

Bazy danych/ Inżynieria Oprogramowania

Dokumentacja Bazy Danych Projektu pt.: „System rezerwacji sal”

Data wykonania: 15.06.2020

Grupa: L4
Kacper Kopczacki
Kamil Łyczko

Spis treści

| | |
|---|---|
| 1. Zapis ról przydzielonych w grupie projektowej..... | 3 |
| 2. Etapy rozwoju diagramów..... | 3 |
| 3. Kod DDL tworzący bazę danych..... | 7 |
| Podsumowanie..... | 9 |

1. Zapis ról przydzielonych w grupie projektowej.

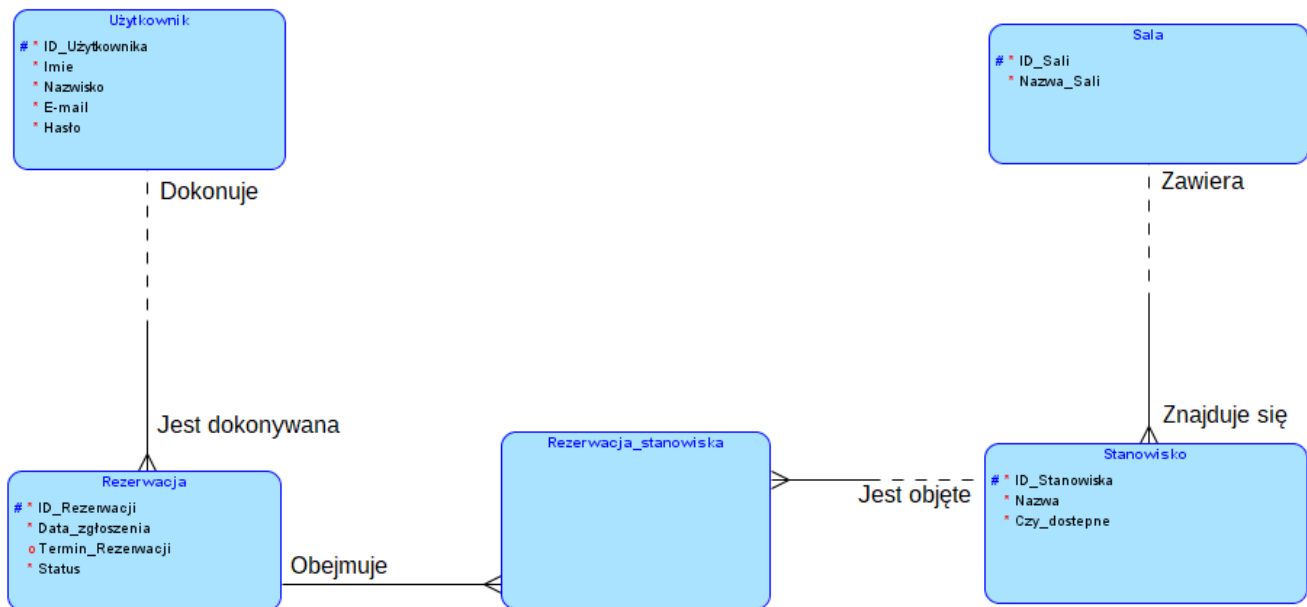
Zaprojektowaniem oraz zaimplementowaniem bazy danych zajęli się Kacper Kopczacki oraz Kamil Łyczko:

