|  |
| --- |
| МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА» (РУТ (МИИТ))  ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Кафедра «Управление и защита информации» |
|  |
| **КУРСОВОЙ ПРОЕКТ**  **по дисциплине «Основы построения защищенных баз данных»** |
|  |
|  |
|  |
| Выполнил: ст. гр. ТКИ – 441  Галыба Л.Д. |
| Проверил: к.т.н., доц.  Васильева М.А. |
|  |
|  |
|  |
| Москва 2024 |

**Оглавление**

[Введение 4](#_Toc167911846)

[Проектирование базы данных 5](#_Toc167911847)

[Создание базы данных 9](#_Toc167911848)

[Создание таблиц 9](#_Toc167911849)

[Изменения таблицы Staff 10](#_Toc167911850)

[Изменения таблицы Vacations 12](#_Toc167911851)

[Изменения таблицы Children 14](#_Toc167911852)

[Изменения таблицы Disability 15](#_Toc167911853)

[Изменения таблицы Posts 15](#_Toc167911854)

[Изменения таблицы PreviousWorks 16](#_Toc167911855)

[Заполнение базы данных 18](#_Toc167911856)

[Создание функций 19](#_Toc167911857)

[Создание процедур 25](#_Toc167911858)

[Создание представлений 34](#_Toc167911859)

[Вывод 36](#_Toc167911860)

[Список литературы 37](#_Toc167911861)

Введение

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) в системе управления базами данных (СУБД) PostgreSQL.

В данном курсовом проекте ставится задача разработать РБД в СУБД PostgreSQL по теме «Отдел кадров», необходимо разработать скрипты на создание и заполнение РБД.

База данных создаётся для информационного обслуживания работников отдела кадров. В фирме работают сотрудники на разных должностях и разных ставках. Сотрудники могут иметь детей, быть пенсионерами или иметь инвалидность. Сотрудникам предоставляет очередной отпуск. Некоторые сотрудники могут быть в отпуске по уходу за ребенком.

Готовые запросы:

* Показывать список сотрудников фирмы по ФИО, по должности, ставке, имеющих детей, пенсионеров, находящихся в очередном отпуске, находящихся в отпуске по уходу за ребенком.
* Выдавать информацию о должностях сотрудников.
* Выдавать информацию о предыдущих местах работы сотрудников.
* Находить сотрудников по ФИО.

Проектирование базы данных

Выделим базовые сущности базы данных «ВУЗ»:

* Сотрудники
* Дети
* Предыдущие места работы
* Отпуска
* Должности
* Группы инвалидности

Были созданы следующие сущности:

1. Таблица "Staff" - содержит информацию о сотрудниках
2. Таблица "Children" – содержит информацию о детях
3. Таблица "PreviousWorks" - содержит информацию о предыдущих местах работы
4. Таблица "Vacations" - содержит информацию об отпусках
5. Таблица "Posts" - содержит информацию о должностях
6. Таблица "Disability" - содержит информацию о группах инволидности

На рисунке 1 представлена ER-диаграмма БД.

|  |
| --- |
|  |
| 1. – ER-диаграмма базы данных «Отдел кадров» |

1. - Схема отношения таблицы Staff

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор | ID | int4 | Первичный ключ, обязательное поле |
| Имя сотрудника | Name | varchar(15) | Обязательное поле |
| Фамилия сотрудника | Surname | varchar(30) | Обязательное поле |
| Отчество сотрудника | Patronymic | varchar(25) | Обязательное поле |
| Идентификатор инвалидности | DisabilityID | int4 | Внешний ключ |
| Идентификатор должности | PostID | int4 | Внешний ключ, обязательное поле |
| Пол | Sex | varchar(1) | Обязательное поле |
| Дата рождения | BirthDay | date | Обязательное поле |

1. - Схема отношения таблицы Children

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор | ID | int4 | Первичный ключ, обязательное поле |
| Имя | Name | varchar(15) | Обязательное поле |
| Фамиия | Surname | varchar(30) | Обязательное поле |
| Отчество | Patronymic | varchar(25) | Обязательное поле |
| Идентификатор сотрудника | StaffID | Int4 | Внешний ключ, обязательное поле |
| Дата рождения | BirthDay | date | Обязательное поле |

