|  |  |
| --- | --- |
| 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 **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***по дисциплине «Базы данных»***

***НА ТЕМУ:***

***\_\_ База данных Партия любителей пива\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-42Б |  | 17.04.2021 | И.С. Марчук |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Руководитель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |

*2021 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 \_\_\_ИУ6\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Пролетарский

« 9 » февраля 2021 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине Базы данных

Студент группы ИУ6-42б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Марчук Иван Сергеевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсовой работы \_\_\_\_\_\_База данных «Партия любителей пива» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_учебная\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

График выполнения работы: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

***Задание:*** разработать базу данных «Партия любителей пива», содержащую не менее 7 таблиц. В основной таблице должно содержаться не менее 100 тыс. записей, в других таблицах должно содержаться не менее 10 записей. В БД должно быть разработано не менее 5 сложных запросов и не менее 2 сложных отчетов.

***Оформление курсовой работы:***

Расчетно-пояснительная записка на 25-30 листах формата А4.

Дата выдачи задания «8» февраля 2021 г.

|  |  |  |  |
| --- | --- | --- | --- |
| **Руководитель** |  |  | Д.А. Миков |
|  |  | (Подпись, дата) | (И.О. Фамилия) |
| **Студент** |  |  | И.С. Марчук |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

Оглавление

[**Введение** 3](#_Toc72083361)

[**Глава 1. Постановка задачи и проектирование** 4](#_Toc72083362)

[**1.1 Описание предметной области** 4](#_Toc72083363)

[**1.2 Выбор средств реализации БД и интерфейса** 5](#_Toc72083364)

[**1.3 Описание сущностей БД** 6](#_Toc72083365)

[**Глава 2. Реализация** 14](#_Toc72083366)

[**2.1 Заполненные таблицы** 14](#_Toc72083367)

[**2.2 Описание функций БД** 20](#_Toc72083368)

[**2.3 Интерфейс пользователя** 21](#_Toc72083369)

[**Заключение** 28](#_Toc72083370)

[**Список источников** 29](#_Toc72083371)

[**Приложение** 30](#_Toc72083372)

# **Введение**

Сейчас существует большое количество политических партий, включающих в себя сотни и тысячи людей. Ежедневно необходимо рассылать партийные новости, вести учет всех работников, осуществлять заказы необходимых для партии вещей и контролировать выплаты партийных взносов. Вести учет всего этого переписываясь по электронной почте или используя бумажные носители очень неудобно, а иногда и невозможно. Необходима система, позволяющая структурировать большое количество записей о пользователях в едином пространстве для поддержания актуальности сведений и имеющая возможность массовой рассылки уведомлений для членов партии. Все эти возможности должны быть доступны в любое время и с любого компьютера, подключенного к Интернету.

# **Глава 1. Постановка задачи и проектирование**

**Цель:** разработать систему, для партии любителей пива[1] позволяющую вести учет партийных работников и их партийных билетов, публиковать общедоступные новости, делать партийные заказы и контролировать выплаты членских взносов.

**Задачи:**

- разработать БД для хранения информации о членах партии и их парт. билетах, о различных марках пива и партийных заказах, о глобальных новостях и о членских взносах;

- разработать функции и запросы в БД, осуществляющие получение и запись необходимой информации через простой интерфейс;

- разработать графический интерфейс пользователя для взаимодействия с реализованной БД с использованием веб-браузера.

# **1.1 Описание предметной области**

Система должна учитывать особенности устройства и взаимодействия между собой частей партии: списки пользователей, уникальный партийный билет для каждого пользователя выдаваемый после одобрения руководством и имеющий также возможность менять свою должность. Необходима система должностей варьирующая доступ к некоторым функциям и регулирующая членский взнос. Также обязательна система рассылки глобальных новостей. Так как партия специализируется на марках пива, то необходим каталог, содержащий в себе наиболее полный список всех сортов с полным описанием и возможность создания партийных заказов пива (в дальнейшем эту систему можно будет расширить до любой специализации).

# **1.2 Выбор средств реализации БД и интерфейса**

В качестве СУБД в данной работе используется ORACLE Database 11g[5]. Это открытая, кроссплатформенная СУБД, поддерживающая PL/SQL[4], индексы и имеющая продвинутый конструктор приложений APEX[3] поддерживающий огромное количество форматов от веб-сайтов до мобильных приложений. В качестве пользовательского интерфейса было выбрано именно приложение, созданное в APEX. Разработка собственного пользовательского интерфейса обусловлена особенностями разработки интерфейсов для данной предметной области. Схема разработанной БД представлена на рисунке 1:

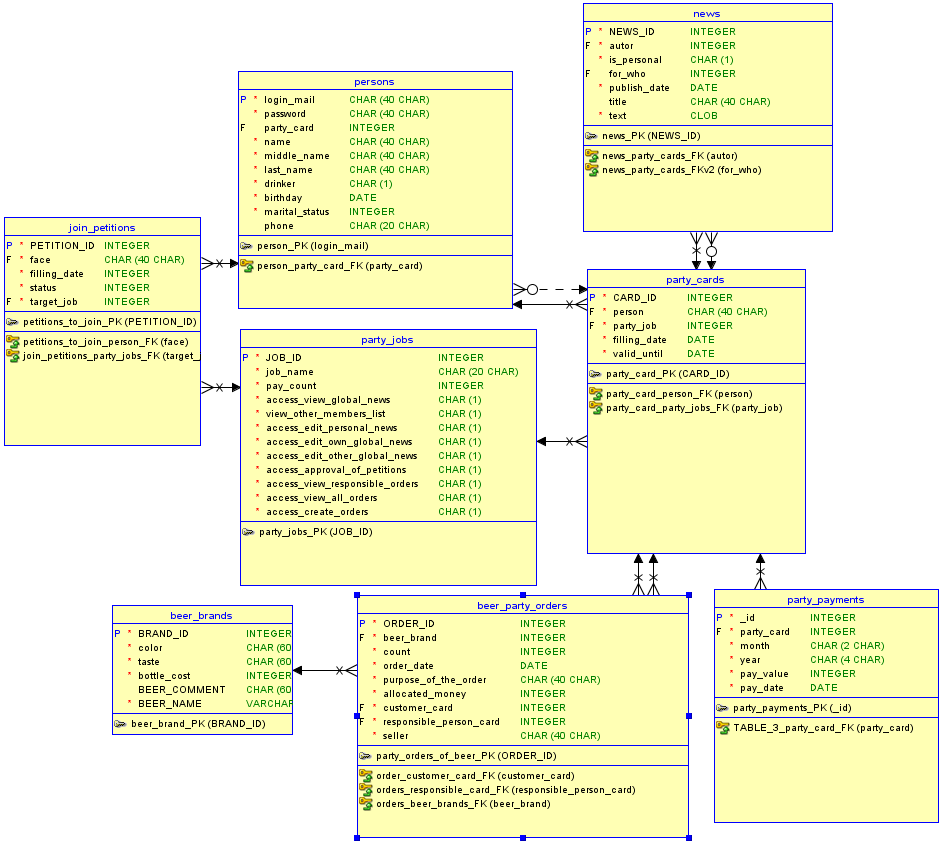


Рисунок 1 - Схема БД

# **1.3 Описание сущностей БД**

Для создания соответствующих таблиц был написан код на языке SQL[4].

