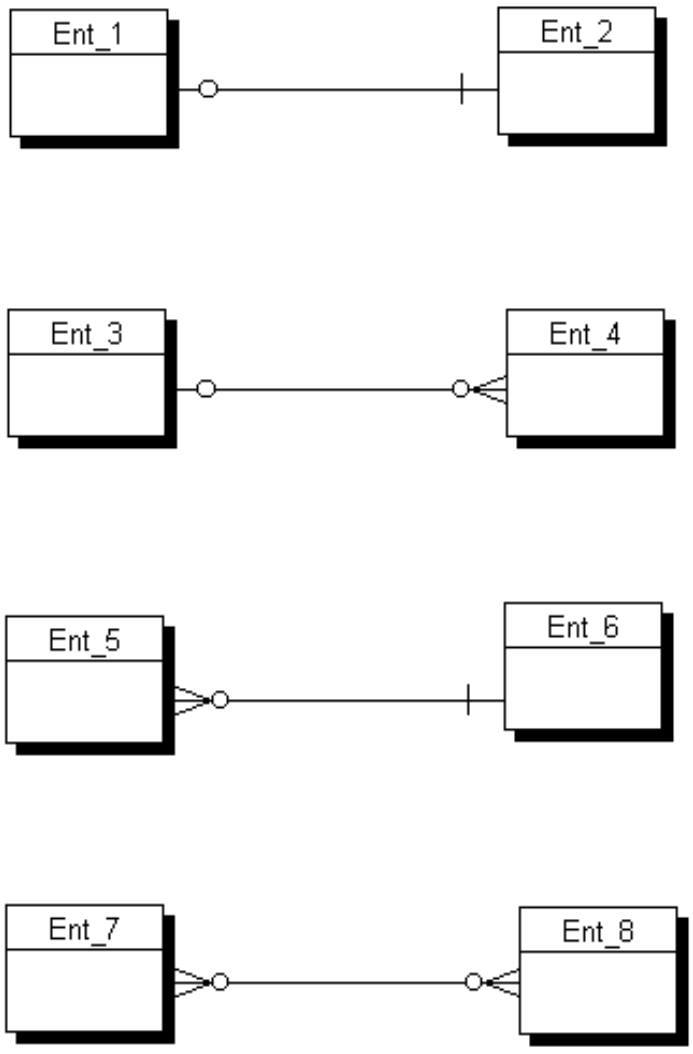
Tworzenie modelu logicznego i fizycznego danych.

W celu stworzenia modelu danych wykorzystamy program Data Architect wchodzący w skład pakietu narzędzi CASE Power Designer, który pozwala utworzyć tzw. *logiczny* (*konceptualny model danych)* (CDM), *fizyczny model danych* (PDM) oraz wygenerować skrypt SQL tworzący bazę danych.

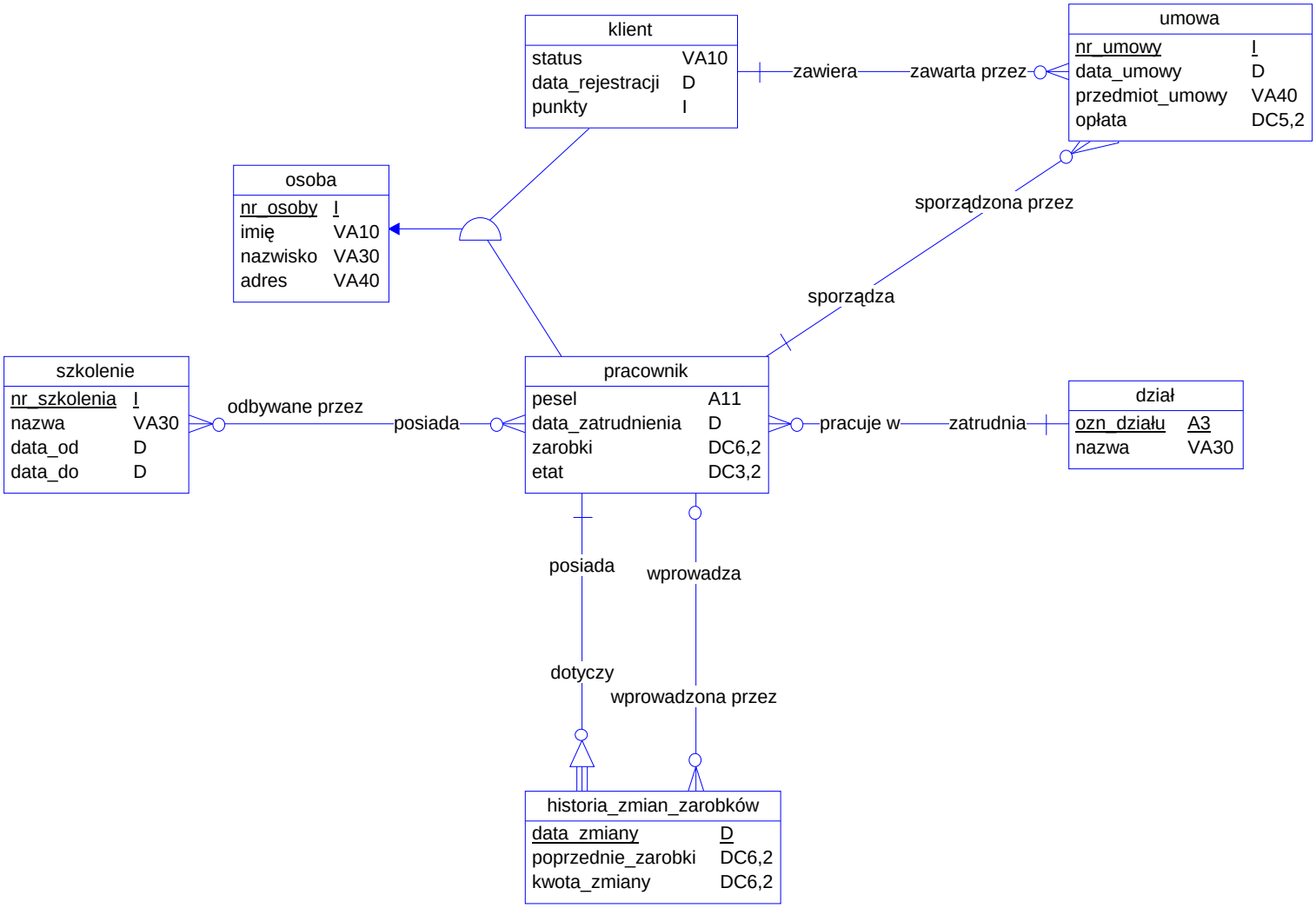
Tworzenie modelu logicznego.

