Transformacja modelu EER do modelu relacyjnego

Pojęcia podstawowe

- Schemat logiczny bazy danych
  - Zbiór schematów relacji, na którym operują aplikacje bazy danych.
- Relacja (tablica)
  - Dwuwymiarowa tablica. Kolumny tablicy nazywamy atrybutami. Wiersze tablicy nazywamy krotkami - każda krotka reprezentuje wystąpienie encji.
- Atrybut
  - Cecha lub własność encji - kolumna tablicy
- Klucz potencjalny
  - Atrybut lub zbiór atrybutów jednoznacznie identyfikujący krotki relacji
- Klucz podstawowy
  - Atrybut lub zbiór atrybutów - wybrany spośród kluczy potencjalnych
- Klucz obcy
  - Atrybut lub zbiór atrybutów będący kluczem podstawowym innej relacji

Reguły transformacji (1)

Podstawowe reguły transformacji modelu EER do modelu relacyjnego generują trzy typy schematów relacji:

1. **Schemat relacji encji**

   Schemat relacji encji jest generowany zawsze dla encji występujących w związkach binarnych typu: wiele-do-wiele, jeden-do-wiele (po stronie "jeden"), lub jeden-do-jeden (po jednej wybranej stronie); dla encji występujących w związkach binarnych rekursywnych typu wiele-do-wiele; oraz dla encji występujących w związkach ternarnych lub związkach wyższych stopni relationship, oraz dla hierarchii generalizacji

Reguły transformacji (2)

2. **Schemat relacji encji z kluczem obcym**

   Schemat relacji encji z kluczem obcym jest generowany zawsze dla encji występujących w związkach binarnych typu jeden-do-wiele (po stronie "wiele") lub jeden-do-jeden (po jednej wybranej stronie); dla encji występujących w związkach binarnych rekursywnych typu jeden-do-wiele lub jeden-do-jeden

Reguły transformacji (3)

3. **Schemat relacji związku**

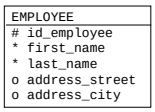
   Schemat relacji związku jest generowany zawsze dla związków binarnych typu wiele-do-wiele, dla związków binarnych rekursywnych typu wiele-do-wiele, oraz wszystkich związków ternarnych oraz związków wyższych stopni

Reguły transformacji (4)

4. **Następujące reguły odnoszą się do wartości pustych generowanych regułami transformacji:**

- Wartości puste są dozwolone w relacjach encji z kluczem obcym dla kluczy obcych odpowiadających encjom należącym do klasy przynależności opcjonalnej
- Wartości puste są zabronione w relacjach encji z kluczem obcym dla kluczy obcych odpowiadających encjom należącym do klasy przynależności obowiązkowej
- Wartości puste są zabronione w relacjach związków dla atrybutów należących do kluczy podstawowych tych relacji

Transformacja encji



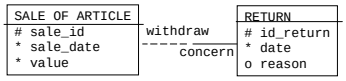
EMPLOYEE (
id_employee *PRIMARY KEY*,
first_name *NOT NULL*,
last_name *NOT NULL*,
address_street *NULL*,
address_city *NULL*)

Reguły transformacji:

- Nazwa encji jest odwzorowywana w nazwę schematu relacji
- Atrybuty encji są odwzorowywane w nazwy atrybutów schematu relacji
- Unikalny identyfikator encji jest transformowany w klucz podstawowy schematu relacji
- Obowiązkowość atrybutów encji jest reprezentowana w schemacie relacji w postaci ograniczenia *NOT NULL* zdefiniowanego na atrybucie schematu relacji
- Opcjonalność atrybutów encji jest reprezentowana w schemacie relacji w postaci własności *NULL* zdefiniowanej na atrybucie schematu relacji
- Pozostałe ograniczenia integralnościowe zdefiniowane dla atrybutów encji są reprezentowane w sposób identyczny dla atrybutów schematu relacji

Transformacja związku 1:1

Związek 1:1 jednostronnie obowiązkowy



Reguły transformacji:

- Klucz obcy jest dodawany do schematu relacji po stronie „obowiązkowy"
- Ograniczenia referencyjne są definiowane dla klucza obcego

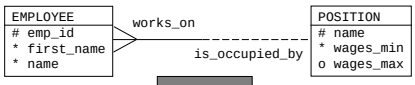
SALES (
sale_id *PRIMARY KEY*,
sale_date *NOT NULL*,
value *NOT NULL*)
RETURNS (
id_return *PRIMARY KEY*,
date *NOT NULL*,
reason *NULL*,
id_sale *NOT NULL*
***REFERENCES* sales(sale_id)**)

Związek 1:1 dwustronnie opcjonalny

Reguły transformacji:

- Klucz obcy jest dodawany do schematu relacji o mniejszej liczbie krotek
- Ograniczenia referencyjne są definiowane dla klucza obcego

Transformacja związku 1:N

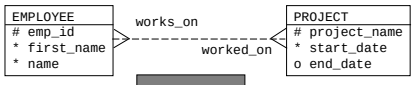


POSITIONS (
name *PRIMARY KEY*,
wages_min *NOT NULL*,
wages_max *NULL*)
EMPLOYEES (
emp_id *PRIMARY KEY*,
first_name *NOT NULL*,
name *NOT NULL*,
position *NOT NULL*,
***REFERENCES* positions(name)**
)

Reguły transformacji:

- Klucz obcy jest dodawany do schematu relacji po stronie „wiele"
- Ograniczenia referencyjne są definiowane dla klucza obcego
- Obowiązkowość związku po stronie "wiele" jest reprezentowana przez ograniczenie *NOT NULL* definiowane na kluczu obcym schematu relacji
- Opcjonalność związku po stronie "wiele" jest reprezentowana przez własność *NULL* definiowaną na kluczu obcym schematu relacji
- Opcjonalność lub obowiązkowość związku po stronie "jeden" nie jest definiowana w modelu relacyjnym

Transformacja związku M:N



EMPLOYEES (
emp_id *PRIMARY KEY*,
first_name *NOT NULL*,
name *NOT NULL*)
PROJECTS (
project_name *PRIMARY KEY*,
start_date *NOT NULL*,
end_date *NULL*)
WORKS-ON (
emp_id ***REFERENCES* employees(emp_id),**
proj_name ***REFERENCES* projects(project_name),**
PRIMARY KEY(emp_id, proj_name))

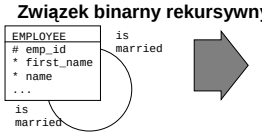
Reguły transformacji:

- Utwórz schemat relacji związku dla związku typu M:N
- Nowy schemat relacji zawiera klucze obce odpowiadające kluczom podstawowym wiązanych encji
- Ograniczenia referencyjne są definiowane dla kluczy obcych
- Klucze obce tworzą klucz podstawowy utworzonego schematu relacji związku

Transformacja binarnych związków rekursywnych

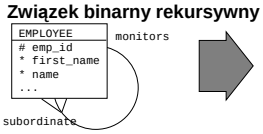
- Reguły transformacji podobne do stosowanych dla związków binarnych typu 1:1, 1:N i M:N.

Związek binarny rekursywny 1:1



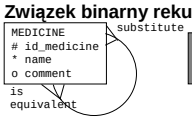
EMPLOYEES (
emp_id *PRIMARY KEY*,
...,
id_spouse *REFERENCES* employees(emp_id))

Związek binarny rekursywny 1:N



EMPLOYEES (
emp_id *PRIMARY KEY*,
...,
id_boss *REFERENCES* employees(emp_id))

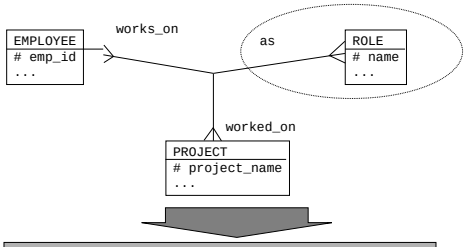
Związek binarny rekursywny M:N



EQUIVALENTS (
id_med *NOT NULL REFERENCES* medicines(id_med),
id_equ *NOT NULL REFERENCES* medicines(id_med),
PRIMARY KEY (id_med, id_equ))

Transformacja związków ternarnych

Typ: „wiele" dla wszystkich encji



Reguły transformacji:

- Utwórz schemat relacji związku dla związku ternarnego ternary
- Nowy schemat relacji zawiera klucze obce odpowiadające kluczom podstawowym wiązanych encji
- Klucze obce tworzą klucz podstawowy utworzonego schematu relacji związku

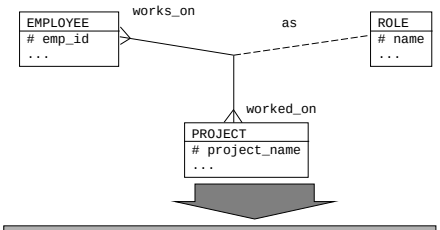
PARTICIPATION_IN_PROJ (
emp_id *NOT NULL REFERENCES* employees(emp_id),
project *NOT NULL REFERENCES* projects(project_name),
role *NOT NULL REFERENCES* roles(name),
PRIMARY KEY **(emp_id, project, role)**)

Transformacja związków ternarnych

Typ: „jeden” dla jednej z encji

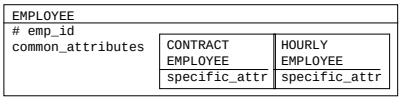
Reguły transformacji:

- Utwórz schemat relacji związku dla związku ternarnego ternary
- Nowy schemat relacji zawiera klucze obce odpowiadające kluczom podstawowym wiązanych encji
- Klucz podstawowy schematu relacji składa się z kluczy obcych do encji po stronie „wiele”



PARTICIPATION_IN_PROJ (
emp_id *NOT NULL REFERENCES* employees(emp_id),
project *NOT NULL REFERENCES* projects(project_name),
role *REFERENCES* roles(name),
PRIMARY KEY **(emp_id, project,)**)

Transformacja hierarchii generalizacji - do trzech schematów relacji



Reguły transformacji:

- Utwórz schemat relacji „supertype” zawierającej klucz podstawowy, atrybuty wspólne oraz jeden atrybut definiujący typ generalizacji.
- Dla każdej encji „subtype” utwórz schemat relacji „subtype” zawierającej atrybuty specyficzne i klucz podstawowy dziedziczony z relacji „supertype”

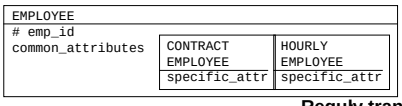
EMPLOYEES (
emp_id *PRIMARY KEY*,
common_attributes,
job_type *NOT NULL*)

CONTRACT_EMPLOYEES (
emp_id *PRIMARY KEY*,
specific_attr)

HOURLY_EMPLOYEES (
emp_id *PRIMARY KEY*,
specific_attr)

CONTRACT_EMPLOYEES
(emp_id, common_attributes, specific_attr) AS
select emp_id, common_attributes, specific_attr
from EMPLOYEES w, CONTRACT_EMPLOYEES cw
where w.emp_id = cw.emp_id;
HOURLY_EMPLOYEES
(emp_id, common_attributes, specific_attr) AS
select emp_id, common_attributes, specific_attr
from EMPLOYEES w, HOURLY_EMPLOYEES fw
where w.emp_id = fw.emp_id;

Transformacja hierarchii generalizacji - do dwóch schematów relacji



Reguły transformacji:

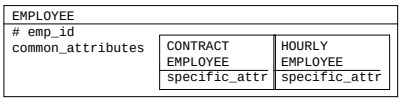
- Dla każdej encji „subtype” utwórz schemat relacji „subtype” zawierającej atrybuty wspólne, atrybuty specyficzne, oraz klucz podstawowy dziedziczony z encji „supertype „

CONTRACT_EMPLOYEES (
emp_id *PRIMARY KEY*,
common_attributes,
specific_attr)

HOURLY_EMPLOYEES (
emp_id *PRIMARY KEY*,
common_attributes,
specific_attr)

EMPLOYEES
(emp_id, common_attributes, job_type) AS
select emp_id, common_attributes, ‘CONTRACT’
from CONTRACT_EMPLOYEES
union
select emp_id, common_attributes, ‘HOURLY’
from HOURLY_EMPLOYEES;

Transformacja hierarchii generalizacji - do jednego schematu relacji



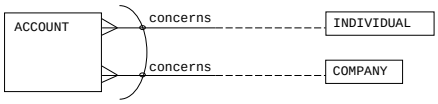
Reguły transformacji:

- Utwórz jeden schemat relacji zawierający: atrybuty wspólne, atrybuty specyficzne oraz atrybut definiujący typ generalizacji

EMPLOYEES (
emp_id *PRIMARY KEY*,
common_attribute,
specific_attr *NULL*,
specific_attr *NULL*,
job_type *NOT NULL*)

CONTRACT_EMPLOYEES
(emp_id, common_attributes, specific_attr) AS
select emp_id, common_attributes, specific_attr
from EMPLOYEES
where job_type = ‘CONTRACT’;
HOURLY_EMPLOYEES
(emp_id, common_attributes, specific_attr) AS
select emp_id, common_attributes, specific_attr
from EMPLOYEES
where job_type = ‘HOURLY’;

Transformacja łuków



Metoda 1. Wspólny klucz obcy

- Wtedy, gdy identyfikatory encji posiadają wspólną domenę

ACCOUNT (
account_number INTEGER(6) *PRIMARY KEY,*
... ,
concerns **CHAR(1)** *NOT NULL*
CHECK(concerns IN (’I’,’C’)),
account_balance **INTEGER(8)** *NOT NULL*)

Metoda 2. Indywidualny klucz obcy

- Wtedy, gdy identyfikatory encji posiadają różne domeny

ACCOUNT (
account_number INTEGER(6) *PRIMARY KEY,*
... ,
id_ind_account **CHAR(5)** *NULL*
REFERENCES individuals(kod_śr),
id_comp_account **INTEGER(8)** *NULL*
REFERENCES companies(id_furn))