

Министерство образования Республики Беларусь

Учреждение образования
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ**

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

ОТЧЁТ
по лабораторной работе №2
по дисциплине

ПРОЕКТИРОВАНИЕ БАЗ ЗНАНИЙ

Вариант №12

Студент гр. 121701
Руководитель

Р. В. Липский
Н. Г. Липницкая

Минск 2023

СОДЕРЖАНИЕ

1	Постановка задачи	2
1.1	Словесное описание предметной области	2
1.2	Требования к приложению	2
2	Логическое проектирование	3
3	Концептуальное проектирование	4
3.1	Варианты использования	4
3.2	Отношения сущностей	4
4	Физическое проектирование	6
4.1	Схема базы данных	6
4.1.1	Таблица разрядов	6
4.1.2	Таблица работников	6
4.1.3	Таблица выплат	6
4.1.4	Таблица настроек	7
5	Реализация	8
5.1	Тексты запросов	8
5.1.1	Получение всех работников	8
5.1.2	Получение конкретного работника	8
5.1.3	Создание нового работника	8
5.1.4	Изменение работника	8
5.1.5	Удаление работника	8
5.1.6	Получение всех разрядов	8
5.1.7	Получение конкретного разряда	8
5.1.8	Добавление нового разряда	9
5.1.9	Изменение разряда	9
5.1.10	Удаление разряда	9
5.1.11	Подсчёт выплат	9
5.1.12	Получение выплат конкретного работника	10
5.1.13	Получение работников с минимальными зарплатами за месяц	10
6	Интерфейс приложения	11
6.1	Интерфейс для взаимодействия с работниками	11
6.2	Интерфейс для взаимодействия с разрядами	12
6.3	Интерфейс для взаимодействия с выплатами	13
7	Инструментальные средства, использованные при разработке	15
7.1	PostgreSQL	15
7.2	Docker, Docker Compose	15
7.3	Python (psycopg2, tabulate)	15
	Список использованных источников	15

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Словесное описание предметной области

Расчет заработной платы производится для сотрудников предприятия, работающих на бюджетной основе. Для сотрудника определена должность и разряд. В соответствии с единой тарифной сеткой каждый разряд имеет свой коэффициент. Оклад рассчитывается как произведение минимального размера оплаты труда (он может меняться со временем) на коэффициент, соответствующий разряду. Также сотрудник имеет доплату – так называемый дополнительный коэффициент в размере 15 процентов от оклада. Удержания – 13 процентов подоходный налог, 1 процент – пенсионный фонд, 1 процент – профсоюзный взнос, если сотрудник член профсоюза. Необходимо вести расчет заработной платы и накапливать информацию по сотруднику на протяжении всего периода его работы. Необходимо за каждый месяц рассчитывать и хранить сколько всего начислено, удержано и к выплате. Расчет больничных и отпускных дней не вести.

1.2 Требования к приложению

Необходимо реализовать выполнения следующих функций:

- Добавление/редактирование/удаление информации о сотрудниках.
- Добавление/редактирование/удаление информации о расчетных показателях для каждого разряда.
- Просмотр списка коэффициентов, соответствующих разрядам на текущую дату – дата, список коэффициентов – разряд, коэффициент (разрядов 18).
- Просмотр списка сотрудников, имеющих минимальную ЗП за выбранный месяц – дата, список сотрудников - ФИО, должность сотрудника, сумма к выдаче.
- Просмотр списка всех начислений, удержаний, выплат для заданного сотрудника на выбранный месяц – дата, ФИО, должность сотрудника, все начисления, удержания, выплата.

2 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

2.1 Варианты использования

Система имеет два основных пользователя – юрист отдела кадров и бухгалтер.

Юрист отдела кадров занимается поддержанием учета всех работников в системе, их разрядов, должностей, личных данных, а также поддерживает актуальную информацию о разрядах (названия, коэффициента) с целью соответствия информации в системе законодательству и внутренним актам предприятия.

Бухгалтер может просматривать список коэффициентов, соответствующих разрядом для получения актуальной информации просматривать список выплат для конкретного сотрудника, рассчитывать выплаты для всех сотрудников предприятия, а также получать информации о наименьших зарплатах на предприятии (рис. 3.1).



Рисунок 2.1 – Диаграмма вариантов использования

2.2 Отношения сущностей

В базе данных определено четыре таблицы: разряды, работники, выплаты и настройки.

