

7. Normální forma

verze z 11. listopadu 2024

1 Schéma databáze

Databázi vždy modelujeme jen vytyčenou část reality. Například filmy, které vlastníme. Jevy, které ve vytyčené části pozorujeme, dělíme na samostatné a průvodní. Samostatné jevy jsme schopni pozorovat přímo. Například film *La dolce vita* nebo režisér *Federico Fellini* jsou samostatné jevy. Samostatné jevy nazýváme objekty. Naproti tomu průvodní jevy nejsou pozorovatelné přímo, ale pouze za přítomnosti objektů. Například jev, že film *La dolce vita* byl vydán v roce 1960 je průvodním jevem. Bez objektu *La dolce vita* bychom jej pozorovat nemohli. Průvodní jevy dále dělíme na vlastnosti a vztahy. Vlastností nazýváme průvodní jev, který je pozorovatelný na jediném objektu. Například vlastností filmu *La dolce vita* je, že byl vydán v roce 1960. Vztahy jsou pak průvodní jevy, které pozorujeme mezi více objekty. Například jev, že *Federico Fellini* režíroval film *La dolce vita* je vztahem mezi objekty *Federico Fellini* a *La dolce vita*.

V databázi nejsme schopni objekty modelovat přímo, ale za použití jejich vlastností. Přitom nás při modelování zajímají pouze vlastnosti, které jsme schopni modelovat komponentami. Takové vlastnosti nazýváme popisné vlastnosti. Například popisnou vlastnost „film byl vydán v roce 1960“ modelujeme komponentou $\langle \text{movie_year}, 1960 \rangle$. Objekt pak můžeme modelovat n -ticí sdružující všechny popisné vlastnosti objektu. Například film *La dolce vita* můžeme modelovat n -ticí

$$\{\langle \text{movie_title}, 'La dolce vita' \rangle, \langle \text{movie_year}, 1960 \rangle, \langle \text{movie_length}, 174 \rangle\}.$$

Řekneme, že množina vlastností určuje objekt, pokud v realitě existuje právě jeden objekt, který má všechny vlastnosti z množiny. Například množina tvořená vlastnostmi „film byl vydán v roce 1960“ a „film má název *La dolce vita*“ určuje film *La dolce vita*.

Vztah také modelujeme n -ticí a to n -ticí tvořenou určujícími vlastnostmi objektů, které do vztahu vstupují. Například vztah *Federico Fellini* režíroval film *La dolce vita* modelujeme n -ticí:

$$\{\langle \text{movie_title}, 'La dolce vita' \rangle, \langle \text{movie_year}, 1960 \rangle, \langle \text{director_name}, 'Federico Fellini' \rangle\}.$$

Množinu objektů, které jsou modelovány n -ticemi nad stejným schématem, modelujeme relací tvořenou těmito n -ticemi. Například množinu tvořenou filmem *La dolce vita* a filmem *Rocco e i suoi fratelli* modelujeme relací:

movie_title	movie_year	movie_length
La dolce vita	1960	174
Rocco e i suoi fratelli	1960	170

Jak jsme zvyklí, relacím musíme být schopni přiřadit jejich charakteristickou vlastnost. Například právě uvedená relace by měla charakteristickou vlastnost „Film s názvem `movie_title` byl vydán v roce `movie_year` a je dlouhý `movie_length`.“ Pokud modelovaná část reality byla vytyčena vlastněnými filmy, pak z relace vyčteme, že nevlastníme film *Le notti bianche*. Podobně relací také modelujeme množinu vztahů, které jsou modelovány n -tici nad stejným schématem.

Databáze je pak tvořena relacemi uloženými v relačních proměnných. Názvy relačních proměnných, jejich schémata a charakteristické vlastnosti nazýváme souhrnně **schéma databáze**.

2 Problémy s návrhem databáze

Návrhem databáze rozumíme proces vytváření jeho schématu. Podíváme se nejprve na čtyři základní problémy, kterými může schéma databáze trpět.