Beer brands – таблица марок пива

CREATE TABLE beer\_brands (

brand\_id INTEGER NOT NULL PRIMARY KEY,

color CHAR(60 CHAR) NOT NULL,

taste CHAR(60 CHAR) NOT NULL,

bottle\_cost INTEGER NOT NULL,

beer\_comment CHAR(60 CHAR),

beer\_name VARCHAR2(60) NOT NULL

);

- brand\_id – идентификатор бренда пива;

- color – цвет пива;

- taste – вкус пива;

- bottle\_cost – цена за бутылку;

- beer\_comment – комментарий;

- beer\_name – название марки.

beer\_party\_orders – таблица партийных заказов пива

CREATE TABLE beer\_party\_orders (

order\_id INTEGER NOT NULL PRIMARY KEY,

beer\_brand INTEGER NOT NULL,

count INTEGER NOT NULL,

order\_date DATE NOT NULL,

purpose\_of\_the\_order CHAR(40 CHAR) NOT NULL,

allocated\_money INTEGER NOT NULL,

customer\_card INTEGER NOT NULL,

responsible\_person\_card INTEGER NOT NULL,

seller CHAR(40 CHAR) NOT NULL,

FOREIGN KEY(customer\_card) REFERENCES party\_cards (card\_id) CASCADE,

FOREIGN KEY (beer\_brand) REFERENCES beer\_brands (brand\_id) CASCADE,

FOREIGN KEY(responsible\_person\_card) REFERENCES party\_cards(card\_id) CASCADE

);

- order\_id – идентификатор заказа;

- beer\_brand – идентификатор марки пива;

- count – количество бутылок;

- order\_date – дата заказа;

- purpose\_of\_the\_order – цель заказа;

- allocated\_money – выделенные деньги;

- customer\_card – идентификатор заказчика;

- responsible\_person\_card – идентификатор ответственного лица;

- seller – продавец.

Join petitions – заявки на изменение партийной должности и продление партийного билета

CREATE TABLE join\_petitions (

petition\_id INTEGER NOT NULL PRIMARY KEY,

face CHAR(40 CHAR) NOT NULL,

filling\_date INTEGER NOT NULL,

status INTEGER NOT NULL,

target\_job INTEGER NOT NULL,

FOREIGN KEY (target\_job) REFERENCES party\_jobs (job\_id) CASCADE,

FOREIGN KEY (face) REFERENCES persons (login\_mail) CASCADE

);

- petition\_id – уникальный номер записи (первичный ключ);

- face – логин лица, подавшего заявление;

- filling\_date – дата подачи;

- status – статус заявления;

- target\_job – целевая должность.

News – партийные новости

CREATE TABLE news (

news\_id INTEGER NOT NULL PRIMARY KEY,

autor INTEGER NOT NULL,

is\_personal CHAR(1) NOT NULL,

for\_who INTEGER,

publish\_date DATE NOT NULL,

title CHAR(40 CHAR),

text CLOB NOT NULL,

FOREIGN KEY (autor) REFERENCES party\_cards (card\_id) CASCADE, FOREIGN KEY (for\_who) REFERENCES party\_cards (card\_id) SET NULL

);

- news\_id – уникальный номер записи (первичный ключ);

- autor – автор записи;

- is\_personal – новость - личная или для массовой рассылки;

- for\_who – в случае личной новости указывается также и получатель;

- publish\_date – дата публикации новости;

- title – заголовок новости;

- text – текст новости.

party\_cards – партийные билеты участников

CREATE TABLE party\_cards (

card\_id INTEGER NOT NULL PRIMARY KEY,

person CHAR(40 CHAR) NOT NULL,

party\_job INTEGER NOT NULL,

filling\_date DATE NOT NULL,

valid\_until DATE NOT NULL,

FOREIGN KEY (party\_job) REFERENCES party\_jobs (job\_id) CASCADE,

FOREIGN KEY (person) REFERENCES persons (login\_mail) CASCADE

);

- card\_id – идентификатор билета;

- person – идентификатор лица использующего билет;

- party\_job – должность;

- filling\_date – дата получения;

- valid\_until – срок годности.

party\_jobs – партийные должности

CREATE TABLE party\_jobs (

job\_id INTEGER NOT NULL PRIMARY KEY,

job\_name CHAR(20 CHAR) NOT NULL,

pay\_count INTEGER NOT NULL,

access\_view\_global\_news CHAR(1) NOT NULL,

view\_other\_members\_list CHAR(1) NOT NULL,

access\_edit\_personal\_news CHAR(1) NOT NULL,

access\_edit\_own\_global\_news CHAR(1) NOT NULL,

access\_edit\_other\_global\_news CHAR(1) NOT NULL,

access\_approval\_of\_petitions CHAR(1) NOT NULL,

access\_view\_responsible\_orders CHAR(1) NOT NULL,

access\_view\_all\_orders CHAR(1) NOT NULL,

access\_create\_orders CHAR(1) NOT NULL

);

- job\_id – идентификатор записи;

- job\_name – название должности;

- pay\_count – членский взнос;

- access\_view\_global\_news – доступ на просмотр глобальных новостей;

- view\_other\_members\_list – доступ на просмотр списка пользователей;

- access\_edit\_personal\_news – доступ на отправку сообщений;

- access\_edit\_own\_global\_news – создание глобальных новостей;

- access\_edit\_other\_global\_news – доступ на редактирование чужих глобальных новостей (модерирование);

- access\_approval\_of\_petitions – доступ на изменение чужих парт билетов;

- access\_view\_responsible\_orders – доступ на просмотр заказов за которые лицо является ответственным;

- access\_view\_all\_orders – доступ на просмотр всех заказов;

- access\_create\_orders – доступ на создание партийных заказов.

party\_payments – членские взносы

CREATE TABLE party\_payments (

"\_id" INTEGER NOT NULL PRIMARY KEY,

party\_card INTEGER NOT NULL,

month CHAR(2 CHAR) NOT NULL,

year CHAR(4 CHAR) NOT NULL,

pay\_value INTEGER NOT NULL,

pay\_date DATE NOT NULL,

FOREIGN KEY (party\_card) REFERENCES party\_cards (card\_id) CASCADE

);

- \_id – идентификатор взноса;

- party\_card – идентификатор билета человека, внесшего взнос;

- month – оплачиваемый месяц;

- year – оплачиваемый год;

