



Національний технічний університет України

“Київський політехнічний інститут”

БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ

Розрахункова робота на тему:

“Створення бази даних відділу кадрів підприємства”

Виконав:

Студент групи ФБ-74

Варіант 1

Брікс Олексій

Перевірив:

Коломицев М.В.

Київ 2019

Мета роботи: Освоєння методів проектування баз даних і роботи з базами даних у середовищі СУБД MS SQL Server.

Варіант 1

Предметна область – відділ кадрів підприємства.

Задачі, що вирішує інформаційна система – облік інформації о співробітниках і складу їх сімей (для надання податкових пільг)

Завдання на обробку даних:

1. Створити впорядковані по відділах списки:
 - Співробітників з їх дітьми;
 - Співробітників, які мають загальний стаж роботи не менше 30-и років;
 - Співробітників, які мають найвищі оклади у своєму відділі;
 - Співробітників, що мають різностатевих дітей.
2. обчислення:
 - "Відомість на отримання зарплати" (оклад за вирахуванням ПДФО). Якщо є діти, одна мінімальна зарплата податком не обкладається. (ПДФО - податок на доходи фіз.осіб. За замовчуванням - 13%, з можливістю зміни. Мін.зарплата - за замовчуванням 1000.00 грн., з можливістю зміни).
3. корекція:
 - Зміна окладу;
 - Переклад співробітника в інший відділ;
 - Зміна посади.
4. Звіти виду:
 - Бездітні співробітники, угруповання за віковими категоріями (До 30 років, від 30 до 50 років, понад 50 років),
 - Кадрові відомості про співробітника.

1. Завдання Створення діаграми потоків даних (DFD-моделі)

Після аналізу завдання, розробляємо можливу діаграму потоків даних. Для створення DFD-діаграми будемо користуватися сайтом draw.io

Для компактнішого і структурного проектування визначемо декілька підсистем.

Результат роботи видно на рисунках 1, 2, 3, 4.



Рисунок 1. Підсистема “Визначення типу звіту”