2.1 Problém nezachytitelného stavu

Problém nezachytitelného stavu reality nastává, když se námi vymezená část reality dostane do stavu, který nejsme schopni databází modelovat. Například předpokládejme, že chceme modelovat část reality vymezenou vlastněnými filmy a režiséry, o které se zajímáme, relační proměnnou `movie` s charakteristickou vlastností „Film `title` režíroval `director`.“ Relace

title	director
Rocco e i suoi fratelli	Luchino Visconti
Le notti bianche	Luchino Visconti
La dolce vita	Federico Fellini

je možnou hodnotou proměnné. Nyní stav reality, ve kterém se zajímáme o režiséra, ale nevlastníme žádný jeho film, nejsme schopni modelovat. Například pokud se začneme zajímat o režiséra *Roberta Rosselliniho*, nemáme způsob, jak tuto informaci do databáze uložit. Podobně, pokud prodáme film *La dolce vita*, a smažeme příslušnou n -tici z proměnné, přestaneme se podle modelu zajímat o režiséra *Federico Fellini*, což je v rozporu s realitou.

Obecnému řešení problému nezachytitelného stavu se budeme věnovat později. Nyní pouze poznamenejme, že v našem příkladě lze problém vyřešit založením nové relační proměnné `director` s charakteristickou vlastností „`name` je režisér.“ Zde můžeme zájem o *Roberta Rosselliniho* modelovat:

name
Luchino Visconti
Federico Fellini
Roberto Rossellini

případně prodat film *La dolce vita*:

title	director
Rocco e i suoi fratelli	Luchino Visconti
Le notti bianche	Luchino Visconti

2.2 Problém nezachyceného vztahu

Problém nezachyceného vztahu nastává, když nejsme schopni modelovat žádaný vztah mezi objekty ve vytyčené části reality. Například předpokládejme, že chceme modelovat herce a filmy, o které máme zájem, a navíc nás zajímá, jakou roli herec ve filmu hrál. Uvažujme schéma databáze tvořené relační proměnnou `cast` danou vlastností „Ve filmu `title` hrál herec `actor`.“ a proměnnou `cast_role` danou vlastností „Herec `actor` hrál roli `role`.“ Možná hodnota proměnné `cast`:

title	actor
Rocco e i suoi fratelli	Alain Delon
La dolce vita	Marcello Mastroianni
Le notti bianche	Marcello Mastroianni

možná hodnota proměnné `cast_role`:

actor	role
Alain Delon	Rocco Parondi
Marcello Mastroianni	Mario
Marcello Mastroianni	Marcello Rubini

Podle našeho modelu nejsme schopni rozhodnout, zda *Marcello Mastroianni* hrál ve filmu *La dolce vita* roli *Mario* nebo *Marcello Rubini*. Problém vyřešíme spojením proměnných. Konkrétně můžeme místo výše popsaných proměnných uvažovat proměnnou `cast` s charakteristickou vlastností „Ve filmu `title` hrál herec `actor` roli `role`.“ Odpovídající možná hodnota:

title	actor	role
Rocco e i suoi fratelli	Alain Delon	Rocco Parondi
La dolce vita	Marcello Mastroianni	Marcello Rubini
Le notti bianche	Marcello Mastroianni	Mario

2.3 Nadbytečnost ve schématu databáze

K problému dochází, pokud ve schématu databáze existují dvě proměnné, které je možné spojit do jedné, aniž by vznikl problém nezachytitelného stavu. Například

vytyčme realitu tak, že chceme modelovat vlastněné filmy pomocí dvou proměnných, kde první proměnná `movie_year` má charakteristickou vlastnost „Film `title` byl vydán roku `year`“ a druhá `movie_length` má charakteristickou vlastnost „Film `title` je dlouhý `length` minut.“ Možná hodnota proměnné `movie_year`:

title	year
Rocco e i suoi fratelli	1960
Le notti bianche	1957
La dolce vita	1960

a proměnné `movie_length`:

title	length
Rocco e i suoi fratelli	170
Le notti bianche	102
La dolce vita	174

Nadbytečnost se zde projeví tak, že snadno můžeme databázi uvést do sporného stavu, kde **sporným stavem** myslíme že z databáze plyne pravdivost výroku i jeho negace. Představme si například, že prodáme film *La dolce vita* a smažeme odpovídající n -tici pouze z proměnné `movie_year`. Pak podle proměnné `movie_year` film nevlastníme, ale podle proměnné `movie_length` jej stále vlastníme.

