

# Projektowanie Aplikacji Bazodanowych

Jarosław Jędruszcak

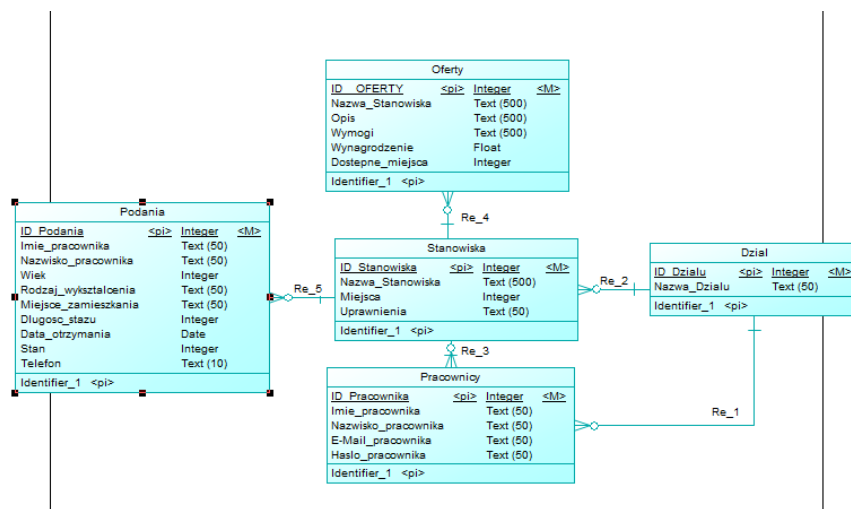
Temat: Obsługa Działu Kadr

Sprawozdanie nr1

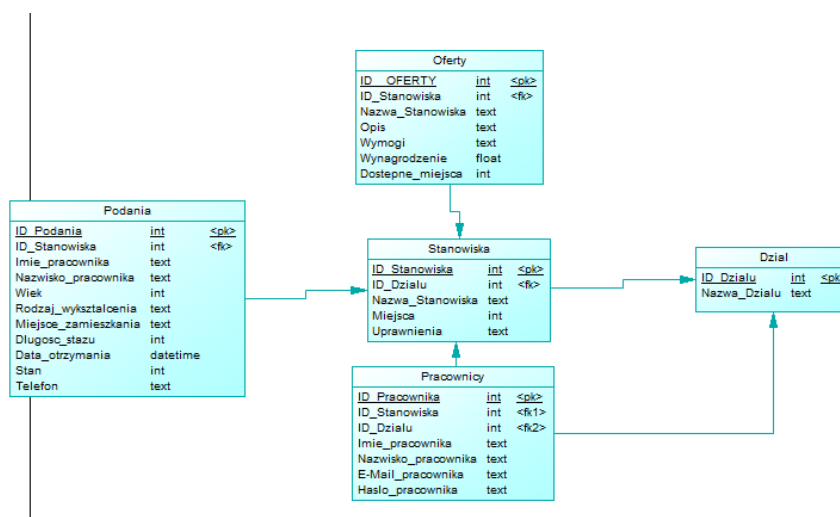
Słownik encji:

- Podania – Encja zawierająca podane podania w odpowiedzi na oferty.
- Oferty - Encja zawierająca zamieszczone oferty o prace, zamieszcza je osoba ze stanowiska posiadającego uprawnienia "rekrutacja".
- Pracownicy – Encja zawierająca liste pracowników pracujących pod stanowiskiem.
- Stanowiska – Encja zawierająca liste dostępnych stanowisk w danym dziale.
- Dział – Encja zawierająca liste Działów.

Model Konceptualny:



Model Fizyczny:



```
/*=====*/
/* DBMS name:   Microsoft SQL Server 2012          */
/* Created on:   06.01.2019 15:18:57                */
/*=====*/
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('OFERTY') and o.name = 'FK_OFERTY_RE_4_STANOWIS')
alter table OFERTY
  drop constraint FK_OFERTY_RE_4_STANOWIS
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('PODANIA') and o.name = 'FK_PODANIA_RE_5_STANOWIS')
alter table PODANIA
  drop constraint FK_PODANIA_RE_5_STANOWIS
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('PRACOWNICY') and o.name = 'FK_PRACOWNI_RE_1_DZIAL')
alter table PRACOWNICY
  drop constraint FK_PRACOWNI_RE_1_DZIAL
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('PRACOWNICY') and o.name =
'FK_PRACOWNI_RE_3_STANOWIS')
alter table PRACOWNICY
  drop constraint FK_PRACOWNI_RE_3_STANOWIS
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')
  where r.fkeyid = object_id('STANOWISKA') and o.name = 'FK_STANOWIS_RE_2_DZIAL')
alter table STANOWISKA
  drop constraint FK_STANOWIS_RE_2_DZIAL
go
```

```
if exists (select 1
  from sysobjects
  where id = object_id('DZIAL')
  and type = 'U')
drop table DZIAL
go
```

```
if exists (select 1
  from sysindexes
  where id = object_id('OFERTY')
```

```
        and name = 'RE_4_FK'
        and indid > 0
        and indid < 255)
drop index OFERTY.RE_4_FK
go
```

```
if exists (select 1
          from sysobjects
          where id = object_id('OFERTY')
          and type = 'U')
drop table OFERTY
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('PODANIA')
          and name = 'RE_5_FK'
          and indid > 0
          and indid < 255)
drop index PODANIA.RE_5_FK
go
```

```
if exists (select 1
          from sysobjects
          where id = object_id('PODANIA')
          and type = 'U')
drop table PODANIA
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('PRACOWNICY')
          and name = 'RE_1_FK'
          and indid > 0
          and indid < 255)
drop index PRACOWNICY.RE_1_FK
go
```

```
if exists (select 1
          from sysindexes
          where id = object_id('PRACOWNICY')
          and name = 'RE_3_FK'
          and indid > 0
          and indid < 255)
drop index PRACOWNICY.RE_3_FK
go
```

```
if exists (select 1
          from sysobjects
          where id = object_id('PRACOWNICY')
          and type = 'U')
drop table PRACOWNICY
```

go

```
if exists (select 1
            from sysindexes
            where id = object_id('STANOWISKA')
            and name = 'RE_2_FK'
            and indid > 0
            and indid < 255)
drop index STANOWISKA.RE_2_FK
go
```

```
if exists (select 1
            from sysobjects
            where id = object_id('STANOWISKA')
            and type = 'U')
drop table STANOWISKA
go
```

```
/*=====*/
/* Table: DZIAL */
/*=====*/
create table DZIAL (
    ID_DZIALU int not null,
    NAZWA_DZIALU text null,
    constraint PK_DZIAL primary key nonclustered (ID_DZIALU)
)
go
```

```
/*=====*/
/* Table: OFERTY */
/*=====*/
create table OFERTY (
    ID_OFERTY int not null,
    ID_STANOWISKA int not null,
    NAZWA_STANOWISKA text null,
    OPIS text null,
    WYMOGI text null,
    WYNAGRODZENIE float null,
    DOSTEPNE_MIEJSCA int null,
    constraint PK_OFERTY primary key nonclustered (ID_OFERTY)
)
go
```

```
/*=====*/
/* Index: RE_4_FK */
/*=====*/
create index RE_4_FK on OFERTY (
    ID_STANOWISKA ASC
)
go
/*=====*/
```

```

/* Table: PODANIA */
/*=====*/
create table PODANIA (
  ID_PODANIA      int          not null,
  ID_STANOWISKA   int          not null,
  IMIE_PRACOWNIKA text         null,
  NAZWISKO_PRACOWNIKA text     null,
  WIEK            int          null,
  RODZAJ_WYKSZTALCENIA text    null,
  MIEJSCE_ZAMIESZKANIA text    null,
  DLUGOSC_STAZU   int          null,
  DATA_OTRZYMANIA datetime    null,
  STAN            int          null,
  TELEFON         text         null,
  constraint PK_PODANIA primary key nonclustered (ID_PODANIA)
)
go

/*=====*/
/* Index: RE_5_FK */
/*=====*/
create index RE_5_FK on PODANIA (
  ID_STANOWISKA ASC
)
go

/*=====*/
/* Table: PRACOWNICY */
/*=====*/
create table PRACOWNICY (
  ID_PRACOWNIKA   int          not null,
  ID_STANOWISKA   int          not null,
  ID_DZIALU        int          not null,
  IMIE_PRACOWNIKA text         null,
  NAZWISKO_PRACOWNIKA text     null,
  E_MAIL_PRACOWNIKA text       null,
  HASLO_PRACOWNIKA text        null,
  constraint PK_PRACOWNICY primary key nonclustered (ID_PRACOWNIKA)
)
go

/*=====*/
/* Index: RE_3_FK */
/*=====*/
create index RE_3_FK on PRACOWNICY (
  ID_STANOWISKA ASC
)
go

/*=====*/
/* Index: RE_1_FK */
/*=====*/