1. - Схема отношения таблицы PreviousWorks

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор | ID | int4 | Первичный ключ, обязательное поле |
| Название организации | OrganizationName | varchar(50) | Обязательное поле |
| Идентификатор сотрудника | StaffID | Int4 | Внешний ключ, обязательное поле |
| Начало работы | Start | date | Обязательное поле |
| Конец работы | End | date | Обязательное поле |

1. - Схема отношения таблицы Vacations

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор | ID | int4 | Первичный ключ, обязательное поле |
| Отпуск по уходу за ребенком | Maternity | varchar(3) | Обязательное поле |
| Идентификатор сотрудника | StaffID | Int4 | Внешний ключ, обязательное поле |
| Начало работы | Start | date | Обязательное поле |
| Конец работы | End | date | Обязательное поле |

1. - Схема отношения таблицы Disability

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор | ID | int4 | Первичный ключ, обязательное поле |
| Группа инвалидности | Group | Int4 | Обязательное поле |

1. - Схема отношения таблицы Posts

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор | ID | int4 | Первичный ключ, обязательное поле |
| Название должности | Name | varchar(30) | Обязательное поле |
| Зарплата | Salary | numeric(8, 2) | Обязательное поле |

Создание базы данных

БД проектируется в среде DBeaver. Напишем скрипты для создания базы данных и таблиц.

Создание базы данных «Отдел кадров»:

**create** **database** **"HumanResourcesDepartment"**

Создание таблиц

Создание таблицы Staff:

**create** **table** **"Staff"**

(

**"ID"** **int**

, **"Name"** **varchar**(15)

, **"Surname"** **varchar**(20)

, **"Patronymic"** **varchar**(20)

, **"BirthDay"** **varchar**(10)

, **"VacationID"** **int**

, **"ChildID"** **int**

, **"DisabilityID"** **int**

, **"PostID"** **int**

, **"PreviousWorkID"** **int**

)

Создание таблицы Vacations:

**create** **table** **"Vacations"**

(

**"ID"** **int**

, **"StartDate"** **varchar**(10)

, **"EndDate"** **varchar**(10)

, **"Maternity"** **varchar**(3)

)

Создание таблицы Disability:

**create** **table** **"Disability"**

(

**"ID"** **int**

, **"Group"** **int**

)

Создание таблицы Children:

**create** **table** **"Children"**

(

**"ID"** **int**

, **"Name"** **varchar**(15)

, **"Surname"** **varchar**(20)

, **"BirthDay"** **varchar**(10)

)

Создание таблицы Posts:

**create** **table** **"Posts"**

(

**"ID"** **int**

, **"Name"** **varchar**(20)

, **"Salary"** **numeric**(8, 2)

)

Создание таблицы PreviousWorks:

**create** **table** **"PreviousWorks"**

(

**"ID"** **int**

, **"OrganizationName"** **varchar**(50)

, **"Start"** **varchar**(10)

, **"End"** **varchar**(10)

, **"Post"** **varchar**(20)

)