Problém vyřešíme spojením proměnných. Proměnné spojíme do jedné proměnné `movie` s charakteristickou vlastností „Film `title` byl vydán roku `year` a je dlouhý `length` minut.“ Odpovídající možná hodnota:

title	year	length
Rocco e i suoi fratelli	1960	170
Le notti bianche	1957	102
La dolce vita	1960	174

3 Nadbytečnost v proměnné

Problém nadbytečnosti v proměnné nastává, když je možné rozdělit proměnnou do dvou tak, aby nevznikl problém nezachytitelného vztahu nebo problém nadbytečnosti ve schématu.

Například chceme modelovat vlastněné filmy, ve kterých hrají herci, o které se zajímáme proměnnou `movie` určenou vlastností „Ve filmu `title`, který režíroval `director`, hraje `actor`.“ Možná hodnota proměnné:

title	director	actor
Rocco e i suoi fratelli	Luchino Visconti	Alain Delon
Le notti bianche	Luchino Visconti	Marcello Mastroianni
Le notti bianche	Luchino Visconti	Maria Schell
La dolce vita	Federico Fellini	Marcello Mastroianni

určenou vlastností „Ve filmu `title`, který režíroval `director`, hraje `actor`.“ Problém nadbytečnosti umožňuje snadno změnit hodnotu proměnné tak, aby databáze byla sporná. Například po přidání herce *Jean Marais*, který hrál ve filmu *Le notti bianche* s chybně uvedeným režisérem *Federico Fellini*:

title	director	actor
Rocco e i suoi fratelli	Luchino Visconti	Alain Delon
Le notti bianche	Luchino Visconti	Marcello Mastroianni
Le notti bianche	Luchino Visconti	Maria Schell
La dolce vita	Federico Fellini	Marcello Mastroianni
Le notti bianche	Federico Fellini	Jean Marais

dostáváme, že film *Le notti bianche* režíroval i nереžíroval *Luchino Visconti*.

4 Boyce-Coddova normální forma

Problémem nadbytečnosti v proměnné se zabývá následující normální forma.

Vezměme relační proměnnou *relation*. Proměnná *relation* je v **Boyceho-Coddově normální formě (BCNF)**, pokud pro každou netriviální funkční závislost $X \rightarrow Y$ relační proměnné *relation* platí, že X je nadklíč proměnné *relation*. Boyce-Coddovu normální formu budeme zkráceně nazývat pouze **normální formou**.

Pokud $X \rightarrow Y$ je netriviální funkční závislost proměnné *relation*, ale X není nadklíč proměnné *relation*, pak říkáme, že $X \rightarrow Y$ **porušuje** normální formu.

Pokud existuje funkční závislost proměnné, která porušuje normální formu, pak proměnná trpí problémem nadbytečnosti v proměnné.

Věta 1. *Pokud relation není v normální formě, pak existuje funkční závislost $X \rightarrow Y$ proměnné relation v základním tvaru, která porušuje normální formu.*

Důkaz: Vezměme funkční závislost $X' \rightarrow Y'$, která porušuje normální formu. Jistě $Y' \not\subseteq X'$ a X' není nadklíč proměnné *relation*. Označme $X \rightarrow Y$ některou funkční závislost proměnné *relation* v normální formě, kde $X \subseteq X'$ a $Y' \subseteq Y$. Z $Y' \not\subseteq X'$ plyne, že $Y \not\subseteq X$ a tedy $X \rightarrow Y$ není triviální. Protože X' není nadklíč *relation* a $X \subseteq X'$, pak ani X není nadklíč *relation*. Tedy $X \rightarrow Y$ porušuje normální formu. \square

Věta 2. *Netriviální funkční závislost $X \rightarrow Y$ v základním tvaru proměnné relation nad S , kde $Y \neq S$, porušuje normální formu.*

Důkaz: Proměnná *relation* nespĺňuje $X \rightarrow S$ a tedy X není nadklíč. \square

5 Rozklad

Vezměme schéma S . Dvě schémata S_1 a S_2 taková, že $S_1 \cup S_2 = S$, jsou **rozkladem schématu** S . Například $\{\text{title}, \text{director}\}$ a $\{\text{title}, \text{actor}\}$ nebo $\{\text{director}\}$ a

$\{\text{title}, \text{actor}\}$ jsou rozklady schématu $\{\text{title}, \text{director}, \text{actor}\}$.

