|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 Робототехника и комплексная автоматизация

КАФЕДРА РК6 (САПР)

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ БАЗЫ ДАННЫХ**

**НА ТЕМУ:**

**Разработка информационной системы**

**«Бонусная программа»**

Студент РК6-43 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Агаев Г.С.**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пивоварова Н.В.**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2020 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 РК6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Карпенко А.П.

(И.О.Фамилия)

« » февраля 2019 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине Базы данных

Студент группы РК6-43 Агаев Г.С.

Тема курсовой работы Разработка информационной системы «Бонусная программа»

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

Источник тематики (кафедра, предприятие, НИР) \_Кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Задание*** Прилагается отдельном листе.

***Оформление курсовой работы:***

1. Расчетно-пояснительная записка на 20 листах формата А4.

1. Реализация Информационной системы на персональном компьютере студента.

Дата выдачи задания « 12 » февраля 2019 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пивоварова Н.В.

(Подпись, дата)

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Агаев Г.С.

(Подпись, дата) (И.О.Фамилия)

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

Оглавление

[Аннотация 2](#_Toc41052451)

[Описание предметной области 3](#_Toc41052452)

[Техническое задание 4](#_Toc41052453)

[Этап проектирования 4](#_Toc41052454)

[Этап реализации 5](#_Toc41052455)

[Разработка инфологической модели предметной области 6](#_Toc41052456)

[Построение логической модели БД 8](#_Toc41052457)

[Разработка оперативных запросов 9](#_Toc41052458)

[Разработка стандартного статического отчета 11](#_Toc41052459)

[Поставленная задача 11](#_Toc41052460)

[План процедуры 11](#_Toc41052461)

[Текст процедуры 12](#_Toc41052462)

[Итоговая логическая модель БД 13](#_Toc41052463)

[Разработка приложения конечного пользователя 14](#_Toc41052464)

[Запуск приложения 14](#_Toc41052465)

[Контроллер запросов 14](#_Toc41052466)

[Контроллер хранимой процедуры 14](#_Toc41052467)

[Заключение 16](#_Toc41052468)

[Список литературы 17](#_Toc41052469)

# Аннотация

В курсовом проекте поставлена задача создания информационной системы «Бонусная программа» для авиакомпании. Реализуется данная задача с помощью базы данных MySQL и такими языками программирования, как HTML и Python. В ходе реализации выполняется следующая работа:

* Разрабатывается инфологическая модель предметной области
* Разрабатывается логическая модель базы данных
* Разрабатываются оперативные запросы
* Разрабатывается стандартный статический отчет
* Разрабатывается хранимая процедура
* Разрабатывается веб-приложение для конечного пользователя

# Описание предметной области

Авиакомпания ведет бонусную программу.

Каждый *пассажир*, купивший хотя бы один *билет* на рейсы авиакомпании, имеет право

вступить в эту программу.

Сведения о пассажирах, вступивших в бонусную программу, сохраняются в БД

авиакомпании.

В соответствии с правилами‚ участники бонусной программы получают бонусы за

каждый купленный и использованный для полета авиабилет. Количество бонусов,

начисляемых за билет, зависит от *рейса* и даты его *вылета*, и определяется по

специальной *шкале*.

Все бонусы‚ накопленные участниками программы, сохраняются в БД, и в дальнейшем

могут быть использованы для покупки новых билетов.

# Техническое задание

## Этап проектирования

1. Разработать инфологическую модель предметной области. Письменно обосновать выделение сущностей и связей между ними, а также назначение атрибутов сущностей.
2. Разработать реляционную логическую модель будущей базы данных.
3. Разработать 6 оперативных запросов.
4. Разработать форму регулярного отчета, который позволит оценить эффективность соответствующего бизнеса.
5. Разработать план хранимой процедуры для реализации регулярного отчета.
6. Разработать интерфейс конечного пользователя для доступа к оперативным запросам и регулярному отчету.

## Этап реализации

1. Создать базу данных на основе разработанной логической модели.
2. Реализовать оперативные запросы, создав для отладки каждого из них тестовый набор данных в соответствующих таблицах.
3. Написать и отладить хранимую процедуру для реализации регулярного отчета.
4. Написать и отладить интерфейс конечного пользователя в форме меню, позволяющего выполнить любой запрос и получить регулярный отчет за любой период времени.

# Разработка инфологической модели предметной области

Перед тем, как создать инфологическую модель, выделим из описания предметной области сущности и атрибуты этих сущностей. При этом среди атрибутов сущностей уникальный идентификатор каждой сущности подчеркнем, а сами сущности напишем полужирным шрифтом.

