Databáze v geoinformatice II.

Mgr. Michal Jakl

ZS 2022

Obsah lekce

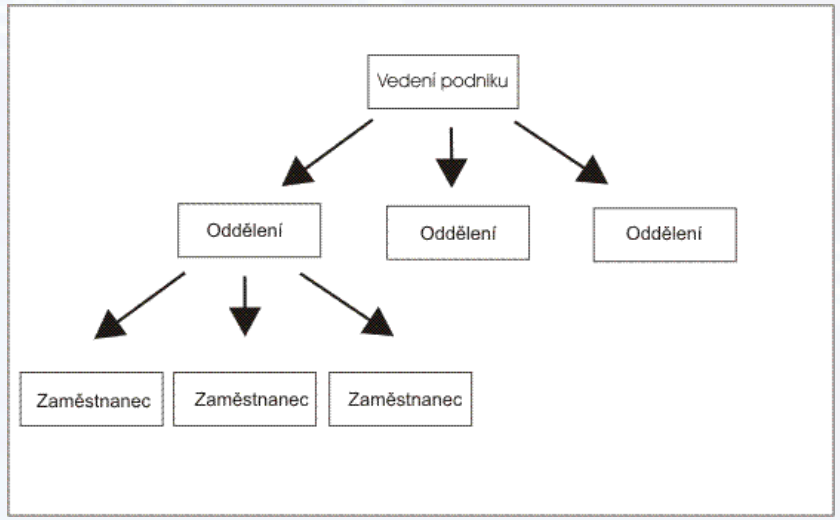
1. Datové modely
2. Relační model
3. Relační algebra
4. Základní datové typy

Praktické cvičení:

Procvičení relačních operací

1. Datové modely – hierarchický model

- Hierarchický model je založen na principu modelování hierarchie mezi entitami se vztahy podřízenosti, nadřízenosti a dědičnosti.
- Jedná se o nejstarší z databázových model
- Vznikl v 60. letech 20. století.
- Lze jej využít pro popis všech informačních systémů, nebo jejich částí, kde lze nalézt vztahy podřízenosti a nadřízenosti.
- Řada informačních systém ale tímto modelem popsat nejde. Nelze například pomocí tohoto modelu popsat všechny možné vazby (vztahy) mezi entitami.
- Příkladem informačního systému, který lze popsat pomocí hierarchického modelu, je informační systém pro rodokmen.
- Vědci i odborníci si nedostatek tohoto modelu uvědomovali a tak nový databázový model na sebe nenechal dlouho čekat…

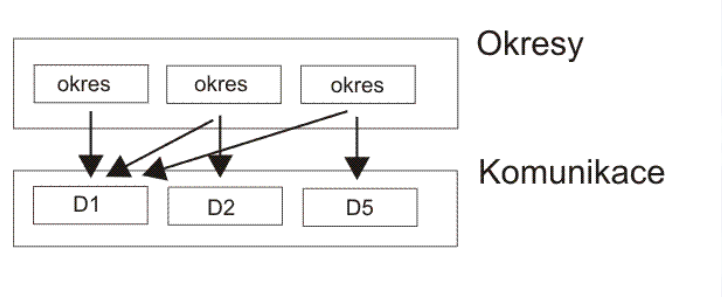


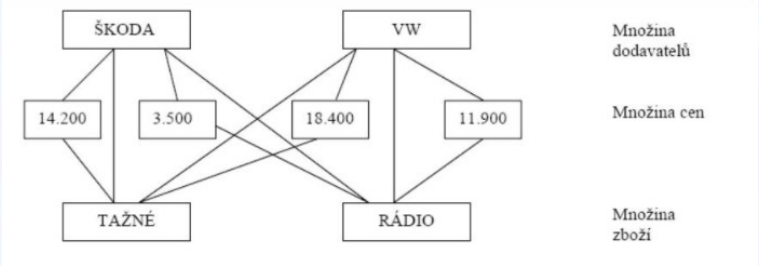
Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