Изменения таблицы Staff

**alter** **table** **"Staff"**

**add** **constraint** **"PK\_Staff"**

**primary** **key** (**"ID"**);

**alter** **table** **"Staff"**

**add** **constraint** **"FK\_Vacations\_Staff"**

**foreign** **key**(**"VacationID"**)

**references** **"Vacations"**(**"ID"**);

**alter** **table** **"Staff"**

**add** **constraint** **"FK\_Children\_Staff"**

**foreign** **key**(**"ChildID"**)

**references** **"Children"**(**"ID"**);

**alter** **table** **"Staff"**

**add** **constraint** **"FK\_Disability\_Staff"**

**foreign** **key**(**"DisabilityID"**)

**references** **"Disability"**(**"ID"**);

**alter** **table** **"Staff"**

**add** **constraint** **"FK\_Posts\_Staff"**

**foreign** **key**(**"PostID"**)

**references** **"Posts"**(**"ID"**);

**alter** **table** **"Staff"**

**add** **constraint** **"FK\_PreviousWorks\_Staff"**

**foreign** **key**(**"PreviousWorkID"**)

**references** **"PreviousWorks"**(**"ID"**);

**alter** **table** **"Staff"**

**alter** **column** **"Name"**

**set** **not** **null**;

**alter** **table** **"Staff"**

**alter** **column** **"Surname"**

**set** **not** **null**;

**alter** **table** **"Staff"**

**alter** **column** **"Patronymic"**

**set** **not** **null**;

**alter** **table** **"Staff"**

**add** **constraint** **"CH\_BirthDay"**

**check**(**"BirthDay"** **similar** **to** **'^(([0-2][0-9])|([3][0-1])).([0-1][0-9]).(([1][9]\d{2})|([2][0]\d{2}))$'**);

**alter** **table** **"Staff"**

**alter** **column** **"BirthDay"**

**set** **not** **null**;

**alter** **table** **"Staff"**

**add** **constraint** **"CH\_Staff\_ID"**

**check**(**"ID"**>0);

**alter** **table** **"Staff"**

**add** **column** **"Sex"** **varchar**(1);

**alter** **table** **"Staff"**

**alter** **column** **"Sex"**

**set** **not** **null**

**alter** **table** **"Staff"**

**drop** **constraint** **"FK\_Vacations\_Staff"**;

**alter** **table** **"Staff"**

**drop** **column** **"VacationID"**;

**alter** **table** **"Staff"**

**drop** **constraint** **"FK\_Children\_Staff"**;

**alter** **table** **"Staff"**

**drop** **column** **"ChildID"**;

**alter** **table** **"Staff"**

**drop** **constraint** **"FK\_PreviousWorks\_Staff"**;

**alter** **table** **"Staff"**

**drop** **column** **"PreviousWorkID"**;

**alter** **table** **"Staff"**

**drop** **constraint** **"CH\_BirthDay"**;

**alter** **table** **"Staff"**

**drop** **column** **"BirthDay"**

**alter** **table** **"Staff"**

**add** **column** **"BirthDay"** **date**

Изменения таблицы Vacations

**alter** **table** **"Vacations"**

**add** **constraint** **"PK\_Vacations"**

**primary** **key** (**"ID"**);

**alter** **table** **"Vacations"**

**alter** **column** **"StartDate"**

**set** **not** **null**;

**alter** **table** **"Vacations"**

**alter** **column** **"EndDate"**

**set** **not** **null**;

**alter** **table** **"Vacations"**

**add** **constraint** **"CH\_StartDate"**

**check**(**"StartDate"** **similar** **to** **'^(([0-2][0-9])|([3][0-1])).([0-1][0-9]).(([1][9]\d{2})|([2][0]\d{2}))$'**);

**alter** **table** **"Vacations"**

**add** **constraint** **"CH\_EndDate"**

**check**(**"EndDate"** **similar** **to** **'^(([0-2][0-9])|([3][0-1])).([0-1][0-9]).(([1][9]\d{2})|([2][0]\d{2}))$'**);

