**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.

# З А Д А Н И Е
## 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** | _____ (Подпись, дата) | Д.А. Миков (И.О. Фамилия) |
| **Студент** | _____ (Подпись, дата) | И.С. Марчук (И.О. Фамилия) |

Содержание

Введение.....3
Глава 1. Техническое задание .....4
1.1 Описание предметной области .....4
Глава 2. Проектирование.....5
2.1 Выбор средств реализации БД и интерфейса.....5
2.2 Описание сущностей БД .....6
2.3 Заполненные таблицы .....14
2.4 Описание функций БД.....19
2.5 Исходные коды веб приложения APEX.....20
Глава 3. Результат .....66
3.1 Интерфейс пользователя .....66
Заключение .....73
Список источников .....74
Приложение .....75

# 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. Техническое задание

**Цель:** разработать систему, для партии любителей пива 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).

# Глава 2. Проектирование

## 2.1 Выбор средств реализации БД и интерфейса

В качестве СУБД в данной работе используется ORACLE Database 11g. Это открытая, кроссплатформенная СУБД, поддерживающая PL/SQL, индексы и имеющая продвинутый конструктор приложений APEX 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:

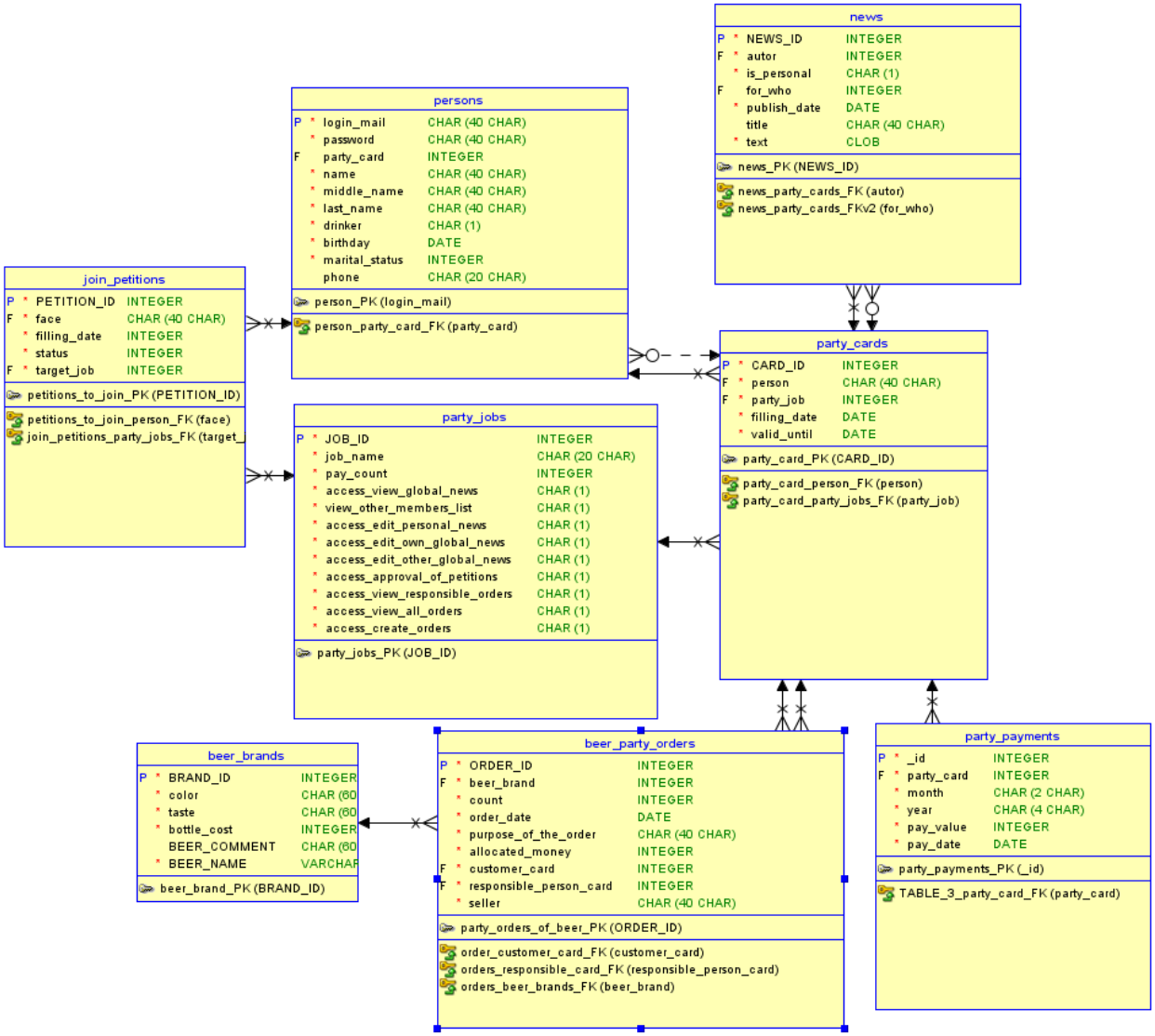



Рисунок 1 - Схема БД

## 2.2 Описание сущностей БД

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.