1. Datové modely – síťový model

- Jeho počátky se datují do 70. let 20. století
- Jedná se o model, který jednotlivé entity organizuje do grafu. Graf je matematická struktura, která je tvořena uzly a hranami.
- Uzly jsou v tomto případě entity informačního systému a hrany mezi uzly jsou dokumentované vazby mezi entitami.
- Prvky v databázi jsou propojeny horizontálně i vertikálně. Je tedy výhodnější než hierarchická databáze a vyznačuje se menší redundancí.
- Tento model je tedy už výrazně obecnější, než model hierarchický, neboť je schopen dokumentovat i jiné vazby, než jen hierarchické vazby (jedna entita je nadřazena druhé entitě). Nicméně stále tento model není dostatečný, neboť nedokáže popsat všechny vazby, zejména pak vazby, kde jednotlivé entity vystupují jako násobné.
- Proto se od tohoto modelu postupně upouští a vzniká poslední, relační model, který je hojně využíván dodnes.

- Pomocí síťového modelu lze popsat některé geografické informační systémy (GIS). Jde o systémy, kde se zpracovávají geografické údaje (o územích, parcelách, rozvod inženýrských sítí, apod.). Konkrétním příkladem může být databáze nejkratších tras mezi jednotlivými městy v České republice.

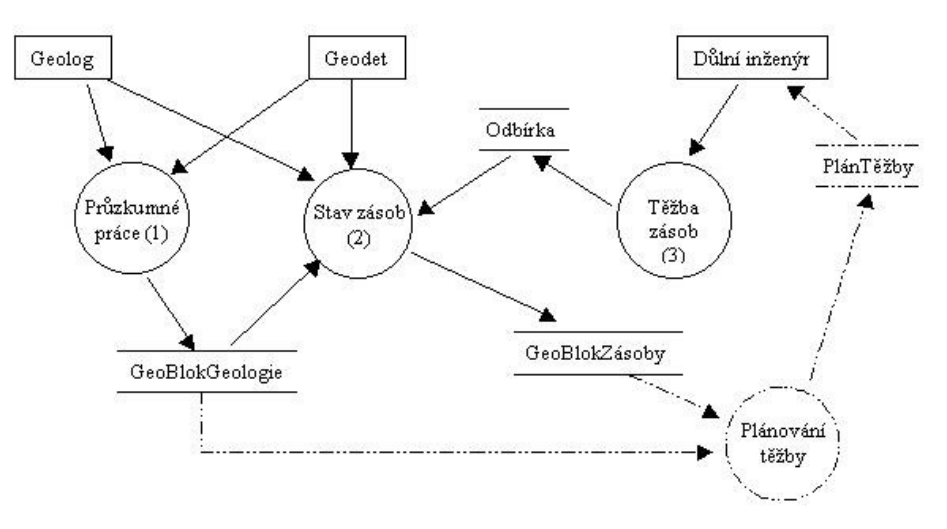




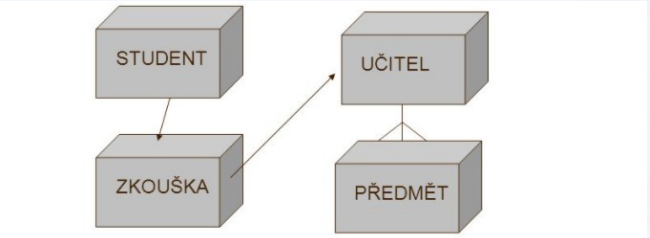
Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

1. Datové modely – objektový model

- Základním prvkem objektového modelu je objekt.
- Každý objekt má své vlastnosti a metody, kterými přistupuje a manipuluje s hodnotami svých vlastností.
- Informační systém je pak popsán jako soubor objektů se svými vlastnostmi, metodami a jak se objekty k sobě vzájemně chovají.
- Tento model je robustnější, než model relační, ale na druhou stranu již poměrně složitý, což ho staví poněkud do nevýhodné pozice.
- Přílišná složitost tohoto modelu způsobila, že neobstál v boji proti jednoduššímu relačnímu modelu.
- Objektového modelu se tedy příliš neužívá.



- Objektově-relační model je vzniklým kompromisem
- Tento model rozšiřuje stávající relační model o dobré a zajímavé vlastnosti modelu objektového. Zůstává mu tedy relativní jednoduchost z relačního modelu a navíc získává „to dobré“ z modelu objektového.