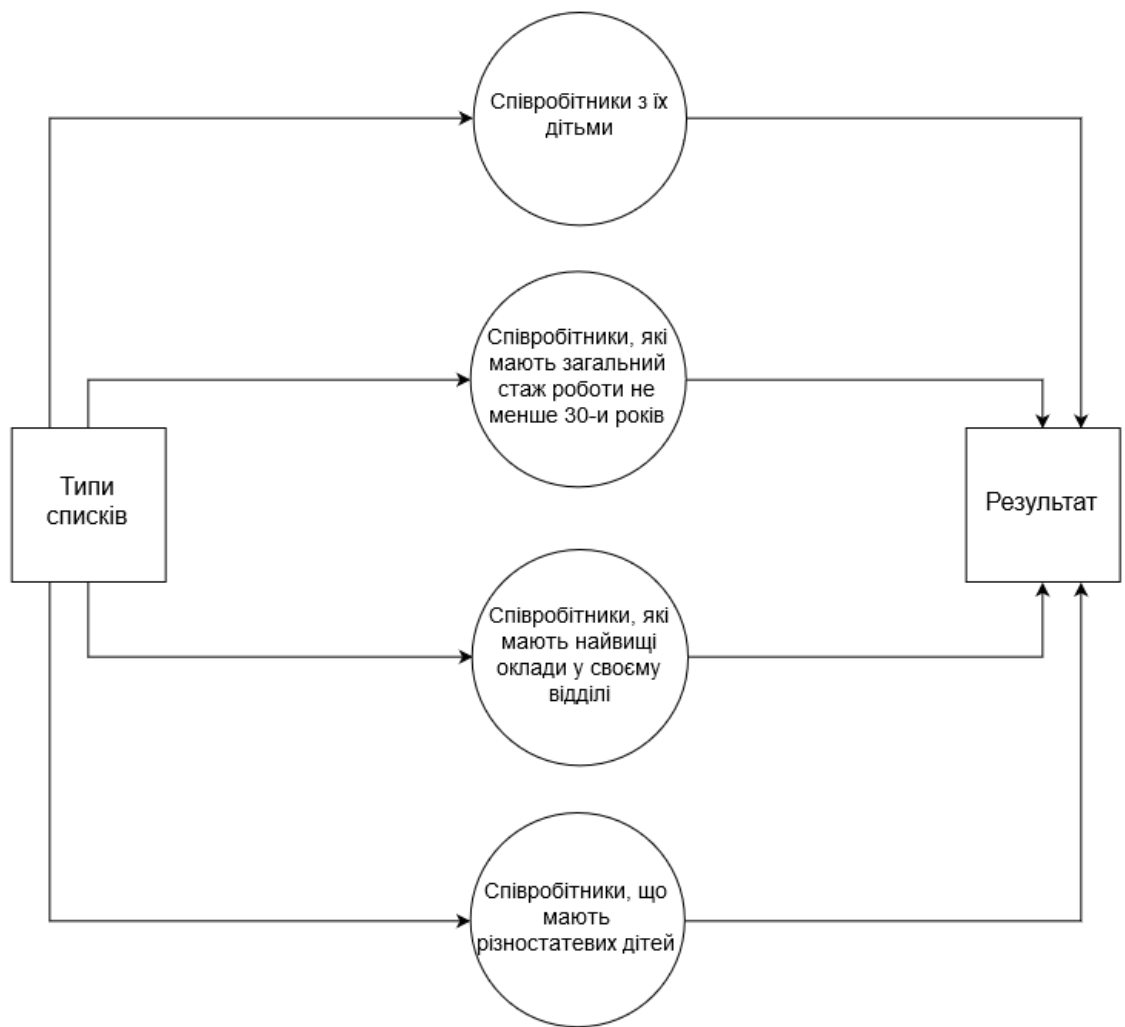


Рисунок 2. Підсистема “Визначення типу списків”

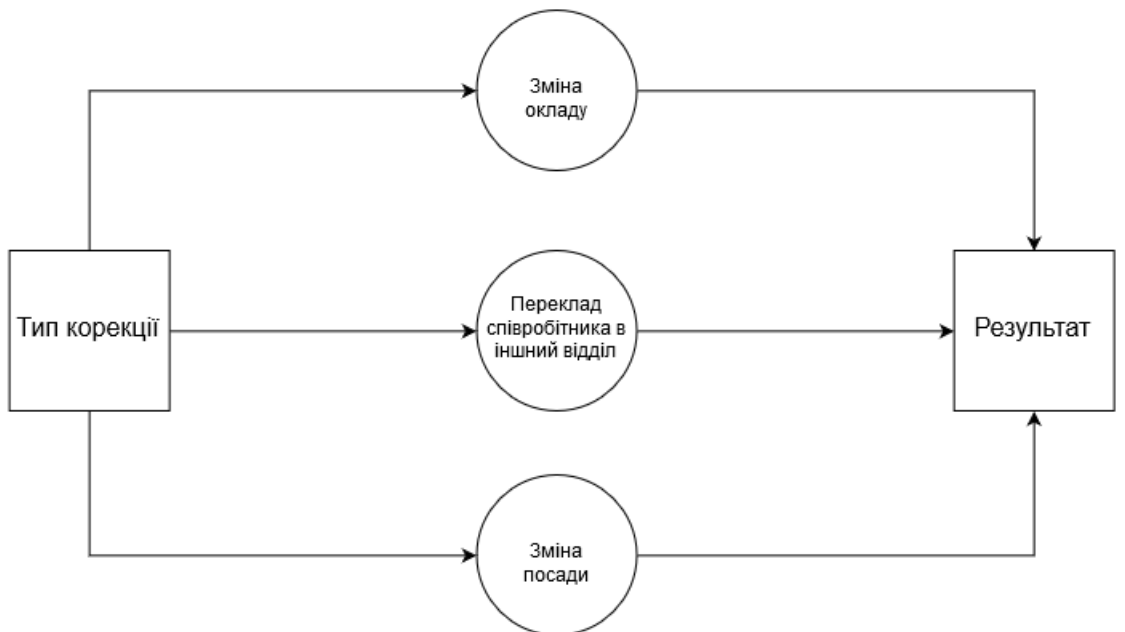


Рисунок 3. Підсистема “Визначення типу корекції”