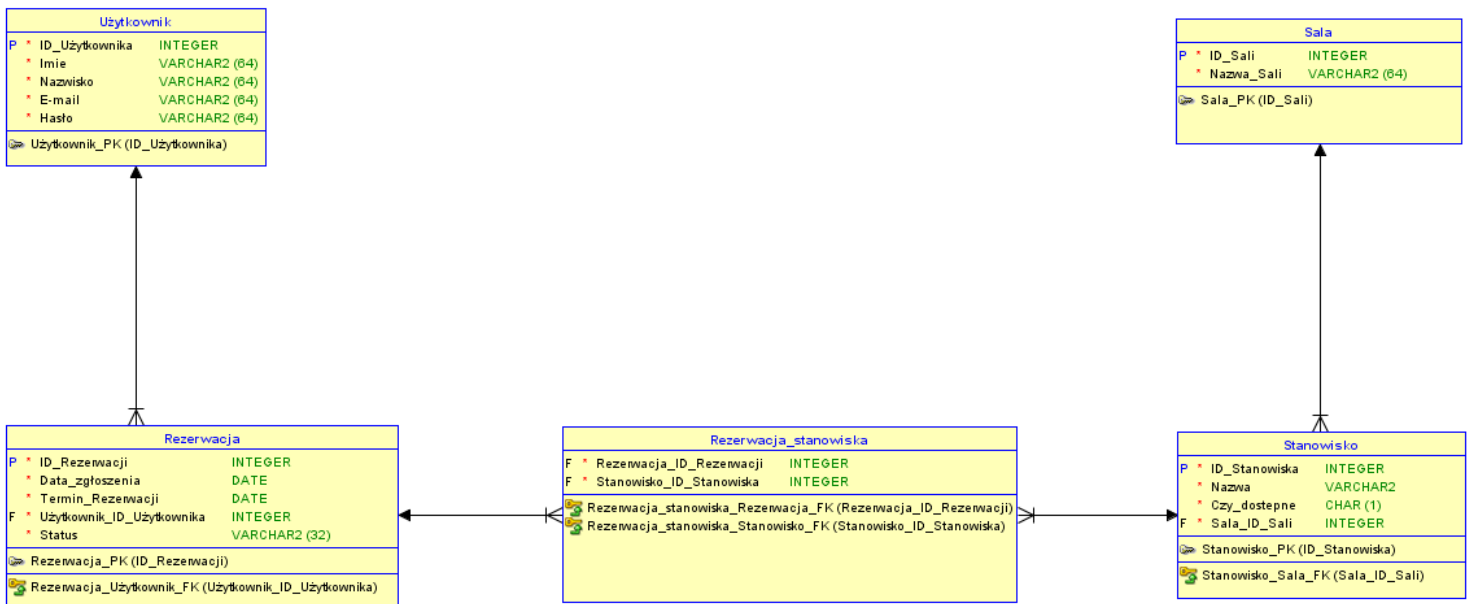
- Tworzenie diagramów ERD oraz Bazy Danych,
- Przygotowanie kodu DDL tworzącego bazę danych,
- Stworzenie tabel w bazie danych,
- Stworzenie modelu na podstawie zdalnej bazy danych.

Standardem wykorzystywanym przy tworzeniu bazy danych był PostgreSQL.

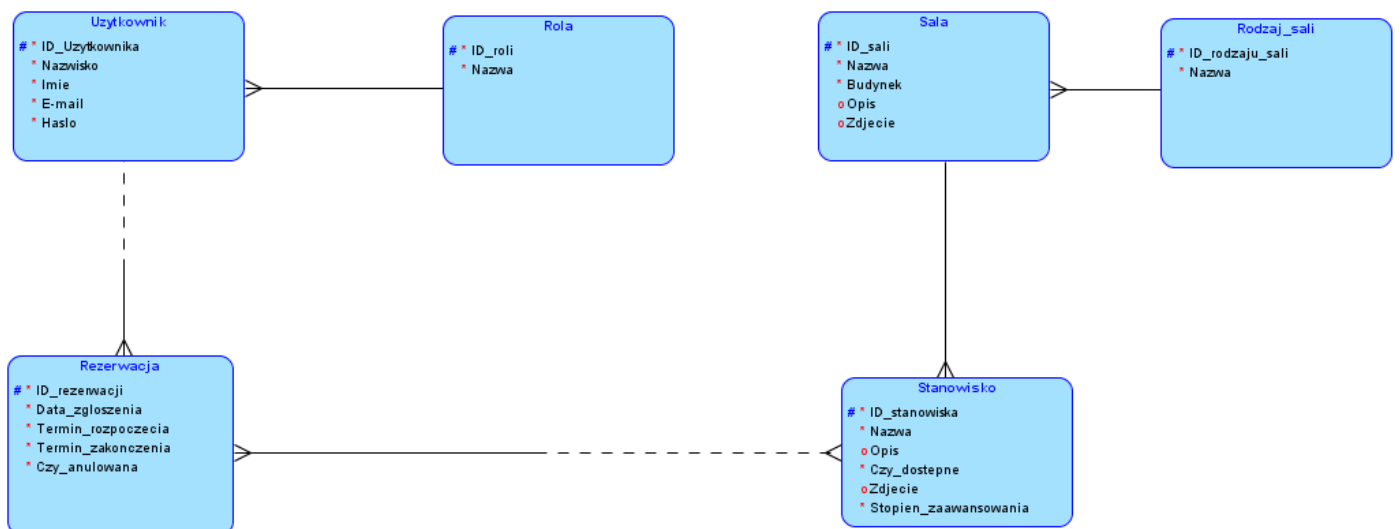
2. Etapy rozwoju diagramów.