Настройки – таблица вида ключ-значение, где хранятся коэффициенты для расчёта подоходного налога, выплат в пенсионный фонд, профсоюз, минимальный размер оплаты труда.

Разряды – имеют ID, имя и коэффициент для расчёта заработной платы.

Работники – содержат имя, фамилию, отчество, должность, статус нахождения в профсоюзе, а также отношение `grade_id:id`, которое указывает на разряд работника из таблицы разрядов.

Выплаты – содержат дату выплаты, оклад, доплату, подоходный налог, выплату в пенсионный фонд, выплату в пользу профсоюза, начисление, а также отношение `worker_id:id`, которое указывает на работника, к которому относится данная выплата (рис. 3.2).

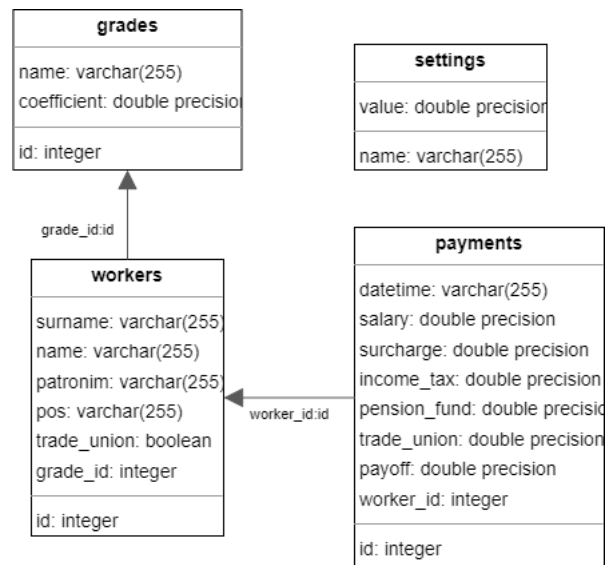


Рисунок 2.2 – ER-диаграмма

3 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Опишем основной бизнес-процесс – расчёт выплат заработной платы для работников. После выхода в систему, пользователь должен решить, желает ли он добавить новых работников в систему (если таковые есть). В случае, если это так, то пользователь вводит данные нового пользователя, а система их сохраняет. Если пользователь желает доавбить еще одного пользователя, то цикл повторяется, иначе пользователь переходит к редактированию разрядов.

В случае, если необходимо отредактировать разряд, пользователь вводит новые данные, если пользователю нужно отредактировать ещё один разряд, цикл повторяется, иначе пользователь переходит к расчёту выплаты.

При расчёте выплат, система собирает список всех работников и рассчитывает их выплаты на основе установленных правил, после чего сохраняет эти выплаты в базу данных. После этого пользователь может просмотреть сохраненные выплаты (рис. 2.1).

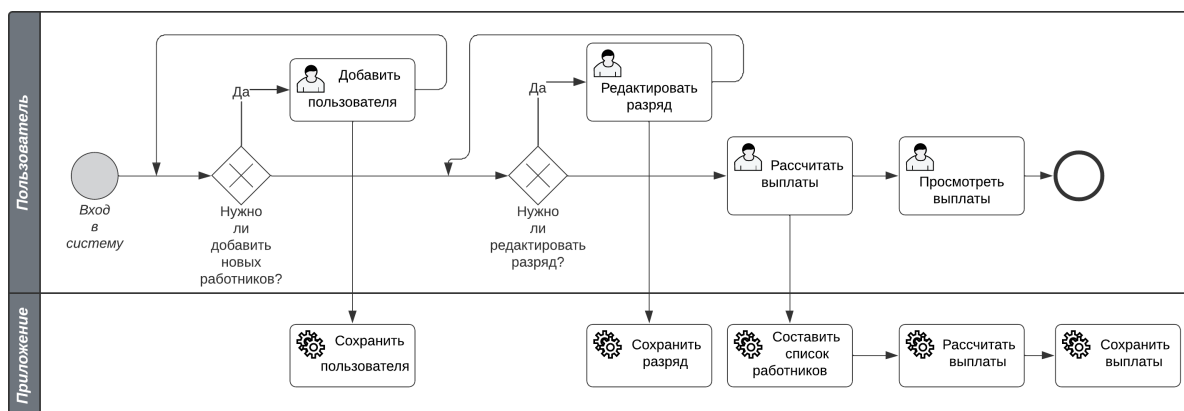


Рисунок 3.1 – BPMN-диаграмма

4 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

4.1 Схема базы данных

4.1.1 Таблица разрядов

```
CREATE TABLE grades(  
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    name VARCHAR(255) NOT NULL,  
    coefficient DOUBLE PRECISION  
);
```