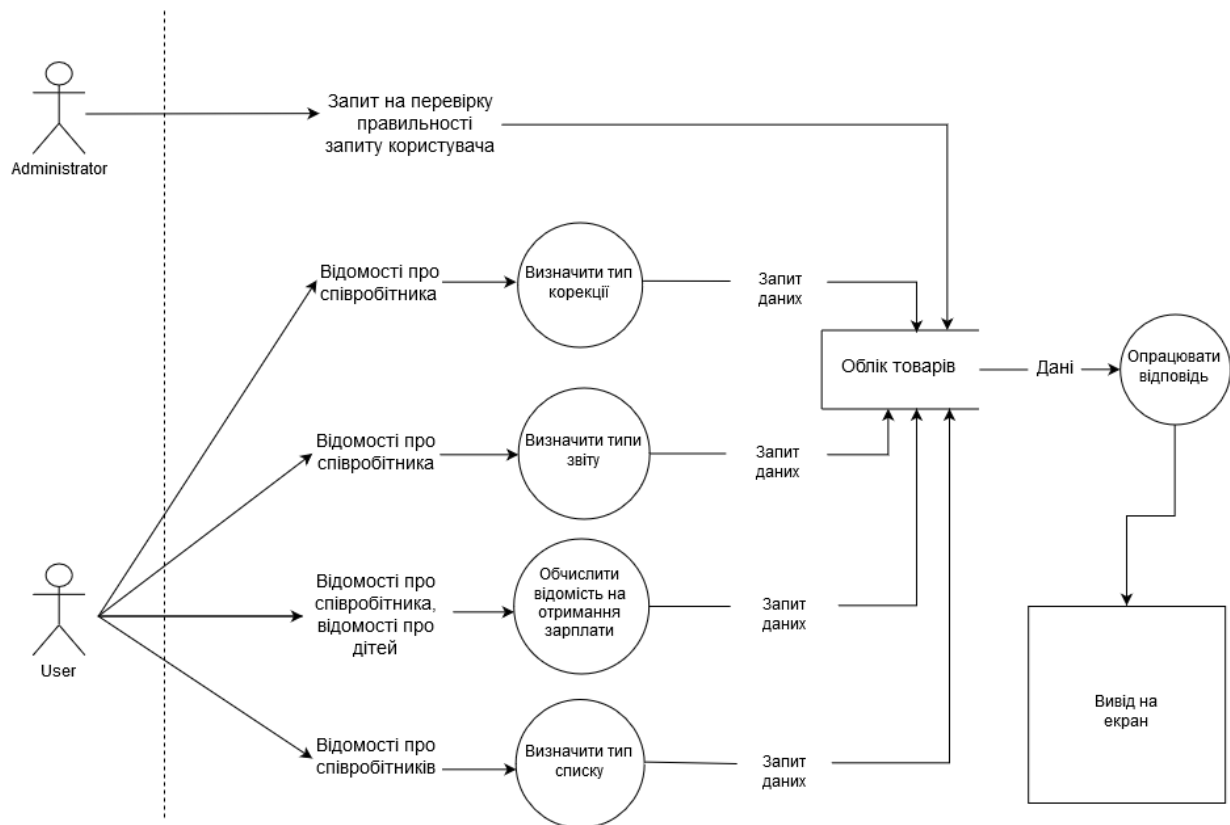


Рисунок 4. Головна діаграма потоків даних

2. Проектування бази даних в MSSQL

```
use rr_var1
go
```

```
drop table employee_child, employee, salary, position, department, department_type
go
```

```
create table department_type (
    department_type_id int primary key identity not null,
    department_type_title varchar(40) not null
)
```

```
create table department (
    department_id int primary key identity not null,
    department_title varchar(40) not null,
    department_type int foreign key references department_type(department_type_id)
)
```

```
create table position (
    position_id int primary key identity not null,
    position_title varchar(40) not null
)
```

```
create table employee (
    employee_id int primary key identity not null,
    employee_firstname varchar(40) not null,
    employee_lastname varchar(40) not null,
    employee_birth_date date not null,
    employee_onboard_date date not null,
    employee_wk_date date not null,
    employee_department int foreign key references department(department_id),
    employee_position int foreign key references position(position_id),
)
```

```
create table employee_child (
```

```

    employee_child_id int primary key identity not null,
    employee_child_firstname varchar(40) not null,
    employee_child_lastname varchar(40) not null,
    employee_child_gender varchar(1) not null
    CONSTRAINT employee_child_gender_check CHECK(employee_child_gender LIKE '[MF]'),
    employee_child_birth_date date not null,
    employee_child_parent int foreign key references employee(employee_id)
)

create table salary (
    salary_id int primary key identity not null,
    salary_position int foreign key references position(position_id),
    salary_department int foreign key references department(department_id),
    salary_value int not null
)

-----Department-Type
insert into department_type (department_type_title) values ('Supply');
insert into department_type (department_type_title) values ('Practice');
insert into department_type (department_type_title) values ('Finance');
-----Department
insert into department (department_title,department_type) values ('Developers', 2);
insert into department (department_title,department_type) values ('QA', 2);
insert into department (department_title,department_type) values ('Operations', 1);
insert into department (department_title,department_type) values ('Security', 1);
insert into department (department_title,department_type) values ('Accounting', 3);
-----Position
insert into position (position_title) values ('Jr. Specialist');
insert into position (position_title) values ('Mid. Specialist');
insert into position (position_title) values ('Sr. Specialist');