Pro relaci R nad S se relace $R_1 = \pi_{S_1}(R)$ a $R_2 = \pi_{S_2}(R)$ nazývají **rozkladem relace** R podle S_1 a S_2 . Například pro relaci R :

title	director	actor
Rocco e i suoi fratelli	Luchino Visconti	Alain Delon
Le notti bianche	Luchino Visconti	Marcello Mastroianni
Le notti bianche	Luchino Visconti	Maria Schell
La dolce vita	Federico Fellini	Marcello Mastroianni

určenou vlastností „Ve filmu **title**, který režíroval **director**, hraje **actor**.“ tvoří relace:

title	director
Rocco e i suoi fratelli	Luchino Visconti
Le notti bianche	Luchino Visconti
La dolce vita	Federico Fellini

a relace:

title	actor
Rocco e i suoi fratelli	Alain Delon
Le notti bianche	Marcello Mastroianni
Le notti bianche	Maria Schell
La dolce vita	Marcello Mastroianni

rozklad R podle $\{\text{title}, \text{director}\}$ a $\{\text{title}, \text{actor}\}$. Dále relace:

director
Luchino Visconti
Federico Fellini

a relace:

title	actor
Rocco e i suoi fratelli	Alain Delon
Le notti bianche	Marcello Mastroianni
Le notti bianche	Maria Schell
La dolce vita	Marcello Mastroianni

rozklad R podle $\{\text{director}\}$ a $\{\text{title}, \text{actor}\}$.

Věta 3. Pokud R_1 a R_2 tvoří rozklad relace R nad S podle rozkladu S_1 a S_2 schématu S , pak $R \subseteq R_1 \bowtie R_2$.

Důkaz: Vezměme libovolnou n -tici $t \in R$, pak máme $t(S_1) \in \pi_{S_1}(R)$ a $t(S_2) \in \pi_{S_2}(R)$ a tedy $t \in \pi_{S_1}(R) \bowtie \pi_{S_2}(R) = R_1 \bowtie R_2$. \square

Pokud $R_1 \bowtie R_2 = R$, pak se rozklad relace nazývá **bezeztrátový**. Například rozklad relace

title	director	actor
Rocco e i suoi fratelli	Luchino Visconti	Alain Delon
Le notti bianche	Luchino Visconti	Marcello Mastroianni
Le notti bianche	Luchino Visconti	Maria Schell
La dolce vita	Federico Fellini	Marcello Mastroianni

podle $\{\text{title}, \text{director}\}$ a $\{\text{title}, \text{actor}\}$ je bezetrátový, ale rozklad podle $\{\text{director}\}$ a $\{\text{title}, \text{actor}\}$ již bezetrátový není.

Funkční závislost $X \rightarrow Y$ nad S **určuje** rozklad $X \cup Y$ a $X \cup (S - Y)$ schématu S . Například funkční závislost $\{\text{title}\} \rightarrow \{\text{director}\}$ nad $\{\text{title}, \text{director}, \text{actor}\}$ určuje rozklad $\{\text{title}, \text{director}\}$ a $\{\text{title}, \text{actor}\}$ schématu $\{\text{title}, \text{director}, \text{actor}\}$.

Věta 4. *Pokud je v relaci R množina Y funkčně závislá na X , pak rozklad R určený $X \rightarrow Y$ je bezetrátový.*

Důkaz: Stačí ukázat, že $\pi_{X \cup Y}(R) \bowtie \pi_{X \cup (S - Y)}(R) \subseteq R$. Obrácená inkluze plyne z věty 3. Vezměme libovolnou n -tici $t \in \pi_{X \cup Y}(R) \bowtie \pi_{X \cup (S - Y)}(R)$. Z definice spojení dostáváme, že $t(X \cup (S - Y)) \in \pi_{X \cup (S - Y)}(R)$. Z definice projekce plyne, že musí existovat n -tice $t' \in R$, která se s t shoduje v X i v $S - Y$. Protože v relaci R je Y funkčně závislé na X , tak se t a t' shodují i v Y . Dohromady dostáváme, že t a t' se shodují v $(S - Y) \cup Y = S$ a proto $t = t'$; tedy $t \in R$.