4.1.2 Таблица работников

```
CREATE TABLE workers(  
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    surname VARCHAR(255) NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    patronim VARCHAR(255) NOT NULL,  
    pos VARCHAR(255) NOT NULL,  
    trade_union BOOLEAN NOT NULL,  
  
    grade_id INT REFERENCES grades(id) ON DELETE CASCADE  
);
```

4.1.3 Таблица выплат

```
CREATE TABLE payments(  
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    datetime VARCHAR(255) NOT NULL,  
    salary DOUBLE PRECISION NOT NULL,  
    surcharge DOUBLE PRECISION NOT NULL,  
    income_tax DOUBLE PRECISION NOT NULL,  
    pension_fund DOUBLE PRECISION NOT NULL,  
    trade_union DOUBLE PRECISION NOT NULL,  
    payoff DOUBLE PRECISION  
    GENERATED ALWAYS AS  
        (salary + surcharge - income_tax -  
         pension_fund - trade_union)  
    STORED,  
  
    worker_id INT REFERENCES workers(id)  
);
```

4.1.4 Таблица настроек

```
CREATE TABLE settings(  
    name VARCHAR(255) PRIMARY KEY,  
    value DOUBLE PRECISION NOT NULL  
);
```


5 РЕАЛИЗАЦИЯ

5.1 Тексты запросов

5.1.1 Получение всех работников

```
SELECT w.id, surname, w.name, patronim, pos, trade_union,  
       g.name  
FROM workers w JOIN grades g ON w.grade_id = g.id
```

5.1.2 Получение конкретного работника

```
SELECT w.id, surname, w.name, patronim, pos, trade_union, g.name  
FROM workers w JOIN grades g ON w.grade_id = g.id  
WHERE w.id = {worker_id}
```

5.1.3 Создание нового работника

```
INSERT INTO workers(surname, name, patronim, pos, trade_union,  
                   grade_id)  
VALUES ({surname}, {name}, {patronim}, {position}, {trade_union},  
       {grade_id})
```

5.1.4 Изменение работника

```
UPDATE workers SET  
  surname = '{surname}',  
  name = '{name}',  
  patronim = '{patronim}',  
  pos = '{position}',  
  trade_union = {trade_union},  
  grade_id={group_id}  
WHERE id = {worker_id}
```

5.1.5 Удаление работника

```
DELETE FROM workers WHERE id = {worker_id}
```

5.1.6 Получение всех разрядов

```
SELECT * FROM grades
```

5.1.7 Получение конкретного разряда

```
SELECT * FROM grades WHERE id = {grade_id}
```

5.1.8 Добавление нового разряда

```
INSERT INTO grades(name, coefficient)
VALUES ('{name}', {coefficient})
```

5.1.9 Изменение разряда

```
UPDATE grades SET
    name = '{name}',
    coefficient = {coefficient}
WHERE id = {grade_id}
```

5.1.10 Удаление разряда

```
DELETE FROM grades WHERE id = {grade_id}
```

5.1.11 Подсчёт выплат

```
WITH settings AS (SELECT
    (SELECT value FROM settings WHERE name = 'minimal_payment')
    AS minimal_payment,
    (SELECT value FROM settings WHERE name = 'surcharge')
    AS surcharge,
    (SELECT value FROM settings WHERE name = 'income_tax')
    AS income_tax,
    (SELECT value FROM settings WHERE name = 'pension_fund')
    AS pension_fund,
    (SELECT value FROM settings WHERE name = 'trade_union')
    AS trade_union
)
INSERT INTO payments (datetime, salary, surcharge,
    income_tax, pension_fund, trade_union,
    worker_id)
SELECT
    '{month}',
    minimal_payment * g.coefficient,
    minimal_payment * g.coefficient * surcharge,
    (minimal_payment * g.coefficient) * income_tax,
    (minimal_payment * g.coefficient) * pension_fund,
    (minimal_payment * g.coefficient) *
        CASE WHEN w.trade_union THEN settings.trade_union ELSE 0 END,
    w.id
FROM workers w JOIN grades g ON w.grade_id = g.id
CROSS JOIN settings;
```

5.1.12 Получение выплат конкретного работника

```
SELECT workers.id, surname, workers.name, patronim, pos,  
       workers.trade_union, g.name, p.id, datetime, salary,  
       surcharge, income_tax, pension_fund, p.trade_union,  
       p.payoff  
FROM workers  
     JOIN payments p on workers.id = p.worker_id  
     JOIN grades g ON workers.grade_id = g.id  
WHERE worker_id = {worker_id}
```