insert into position (position_title) values ('Ld. Specialist');
-----Salary
insert into salary(salary_position, salary_department, salary_value) values (1, 1, 1200);
insert into salary(salary_position, salary_department, salary_value) values (1, 2, 1100);
insert into salary(salary_position, salary_department, salary_value) values (1, 3, 1100);
insert into salary(salary_position, salary_department, salary_value) values (1, 4, 1000);
insert into salary(salary_position, salary_department, salary_value) values (1, 5, 1300);
insert into salary(salary_position, salary_department, salary_value) values (2, 1, 2300);
insert into salary(salary_position, salary_department, salary_value) values (2, 2, 2050);
insert into salary(salary_position, salary_department, salary_value) values (2, 3, 2100);
insert into salary(salary_position, salary_department, salary_value) values (2, 4, 1800);
insert into salary(salary_position, salary_department, salary_value) values (2, 5, 2200);
insert into salary(salary_position, salary_department, salary_value) values (3, 1, 3400);
insert into salary(salary_position, salary_department, salary_value) values (3, 2, 3150);
insert into salary(salary_position, salary_department, salary_value) values (3, 3, 3400);
insert into salary(salary_position, salary_department, salary_value) values (3, 4, 2500);
insert into salary(salary_position, salary_department, salary_value) values (3, 5, 3200);
insert into salary(salary_position, salary_department, salary_value) values (4, 1, 4000);
insert into salary(salary_position, salary_department, salary_value) values (4, 2, 3800);
insert into salary(salary_position, salary_department, salary_value) values (4, 3, 3900);
insert into salary(salary_position, salary_department, salary_value) values (4, 4, 3000);
insert into salary(salary_position, salary_department, salary_value) values (4, 5, 3800);
-----Employee
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Пётр', 'Петров', '1998-05-24', '2019-06-06', '2018-08-10',1, 1)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Анастасия', 'Шукина', '1999-03-18', '2019-04-23', '2018-06-11', 2, 1)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Андрей', 'Белов', '1995-02-11', '2019-01-20', '2015-02-10', 3, 1)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Злата', 'Молчанова', '1993-02-11', '2018-11-29', '2012-01-13' ,4, 1)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Ульяна', 'Ковальчук', '1998-08-20', '2019-08-20', '2019-08-20', 5, 1)

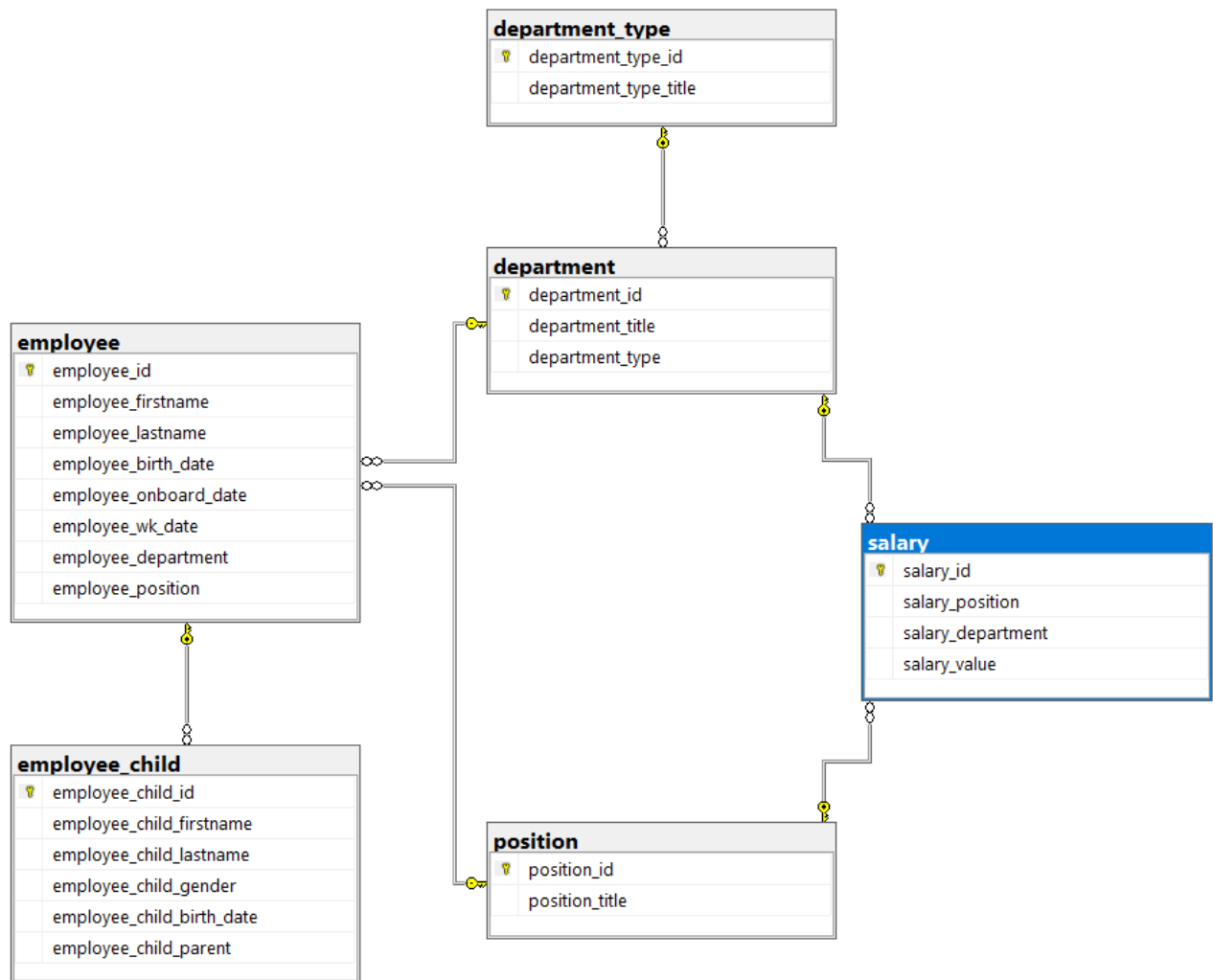
```

```

insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Устин', 'Данилов', '1995-02-12', '2018-05-11', '2015-03-20', 1, 2)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Елена', 'Лыткина', '1996-12-29', '2018-03-20', '2016-03-10', 2, 2)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Константин', 'Жданов', '1995-07-11', '2017-11-21', '2015-02-18', 3, 2)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Проход', 'Кабанов', '1992-01-21', '2017-04-28', '2011-05-20', 4, 2)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Харитон', 'Погомий', '1994-10-14', '2018-03-29', '2013-06-23', 5, 2)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Леонард', 'Яковлев', '1992-04-24', '2018-04-11', '2011-07-03', 1, 3)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Рафаил', 'Рыбаков', '1992-05-10', '2017-10-01', '2011-06-29', 2, 3)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Зинаида', 'Дзюба', '1991-01-16', '2016-05-20', '2010-04-24', 3, 3)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Богдан', 'Палий', '1990-08-09', '2015-10-04', '2009-03-14', 4, 3)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Таисия', 'Шевченко', '1988-05-19', '2015-09-29', '2007-01-11', 5, 3)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Игнатий', 'Суханов', '1985-12-10', '2015-01-05', '2004-01-21', 1, 4)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Яромир', 'Романенко', '1987-06-12', '2015-02-11', '2006-03-22', 2, 4)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Жигер', 'Шухевич', '1983-03-19', '2014-12-21', '2002-04-12', 3, 4)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Андреев', 'Алексей', '1978-05-30', '2014-10-11', '1997-01-21', 4, 4)
insert into
    employee(employee_firstname, employee_lastname, employee_birth_date, employee_onboard_date,
employee_wk_date, employee_department, employee_position)
    values ('Алексей', 'Новиков', '1970-05-30', '2015-01-18', '1988-01-21', 5, 4)
-----Employee-children
insert into employee_child(employee_child_firstname, employee_child_lastname, employee_child_gender,
employee_child_birth_date, employee_child_parent) values ('Андрей', 'Шевченко', 'М', '2010-04-01', 15)
insert into employee_child(employee_child_firstname, employee_child_lastname, employee_child_gender,
employee_child_birth_date, employee_child_parent) values ('Анна', 'Шевченко', 'F', '2003-10-11', 15)
insert into employee_child(employee_child_firstname, employee_child_lastname, employee_child_gender,
employee_child_birth_date, employee_child_parent) values ('Святослав', 'Романенко', 'М', '2015-04-20', 17)
insert into employee_child(employee_child_firstname, employee_child_lastname, employee_child_gender,
employee_child_birth_date, employee_child_parent) values ('Марья', 'Погомий', 'F', '2017-08-29', 10)
insert into employee_child(employee_child_firstname, employee_child_lastname, employee_child_gender,
employee_child_birth_date, employee_child_parent) values ('Анастасия', 'Яковлева', 'F', '2017-08-29', 11)

