

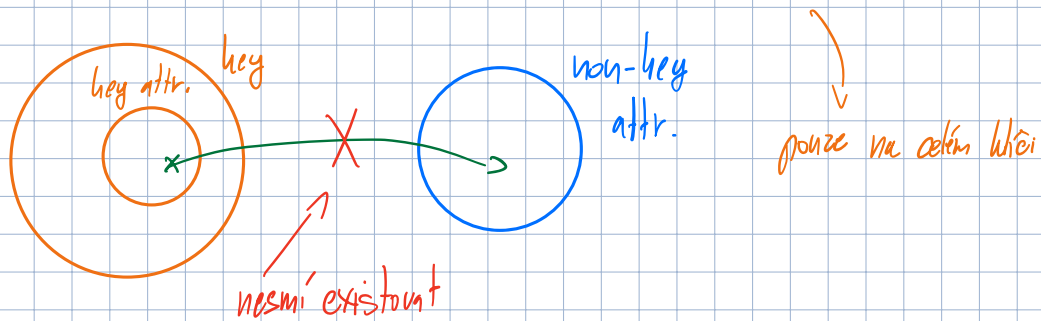
NORMÁLOVÉ FORMY

1. NF:

- všechny atributy jsou v základním tvaru.

2. NF

- žádný neklíčový atribut není závislý na jakémkoli klíčovém atributu.



Řešení: rozdělit tabulky:

Company	Server	HQ	Date
...	...	...	...

klíč Company, Server → everything
Company → HQ → violace s 2NF

Rozdělím na:

Company	Server	Date
...	...	...

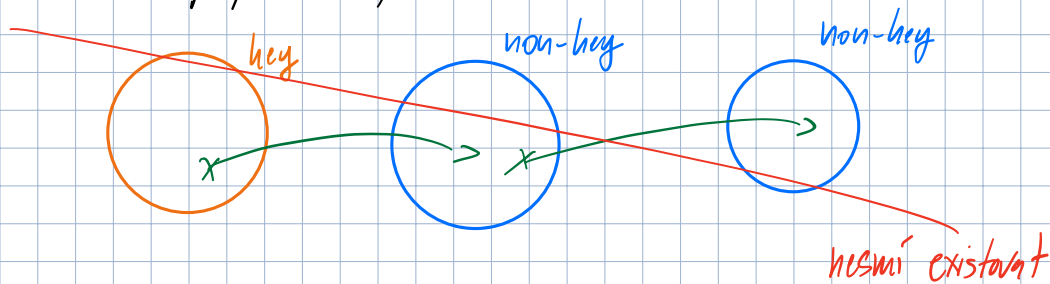
Company	HQ
...	...

Company, Server → everything

Company → HQ

3. NF

- žádný neklíčový atribut nesmí být tranzitivně závislý na klíči skrz nějaký neklíčový atribut.



Řešení: rozdělit tabulky:

Company	PostCode	HQ
...	...	...

Company $\rightarrow$ everything

PostCode $\rightarrow$ HQ $\rightarrow$ violace s 3NF

Rozděním na:

Company	PostCode
...	...

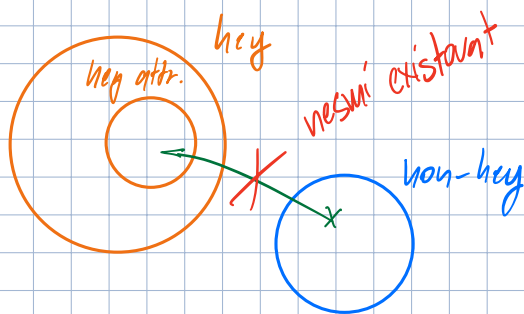
Company $\rightarrow$ everything

PostCode	HQ
...	...

PostCode $\rightarrow$ HQ

BCNF:

- nějaký klíč závisí na neklíčovém atributu.



Řešení: rozdělit tabulky

Plane	Destination	Pilot	Day
...	...	...	...

