Databázové systémy

(vybrané kapitoly)

Jan Outrata

Olomouc, leden, 2007

1 Dotazovací systémy

Operace přidání řádku (ADD), zrušení řádku (DEL) a změna řádku (CH) tabulky pro definici dat tvoří základ *update systému*. Výrazy relační algebry reprezentující výpočet nad relacemi vyjadřují dotaz relační databáze. Relační algebra je součástí *dotazovacího (query) systému* a je tradičně základem dotazovacích jazyků, např. SQL. Formalizací množinových operací relační algebry pomocí aparátu predikátové logiky 1. řádu dostaneme *relační kalkul*. Hodnoty proměnných *n-ticového (tuple) relačního kalkulu* jsou n-tice hodnot z domén atributů ze schématu relace, proměnné *doménového relačního kalkulu* nabývají přímo jednotlivých hodnot z domén atributů. Oba kalkuly jsou ekvivalentní relační algebře (viz dále). Dotazovací jazyky tak mohou být založené jak na relační algebře, tak i na relačním kalkulu, nejčastěji kombinující obojí. Zatímco relační algebra je procedurální jazyk, tj. výraz relační algebry přesně popisuje pořadí jednotlivých operací nad relacemi, tzn. jak se dobrat výsledku vyhodnocení výrazu, relační kalkuly jsou neprodurální jazyky, tzn. výraz relačního kalkulu popisuje co má být výsledkem vyhodnocení a ne jak jej vypočítat. Z tohoto podledu jsou relační kalkuly řazeny k jazykům vyšší úrovně, protože způsob výpočtu může místo uživatele jazyka efektivně (optimalizovaně) řešit databázový systém.

Řekneme, že výrazy (relační algebry nebo relačního kalkulu) E_1 a E_2 jsou *ekvivalentní*, ozn. $E_1 \equiv E_2$, jestliže $E_1(d) = E_2(d)$ pro každý stav databáze d. Takto definovaná ekvivalnce výrazů je závislá na schématu relační databáze, např. výrazy $\pi_{AB}(r \bowtie s)$ a $\pi_{AB}(r) \bowtie \pi_{AB}(s)$ jsou ekvivalentní, pokud r je nad schématem např. ABC a s je nad schématem např. ABD, ale nejsou ekvivalentní např. pro $r(ABCD)$ a $s(ABDE)$. Naproti tomu $\sigma_{A=a}(r \bowtie s) \equiv \sigma_{A=a}(r) \bowtie s$ pro libovolné schéma relace r obsahující atribut A, tj. lib. schéma kompatibilní s výrazy. Ekvivalence výrazů zesílená o požadavek rovnosti vyhodnocení také pro lib. schéma relační databáze kompatibilní s výrazy už ale není tranzitivní, např. pro $E_1 = \pi_{AB}(r \cap s)$, $E_2 = \pi_{AB}(r) \cap s$ a $E_3 = \pi_{AB}(r) \cap \pi_{AB}(s)$ je $E_1 \equiv E_2$ a $E_2 \equiv E_3$, ale není $E_1 \equiv E_3$, protože neplatí např. pro schéma ABC relací r a s. Dotazovací systém QS_1 je *stejně silný (expressive)* jako dotazovací systém QS_2, jestliže pro každý výraz $E_2 \in QS_2$ a každé databázové schéma kompatibilní s E_2 existuje výraz $E_1 \in QS_1$ takový, že $E_1 \equiv E_2$. Dotazovací systém je *úplný*, jestliže je stejně silný jako relační algebra. Úplnými dotazovacími systémy jsou např. n-ticový a doménový relační kalkul nebo úplná podalgebra relační algebry (viz přednášky).

Relační kalkuly zavedl a používal již E. F. Codd v roce 1971 spolu s relační algebrou pro ověření úplnosti (prvního) dotazovacího jazyka APLHA. Nesporný význam relačních kalkulů tví ve spojení predikátové logiky 1. řádu a databází.