- pay\_value – сумма взноса;

- pay\_date – дата оплаты.

persons – члены партии

CREATE TABLE persons (

login\_mail CHAR(40 CHAR) NOT NULL PRIMARY KEY,

password CHAR(40 CHAR) NOT NULL,

party\_card INTEGER,

name CHAR(40 CHAR) NOT NULL,

middle\_name CHAR(40 CHAR) NOT NULL,

last\_name CHAR(40 CHAR) NOT NULL,

drinker CHAR(1) NOT NULL,

birthday DATE NOT NULL,

marital\_status INTEGER NOT NULL,

phone CHAR(20 CHAR),

FOREIGN KEY (party\_card) REFERENCES party\_cards (card\_id) SET NULL

);

- login\_mail – электронная почта-логин-идентификатор;

- password – пароль;

- party\_card – идентификатор партийного билета;

- name – имя;

- middle\_name – фамилия;

- last\_name – отчество;

- drinker – пьющий/не пьющий;

- birthday – дата рождения;

- marital\_status – статус отношений;

- phone – контактный телефон.

# **Глава 2. Реализация**

# **2.1 Заполненные таблицы**

Заполненная таблица «Пользователи» на рисунке 2

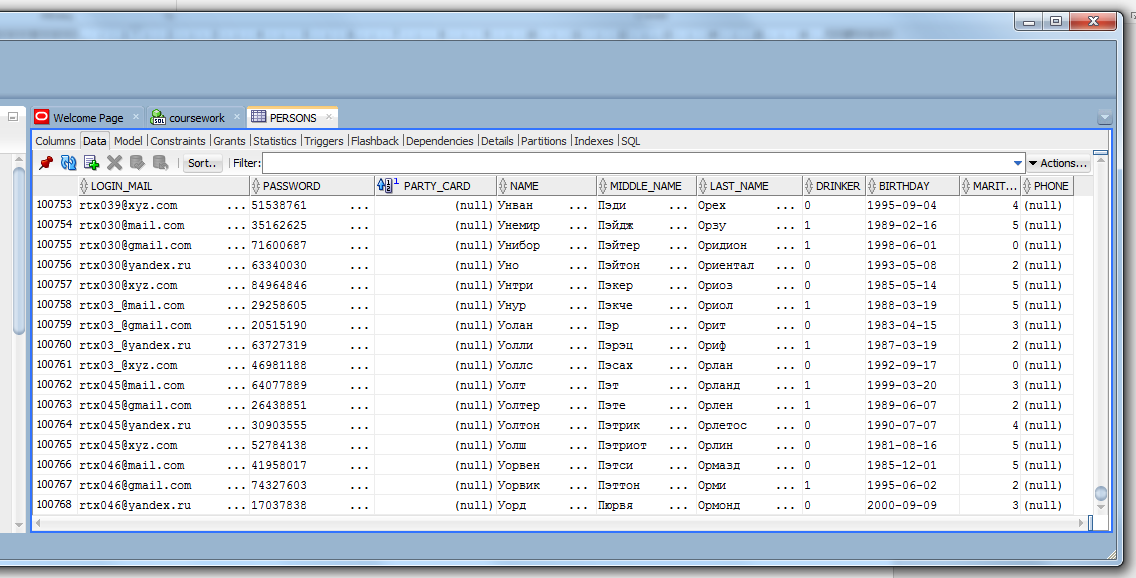


Рисунок 2 – PERSONS