* **Пассажир**
  + *id\_passenger*
  + ФИО
  + День рождения
  + № паспорта
  + Количество бонусов
  + Дата последнего обновления данных
* **История**
  + *id\_history*
  + Новая история
  + Старая история
* **Билет**
  + *id\_ticket*
  + Класс
  + Дата покупки
  + Цена
* **Вылет**
  + *id\_departure*
  + Аэропорт вылета
  + Аэропорт прилета
  + Рейс
  + Дата и время вылета
* **Шкала**
  + *id\_scale*
  + Начальная цена билета
  + Конечная цена билета
  + Количество начисляемых бонусов

Шкала

Вылет

Вылет

Изменение

Купил

Билет

История

История

Пассажир

M

1

1

1

1

М

М

1

Рисунок 1. Инфологическая модель

# Построение логической модели БД

Для построения логической модели будущей базы данных применим следующие

формальные правила.

1. Каждая сущность инфологической модели отображается отдельной таблицей.
2. Каждая связь типа многие-ко-многим отображается отдельной таблицей.
3. Каждая связь типа один-ко-многим – дополнительным столбцом в дочерней

таблице, и этот столбец становится внешним ключом, связанным с первичным ключом соответствующей родительской таблицы.

В результате применения этих правил получим следующий набор реляционных таблиц.

Таблица 1. Пассажир - **Passenger**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id\_passenger | Full\_name | Birthday | Bonuses | Data\_update\_date | Passport\_N |
| PK |  |  |  |  |  |

Таблица 2. История - **History**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_history | id\_passenger | Old\_history | New\_history |
| PK | FK |  |  |

Таблица 3. Билет - **Ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id\_ticket | id\_passenger | id\_departure | Class | Date\_of\_purchase | price |
| PK | FK | FK |  |  |  |

Таблица 4. Вылет - **Departure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_departure | Flight | Airport 1 | Airport 2 | Date\_and\_Time |
| PK |  |  |  |  |

Таблица 5. Шкала - **Scale**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_scale | Starting\_price | Final\_price | Number\_of\_bonuses |
| PK |  |  |  |

# Разработка оперативных запросов

**1. Создать отчет о начислении бонусных миль по форме: Номер рейса, дата вылета, общее количество начисленных бонусных миль.**

SELECT Flight, Date\_and\_Time, SUM(Number\_of\_bonuses)

FROM departure d JOIN (SELECT Number\_of\_bonuses, id\_departure

FROM ticket t JOIN scale s ON s.Starting\_price <= t.price AND

t.price <= s.Final\_price) n USING (id\_departure)

GROUP BY Flight;

**2. Создать отчет о продаже билетов на каждый рейс за 2017 года по форме: Название аэропорта, месяц вылета, класс билета, общее число проданных билетов.**

SELECT Airport\_1, Airport\_2, MONTHNAME(Date\_and\_Time), Class, COUNT(\*)

FROM ticket JOIN departure USING (id\_departure)

WHERE YEAR(Date\_and\_Time) = 2017

GROUP BY Airport\_1, Airport\_2, MONTH(Date\_and\_Time), Class;

**3. Покажите все сведенья о пассажирах, покупавших самые дорогие билеты на рейс ХХХ.**

SELECT passenger.\*

FROM passenger JOIN ticket USING (id\_passenger)

WHERE Price = (SELECT MAX(Price)

FROM ticket JOIN departure USING (id\_departure)

WHERE Flight = "XXX");

**4. Покажите все сведенья о пассажирах, которые зарегистрированы в БД , но ни разу не покупавших билеты(с помощью левостороннего соединения).**

SELECT passenger.\*

FROM passenger LEFT JOIN ticket USING (id\_passenger)

WHERE id\_ticket IS NULL;

**5. Покажите все сведенья о пассажирах, не покупавшие билеты в марте 2014 года.**

SELECT p.\*

FROM passenger p LEFT JOIN (SELECT id\_passenger, id\_ticket

FROM ticket

WHERE YEAR(Date\_of\_purchase) = 2014 AND

MONTH(Date\_of\_purchase) = 3) t USING (id\_passenger)

WHERE id\_ticket IS NULL;

**6. Покажите все сведенья о пассажирах, чаще всех покупавших билеты в марте-апреле 2013 года( с помощью view).**

CREATE VIEWpass\_ticket(id\_passenger, Full\_name, Birthday, Bonuses, Data\_update\_date, Passport\_N, count\_ticket) AS

SELECT p.id\_passenger, p.Full\_name, p.Birthday, p.Bonuses, p.Data\_update\_date, p.Passport\_N, COUNT(t.id\_passenger)

FROM passenger p, ticket t

WHERE p.id\_passenger = t.id\_passenger AND DATE(Date\_of\_purchase) >= '2013-03-01' AND DATE(Date\_of\_purchase) <= '2013-04-31'

GROUP BY id\_passenger;

SELECT \*

FROM pass\_ticket

WHERE count\_ticket = (SELECT MAX(count\_ticket)

FROM pass\_ticket);

# Разработка стандартного статического отчета

## Поставленная задача

Разработать процедуру, которая будет принимать на вход месяц и год. В процедуре необходимо посчитать общую стоимость билетов на каждый рейс, проданный в этот месяц, а также общее количество начисленных бонусов пассажирам этого рейса. Результат должен быть представлен в виде отдельной таблице.