**alter** **table** **"Vacations"**

**add** **constraint** **"CH\_Vacations\_ID"**

**check**(**"ID"**>0);

**alter** **table** **"Vacations"**

**add** **column** **"StaffID"** **int**;

**alter** **table** **"Vacations"**

**add** **constraint** **"FK\_Staff\_Vacations"**

**foreign** **key** (**"StaffID"**)

**references** **"Staff"**(**"ID"**)

**alter** **table** **"Vacations"**

**drop** **constraint** **"CH\_StartDate"**

**alter** **table** **"Vacations"**

**drop** **constraint** **"CH\_EndDate"**

**alter** **table** **"Vacations"**

**drop** **column** **"StartDate"**

**alter** **table** **"Vacations"**

**drop** **column** **"EndDate"**

**alter** **table** **"Vacations"**

**add** **column** **"Start"** **date**

**alter** **table** **"Vacations"**

**add** **column** **"End"** **date**

Изменения таблицы Children

**alter** **table** **"Children"**

**add** **constraint** **"PK\_Children"**

**primary** **key** (**"ID"**);

**alter** **table** **"Children"**

**add** **column** **"Patronymic"** **varchar**(20);

**alter** **table** **"Children"**

**alter** **column** **"Name"**

**set** **not** **null**;

**alter** **table** **"Children"**

**alter** **column** **"Surname"**

**set** **not** **null**;

**alter** **table** **"Children"**

**alter** **column** **"Patronymic"**

**set** **not** **null**;

**alter** **table** **"Children"**

**alter** **column** **"BirthDay"**

**set** **not** **null**;

**alter** **table** **"Children"**

**add** **constraint** **"CH\_BirthDay"**

**check**(**"BirthDay"** **similar** **to** **'^(([0-2][0-9])|([3][0-1])).([0-1][0-9]).(([1][9]\d{2})|([2][0]\d{2}))$'**);