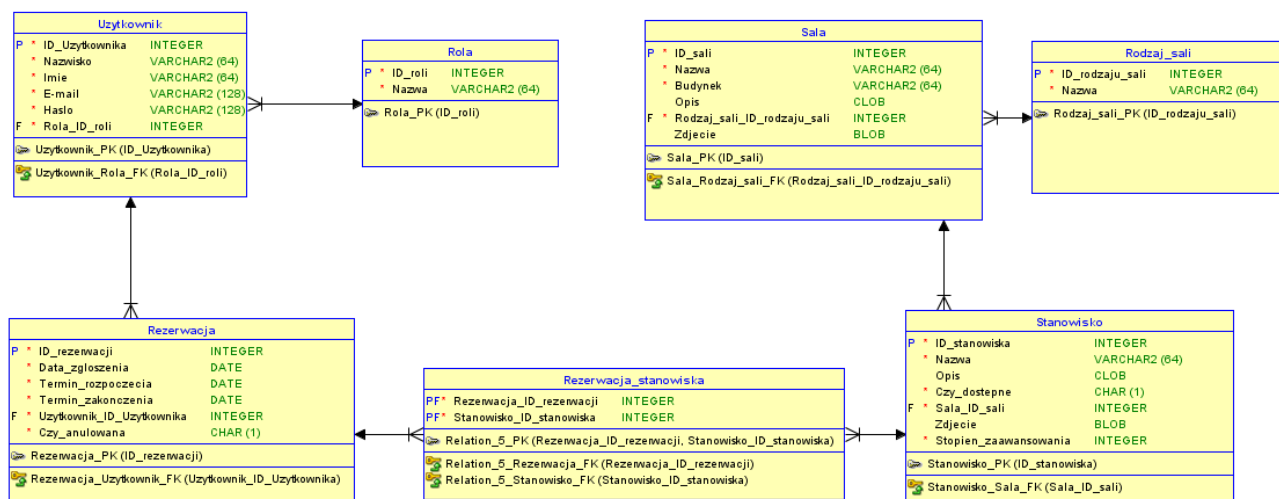
Etap 1:



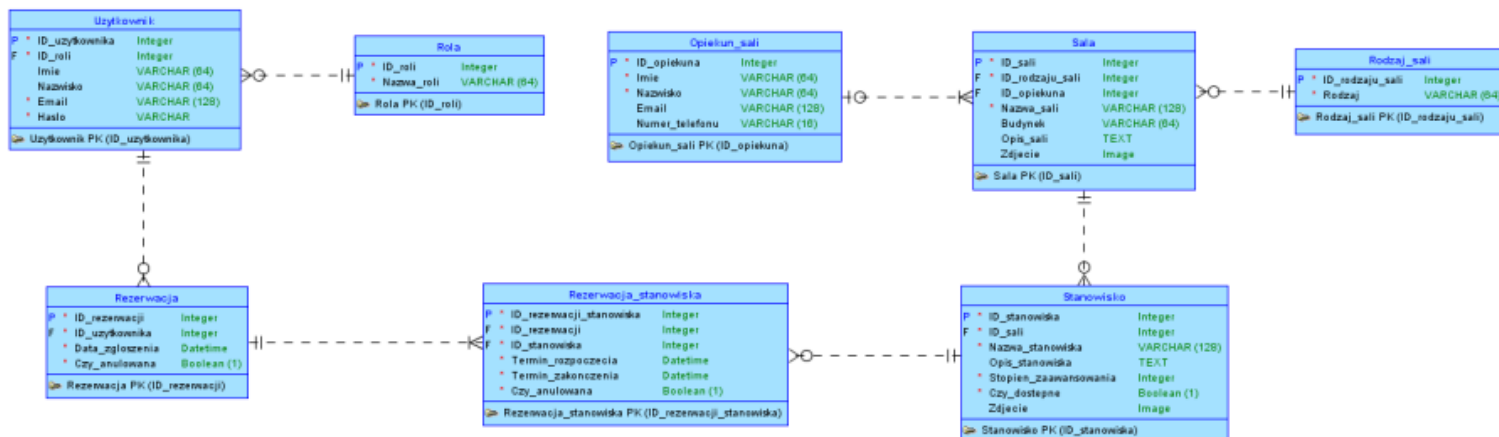


Etap 2:





Etap 3:



Etap 4 (postać ostateczna):