5.1.13 Получение работников с минимальными зарплатами за месяц

```
SELECT w.id, w.surname, w.name, w.patronim, w.pos,  
       w.trade_union, g.name, p.payoff  
FROM workers w  
     JOIN payments p ON w.id = p.worker_id  
     JOIN public.grades g on w.grade_id = g.id  
WHERE p.datetime LIKE '{datetime}'  
     AND p.payoff = (  
       SELECT MIN(payoff)  
       FROM payments  
       WHERE datetime LIKE '{datetime}'  
)
```

6 ИНТЕРФЕЙС ПРИЛОЖЕНИЯ

Интерфейс приложения реализован в виде CLI (интерфейс командной строки).

6.1 Интерфейс для взаимодействия с работниками

При входе на страницу Workers, пользователю отображается таблица всех работников, включающая их ID, фамилию, имя, отчество, должность, статус нахождения в профсоюзе и разряд (рис. 6.1).

Workers	Grades	Payday	Settings
Add new worker			
Lowest salary: Oleshkevich Alexey Sergeevich			
Payoff: 0, 1C Developer (Intern)			
Dyatlov Evgeny Alexandrovich			
Cabman (III)			
Protas Alexander Andreevich			
Office Manager (IV)			
Utkin Ilya Alexandrovich			
DevOps Engineer (Tech Lead) , Trade Union Member			
Makarenko Anna Ivanovna			
C/C++ Developer (Team Lead)			
Strongin Andrew Vadimovich			
Python Developer (Team Lead)			
Lipski Rastislau Uladzimiravich			
Java/Kotlin Developer (Senior Architect)			
Oleshkevich Alexey Sergeevich			
1C Developer (Intern)			
Kondratieva Ilona			
Data Scientist (Middle)			
Galuga Maxim Vladimirovich			
Angular/Vue Developer (Middle)			
Chvilev Ilya Alexandrovich			
.NET Developer (Junior) , Trade Union Member			
Korsakova Svetlana Konstantinovna			
Project Manager (Senior)			

Рисунок 6.1 – Получение всех работников

При нажатии на кнопку Add new worker, у пользователя запрашиваются фамилия, имя, отчество, должность, статус нахождения в профсоюзе и разряд нового работника, после чего новый работник сохраняется в базу данных (рис. 6.2).

Add new worker

Surname...	Name...	Patronym...	Position...	<input type="checkbox"/> Trade Union
------------	---------	-------------	-------------	--------------------------------------

|

[Add worker](#)

Рисунок 6.2 – Добавление нового работника

При нажатии на работника, у пользователя запрашиваются новые фамилия, имя, отчество, должность, статус нахождения в профсоюзе и разряд работника. Также можно удалить работника. После ввода всех данных, изменения сохраняются в базу данных (рис. 6.3).

Utkin Ilya Alexandrovich

DevOps Engineer (Tech Lead) , Trade Union Member

Utkin

Ilya

Alexandrovich

DevOps Engineer

☒ Trade Union

Tech Lead

[Save worker](#) [View payments](#) [Delete this worker](#)

Рисунок 6.3 – Изменение работника

6.2 Интерфейс для взаимодействия с разрядами

При входе на страницу Grades, пользователю отображается таблица всех разрядов, включающая в себя название и коэффициент разряда (рис. 6.4).

Workers	Grades	Payday	Settings
Add new grade			
I	Coefficient: 1		
II	Coefficient: 1.15		
III	Coefficient: 1.3		
IV	Coefficient: 1.45		
V	Coefficient: 1.6		
Tech Lead	Coefficient: 15		
Team Lead	Coefficient: 25		
Senior Architect	Coefficient: 30		
Intern	Coefficient: 0		
Middle	Coefficient: 5		
Junior	Coefficient: 2.5		
Senior	Coefficient: 10		

Рисунок 6.4 – Получение всех разрядов

При нажатии на кнопку Add new grade, у пользователя запрашивается название разряда и его коэффициент для расчёта заработной платы. После ввода данных, новый разряд сохраняется в базу данных (рис. 6.5).

Add new grade

Name...

0

Add grade

Рисунок 6.5 – Добавление нового разряда

При нажатии на разряд, у пользователя запрашиваются новое название и коэффициент для расчёта заработной платы. Также есть возможность

удалить разряд. После ввода всех данных, изменения сохраняются в базу данных (рис. 6.6).