**alter** **table** **"Children"**

**add** **constraint** **"CH\_Children\_ID"**

**check**(**"ID"**>0);

**alter** **table** **"Children"**

**add** **column** **"StaffID"** **int**;

**alter** **table** **"Children"**

**add** **constraint** **"FK\_Staff\_Children"**

**foreign** **key** (**"StaffID"**)

**references** **"Staff"**(**"ID"**);

**alter** **table** **"Children"**

**drop** **constraint** **"CH\_BirthDay"**

**alter** **table** **"Children"**

**drop** **column** **"BirthDay"**

**alter** **table** **"Children"**

**add** **column** **"BirthDay"** **date**

Изменения таблицы Disability

**alter** **table** **"Disability"**

**add** **constraint** **"PK\_Disability"**

**primary** **key** (**"ID"**);

**alter** **table** **"Disability"**

**add** **constraint** **"CH\_Disability\_ID"**

**check**(**"ID"**>0);

Изменения таблицы Posts

**alter** **table** **"Posts"**

**add** **constraint** **"PK\_Posts"**

**primary** **key** (**"ID"**);

**alter** **table** **"Posts"**

**alter** **column** **"Name"**

**set** **not** **null**;

**alter** **table** **"Posts"**

**alter** **column** **"Salary"**

**set** **not** **null**;

**alter** **table** **"Posts"**

**add** **constraint** **"CH\_Posts\_ID"**

**check**(**"ID"**>0);

**alter** **table** **"Posts"**

**add** **constraint** **"CH\_Posts\_Salary"**

**check**(**"Salary"**>0);

Изменения таблицы PreviousWorks

**alter** **table** **"PreviousWorks"**

**add** **constraint** **"PK\_PreviousWorks"**

**primary** **key** (**"ID"**);

**alter** **table** **"PreviousWorks"**

**alter** **column** **"OrganizationName"**

**set** **not** **null**;

**alter** **table** **"PreviousWorks"**

**alter** **column** **"Start"**

**set** **not** **null**;

**alter** **table** **"PreviousWorks"**

**alter** **column** **"End"**

**set** **not** **null**;

**alter** **table** **"PreviousWorks"**

**alter** **column** **"Post"**

**set** **not** **null**;

**alter** **table** **"PreviousWorks"**

**add** **constraint** **"CH PreviousWork\_ID"**

**check**(**"ID"**>0);

**alter** **table** **"PreviousWorks"**

**add** **constraint** **"CH\_Start"**

**check**(**"Start"** **similar** **to** **'^(([0-2][0-9])|([3][0-1])).([0-1][0-9]).(([1][9]\d{2})|([2][0]\d{2}))$'**);

**alter** **table** **"PreviousWorks"**

**add** **constraint** **"CH\_End"**

**check**(**"End"** **similar** **to** **'^(([0-2][0-9])|([3][0-1])).([0-1][0-9]).(([1][9]\d{2})|([2][0]\d{2}))$'**);

**alter** **table** **"PreviousWorks"**

**add** **column** **"StaffID"** **int**;

**alter** **table** **"PreviousWorks"**

**add** **constraint** **"FK\_Staff\_PreviousWorks"**

**foreign** **key**(**"StaffID"**)

**references** **"Staff"**(**"ID"**);

**alter** **table** **"PreviousWorks"**

**drop** **constraint** **"CH\_Start"**

**alter** **table** **"PreviousWorks"**

**drop** **constraint** **"CH\_End"**

**alter** **table** **"PreviousWorks"**

**drop** **column** **"Start"**

**alter** **table** **"PreviousWorks"**

**drop** **column** **"End"**

**alter** **table** **"PreviousWorks"**

**add** **column** **"Start"** **date**

**alter** **table** **"PreviousWorks"**

**add** **column** **"End"** **date**

Заполнение базы данных

Создадим скрипты для заполнения базы данных «ВУЗ» с целью дальнейшего формирования запросов.

**insert** **into** **"Disability"** **values** (1, 1)

**insert** **into** **"Disability"** **values** (2, 2)

**insert** **into** **"Disability"** **values** (3, 3)

**insert** **into** **"Posts"** **values** (1, **'Разработчик'**, 100000)

**insert** **into** **"Posts"** **values** (2, **'Аналитик'**, 90000)

**insert** **into** **"Posts"** **values** (3, **'Уборщик'**, 50000)

Создание функций

1. Функция получения идентификатора сотрудника

**create** **function** **"fGetStaffID"**(

**"surname"** **varchar**(20)

, **"name"** **varchar**(15)

, **"patronymic"** **varchar**(20)

)

**returns** **integer** **as**

**$$**

**select** s.**"ID"**

**from** **"Staff"** s

**where** s.**"Surname"** = **"surname"**

**and** s.**"Name"** = **"name"**

**and** s.**"Patronymic"** = **"patronymic"**;

**$$** **language** **sql**;

|  |
| --- |
|  |
| 1. – Результат вызова функции |

2. Функция получения идентификатора должности

**create** **function** **"fGetPostID"**(**"name"** **varchar**(15))

**returns** **integer** **as**

**$$**

**select** p.**"ID"**

**from** **"Posts"** p

**where** p.**"Name"** = **"name"**;

**$$** **language** **sql**;

|  |
| --- |
|  |
| 1. – Результат вызова функции |

3. Функция получения идентификатора инвалидности

**create** **function** **"fGetDisabilityID"**(**"group"** **integer**)

**returns** **integer** **as**

**$$**

**select** d.**"ID"**

**from** **"Disability"** d

**where** d.**"Group"** = **"group"**;

**$$** **language** **sql**

|  |
| --- |
|  |
| 1. – Результат вызова функции |

4. Функция получения следующего идентификатора для сотрудника

**create** **function** **"fGetNextStaffID"**()

**returns** **integer**

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

**BEGIN**

**select** **max**(s.**"ID"**)

**into** id

**from** **"Staff"** s;

**if** id **is** **null**

**then**

id = 1;

**else**

id = id + 1;

**end** **if**;

**return** id;

**end**;

**$$**

1. Функция получения следующего идентификатора для отпуска

**create** **function** **"fGetNextVacationID"**()

**returns** **integer**

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

**BEGIN**

**select** **max**(v.**"ID"**)

**into** id

**from** **"Vacations"** v;

**if** id **is** **null**

**then**

id = 1;

**else**

id = id + 1;

**end** **if**;

**return** id;

**end**;

**$$**

1. Функция получения следующего идентификатора для ребенка

**create** **function** **"fGetNextChildrenID"**()

**returns** **integer**

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

**BEGIN**

**select** **max**(c.**"ID"**)

**into** id

**from** **"Children"** c;

**if** id **is** **null**

**then**

id = 1;

**else**

id = id + 1;

**end** **if**;

**return** id;

**end**;

**$$**

1. Функция получения следующего идентификатора для предыдущего места работы

**create** **function** **"fGetNextPreviousWorkID"**()

**returns** **integer**

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

**BEGIN**

**select** **max**(pw.**"ID"**)

**into** id

**from** **"PreviousWorks"** pw;

**if** id **is** **null**

**then**

id = 1;

**else**

id = id + 1;