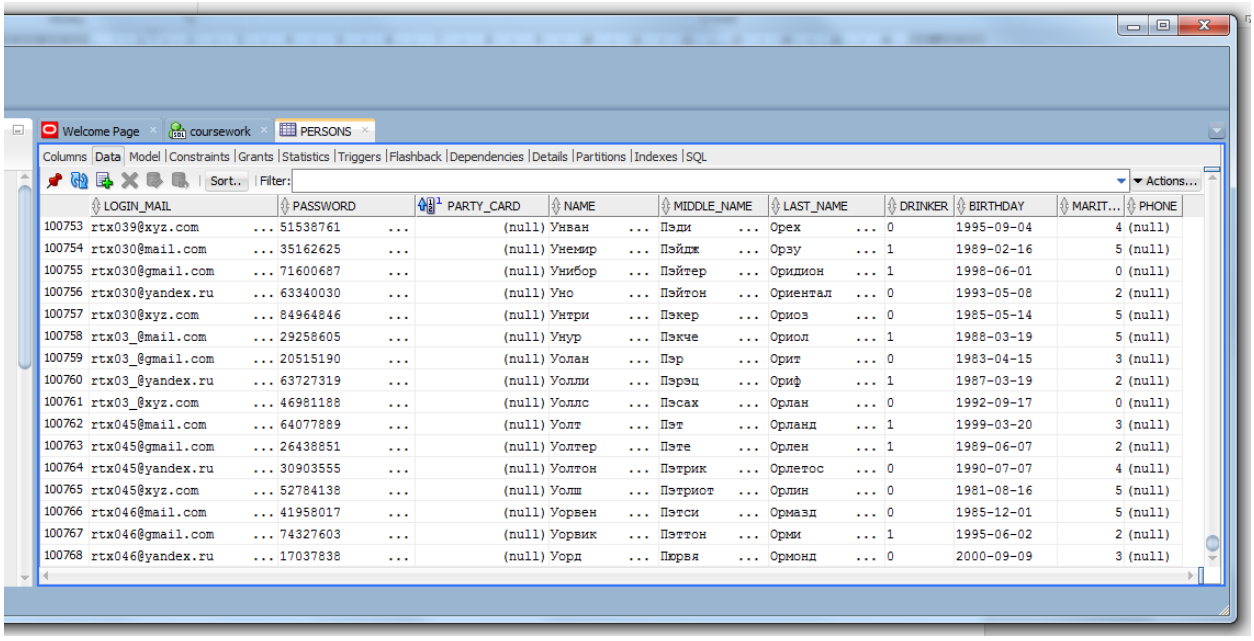
persons – члены партии

```
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.

## 2.3 Заполненные таблицы

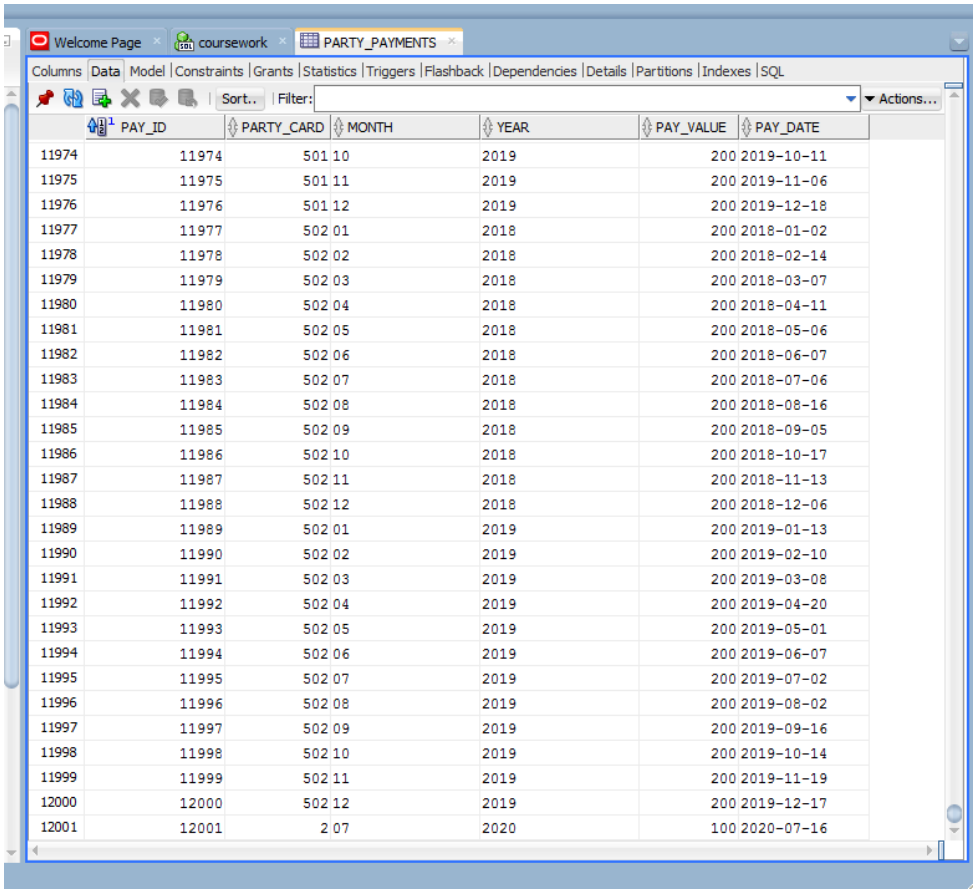
Заполненная таблица «Пользователи» на рисунке 2



| | LOGIN_MAIL | PASSWORD | PARTY_CARD | NAME | MIDDLE_NAME | LAST_NAME | DRINKER | BIRTHDAY | MARIT... | PHONE |
|---|---|---|---|---|---|---|---|---|---|---|
| 100753 | rtx039@xyz.com | 51538761 | (null) | Унван | Пэди | Орех | 0 | 1995-09-04 | 4 | (null) |
| 100754 | rtx030@mail.com | 35162625 | (null) | Унемир | Пэйдж | Орэу | 1 | 1989-02-16 | 5 | (null) |
| 100755 | rtx030@gmail.com | 71600687 | (null) | Унибор | Пэйтер | Оридион | 1 | 1998-06-01 | 0 | (null) |
| 100756 | rtx030@yandex.ru | 63340030 | (null) | Уно | Пэйтон | Ориентал | 0 | 1993-05-08 | 2 | (null) |
| 100757 | rtx030@xyz.com | 84964846 | (null) | Унтри | Пэкер | Ориоз | 0 | 1985-05-14 | 5 | (null) |
| 100758 | rtx03_@mail.com | 29258605 | (null) | Унур | Пэкче | Ориол | 1 | 1988-03-19 | 5 | (null) |
| 100759 | rtx03_@gmail.com | 20515190 | (null) | Уолан | Пэр | Орит | 0 | 1983-04-15 | 3 | (null) |
| 100760 | rtx03_@yandex.ru | 63727319 | (null) | Уолли | Пэрэц | Ориф | 1 | 1987-03-19 | 2 | (null) |
| 100761 | rtx03_@xyz.com | 46981188 | (null) | Уоллс | Пэсах | Орлан | 0 | 1992-09-17 | 0 | (null) |
| 100762 | rtx045@mail.com | 64077889 | (null) | Уолт | Пэт | Орланд | 1 | 1999-03-20 | 3 | (null) |
| 100763 | rtx045@gmail.com | 26438851 | (null) | Уолтер | Пэте | Орлен | 1 | 1989-06-07 | 2 | (null) |
| 100764 | rtx045@yandex.ru | 30903555 | (null) | Уолтон | Пэтрик | Орлетос | 0 | 1990-07-07 | 4 | (null) |
| 100765 | rtx045@xyz.com | 52784138 | (null) | Уолш | Пэтриот | Орлин | 0 | 1981-08-16 | 5 | (null) |
| 100766 | rtx046@mail.com | 41958017 | (null) | Уорвен | Пэтси | Ормазд | 0 | 1985-12-01 | 5 | (null) |
| 100767 | rtx046@gmail.com | 74327603 | (null) | Уорвик | Пэттон | Орми | 1 | 1995-06-02 | 2 | (null) |
| 100768 | rtx046@yandex.ru | 17037838 | (null) | Уорд | Пюрвя | Ормонд | 0 | 2000-09-09 | 3 | (null) |

Рисунок 2 – PERSONS

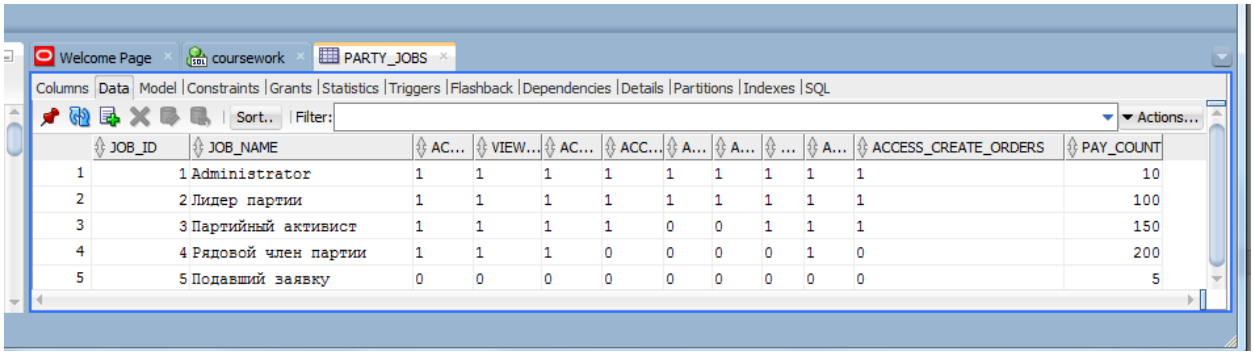
Заполненная таблица «Членские взносы» на рисунке 3



| | PAY_ID | PARTY_CARD | MONTH | YEAR | PAY_VALUE | PAY_DATE |
|---|---|---|---|---|---|---|
| 11974 | 11974 | 501 | 10 | 2019 | 200 | 2019-10-11 |
| 11975 | 11975 | 501 | 11 | 2019 | 200 | 2019-11-06 |
| 11976 | 11976 | 501 | 12 | 2019 | 200 | 2019-12-18 |
| 11977 | 11977 | 502 | 01 | 2018 | 200 | 2018-01-02 |
| 11978 | 11978 | 502 | 02 | 2018 | 200 | 2018-02-14 |
| 11979 | 11979 | 502 | 03 | 2018 | 200 | 2018-03-07 |
| 11980 | 11980 | 502 | 04 | 2018 | 200 | 2018-04-11 |
| 11981 | 11981 | 502 | 05 | 2018 | 200 | 2018-05-06 |
| 11982 | 11982 | 502 | 06 | 2018 | 200 | 2018-06-07 |
| 11983 | 11983 | 502 | 07 | 2018 | 200 | 2018-07-06 |
| 11984 | 11984 | 502 | 08 | 2018 | 200 | 2018-08-16 |
| 11985 | 11985 | 502 | 09 | 2018 | 200 | 2018-09-05 |
| 11986 | 11986 | 502 | 10 | 2018 | 200 | 2018-10-17 |
| 11987 | 11987 | 502 | 11 | 2018 | 200 | 2018-11-13 |
| 11988 | 11988 | 502 | 12 | 2018 | 200 | 2018-12-06 |
| 11989 | 11989 | 502 | 01 | 2019 | 200 | 2019-01-13 |
| 11990 | 11990 | 502 | 02 | 2019 | 200 | 2019-02-10 |
| 11991 | 11991 | 502 | 03 | 2019 | 200 | 2019-03-08 |
| 11992 | 11992 | 502 | 04 | 2019 | 200 | 2019-04-20 |
| 11993 | 11993 | 502 | 05 | 2019 | 200 | 2019-05-01 |
| 11994 | 11994 | 502 | 06 | 2019 | 200 | 2019-06-07 |
| 11995 | 11995 | 502 | 07 | 2019 | 200 | 2019-07-02 |
| 11996 | 11996 | 502 | 08 | 2019 | 200 | 2019-08-02 |
| 11997 | 11997 | 502 | 09 | 2019 | 200 | 2019-09-16 |
| 11998 | 11998 | 502 | 10 | 2019 | 200 | 2019-10-14 |
| 11999 | 11999 | 502 | 11 | 2019 | 200 | 2019-11-19 |
| 12000 | 12000 | 502 | 12 | 2019 | 200 | 2019-12-17 |
| 12001 | 12001 | 2 | 07 | 2020 | 100 | 2020-07-16 |

Рисунок 3 – PARTY_PAYMENTS

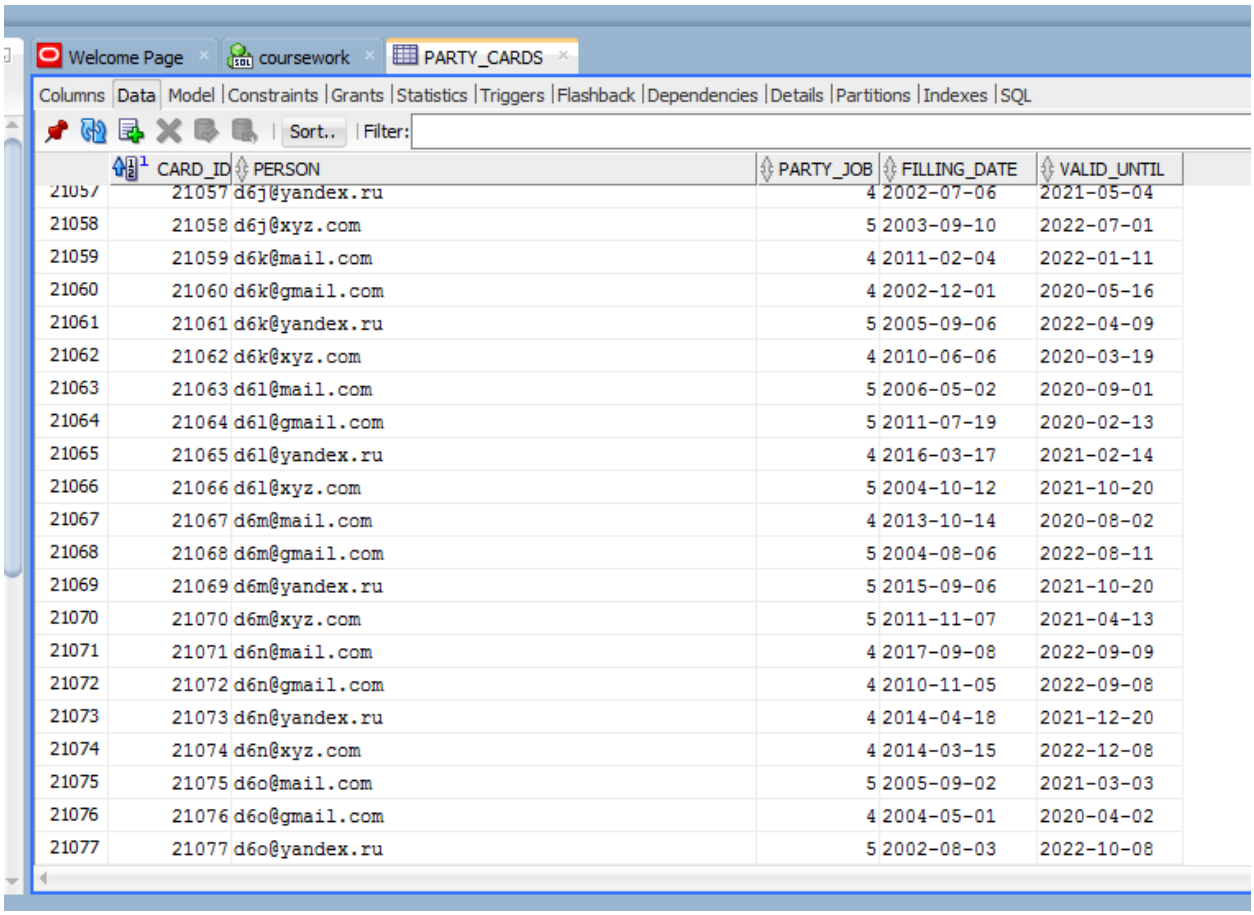
Заполненная таблица «Партийные должности» на рисунке 4



| | JOB_ID | JOB_NAME | AC... | VIEW... | AC... | ACC... | A... | A... | ... | A... | ACCESS_CREATE_ORDERS | PAY_COUNT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Administrator | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 |
| 2 | 2 | Лидер партии | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| 3 | 3 | Партийный активист | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 150 |
| 4 | 4 | Рядовой член партии | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 200 |
| 5 | 5 | Подавший заявку | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 |

Рисунок 4 – PARTY_JOBS

Заполненная таблица «Партийные билеты» на рисунке 5



| | CARD_ID | PERSON | PARTY_JOB | FILLING_DATE | VALID_UNTIL |
|---|---|---|---|---|---|
| 21057 | 21057 | d6j@yandex.ru | 4 | 2002-07-06 | 2021-05-04 |
| 21058 | 21058 | d6j@xyz.com | 5 | 2003-09-10 | 2022-07-01 |
| 21059 | 21059 | d6k@mail.com | 4 | 2011-02-04 | 2022-01-11 |
| 21060 | 21060 | d6k@gmail.com | 4 | 2002-12-01 | 2020-05-16 |
| 21061 | 21061 | d6k@yandex.ru | 5 | 2005-09-06 | 2022-04-09 |
| 21062 | 21062 | d6k@xyz.com | 4 | 2010-06-06 | 2020-03-19 |
| 21063 | 21063 | d6l@mail.com | 5 | 2006-05-02 | 2020-09-01 |
| 21064 | 21064 | d6l@gmail.com | 5 | 2011-07-19 | 2020-02-13 |
| 21065 | 21065 | d6l@yandex.ru | 4 | 2016-03-17 | 2021-02-14 |
| 21066 | 21066 | d6l@xyz.com | 5 | 2004-10-12 | 2021-10-20 |
| 21067 | 21067 | d6m@mail.com | 4 | 2013-10-14 | 2020-08-02 |
| 21068 | 21068 | d6m@gmail.com | 5 | 2004-08-06 | 2022-08-11 |
| 21069 | 21069 | d6m@yandex.ru | 5 | 2015-09-06 | 2021-10-20 |
| 21070 | 21070 | d6m@xyz.com | 5 | 2011-11-07 | 2021-04-13 |
| 21071 | 21071 | d6n@mail.com | 4 | 2017-09-08 | 2022-09-09 |
| 21072 | 21072 | d6n@gmail.com | 4 | 2010-11-05 | 2022-09-08 |
| 21073 | 21073 | d6n@yandex.ru | 4 | 2014-04-18 | 2021-12-20 |
| 21074 | 21074 | d6n@xyz.com | 4 | 2014-03-15 | 2022-12-08 |
| 21075 | 21075 | d6o@mail.com | 5 | 2005-09-02 | 2021-03-03 |
| 21076 | 21076 | d6o@gmail.com | 4 | 2004-05-01 | 2020-04-02 |
| 21077 | 21077 | d6o@yandex.ru | 5 | 2002-08-03 | 2022-10-08 |

Рисунок 5 – PARTY_CARDS

Заполненная таблица «Новости» на рисунке 6

Welcome Page | coursework | NEWS

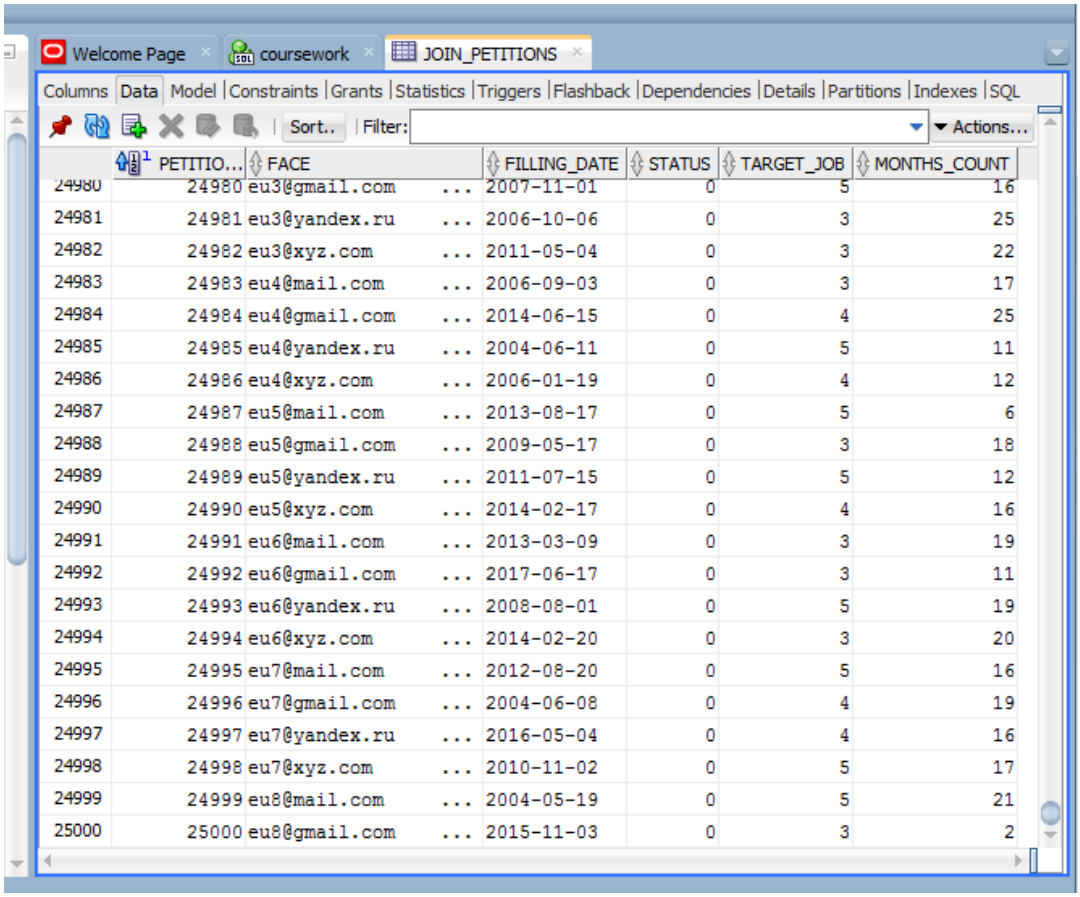
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

| | NEWS_ID | AUTOR | IS... | FOR_... | PUBLISH_DATE | TITLE | TEXT |
|---|---|---|---|---|---|---|---|
| 10960 | 10980 | 932 | 0 | (null) | 2017-08-03 | Протест ... | Также как реализация намеченных плановых заданий вынуждае... |
| 10961 | 10981 | 933 | 0 | (null) | 2017-09-01 | Идея! ... | Позвольте Вам напомнить, что рамки и место обучения кадро... |
| 10962 | 10982 | 934 | 0 | (null) | 2018-10-20 | Идеи ... | Равным образом экономическая повестка сегодняшнего дня вы... |
| 10963 | 10983 | 936 | 0 | (null) | 2017-02-07 | Про потребн... | Также как новая модель организационной деятельности требу... |
| 10964 | 10984 | 938 | 0 | (null) | 2017-06-10 | Протест ... | Товарищи! перспективное планирование требуют от нас анали... |
| 10965 | 10985 | 941 | 0 | (null) | 2018-09-20 | Идеи ... | Не следует, однако, забывать, что оптимизация основных це... |
| 10966 | 10986 | 942 | 0 | (null) | 2018-12-05 | Внедрение п... | Повседневная практика показывает, что внедрение современн... |
| 10967 | 10987 | 943 | 0 | (null) | 2018-05-15 | Изменения с... | Позвольте Вам напомнить, что постоянный количественный ро... |
| 10968 | 10988 | 946 | 0 | (null) | 2017-02-03 | Про потребн... | Таким образом внедрение современных подходов способствует... |
| 10969 | 10989 | 951 | 0 | (null) | 2017-12-01 | Направление... | Не следует, однако, забывать, что экономическая повестка ... |