Заполненная таблица «Членские взносы» на рисунке 3

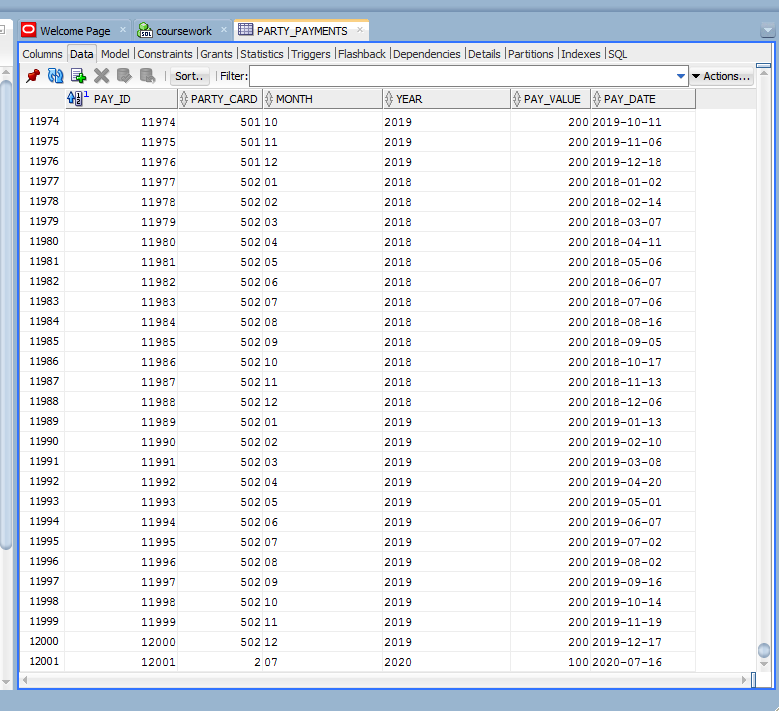


Рисунок 3 – PARTY\_PAYMENTS

Заполненная таблица «Партийные должности» на рисунке 4

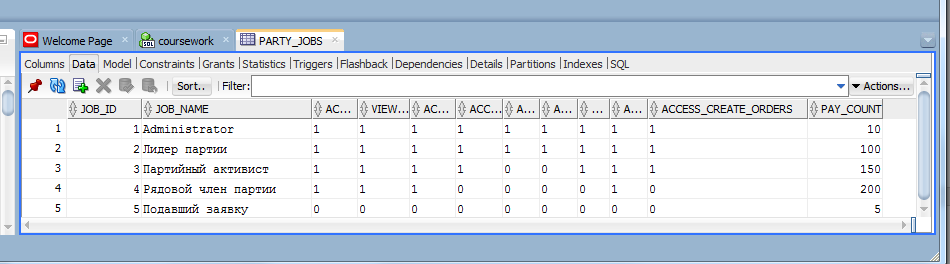


Рисунок 4 – PARTY\_JOBS

Заполненная таблица «Партийные билеты» на рисунке 5

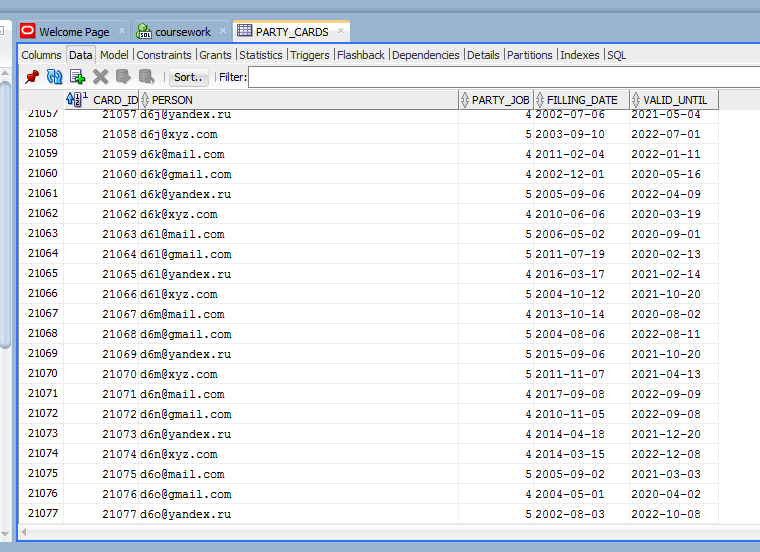


Рисунок 5 – PARTY\_CARDS

Заполненная таблица «Новости» на рисунке 6

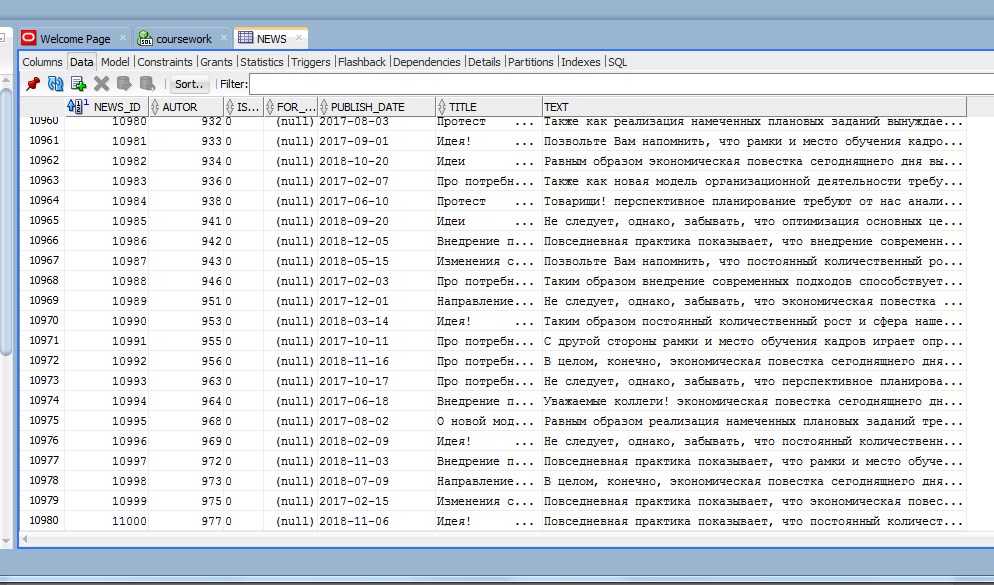


Рисунок 6 – NEWS

Заполненная таблица «Заявки на изменение или продление парт билета» на рисунке 7.

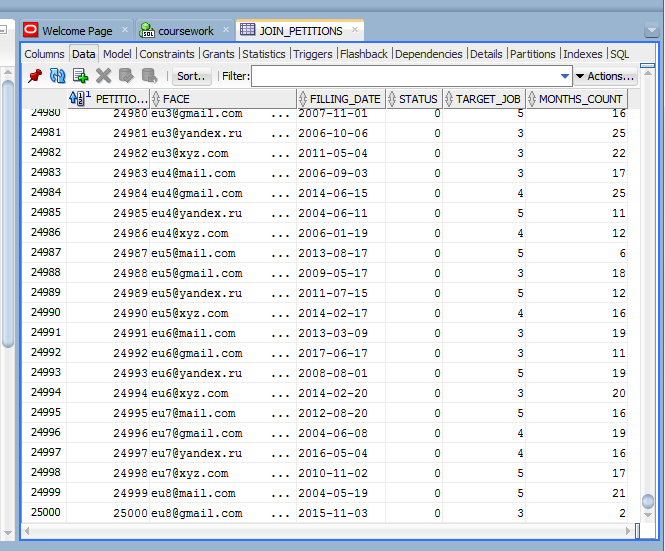


Рисунок 7 – JOIN\_PETITIONS

Заполненная таблица «Партийные заказы» на рисунке 8

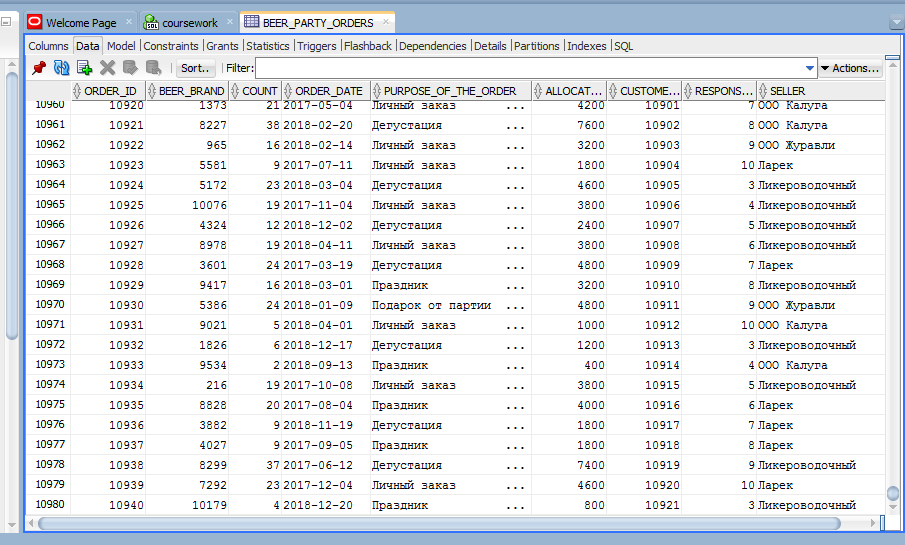


Рисунок 8 – BEER\_PARTY\_ORDERS

Заполненная таблица «Марки пива» на рисунке 9

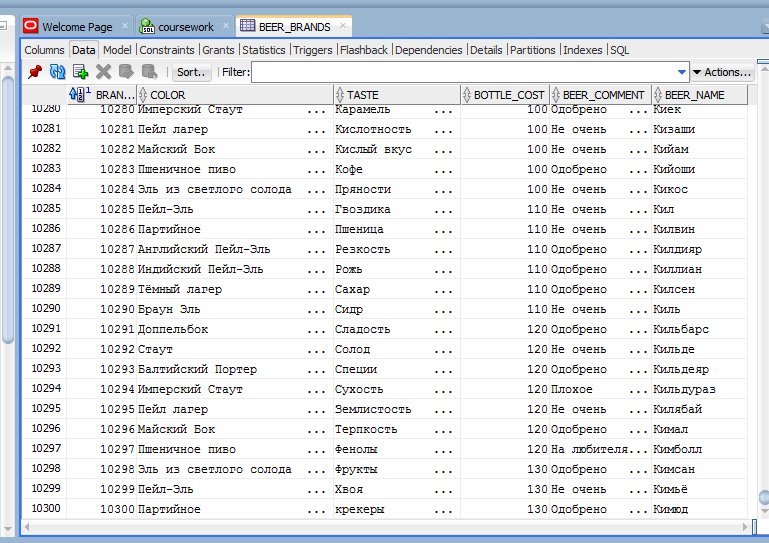


Рисунок 9 – BEER\_BRANDS

# **2.2 Описание функций БД**

Были разработаны функции, помогающие осуществлять контроль и поддерживать целостность БД. Они представлены в приложении.