Pilot, Day $\rightarrow$ everything

Plane, Day $\rightarrow$ everything

Destination $\rightarrow$ Pilot

Rozděním na:

Plane	Destination	Day
...	...	...

Plane, Day $\rightarrow$ everything

Destination	Pilot
...	...

Destination $\rightarrow$ Pilot

violace s BCNF $\rightarrow$

Atributový uzavěr:

- snaží se pomocí funkčních pravidel maximálně rozšířit množinu dosažitelných atributů.
- řeším $\{b, c\}^+ = ?$

Testování nálezitosti

- máme-li $X \rightarrow Y$ a funkci závislosti F , chceme vědět, jestli $Y \in X^+$ vzhledem k F .
- řeším pomocí atributového uzavěru.

Testování redundance

- máme-li $X \rightarrow Y$ a funkci závislosti F , chceme vědět, jestli $\{F - \{X \rightarrow Y\}\}^+ = \{F\}^+$, resp. jestli $Y \in \{F - \{X \rightarrow Y\}\}^+$
- řeším pomocí testování nálezitosti (pokud je Y i v redukovaném F , je $X \rightarrow Y$ redundantní)

Redukce atributů

- chceme co nejmenší levé strany všech funkčních pravidel
- zkusíme, jestli zmenšením levé strany čím dál více v uzavěru celou pravou stranu.

Získání minimálního pokrytí

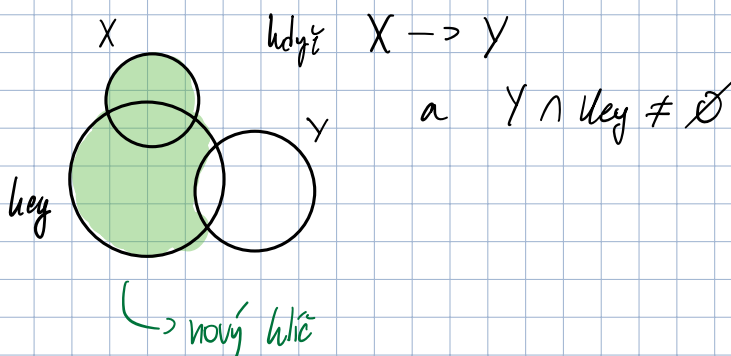
- rozdělím na elementární pravidla: vpravo jenom vždy jeden atribut
- zredukuje atributy levých stran funkčních pravidel $\rightarrow$ pomocí redukce atributů
- odstraní redundantní pravidla $\rightarrow$ pomocí testování redundance

Získání klíče:

- Provedu redukci atributů z everything $\rightarrow$ everything vzhledem k množině funkčních závislostí F .

Nalezení všech klíčů

- nalezneme jeden klíč a pak provádíme:



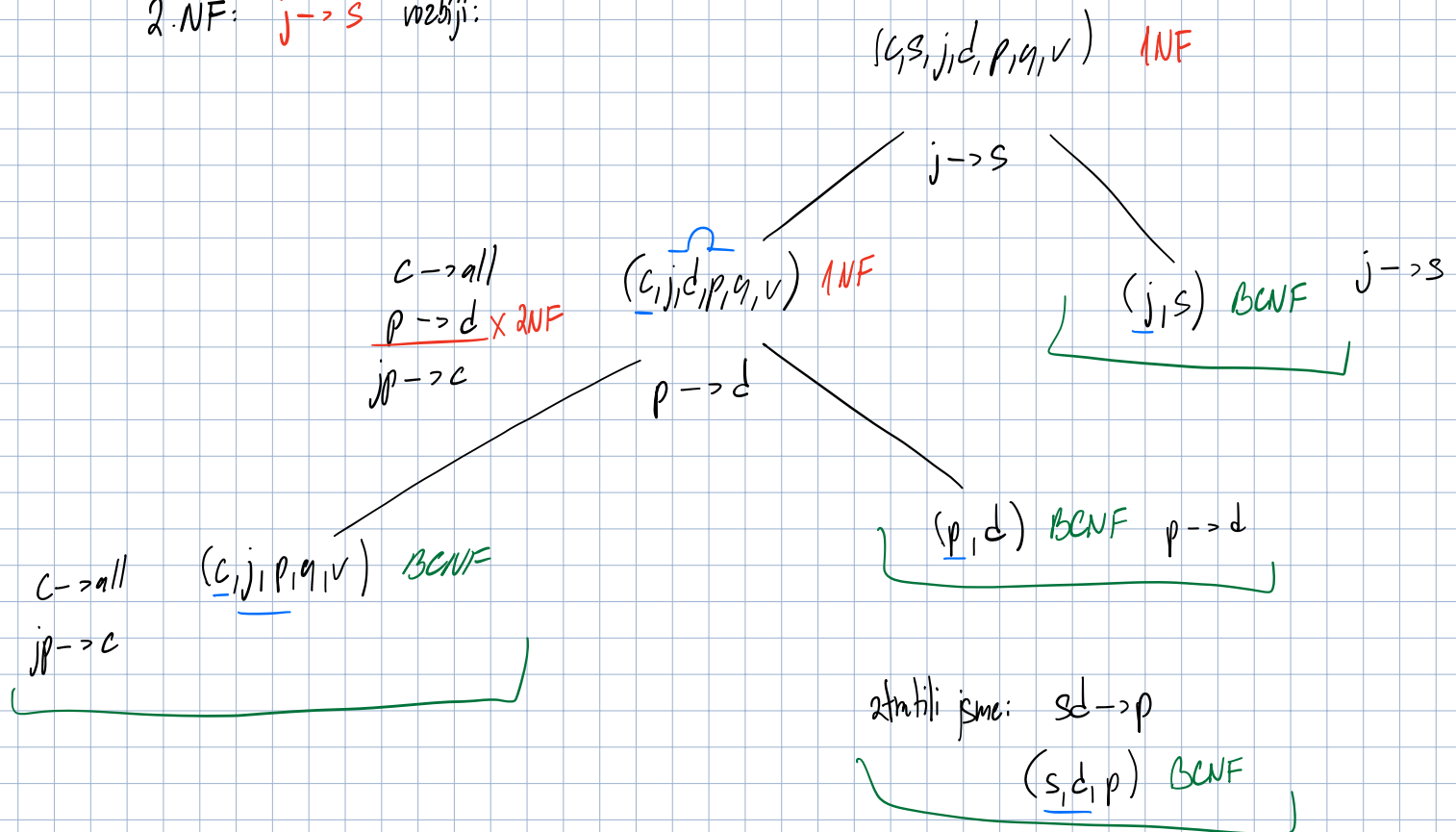
PROCVIČENÍ

$R(\underline{c}, s, \underline{j}, d, p, q, v)$ *útlík*

$F = \{ c \rightarrow all, sd \rightarrow p, p \rightarrow d, jp \rightarrow c, j \rightarrow s \}$

1.NF: 2 předpoklady

2.NF: $j \rightarrow s$ *vezbít:*



$R(a, b, c, d, e, f)$

$F = \{ a \rightarrow b, bd \rightarrow ae, ef \rightarrow b \}$

1) nalezení všech klíčů

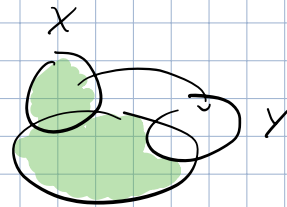
$abedef \rightarrow abedef \sim acdef \rightarrow abedef \sim \underline{acdf} \rightarrow abedef$

$\underline{bcdf}$ $a \rightarrow b$

- zkusím minimalizovat

$cdf^+ = cdf$ $\times$
 $bdf^+ = abdef^+ = abdef$ $\times$
 $bef^+ = bef$ $\times$
 $bcd^+ = abede^+ = abede$ $\times$