1.1 N-ticový relační kalkul

N-ticový relační kalkul $\mathcal{TC} = (\mathbf{U}, \mathbf{D}, dom, \mathbf{R}, d, \Theta)$, kde $\mathbf{U}$ je množina atributů (universum), $\mathbf{D}$ je množina domén (množin hodnot atributů), *dom* je zobrazení z $\mathbf{U}$ do $\mathbf{D}$ přiřazující každému atributu jeho doménu (přípustnou množninu jeho hodnot), $\mathbf{R}$ je schéma relační databáze, d je relační databáze nad schématem $\mathbf{R}$ a Θ je množina relací srovnání nad $D_i \in \mathbf{D}$ obsahující alespoň rovnost a nerovnost pro každou doménu D_i, je podobný množinovým operacím relační algebry. Zatímco základními prvky relační algebry jsou relace, základními prvky n-ticového relačního kalkulu jsou n-tice hodnot z domén atributů ze schématu relace. *Výrazy* mají tvar $\{x(R) \mid f(x)\}$, kde f je Boolovský predikát nad n-ticovou proměnnou x nad relačním schématem $R \subseteq \mathbf{U}$. Hodnotou $E(d)$ výrazu je relace $r(R)$ obsahující n-tice $t \in dom(R)$ splňující predikát $f(x)$.

Příklad 1 Výrazy n-ticového relačního kalkulu:
(1) $\{x(\text{jméno příjmení}) \mid x \in \text{Osoba} \ \wedge \ x(\text{pohlaví}) = \text{“žena”}\}$
(2) $\{x(\text{název}) \mid x \in \text{Předmět} \ \wedge \ \exists y \in \text{Předměty\_Oboru}(x(\text{id}) = y(\text{předmět})) \ \wedge \ \exists z \in \text{Obor}(y(\text{obor}) = z(\text{id})) \ \wedge \ z(\text{zkratka}) = \text{“INF”}\}$.

V duchu predikátové logiky (relačními symboly jsou relace r_i) zavádíme pojmy jako *formule* (relativně k atributům z $\mathbf{U}$, doménám z $dom(\mathbf{U})$, relacím srovnání z Θ a relacím r_i schématu R_i), *atomy* $true$, $false$, $r(x)$ alias $x \in r$, $x(R)\theta y(S)$ a $c\theta x(R)$, *volný* a *vázaný výskyt* n-ticové proměnné ve formuli (analogie globální a lokální proměnné funkce programovacího jazyka, výskyt svazují kvantifikátory $\exists$ a $\forall$), *legální* formule, formule $f(t/x)$ se *substitucí* n-tice t za proměnnou x s volným výskytem v f, *vyhodnocení (interpretace)* $I(f(t/x))$ v $\{true, false\}$ formule f bez proměnných s volným výskytem, apod. Ve výrazu $\{x(R) \mid f(x)\}$ jsou potom f legální formule a x jediná volná proměnná v f a hodnotou $E(d)$ výrazu je relace $r(R)$ obsahující n-tice $t \in dom(R)$, pro které platí $I(f(t/x)) = true$.

Věta 1 *Nechť $\mathcal{R}$ je relační algebra s komplementem a $\mathcal{TC}$ je n-ticový kalkul. Pro každý relační výraz E nad $\mathcal{R}$ existuje ekvivalentní výraz F nad $\mathcal{TC}$, tj. pro každý stav databáze d platí $E(d) = F(d)$.*

Důkaz. Konstrukcí F indukcí pro jednotlivé operace E' nad $\mathcal{R}'$ (úplná podalgebra $\mathcal{R}$) podle jejich počtu. $\square$

Příklad 2 Výrazy z příkladu 1 jsou ekvivalentní relačním výrazům
(1) $\pi_{\text{jméno, příjmení}}(\sigma_{\text{pohlaví=“žena”}}(\text{Osoba}))$
(2) $\pi_{\text{název}}(\text{Předmět}[\text{id} = \text{předmět}]\text{Předměty\_Oboru}[\text{obor} = \text{id}]\sigma_{\text{zkratka=“INF”}}(\text{Obor}))$.

Výraz E nad $\mathcal{TC}$ se může vyhodnotit na nekonečnou relaci (v případě nekonečných domén), mluvíme o *neomezené interpretaci.* Při *omezené interpretaci* obsahují n-tice výsledné relace jen hodnoty z relací nebo konstantní hodnoty ve formuli f výrazu, tj. $t \in edom(R, f)$. Výsledná relace je potom konečná.

Věta 2 *Nechť $\mathcal{R}$ je relační algebra bez komplementu a $\mathcal{TC}$ je n-ticový kalkul. Pro každý relační výraz E nad $\mathcal{R}$ existuje výraz F nad $\mathcal{TC}$ takový, že E je ekvivalentní F pod omezeným vyhodnocením.*

Bezpečný výraz je výraz, jehož hodnotou při neomezeném vyhodnocováním je konečná relace a při vyhodnocování podvýrazů $\exists y(S)$ a $\forall y(S)$ je potřeba zpracovat jen konečný počet n-tic. Výrazy z příkladu 1 jsou bezpečné, nebezpečné by byly při zápisu $\ldots \vee \exists \ldots$ místo $\ldots \wedge \exists \ldots$.