```

3. ER-діаграма



4. Створення необхідного функціоналу

4.1. Створення впорядкованих списків

А) Співробітників з їх дітьми;

```
create view list_first
as
    select
        emp.employee_firstname, emp.employee_lastname, ec.employee_child_id,
        ec.employee_child_firstname, ec.employee_child_lastname, DATEDIFF(year,
        ec.employee_child_birth_date, getdate()) as employee_child_age
    from
        employee emp
        inner join
            employee_child ec
        on ec.employee_child_parent = emp.employee_id
with check option
go

select * from list_first
```


go

В) Співробітників, які мають загальний стаж роботи не менше 30-и років;

```
create view list_second
as
    select
        emp.employee_firstname, emp.employee_lastname, datediff(year,
emp.employee_wk_date, getdate()) as work_experience
    from
        employee emp
    where
        datediff(year, emp.employee_wk_date, getdate())>=30
with check option
go
```

```
select * from list_second
go
```

С) Співробітників, які мають найвищі оклади у своєму відділі;

```
create view list_third
as
    select
        emp.employee_firstname, emp.employee_lastname, d.department_title, tt.maximum
as employee_salary
    from
        employee emp
    inner join
        position p
    on p.position_id = emp.employee_position
    inner join
        department d
    on d.department_id = emp.employee_department
    inner join
        salary s
    on emp.employee_position = s.salary_position and emp.employee_department
= s.salary_department
    inner join
        (
            select
                d.department_id, max(s.salary_value) as maximum
            from
                employee emp
            inner join
                position p
            on p.position_id = emp.employee_position
            inner join
                department d
            on d.department_id = emp.employee_department
            inner join
                salary s
            on emp.employee_position = s.salary_position and
emp.employee_department = s.salary_department
        ) as tt
    on tt.department_id = d.department_id and s.salary_value = tt.maximum
    group by
        emp.employee_firstname, emp.employee_lastname, d.department_title, tt.maximum
with check option
go
```

```
select * from list_third
go
```

D) Співробітників, що мають різностатевих дітей.

```
create view list_fourth
as
select
    e.employee_firstname, e.employee_lastname
from
    employee e
    inner join
        employee_child ec
        on ec.employee_child_parent = e.employee_id
group by
    e.employee_firstname, e.employee_lastname
having e.employee_firstname in (
select
    e.employee_firstname
from
    employee e
    inner join
        employee_child ec
        on ec.employee_child_parent = e.employee_id
where
    ec.employee_child_gender = 'M'
)
and e.employee_firstname in (
select
    e.employee_firstname
from
    employee e
    inner join
        employee_child ec
        on ec.employee_child_parent = e.employee_id
where
    ec.employee_child_gender = 'F'
)
with check option
go

select * from list_fourth
go
```