Tedy nelze minimalizovat a jde o celý klíč



$\underline{cdef}$ - zkusím minimalizovat

$def^+ = bdef^+ = abdef^+ = abdef$ $\times$
 $cef^+ = bcef^+ = bcef$ $\times$
 $cdf^+ = cdf$ $\times$
 $cde = cde$ $\times$

Nelze minimalizovat, jde o klíč

převod na BCNF:

(a, b, c, d, e, f)

$a \rightarrow b$
 $bd \rightarrow ac$
 $ef \rightarrow b$

1NF

2NF

3NF

Všechno jsou neklíčové atributy

BCNF: $a \rightarrow b$ pro $bcd \rightarrow all$

(a, c, d, e, f) $cdef \rightarrow a$
 $acdf \rightarrow e$

BCNF

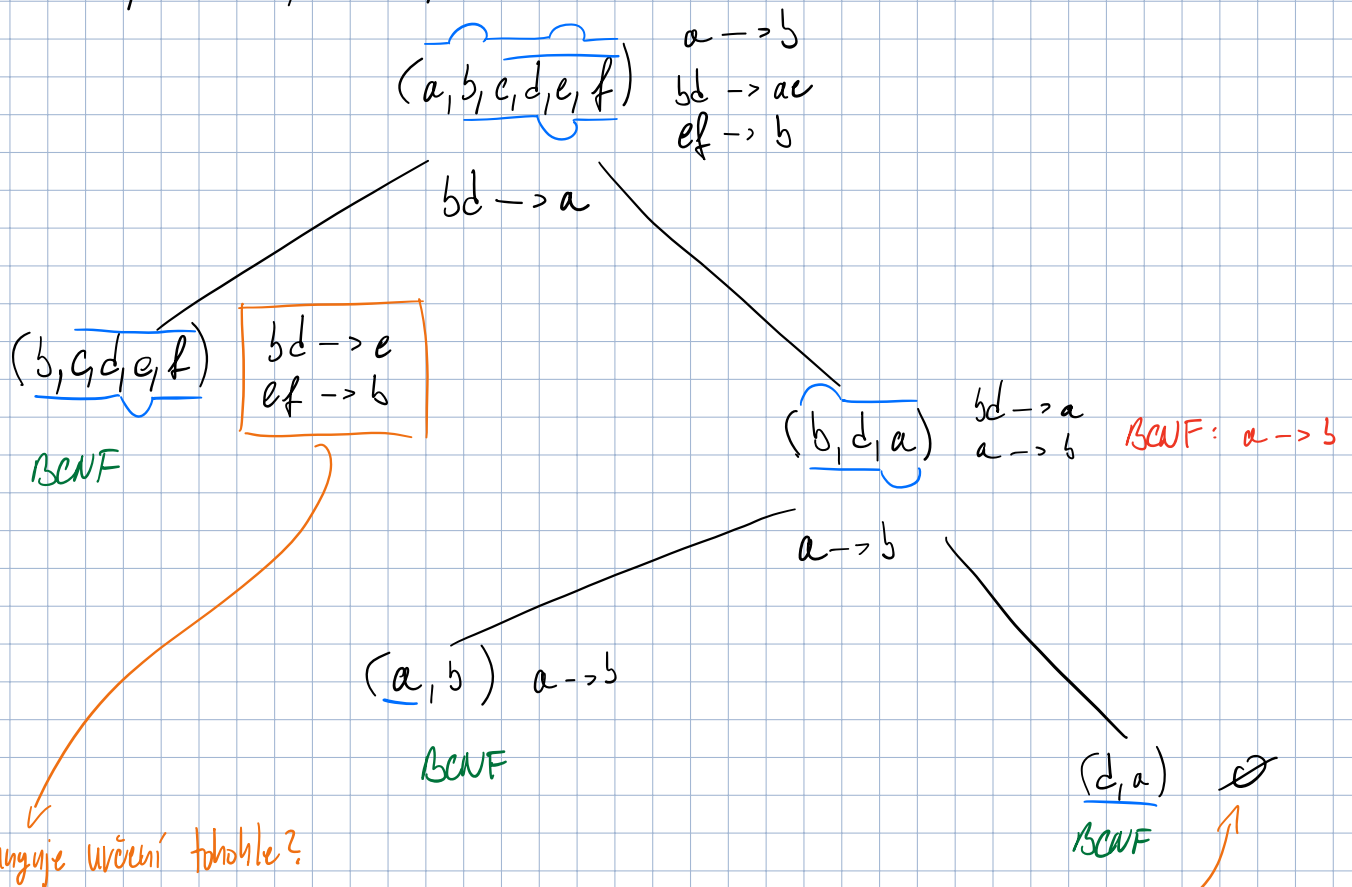
(a, b) $a \rightarrow b$

BCNF

$\angle$ neexistuje žádné pravidlo
 neklíčový atribut $\rightarrow$ klíčový atribut

Můžeme zkusit rozdělit podle jiného pravidla:

$a \rightarrow b$, $bd \rightarrow a$, $bd \rightarrow c$, $ef \rightarrow b$



Jak funguje určení tabulky?

již mi nechýl žádná závislost pro a, d

Funkční závislosti:

$$A = \{a, b, c, d, e, f, g\}$$