OBJEKTY STEJNÉHO TYPU TVOŘÍ TŘÍDU OBJEKTŮ NAPŘÍKLAD TŘÍDA „ZKOUŠEK“. KONKRÉTNÍ ZÁZNAM O ZKOUŠCE NAZÝVÁME INSTANCE OBJEKTU. KAŽDÝ OBJEKT V DATABÁZI MÁ PŘIDĚLEN SVŮJ UNIKÁTNÍ IDENTIFIKÁTOR – OID, POMOCÍ KTERÉHO MŮŽEME MEZI OBJEKTY VÉST PŘÍMÉ VAZBY OBDOBNĚ JAKO V SÍŤOVÉM MODELU. KROMĚ TOHO V OBJEKTOVÉM MODELU MOHOU EXISTOVAT I RELAČNÍ VAZBY.

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

2. Relační model – základní informace

- Dr. E. F. Codd, 1969
- Základem pojem - relace
  - Každá entita informačního systému je popsána relací a rovněž každá vazba mezi entitami je opět popsána relací. Soubor relací pak tvoří tzv. relační schéma
- Důraz na integritu
- Snadná pochopitelnost
- Lidská přirozenost
- Do souvislosti se vznikem relačního modelu je dáván také vznik jazyka SQL (Structured Query Language), který se stal standardem pro relační databáze

entita – cokoliv, o čem potřebujeme v systému uchovávat informace

atributy – určité sledované skutečnosti týkající se dané entity

domény – popisují typ dat, obor hodnot = spojení datového typu a validačního pravidla

vztahy – jisté asociace mezi entitami, nemusí mezi určitými entitami existovat

2. Relační model - relace

Relační schéma R je výraz tvaru R(A, f), kde R je jméno schématu, A = {A1, A2,..., An} je konečná množina jmen atributů, f je zobrazení přiřazující každému jménu atributu Ai neprázdnou množinu (obor hodnot atributu), kterou nazýváme doménou atributu Di, tedy f(Ai) = Di.

Relace R s relačním schématem R je konečná podmnožina kartézského součinu domén Di, příslušejících jednotlivým atributům Ai, tedy R ⊂ D1 x D2 x ...x Dn. Číslo n nazýváme stupněm relace, o relaci R říkáme, že je typu R

Relační schéma definuje nám již známý pojem typ entity (zobrazený jako název a hlavička datové tabulky), relace pak odpovídá množině entit (zobrazené jako obsah tabulky, množina jejích řádků).

Již víme, že relaci je výhodné znázorňovat jako dvourozměrnou tabulku, kde každý řádek odpovídá jedné entitě, každý sloupec jednomu atributu. Jména atributů musí být v rámci relace navzájem různá. Často se tedy v relačním modelu místo pojmu relace používá pojem tabulka.

Zdroj: http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf

2. Relační model - relace

Jméno relačního schématu: **Učitel**

Jména atributů: `ČU,jméno,funkce,plat`

Domény: D1 je množina řetězců tvaru Un, kde n je přirozené číslo; prvky množiny jsou osobní čísla učitelů;
D2 je množina možných jmen učitelů;
D3 množina textů označujících funkční zařazení učitelů v zaměstnání;
D4 množina nezáporných celých čísel<10000 *(IO definované jako interval)*

Zobrazení: `f(ČU)=D1, f(jméno)=D2, ...`

Relační schéma: **Učitel**`(ČU,jméno,funkce,plat)`

Kartézský součin domén: D1 x D2 x D3 x D4 *je množina* ***všech možných*** *čtveřic takových, že první prvek čtveřice je hodnota z* D1, …, *poslední je hodnota z* D4.

Relace:

```
Učitel={(U12,Čáp Ladislav,docent,24000),
        (U27,Holub Stanislav,asistent,23000),
        (U39,Kos Zbyněk,asistent,13000),
        (U43,Orel David,docent,24000), …}
```

přičemž do relace patří jen ty čtveřice, které odpovídají skutečným hodnotám 4 atributů odpovídajících evidovaným učitelům.

