



Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky



Úvod do databázových systémů


Cvičení 12

Ing. Petr Lukáš
petr.lukas@vsb.cz
Ostrava, 2014

- **Univerzální relační schéma**
- **Funkční závislost**
- **Armstrongovy axiomy**
- **Uzávěry množiny atributů**
- **Klíč schématu**
- **Minimální neredundantní pokrytí**



- **Univerzální relační schéma**
„široká“ nepřehledná tabulka, která obsahuje všechny atributy (např. modelovaného systému)
- **Funkční závislost**
ze znalosti hodnot nějaké množiny hodnot atributů (a samozřejmě obsahu databáze) znám i množinu hodnot jiných atributů.
- **Armstrongovy axiomy**
odvozovací pravidla pro funkční závislosti.
- **Uzávěry množiny atributů**
hodnoty kterých všech atributů jsem schopný získat na základě určité dané množiny atributů
- **Klíč schématu**
atributy, kterými jednoznačně identifikuji celý záznam, tzn. pokud znám hodnoty těchto atributů, umím v univerzálním schématu dohledat obsah celého záznamu
- **Minimální neredundantní pokrytí**
rozložím FZ na elementární FZ, z levých stran odstraním nadbytečné atributy a nakonec celé nadbytečné FZ.

- **Redundance, konsistence, integrita**
 - **Normální formy relací**
 - **Algoritmický návrh databáze**
- 

Redundance, konsistence, integrita

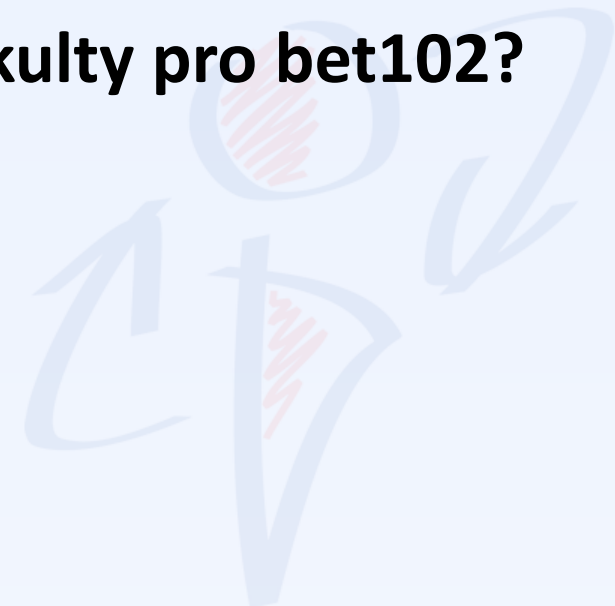


login	jmeno	prijmeni	id_fakulty	nazev_fakulty
kab242	Prokop	Kabel	1	FEI
bet102	Zdislava	Betonová	1	FEI
jed135	Tomáš	Jedno	2	FBI



login	jmeno	prijmeni	id_fakulty	nazev_fakulty
kab242	Prokop	Kabel	1	FEI
bet102	Zdislava	Betonová	1	FEI → EKF
jed135	Tomáš	Jedno	2	FBI

- **Co se stane, když změním název fakulty pro bet102?**



login	jmeno	prijmeni	id_fakulty	nazev_fakulty
kab242	Prokop	Kabel	1	FEI
bet102	Zdislava	Betonová	1	FEI → EKF
jed135	Tomáš	Jedno	2	FBI

- **Co se stane, když změním název fakulty pro bet102?**
Stejnému ID fakulty bude pokaždé odpovídat jiný název a to je zjevně nesmysl. Data budou tzv. nekonzistentní.

login	jmeno	prijmeni	id_fakulty	nazev_fakulty
kab242	Prokop	Kabel	1	FEI
bet102	Zdislava	Betonová	1	FEI → EKF
jed135	Tomáš	Jedno	2	FBI