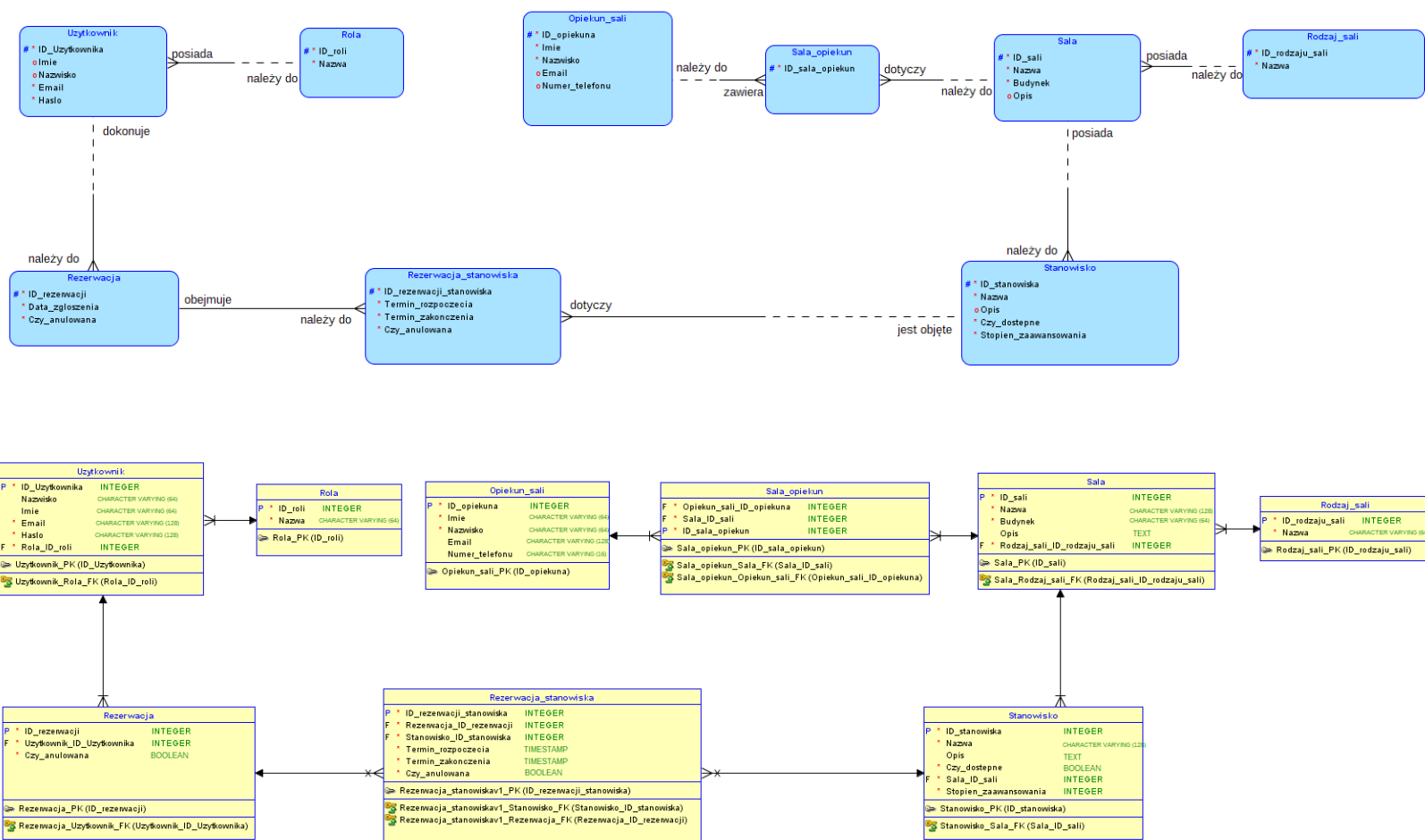


Tabela użytkownik przechowuje dane użytkowników korzystających z systemu. Tabela Rola służy do przechowywania ról nadawanych użytkownikom. Tabela Sala przechowuje dane o pracownikach i innych pomieszczeniach użytkowych w ProtoLab. Sala posiada rodzaj, który jest zawarty w encji Rodzaj_sali. Tabela Opiekun_sali zawiera dane wszystkich pracowników ProtoLab, którzy są opiekunami sal. Tabela Sala_opiekun zawiera informacje o przydzielach opiekunów do sal. Encja Stanowisko posiada dane dotyczące każdego stanowiska znajdujących się w salach. Tabela Rezerwacja posiada dane użytkownika, który dokonuje rezerwacji. Jedna rezerwacja może obejmować wiele rezerwacji stanowisk w różnych terminach, które są zawarte w tabeli Rezerwacja_stanowiska.

3. Kod DDL tworzący bazę danych.

```
CREATE TABLE sala_opiekun (  
    id_sala_opiekun    SERIAL PRIMARY KEY,  
    id_sali            INTEGER NOT NULL,  
    id_opiekuna_sali   INTEGER NOT NULL  
);
```

```
CREATE TABLE opiekun_sali (  
    id_opiekuna    SERIAL PRIMARY KEY,  
    imie          VARCHAR(64) NOT NULL,  
    nazwisko      VARCHAR(64) NOT NULL,  
    email         VARCHAR(128),  
    numer_telefonu VARCHAR(16)  
);
```

```
CREATE TABLE rezerwacja (  
    id_rezerwacji    SERIAL PRIMARY KEY,  
    id_uzytkownika   INTEGER NOT NULL,  
    data_zgloszenia  timestamp(3) NOT NULL,  
    czy_anulowana    boolean NOT NULL  
);
```

```
CREATE TABLE rezerwacja_stanowiska (  
    id_rezerwacji_stanowiska SERIAL PRIMARY KEY,  
    id_rezerwacji           INTEGER NOT NULL,  
    id_stanowiska           INTEGER NOT NULL,  
    termin_rozpoczecia     timestamp(3) NOT NULL,  
    termin_zakonczenia     timestamp(3) NOT NULL,  
    czy_anulowana          boolean NOT NULL  
);
```

```
CREATE TABLE rodzaj_sali (  
    id_rodzaju_sali SERIAL PRIMARY KEY,  
    rodzaj          VARCHAR(64) NOT NULL  
);
```

```
CREATE TABLE rola (  
    id_rola    SERIAL PRIMARY KEY,  
    nazwa_rola VARCHAR(64) NOT NULL  
);
```