Do relace nepatří např. čtveřice `(U27,Holub Stanislav,docent,3000)`, *neboť nikdo takový neexistuje, případně nepatří k našim evidovaným učitelům.*

Zobrazení relace tabulkou

Učitel

ČU	jméno	fce	plat
U12	Čáp Ladislav	docent	24000
U27	Holub Stanislav	asistent	23000
U39	Kos Zbyněk	asistent	13000
U43	Orel David	docent	24000
…			

Pro schéma **Učitel** (ČU, jméno, fce, plat) *je relace* Učitel ⊂ ČU x jméno x fce x plat

Kartézský součin obsahuje všechny kombinace hodnot všech domén. Z nich jen některé patří do relace = podmnožiny kartézského součinu. Jsou to ty prvky (řádky), které odrážejí evidovanou skutečnost formulovanou integritním omezením o prvcích relace. To znamená ty, které přísluší skutečným učitelům naší evidence (Černohorský je docentem s osobním číslem U66 a platem 20000, Vondrák profesorem … v evidenci školy).

Učitel

ČU	jméno	fce	plat
U1	Anděl Jiří	asistent	10000
U1	Anděl Jiří	asistent	20000
…			
U128	Anděl Jiří	profesor	30000
…			
U66	Černohorský Jindřich	asistent	10000
U66	Černohorský Jindřich	asistent	20000
U66	Černohorský Jindřich	asistent	30000
U66	Černohorský Jindřich	docent	20000
…			
U888	Vondrák Ivo	asistent	10000
…			
U888	Vondrák Ivo	profesor	30000

Zdroj: http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf

2. Relační model – podmínky relačnosti

- Logická úroveň jedním způsobem - tabulky
- Zaručený přístup (tabulka, sloupec, primární klíč)
- Podpora neznámé hodnoty (NULL)
- Přístup shodný data a metadata
- Přívětivý a komplexní datový umožňující
- Určení, zda je pohled i pro zápis
  - = všechny aktualizovatelné pohledy je možno aktualizovat systémově
- Množinové operace s celými tabulkami
- Fyzická datová nezávislost
  - = data jsou nezávislá na konkrétním fyzickém uložení
- Logická datová nezávislost
  - = můžeme provádět změny v logické struktuře databáze, aniž by to uživatel poznal
- Nezávislost na integritních omezeních
  - = integritní omezení je definováno v datovém jazyku, nikoliv v aplikaci
- Podpora distribuovaného zpracování
  - = databázový jazyk musí být schopen manipulovat s daty umístěnými na jiném počítačovém systému
- Procedurální jazyk nemůže být mocnější
  - = žádný z jazyků nesmí mít možnost manipulovat s daty v rozporu s integritními omezeními

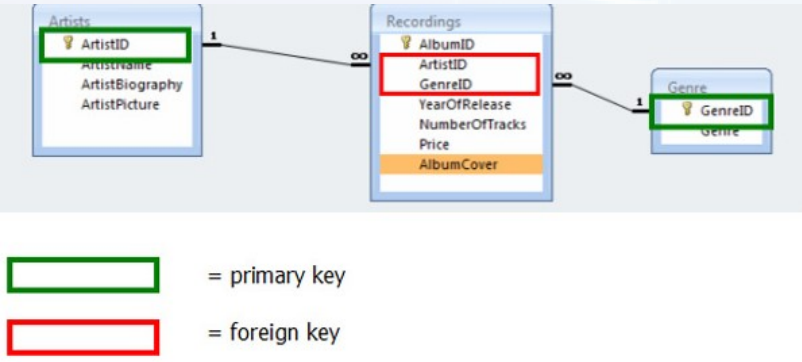
- ✓ definici dat,
- ✓ definici pohledu,
- ✓ manipulaci s daty,
- ✓ omezení integrity,
- ✓ autorizaci,
- ✓ vymezení transakce.