- **Co se stane, když změním název fakulty pro bet102?**
Stejnému ID fakulty bude pokaždé odpovídat jiný název a to je zjevně nesmysl. Data budou tzv. nekonzistentní.
- **Co se stane, když smažu záznam pro jed135?**

login	jmeno	prijmeni	id_fakulty	nazev_fakulty
kab242	Prokop	Kabel	1	FEI
bet102	Zdislava	Betonová	1	FEI → EKF
jed135	Tomáš	Jedno	2	FBI

- **Co se stane, když změním název fakulty pro bet102?**
Stejnému ID fakulty bude pokaždé odpovídat jiný název a to je zjevně nesmysl. Data budou tzv. nekonzistentní.
- **Co se stane, když smažu záznam pro jed135?**
Pokud nebudu mít k dispozici jinou tabulku s fakultami, nadobro ztratím informace o fakultě FBI.

- **Redundance**

Znamená obecně nadbytečnost dat. Stejná data jsou v databázi uchovávána vícekrát a to je obvykle **nežádoucí jev**.

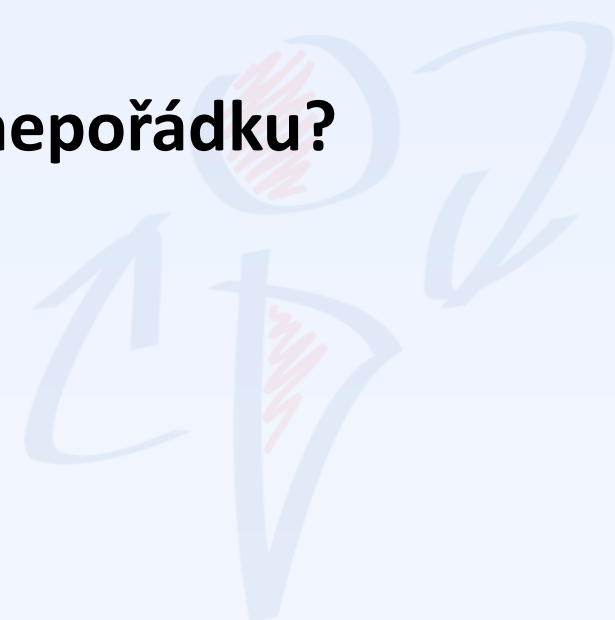
- **Konzistence**

Když už redundance nastane (což se prakticky může stát), musíme zajistit, aby data nebyla nekonzistentní (viz příklad na předchozím slidu). Tzn. konzistence je naopak **žádoucí jev**.

Nekonzistence je důsledkem redundance.

id_republiky	nazev	hlavni_mesto	Prezident
1	Česká republika	Praha	Václav Havel
2	USA	Washington	Arnold Schwarzenegger
3	Německo	Bonn	Richard von Weizsäcker

- **Co je na datech na první pohled v nepořádku?**



id_republiky	nazev	hlavni_mesto	Prezident
1	Česká republika	Praha	Václav Havel
2	USA	Washington	Arnold Schwarzenegger
3	Německo	Bonn	Richard von Weizsäcker

- **Co je na datech na první pohled v nepořádku?**
Při troše základního vzdělání minimálně víme, že dnes, v roce 2013, Václav Havel určitě už není současný prezident ČR, Arnold není prezidentem USA (i když by si to asi přál) a Bonn není současným hlavním městem Německa. Data nejsou integritní.

- **Integrita**

Integrita dat znamená soulad se zobrazovanou realitou. Integrita je **žádoucí jev**.

Integritou také rozumí, že jsou dodržena všechna definovaná integritní omezení v databázi, tzn. např. neexistuje cizí klíč, který by se odkazoval na neexistující ID.

login	jmeno	prijmeni	id_fakulty		id_faulkty	nazev_fakulty
kab242	Prokop	Kabel	1	→	1	FEI
bet102	Zdislava	Betonová	2	→	2	FBI
jed135	Tomáš	Jedno	3	→		?