□

Například rozklad relace

title	director	actor
Rocco e i suoi fratelli	Luchino Visconti	Alain Delon
Le notti bianche	Luchino Visconti	Marcello Mastroianni
Le notti bianche	Luchino Visconti	Maria Schell
La dolce vita	Federico Fellini	Marcello Mastroianni

určený funkční závislostí $\text{title} \rightarrow \text{director}$ je bezetrátový, ale rozklad podle $\text{director} \rightarrow \text{actor}$ bezetrátový není. Skutečně, relaci bychom rozložili na relace

director	actor
Luchino Visconti	Alain Delon
Luchino Visconti	Marcello Mastroianni
Luchino Visconti	Maria Schell
Federico Fellini	Marcello Mastroianni

a

title	director
Rocco e i suoi fratelli	Luchino Visconti
Le notti bianche	Luchino Visconti
Le notti bianche	Luchino Visconti
La dolce vita	Federico Fellini

jejichž spojení není rovné původní relaci.

Uvažujme relační proměnné *relation* nad S , *relation1* nad S_1 a *relation2* nad S_2 , pak *relation1* a *relation2* jsou **rozkladem proměnné *relation***, pokud vždy R_1 a R_2 je rozklad R podle S_1 a S_2 , kde R, R_1, R_2 jsou postupně hodnoty proměnných *relation, relation1, relation2*. Navíc řekneme, že tento **rozklad proměnné je bezeztrátový**, pokud se vždy jedná o bezeztrátový rozklad.

Například proměnná *movie* určená vlastností „Film *title* režíroval *director*.“ a proměnná *cast* určená vlastností „Ve filmu *title* hraje *actor*.“ tvoří bezeztrátový rozklad proměnné *movie_cast* určené vlastností „Ve filmu *title*, který režíroval *director*, hraje *actor*.“

6 Návrh databáze

Předpokládejme, že chceme modelovat jistou část reality. Návrh databáze budeme demonstrovat na příkladu databáze filmů, které vlastníme.

6.1 Shromáždění atributů

Vy vytyčené části reality najdeme typy objektů a určíme jejich popisné vlastnosti:

1. film
 - (a) název filmu
 - (b) rok vydání filmu
 - (c) délka filmu
2. režisér
 - (a) jméno režiséra
 - (b) počet Oskarů, které režisér získal
3. scénář
 - (a) název filmu
 - (b) rok vydání filmu
 - (c) počet scén ve scénáři
4. herec
 - (a) jméno herce
 - (b) rok narození herce
5. asistent
 - (a) jméno asistenta

Popisným vlastnostem přiřadíme atributy:

1. `movie_title` ...název filmu
2. `movie_year` ...rok vydání filmu
3. `movie_length` ...délka filmu
4. `director_name` ...jméno režiséra
5. `oscar_count` ...počet Oskarů, které režisér získal
6. `scene_count` ...počet scén ve scénáři k filmu
7. `actor_name` ...jméno herce
8. `actor_born` ...rok narození herce
9. `asistent_name` ...jméno asistenta režiséra

Atributům přiřadíme skalární typ určující jejich doménu:

1. `movie_title` ...text
2. `movie_year` ...integer
3. `movie_length` ...integer
4. `director_name` ...text
5. `oscar_count` ...integer
6. `scene_count` ...integer
7. `actor_name` ...text
8. `actor_born` ...integer
9. `asistent_name` ...text

6.2 Vytvoření jedné proměnné

Vytvoříme jednu základní relaci nad všemi shromážděnými atributy a určíme její charakteristickou vlastnost. Dále stanovíme omezení na možné hodnoty základní relace.

Vytvoříme základní relaci `movie_database` nad

```
{movie_title, movie_year, movie_length  
director_name, oscar_count, asistent_name, scene_count,  
actor_name, actor_born}.
```

Charakteristická vlastnost proměnné je:

„Film `movie_title` vydaný roku `movie_year` je dlouhý `movie_length`, režíruje jej `director_name`, který získal `oscar_count` Oskarů, s asistentem `asistent_name` podle scénáře s `scene_count` scénami a ve filmu hraje herec `actor_name` narozený roku `actor_born`.“

Stanovme si omezení na možné hodnoty proměnné `movie_database`:

1. Člověk (herec, režisér i asistent) je určen svým jménem.
2. Film je určen názvem a rokem vydání.
3. Scénář je určen názvem filmu a rokem jeho vydání.
4. Film režíroval právě jeden režisér.
5. Režisér má nejvýše jednoho asistenta.
6. Asistent pomáhá právě jednomu režisérovi.

