

DBS – BI-WSI-SI-2

Transformace konceptuálního schématu (v ER nebo jiné notaci) na relační.

Obsah

1	Popis	2
2	Postup transformace	2
2.1	Transformace silné entity	2
2.2	Vztah 1:1, obě části jsou ve vztahu povinné	2
2.3	Vztah 1:1, kde jen jedna část je povinná	3
2.4	Vztah 1:1, obě nepovinné	3
2.5	Vztah 1:N, povinná účast stany N (determinant)	3
2.6	Vztah 1:N, nepovinná účast stany N (determinant)	3
2.7	Vztah M:N	4
2.8	Dekompozice vztahu M:N	4
2.9	Slabá entita	4
2.10	Identifikační závislost	5
3	ISA hierarchie	5

1 Popis

Algoritmus na transformaci (převod) konceptuálního schématu na relační bývá součástí modelovacích nástrojů (oracle, EA..). Některé konstrukce mohou mít několik variant převedení (několik různých).

2 Postup transformace

Jediná možnost provázání dat ze dvou relací je referenční integritou (FOREIGN KEY).

- název entity → název relace
- atribut entity → atribut relace
- povinnost atributu entity → NOT NULL
- atributy edentifikátoru entity → PRIMARY KEY
- alternativní klíče → UNIQUE
- u slabých entit → identifikátor vlastníka

2.1 Transformace silné entity

Daná entita se převede na tabulku s hodnotami **entita**(identifikator, povinny_atribut, nepovinny_atribut). Podtržení znamená, že je to primární klíč.



2.2 Vztah 1:1, obě části jsou ve vztahu povinné

Dojde k sjednocení daných entit do jedné tabulky **zamestnanec_vuz**(cislo_z, jmneo_z, adresa, spz, vyrobce, model). Kde navíc **spz** je NOT NULL a UNIQUE (protože byl identifikátorem).



2.3 Vztah 1:1, kde jen jedna část je povinná

Entity se nemohou spojit kvůli nepovinnosti jedné strany. Převědou se tedy na dvě tabulky, kde tabulka, jež je nepovinnou částí, bude mít ID odkaz na povinnou část.



- `zamestnanec(cislo_z, jmeno_z, adresa)`
- `vuz(spz, vyrobce, model, cislo_z)` a `cislo_z` je NOT NULL UNIQUE (kvůli zachování vazby 1:1)
- `vuz[cislo_z] ⊆ zamestnanec[cislo_z]`

2.4 Vztah 1:1, obě nepovinné

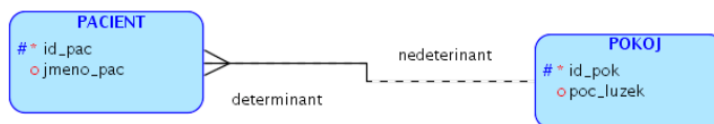


Lze vyřešit dvěma způsoby. **První řešení** odpovídá předchozímu případu, akorát `cislo_z` v tabulce `vuz` bude **nepovinný**.

Druhé řešení přidává relační tabulky:

- `zamestnanec(cislo_z, jmeno_z, adresa)`
- `vuz(spz, vyrobce, model)`
- `zamestnanec_vuz(cislo_z, spz)`, kde `SPZ` je NOT NULL a UNIQUE
- `zamestnanec_vuz[spz] ⊆ vuz[spz]`
- `zamestnanec_vuz[cislo_z] ⊆ zamestnanec[cislo_z]`

2.5 Vztah 1:N, povinná účast stany N (determinant)

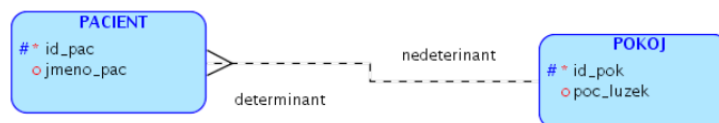


- `pacient(id_pac, jmeno_pac, id_pok)`
- `pokoj(id_pok, poc_luzek)`
- `pacient[id_pok] ⊆ pokoj[id_pok]`, kde `id_pok` je NOT NULL

2.6 Vztah 1:N, nepovinná účast stany N (determinant)

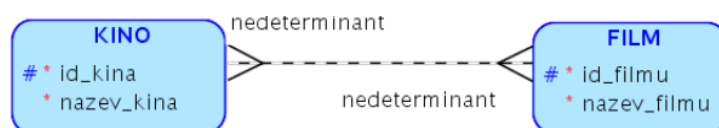
Opět dva způsoby jak se dá řešit, první je vzít řešení z předchozí ukázky a nastavit `id_pok` na **nepovinný**. A druhé řešení je opět vytvoření vazební tabulky