4.2 Обчислення

A) "Відомість на отримання зарплати" (оклад за вирахуванням ПДФО).
 Якщо є діти, одна мінімальна зарплата податком не обкладається. (ПДФО -
 податок на доходи фіз.осіб. За замовчуванням - 13%, з можливістю зміни.
 Мін.зарплата - за замовчуванням 1000.00 грн., з можливістю зміни).

```
create proc calculation (
    @employee_id int
)
as
    if exists(
        select
            emp.employee_id, emp.employee_firstname
        from
            employee emp
            inner join
                employee_child ec
```

```

        on ec.employee_child_parent = emp.employee_id
    group by
        emp.employee_id, emp.employee_firstname
    having
        emp.employee_id = @employee_id
)
begin
    print('This employee has child/children. So final salary`ll be: ')
    select
        (s.salary_value/0.87) as salary_without_pdfo
    from
        employee emp
        inner join
            salary s
            on s.salary_position = emp.employee_position and
s.salary_department = emp.employee_department
    where
        emp.employee_id = @employee_id
end;
else
    begin
    print('This employee does not have child/children. So final salary`ll be: ')
    select
        s.salary_value as salary
    from
        employee emp
        inner join
            salary s
            on s.salary_position = emp.employee_position and
s.salary_department = emp.employee_department
    where
        emp.employee_id = @employee_id
    end
go

exec calculation 10
go

```

4.3 Корекція

А) Зміна окладу;

```

create proc correction_first(
    @position_id int = 1,
    @department_id int = 1,
    @salary_value int = 1000
)
as
    begin
        update salary set salary_value = @salary_value where salary_position = @position_id
and salary_department = @department_id
    end
go

exec correction_first 1, 1, 1400
go

```

В) Переклад співробітника в інший відділ;

```

create proc correction_second(
    @employee_id int,
    @employee_department int

```

```

)
as
    begin
    update employee set employee_department = @employee_department where employee_id =
@employee_id
    end
go

exec correction_second 1, 2
go

```

С) Зміна посади.

```

create proc correction_third
(
    @employee_id int,
    @employee_position int
)
as
    begin
    update employee set employee_position = @employee_position where employee_id =
@employee_id
    end
go

exec correction_third 1, 2
go

```

4.4 Створення звіту

А) Бездітні співробітники, угруповання за віковими категоріями (До 30 років, від 30 до 50 років, понад 50 років);

```

select
    emp.employee_firstname, emp.employee_lastname, DATEDIFF(year,
emp.employee_birth_date, GETDATE()) as age
from
    employee emp
where emp.employee_id not in
(
    select
        emp.employee_id
    from
        employee emp
        inner join
            employee_child ec
        on ec.employee_child_parent = emp.employee_id
    group by
        emp.employee_id
) and DATEDIFF(year, emp.employee_birth_date, GETDATE()) between 20 and 30
-- changeable parameter
order by
    age

```

В) Кадрові відомості про співробітника.

```

select
    emp.employee_firstname, emp.employee_lastname, d.department_title,
p.position_title, s.salary_value, tt.number_of_children
from
    employee emp

```

```

inner join
    position p
    on p.position_id = emp.employee_position
inner join
    department d
    on d.department_id = emp.employee_department
inner join
    salary s
    on emp.employee_position = s.salary_position and emp.employee_department
= s.salary_department
left join
(
    select
        emp.employee_id, count(ec.employee_child_id) as
number_of_children
    from
        employee emp
        inner join
            employee_child ec
            on ec.employee_child_parent = emp.employee_id
    group by
        emp.employee_id
) as tt
on emp.employee_id = tt.employee_id

```

5. Перевірка нормалізації розробленої моделі

Для цього необхідно перевірити чи належить дана модель до третьої нормальної форми. Для цього модель повинна належати до першої і другої форми, а також не повинна мати транзитивних відношень, тобто залежностей не ключового атрибуту від іншого не ключового.

Для початку перевіримо належність до першої нормальної форми. Для цього усі атрибути повинні бути атомарні і повинні бути відсутні групи, які повторюються. Виконується.

Далі перевіримо належність до другої нормальної форми. Це буде виконуватись, якщо модель належить до першої нормальної форми (це вже перевірено), а також будуть відсутні неповні функціональні залежності не ключових атрибутів первинного ключа. Виконується.

Залишилось перевірити відсутність транзитивних відношень, тобто ні одне з ключових полів не має ідентифікуватися за допомогою іншого ключового поля. Значить розроблена модель належить до третьої нормальної форми. Виконується.

Висновок

У ході виконання розрахунково-графічної роботи, я дослідив методи проектування баз даних, навчився проектувати власну базу даних на сайті draw.io, а також створювати функціонал для потреб користувача в MS SQL SERVER.