**end** **if**;

**return** id;

**end**;

**$$**

1. Функция показывает сотрудников по должности

**create** **function** **"fShowByPost"**(**in** post **varchar**(30))

**returns** **table** (

**"Surname"** **varchar**(15)

, **"Name"** **varchar**(30)

, **"Patronymic"** **varchar**(25)

)

**language** plpgsql

**as** **$$**

**declare**

id\_post **integer**;

**begin**

**select** \* **into** id\_post **from** **"fGetPostID"**(post);

**return** query

**select** s.**"Surname"**

, s.**"Name"**

, s.**"Patronymic"**

**from** **"Staff"** s

**where** **"id\_post"** = s.**"PostID"** ;

**end**;

**$$**

|  |
| --- |
|  |
|  |
| 1. – Результат вызова функции |

Создание процедур

1. Процедура добавления записи в таблицу "Staff".

**create** **procedure** **"pMakeStaff"**(

**in** surname **varchar**(25)

, **in** **name** **varchar**(15)

, **in** patronymic **varchar**(25)

, **in** birthday **date**

, **in** sex **varchar**(1)

, **in** post **varchar**(20)

, **in** disability\_group **integer**

)

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

id\_disability **integer**;

id\_post **integer**;

**begin**

**if** **name** **is** **null** **or** **name** = **''**

**then**

**raise** **exception** **'Неправильное имя сотрудника'**;

**end** **if**;

**if** surname **is** **null** **or** surname = **''**

**then**

**raise** **exception** **'Неправильная фамилия сотрудника'**;

**end** **if**;

**if** patronymic **is** **null** **or** patronymic = **''**

**then**

**raise** **exception** **'Неправильное отчество сотрудника'**;

**end** **if**;

**if** post **is** **null** **or** post = **''**

**then**

**raise** **exception** **'Неправильная должность сотрудника'**;

**end** **if**;

**if** birthday = **now**()

**then**

**raise** **exception** **'Неправильный формат даты'**;

**end** **if**;

**if** sex **not** **similar** **to** **'М|Ж'**

**then**

**raise** **exception** **'Неправильный формат пола'**;

**end** **if**;

**if** disability\_group < 0 **or** disability\_group > 3

**then**

**raise** **exception** **'Неправильный формат группы инвалидности'**;

**end** **if**;

**select** \* **into** id **from** public.**"fGetNextStaffID"**();

**select** \*

**into** id\_post **from** public.**"fGetPostID"**(post);

**if** id\_post **is** **null**

**then**

**raise** **exception** **'Не найдена должность %'**, post;

**end** **if**;

**select** \*

**into** id\_disability **from** public.**"fGetDisabilityID"**(disability\_group);

**insert** **into** **"Staff"**(**"ID"**, **"Surname"**, **"Name"**, **"Patronymic"**, **"BirthDay"**, **"DisabilityID"**, **"PostID"**, **"Sex"**)

**values** (id, surname, **name**, patronymic, birthday, id\_disability, id\_post, sex);

**raise** **notice** **'Добавлен новый сотрудник'**;

**end**;

**$$**

|  |
| --- |
|  |
|  |
| 1. – Результат вызова процедуры |

1. Процедура добавления записи в таблицу "Vacations".

**create** **procedure** **"pMakeVacation"**(

**in** start\_date **date**

, **in** end\_date **date**

, **in** staff\_surname **varchar**(25)

, **in** staff\_name **varchar**(15)

, **in** staff\_patronimic **varchar**(25)

, **in** maternity **varchar**(3)

)

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

id\_staff **integer**;