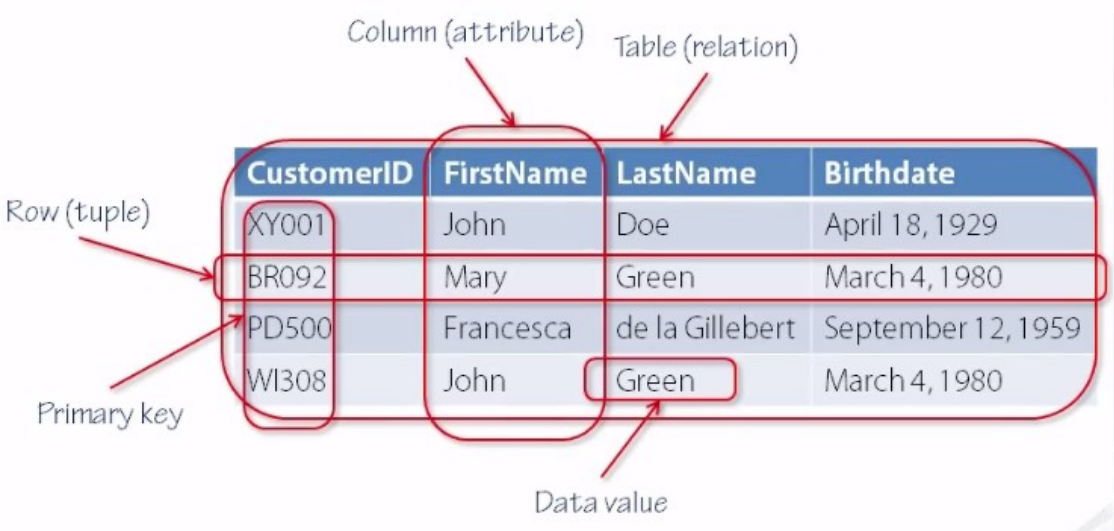
Podmínky minimální relačnosti

- veškerá data v databázi jsou uložena v databázových tabulkách
- uživatele nezajímá fyzická reprezentace dat
- je k dispozici neprocedurální dotazovací jazyk podporující základní operace z RA

2. Relační model - klíče

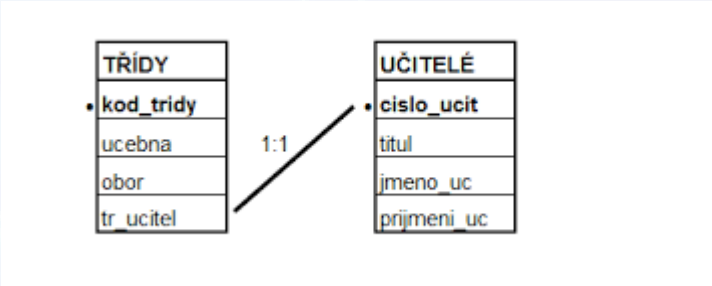
- Každá n-tice je jednoznačně určena hodnotami atributů tvořících klíč
- Může existovat více klíčů
- Minimalita – žádný atribut nelze odebrat
  - Primární klíč
  - MĚSTO (PSČ, název, kraj, stát, kontinent)





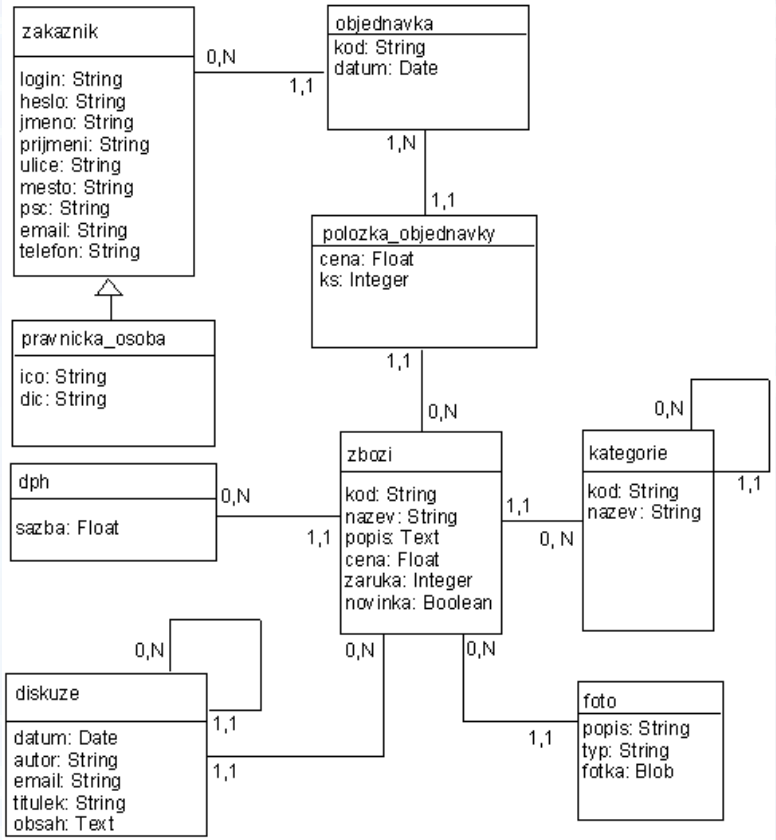
Zdroj: http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf

2. Relační model - vazby



- 1:1
- 1:n
- n:m
- povinný
- nepovinný

- Charakteristickou vlastností relačního modelu je realizace vazeb opět pomocí relací ~ tabulek
- Relační model obecně realizuje vazbu mezi relacemi pomocí relace. V ní každou entitu do vazby vstupující zastupuje její primární klíč.
- V některých případech (vazby 1:1, 1:M) však není nutné použít samostatnou vazební tabulku, ale stačí doplnění tabulky na straně M o další atribut - cizí klíč.
    - Celkově můžeme konstatovat, že vazba mezi relacemi se obecně vždy dá realizovat pomocí vazební tabulky. V některých případech je možné vazební tabulku sloučit s jednou z tabulek do vazby vstupujících a spojit dva totožné sloupce. Nově připojené sloupce k původní tabulce nazýváme cizí klíče.
- Velká výhoda relačního modelu je však také v tom, že i předem nepředpokládané vazby můžeme realizovat bez zásahu do původní struktury relací.
    - Pokud se až později ukáže potřeba evidence nového, nepředpokládaného typu vazby mezi existujícími entitami, stačí definovat novou vazební relaci. Jejími atributy budou primární klíče provázaných relací, případně další atributy vazby



Zdroj: http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf

2. Relační model - normální formy

Normalizace databáze je v informatice označení postupu, kdy je struktura dat v relační databázi přeorganizována tak, aby využívala výhody relačního modelu dat. Normalizace databáze umožňuje data efektivněji ukládat, prohledávat, třídit i zpracovávat. Při normalizaci jsou v databázi měněny atributy (sloupce) jednotlivých tabulek a zaváděny mezi nimi výhodné vztahy, je omezována redundance uložených dat (vět) a je brán ohled na řešení problému s případnou nekonzistencí dat.

Autorem termínu normalizace databáze je britsko-americký matematik a informatik Edgar. F. Codd.

- V relačním modelu jsou data uložena v tabulkách, na které má jisté požadavky.
- Při splnění požadavků je tabulka označována jako normalizovaná.
- Pokud nejsou tyto požadavky splněny, jsou označovány jako nenormalizované (= jsou v 0. NF) a proces jejich převodu na tabulky se označuje jako normalizace.
- Při tomto procesu dochází k odstraňování nedostatků tabulek jako je redundance nebo možnost vzniku aktualizační anomálie, tj. nechtěného vedlejšího efektu operace nad databází, při kterém dojde ke ztrátě nebo nekonzistenci dat.
- Postup normalizace je rozdělen do několika kroků a po dokončení každého z nich se tabulka nachází v určité normální formě.
- V praxi se většinou normalizuje do třetí normální formy, vyšší normální formy je vcelku obtížné porušit a vyžadují relativně velké znalosti, stejně jako návrh databází takové velikosti, kde je možné je porušit.
- Přestože existuje řada normálních forem, v praxi se za normalizovanou databázi považuje taková, která splňuje alespoň první tři normální formy.
- Důležité také je, že každá z normálních forem obsahuje a vyžaduje splnění všech pravidel obsažených ve formách předchozích.

Zdroj: https://cs.wikipedia.org/wiki/Normalizace_databáze

2. Relační model - 1. NF

Tabulka splňuje 1NF právě když všechny její atributy (sloupce) jsou atomické hodnoty.