Normální formy relací



Relace je v 1. normální formě, jestliže obsahuje pouze atomické (dále nedělitelné) atributy.

login	jmeno	adresa
kab242	Prokop	Za Mostem 23, Ostrava-Svinov
bet102	Zdislava	710 00, Slezská Ostrava, Bohumínská 22
jed135	Tomáš	U točny 2, Psojedy, Česká republika

- Ukázková relace porušuje pravidlo pro 1. normální formu. Pokud bychom např. chtěli hromadně předtisknout obálky, asi bychom s takto uvedenou adresou moc dobře nepořídili.
- Na druhou stranu co je to „atomický“ atribut není jednoznačně dáno. Co má např. obsahovat atribut *ulice*? Atomicita je potřeba citlivě volit podle účelu systému.

Relace je ve 2. normální formě, jestliže je v 1. NF a každý neklíčový atribut (tj. ten, který není součástí žádného klíče) je úplně závislý na každém klíči. Tzn. není závislý na žádném podklíči.

<u>hodina</u>	<u>ucebna</u>	predmet	kapacita
10:45	A1033	UDBS	16
7:15	G317a	DAIS	20
9:30	A1033	MAIT	16

- Ukázka opět porušuje 2. NF, protože platí funkční závislost **ucebna** → **kapacita**, ale klíčem relace jsou dohromady **hodina** a **ucebna**. Tzn. kapacita je závislá na podklíči a to je špatně.

Relace je ve 3. normální formě, jestliže je ve 2. NF a neexistují netriviální závislosti mezi neklíčovými atributy.

<u>login</u>	jmeno	prijmeni	id_fakulty	nazev_fakulty
kab242	Prokop	Kabel	1	FEI
bet102	Zdislava	Betonová	1	FEI
jed135	Tomáš	Jedno	2	FBI

- Název fakulty a její id jsou dva neklíčové atributy, mezi kterými existuje funkční závislost **id_fakulty** → **nazev_fakulty**. Tato relace tedy není ve 3. normální formě.

Relace je v BCNF, jestliže pro každou netriviální funkční závislost $X \rightarrow Y$ platí, že X je klíč nebo nadklíč.

<u>sluzba</u>	<u>zakaznik</u>	tarif
internet	Petr	10 GBit/s
internet	Jan	100 GBit/s
telefon	Petr	SUPER
telefon	Pavel	EXTRA

- V uvedeném příkladu je problém v tom, že existuje funkční závislost **tarif** \rightarrow **sluzba**. Tarify *SUPER* a *EXTRA* jsou vždy pro telefonní službu, zatímco *10 GBit/s* a *100 GBit/s* se vždy týkají internetu. V relaci tedy existuje závislost, kde levá strana netvoří klíč nebo nadklíč.

Algoritmický návrh databáze

A faint, light blue watermark logo is visible in the background on the right side of the slide. It consists of several curved lines and a central red hatched area, resembling a stylized letter or symbol.

- **Univerzální schéma**

Ať už zvolíme kterýkoli algoritmus pro návrh databáze, prvním krokem je vždy sestavení univerzálního schématu ***RU***, tj. předpisu pro tabulku s komplet všemi atributy.

- **Dekompozice univerzálního schématu**

Existují dva základní algoritmy

- **Algoritmus dekompozice do BCNF**
- **Algoritmus dekompozice do 3.NF** (často též nazýván jako algoritmus syntézy)

- Najdeme FZ $X \rightarrow Y$, která porušuje podmínku BCNF, a provedeme rozklad schématu R na dvě schémata R_1 a R_2 .

Schéma R_1 bude obsahovat všechny atributy R bez atributů Y , druhé schéma R_2 bude obsahovat dohromady atributy X a Y .

- Rozkládáme tak dlouho, dokud existují relace a FZ, které porušují podmínku BCNF.
- Bohužel, záleží na tom, v jakém pořadí vytahujeme jednotlivé FZ a může dojít ke **ztrátě informace** nebo **ztrátě funkční závislosti**.