| 10970 | 10990 | 953 | 0 | (null) | 2018-03-14 | Идея! ... | Таким образом постоянный количественный рост и сфера наше... |
| 10971 | 10991 | 955 | 0 | (null) | 2017-10-11 | Про потребн... | С другой стороны рамки и место обучения кадров играет опр... |
| 10972 | 10992 | 956 | 0 | (null) | 2018-11-16 | Про потребн... | В целом, конечно, экономическая повестка сегодняшнего дня... |
| 10973 | 10993 | 963 | 0 | (null) | 2017-10-17 | Про потребн... | Не следует, однако, забывать, что перспективное планирова... |
| 10974 | 10994 | 964 | 0 | (null) | 2017-06-18 | Внедрение п... | Уважаемые коллеги! экономическая повестка сегодняшнего дн... |
| 10975 | 10995 | 968 | 0 | (null) | 2017-08-02 | О новой мод... | Равным образом реализация намеченных плановых заданий тре... |
| 10976 | 10996 | 969 | 0 | (null) | 2018-02-09 | Идея! ... | Не следует, однако, забывать, что постоянный количественн... |
| 10977 | 10997 | 972 | 0 | (null) | 2018-11-03 | Внедрение п... | Повседневная практика показывает, что рамки и место обуче... |
| 10978 | 10998 | 973 | 0 | (null) | 2018-07-09 | Направление... | В целом, конечно, экономическая повестка сегодняшнего дня... |
| 10979 | 10999 | 975 | 0 | (null) | 2017-02-15 | Изменения с... | Повседневная практика показывает, что экономическая повес... |
| 10980 | 11000 | 977 | 0 | (null) | 2018-11-06 | Идея! ... | Повседневная практика показывает, что постоянный количест... |

Рисунок 6 – NEWS

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



| | PETITIO... | FACE | | FILLING_DATE | STATUS | TARGET_JOB | MONTHS_COUNT |
|---|---|---|---|---|---|---|---|
| 24980 | 24980 | eu3@gmail.com | ... | 2007-11-01 | 0 | 5 | 16 |
| 24981 | 24981 | eu3@yandex.ru | ... | 2006-10-06 | 0 | 3 | 25 |
| 24982 | 24982 | eu3@xyz.com | ... | 2011-05-04 | 0 | 3 | 22 |
| 24983 | 24983 | eu4@mail.com | ... | 2006-09-03 | 0 | 3 | 17 |
| 24984 | 24984 | eu4@gmail.com | ... | 2014-06-15 | 0 | 4 | 25 |
| 24985 | 24985 | eu4@yandex.ru | ... | 2004-06-11 | 0 | 5 | 11 |
| 24986 | 24986 | eu4@xyz.com | ... | 2006-01-19 | 0 | 4 | 12 |
| 24987 | 24987 | eu5@mail.com | ... | 2013-08-17 | 0 | 5 | 6 |
| 24988 | 24988 | eu5@gmail.com | ... | 2009-05-17 | 0 | 3 | 18 |
| 24989 | 24989 | eu5@yandex.ru | ... | 2011-07-15 | 0 | 5 | 12 |
| 24990 | 24990 | eu5@xyz.com | ... | 2014-02-17 | 0 | 4 | 16 |
| 24991 | 24991 | eu6@mail.com | ... | 2013-03-09 | 0 | 3 | 19 |
| 24992 | 24992 | eu6@gmail.com | ... | 2017-06-17 | 0 | 3 | 11 |
| 24993 | 24993 | eu6@yandex.ru | ... | 2008-08-01 | 0 | 5 | 19 |
| 24994 | 24994 | eu6@xyz.com | ... | 2014-02-20 | 0 | 3 | 20 |
| 24995 | 24995 | eu7@mail.com | ... | 2012-08-20 | 0 | 5 | 16 |
| 24996 | 24996 | eu7@gmail.com | ... | 2004-06-08 | 0 | 4 | 19 |
| 24997 | 24997 | eu7@yandex.ru | ... | 2016-05-04 | 0 | 4 | 16 |
| 24998 | 24998 | eu7@xyz.com | ... | 2010-11-02 | 0 | 5 | 17 |
| 24999 | 24999 | eu8@mail.com | ... | 2004-05-19 | 0 | 5 | 21 |
| 25000 | 25000 | eu8@gmail.com | ... | 2015-11-03 | 0 | 3 | 2 |

Рисунок 7 – JOIN_PETITIONS

Заполненная таблица «Партийные заказы» на рисунке 8

Welcome Page | coursework | BEER_PARTY_ORDERS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter: | Actions...

| | ORDER_ID | BEER_BRAND | COUNT | ORDER_DATE | PURPOSE_OF_THE_ORDER | ALLOCAT... | CUSTOME... | RESPONS... | SELLER |
|---|---|---|---|---|---|---|---|---|---|
| 10960 | 10920 | 1373 | 21 | 2017-05-04 | Личный заказ ... | 4200 | 10901 | 7 | 000 Калуга |
| 10961 | 10921 | 8227 | 38 | 2018-02-20 | Дегустация ... | 7600 | 10902 | 8 | 000 Калуга |
| 10962 | 10922 | 965 | 16 | 2018-02-14 | Личный заказ ... | 3200 | 10903 | 9 | 000 Журавли |
| 10963 | 10923 | 5581 | 9 | 2017-07-11 | Личный заказ ... | 1800 | 10904 | 10 | Ларек |
| 10964 | 10924 | 5172 | 23 | 2018-03-04 | Дегустация ... | 4600 | 10905 | 3 | Ликероводочный |
| 10965 | 10925 | 10076 | 19 | 2017-11-04 | Личный заказ ... | 3800 | 10906 | 4 | Ликероводочный |
| 10966 | 10926 | 4324 | 12 | 2018-12-02 | Дегустация ... | 2400 | 10907 | 5 | Ликероводочный |
| 10967 | 10927 | 8978 | 19 | 2018-04-11 | Личный заказ ... | 3800 | 10908 | 6 | Ликероводочный |
| 10968 | 10928 | 3601 | 24 | 2017-03-19 | Дегустация ... | 4800 | 10909 | 7 | Ларек |
| 10969 | 10929 | 9417 | 16 | 2018-03-01 | Праздник ... | 3200 | 10910 | 8 | Ликероводочный |
| 10970 | 10930 | 5386 | 24 | 2018-01-09 | Подарок от партии ... | 4800 | 10911 | 9 | 000 Журавли |
| 10971 | 10931 | 9021 | 5 | 2018-04-01 | Личный заказ ... | 1000 | 10912 | 10 | 000 Калуга |
| 10972 | 10932 | 1826 | 6 | 2018-12-17 | Дегустация ... | 1200 | 10913 | 3 | Ликероводочный |
| 10973 | 10933 | 9534 | 2 | 2018-09-13 | Праздник ... | 400 | 10914 | 4 | 000 Калуга |
| 10974 | 10934 | 216 | 19 | 2017-10-08 | Личный заказ ... | 3800 | 10915 | 5 | Ликероводочный |
| 10975 | 10935 | 8828 | 20 | 2017-08-04 | Праздник ... | 4000 | 10916 | 6 | Ларек |
| 10976 | 10936 | 3882 | 9 | 2018-11-19 | Дегустация ... | 1800 | 10917 | 7 | Ларек |
| 10977 | 10937 | 4027 | 9 | 2017-09-05 | Праздник ... | 1800 | 10918 | 8 | Ларек |
| 10978 | 10938 | 8299 | 37 | 2017-06-12 | Дегустация ... | 7400 | 10919 | 9 | Ликероводочный |
| 10979 | 10939 | 7292 | 23 | 2017-12-04 | Личный заказ ... | 4600 | 10920 | 10 | Ларек |
| 10980 | 10940 | 10179 | 4 | 2018-12-20 | Праздник ... | 800 | 10921 | 3 | Ликероводочный |

Рисунок 8 – BEER_PARTY_ORDERS

Заполненная таблица «Марки пива» на рисунке 9

Welcome Page | coursework | BEER_BRANDS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter: | Actions...

| | BRAN... | COLOR | TASTE | BOTTLE_COST | BEER_COMMENT | BEER_NAME |
|---|---|---|---|---|---|---|
| 10280 | 10280 | Имперский Стаут ... | Карамель ... | 100 | Одобрено ... | Киек |
| 10281 | 10281 | Пейл лагер ... | Кислотность ... | 100 | Не очень ... | Кизаши |
| 10282 | 10282 | Майский Бок ... | Кислый вкус ... | 100 | Не очень ... | Кийам |
| 10283 | 10283 | Пшеничное пиво ... | Кофе ... | 100 | Одобрено ... | Кийоши |
| 10284 | 10284 | Эль из светлого солода ... | Пряности ... | 100 | Не очень ... | Кикос |
| 10285 | 10285 | Пейл-Эль ... | Гвоздика ... | 110 | Не очень ... | Кил |
| 10286 | 10286 | Партийное ... | Пшеница ... | 110 | Не очень ... | Килвин |
| 10287 | 10287 | Английский Пейл-Эль ... | Резкость ... | 110 | Одобрено ... | Килдияр |
| 10288 | 10288 | Индийский Пейл-Эль ... | Рожь ... | 110 | Одобрено ... | Киллиан |
| 10289 | 10289 | Тёмный лагер ... | Сахар ... | 110 | Одобрено ... | Килсен |
| 10290 | 10290 | Браун Эль ... | Сидр ... | 110 | Не очень ... | Киль |
| 10291 | 10291 | Доппельбок ... | Сладость ... | 120 | Одобрено ... | Кильбарс |
| 10292 | 10292 | Стаут ... | Солод ... | 120 | Не очень ... | Кильде |
| 10293 | 10293 | Балтийский Портер ... | Специи ... | 120 | Одобрено ... | Кильдеяр |
| 10294 | 10294 | Имперский Стаут ... | Сухость ... | 120 | Плохое ... | Кильдураз |
| 10295 | 10295 | Пейл лагер ... | Землистость ... | 120 | Не очень ... | Килябай |
| 10296 | 10296 | Майский Бок ... | Терпкость ... | 120 | Одобрено ... | Кимал |
| 10297 | 10297 | Пшеничное пиво ... | Фенолы ... | 120 | На любителя... | Кимболл |
| 10298 | 10298 | Эль из светлого солода ... | Фрукты ... | 130 | Одобрено ... | Кимсан |
| 10299 | 10299 | Пейл-Эль ... | Хвоя ... | 130 | Не очень ... | Кимьё |
| 10300 | 10300 | Партийное ... | крекеры ... | 130 | Одобрено ... | Кимюд |

Рисунок 9 – BEER_BRANDS

## 2.4 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.

## 2.5 Исходные коды веб приложения APEX

**Страница входа**

PageProcessing: **SetUserNameCookie** – помещает ссылку на текущего пользователя в куки браузера