Na wstępie określa się zbiory **encji** oraz ich **atrybuty** (wraz z określeniem typu danych, wymagalności, ograniczeń) i klucze główne. Pomiędzy tak zdefiniowanymi zbiorami encji kreśli się relacje o określonych własnościach. Wszystko to odbywa się w trybie graficznym. Rysunek 1 przedstawia symbole wykorzystywane na schematach modeli CDM (prostokąty odpowiadają encjom, związki są prezentowane za pomocą linii łączących odpowiednie encje).



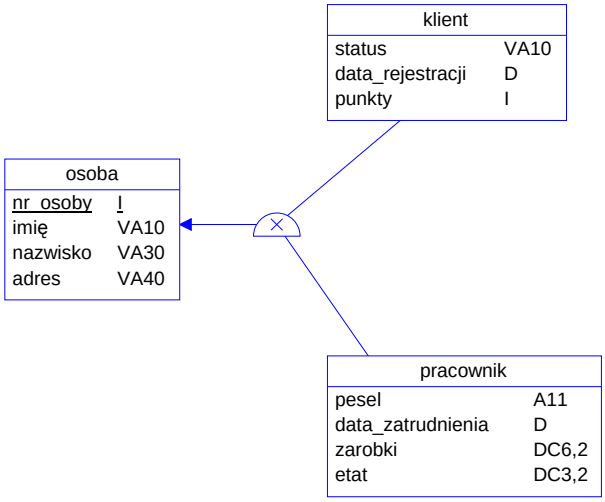
Rys.1. Symbole związków stosowane w ***modelu logicznym (CDM):*** jeden do jednego (wymagany z jednej strony), jeden do wielu (nie wymagany z żadnej strony),wiele do jednego (wymagany z jednej strony), wiele do wielu (nie wymagany).

Rysunek 2 przedstawia przykładowy model logiczny bazy danych do rejestracji danych klientów i pracowników, z wyszczególnieniem atrybutów poszczególnych encji (wraz z ich dziedzinami), kluczy głównych (podkreślone atrybuty) oraz związków pomiędzy encjami.



Rys. .2 Schemat przykładowego modelu logicznego utworzony za pomocą programu Data Architect z pakietu Power Designer.