**begin**

**if** maternity **not** **similar** **to** **'^Да|Нет'**

**then**

**raise** **exception** **'Неправильный формат статуса отпуска по уходу за ребенком'**;

**end** **if**;

**if** staff\_name **is** **null** **or** staff\_name = **''**

**then**

**raise** **exception** **'Неправильное имя сотрудника'**;

**end** **if**;

**if** staff\_surname **is** **null** **or** staff\_surname = **''**

**then**

**raise** **exception** **'Неправильная фамилия сотрудника'**;

**end** **if**;

**if** staff\_patronimic **is** **null** **or** staff\_patronimic = **''**

**then**

**raise** **exception** **'Неправильное отчество сотрудника'**;

**end** **if**;

**select** \* **into** id **from** public.**"fGetNextVacationID"**();

**select** \*

**into** id\_staff **from** public.**"fGetStaffID"**(staff\_surname, staff\_name, staff\_patronimic);

**if** id\_staff **is** **null**

**then**

**raise** **exception** **'Не найден сотрудник'**;

**end** **if**;

**insert** **into** **"Vacations"** (**"ID"**, **"Start"**, **"End"**, **"Maternity"**, **"StaffID"**)

**values** (id, start\_date, end\_date, maternity, id\_staff);

**raise** **notice** **'Добавлены новые данные об отпуске'**;

**end**;

**$$**

|  |
| --- |
|  |
|  |
| 1. – Результат вызова процедуры |

1. Процедура добавления записи в таблицу "Children".

**create** **procedure** **"pMakeChildren"**(

**in** surname **varchar**(25)

, **in** **name** **varchar**(15)

, **in** patronymic **varchar**(25)

, **in** birthday **date**

, **in** parent\_surname **varchar**(25)

, **in** parent\_name **varchar**(15)

, **in** parent\_patronimic **varchar**(25)

)

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

id\_staff **integer**;

**begin**

**if** **name** **is** **null** **or** **name** = **''**

**then**

**raise** **exception** **'Неправильное имя ребенка'**;

**end** **if**;

**if** surname **is** **null** **or** surname = **''**

**then**

**raise** **exception** **'Неправильная фамилия ребенка'**;

**end** **if**;

**if** patronymic **is** **null** **or** patronymic = **''**

**then**

**raise** **exception** **'Неправильное отчество ребенка'**;

**end** **if**;

**if** birthday > **now**()

**then**

**raise** **exception** **'Неправильный формат даты'**;

**end** **if**;

**if** parent\_name **is** **null** **or** parent\_name = **''**

**then**

**raise** **exception** **'Неправильное имя сотрудника'**;

**end** **if**;

**if** parent\_surname **is** **null** **or** parent\_surname = **''**

**then**

**raise** **exception** **'Неправильная фамилия сотрудника'**;

**end** **if**;

**if** parent\_patronimic **is** **null** **or** parent\_patronimic = **''**

**then**

**raise** **exception** **'Неправильное отчество сотрудника'**;

**end** **if**;

**select** \* **into** id **from** public.**"fGetNextChildrenID"**();

**select** \*

**into** id\_staff **from** public.**"fGetStaffID"**(parent\_surname, parent\_name, parent\_patronimic);

**if** id\_staff **is** **null**

**then**

**raise** **exception** **'Не найден сотрудник'**;

**end** **if**;

**insert** **into** **"Children"**(**"ID"**, **"Surname"**, **"Name"**, **"BirthDay"**, **"Patronymic"**,**"StaffID"**)

**values** (id, surname, **name**, birthday, patronymic, id\_staff );

**raise** **notice** **'Добавлен новый ребенок'**;

**end**;

**$$**

|  |
| --- |
|  |
|  |
| 1. – Результат вызова процедуры |

1. Процедура добавления записи в таблицу "Departments".

**create** **procedure** **"pMakePreviousWork"**(

**in** org\_name **varchar**(30)

, **in** start\_date **date**

, **in** end\_date **date**

, **in** staff\_surname **varchar**(25)

, **in** staff\_name **varchar**(15)

, **in** staff\_patronimic **varchar**(25)

, **in** post **varchar**(30)

)

**language** plpgsql

**as** **$$**

**declare**

id **integer**;

id\_staff **integer**;