$$F2 = \{a \rightarrow b, b \rightarrow a, c \rightarrow d, cde \rightarrow ag, d \rightarrow ef\}$$

1) Nalezněte minimální pokrytí:

$$a \rightarrow b$$

$$b \rightarrow a$$

$$c \rightarrow d$$

$$cde \rightarrow a$$

$$cde \rightarrow g$$

$$d \rightarrow e$$

$$d \rightarrow f$$

$$cd \rightarrow a, c \rightarrow a$$

$$cd \rightarrow g, c \rightarrow g$$

$$cde \xrightarrow{?} a : de^+ = defx$$

$$cde \xrightarrow{?} a : ce^+ = cedagfb \checkmark$$

$$cd \xrightarrow{?} a : c^+ = cdefagb \checkmark$$

Podobně

$$F2' = \{$$

$$a \rightarrow b$$

$$b \rightarrow a,$$

$$c \rightarrow adg$$

$$d \rightarrow ef\}$$

2) Nalezněte všechny klíče

$$abcdefg \rightarrow abcdefg$$

$$acdefg \rightarrow abcdefg$$

$$cefg \rightarrow abcdefg$$

$$\text{První klíč} = \underline{ce}$$

nemáme další pravidla,

klíč by mohl c jako podmnožinu

první strany.

kandidátní klíče cef

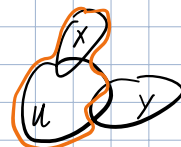
prvotní redukce:

$$ef^+ = ef$$

$$cf^+ = abcdefg \checkmark$$

$$f^+ = f$$

$$\underline{c^+ = abcdg^+ = abcdefg}$$



Určete, které F2 pokrývá [2/3/BC] NF

$$a \rightarrow b \rightarrow 3NF$$

$$b \rightarrow a \rightarrow 3NF$$

$$c \rightarrow d \checkmark$$

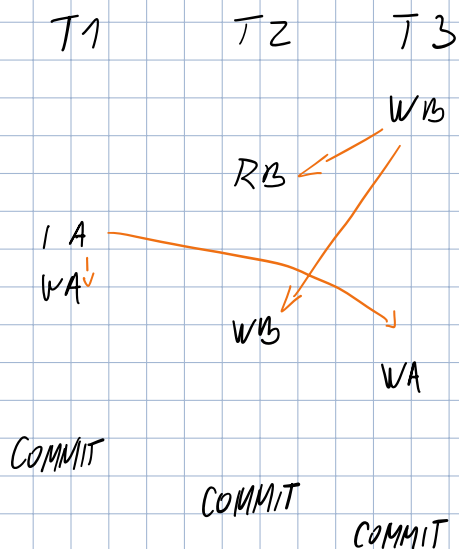
$$c \rightarrow a \checkmark$$

$$c \rightarrow g \checkmark$$

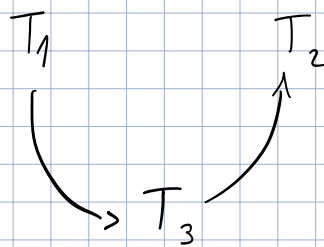
$$d \rightarrow e \rightarrow 3NF$$

$$d \rightarrow f \rightarrow 3NF$$

Transakce



Konfliktní uspořádatelnost? **ANO**



Pohlednutí -/-? **ANO** protože konfliktů

Uspořádatelnost? **ANO** protože pohlednutí

Zatvrditelnost? **NE**, T2 čte B před commitem T3.

S2PL? **NE**, jinak by to bylo zatvrditelné

NO-SQL

Mějme $q = (2, 4)$, $s_1 = (1, 14)$, $s_2 = (8, 10)$

Učíte vzdálenosti Manhattan a Euklid.

			M	E
q	2	4		
s ₁	1	14	11	$\sqrt{101}$
s ₂	8	10	12	$\sqrt{72}$

SQL $\rightarrow$ RA

Select * from Obchod O inner join Prodává X on (X.OID = O.OID) where Cena < 10

Obchod [O.OID = P.OID] Proává (Cena < 10) ✓

(Obchod [Obchod.OID = Proává.OID] Proává) (Cena < 10)

Select * from Produkt P where PID not in (Select PID from Proává)

Produkt - Produkt [PID = PID] Proává [PID, název, váha] ✓

Produkt - Produkt [Produkt.PID = Proává.PID] Proává [PID, název, váha]

Produkt * Proává - protože mám jen jeden společný sloupec

Select OID from Obchod O where (Select Count(*) from Proává X where X.OID = O.OID) = (Select Count(*) from Produkt)

- vybrání všech obchodů, co prodávají všechny produkty

[OID] = [OID, PID] $\div$ [PID]

Proává [OID, PID] $\div$ Produkt [PID]

$\rightarrow$ všechny produkty

(Proává $\div$ Produkt [PID]) [OID] DEFINICE !

$\hookrightarrow$ tahle vna všechny dvojice OID, Cena

Relační algebra

Základní operace:

přejmenování	$R \langle a \rightarrow x, b \rightarrow y \dots \rangle$
selece	$R(\text{podmínka})$
projekce	$R[a, b, \dots]$
spojení	$R * S$, resp. $R[\text{podm.}] S$
dělení	$R \div S$ - jméno lin, co promítalo všechny filmy: $\text{Promítání}(\text{Kino}, \text{Název}) \div \text{Filmy}(\text{Název})$
rozdíl	$R - S$ - musí být shodný schéma
součet	$R + S$ - musí být shodný schéma

↳ pokud dělení a pak
zpět spojíme všem možným
vzhledům, výslední mi vznikne
nadmnožina.

Příklady:

Seznam promítaných kin:

Kino (Město = Praha)

Filmy od Bensona

Filmy (Director = Benson)

Seznam jmen kin, co promítali něco od Bensona

Filmy [Film.program = Program.jmeno] Program (Film.Director = Benson) [NázevKino]

Seznam jmen kin, co nepromítali nic od Bensona

Kino [NázevKino] - Filmy [Film.program = Program.jmeno] Program (Film.Director = Benson) [NázevKino]

Seznam jmen kin, co promítali jenom Bensona

Kino [NázevKino] - Filmy [Film.program = Program.jmeno] Program (Film.Director $\neq$ Benson) [NázevKino]

Seznam jmen kin, co promítali všechny filmy od Bensona

Program [NázevKino, Název] $\div$ Filmy (Director = Benson) [Název]