Таблица 6. Отчет - **Report**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_report | Flight | Total\_cost | Bonus\_amount |
| PK |  |  |  |

## План процедуры

1. В декларативной части процедуры задекларируем:

* Переменные s\_price для хранения общей стоимости билетов, s\_bonuses для хранения общего количество начисленных бонусов и n\_flight для хранения номера рейса;
* Курсор для извлечения номера рейса, общей стоимости билетов и общего количество начисленных бонусов из столбцов Flight, Price и Number\_of\_bonuses, которые расположены в таблицах departure, ticket и scale соответственно;
* Обработчик, который должен отслеживать достижение конца курсора;

1. Очистим полностью таблицу report для того, чтобы поместить новый отчет заместо старого;
2. Откроем курсор;
3. Организуем цикл, в котором будем извлекать записи из курсора по одной. Исходя из извлеченных данных заполним таблицу report. Год и месяц являются входными параметрами процедуры;
4. По окончанию цикла закроем курсор.

## Текст процедуры

CREATE DEFINER=`root`@`localhost` PROCEDURE `report`(month\_of\_purchase INTEGER, year\_of\_purchase INTEGER)

BEGIN

DECLARE s\_price FLOAT;

DECLARE s\_bonuses INTEGER;

DECLARE n\_flight VARCHAR(8);

DECLARE done INTEGER DEFAULT 0;

DECLARE rpt CURSOR FOR SELECT Flight, SUM(Price),SUM(Number\_of\_bonuses)

FROM departure JOIN (SELECT id\_departure, Date\_of\_purchase, Price, Number\_of\_bonuses

FROM ticket t JOIN scale s ON s.Starting\_price <= t.price AND

t.price <= s.Final\_price) n USING (id\_departure)

WHERE YEAR(Date\_of\_purchase) = year\_of\_purchase AND

MONTH(Date\_of\_purchase) = month\_of\_purchase

GROUP BY Flight;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

TRUNCATE TABLE bonus\_program.report;

OPEN rpt;

REPEAT

FETCH rpt INTO n\_flight, s\_price, s\_bonuses;

IF (done = 0) THEN

INSERT INTO bonus\_program.report VALUES(NULL, n\_flight, s\_price, s\_bonuses);

END IF;

UNTIL done = 1 END REPEAT;

CLOSE rpt;

END

# Итоговая логическая модель БД

Итоговая модель базы данных получена объединением таблиц, описанных при первом построении логическая модели, и таблицы, добавленной в ходе разработки структуры отчёта.

Таблица 1. Пассажир - **Passenger**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id\_passenger | Full\_name | Birthday | Bonuses | Data\_update\_date | Passport\_N |
| PK |  |  |  |  |  |

Таблица 2. История - **History**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_history | id\_passenger | Old\_history | New\_history |
| PK | FK |  |  |

Таблица 3. Билет - **Ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id\_ticket | id\_passenger | id\_departure | Class | Date\_of\_purchase | price |
| PK | FK | FK |  |  |  |

Таблица 4. Вылет - **Departure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_departure | Flight | Airport 1 | Airport 2 | Date\_and\_Time |
| PK |  |  |  |  |

Таблица 5. Шкала - **Scale**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_scale | Starting\_price | Final\_price | Number\_of\_bonuses |
| PK |  |  |  |

Таблица 6. Отчет - **Report**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_report | Flight | Total\_cost | Bonus\_amount |
| PK |  |  |  |

# Разработка приложения конечного пользователя

Все файлы полученные в ходе разработки web-приложения хранятся в директории D:\Documents\Database\CourseWork. В корне данной директории находится основной модуль menu.py, в котором происходит вызов HTML страниц и обработка событий. Файлы со всеми статическими HTML страницами и шаблонами для динамических HTML страниц хранятся в поддиректории Templates. Всего таких файлов 10, 7 из них отвечают за вывод результата оперативных запросов и процедуры, еще один представляет из себя пользовательское(menu.html), оставшиеся отвечают за сообщения об ошибках таких, как данные не найдены и соединения с базой данных не установлено. За само соединения с базой данных отвечает процедура db\_connect расположенная в одноименном модуле, которая в свою очередь располагается в поддиректории utils. Данная процедура импортируется в основной модуль menu.py при помощи from…import, после чего используется во всех процедурах данного модуля, кроме menu(), так как menu() отвечает за вызов нужной процедуры в соответствии с действиями пользователя или же сразу HTML страницы. Чтобы сократить HTML код общая стилистическая часть HTML страниц вынесена в отдельный файл под названием style.css, которая расположена в поддиректории “Static”.

# Заключение

Таким образом, в ходе выполнения курсовой работы по БД была создана информационная система с помощью базы данных MySQL, а также web приложение для данной информационной системы. Интерфейс web приложения создавался с использованием языков программирования HTML и CSS, управляющий скрипт писался на Python.

# Список литературы

* Белоус В.В., Пивоварова Н.В. Методические указания к курсовой работе по дисциплине «Базы данных» – Москва, 2020 – МГТУ им. Н.Э. Баумана – 22 с.