- `pacient(id_pac, jmeno_pac),`



- pokoj(id_pok, poc_luzek)
- umisteni(id_pac, id_pok), kde id_pok je NOT NULL
- umisteni[id_pac] $\subseteq$ pacient[id_pac]
- umisteni[id_pok] $\subseteq$ pokoj[id_pok]

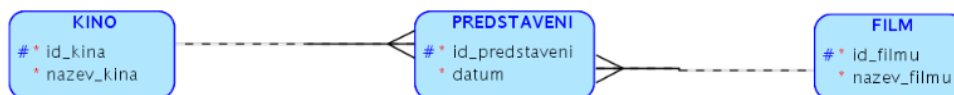
2.7 Vztah M:N



- kino(id_kina, nazev_kina),
- film(id_filmu, nazev_filmu)
- predstaveni(id_kina, id_filmu)
- predstaveni[id_kina] $\subseteq$ kino[id_kina]
- predstaveni[id_filmu] $\subseteq$ film[id_filmu]

Toto řešení způsobuje to, že v jendom kině se může hrát film pouze jednou. Tento problém se řeší pomocí dekompozice.

2.8 Dekompozice vztahu M:N

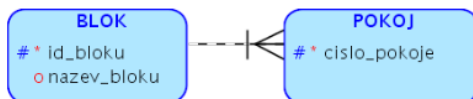


- kino(id_kina, nazev_kina),
- film(id_filmu, nazev_filmu)
- predstaveni(id_predstaveni, datum, id_kina, id_filmu), ID kina a filmu jsou NOT NULL
- predstaveni[id_kina] $\subseteq$ kino[id_kina]
- predstaveni[id_filmu] $\subseteq$ film[id_filmu]

2.9 Slabá entita

Vezmeme v příklad čísla bloků a v jednotlivých blocích jsou pokoje. Bloky jsou číslovány 1, 2,...,n a v každém bloku jsou číslovány pokoje od 1 do n. Pomocí samotného čísla pokoje se nedá identifikovat, kde se nachází, protože každý blok má pokoje stejně číslován. Ale pomocí kombinace číslo bloku + číslo pokoje jsmě již schopní pokoj najít.

- blok(id_bloku, nazev_bloku),
- pokoj(id_pokoje, id_bloku)



- $\text{pokoj}[\text{id_bloku}] \subseteq \text{blok}[\text{id_bloku}]$

2.10 Identifikační závislost



Každý uživatel má pouze jeden profil a ten profil patří pouze jednomu uživateli. Uživatel nemusí mít profil, ale pokud ho má, tak je jasně identifikován pomocí uživatele.

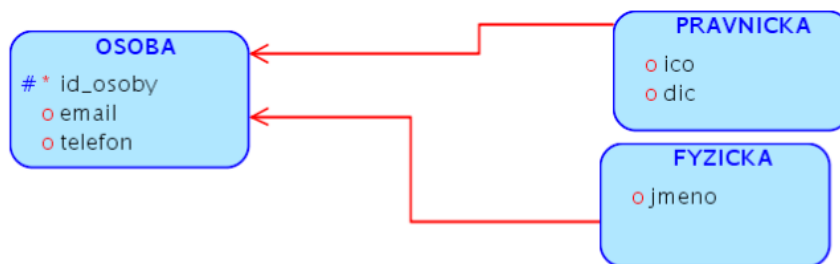
- $\text{osoba}(\text{id_osoby}, \text{jmeno_osoby})$
- $\text{profil}(\text{id_osoby}, \text{fotka})$
- $\text{profil}[\text{id_osoby}] \subseteq \text{osoba}[\text{id_osoby}]$

3 ISA hierarchie

Jedná se o způsob jak vyřešit více vztahů při převodu. Rozdělují se na tři typy:

- všechny vazby do jedné tabulky
- polymorfismus
- více oddělených tabulek

Příklad:



- vše dohromady
 - $\text{osoba}(\text{id_osoby}, \text{email}, \text{telefon}, \text{jmeno}, \text{ico}, \text{dic})$
- polymorfismus
 - $\text{osoba}(\text{id_osoby}, \text{email}, \text{telefon})$, kde ID osoby je UNIQUE
 - $\text{fyzicka}(\text{id_osoby}, \text{jmeno})$
 - $\text{pravnicka}(\text{id_osoby}, \text{ico}, \text{dic})$
 - $\text{fyzicka}[\text{id_osoby}] \subseteq \text{osoba}[\text{id_osoby}]$
 - $\text{pravnicka}[\text{id_osoby}] \subseteq \text{osoba}[\text{id_osoby}]$

- oddělené tabulky
 - fyzicka(id_osoby, email, telefon, jmneo)
 - pravnicka(id_osoby, email, telefon, ico, dic)