Система может подстраиваться под особенности устройства и взаимодействия между собой частей партии, так как она имеет в себе: списки пользователей, уникальный партийный билет для каждого пользователя, выдаваемый после одобрения руководством и имеющий также возможность менять свою должность. Имеется настраиваемая система должностей, варьирующая доступ к некоторым функциям и регулирующая членский взнос. Также реализована система рассылки глобальных новостей. Так как партия специализируется на марках пива, имеется также каталог, содержащий в себе наиболее полный список всех сортов с детальным описанием, а также есть возможность создания партийных заказов пива, которую в дальнейшем можно будет доработать как систему учета и других заказов для удовлетворения нужд партии.

# **2.3 Интерфейс пользователя**

Разработанный графический интерфейс предназначен исключительно для пользователей и имеет 4 страницы:

- Страница входа и регистрации – позволяет пользователю создать аккаунт в системе и авторизоваться, для этого я применил достаточно распространенный способ регистрации пользователей[2].

- Персональная страница – позволяет просматривать и редактировать сведения о текущем пользователе и его партийном билете

- Новости партии – позволяет получать информацию об актуальных новостях, а также, при наличии прав, изменять чужие партийные билеты и просматривать выплаты

- Каталог пива – содержит информацию обо всех марках пива известных партии любителей пива, при наличии необходимых прав, пользователь может на этой странице сделать заказ для партии.

На страницах используется много таблиц, которые пользователь может отфильтровать и настроить под вои конкретные нужды, что обеспечивает хорошую гибкость интерфейса.

Рассмотрим разработанный интерфейс.