```
apex_authentication.send_login_username_cookie(
p_username=>lower(:P101_USERNAME)
);
```

PageProcessing: **Login** – вызывает из БД функцию авторизации, сама вызывается по нажатию кнопки «Войти»

```
Pkg_Security.Process_Login(
  :P101_USERNAME,
  :P101_PASSWORD,
  :APP_ID
);
```

**Страница регистрации**

DynamicAction(PL/SQL): **CreatePress** – вызывается при нажатии на кнопку «создать учетную запись»

```
declare
   p_count       INTEGER(10);
   answer_string  VARCHAR2(300);
   error_flag     boolean;
begin
  error_flag := false;

  -- логин
  if(TRIM(:P102_MAIL) IS NULL)then
    answer_string := answer_string || 'Логин не может быть пустым; ';
    error_flag := true;
```

```
end if;

SELECT
  COUNT(*)
INTO
  p_count
FROM PERSONS p_persons
WHERE TRIM(p_persons.LOGIN_MAIL) = TRIM(:P102_MAIL);

--answer_string := answer_string || to_char(p_count);

if(p_count <> 0)then
  answer_string := answer_string || 'Такой логин уже занят; ';
  error_flag := true;
end if;

-- пароль
if(TRIM(:P102_PASSWORD) IS NULL)then
  answer_string := answer_string || 'Поле Пароль не может быть пустым; ';
  error_flag := true;
end if;

   -- эта проверка выдает false, если хотябы один из операндов null
if(TRIM(:P102_PASSWORD) <> TRIM(:P102_PASSWORD_CONFIRM))then
  answer_string := answer_string || 'Пароли должны совпадать; ';
  error_flag := true;
end if;

-- Имя
if(TRIM(:P102_NAME) IS NULL)then
```

```
  answer_string := answer_string || 'Поле Имя не может быть пустым; ';
  error_flag := true;
end if;

-- Отчество
if(TRIM(:P102_MIDDLE_NAME) IS NULL)then
  answer_string := answer_string || 'Поле Отчество не может быть пустым; ';
  error_flag := true;
end if;

-- Фамилия
if(TRIM(:P102_LAST_NAME) IS NULL)then
  answer_string := answer_string || 'Поле Фамилия не может быть пустым; ';
  error_flag := true;
end if;

-- День рождения
if(TRIM(:P102_BIRTHDAY) IS NULL)then
  answer_string := answer_string || 'Поле День рождения не может быть пустым;
';
  error_flag := true;
end if;
-- создание аккаунта
if(not error_flag)then
 INSERT INTO "PERSONS" (
   LOGIN_MAIL,
   PASSWORD,
   NAME,
   MIDDLE_NAME,
   LAST_NAME,
```

```
    DRINKER,
    BIRTHDAY,
    MARITAL_STATUS,
    PHONE
  ) VALUES (
    TRIM(:P102_MAIL),
    TRIM(:P102_PASSWORD),
    TRIM(:P102_NAME),
    TRIM(:P102_MIDDLE_NAME),
    TRIM(:P102_LAST_NAME),
    TRIM(:P102_DRINKER),
    TRIM(:P102_BIRTHDAY),
    TRIM(:P102_MATRIAL_STATUS),
    TRIM(:P102_PHONE)

  );
  return 'Проверка пройдена успешно. Создаем аккаунт...';
 end if;
 return answer_string;
end;
```

**Персональная страница**

SQL Query: **P_MAIL** – запрос получения логина пользователя

```
select
 PERSONS.LOGIN_MAIL as "Mail"
 from PERSONS PERSONS
 where ( APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME') = TRIM(PERSONS.LOGIN_MAIL))
```

PL/SQL: **P1_NAME** – запрос в базу данных для получения имени пользователя

```
declare
   lastName    PERSONS.LAST_NAME%Type;
        name        PERSONS.NAME%Type;
        middleName  PERSONS.MIDDLE_NAME%Type;
begin
  Begin
   Select
     TRIM(person.LAST_NAME),
     TRIM(person.NAME),
     TRIM(person.MIDDLE_NAME)
   Into   lastName,
                  name,
         middleName
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
   return lastName||' '||name||' '||middleName;
  Exception
   When No_Data_Found Then
     return '';
  End;
end;
```

PL/SQL: **P1_DRINKER**– запрос в базу данных для получения статуса отношения к алкоголю у пользователя

```
declare
   drinker    PERSONS.DRINKER%Type;
begin
```

```
  Begin
    Select
      TRIM(person.DRINKER)
    Into
      drinker
    From   PERSONS person
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);

    if(drinker = 0)then
      return 'Не пьющий';
    else
      return 'Любитель выпить';
    end if;

  Exception
    When No_Data_Found Then
      return '';
  End;
end;
```

PL/SQL: **P1_BIRTHDAY**– запрос в базу данных для получения даты рождения пользователя

```
select
    PERSONS.BIRTHDAY
 from PERSONS PERSONS
 where   (   APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(PERSONS.LOGIN_MAIL))
```

PL/SQL: **P1_MARITAL_STATUS** – запрос в базу данных для получения статуса отношений пользователя

```
declare
   status    PERSONS.MARITAL_STATUS%Type;
begin
  Begin
    Select
      person.MARITAL_STATUS
    Into
      status
    From   PERSONS person
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
    return
    CASE status
      WHEN 1 THEN 'Не состою в браке'
      WHEN 2 THEN 'Встречаюсь'
      WHEN 3 THEN 'Помолвлен(а)'
      WHEN 4 THEN 'Женат/Замужем'
      WHEN 5 THEN 'В гражданском браке'
      WHEN 6 THEN 'Влюблен(а)'
      WHEN 7 THEN 'Все сложно'
      WHEN 8 THEN 'В активном поиске'
      ELSE 'Не выбрано'
    END;
  Exception
    When No_Data_Found Then
      return '';
  End;
end;
```

PL/SQL: **P1_PHONE** – запрос в базу данных для получения номера телефона пользователя

```
select
   PERSONS.PHONE
 from PERSONS PERSONS
 where ( APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME') = TRIM(PERSONS.LOGIN_MAIL))
```

PL/SQL: **P1_PARTYCARD_ID** – запрос в базу данных для получения номера партийного билета пользователя

```
declare
   card_id    PERSONS.PARTY_CARD%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     card_id
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME') = TRIM(person.LOGIN_MAIL);
   if(card_id IS NULL)then
     return 'У вас пока нет парт. билета';
   else
     return card_id;
   end if;
  Exception
   When No_Data_Found Then
```

```
    return '';
  End;
end;
```

PL/SQL: **P1_FILLING_DATE** – запрос в базу данных для получения даты оформления партийного билета пользователя

```
declare
   card_id_p      PERSONS.PARTY_CARD%Type;
   card_date_p    PARTY_CARDS.FILLING_DATE%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     card_id_p
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
   if(card_id_p IS NULL) then
     return '';
   end if;
   Select
     PARTY_CARDS.FILLING_DATE
   Into
     card_date_p
   From PARTY_CARDS
   Where PARTY_CARDS.CARD_ID = card_id_p;
   if(card_date_p IS NULL)then
     return '';
   else
```

```
      return card_date_p;
    end if;
  Exception
    When No_Data_Found Then
      return '';
  End;
end;
```

PL/SQL: **P1_VALID_UNTIL** – запрос в базу данных для получения даты действия партийного билета пользователя

```
declare
    card_id_p      PERSONS.PARTY_CARD%Type;
    card_date_p    PARTY_CARDS.FILLING_DATE%Type;
begin
  Begin
    Select
      person_p.PARTY_CARD
    Into
      card_id_p
    From   PERSONS person_p
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person_p.LOGIN_MAIL);
    Select
      p_card.VALID_UNTIL
    Into
      card_date_p
    From   PARTY_CARDS p_card
    Where card_id_p = p_card.CARD_ID;
          if(card_id_p IS NULL) then
```

```
      return '';
    end if;
    if(card_date_p IS NULL)then
      return '';
    else
      if(sysdate > card_date_p) then
        return card_date_p||', not valid';
      else
        return card_date_p||', valid';
      end if;
    end if;
  Exception
    When No_Data_Found Then
      return '';
  End;
end;
```

PL/SQL: **P1_JOB_NAME** – запрос в базу данных для получения работы указанной в партийном билете пользователя

```
declare
    p_card_id      PERSONS.PARTY_CARD%Type;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    p_job_name     PARTY_JOBS.JOB_NAME%Type;
begin
  Begin
    Select
      person.PARTY_CARD
    Into
      p_card_id
    From   PERSONS person
```

