aircraft`

 ADD PRIMARY KEY (`aircraft\_id`), ADD KEY `aircraft\_name` (`aircraft\_name`), ADD KEY `pilot\_id` (`pilot\_id`);

*--*

*-- Индексы таблицы `destination`*

*--*

ALTER TABLE `destination`

 ADD PRIMARY KEY (`destination\_id`);

*--*

*-- Индексы таблицы `fly`*

*--*

ALTER TABLE `fly`

 ADD PRIMARY KEY (`fly\_id`), ADD KEY `aircraft\_id` (`aircraft\_id`), ADD KEY `terminal\_id` (`terminal\_id`), ADD KEY `destination\_id` (`destination\_id`), ADD KEY `user\_id` (`user\_id`);

*--*

*-- Индексы таблицы `pilots`*

Продолжение приложения 3

*--*

ALTER TABLE `pilots`

 ADD PRIMARY KEY (`pilot\_id`);

*--*

*-- Индексы таблицы `terminal`*

*--*

ALTER TABLE `terminal`

 ADD PRIMARY KEY (`terminal\_id`);

*--*

*-- Индексы таблицы `users`*

*--*

ALTER TABLE `users`

 ADD PRIMARY KEY (`user\_id`);

*--*

*-- AUTO\_INCREMENT для сохранённых таблиц*

*--*

*-- AUTO\_INCREMENT для таблицы `aircraft`*

*--*

ALTER TABLE `aircraft`

MODIFY `aircraft\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=7;

*--*

*-- AUTO\_INCREMENT для таблицы `destination`*

*--*

ALTER TABLE `destination`

MODIFY `destination\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=9;

*--*

*-- AUTO\_INCREMENT для таблицы `fly`*

Продолжение приложения 3

*--*

ALTER TABLE `fly`

MODIFY `fly\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=15;

*--*

*-- AUTO\_INCREMENT для таблицы `pilots`*

*--*

ALTER TABLE `pilots`

MODIFY `pilot\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=9;

*--*

*-- AUTO\_INCREMENT для таблицы `terminal`*

*--*

ALTER TABLE `terminal`

MODIFY `terminal\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=9;

*--*

*-- AUTO\_INCREMENT для таблицы `users`*

*--*

ALTER TABLE `users`

MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=3;

*--*

*-- Ограничения внешнего ключа сохраненных таблиц*

*--*

*-- Ограничения внешнего ключа таблицы `aircraft`*

*--*

ALTER TABLE `aircraft`

Продолжение приложения 3

ADD CONSTRAINT `aircraft\_ibfk\_1` FOREIGN KEY (`pilot\_id`) REFERENCES `pilots` (`pilot\_id`);

*--*

*-- Ограничения внешнего ключа таблицы `fly`*

*--*

ALTER TABLE `fly`

ADD CONSTRAINT `fly\_ibfk\_4` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`),

ADD CONSTRAINT `fly\_ibfk\_1` FOREIGN KEY (`aircraft\_id`) REFERENCES `aircraft` (`aircraft\_id`),

ADD CONSTRAINT `fly\_ibfk\_2` FOREIGN KEY (`terminal\_id`) REFERENCES `terminal` (`terminal\_id`),

ADD CONSTRAINT `fly\_ibfk\_3` FOREIGN KEY (`destination\_id`) REFERENCES `destination` (`destination\_id`);