Скриншоты страницы «Входа и регистрации» на рисунках 10 и 11

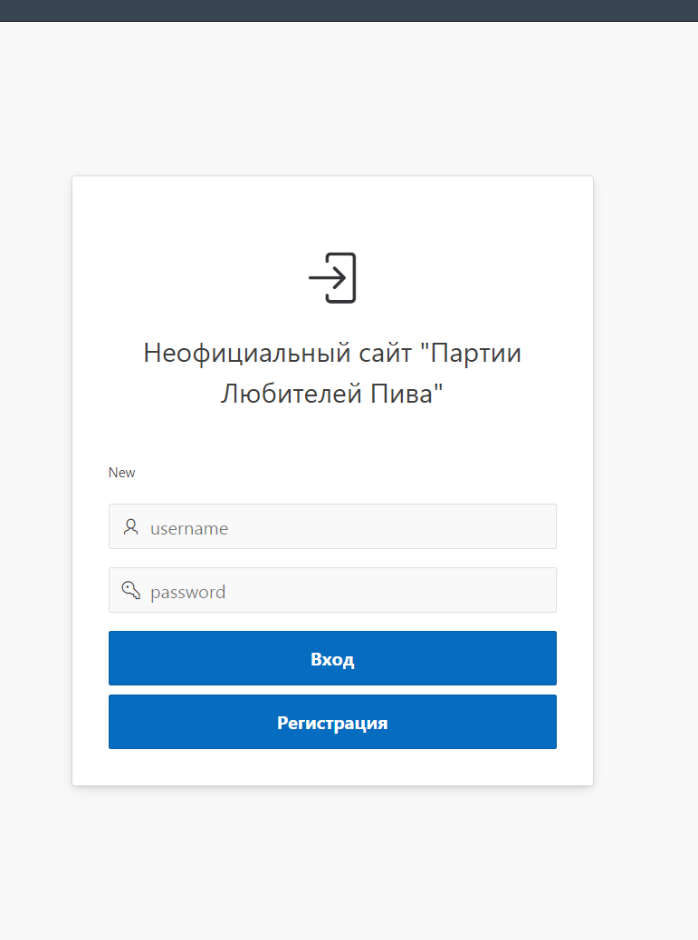


Рисунок 10 – Страница входа

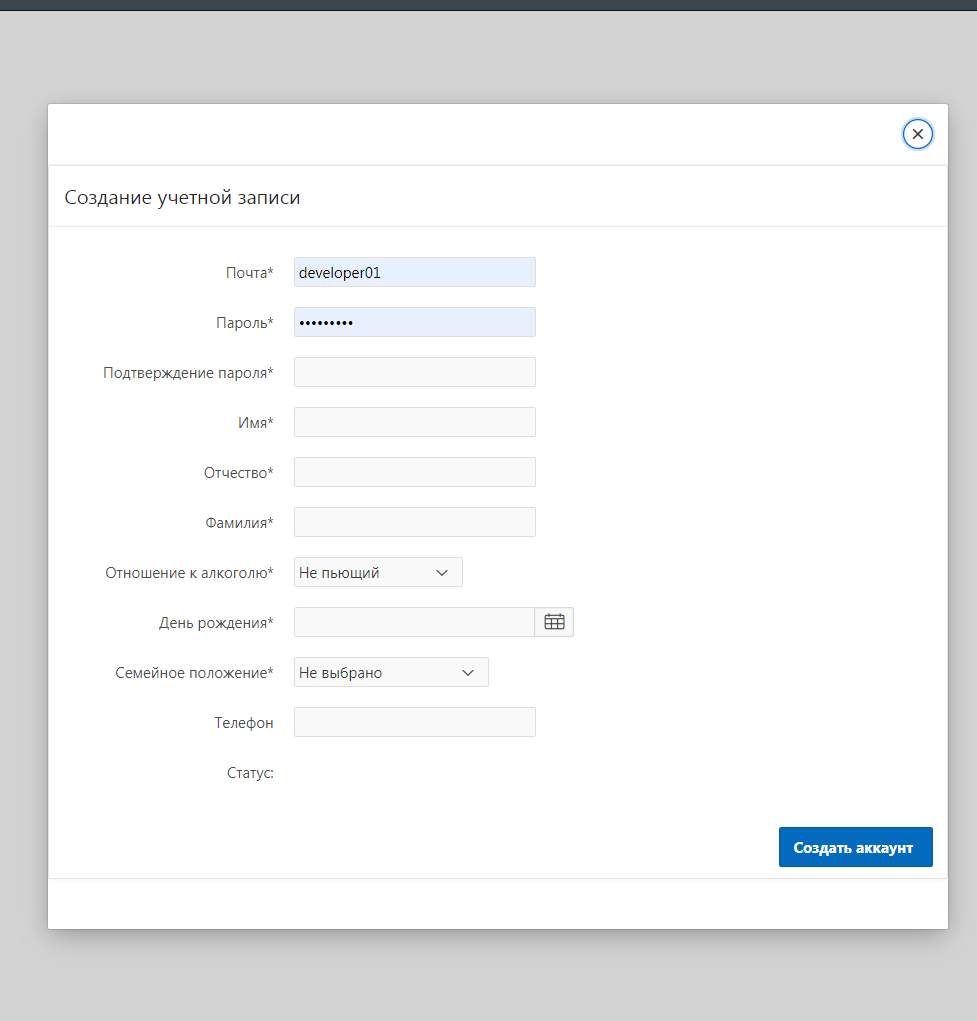


Рисунок 11 – Страница регистрации

Как только пользователь войдет в аккаунт, он будет перенаправлен на свою персональную страницу (рисунок 12). Здесь он может отредактировать свои данные, оплатить членские взносы, а также подать заявку на изменение или продление своего партийного билета.

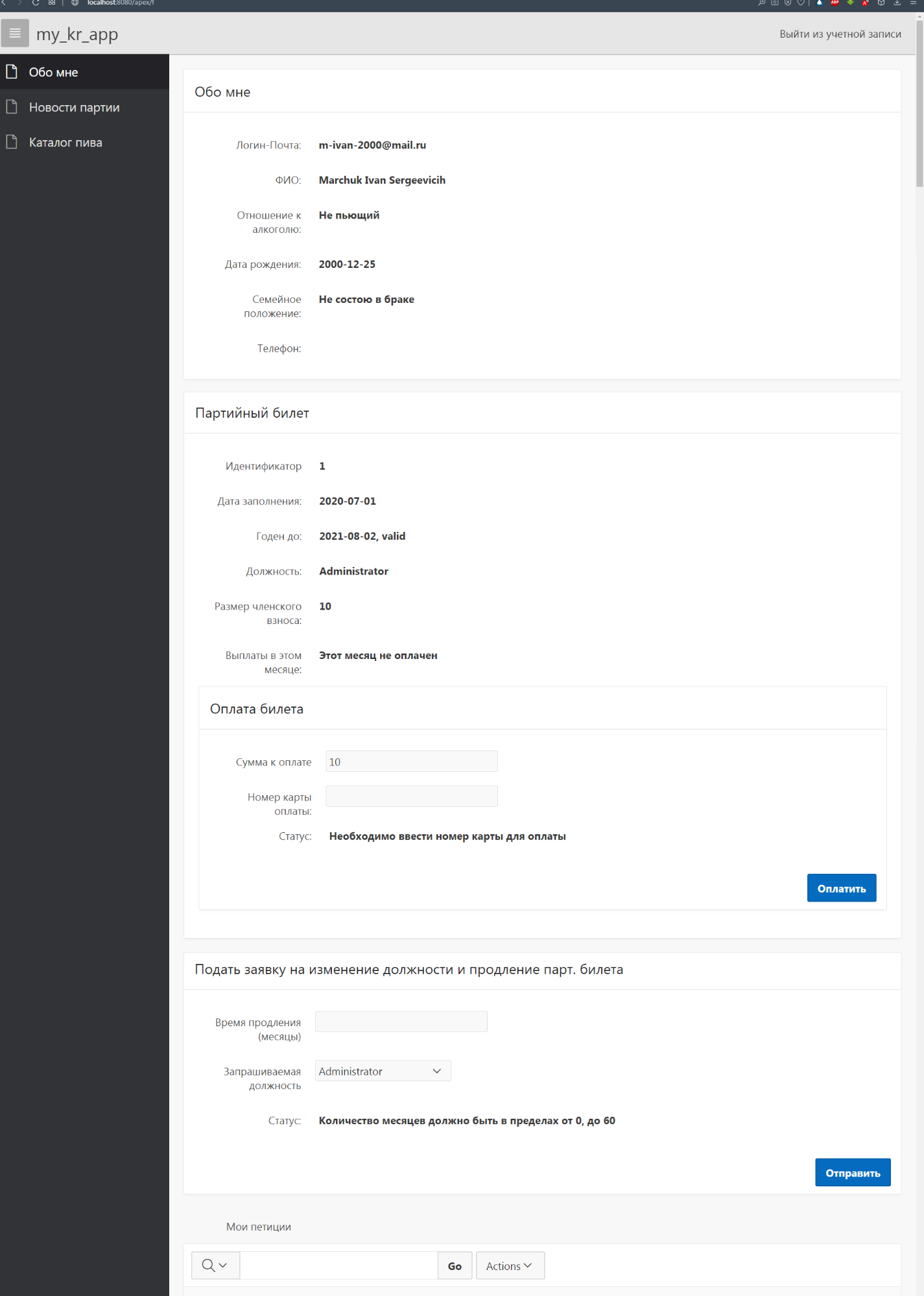


Рисунок 12 – «Персональная страница»

На странице новостей (рисунок 13) пользователь может читать актуальные новости партии, а также, при наличии прав, он может публиковать свои новости. Еще на этой странице пользователь может отвечать на заявки, на продление или изменение партийных билетов и просматривать членские взносы участников.