Tabulka: ZAMĚSTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ
1	Jan	Novák	15.10.1975
2	Petr	Nový	1.4.1978
3	Jan	Nováček	6.9.1965
4	David	Vokurka	5.12.1973

Tabulka: ZAMĚSTNANEC_ADRESA

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ADRESA
1	Jan	Novák	15.10.1975	Václavská 19, Brno, 60200
2	Petr	Nový	1.4.1978	Novákova 114, Brno, 62100
3	Jan	Nováček	6.9.1965	Hybešova 10, Brno, 60100
4	David	Vokurka	5.12.1973	Kartouzská 7, Brno, 61200

Tabulka: ZAMĚSTNANEC_ADRESA2

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ULICE	CISLO	MESTO	PSC
1	Jan	Novák	15.10.1975	Václavská	19	Brno	60200
2	Petr	Nový	1.4.1978	Novákova	114	Brno	62100
3	Jan	Nováček	6.9.1965	Hybešova	10	Brno	60100
4	David	Vokurka	5.12.1973	Kartouzská	7	Brno	61200

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

2. Relační model - 2. NF

Tabulka splňuje 2NF právě když:

- splňuje 1NF
- každý atribut, který není primárním klíčem, je na primárním klíči **úplně** závislý

Tabulka: HRA_SACH

CISLO_HRAC1	CISLO_HRAC2	DATUM	CAS	VYSLEDEK	JM_HRAC1	JM_HRAC2
10	12	12.1.2006	12:00	1:0	Jan	Pavel
11	12	14.1.2006	13:05	0:1	Karel	Pavel
12	14	14.1.2006	15:13	pat	Pavel	Petr

Poznámka:
Je-li v tabulce označen jenom jeden sloupec jako primární klíč a jsou-li hodnoty ve všech sloupcích atomické, pak je 2NF splněna triviálně.

Tabulka: HRA_SACHY

CISLO_HRAC1	CISLO_HRAC2	DATUM	CAS	VYSLEDEK
10	12	12.1.2006	12:00	1:0
11	12	14.1.2006	13:05	0:1
12	14	14.1.2006	15:13	pat

Tabulka: HRAC

CISLO	JM_HRAC
10	Jan
11	Karel
12	Pavel
14	Petr

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

2. Relační model - 3. NF

Tabulka splňuje 3NF, právě když:

- splňuje 2NF
- žádný atribut, který není primárním klíčem, není tranzitivně závislý na žádném klíči.

Když A->B a zároveň B->C pak plyne, že A->C.

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	CISLO_PRAC	NAZEV_PRAC
1	Jan	Novák	15.10.1975	10	studovna
2	Petr	Nový	1.4.1978	15	vrátnice
3	Jan	Nováček	6.9.1965	10	studovna
4	David	Vokurka	5.12.1973	10	studovna
5	Vít	Mrkvička	30.1.1970	12	ředitelna

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	CISLO_PRAC
1	Jan	Novák	15.10.1975	10
2	Petr	Nový	1.4.1978	15
3	Jan	Nováček	6.9.1965	10
4	David	Vokurka	5.12.1973	10
5	Vít	Mrkvička	30.1.1970	12

Tabulka: PRACOVISTE

CISLO	NAZEV_PRAC
10	studovna
15	vrátnice
12	ředitelna

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

2. Relační model - relace - vlastnosti

Z definice relace vyplývají tyto jejich tabulkové vlastnosti

- Homogenita sloupců (hodnoty odpovídají příslušné doméně)
- Každý údaj (hodnota atributu ve sloupci) je atomickou položkou
- Na pořadí řádků nezáleží (relace je množina)
- Na pořadí sloupců nezáleží (pojmenované atributy také tvoří množinu)
- Každý řádek tabulky je jednoznačně identifikovatelný hodnotami jednoho nebo několika atributů (primárního klíče)
- Žádné dva řádky nejsou shodné
- Atributy jsou reprezentovány záhlavím

Zdroj: http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf

3. Relační algebra

Relační algebra je matematický nástroj, který pracuje a manipuluje s relacemi. Vstupem do operací relační algebry jsou relace a výstupem jsou opět relace. Relace jsou v podstatě množiny (množina n-tic splňující danou relaci). Mžeme tedy říci, že do operací vstupují množiny a výstupem jsou opět množiny.

- Relační algebra jako vyspělý prostředek pro práci s relacemi
- Problematika získávání informací z tabulek
- Pochopení pomůže snáze vstřebat standard pro relační databáze, jazyk SQL
- Dále uvedené operace slouží k formulaci dotazů

Dotaz je formule specifikující, jaké informace chceme z databáze získat. Výsledek dané operace je odpovědí na náš dotaz.