R (A, B, C, D, E, F, G)

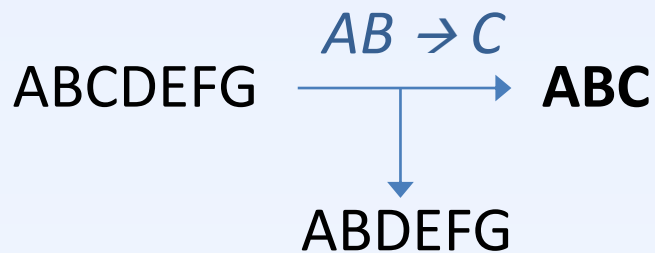
F: {AB \rightarrow C, C \rightarrow D, B \rightarrow E, E \rightarrow F, C \rightarrow G}

ABCDEFGG



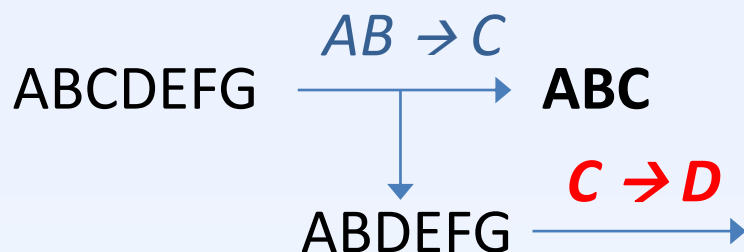
R (A, B, C, D, E, F, G)

F: { $AB \rightarrow C$, $C \rightarrow D$, $B \rightarrow E$, $E \rightarrow F$, $C \rightarrow G$ }



R (A, B, C, D, E, F, G)

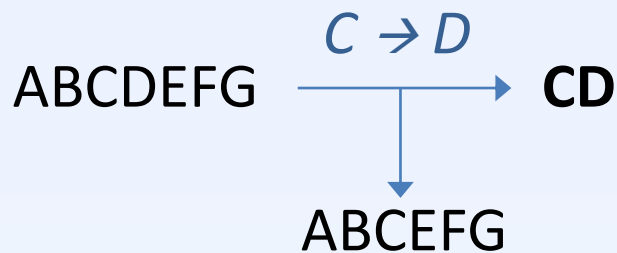
F: { $AB \rightarrow C$, $C \rightarrow D$, $B \rightarrow E$, $E \rightarrow F$, $C \rightarrow G$ }



Problém, pokud v prvním kroku použiju neuváženě $AB \rightarrow C$, dostanu se do pasti, protože už nebudu moct nikde uplatnit $C \rightarrow D$. To znamená, že jsem přišel o funkční závislost a poruším tedy zákon o zachování množiny FZ.

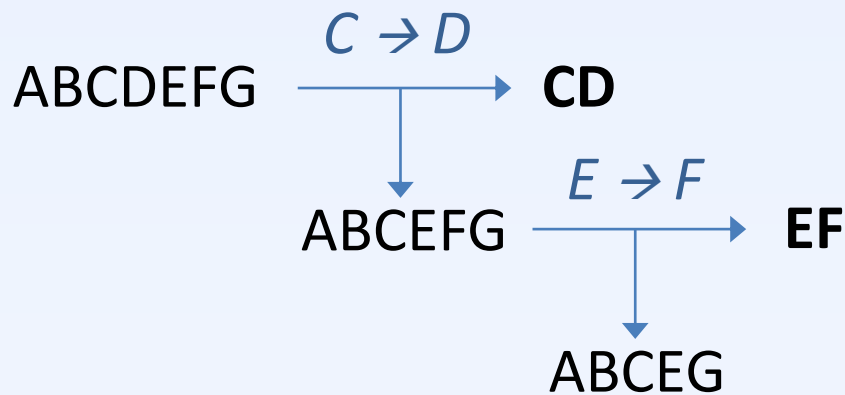
R (A, B, C, D, E, F, G)

F: {AB \rightarrow C, C \rightarrow D, B \rightarrow E, E \rightarrow F, C \rightarrow G}