```
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
    if(p_card_id IS NULL) then
      return '';
    end if;
    Select
      p_card.PARTY_JOB
    Into
      p_job_id
    From   PARTY_CARDS p_card
    Where p_card_id = p_card.CARD_ID;
    if(p_job_id IS NULL)then
      return '';
    end if;
    Select
      jobs.JOB_NAME
    Into
      p_job_name
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;
    if(p_job_name IS NULL)then
      return '';
    else
      return p_job_name;
    end if;
  Exception
    When No_Data_Found Then
      return '';
  End;
end;
```

PL/SQL: **P1_PAY_COUNT** – запрос в базу данных для получения размера членского взноса пользователя

```
declare
   p_card_id      PERSONS.PARTY_CARD%Type;
   p_job_id       PARTY_CARDS.PARTY_JOB%Type;
   p_pay_count    PARTY_JOBS.PAY_COUNT%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     p_card_id
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
   if(p_card_id IS NULL) then
     return '';
   end if;
   Select
     p_card.PARTY_JOB
   Into
     p_job_id
   From   PARTY_CARDS p_card
   Where p_card_id = p_card.CARD_ID;

   if(p_job_id IS NULL)then
     return '';
   end if;
```

```
    Select
      p_job.PAY_COUNT
    Into
      p_pay_count
    From   PARTY_JOBS p_job
    Where p_job_id = p_job.JOB_ID;
    if(p_pay_count IS NULL)then
      return '';
    else
      return p_pay_count;
    end if;
  Exception
    When No_Data_Found Then
      return '';
  End;
end;
```

PL/SQL: **P1_PAY_STATUS** – запрос в базу данных для получения статуса выплаты членского взноса пользователя в этом месяце

```
declare
    p_pays_count      integer;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    p_pay_count    PARTY_JOBS.PAY_COUNT%Type;
begin
    select
      count(PARTY_PAYMENTS.PAY_ID)
    INTO
      p_pays_count
    FROM
```

```
    PARTY_PAYMENTS
     JOIN PARTY_CARDS ON PARTY_CARDS.CARD_ID = PARTY_PAYMENTS.PARTY_CARD
     JOIN PARTY_JOBS ON PARTY_JOBS.JOB_ID = PARTY_CARDS.PARTY_JOB
   WHERE
    PARTY_PAYMENTS.YEAR = SUBSTR(TO_CHAR(SYSDATE), 1, 4) AND
     PARTY_PAYMENTS.MONTH = SUBSTR(TO_CHAR(SYSDATE), 6, 2) AND
     PAY_VALUE >= PARTY_JOBS.PAY_COUNT;

   if(p_pays_count > 0)then
    RETURN 'Этот месяц оплачен';
   else
    RETURN 'Этот месяц не оплачен';
   end if;
end;
```

SQL запрос: **Таблица заявок, поданных пользователем** – запрос в базу данных для получения нерассмотренных заявок отправленных этим пользователем

```
select
     j_petitions.PETITION_ID AS "Номер петиции",
     j_petitions.FILLING_DATE AS "Дата оформления",
     PARTY_JOBS.JOB_NAME AS "Должность",
     j_petitions.MONTHS_COUNT || ' месяцев' AS "Время продления",
     CASE j_petitions.STATUS
       WHEN 1 THEN 'Одобрено'
       WHEN 2 THEN 'Отвергнуто'
       ELSE 'На рассмотрении'
```

```
    END
      AS "Статус"
  from JOIN_PETITIONS j_petitions
    JOIN PARTY_JOBS ON PARTY_JOBS.JOB_ID = j_petitions.TARGET_JOB
  WHERE trim(j_petitions.FACE) =
TRIM(APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME'))
```

DynamicAction(PL/SQL): **HidePayFields** – отрабатывает при загрузке страницы, и скрывает поле оплаты членского взноса, если этот месяц оплачен

```
declare
   p_pays_count      integer;
   card_id    PERSONS.PARTY_CARD%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     card_id
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
   if(card_id IS NULL)then
     return true;
   end if;
  Exception
   When No_Data_Found Then
     return true;
  End;
   select
     count(PARTY_PAYMENTS.PAY_ID)
```

```
  INTO
    p_pays_count
  FROM
    PARTY_PAYMENTS
      JOIN PARTY_CARDS ON PARTY_CARDS.CARD_ID =
PARTY_PAYMENTS.PARTY_CARD
      JOIN PARTY_JOBS ON PARTY_JOBS.JOB_ID =
PARTY_CARDS.PARTY_JOB
  WHERE
    PARTY_PAYMENTS.YEAR = SUBSTR(TO_CHAR(SYSDATE), 1, 4) AND
      PARTY_PAYMENTS.MONTH = SUBSTR(TO_CHAR(SYSDATE), 6, 2)
AND
      PAY_VALUE >= PARTY_JOBS.PAY_COUNT;
  if(p_pays_count > 0)then
    RETURN true;
  else
    RETURN false;
  end if;
end;
```

DynamicAction(PL/SQL): **Filling** – при нажатии кнопки обозначающей подачу заявления на изменение или продление парт билета, эта программа проверяет введенные поля, и сохраняет данные о запросе в базу.

```
declare
begin
  if(to_number(:P1_MONTHS_COUNT)>=0
AND to_number(:P1_MONTHS_COUNT) <= 60) then
      INSERT INTO "JOIN_PETITIONS" (
         PETITION_ID,
         FACE,
```

```
            FILLING_DATE,
            STATUS,
            TARGET_JOB,
            MONTHS_COUNT
        ) VALUES (
            null,
            trim(APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')),
            SYSTIMESTAMP,
            0,
            :P1_JOB,
            :P1_MONTHS_COUNT
        );
        return 'Заявка успешно отправлена';
    else
     return 'Количество месяцев должно быть в пределах от 0, до 60';
    end if;
end;
```

DynamicAction(PL/SQL):**PayButonPress** – при нажатии кнопки оплатить членский взнос, программа проверяет введенные данные, а затем сохраняет данные об оплате в базу.

```
declare
    p_card_id      PERSONS.PARTY_CARD%Type;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    p_pay_count    PARTY_JOBS.PAY_COUNT%Type;
Begin
  Begin
    Select
      person.PARTY_CARD
    Into
```

```
    p_card_id
  From   PERSONS person
  Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);

  if(p_card_id IS NULL) then
    return 'Парт билет не найден!';
  end if;
  Select
    p_card.PARTY_JOB
  Into
    p_job_id
  From   PARTY_CARDS p_card
  Where p_card_id = p_card.CARD_ID;

  if(p_job_id IS NULL)then
    return 'Не найдена должность';
  end if;
   Select
    p_job.PAY_COUNT
  Into
    p_pay_count
  From   PARTY_JOBS p_job
  Where p_job_id = p_job.JOB_ID;

  if(p_pay_count IS NULL)then
    p_pay_count := 0;
  end if;
 Exception
  When No_Data_Found Then
```

```
      return 'Не найден парт билет';
  End;
  if(:P1_PAY_VALUE < p_pay_count)then
    return 'Размер выплаты не может быть меньше чем размер членского
взноса!';
  end if;
  if(:P1_PAY_FIELD_CARD IS NULL)then
    return 'Необходимо ввести номер карты для оплаты';
  end if;
 INSERT INTO "KR_ADMIN"."PARTY_PAYMENTS" (
   PAY_ID,
   PARTY_CARD,
   MONTH,
   YEAR,
   PAY_VALUE,
   PAY_DATE
 ) VALUES (
   null,
   p_card_id,
   SUBSTR(TO_CHAR(SYSDATE), 6, 2),
   SUBSTR(TO_CHAR(SYSDATE), 1, 4),
   p_pay_count,
   SYSTIMESTAMP
 );
  :P1_PAY_FIELD_CARD := '';
  return 'Оплата прошла успешно!';
end;
```

**Страница с новостями**

SQL запрос: **Таблица GlobalNewsPage** – запрос в базу данных для получения глобальных новостей

```
select
    NEWS.PUBLISH_DATE as "Дата публикации",
    PERSONS.LAST_NAME || PERSONS.NAME ||'('||
PERSONS.LOGIN_MAIL || ')' as "Автор",
    NEWS.TITLE as "Заголовок",
    NEWS.TEXT as "Текст"
FROM NEWS
  JOIN PARTY_CARDS ON NEWS.AUTOR=PARTY_CARDS.CARD_ID
  JOIN PERSONS ON
PARTY_CARDS.PERSON=PERSONS.LOGIN_MAIL
```

SQL запрос: **Таблица Заявления** – запрос в базу данных для получения всех не рассмотренных заявлений пользователей, на изменение или продление парт билета

```
select
   JOIN_PETITIONS.PETITION_ID as "Изменить",
   JOIN_PETITIONS.PETITION_ID as "Номер заявления",
   JOIN_PETITIONS.FACE as "Подавшее лицо",
   JOIN_PETITIONS.FILLING_DATE as "Дата подачи",
   PARTY_JOBS.JOB_NAME as "На должность",
   JOIN_PETITIONS.MONTHS_COUNT || ' месяцев' as "Продление на"
FROM JOIN_PETITIONS
 JOIN PARTY_JOBS ON JOIN_PETITIONS.TARGET_JOB =
PARTY_JOBS.JOB_ID
WHERE JOIN_PETITIONS.STATUS = 0
```

SQL запрос: **Таблица партийные выплаты** – запрос в базу данных для получения сведений о всех членских взносах пользователей

```
select
   PARTY_PAYMENTS.PAY_ID as "Уник. номер",
   PERSONS.LOGIN_MAIL || '(' || PERSONS.NAME || ' ' || PERSONS.MIDDLE_NAME || ' ' || PERSONS.LAST_NAME || ')' as "Владелец билета",
   PARTY_PAYMENTS.YEAR || '-' || PARTY_PAYMENTS.MONTH as "Дата месяца",
   PARTY_PAYMENTS.PAY_VALUE as "Сумма оплаты",
   PARTY_PAYMENTS.PAY_DATE as "Дата оплаты"
FROM PARTY_PAYMENTS
 JOIN PERSONS ON (PARTY_PAYMENTS.PARTY_CARD = PERSONS.PARTY_CARD)
```

DynamicAction(PL/SQL):**HideNews** – при отсутствии у пользователя прав на чтение новостей, программа скрывает блок «новости»

```
declare
   p_card_id      PERSONS.PARTY_CARD%Type;
   p_job_id       PARTY_CARDS.PARTY_JOB%Type;
   p_job_access   PARTY_JOBS.ACCESS_VIEW_GLOBAL_NEWS%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     p_card_id
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME') = TRIM(person.LOGIN_MAIL);
```

```
    if(p_card_id IS NULL) then
      return true;
    end if;
    Select
      p_card.PARTY_JOB
    Into
      p_job_id
    From   PARTY_CARDS p_card
    Where p_card_id = p_card.CARD_ID;
    if(p_job_id IS NULL)then
      return true;
    end if;
    Select
      jobs.ACCESS_VIEW_GLOBAL_NEWS
    Into
      p_job_access
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;
    if(p_job_access = 0)then
      return true;
    else
      return false;
    end if;
  Exception
    When No_Data_Found Then
      return true;
  End;
end;
```

DynamicAction(PL/SQL):**HideNotification** – если у пользователя есть разрешение на просмотр новостей партии, то программа скроет стоящее по умолчанию сообщение «у вас нет доступа к новостям»

```
declare
   p_card_id      PERSONS.PARTY_CARD%Type;
   p_job_id       PARTY_CARDS.PARTY_JOB%Type;
   p_job_access   PARTY_JOBS.ACCESS_VIEW_GLOBAL_NEWS%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     p_card_id
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
   if(p_card_id IS NULL) then
     return false;
   end if;
   Select
     p_card.PARTY_JOB
   Into
     p_job_id
   From   PARTY_CARDS p_card
   Where p_card_id = p_card.CARD_ID;
   if(p_job_id IS NULL)then
     return false;
   end if;
   Select
     jobs.ACCESS_VIEW_GLOBAL_NEWS
```