Lemma 3 *Hodnoty bezpečného výrazu nad $\mathcal{TC}$ při neomezeném i omezeném vyhodnocování se rovnají.*

Věta 4 *Pro každý výraz E nad $\mathcal{TC}$ existuje bezpečný výraz F nad $\mathcal{TC}$ ekvivalentní E při omezeném vyhodnocování.*

1.2 Doménový relační kalkul

Doménový relační kalkul $\mathcal{DC} = (\mathbf{U}, \mathbf{D}, dom, \mathcal{R}, d, \Theta)$, kde prvky šestice mají stejný význam jako u n-ticového relačního kalkulu, je podobný n-ticovému relačnímu kalkulu. Základními prvky doménového relačního kalkulu jsou místo n-tic hodnot přímo jednotlivé hodnoty z domén atributů ze schématu relace. *Výrazy* mají tvar $\{x_1(A_1) \ldots x_n(A_n) \mid f(x_1, \ldots, x_n)\}$, kde f je Boolovský predikát nad n proměnnými x_i nabývajících hodnot z domén atributů $A_i \in \mathbf{U}$. Hodnotou $E(d)$ výrazu je relace $r(\{A_i\})$ obsahující n-tice $\langle c_1, \ldots, c_n \rangle, c_i \in dom(A_i)$ splňující predikát $f(x_1, \ldots, x_n)$.

Příklad 3 Výrazy doménového relačního kalkulu:
(1) {jméno příjmení | Osoba(jméno příjmení pohlaví) $\wedge$ pohlaví = "žena"} $\equiv$
{jméno příjmení | Osoba(jméno příjmení "žena")}
(2) {název | $\exists y$ Předmět(y název) $\wedge$ $\exists z$ Předměty_Oboru(y z) $\wedge$ Obor(z "INF")}.

Analogicky n-ticovému relačnímu kalkulu zavádíme v duchu predikátové logiky (relačními symboly jsou n-tice t_j) pojmy jako *formule* (relativně k atributům z $\mathbf{U}$, doménám z $dom(\mathbf{U})$, relacím srovnání z Θ a n-ticím t_j relací r_k schématu R_k), *atomy* $true$, $false$, $r(x_1, \ldots, x_n)$ alias $\langle x_1, \ldots, x_n \rangle \in r$, $x(A)\theta y(B)$ a $c\theta x(A)$, *volný* a *vázaný výskyt* proměnné ve formuli, *legální* formule, formule $f(c/x)$ se *substitucí* konstanty c za proměnnou x s volným výskytem v f, *vyhodnocení (interpretace)* $I(f(c/x))$ v $\{true, false\}$ formule f bez proměnných s volným výskytem, apod. Ve výrazu $\{x_1(A_1) \ldots x_n(A_n) \mid f(x_1, \ldots, x_n)\}$ jsou potom f legální formule a x_i jediné volné proměnné v f a hodnotou $E(d)$ výrazu je relace $r(\{A_i\})$ obsahující n-tice $\langle c_1, \ldots, c_n \rangle, c_i \in dom(A_i)$, pro které platí $I(f(c_i/x_i)) = true$.

Analogicky můžeme definovat také *neomezenou* a *omezené interpretaci* ($c_i \in edom(A_i, f)$) a *bezpečný výraz* a formulovat odpovídající Lemma a Větu z části n-ticového relačního kalkulu pro doménový relační kalkul.

Výraz n-ticového relačného kalkulu lze jednoduše převést na výraz doménového relačního kalkulu tak, že n-ticové proměnné $x(R)$ nahradíme proměnnými $x_i(A_i), A_i \in R$ (a atomy $r(x)$ atomy $r(x_1, \ldots, x_n)$).

Příklad 4 Výrazy z příkladu 3 jsou převedené z výrazů z příkladu 1.

Věta 5 *Výraz F nad $\mathcal{DC}$ získaný převodem z výrazu E nad $\mathcal{TC}$ je výrazu E ekvivalentní při omezeném i neomezeném vyhodnocování. Jestliže je navíc E bezpečný, pak F také.*

Jako důsledek dostáváme stejnou sílu doménového relačního kalkulu $\mathcal{DC}$ jako n-ticového relačního kalkulu $\mathcal{TC}$ a přeformulováním odpovídající vět o úplnosti $\mathcal{TC}$ i úplnost $\mathcal{DC}$.