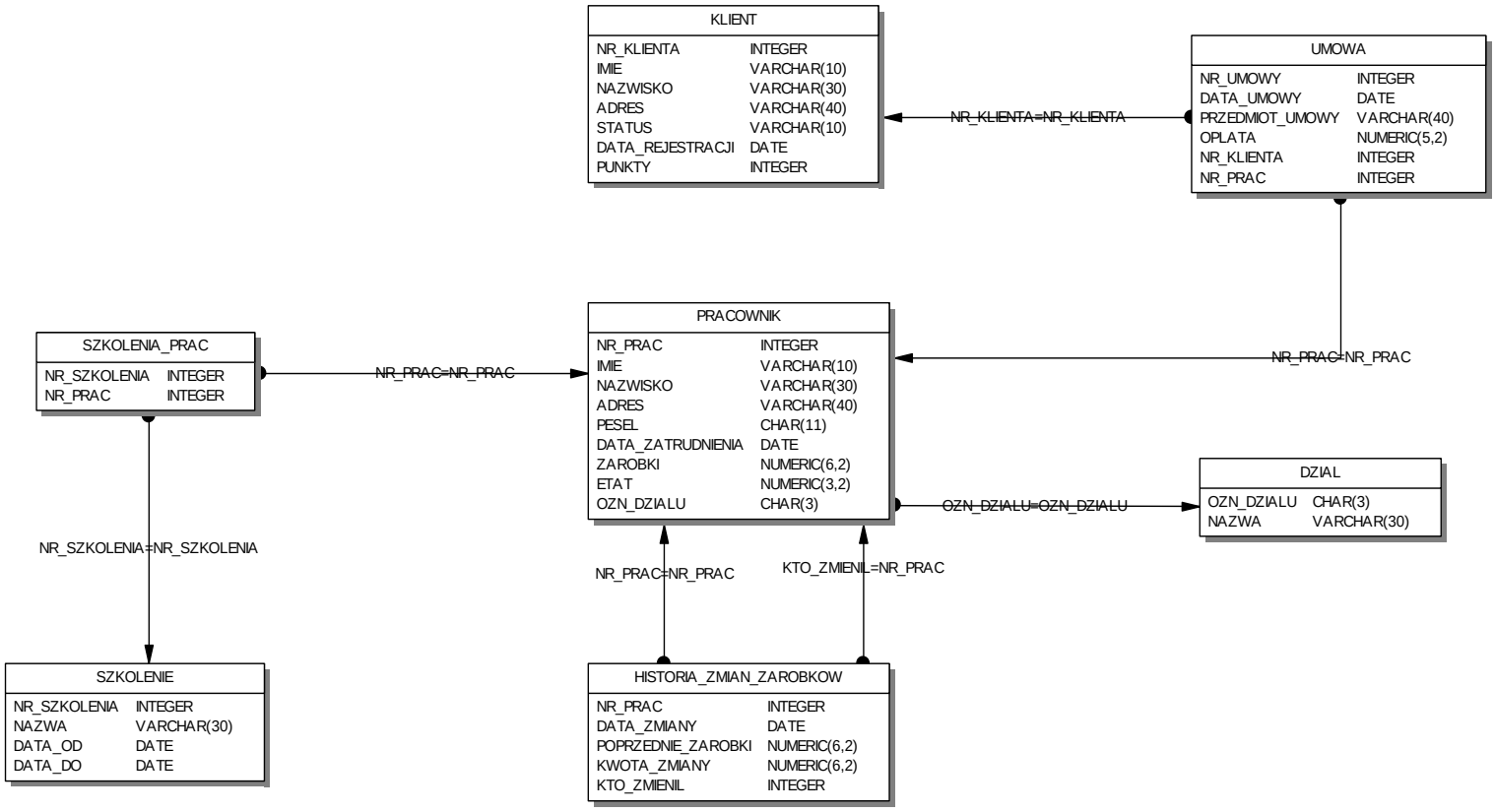
Encja OSOBA posiada dwa **podtypy** – KLIENT oraz PRACOWNIK. Oznacza to, że dana osoba może być klientem lub pracownikiem. Związki tego typu określa się mianem *dziedziczenia*. W powyższym przykładzie osoba może być jednocześnie pracownikiem i klientem, natomiast rysunek 3 przedstawia dziedziczenie, w którym osoba może być **albo** pracownikiem, **albo** klientem, tzn. oba te podtypy się wzajemnie wykluczają.



Rys. 3. Przykład dziedziczenia.

Tworzenie modelu fizycznego.

Model logiczny jest podstawą do wygenerowania tzw. fizycznego modelu danych (PDM) - Rys.4. Model ten uwzględnia konkretny system baz danych. W tym przypadku – *Interbase 4.x*.



Rys. 4. Przykładowy fizyczny model danych (PDM).

W modelu fizycznym tabele zostały uzupełnione o kolumny **kluczy obcych** – na podstawie odpowiednich związków istniejących pomiędzy encjami w modelu fizycznym. Istnienie związku wiele do wielu pomiędzy encjami PRACOWNIK oraz SZKOLENIE spowodowało powstanie dodatkowej tabeli SZKOLENIA_PRAC.