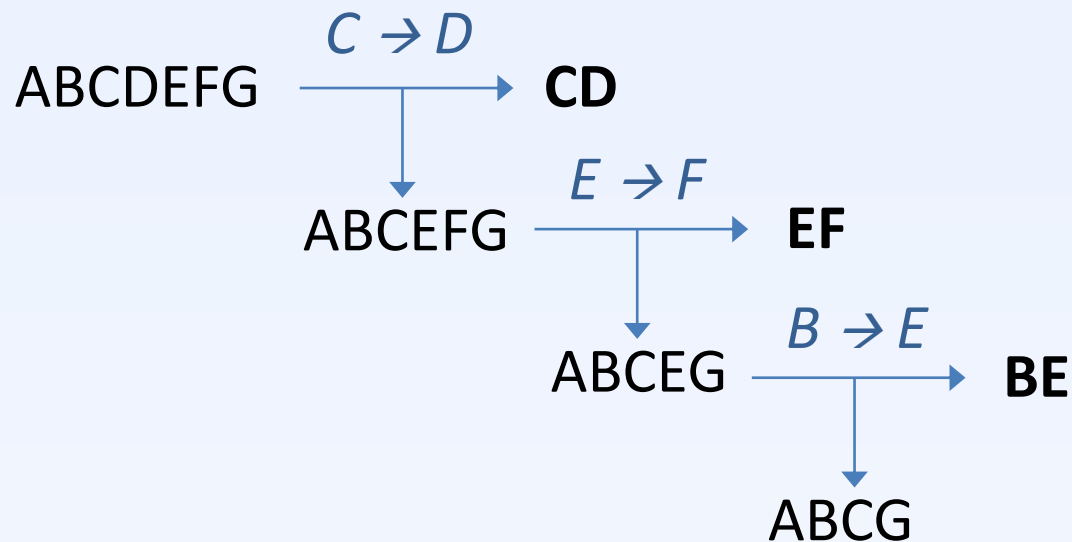
R (A, B, C, D, E, F, G)

F: { $AB \rightarrow C$, $C \rightarrow D$, $B \rightarrow E$, $E \rightarrow F$, $C \rightarrow G$ }



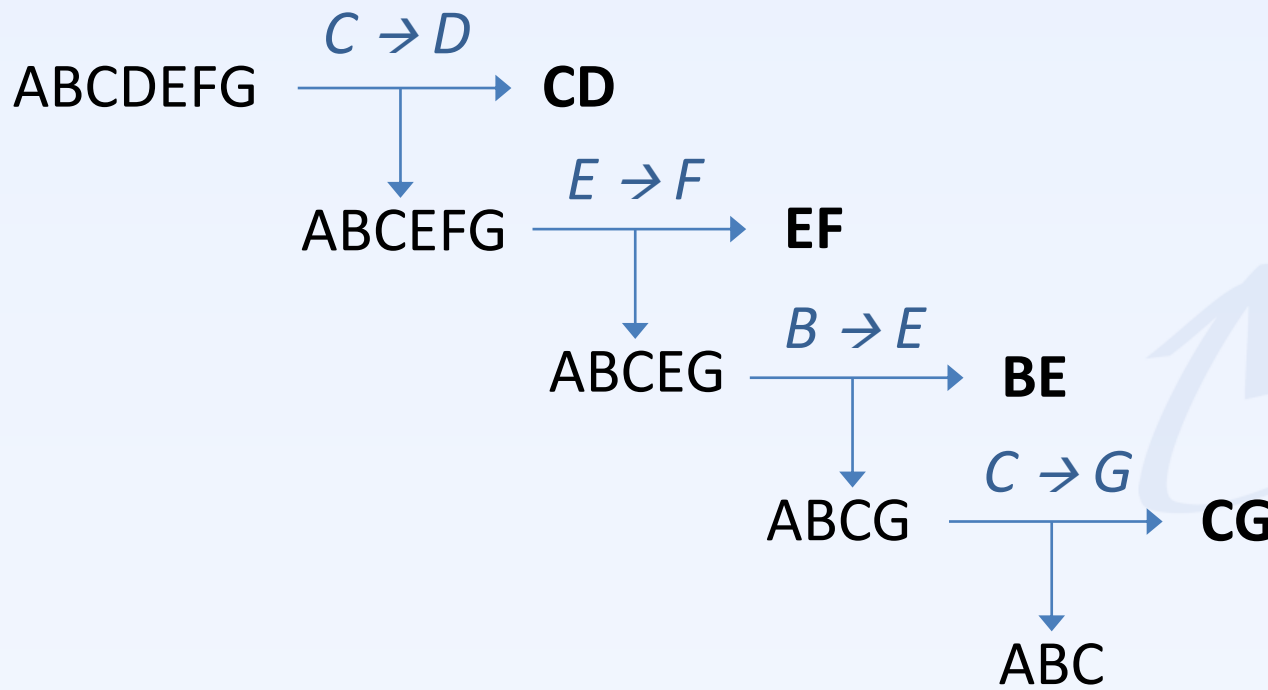
R (A, B, C, D, E, F, G)