2 Normální formy

Klíč relace můžeme jednoduše zobecnit na klíč relační relačního schématu. *Klíč* relačního schématu R je množina atributů $K \subseteq R$ taková, že pro každou přípustnou relaci $r(R)$ neexistují dvě rozdílné n-tice $t_1, t_2 \in R$ takové, že $t_1(K) = t_2(K)$ a žádná $K' \subset K$ není této vlastnosti. *Nadklíč* je množina atributů z R obsahující klíč.

Příklad 5 Klíčem relačního schématu $R_0 = \{$učebna, čas, předmět, adresa$\}$ může být množina atributů $\{$učebna, čas$\}$, nadklíčem by pak byla např. množina $\{$učebna, čas, adresa$\}$.

Převedeno do řeči funkčních závislostí (functional dependencies, FDs, viz přednášky): *Klíč* relačního schématu R je množina atributů $K \subseteq R$ taková, že každá přípustná relace $r(R)$ splňuje funkční závislost $K \to R$ a žádná $K' \subset K$ není této vlastnosti. Při testování splnění funkční závislosti $K \to R$ relací r tedy stačí ověřit implikaci: jestliže $t_1(K) = t_2(K)$, pak $t_1(R) = t_2(R)$ pro lib. dvě rozdílné n-tice $t_1, t_2 \in R$.

Předpokládejme nyní, že relační schéma R má dvě části, tj. $R = \langle S, \mathbf{K} \rangle$, kde S je množina atributů z universa $\mathbf{U}$ a $\mathbf{K}$ je množina kandidátních klíčů (kandidátní klíč může být nadklíč). *Schéma relační databáze* $\mathbf{R}$ nad $\mathbf{U}$ je potom množina relačních schémat $\{R_1, R_2, \ldots, R_p\}$, kde $R_i = \langle S_i, \mathbf{K_i} \rangle$ a $\bigcup_{i=1}^{p} S_i = \mathbf{U}$ (a $S_i \neq S_j$ pro $i \neq j$).

Příklad 6 Schéma relační databáze se dvěma relačními schématy (s množinami kandidátních klíčů):
$\mathbf{R_1} = \{\langle\{$učebna, čas, předmět$\}, \{$učebna, čas$\}\rangle, \langle\{$učebna, adresa$\}, \{$učebna$\}\rangle\}$.

Relační databáze d nad relačním schématem $\mathbf{R}$ je množina relací $\{r_1, r_2, \ldots, r_p\}$ taková, že pro každé relační schéma $R = \langle S, \mathbf{K} \rangle$ z $\mathbf{R}$ existuje v d relace $r(S, \mathbf{K})$. Relační schéma $R = \langle S, \mathbf{K} \rangle$ *reprezentuje* funkční závislost $K \to R$, jestliže K je kandidátní klíč z $\mathbf{K}$. Schéma relační databáze $\mathbf{R} = \{R_1, R_2, \ldots, R_p\}$ potom *reprezentuje* (nebo také *úplně charakterizuje*) množinu funkčních závislostí $F = \{X \to Y \mid$ nějaké R_i reprezentuje $X \to Y\}$.

Příklad 7 Schéma relační databáze $\mathbf{R_1}$ z příkladu 6 reprezentuje množinu funkčních závislostí
$F_1 = \{\{$učebna, čas$\} \to \{$učebna, čas, předmět$\}, \{$učebna$\} \to \{$učebna, adresa$\}\}$.

Databáze d nad schématem $\mathbf{R}$ *splňuje* množinu funkčních závislostí F, jestliže pro každou funkční závisost $X \to Y \in F^+$ (F^+ je uzávěr množiny F funkčních závislostí, viz přednášky) a každé relační schéma $R \in \mathbf{R}$ platí, jestliže $X \to Y$ je nad R, pak relace $r(R)$ splňuje $X \to Y$.

Nyní můžeme definovat normální formy (NF) relačních databází. *Normální forma* je omezení schématu relační databáze, které odstraňuje jisté nechtěné vlastnosti databáze. Normální formy jsou definovány pro samotné relační schéma i pro celé schéma relační databáze. Říkáme, že relační schéma je v normální formě nebo že schéma relační databáze je v normální formě. Schéma relační databáze $\mathbf{R}$ je v normální formě, jestliže každé relační schéma $R \in \mathbf{R}$ je v normální formě.