**begin**

**if** start\_date = **now**()

**then**

**raise** **exception** **'Неправильный формат даты'**;

**end** **if**;

**if** end\_date = **now**()

**then**

**raise** **exception** **'Неправильный формат даты'**;

**end** **if**;

**if** staff\_name **is** **null** **or** staff\_name = **''**

**then**

**raise** **exception** **'Неправильное имя сотрудника'**;

**end** **if**;

**if** staff\_surname **is** **null** **or** staff\_surname = **''**

**then**

**raise** **exception** **'Неправильная фамилия сотрудника'**;

**end** **if**;

**if** staff\_patronimic **is** **null** **or** staff\_patronimic = **''**

**then**

**raise** **exception** **'Неправильное отчество сотрудника'**;

**end** **if**;

**if** post **is** **null** **or** post = **''**

**then**

**raise** **exception** **'Неправильное название должности'**;

**end** **if**;

**select** \* **into** id **from** public.**"fGetNextPreviousWorkID"**();

**select** \*

**into** id\_staff **from** public.**"fGetStaffID"**(staff\_surname, staff\_name, staff\_patronimic);

**if** id\_staff **is** **null**

**then**

**raise** **exception** **'Не найден сотрудник'**;

**end** **if**;

**insert** **into** **"PreviousWorks"** (**"ID"**, **"OrganizationName"**,**"Start"**, **"End"**, **"Post"**, **"StaffID"**)

**values** (id, org\_name, start\_date, end\_date, post, id\_staff);

**raise** **notice** **'Добавлены новые данные о предыдущем месте работы сотрудника'**;

**end**;

**$$**

|  |
| --- |
|  |
|  |
| 1. – Результат вызова процедуры |

Создание Представлений

1. Представление сотрудников имеющих детей

**create** **view** **"vHaveChild"** **as**

**select** s."Surname"

, s."Name"

, s."Patronymic"

, c."Surname" **as** **"ChildSurname"**

, c."Name" **as** **"ChildName"**

, c."Patronymic" **as** **"ChildPatronymic"**

**from** **"Staff"** s

**inner** **join** **"Children"** c

**on** c."StaffID" = s."ID"

|  |
| --- |
|  |
|  |
| 1. – Результат вызова представления |

2. Представление сотрудников, находящихся в отпуске

**create** **view** **"vHaveChild"** **as**

**select** s."Surname"

, s."Name"

, s."Patronymic"

, c."Surname" **as** **"ChildSurname"**

, c."Name" **as** **"ChildName"**

, c."Patronymic" **as** **"ChildPatronymic"**

**from** **"Staff"** s

**inner** **join** **"Children"** c

**on** c."StaffID" = s."ID"

|  |
| --- |
|  |
|  |
| 1. – Результат вызова представления |

Вывод

В результате создания курсового проекта была спроектирована, создана и заполнена база данных для информационного обслуживания работников деканата. Все тестовые запросы к базе данных «ВУЗ» выполняются корректно.

Были изучены операторы SQL на примере диалекта СУБД Postgres Pro, необходимые для фильтрации набора данных; операторы, необходимые для работы с данными, находящимися в разных таблицах, получен навык создания запросов на соединение данных из множества таблиц. Изучены операторы языка определения данных DDL и операторы SQL диалекта PostgreSQL, необходимые для вставки, удаления и модификации данных.

Список литературы

1. Балакина Е. П., Васильева М. А., Филипченко К. М. Информационное обеспечение систем управления. Методические указания к курсовому проектированию. Учебно-методическое пособие. Издание второе, исправленное и дополненное, 2023. 102 –с.
2. Васильева М. А., Степанов Д. Е. Операторы языка определения данных. Учебно– методическое пособие. – М.: РУТ (МИИТ). 2024. – 33с
3. Васильева М. А., Балакина Е. П., Васильева М. В. Соединение данных из множества таблиц. Рекомендации по выполнению работы и перечень типовых заданий. Учебно–методическое пособие. Издание второе, исправленное. – М.: РУТ (МИИТ).2024. – 72с.
4. Васильева М. А, Филипченко К. М., Пугачёв В. А. Фильтрация набора данных. Рекомендации по выполнению работы и перечень типовых заданий: Учебно-методическое пособие. Издание третье, исправленное и дополненное. Москва–2024, 133 с.
5. Васильева М. А., Кузнецов К. К. Операторы языка управления данными. Учебно– методическое пособие. – М.: РУТ (МИИТ).2024. – 36с.