F: { $AB \rightarrow C$, $C \rightarrow D$, $B \rightarrow E$, $E \rightarrow F$, $C \rightarrow G$ }



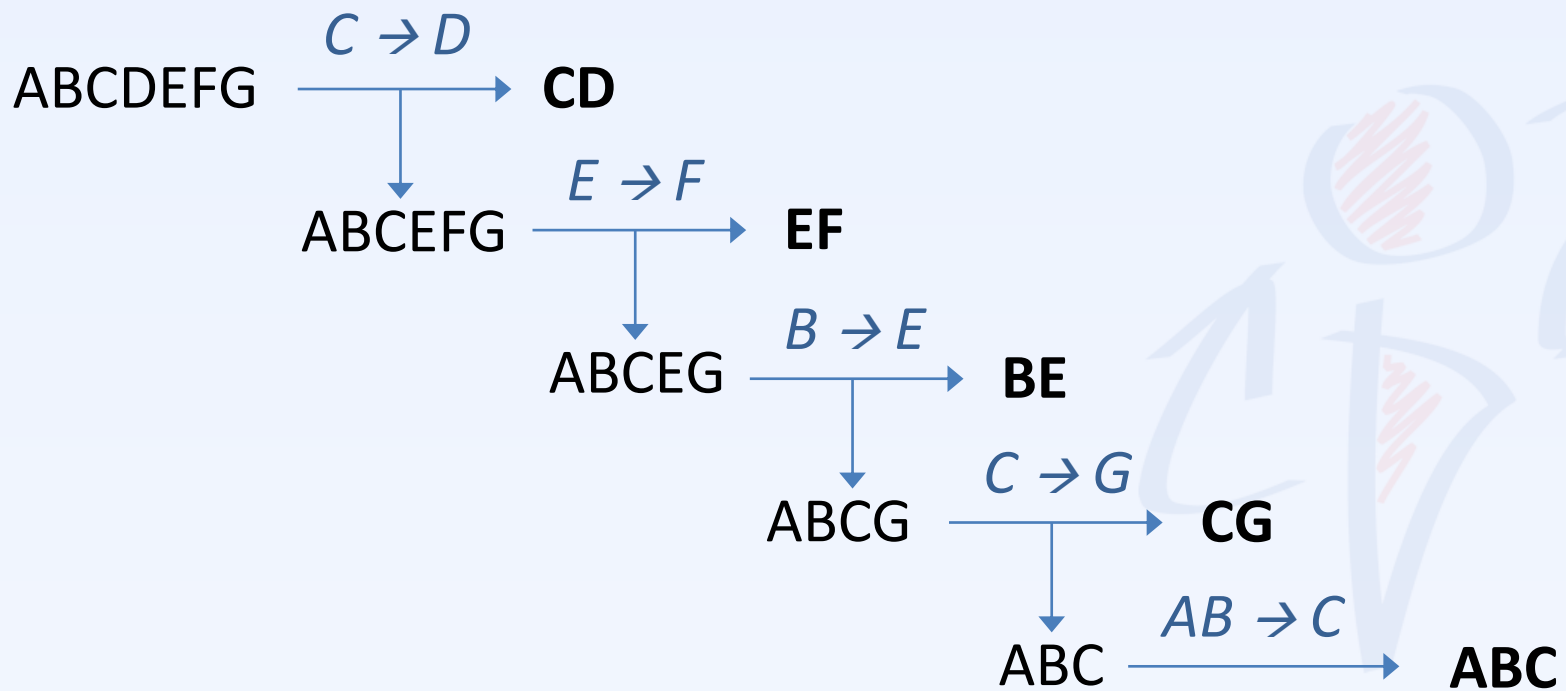
R (A, B, C, D, E, F, G)