2.1 První normální forma

Relační schéma R je v *první normální formě (1NF)*, jestliže hodnoty z $dom(A)$ jsou pro každý atribut $A \in R$ atomické. To znamená, že to nejsou seznamy, množiny hodnot nebo jiné složené hodnoty. Všechny uvedené příklady byly v 1NF. Definice atomické hodnoty je vágní; hodnota může být atomická pro jednu aplikaci, ale nemusí být atomická pro jinou aplikaci. Hodnotu uvažujeme jako neatomickou, jestliže aplikace pracuje jen s částí hodnoty. Např. ve schématu {příjmení, datum_narození} mohou být hodnoty atributu datum_narození složeny z hodnot pro den, měsíc a rok; pokud by aplikace někdy pracovala jen s měsícem a/nebo rokem, pak by toto schéma nebylo v 1NF a pro dosažení 1NF by se muselo nahradit např. schématem {příjmení, den_narození, měsíc_narození, rok_narození}.

Výhody 1NF jsou zřejmé. Pokud schéma není v 1NF, dříve nebo později narazíme na problémy s funkčními závislostmi, modifikací dat aj. Např. chceme do neatomického schématu výše přidat atribut znamení. Tento atribut je jistě funkčně závislý na měsíci a dnu z atributu datum_narození, ale závislost lze vyjádřit jen na celém atributu datum_narození. Aplikace potom ale dovoluje pro dva různé lidi se stejným dnem a měsícem, ale různým rokem narození, tzn. nicméně stejným znamením, uložit různá znamení.

2.2 Druhá normální forma

Další normální formy zavádíme pro zabránění anomáliím při modifikaci dat nebo pro eliminaci redundance dat. Např. relace r nad relačním schématem R_0 z příkladu 5 by měla splňovat také funkční závislost {učebna} $\to$ {adresa}. Při přestěhování učebny na jinou adresu, tj. operaci CH(r; učebna = "..."; adresa = "..."), se ale musí upravit adresa u všech záznamů s danou učebnou, místo úpravy jednoho záznamu (mění se jen jedna informace, adresa učebny). Problém je v duplikaci informace o vztahu mezi učebnou a adresou. Způsobuje ho závislost atributu adresa na části klíče, atributu učebna, a ne celém klíči {učebna, čas}. Řešením je vydělit tento vztah do nové relace, tj. vytvořit schéma (relační databáze) $\mathbf{R_1}$ z příkladu 6. Původní relaci nad schématem R_0 pak můžeme získat (přirozeným) spojením relací nad novými relačními schématy z $\mathbf{R_1}$.

Nechť $X \to Y \in F^+$, kde F je nějaká množina funkčních závislostí nad relačním schématem R. Řekneme, že Y *částečně závisí* na X v F, jestliže existuje $X' \subset X$ taková, že

$X' \to Y \in F^+$. Jinak Y *plně závisí* na X v F.

Příklad 8 V množině funkčních závislostí
$F_0 = \{\{\text{učebna, čas}\} \to \{\text{předmět, adresa}\}, \{\text{učebna}\} \to \{\text{adresa}\}\}$ nad relačním schématem R_0 z příkladu 5 atribut adresa částečně závisí na {učebna, čas}, zatímco atribut předmět plně závisí na {učebna, čas}.

Dále atribut $A \in R$ je *klíčový* v R vzhledem k F, jestliže je obsažen v nějakém klíči R; jinak je *neklíčový*. Např. atributy učebna a čas jsou klíčové a atributy předmět a adresa jsou neklíčové v R_0 vzhledem k F_0 z příkladu 8.

Relační schéma R je v *druhé normální formě (2NF)* vzhledem k množině F funkčních závislostí, jestliže je v 1NF a každý neklíčový atribut plně závisí na každém klíči R. Nebo řečeno jinak, žádný neklíčový atribut nezávisí na vlastní podmnožině žádného klíče R. Např. relační schéma R_0 není v 2NF vzhledem k F_0, protože neklíčový atribut adresa částečně závisí na klíči {učebna, čas}. Nové schéma relační databáze $\mathbf{R_1}$ je v 2NF vzhledem k F_1 z příkladu 7.

2.3 Třetí normální forma

Příklad 9 Relační schéma
$R'_0 = \langle\{\text{učebna, čas, zkratka\_předmětu, jméno\_předmětu, obor}\}, \{\text{učebna, čas}\}\rangle$, které není v třetí normální formě vzhledem k množině funkčních závislostí
$F'_0 = \{\{\text{učebna, čas}\} \to \{\text{zkratka\_předmětu, jméno\_předmětu, obor}\}, \{\text{zkratka\_předmětu}\} \to \{\text{jméno\_předmětu, obor}\}, \{\text{jméno\_předmětu, obor}\} \to \{\text{zkratka\_předmětu}\}\}$.