Senior Architect

Coefficient: 30

Senior Architect

30

Save grade

Рисунок 6.6 – Редактирование разряда

6.3 Интерфейс для взаимодействия с выплатами

При переходе на вкладку Payday, у пользователя запрашивается месяц, для которого необходимо рассчитать зарплату, после чего рассчитываются выплаты для всех работников в базе данных (рис. 6.7).

Payday

September 2023

Make payments

Рисунок 6.7 – Расчёт выплат

При нажатии кнопки View Payments отображается таблица с фамилией, именем, должностью, статусом нахождения в профсоюзе, ID платежа, месяцем платежа, зарплатой, надбавкой, подоходным налогом, вычетом в пенсионный фонд, вычетом в пользу профсоюза и конечной выплатой (рис. 6.8).

Utkin Ilya Alexandrovich

Grade: Tech Lead
Position: DevOps Engineer
Trade Union

Date	Payment ID	Salary	Surcharge	Income tax	Pension fund	Trade union	Payoff
July 2024	106	5250.00	787.50	682.50	52.50	52.50	5250.00
August 2024	117	5250.00	787.50	682.50	52.50	52.50	5250.00
September 2023	128	5250.00	787.50	682.50	52.50	52.50	5250.00
March 2024	69	5250.00	787.50	682.50	52.50	52.50	5250.00
April 2024	79	5250.00	787.50	682.50	52.50	52.50	5250.00
May 2024	89	5250.00	787.50	682.50	52.50	52.50	5250.00
June 2024	100	5250.00	787.50	682.50	52.50	52.50	5250.00

Рисунок 6.8 – Получение выплат работника

При нажатии на кнопку Lowest salary, у пользователя запрашивается месяц, для которого необходимо найти работников, после чего отображаются

работники с наименьшей выплатой в этот месяц, включающая фамилию, имя, отчество, статус нахождения в профсоюзе, разряд и выплату.

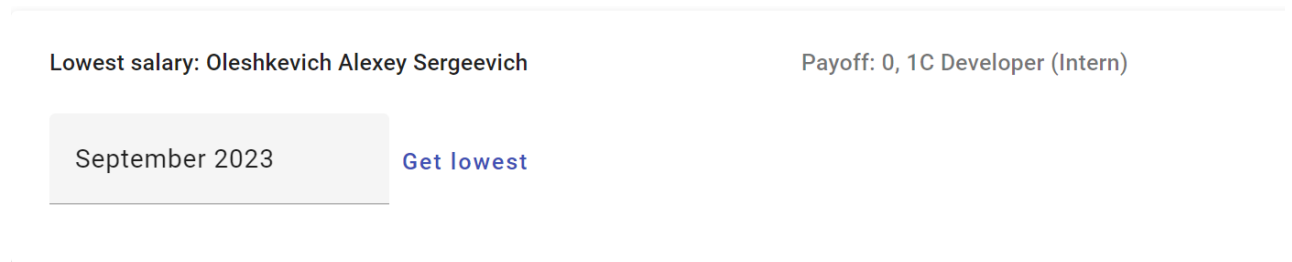


Рисунок 6.9 – Получение работников с наименьшей выплатой в месяце

7 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

7.1 PostgreSQL

В качестве базы данных для реализации приложения была выбрана PostgreSQL - одна из наиболее популярных реляционных баз данных на данный момент. Её преимуществами являются открытый исходный код, большое сообщество, возможность интеграции с большинством современных языков программирования, а также наличие качественной документации.

7.2 Docker, Docker Compose

Для удобства запуска и конфигурации PostgreSQL использовалась технология контейнеризации Docker, которая позволяет быстро и надежно запускать любые программные системы в любых условиях. Преимуществами Docker является кросс-платформенность и большое количество обучающих материалов. При помощи Docker возможно быстро установить и настроить PostgreSQL (или любое иное программное обеспечение).

7.3 Python (psycopg2, tabulate)

Для разработки интерфейса приложения использовался язык Python версии 3.11. Выбор данного языка программирования обусловлен его высокоуровневостью, удобством в разработке и поддержке, наличии большого количества обучающих материалов по нему в сети Интернет.

Внимания также заслуживает огромная коллекция библиотек для Python: например, psycopg2, которая позволяет удобно интегрировать код на Python с базой данных PostgreSQL, а также tabulate, предоставляющая инструменты для отображения таблиц в режиме командой строки.