Přirozeně tento výchozí návrh většinou bude trpět problémem nezachytitelného stavu a nadbytečnosti v proměnné. Naproti tomu je oproštěn od problému nezachyceného vztahu, protože jsme dali do vztahu všechny atributy, a problému nadbytečnosti ve schématu databáze a to z toho důvodu, že máme pouze jednu proměnnou.

6.3 Normalizace

Nejprve se pokusíme vyřešit problém nadbytečnosti v proměnné pomocí **normalizace**. To znamená, že budeme převádět schéma databáze do normální formy. K normalizaci databáze budeme používat následující postup. Normalizace relační proměnné *relation* nad S :

1. Najdeme všechny netriviální funkční závislosti v základním tvaru proměnné *relation*.
2. Závislost tvaru $X \rightarrow S$ určuje kandidátní klíč X proměnné *relation*. Pokud žádná taková funkční závislost neexistuje, pak je S jediným kandidátním klíčem proměnné *relation*.
3. Ostatní funkční závislosti porušují normální formu. Pokud taková funkční závislost existuje, jednu vybereme a rozložíme podle ní *relation* na *relation1* a *relation2*.
4. Rekurzivně provedeme normalizaci proměnných *relation1* a *relation2*.

Podle předchozího postupu nyní normalizujeme naši jedinou základní relaci.

Provedeme normalizaci proměnné `movie_database`. Najdeme všechny netriviální funkční závislosti v základním tvaru proměnné `movie_database`:

1. `director_name` → `director_name, oscar_count, asistent_name`
2. `asistent_name` → `asistent_name, director_name, oscar_count`
3. `actor_name` → `actor_name, actor_born`
4. `movie_title, movie_year`
 → `movie_title, movie_year,`
 `movie_length, director_name,`
 `oscar_count, asistent_name, scene_count`
5. `movie_title, movie_year, actor_name`
 → `movie_title, movie_year, actor_name,`
 `movie_length, director_name,`
 `oscar_count, asistent_name,`
 `actor_born, scene_count`

Jenom pátá funkční závislost určuje kandidátní klíč `{movie_title, movie_year, actor_name}` proměnné `movie_database`. Ostatní funkční závislosti porušují normální formu. Rozložíme `movie_database` podle jedné z nich. Například podle

```
movie_title, movie_year → movie_title, movie_year,
                           movie_length, director_name,
                           oscar_count, asistent_name, scene_count
```

na `movie_database1` nad `{movie_title, movie_year, movie_length, director_name, oscar_count, asistent_name, scene_count}` a `movie_database2` nad `{movie_title, movie_year, actor_name, actor_born}`.

Charakteristická vlastnost proměnné `movie_database1` je

„Film `movie_title` vydaný roku `movie_year` je dlouhý `movie_length`, režíruje jej `director_name`, který získal `oscar_count` Oskarů, s asistentem `asistent_name` podle scénáře s `scene_count` scénami.“

a proměnné `movie_database2` je

„Ve film `movie_title` vydaném roku `movie_year` hraje herec `actor_name` narozený roku `actor_born`.“

Netriviální funkční závislosti v základním tvaru proměnné `movie_database1`:

1. `director_name` → `director_name, oscar_count, asistent_name`
2. `asistent_name` → `asistent_name, director_name, oscar_count`
3. `movie_title, movie_year`
 → `movie_title, movie_year,`
 `movie_length, director_name,`
 `oscar_count, asistent_name, scene_count`

Jediný kandidátní klíč $\{\text{movie_title}, \text{movie_year}\}$ proměnné movie_database1 je určený třetí funkční závislostí. Ostatní funkční závislosti porušují normální formu. Rozložíme movie_database1 třeba podle

$$\text{director_name} \rightarrow \text{director_name}, \text{oscar_count}, \text{asistent_name}$$

na director nad $\{\text{director_name}, \text{oscar_count}, \text{asistent_name}\}$ a movie nad $\{\text{movie_title}, \text{movie_year}, \text{movie_length}, \text{director_name}, \text{scene_count}\}$.