Mějme nyní relaci r nad relačním schématem R'_0 z příkladu 9, která by měla splňovat také funkční závislosti $\{\{\text{zkratka}\} \to \{\text{jméno, obor}\}, \{\text{jméno, obor}\} \to \{\text{zkratka}\}\}$. Při změně předmětu v daný čas na dané učebně (např. při "prohození" předmětů), tj. operaci CH(r; učebna = "...", čas = "..."; zkratka_předmětu = "...", jméno_předmětu = "...", obor = "..."), se ale musí upravit zkratka předmětu, jméno předmětu i obor u všech záznamů s danou učebnou a časem, místo úpravy jednoho záznamu (mění se jen jedna informace, předmět v daný čas na dané učebně). Navíc nové hodnoty nesmí porušit uvedené závislosti. Opět je problém v duplikaci informace, tentokrát o vztahu mezi zkratkou předmětu a jménem předmětu spolu s oborem. Nyní ho ale nezpůsobuje částečná závislost některého neklíčového atributu na klíči. Všechny neklíčové atributy, zkratka_předmětu, jméno_předmětu a obor, jsou plně závislé na klíči {učebna, čas}; schéma tedy je v 2NF. Způsobuje ho závislost neklíčových atributů jméno_předmětu a obor na neklíčovém atributu zkratka_předmětu (který je plně závislý na klíči). Tento typ závislosti 2NF neodstraňuje. Řešení je nicméně stejné, vydělit tento vztah do nové relace, tj. vytvořit schéma (relační databáze) $\mathbf{R_2}$ z příkladu 10. Původní relaci nad schématem R'_0 pak můžeme opět získat (přirozeným) spojením relací nad novými relačními schématy z $\mathbf{R_2}$.

Příklad 10 Schéma relační databáze
$\mathbf{R_2} = \{\langle\{\text{učebna, čas, zkratka\_předmětu}\}, \{\text{učebna, čas}\}\rangle,$

$\langle\{$zkratka_předmětu, jméno_předmětu, obor$\}, \{$zkratka_předmětu$\}\rangle\}$, které je v třetí normální formě vzhledem k množine funkčních závislostí
$F_2 = \{\{$učebna, čas$\} \rightarrow \{$zkratka_předmětu$\}, \{$zkratka_předmětu$\} \rightarrow \{$jméno_předmětu, obor$\}$, $\{$jméno_předmětu, obor$\} \rightarrow \{$zkratka_předmětu$\}\}$ a které je výsledkem normalizace relačního schématu R'_0 vzhledem k množině F'_0 funkčních závislostí z příkladu 9.

Mějme relační schéma R a množinu F funkčních závislostí nad R. Pak atribut $A \in R$ je *tranzitivně závislý* na $X \subseteq R$ v R, jestliže existuje $Y \subseteq R$ taková, že $X \rightarrow Y, Y \nrightarrow X, Y \rightarrow A \in F$ a $A \notin X \cup Y$.

Příklad 11 V množině F'_0 funkčních závislostí nad relačním schématem R'_0 z příkladu 9 jsou atributy jméno_předmětu a obor tranzitivně závislé na $\{$učebna, čas$\}$, protože $\{$učebna, čas$\} \rightarrow \{$zkratka_předmětu$\}$, $\{$zkratka_předmětu$\} \nrightarrow \{$učebna, čas$\}$, $\{$zkratka_předmětu$\} \rightarrow \{$jméno_předmětu, obor$\}$ a $\{$jméno_předmětu, obor$\} \nsubseteq$ $\{$učebna, čas, zkratka_předmětu$\}$.

Relační schéma R je v *třetí normální formě (3NF)* vzhledem k množině F funkčních závislostí, jestliže je v 1NF a žádný neklíčový atribut není tranzitivně závislý na klíči R. Např. relační schéma R'_0 není v 3NF vzhledem k F'_0, protože atributy jméno_předmětu a obor jsou tranzitivně závislé na klíči $\{$učebna, čas$\}$ (skrze $\{$zkratka_předmětu$\}$). Nové schéma relační databáze $\mathbf{R_2}$ je v 3NF vzhledem k množine F_2 funkčních závislostí z příkladu 10. Víme již, že 2NF odstraňuje částečné závislosti, ale neodtraňuje tranzitivní závislosti na klíči, ale obráceně 3NF odstraňuje kromě tranzitivních závislostí také částečné závislosti na klíči. 3NF tedy pokrývá i 2NF.