```
    Into
      p_job_access
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;
    if(p_job_access = 0)then
      return false;
    else
      return true;
    end if;
  Exception
    When No_Data_Found Then
      return false;
  End;
end;
```

DynamicAction(PL/SQL):**HideCreateNews** – если у пользователя нет прав на создание собственных новостей, программа скроет блок отвечающий за это

```
declare
    p_card_id      PERSONS.PARTY_CARD%Type;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    p_job_access
PARTY_JOBS.ACCESS_EDIT_OWN_GLOBAL_NEWS%Type;
begin
  Begin
    Select
      person.PARTY_CARD
    Into
      p_card_id
    From   PERSONS person
```

```
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
    if(p_card_id IS NULL) then
      return true;
    end if;
    Select
      p_card.PARTY_JOB
    Into
      p_job_id
    From   PARTY_CARDS p_card
    Where p_card_id = p_card.CARD_ID;
    if(p_job_id IS NULL)then
      return true;
    end if;
    Select
      jobs.ACCESS_EDIT_OWN_GLOBAL_NEWS
    Into
      p_job_access
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;

    if(p_job_access = 0)then
      return true;
    else
      return false;
    end if;
  Exception
    When No_Data_Found Then
      return true;
  End;
```

```
end;
```

DynamicAction(PL/SQL): **HidePetitions** – если у пользователя нет прав на разбор петиций других пользователей, то программа скроет это сообщение

```
declare
   p_card_id      PERSONS.PARTY_CARD%Type;
   p_job_id       PARTY_CARDS.PARTY_JOB%Type;
   p_job_access   PARTY_JOBS.ACCESS_VIEW_GLOBAL_NEWS%Type;
begin
  Begin
   Select
     person.PARTY_CARD
   Into
     p_card_id
   From   PERSONS person
   Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
   if(p_card_id IS NULL) then
     return true;
   end if;
   Select
     p_card.PARTY_JOB
   Into
     p_job_id
   From   PARTY_CARDS p_card
   Where p_card_id = p_card.CARD_ID;

   if(p_job_id IS NULL)then
     return true;
   end if;
```

```
    Select
      jobs.ACCESS_APPROVAL_OF_PETITIONS
    Into
      p_job_access
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;
    if(p_job_access = 0)then
      return true;
    else
      return false;
    end if;
  Exception
    When No_Data_Found Then
      return true;
  End;
end;
```

DynamicAction(PL/SQL): **Create_New** – отрабатывает при нажатии на кнопку «опубликовать новость»

```
declare
    p_card_id      PERSONS.PARTY_CARD%Type;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    access         PARTY_JOBS.ACCESS_EDIT_OWN_GLOBAL_NEWS%Type;
begin
  Begin
    Select
      person.PARTY_CARD
    Into
      p_card_id
    From   PERSONS person
```

```
Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
  if(p_card_id IS NULL) then
    return 'Не найден ваш парт. билет!';
  end if;
  Select
    p_card.PARTY_JOB
  Into
    p_job_id
  From   PARTY_CARDS p_card
  Where p_card_id = p_card.CARD_ID;
  if(p_job_id IS NULL)then
    return 'Не найдена партийная должность!';
  end if;
   Select
    p_job.ACCESS_EDIT_OWN_GLOBAL_NEWS
  Into
    access
  From   PARTY_JOBS p_job
  Where p_job_id = p_job.JOB_ID;
  if(access = 0)then
    return 'У вас недостаточно прав для публикации глобальных новостей';
  else
    if(:P2_NEWS_TEXT is not null) then
      INSERT INTO "NEWS" (
          AUTOR,
          IS_PERSONAL,
          PUBLISH_DATE,
          TITLE,
          TEXT
```

```
      ) VALUES (
          p_card_id,
          '0',
          SYSTIMESTAMP,
          :P2_NEWS_TITLE,
          :P2_NEWS_TEXT
      );
      return 'Опубликовано в ' || SYSTIMESTAMP;
    else
      return 'Текст новости не может быть пустым!';
    end if;
  end if;
 Exception
  When No_Data_Found Then
    return 'Ошибка публикации, одна из записей не найдена';
 End;
end;
```

**Каталог пива**

SQL запрос: **Таблица BeerBrandsList** – запрос в базу данных для получения информации о марках пива хранящихся в базе данных.

```
select BRAND_ID AS "Уник. номер",
     BEER_NAME AS "Название марки",
     COLOR AS "Оттенок",
     TASTE AS "Оттенки вкуса",
     BOTTLE_COST AS "Стоимость бут(₽)",
     BEER_COMMENT AS "Комментарий"
 from BEER_BRANDS
```

SQL запрос: **Таблица партийные заказы** – запрос в базу данных для получения информации о заказах пива, сделанных пользователями.

```
select BEER_PARTY_ORDERS.ORDER_ID AS "Номер заказа",
    BEER_PARTY_ORDERS.BEER_BRAND AS "Номер бренда",
    BEER_BRANDS.BEER_NAME AS "Название марки",
    BEER_PARTY_ORDERS.COUNT AS "Количество бут.",
    BEER_PARTY_ORDERS.ORDER_DATE AS "Дата заказа",
    BEER_PARTY_ORDERS.PURPOSE_OF_THE_ORDER AS "Цель заказа",
    BEER_PARTY_ORDERS.ALLOCATED_MONEY AS "Выделенные сред.",
    person1.LAST_NAME || person1.NAME ||'('|| person1.LOGIN_MAIL || ')' AS "Заказчик",
    person1.LAST_NAME || person1.NAME ||'('|| person1.LOGIN_MAIL || ')' AS "Ответств. л.",
    BEER_PARTY_ORDERS.SELLER AS "Продавец"
 from BEER_PARTY_ORDERS
  JOIN BEER_BRANDS ON BEER_PARTY_ORDERS.BEER_BRAND = BEER_BRANDS.BRAND_ID
  JOIN PARTY_CARDS card1 ON BEER_PARTY_ORDERS.CUSTOMER_CARD = card1.CARD_ID
  JOIN PERSONS person1 ON card1.PERSON=person1.LOGIN_MAIL
  JOIN PARTY_CARDS card2 ON BEER_PARTY_ORDERS.RESPONSIBLE_PERSON_CARD = card2.CARD_ID
  JOIN PERSONS person2 ON card2.PERSON=person2.LOGIN_MAIL
```

DynamicAction(PL/SQL): **HideOrderCreate** – отрабатывает при загрузке страницы, в случае отсутствия у пользователя прав на создание партийного заказа, убирает блок заказов

```
declare
  p_card_id     PERSONS.PARTY_CARD%Type;
```

```
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    p_job_access   PARTY_JOBS.ACCESS_CREATE_ORDERS%Type;
begin
  Begin
    Select
      person.PARTY_CARD
    Into
      p_card_id
    From   PERSONS person
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
    if(p_card_id IS NULL) then
      return true;
    end if;
    Select
      p_card.PARTY_JOB
    Into
      p_job_id
    From   PARTY_CARDS p_card
    Where p_card_id = p_card.CARD_ID;
    if(p_job_id IS NULL)then
      return true;
    end if;
    Select
      jobs.ACCESS_CREATE_ORDERS
    Into
      p_job_access
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;
    if(p_job_access = 0)then
```

```
      return true;
    else
      return false;
    end if;
  Exception
    When No_Data_Found Then
      return true;
  End;
end;
```

DynamicAction(PL/SQL): **HideOrders** – отрабатывает при загрузке страницы, в случае отсутствия у пользователя прав на просмотр партийных заказов, убирает блок просмотра заказов

```
declare
    p_card_id      PERSONS.PARTY_CARD%Type;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    p_job_access   PARTY_JOBS.ACCESS_CREATE_ORDERS%Type;
begin
  Begin
    Select
      person.PARTY_CARD
    Into
      p_card_id
    From   PERSONS person
    Where     APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')  =
TRIM(person.LOGIN_MAIL);
    if(p_card_id IS NULL) then
      return true;
    end if;
    Select
```

```
      p_card.PARTY_JOB
    Into
      p_job_id
    From   PARTY_CARDS p_card
    Where p_card_id = p_card.CARD_ID;
    if(p_job_id IS NULL)then
      return true;
    end if;
    Select
      jobs.ACCESS_VIEW_RESPONSIBLE_ORDERS
    Into
      p_job_access
    From   PARTY_JOBS jobs
    Where p_job_id = jobs.JOB_ID;
    if(p_job_access = 0)then
      return true;
    else
      return false;
    end if;
  Exception
    When No_Data_Found Then
      return true;
  End;
end;
```

DynamicAction(PL/SQL): **CheckBottlesCost** – отрабатывает когда пользователь в партийном заказе изменяет количество или тип бутылок пива, и расчитывает общую стоимость заказа.

```
declare
    p_cost      BEER_BRANDS.BOTTLE_COST%Type;
```

```
    p_count     integer;
begin
  Begin
    Select
      p_brand.BOTTLE_COST
    Into
      p_cost
    From BEER_BRANDS p_brand
    Where TO_NUMBER(:P3_BRAND) = p_brand.BRAND_ID;
    if(p_cost IS NULL)then
      return 'Ошибка выбора бренда пива!';
    end if;
    return ''||(p_cost * TO_NUMBER(:P3_BOTTLES_COUNT));
  Exception
    When No_Data_Found Then
      return 'Ошибка введенных значений!';
  End;
end;
```

DynamicAction(PL/SQL): **CreateOrder** – отрабатывает при нажатии кнопки создать заказ, отправляет сведения о заказе в базу данных и изменяет статус заказа.

```
declare
    p_card_id      PERSONS.PARTY_CARD%Type;
    p_job_id       PARTY_CARDS.PARTY_JOB%Type;
    access         PARTY_JOBS.ACCESS_CREATE_ORDERS%Type;
    p_resp_card_id      PERSONS.PARTY_CARD%Type;
begin
```

```
  Begin
    Select
      person.PARTY_CARD
    Into
      p_card_id
    From   PERSONS person
    Where    APEX_UTIL.GET_SESSION_STATE('SESSION_USER_NAME')    =
TRIM(person.LOGIN_MAIL);
    if(p_card_id IS NULL) then
      return 'Не найден ваш парт. билет!';
    end if;
    Select
      person.PARTY_CARD
    Into
      p_resp_card_id
    From   PERSONS person
    Where :P3_RESPONSIBLE_PERSON = TRIM(person.LOGIN_MAIL);
    if(p_card_id IS NULL) then
      return 'Не найден парт. билет ответственного за заказ человека!';
    end if;
    Select
      p_card.PARTY_JOB
    Into
      p_job_id
    From   PARTY_CARDS p_card
    Where p_card_id = p_card.CARD_ID;
    if(p_job_id IS NULL)then
      return 'Не найдена партийная должность!';
    end if;
     Select
```

```
  p_job.ACCESS_CREATE_ORDERS
Into
  access
From   PARTY_JOBS p_job
Where p_job_id = p_job.JOB_ID;
if(access = 0)then
  return 'У вашей должности недостаточно прав для создания заказов';
else
  if(
    :P3_BRAND IS NOT NULL AND
    :P3_BOTTLES_COUNT IS NOT NULL AND
    :P3_ALLOCATED_MONEY IS NOT NULL AND
    :P3_PURPOSE IS NOT NULL AND
    :P3_RESPONSIBLE_PERSON IS NOT NULL AND
    :P3_SELLER IS NOT NULL
    )then
    INSERT INTO "BEER_PARTY_ORDERS" (
        BEER_BRAND,
        COUNT,
        ORDER_DATE,
        PURPOSE_OF_THE_ORDER,
        ALLOCATED_MONEY,
        CUSTOMER_CARD,
        RESPONSIBLE_PERSON_CARD,
        SELLER
    ) VALUES (
        :P3_BRAND,
        :P3_BOTTLES_COUNT,
        SYSDATE,
        :P3_PURPOSE,
```

```
          :P3_ALLOCATED_MONEY,
          p_card_id,
          p_resp_card_id,
          :P3_SELLER
        );
        return 'Заказ добавлен в ' || SYSTIMESTAMP;
      else
        return 'Одно из полей пустое!';
      end if;
    end if;
  Exception
    When No_Data_Found Then
      return 'Ошибка в заказе, одна из записей не найдена!';
  End;
end;
```

**Страница обработки заявлений**

PL/SQL:**P4_PETITION_ID** – запрос получения номера петиции.