Tworzenie skryptu SQL

Na podstawie modelu fizycznego można wygenerować, w postaci pliku tekstowego, skrypt SQL opisujący schemat bazy danych (poniżej).

```
/* ============================================================
*/
/*   Database name:  MODEL_1                                  */
/*   DBMS name:      InterBase 4.0                            */
/*   Created on:     2008-02-15  07:16                        */
/* ============================================================
*/

/* ============================================================
*/
/*   Table: KLIENT                                            */
/* ============================================================
*/
create table KLIENT
(
   NR_KLIENTA       INTEGER              not null,
   IMIE             VARCHAR(10)                ,
   NAZWISKO         VARCHAR(30)                ,
   ADRES            VARCHAR(40)                ,
   STATUS           VARCHAR(10)                ,
   DATA_REJESTRACJI DATE                       ,
   PUNKTY           INTEGER                    ,
   constraint PK_KLIENT primary key (NR_KLIENTA)
);

/* ============================================================
*/
/*   Table: SZKOLENIE                                         */
/* ============================================================
*/
create table SZKOLENIE
(
   NR_SZKOLENIA     INTEGER              not null,
   NAZWA            VARCHAR(30)                ,
   DATA_OD          DATE                       ,
   DATA_DO          DATE                       ,
   constraint PK_SZKOLENIE primary key (NR_SZKOLENIA)
);
```

```
/* ==============================================================
*/
/*   Table: DZIAL                                */
/* ==============================================================
*/
create table DZIAL
(
   OZN_DZIALU         CHAR(3)              not null,
   NAZWA              VARCHAR(30)                 ,
   constraint PK_DZIAL primary key (OZN_DZIALU)
);

/* ==============================================================
*/
/*   Table: PRACOWNIK                            */
/* ==============================================================
*/
create table PRACOWNIK
(
   NR_PRAC            INTEGER              not null,
   IMIE               VARCHAR(10)                 ,
   NAZWISKO           VARCHAR(30)                 ,
   ADRES              VARCHAR(40)                 ,
   PESEL              CHAR(11)                    ,
   DATA_ZATRUDNIENIA  DATE                        ,
   ZAROBKI            DECIMAL(6,2)                ,
   ETAT               DECIMAL(3,2)                ,
   OZN_DZIALU         CHAR(3)              not null,
   constraint PK_PRACOWNIK primary key (NR_PRAC)
);

/* ==============================================================
*/
/*   Table: UMOWA                                */
/* ==============================================================
*/
create table UMOWA
(
   NR_UMOWY           INTEGER              not null,
   DATA_UMOWY         DATE                        ,
   PRZEDMIOT_UMOWY    VARCHAR(40)                 ,
   OPLATA             DECIMAL(5,2)                ,
   NR_KLIENTA         INTEGER              not null,
   NR_PRAC            INTEGER              not null,
   constraint PK_UMOWA primary key (NR_UMOWY)
);
```

```
/* ==========================================================
*/
/*   Table: HISTORIA_ZMIAN_ZAROBKOW                         */
/* ==========================================================
*/
create table HISTORIA_ZMIAN_ZAROBKOW
(
   NR_PRAC           INTEGER              not null,
   DATA_ZMIANY       DATE                 not null,
   POPRZEDNIE_ZAROBKI  DECIMAL(6,2)                ,
   KWOTA_ZMIANY      DECIMAL(6,2)                ,
   KTO_ZMIENIL       INTEGER                     ,
   constraint PK_HISTORIA_ZMIAN_ZAROBKOW primary key (NR_PRAC, DATA_ZMIANY)
);

/* ==========================================================
*/
/*   Table: SZKOLENIA_PRAC                                  */
/* ==========================================================
*/
create table SZKOLENIA_PRAC
(
   NR_SZKOLENIA      INTEGER              not null,
   NR_PRAC           INTEGER              not null,
   constraint PK_SZKOLENIA_PRAC primary key (NR_SZKOLENIA, NR_PRAC)
);

alter table PRACOWNIK
   add constraint FK_PRAC_DZIAL foreign key  (OZN_DZIALU)
      references DZIAL;

alter table UMOWA
   add constraint FK_UMOWA_KLIENT foreign key  (NR_KLIENTA)
      references KLIENT;

alter table UMOWA
   add constraint FK_UMOWA_PRAC foreign key  (NR_PRAC)
      references PRACOWNIK;

alter table HISTORIA_ZMIAN_ZAROBKOW
   add constraint FK_HISTORIA_ZMIANY_PRAC foreign key  (NR_PRAC)
      references PRACOWNIK;

alter table HISTORIA_ZMIAN_ZAROBKOW
   add constraint FK_HISTORIA_KTO_ZMIEN foreign key  (KTO_ZMIENIL)
      references PRACOWNIK(NR_PRAC);

alter table SZKOLENIA_PRAC
   add constraint FK_ODBYWANE_SZKOL foreign key  (NR_SZKOLENIA)
      references SZKOLENIE;

alter table SZKOLENIA_PRAC
   add constraint FK_SZKOL_PRAC foreign key  (NR_PRAC)
      references PRACOWNIK;
```