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



**Katedra
Informatyki i Automatyki**
Politechnika Rzeszowska

```
CREATE TABLE sala (  
    id_sali      SERIAL PRIMARY KEY,  
    id_rodzaju_sali  INTEGER NOT NULL,  
    nazwa_sali   VARCHAR(128) NOT NULL,  
    budynek      VARCHAR(64),  
    opis_sali    text,  
);
```

```
CREATE TABLE stanowisko (  
    id_stanowiska SERIAL PRIMARY KEY,  
    id_sali        INTEGER NOT NULL,  
    nazwa_stanowiska  VARCHAR(128) NOT NULL,  
    opis_stanowiska   text,  
    stopien_zaaawansowania  INTEGER NOT NULL,  
    czy_dostepne      boolean NOT NULL,  
);
```

```
CREATE TABLE uzytkownik (  
    id_uzytkownika SERIAL PRIMARY KEY,  
    id_roli         INTEGER NOT NULL,  
    imie           VARCHAR(64),  
    nazwisko       VARCHAR(64),  
    email          VARCHAR(128) NOT NULL,  
    haslo          VARCHAR(128) NOT NULL  
);
```

```
ALTER TABLE sala_opiekun  
    ADD CONSTRAINT sala_opiekun_id_opiekuna_fk FOREIGN KEY ( id_opiekuna_sali )  
        REFERENCES opiekun_sali ( id_opiekuna )  
        ON DELETE CASCADE  
        ON UPDATE CASCADE;
```

```
ALTER TABLE sala_opiekun  
    ADD CONSTRAINT sala_opiekun_id_sali_fk FOREIGN KEY ( id_sali )  
        REFERENCES sala ( id_sali )  
        ON DELETE CASCADE  
        ON UPDATE CASCADE;
```

```
ALTER TABLE Rezerwacja_stanowiska  
    ADD CONSTRAINT rezerwacja_stanowiska_rezerwacja_fk FOREIGN KEY ( id_rezerwacji )  
        REFERENCES rezerwacja ( id_rezerwacji )  
        ON DELETE CASCADE  
        ON UPDATE CASCADE;
```

```
ALTER TABLE Rezerwacja_stanowiska  
    ADD CONSTRAINT rezerwacja_stanowiska_stanowisko_fk FOREIGN KEY ( id_stanowiska )  
        REFERENCES stanowisko ( id_stanowiska )
```



```
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
ALTER TABLE Rezerwacja  
ADD CONSTRAINT rezerwacja_uzytkownik_fk FOREIGN KEY ( id_uzytkownika )  
REFERENCES uzytkownik ( id_uzytkownika )  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
ALTER TABLE Sala  
ADD CONSTRAINT sala_opiekun_sali_fk FOREIGN KEY ( id_opiekuna )  
REFERENCES opiekun_sali ( id_opiekuna )  
ON DELETE SET NULL  
ON UPDATE CASCADE;
```

```
ALTER TABLE Sala  
ADD CONSTRAINT sala_rodzaj_sali_fk FOREIGN KEY ( id_rodzaju_sali )  
REFERENCES rodzaj_sali ( id_rodzaju_sali )  
ON DELETE RESTRICT  
ON UPDATE CASCADE;
```

```
ALTER TABLE Stanowisko  
ADD CONSTRAINT stanowisko_sala_fk FOREIGN KEY ( id_sali )  
REFERENCES sala ( id_sali )  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
ALTER TABLE Uzytkownik  
ADD CONSTRAINT uzytkownik_rola_fk FOREIGN KEY ( id_rola )  
REFERENCES rola ( id_rola )  
ON DELETE RESTRICT  
ON UPDATE CASCADE;
```

Opis powyższego kodu:

Poleceniami CREATE TABLE tworzymy tabele na wzór encji widocznych na diagramie ERD. Poleceniami ALTER TABLE definiujemy klucze obce tabel widoczne w diagramie bazy danych. Zostały również ustawione reguły postępowania w przypadku wykonywania operacji DELETE oraz UPDATE dla konkretnego obiektu z bazy danych.

Podsumowanie

Tabele bazy danych zostały wygenerowane na zdalnym serwerze z wykorzystaniem narzędzia pgAdmin. Wszelkie operacje dodawania, usuwania, pobierania i edytowania danych działają poprawnie. Operacje są możliwe do wykonywania z poziomu aplikacji.