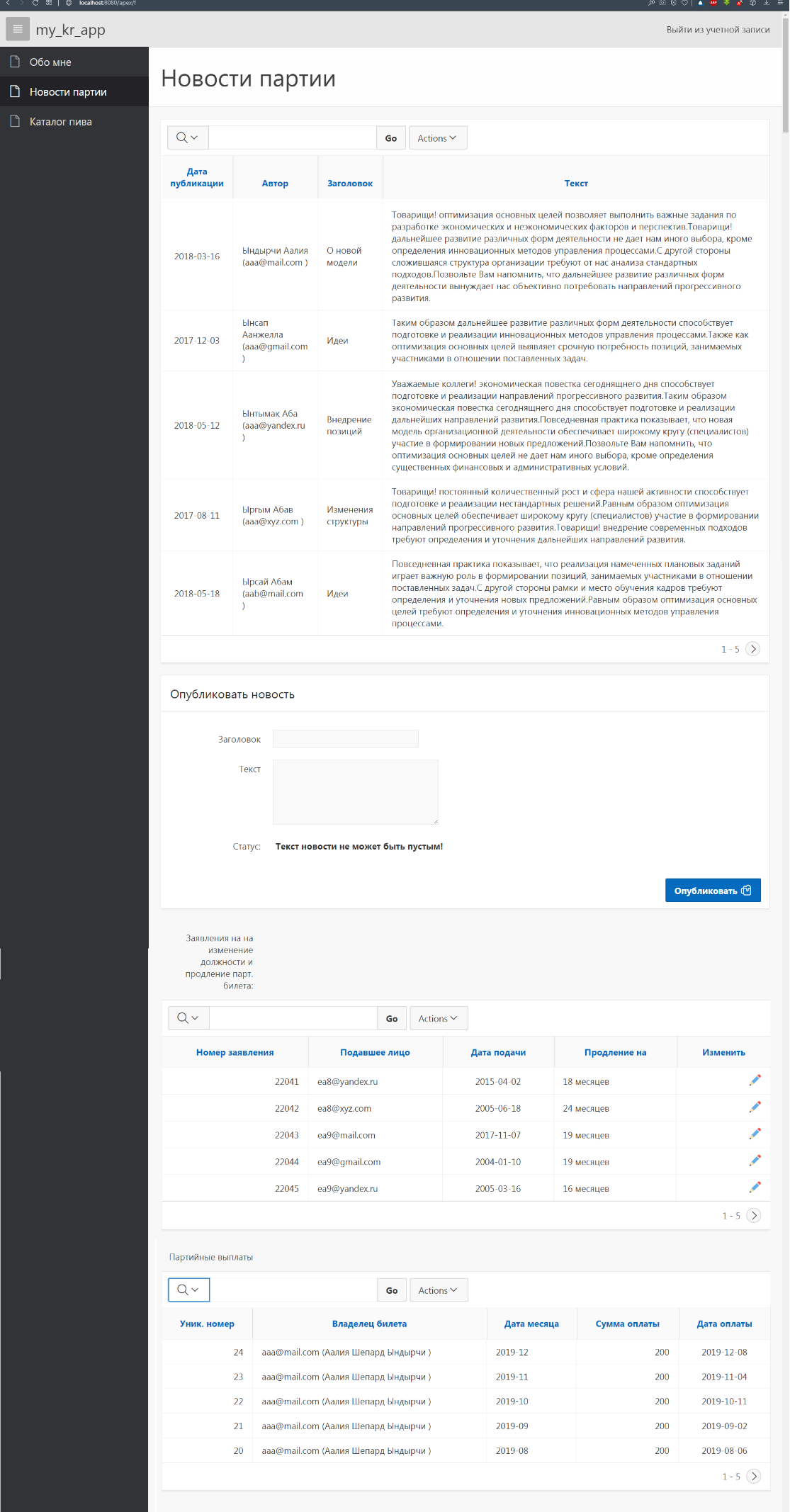


Рисунок 13 – «Новости партии»

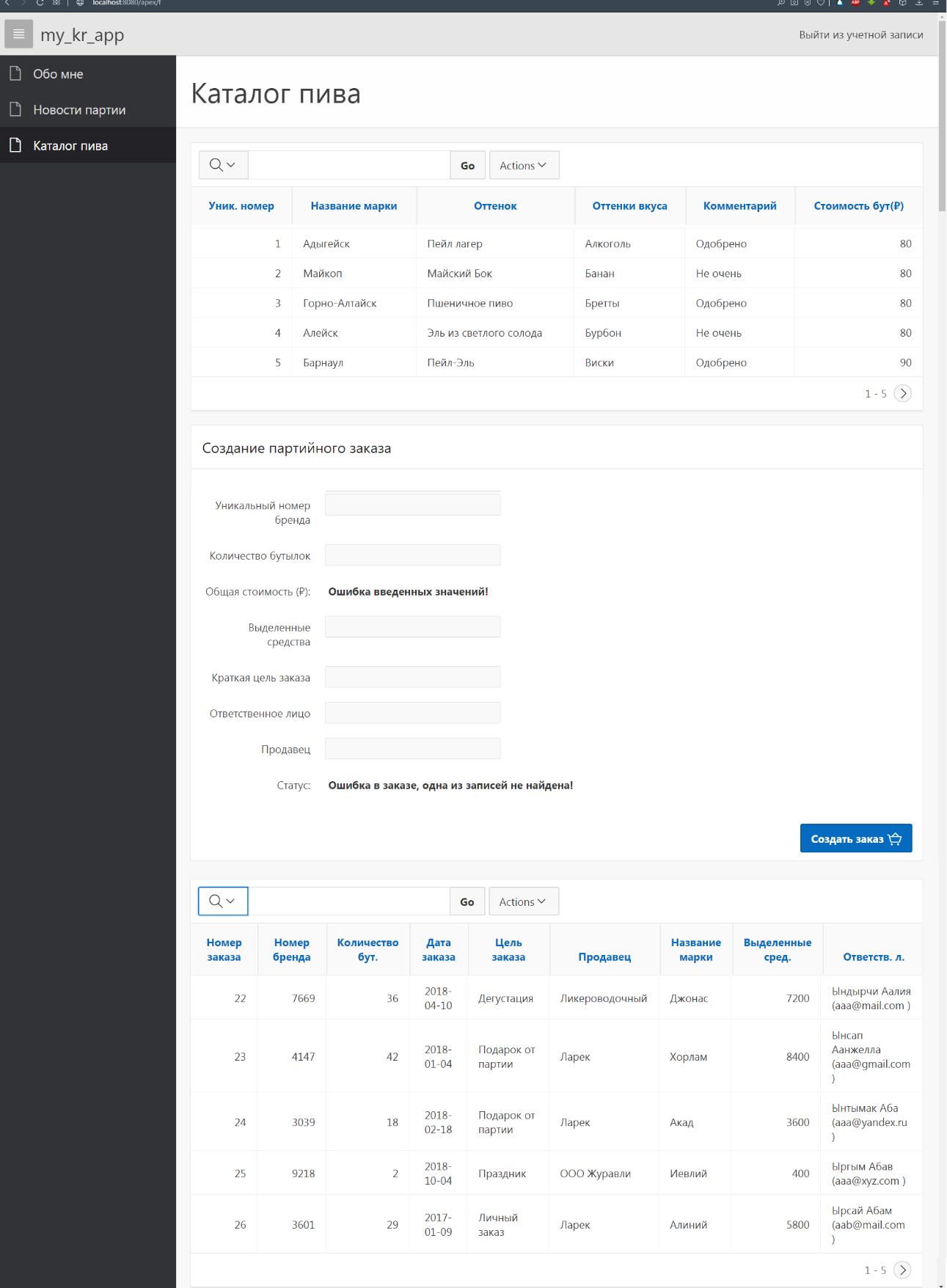


Рисунок 14 – «Каталог пива»

В каталоге пива (рисунок 14) пользователь может посмотреть информацию обо всех марках пива, известных партии. А также, при наличии необходимых прав, может сделать заказ для партийных нужд.