Lemma 6 *Jestliže je schéma (relační nebo relační databáze) v 3NF, pak je také v 2NF.*

Důkaz. Obměnou, částečná závislost neklíčového atributu na klíči implikuje tranzitivní závislost tohoto atributu na klíči. □

2.4 Boyce-Coddova normální forma

Příklad 12 Relační schéma
$R''_0 = \langle\{$učebna, čas, učitel, adresa$\}, \{$učebna, čas$\}\rangle$, které není v Boyce-Coddově normální formě vzhledem k množině funkčních závislostí
$F''_0 = \{\{$učebna, čas$\} \rightarrow \{$učitel, adresa$\}, \{$učebna$\} \rightarrow \{$adresa$\}, \{$čas, učitel, adresa$\} \rightarrow$ $\{$učebna$\}\}$.

2NF odtraňuje částečné závislosti neklíčových atributů na klíči, 3NF dále tranzitivní závislosti neklíčových atributů na klíči. Ale co závisloti klíčových atributů na klíči? Např. relace r nad relačním schématem R''_0 z příkladu 12 by měla kromě funkční závislosti $\{$učebna$\} \rightarrow \{$adresa$\}$ splňovat také funkční závislost $\{$čas, učitel, adresa$\} \rightarrow \{$učebna$\}$. Bez této druhé funkční závislosti by byl neklíčový atribut adresa částečně závislý na klíči $\{$učebna, čas$\}$ (a schéma by nebylo v 2NF), ale druhou závislostí se stává klíčovým atributem (v klíči {čas, učitel, adresa}, v řeči funkčních závislosti, viz úvod k normálním formám). Relační schéma R_0'' již neobsahuje částečnou ani tranzitivní závislost neklíčového atributu na klíči, proto je v 2NF i 3NF vzhledem k množině F_0'' funkčních závislostí. Přesto cítíme, že i přesto, že je to nyní klíčový atribut, by měla být odstraněna jeho částečná závislost na klíči {učebna, čas} v F_0'', podobně jako v příkladu 5. Vytvoříme tedy nové schéma relační databáze $\mathbf{R_3}$, podobně jako v příkladu 6.

Příklad 13 Schéma relační databáze
$\mathbf{R_3} = \{\langle\{\text{učebna, čas, učitel}\}, \{\text{učebna, čas}\}\rangle, \langle\{\text{učebna, adresa}\}, \{\text{učebna}\}\rangle\}$, které je v Boyce-Coddově normální formě vzhledem k množine funkčních závislostí
$F_3 = \{\{\text{učebna, čas}\} \to \{\text{učebna, čas, učitel}\}, \{\text{učebna}\} \to \{\text{učebna, adresa}\},$
$\{\text{čas, učitel, adresa}\} \to \{\text{učebna}\}\}$ a které je výsledkem normalizace relačního schématu R_0'' vzhledem k množině F_0'' funkčních závislostí z příkladu 12.

Relační schéma R je v *Boyce-Coddově normální formě (BCNF)* vzhledem k množině F funkčních závislostí, jestliže je v 1NF a žádný atribut není tranzitivně závislý na klíči R. Alternativně, jestliže pro každou $Y \subseteq R$ a každý atribut $A \in R \setminus Y$ platí, jestliže $Y \to A \in F^+$, pak $Y \to R \in F^+$, tzn. jestliže Y netriviálně implikuje atribut z R, pak je Y nadklíč R. Např. relační schéma R_0'' není v BCNF vzhledem k F_0'', protože klíčový atribut adresa částečně (a tedy i tranzitivně) závisí na klíči {učebna, čas}. Nové schéma relační databáze $\mathbf{R_3}$ je v BCNF vzhledem k množině F_3 funkčních závislostí z příkladu 13. BCNF odtraňuje částečné i tranzitivní závislosti všech atributů na klíči, tedy i neklíčových. Z toho je zřejmé, že BCNF pokrývá 3NF (a tím i 2NF).