Charakteristická vlastnost proměnné director je

„Režisér director_name získal oscar_count Oskarů a pracuje s asistentem asistent_name .“

a proměnné movie je

„Film movie_title vydaný roku movie_year je dlouhý movie_length , režíruje jej director_name podle scénáře s scene_count scénami.“

Obě netriviální funkční závislosti v základním tvaru proměnné director

1. $\text{director_name} \rightarrow \text{director_name}, \text{oscar_count}, \text{asistent_name}$
2. $\text{asistent_name} \rightarrow \text{asistent_name}, \text{director_name}, \text{oscar_count}$

určují kandidátní klíče $\{\text{director_name}\}$ a $\{\text{asistent_name}\}$. Proměnná je tedy v normální formě.

Jediná netriviální funkční závislost v základním tvaru proměnné movie

1. $\text{movie_title}, \text{movie_year} \rightarrow \text{movie_title}, \text{movie_year}, \text{movie_length}, \text{director_name}, \text{scene_count}$

určuje kandidátní klíč $\{\text{movie_title}, \text{movie_year}\}$ a tedy proměnná je v normální formě.

Netriviální funkční závislosti v základním tvaru proměnné movie_database2 :

1. $\text{actor_name} \rightarrow \text{actor_name}, \text{actor_born}$
2. $\text{movie_title}, \text{movie_year}, \text{actor_name} \rightarrow \text{movie_title}, \text{movie_year}, \text{actor_name}, \text{actor_born}$

Druhá funkční závislost určuje kandidátní klíč $\{\text{movie_title}, \text{movie_year}, \text{actor_name}\}$ proměnné movie_database2 . První porušuje normální formu a rozkládá movie_database2 na actor nad $\{\text{actor_name}, \text{actor_born}\}$ a movie_cast nad $\{\text{movie_title}, \text{movie_year}, \text{actor_name}\}$.

Charakteristická vlastnost proměnné movie_cast je

„Ve film movie_title vydaném roku movie_year hraje herec actor_name .“

a proměnné `actor` je

„Herec `actor_name` se narodil roku `actor_born`.“

Obě proměnné jsou v normální formě.

Normalizací jsme získali následující základní relace v normální formě.

1. `director` nad `{director_name, oscar_count, asistent_name}`

„Režisér `director_name` získal `oscar_count` Oskarů a pracuje s asistentem `asistent_name`.“

2. `movie` nad `{movie_title, movie_year, movie_length, director_name, scene_count}`

„Film `movie_title` vydaný roku `movie_year` je dlouhý `movie_length`, režíruje jej `director_name` podle scénáře s `scene_count` scénami.“

3. `actor` nad `{actor_name, actor_born}`

„Herec `actor_name` se narodil roku `actor_born`.“

4. `movie_cast` nad `{movie_title, movie_year, actor_name}`

„Ve filmu `movie_title` vydaném roku `movie_year` hraje herec `actor_name`.“

Ve výjimečných případech se může stát, že normalizované schéma databáze stále trpí problémem nadbytečnosti v proměnné. V takovém případě musíme dále provést bezetrátový rozklad postižené proměnné. Jako příklad se podívejme na proměnnou `director_actor` danou vztahem: „`director_name` je režisér a `actor_name` je herec.“ Možná hodnota proměnné:

<code>director_name</code>	<code>actor_name</code>
Luchino Visconti	Alain Delon
Luchino Visconti	Marcello Mastroianni
Federico Fellini	Alain Delon
Federico Fellini	Marcello Mastroianni

Proměnnou můžeme rozdělit na proměnné `director`:

<code>director_name</code>
Luchino Visconti
Federico Fellini

a `actor`:

<code>actor_name</code>
Alain Delon
Marcello Mastroianni

6.4 Další rozklady

Zkontrolujeme, zda schéma databáze netrpí problémem nezachytitelného stavu. Pokud by k problému došlo, musíme postiženou proměnnou rozložit.