# **Заключение**

Была разработана система, для партии любителей пива позволяющая вести учет партийных работников и их партийных билетов, публиковать общедоступные новости, делать партийные заказы и контролировать выплаты членских взносов. Система позволяет быстро и оперативно рассылать всем участникам партии самые важные новости. В качестве доработки можно добавить чаты, позволяющие пользователям общаться непосредственно друг с другом. А также, реализовать заказы не только пива но и любых других частых товаров из интернет магазинов сформированных в удобные списки. В таком случае, приложение сможет объединить в себе все необходимые функции. Возможна также разработка мобильного приложения (или десктопного) за счет использования простого API-интерфейса. С точки зрения графического интерфейса пользователя реализовано современное веб-приложение, оптимизированное как под ПК, так и под мобильные устройства и позволяющее вести рассылку партийных новостей, вести учет всех работников, осуществлять заказы необходимой для партии продукции и контролировать выплаты партийных взносов. Данное приложение призвано оптимизировать работу политических партий, сэкономить их время и ресурсы.

# **Список источников**

1. Партия любителей пива (Россия) [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Партия_любителей_пива_(Россия)> (дата обращения 16.05.2021)

2. Руководства Oracle APEX [Электронный ресурс] URL: <https://betacode.net/10443/custom-authentication-in-oracle-apex> (дата обращения 16.05.2021)

3. Туториал по Oracle Application Express. Обзор IDE [Электронный ресурс] URL: <https://habr.com/ru/post/445128/> (дата обращения 16.05.2021)

4. Oracle PL/SQL учебник [Электронный ресурс] URL: <https://oracleplsql.ru/contents-oracle-plsql.html> (дата обращения 16.05.2021)

5. Oracle Database Online Documentation 12cRelease1(12.1) [Электронный ресурс] URL: <https://docs.oracle.com/database/121/nav/portal_4.htm> (дата обращения 16.05.2021)

# **Приложение**

**Функции БД**

**1. Pkg\_Security** – функция авторизации пользователя

Procedure Process\_Login(pUserMail PERSONS.LOGIN\_MAIL%Type, pPassword PERSONS.PASSWORD%Type, p\_App\_Id Number);

create or replace Package Body Pkg\_Security Is

--------------------------------------

Procedure Process\_Login(pUserMail PERSONS.LOGIN\_MAIL%Type, pPassword PERSONS.PASSWORD%Type, p\_App\_Id Number) As

v\_Result Boolean := true;

v\_Password PERSONS.PASSWORD%Type;

v\_Email PERSONS.LOGIN\_MAIL%Type;

Begin

-- authenticate in data base

If pUserMail Is Null Or pPassword Is Null Then

-- Write to Session, Notification must enter a username and password

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Please enter Username and password.');

v\_Result := False;

End If;

Begin

Select TRIM(TRIM(person.PASSWORD)),

TRIM(person.LOGIN\_MAIL)

Into v\_Password,

v\_Email

From PERSONS person

Where TRIM(person.LOGIN\_MAIL) = pUserMail;

Exception

When No\_Data\_Found Then

-- Write to Session, User not found.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'User not found');

v\_Result := False;

End;

IF v\_Password <> pPassword THEN

-- Write to Session, Password incorrect.

Apex\_Util.Set\_Session\_State('LOGIN\_MESSAGE'

,'Password incorrect');

v\_Result := False;

End If;

If v\_Result = True Then -- authenticate is success

-- Write user information to Session.

Apex\_Util.Set\_Session\_State('SESSION\_USER\_NAME',pUserMail);

-- Redirect to Page 1 (Home Page).

Wwv\_Flow\_Custom\_Auth\_Std.Post\_Login(pUserMail,

pPassword,

v('APP\_SESSION'), -- p\_Session\_Id

p\_App\_Id || ':1'); -- p\_Flow\_page

Else

-- Login Failure, redirect to page 101 (Login Page).

Owa\_Util.Redirect\_Url('f?p=103:101:12345');

End If;

End;

End Pkg\_Security;

**Триггеры БД**

**1. BEER\_PARTY\_ORDERS\_id\_trg** – триггер инкрементирующий первичные ключи в таблице заказов пива

create or replace trigger BEER\_PARTY\_ORDERS\_id\_trg

before insert on BEER\_PARTY\_ORDERS

for each row

begin

if :new.ORDER\_ID is null then

select BEER\_PARTY\_ORDERS\_seq.nextval into :new.ORDER\_ID from dual;

end if;

end;

**2. bi\_apex\_access\_control** – триггер сохраняющий данные о пользователях которые работали с таблицей доступа

create or replace trigger bi\_apex\_access\_control

before insert or update on apex\_access\_control

for each row

begin

if inserting and :new.id is null then

select apex\_access\_control\_seq.nextval into :new.id from sys.dual;

end if;

if inserting then

:new.created\_by := v('USER');

:new.created\_on := sysdate;

end if;

if updating then

:new.updated\_by := v('USER');

:new.updated\_on := sysdate;

end if;

end;

**3. bi\_apex\_access\_setup** – триггер срабатывающий при начальной настройке доступа и инициализирующий начальные значения

create or replace trigger bi\_apex\_access\_setup

before insert or update on apex\_access\_setup

for each row

begin

if inserting and :new.id is null then

select apex\_access\_control\_seq.nextval into :new.id from sys.dual;

end if;

if :new.application\_id is null then

:new.application\_id := v('APP\_ID');

end if;

end;

**4.** **JOIN\_PETITIONS\_id\_trg** – триггер первичных ключей таблицы JOIN\_PETITIONS

create or replace trigger JOIN\_PETITIONS\_id\_trg

before insert on JOIN\_PETITIONS

for each row

begin

if :new.PETITION\_ID is null then

select JOIN\_PETITIONS\_seq.nextval into :new.PETITION\_ID from dual;

end if;

end;

**5.** **NEWS\_id\_trg** – триггер первичных ключей таблицы JOIN\_PETITIONS

create or replace trigger NEWS\_id\_trg

before insert on NEWS

for each row

begin

if :new.NEWS\_ID is null then

select NEWS\_seq.nextval into :new.NEWS\_ID from dual;

end if;

end;

**6.** **PARTY\_CARDS\_id\_trg** – триггер первичных ключей таблицы PARTY\_CARDS

create or replace trigger PARTY\_CARDS\_id\_trg

before insert on PARTY\_CARDS

for each row

begin

if :new.CARD\_ID is null then

select PARTY\_CARDS\_seq.nextval into :new.CARD\_ID from dual;

end if;

end;

**7.** **PARTY\_PAYMENTS\_id\_trg** – триггер первичных ключей таблицы PARTY\_PAYMENTS

create or replace trigger PARTY\_PAYMENTS\_id\_trg

before insert on PARTY\_PAYMENTS

for each row

begin

if :new.PAY\_ID is null then

select PARTY\_PAYMENTS\_seq.nextval into :new.PAY\_ID from dual;

end if;

end;