```

```
create index RE_1_FK on PRACOWNICY (  
ID_DZIALU ASC  
)  
go
```

```
/*=====*/  
/* Table: STANOWISKA */  
/*=====*/
```

```
create table STANOWISKA (  
ID_STANOWISKA int not null,  
ID_DZIALU int not null,  
NAZWA_STANOWISKA text null,  
MIEJSCA int null,  
UPRAWNIENIA text null,  
constraint PK_STANOWISKA primary key nonclustered (ID_STANOWISKA)  
)  
go
```

```
/*=====*/  
/* Index: RE_2_FK */  
/*=====*/
```

```
create index RE_2_FK on STANOWISKA (  
ID_DZIALU ASC  
)  
go
```

```
alter table OFERTY  
add constraint FK_OFERTY_RE_4_STANOWIS foreign key (ID_STANOWISKA)  
references STANOWISKA (ID_STANOWISKA)  
go
```

```
alter table PODANIA  
add constraint FK_PODANIA_RE_5_STANOWIS foreign key (ID_STANOWISKA)  
references STANOWISKA (ID_STANOWISKA)  
go
```

```
alter table PRACOWNICY  
add constraint FK_PRACOWNI_RE_1_DZIAL foreign key (ID_DZIALU)  
references DZIAL (ID_DZIALU)  
go
```

```
alter table PRACOWNICY  
add constraint FK_PRACOWNI_RE_3_STANOWIS foreign key (ID_STANOWISKA)  
references STANOWISKA (ID_STANOWISKA)  
go
```

```
alter table STANOWISKA  
add constraint FK_STANOWIS_RE_2_DZIAL foreign key (ID_DZIALU)  
references DZIAL (ID_DZIALU)  
go
```

Podstawowe rodzaje zapytań:

1. Otrzymać listę pracowników wyznaczonego działu.

```
SELECT Pracownicy.*, Dzial.Nazwa_Dzialu
FROM Dzial INNER JOIN Pracownicy ON Dzial.ID_Dzialu = Pracownicy.ID_Dzial
WHERE ((Dzial.Nazwa_Dzialu)="TU WYZNACZAM DZIAŁ");
```

2. Otrzymać listę kierowników działów.

```
SELECT Pracownicy.*, Stanowiska.Uprawnienia
FROM Stanowiska INNER JOIN Pracownicy ON Stanowiska.ID_Stalowiska =
Pracownicy.ID_Stalowiska
WHERE ((Stanowiska.Uprawnienia)="Kierownik");
```

3. Otrzymać listę posadowych stanowisk działu.

```
SELECT Oferty.*, Stanowiska.Nazwa_Stalowiska, Dzial.Nazwa_Dzialu
FROM (Dzial INNER JOIN Stanowiska ON Dzial.ID_Dzialu = Stanowiska.ID_Dzialu) INNER
JOIN Oferty ON Stanowiska.ID_Stalowiska = Oferty.ID_Stalowiska
WHERE ((Dzial.Nazwa_Dzialu)="TU WYZNACZAM DZIAŁ");
```

4. Otrzymać listę wolnych etatów.

```
SELECT Stanowiska.*
FROM Stanowiska
WHERE (Stanowiska.Miejsca > 0);
```

5. Otrzymać listę zgłoszonych ofert na wyznaczony termin

```
SELECT Podania.*
FROM Podania
WHERE ((Podania.Data_Otrzymania) = "TU WYZNACZAM TERMIN")
```

6. Otrzymać listę oraz liczbę osób zatrudnionych na odpowiednich stanowiskach w różnych działach.

```
SELECT Pracownicy.Imie_pracownika, Stanowiska.Nazwa_Stalowiska
FROM Stanowiska INNER JOIN Pracownicy ON Stanowiska.ID_Stalowiska = Pracownicy.ID_Stalowiska
WHERE (((Stanowiska.Nazwa_Stalowiska)="TU PODAC NAZWE"));
```

```
SELECT Count(Pracownicy.Imie_pracownika) AS PoliczOfImie_pracownika,
Stanowiska.Nazwa_Stalowiska
FROM Stanowiska INNER JOIN Pracownicy ON Stanowiska.ID_Stalowiska =
Pracownicy.ID_Stalowiska
GROUP BY Stanowiska.Nazwa_Stalowiska
HAVING (((Stanowiska.Nazwa_Stalowiska)="TU PODAC NAZWE"));
```

7. Otrzymać listę kandydatów na odpowiedni stanowiska według zgłoszonych ofert.

```
SELECT Podania.*, Stanowiska.Nazwa_Stalowiska
FROM Stanowiska INNER JOIN Podania ON Stanowiska.ID_Stalowiska =
Podania.ID_Stalowiska
WHERE (((Stanowiska.Nazwa_Stalowiska)="TU PODAC NAZWE"));
```