Například nyní nejsme schopni modelovat stav, kdy režisér nemá asistenta. Proto proměnnou `director` dále rozložíme podle funkční závislosti `director_name` → `asistent_name` na `asistent` nad `{asistent_name, director_name}` a `director` nad `{director_name, oscar_count}`. Proměnná `asistent` má charakteristickou vlastnost:

„Asistent `asistent_name` pracuje s režisérem `director_name`.“

a nová verze proměnné `director` má charakteristickou vlastnost:

„Režisér `director_name` získal `oscar_count` Oskarů.“

Netriviální funkční závislosti v základním tvaru proměnné `asistent`

1. `director_name` → `director_name, asistent_name`
2. `asistent_name` → `asistent_name, director_name`

určují kandidátní klíče `{director_name}` a `{asistent_name}`.

Netriviální funkční závislosti v základním tvaru nové verze proměnné `director`

1. `director_name` → `director_name, oscar_count`

určuje kandidátní klíč `{director_name}`.

6.5 Integritní omezení

K definicím základních relací přidáme primární, alternativní a cizí klíče. Výsledný model v SQL:

```
CREATE TABLE director (  
    director_name text PRIMARY KEY,  
    oscar_count integer  
);  
-- Režisér "director_name" získal "oscar_count" Oskarů.
```

```
CREATE TABLE movie (  
    movie_title text,  
    movie_year integer,  
    movie_length integer,  
    director_name text REFERENCES director,  
    scene_count integer,
```

```

        PRIMARY KEY ( movie_title, movie_year )
    );

-- Film "movie_title" vydaný roku "movie_year"
-- je dlouhý "movie_length"
-- a režíruje jej "director_name"
-- podle scénáře s "scene_count" scénami.

CREATE TABLE actor (
    actor_name text PRIMARY KEY,
    actor_born integer
);
-- Herec "actor_name" se narodil roku "actor_born".

CREATE TABLE movie_cast (
    movie_title text,
    movie_year integer,
    actor_name text REFERENCES actor,
    FOREIGN KEY ( movie_title, movie_year ) REFERENCES movie,
    PRIMARY KEY ( movie_title, movie_year, actor_name )
);
-- Ve film "movie_title" vydaném roku "movie_year"
-- hraje herec "actor_name".

CREATE TABLE asistent (
    asistent_name text PRIMARY KEY,
    director_name text UNIQUE REFERENCES director
);
-- Asistent "asistent_name" pracuje s režisérem "director_name".

```

Otázky a úkoly na cvičení

1. Rozhodněte, zda jsou následující proměnné v normální formě. Pokud v normální formě nejsou, uveďte všechny funkční závislosti proměnné v základním tvaru, které normální formu porušují. Objekty (filmy, města, kina, skladatele, ...) určíme jménem a dny datem.
 - (a) `cinema_movie` je dáno „V městě `town` je kino `cinema`, které někdy promítalo film `movie`“
 - (b) `director_movie` je dáno „Režisér `director` natočil film `movie1` a hrál ve filmu `movie2`.“
 - (c) `composer_movie` je dáno „Skladatel `composer` narozený v zemi `country` složil v roce `year` hudbu k filmu `movie`.“

- (d) `critic_movie` je dáno „Kritik `critic` ohodnotil film `movie` `star_count` hvězdičkami.“
- (e) `cinema_program` je dáno „Kino `cinema` promítá film `movie` od režiséra `director` v den `day`.“

2. Určete všechny bezztrátové rozklady relace:

<code>town</code>	<code>cinema</code>	<code>movie</code>	<code>year</code>
Olomouc	Metropol	Anora	2024
Olomouc	Metropol	Vlny	2024
Olomouc	Premiere Šantovka	Anora	2024
Krnov	Mír	Pulp Fiction	1994

- 3. Určete všechny rozklady proměnné `cinema` určené netriviálními funkčními závislostmi v základním tvaru. Proměnná `cinema` má charakteristickou vlastnost „V městě `town` nalézající se kino `cinema` promítá film `movie` z roku `year`.“
- 4. Normalizujte proměnné z prvního úkolu.
- 5. Vytvořte podle postupu z šesté kapitoly relační model pro jisté sdružení kin. Předpokládejte, že kino je určeno jménem a městem, ve kterém se nalézá. Z vlastností kina nás dále zajímá jeho kapacita. Model musí zachycovat programy jednotlivých kin. Program kina udává datum a čas promítání jednotlivých filmů. Film je určen jménem a rokem vydáním. U filmu nás dále zajímá, jak je dlouhý a jaký režisér ho natočil.