```
begin
   return APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
end;
```

PL/SQL: **P4_PETITION_DATE** – запрос получения даты петиции.

```
Declare
  temp_p JOIN_PETITIONS.FILLING_DATE%TYPE;
begin
  select
    JOIN_PETITIONS.FILLING_DATE
  INTO
    temp_p
```

```
  FROM JOIN_PETITIONS
  WHERE JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  return temp_p;
end;
```

PL/SQL: **P4_FACE** – запрос получения почты пользователя подавшего петицию.

```
Declare
  temp_p JOIN_PETITIONS.FACE%TYPE;
begin
  select
    JOIN_PETITIONS.FACE
  INTO
    temp_p
  FROM JOIN_PETITIONS
  WHERE JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  return temp_p;
end;
```

PL/SQL: **P4_CURRENT_UNTIL** – запрос получения срока годности текущего партийного билета.

```
Declare
  face_p JOIN_PETITIONS.FACE%TYPE;
  card_id_p PERSONS.PARTY_CARD%TYPE;
  valid_p PARTY_CARDS.VALID_UNTIL%TYPE;
begin
  -- получаем человека
  select
```

```
    JOIN_PETITIONS.FACE
  INTO
    face_p
  FROM JOIN_PETITIONS
  WHERE JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  -- получаем id его билета
  select
    PERSONS.PARTY_CARD
  INTO
    card_id_p
  FROM PERSONS
  WHERE PERSONS.LOGIN_MAIL = face_p;
  -- если билет существует
  if(card_id_p is not null)then
    -- получаем срок годности
    select
      PARTY_CARDS.VALID_UNTIL
    INTO
      valid_p
    FROM PARTY_CARDS
    WHERE PARTY_CARDS.CARD_ID = card_id_p;
    return valid_p;
  else
    return '---';
  end if;
end;
```

PL/SQL:**P4_ADD_UNTIL** – запрос получения времени продления партийного билета.

```
Declare
  temp_p JOIN_PETITIONS.MONTHS_COUNT%TYPE;
begin
  select
    JOIN_PETITIONS.MONTHS_COUNT
  INTO
    temp_p
  FROM JOIN_PETITIONS
  WHERE JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  return to_char(temp_p)||' месяцев';
end;
```

PL/SQL:**P4_CURRENT_JOB** – запрос получения текущей должности указанной в парт билете.

```
Declare
  face_p JOIN_PETITIONS.FACE%TYPE;
  card_id_p PERSONS.PARTY_CARD%TYPE;
  job_id_p PARTY_CARDS.PARTY_JOB%TYPE;
  job_name_p PARTY_JOBS.JOB_NAME%TYPE;
begin
  -- получаем человека
  select
    JOIN_PETITIONS.FACE
  INTO
    face_p
  FROM JOIN_PETITIONS
  WHERE JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  -- получаем id его билета
```

```
  select
    PERSONS.PARTY_CARD
  INTO
    card_id_p
  FROM PERSONS
  WHERE PERSONS.LOGIN_MAIL = face_p;
  -- если билет существует
  if(card_id_p is not null)then
    -- получаем id работы
    select
      PARTY_CARDS.PARTY_JOB
    INTO
      job_id_p
    FROM PARTY_CARDS
    WHERE PARTY_CARDS.CARD_ID = card_id_p;
    -- получаем работу
    select
      PARTY_JOBS.JOB_NAME
    INTO
      job_name_p
    FROM PARTY_JOBS
    WHERE PARTY_JOBS.JOB_ID = job_id_p;
    return job_name_p;
  else
    return '---';
  end if;
end;
```

PL/SQL: **P4_NEW_JOB** – запрос получения указанной в петиции должности.

```
Declare
  temp_p JOIN_PETITIONS.TARGET_JOB%TYPE;
  job_name_p PARTY_JOBS.JOB_NAME%TYPE;
begin
  -- id работы
  select
    JOIN_PETITIONS.TARGET_JOB
  INTO
    temp_p
  FROM JOIN_PETITIONS
  WHERE JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  -- получаем работу
  select
    PARTY_JOBS.JOB_NAME
  INTO
    job_name_p
  FROM PARTY_JOBS
  WHERE PARTY_JOBS.JOB_ID = temp_p;
  return job_name_p;
end;
```

DynamicAction(PL/SQL): **CANCEL** – отрабатывает при нажатии кнопки «отклонить петицию», отправляет статус петиции в базу данных.

```
Declare
  temp_p JOIN_PETITIONS.FILLING_DATE%TYPE;
begin
  UPDATE
    "KR_ADMIN"."JOIN_PETITIONS"
  SET
```

```
    STATUS = '2'
  WHERE
    JOIN_PETITIONS.PETITION_ID =
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
end;
```

DynamicAction(PL/SQL): **ACCEPT** – отрабатывает при нажатии кнопки «принять петицию», отправляет статус петиции в базу данных.

```
Declare
  job_id_p JOIN_PETITIONS.TARGET_JOB%TYPE;
  month_p JOIN_PETITIONS.MONTHS_COUNT%TYPE;
  mail_p PERSONS.LOGIN_MAIL%TYPE;
  card_id_p PERSONS.PARTY_CARD%TYPE;
begin
  -- ищем новую работу
  select
    JOIN_PETITIONS.TARGET_JOB
  INTO
    job_id_p
  FROM JOIN_PETITIONS
  WHEREJOIN_PETITIONS.PETITION_ID=
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
  -- ищем срок продления
  select
    JOIN_PETITIONS.MONTHS_COUNT
  INTO
    month_p
  FROM JOIN_PETITIONS
  WHEREJOIN_PETITIONS.PETITION_ID=
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
```

```
-- ищем парт. билет
select
  PERSONS.LOGIN_MAIL,
  PERSONS.PARTY_CARD
INTO
  mail_p,
  card_id_p
FROM JOIN_PETITIONS
  JOIN PERSONS ON JOIN_PETITIONS.FACE = PERSONS.LOGIN_MAIL
WHERE JOIN_PETITIONS.PETITION_ID=
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
if(card_id_p is not null)then
  -- если парт билет существует, просто меняем его данные
  UPDATE
    "KR_ADMIN"."PARTY_CARDS"
  SET
    PARTY_JOB = job_id_p,
    VALID_UNTIL = ADD_MONTHS(PARTY_CARDS.VALID_UNTIL,
month_p)
  WHERE
    PARTY_CARDS.CARD_ID = card_id_p;
else
  -- если парт билет не существует, создаем его
  INSERT INTO "PARTY_CARDS" (
        PERSON,
        PARTY_JOB,
        FILLING_DATE,
        VALID_UNTIL
  ) VALUES (
        mail_p,
```

```
        job_id_p,
        SYSTIMESTAMP,
        ADD_MONTHS(SYSTIMESTAMP, month_p)
  );
  -- получаем id карты
  select
    PARTY_CARDS.CARD_ID
  INTO
    card_id_p
  FROM PARTY_CARDS
  WHERE PARTY_CARDS.PERSON = mail_p;
  -- записываем id пользователю
  UPDATE
    "KR_ADMIN"."PERSONS"
  SET
    PARTY_CARD = card_id_p
  WHERE PERSONS.LOGIN_MAIL = mail_p;