Lemma 7 *Jestliže je schéma (relační nebo relační databáze) v BCNF, pak je také v 3NF.*

Zdá se, že BCNF je ta "nejlepší" normální forma, protože pokrývá všechny předchozí normální formy. BCNF ale trpí vážnými problémy, které jí brání úspěšnému použití v praxi. Za prvé, pro nějaké množiny funkčních závislostí F nad relačním schématem R nemusí vůbec existovat schéma relační databáze v BCNF (vzhledem k F), které reprezentuje (úplně charakterizuje) F. Naproti tomu, schéma relační databáze v 3NF (vzhledem k F), které reprezentuje F, je možné nalézt pro libovolné množiny funkčních závislostí F nad relačním schématem R, viz normalizace dekompozicí relačních schémat níže. Za druhé, rozhodnutí, zda schéma relační databáze je v BCNF, je NP-úplný problém.

2.5 Normalizace dekompozicí relačních schémat

Normalizace relačního schématu nebo schématu relační databáze, které není v normální formě vzhledem k množině funkčních závislostí je transformace na schéma relační databáze, které je v normální formě vzhledem k téže množině funkčních závislostí a které reprezentuje tuto množinu funkčních závislostí. Každé schéma, které není v 1NF, 2NF nebo 3NF, je možné normalizovat na schéma, které je v 1NF, 2NF nebo 3NF. Pro BCNF nemusí být normalizace pro některá schémata a množiny funkčních závislotí možná. Normalizace do 1NF znamená jednoduché nahrazení atributů s neatomickými hodnotami atributy s atomickými hodnotami, které dohromady tvoří původní neatomické hodnoty. Pro 2NF, 3NF a BCNF normalizace znamená odstranění všech častečných (2NF) a tranzitivních (3NF) závislostí neklíčových nebo i klíčových (BCNF) atributů na klíči a v předchozích částech věnovaných jednotlivým normálním formám byla na příkladech naznačena normalizace dekompozicí relačních schémat. Dekompozice relačního schématu R znamená rozložení schématu na dvě relační schémata R_1 a R_2 (případně se překrývající) taková, že každá relace nad schématem R splňující množinu funkčních závislostí F je bezeztrátově rozložena (dekomponována) na projekce relace na schémata R_1 a R_2. Bezeztrátově znamená, že (přirozeným) spojením projekcí získáme zpět původní relaci, tj. $\pi_{R_1}(r) \bowtie \pi_{R_2}(r) = r$. V procesu normalizace dekompozici opakujeme pro nová schémata R_1 a R_2, pokud tato nejsou v normální formě (vzhledem k F); relační schémata dekomponujeme, dokud nejsou všechna v normální formě (vzhledem k F). Při každé dekompozici obsahují nová relační schémata méně atributů a vzhledem k tomu, že relační schéma obsahující pouze dva atributy nemůže porušit podmínky žádné z normálních forem 2NF, 3NF a BCNF, normalizace dekompozicí relačních schémat vždy skončí.

Dekompozice relačního schématu probíhá následovně. Mějme relační schéma R a množinu F funkčních závislostí nad R s tranzitivní nebo částečnou (a tudíž také tranzitivní) závislostí atributu $A \in R$ na klíči $K \subseteq R$, tzn. existuje $Y \subseteq R$, kde $K \to Y, Y \nrightarrow K, Y \to A \in F$ a $A \notin K \cup Y$. Definujeme nová relační schémata $R_1 = R \setminus A$ a $R_2 = Y \cup A$, kde množina kandidátních klíčů schématu R_1 je stejná jako množina $\mathbf{K}$ kandidátních klíčů schématu R a množina kandidátních klíčů schématu R_2 je $\{Y\}$. Tímto jsme odstranili jednu tranzitivní závislost z R a přitom pro každou relaci $r(R)$ splňující F platí $\pi_{R_1}(r) \bowtie \pi_{R_2}(r) = r$. Dekompozici můžeme mírně vylepšit tím, že místo ostranění tranzitivní závislosti jednoho atributu A na klíči K odstraníme najednou závislost všech atributů $A_1, A_2, \ldots, A_m$, které jsou závislé na Y, protože ty jsou také tranzitivně závislé na klíči K: $R_1 = R \setminus \{A_1, A_2, \ldots, A_m\}$ a $R_2 = Y \cup \{A_1, A_2, \ldots, A_m\}$. Příklady dekompozicí jsme viděli v předchozích částech věnovaných jednotlivým normálním formám, 2NF, 3NF a BCNF.

Dodejme, že výsledné normalizované schéma relační databáze získané normalizací dekompozicí relačních schémat není jednoznačné, tj. k jednomu nenormalizovanému schématu může existovat více různých normalizovaných schémat. Důvodem je možnost volby tranzitivní závislosti, pokud je jich víc, kterou chceme v dekompozičním kroku normalizace ostranit, tzn. volby, na jaká podschémata relační schéma dekomponovat.