F: { $AB \rightarrow C$, $C \rightarrow D$, $B \rightarrow E$, $E \rightarrow F$, $C \rightarrow G$ }



R (A, B, C, D, E, F, G)

F: { $AB \rightarrow C$, $C \rightarrow D$, $B \rightarrow E$, $E \rightarrow F$, $C \rightarrow G$ }



Získali jsme tedy rozklad

$R_1(CD), R_2(EF), R_3(BE), R_4(CG), R_5(ABC)$

- Je jasné, že výsledná schémata musí být v BCNF, protože jestliže existovala FZ, která porušovala BCNF, pak podle této FZ musel proběhnout rozklad.

1. Sestavíme minimální neredundantní pokrytí množiny FZ a nalezneme klíč. Pro každou FZ vytvoříme zvlášť relaci. V prvním kroku je přesně tolik „tabulek“, kolik je FZ v min. nered. pokrytí.
2. Spojíme relace které vznikly z FZ se stejnými levými stranami.
3. Spojíme relace s ekvivaltními klíči, tzn. těmi klíči, jejichž uzávěr je stejný. V tomto kroku může dojít k porušení podmínek pro BCNF!
4. Pokud existuje atribut univerzálního schématu, který dosud není zařazen v žádné ze vzniklých relací (tzn. nebyl obsažen v žádné FZ), pak jej přidáme do libovolné vzniklé relace.
5. Pokud žádná z relací neobsahuje celý klíč univerzálního schématu, pak vytvoříme novou relaci, která se bude skládat z atributů tohoto klíče.

R (A, B, C, D, E, F, G)

F: {AB → C, C → D, B → E, E → F, C → G}

1. Minimální neredundantní pokrytí a vytvoření mnoha „malých“ relací. Klíč schématu je **AB**.
 $RO_1 = (R_1(ABC), R_2(CD), R_3(BE), R_4(EF), R_5(CG))$
2. Spojíme relace, které vznikly z FZ se stejnými levými stranami.
 Žádné takové nejsou, takže $RO_2 = RO_1$
3. Spojíme relace s ekvivalentními klíči
 Opět žádné nejsou, takže $RO_3 = RO_2$
4. Pokud zbyl nějaký nezpracovaný atribut, přidáme jej do kterékoli z relací.
 Nezbyl, takže nic, $RO_4 = RO_3$
5. Není-li klíč obsažen v žádné z relací, pak vytvoříme novou relaci obsahující atributy klíče.
 Klíč **AB** je obsažen v R_1 , takže $RO_5 = RO_4$ a máme výsledek
 $R_1(ABC), R_2(CD), R_3(BE), R_4(EF), R_5(CG)$

- **Redundance**
Nadbytečnost dat, stejná data jsou uchována vícekrát
- **Konsistence**
Když už k redundanci dojde, musíme zajistit konzistenci dat
- **Integrita**
Znamená soulad se zobrazovanou realitou
- **1. Normální forma**
Všechny atributy jsou atomické
- **2. Normální forma**
Každý neklíčový atribut je úplně závislý na klíči
- **3. Normální forma**
Neexistují netriviální závislosti mezi neklíčovými atributy
- **BCNF**
Pro každou FZ platí, že levá strana je klíč nebo nadklíč
- **Dekompozice**
Musí obsahovat všechny atributy, nesmí dojít ke ztrátě informace nebo funkční závislosti
- **Algoritmus dekompozice do BCNF**
- **Algoritmus dekompozice do 3.NF**

www.dbedu.cs.vsb.cz

- Přihlášení přes **jednotný login a heslo**
- Vpravo sloupec -> *České kurzy* -> *UDBS*