 end if;
 UPDATE
  "KR_ADMIN"."JOIN_PETITIONS"
 SET
  STATUS = '1'
 WHERE
  JOIN_PETITIONS.PETITION_ID=
APEX_UTIL.GET_SESSION_STATE('IN_BUFFER_1');
end;
```

# Глава 3. Результат

## 3.1 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.

- 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

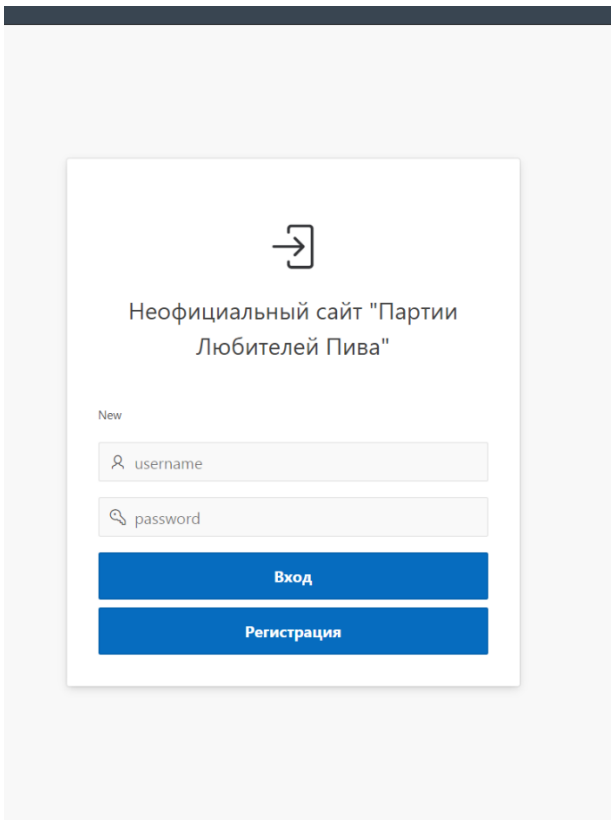



Рисунок 10 – Страница входа

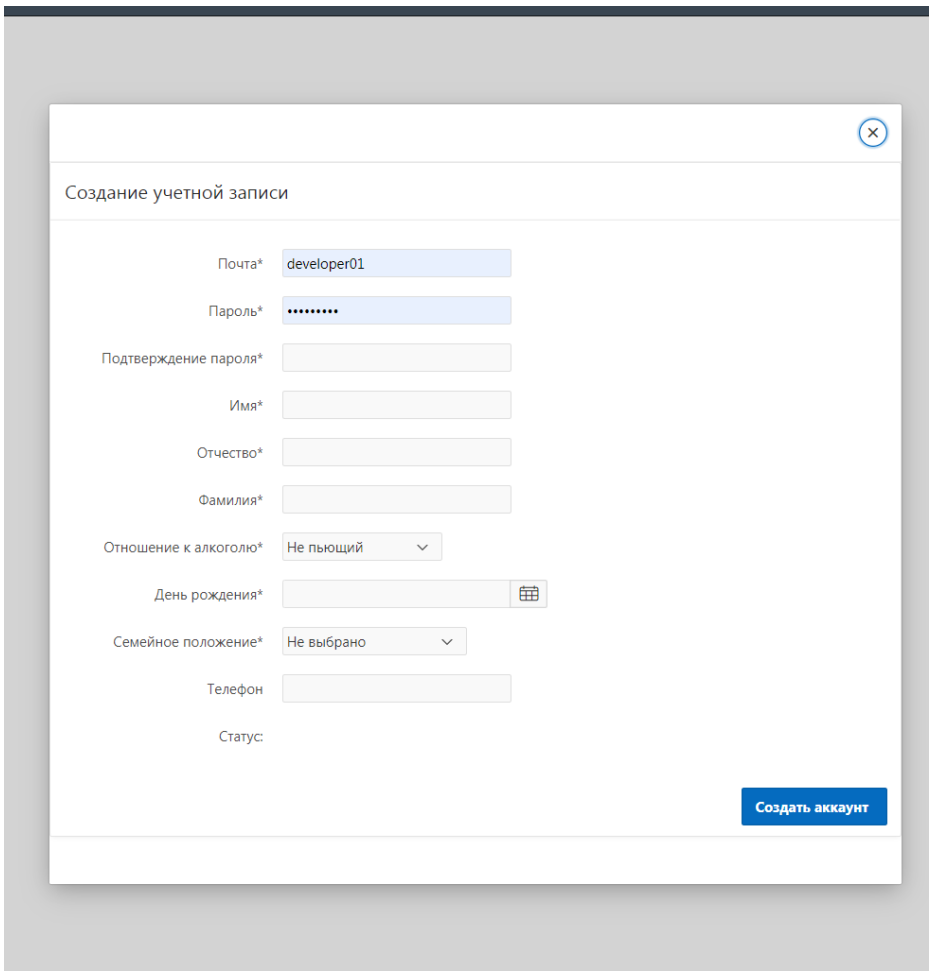



Рисунок 11 – Страница регистрации

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.

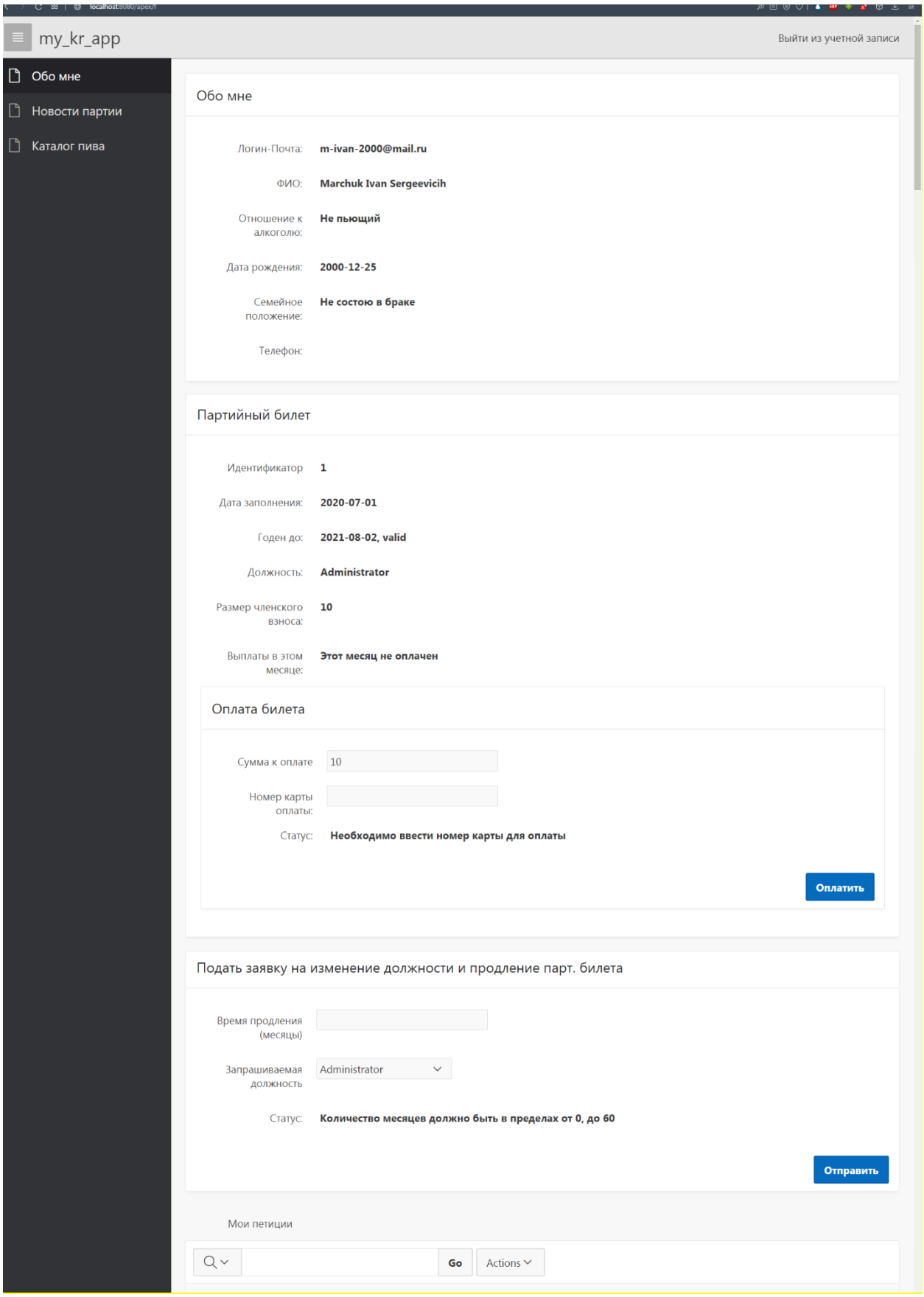



Рисунок 12 – «Персональная страница»

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.

my_kr_app

Выйти из учетной записи

Обо мне

Новости партии

Каталог пива

## Новости партии

Go Actions

| Дата публикации | Автор | Заголовок | Текст |
|---|---|---|---|
| 2018-03-16 | Ындырчи Аалия (aaa@mail.com ) | О новой модели | Товарищи! оптимизация основных целей позволяет выполнить важные задания по разработке экономических и неэкономических факторов и перспектив.Товарищи! дальнейшее развитие различных форм деятельности не дает нам иного выбора, кроме определения инновационных методов управления процессами.С другой стороны сложившаяся структура организации требуют от нас анализа стандартных подходов.Позвольте Вам напомнить, что дальнейшее развитие различных форм деятельности вынуждает нас объективно потребовать направлений прогрессивного развития. |
| 2017-12-03 | Ынсап Аанжелла (aaa@gmail.com ) | Идеи | Таким образом дальнейшее развитие различных форм деятельности способствует подготовке и реализации инновационных методов управления процессами.Также как оптимизация основных целей выявляет срочную потребность позиций, занимаемых участниками в отношении поставленных задач. |
| 2018-05-12 | Ынтымак Аба (aaa@yandex.ru ) | Внедрение позиций | Уважаемые коллеги! экономическая повестка сегодняшнего дня способствует подготовке и реализации направлений прогрессивного развития.Таким образом экономическая повестка сегодняшнего дня способствует подготовке и реализации дальнейших направлений развития.Повседневная практика показывает, что новая модель организационной деятельности обеспечивает широкому кругу (специалистов) участие в формировании новых предложений.Позвольте Вам напомнить, что оптимизация основных целей не дает нам иного выбора, кроме определения существенных финансовых и административных условий. |
| 2017-08-11 | Ыргым Абав (aaa@xyz.com ) | Изменения структуры | Товарищи! постоянный количественный рост и сфера нашей активности способствует подготовке и реализации нестандартных решений.Равным образом оптимизация основных целей обеспечивает широкому кругу (специалистов) участие в формировании направлений прогрессивного развития.Товарищи! внедрение современных подходов требуют определения и уточнения дальнейших направлений развития. |
| 2018-05-18 | Ырсай Абам (aab@mail.com ) | Идеи | Повседневная практика показывает, что реализация намеченных плановых заданий играет важную роль в формировании позиций, занимаемых участниками в отношении поставленных задач.С другой стороны рамки и место обучения кадров требуют определения и уточнения новых предложений.Равным образом оптимизация основных целей требуют определения и уточнения инновационных методов управления процессами. |

1 - 5

### Опубликовать новость

Заголовок

Текст

Статус: **Текст новости не может быть пустым!**

Опубликовать

Заявления на на изменение должности и продление парт. билета:

Go Actions

| Номер заявления | Подавшее лицо | Дата подачи | Продление на | Изменить |
|---|---|---|---|---|
| 22041 | ea8@yandex.ru | 2015-04-02 | 18 месяцев | |
| 22042 | ea8@xyz.com | 2005-06-18 | 24 месяцев | |
| 22043 | ea9@mail.com | 2017-11-07 | 19 месяцев | |
| 22044 | ea9@gmail.com | 2004-01-10 | 19 месяцев | |
| 22045 | ea9@yandex.ru | 2005-03-16 | 16 месяцев | |

1 - 5

Партийные выплаты

Go Actions

| Уник. номер | Владелец билета | Дата месяца | Сумма оплаты | Дата оплаты |
|---|---|---|---|---|
| 24 | aaa@mail.com (Аалия Шепард Ындырчи ) | 2019-12 | 200 | 2019-12-08 |
| 23 | aaa@mail.com (Аалия Шепард Ындырчи ) | 2019-11 | 200 | 2019-11-04 |
| 22 | aaa@mail.com (Аалия Шепард Ындырчи ) | 2019-10 | 200 | 2019-10-11 |
| 21 | aaa@mail.com (Аалия Шепард Ындырчи ) | 2019-09 | 200 | 2019-09-02 |
| 20 | aaa@mail.com (Аалия Шепард Ындырчи ) | 2019-08 | 200 | 2019-08-06 |

1 - 5

Рисунок 13 – «Новости партии»

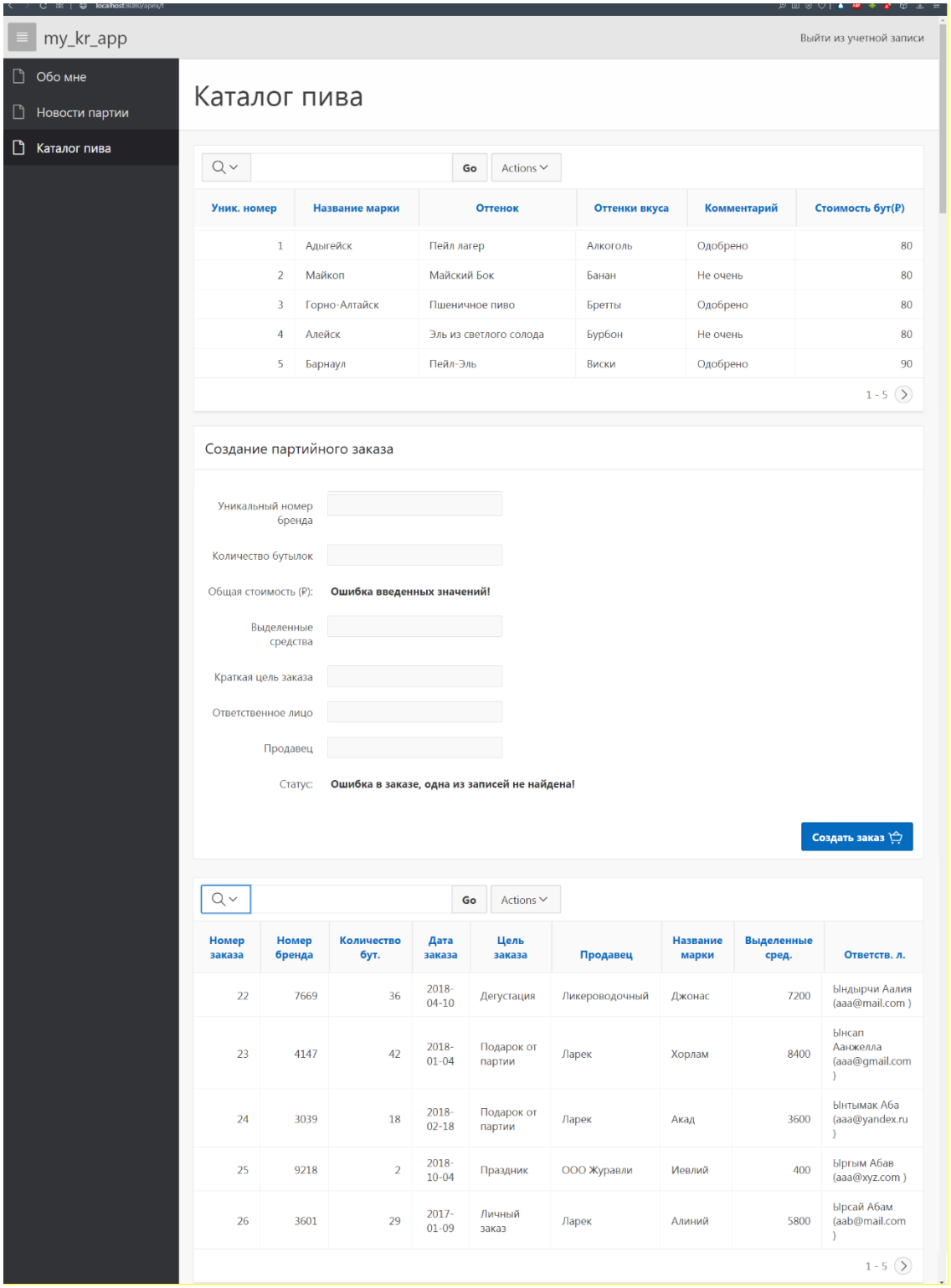



Рисунок 14 – «Каталог пива»

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://o7planning.org/ru/11003/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;
```