3. Relační algebra - operace

- Množinové operace:
  - Sjednocení (union)
  - Rozdíl
  - Průnik
  - Kartézský součin
- Databázové operace
  - Projekce
  - Restrikce (selekce)
  - Spojení (join)
    - Left, right, full join

Jsou dány relace R a S.

Množinové operace

sjednocení relací téhož stupně $R \cup S = \{x \mid x \in R \vee x \in S\}$

průnik relací $R \cap S = \{x \mid x \in R \wedge x \in S\}$

rozdíl relací $R - S = \{x \mid x \in R \wedge x \notin S\}$

kartézský součin relace R stupně m a relace S stupně n

$$R \times S = \{rs \mid r \in R \wedge s \in S\}, \text{ kde } rs = \{r_1,\ldots,r_m,s_1,\ldots s_n\}$$

Další operace relační

projekce (výběr sloupců) relace R na atributy $A=\{A_{i1},\ldots A_{im}\}$, $1 \le i_m \le n$, $i_j \ne i_k$ pro $j \ne k$

$$R[A] = \{r[A] \mid r \in R\}, \text{ kde } R[A] = (r_{i1}, \ldots, r_{im}) \text{ pro } r \in R$$

selekce (výběr řádků) z relace R podle podmínky P je

$$R(P) = \{r \mid r \in R \wedge P(r)\}$$

spojení relací R s atributy A a relace S s atributy B dle relačního operátoru

$\Theta = \{<,=,>,<=,>=,<>\}$ v atributu A_i relace R a v atributu B_j relace S je

$$R[A\Theta B]S = \{rs \mid r \in R \wedge s \in S \wedge r[A_i]\ \Theta\ s[B_j]\}$$

Θ-spojení lze definovat jako kartézský součin relací R a S následovaný selekcí s podmínkou AΘB.

Nejčastěji se používá spojení relací podle operátoru "="; v tom případě by ve výsledné relaci byly dva sloupce shodné, proto se zavádí operace

$$R\ [A*B]\ S = \{R\ [A=B]\ S\},$$

která automaticky jeden ze shodných sloupců vypouští.

Příklady uvedené dále jsou několik let staré, proto například data nemusí úplně odpovídat. Na logiku dotazů to však nemá žádný vliv ☺

Zdroj: http://wiki.cs.vsb.cz/images/2/2a/Inz2.pdf

3. Relační algebra - projekce

Projekce **P** nad relací **R** s množinou atributů **A** je relace **R*** s množinou atributů **B**, kde **B** je podmnožinou původní množiny atributů **A**. Jako příklad si vezměme relaci **ZAMESTNANEC** s atributy **CISLO**, **JMENO**, **PRIJMENI**, **DAT_NAROZ**, **SMLOUVA_OD**. Množina původních atributů **A = {CISLO, JMENO, PRIJMENI, DAT_NAROZ, SMLOUVA_OD}**. Nechť projekce **P** je zaměřena pouze na jména a příjmení zaměstnanců. Tedy množina **B = {JMENO, PRIJMENI}**. Výsledná relace **ZAMESTNANEC*** bude obsahovat pouze 2 atributy **JMENO** a **PRIJMENI**.

Projekci zapisujeme do hranatých závorek **[,]**, do kterých uvedeme jména sloupců, na které chceme projekci zrealizovat. Původní relace (tabulka) **ZAMESTNANEC** vypadala takto:

U projekce je třeba ještě zmínit, že pokud by vyprojektováním určitých sloupců ve výsledné relaci došlo k duplicitě některých řádků, pak tato duplicita je automaticky odstraněna. Vezměme si náš příklad relace **ZAMESTNANEC** a nechť v naší firmě pracují dva lidé jménem Jan Novák, s tím rozdílem, že první se narodil 15.10.1975 a druhý 3.2.1960. Projekcí na sloupce **JMENO** a **PRIJMENI** získáme ve výsledné relaci pouze jeden záznam Jan Novák.

Tabulka: ZAMESTNANEC

CISLO	JMENO	PRIJMENI	DAT_NAROZ	SMLOUVA_OD
1	Jan	Novák	15.10.1975	1.1.2000
2	Petr	Nový	1.4.1978	12.5.1999
3	Jan	Nováček	6.9.1965	7.7.1998
4	David	Vokurka	5.12.1973	1.10.2002

Tabulka: ZAMESTNANEC*

JMENO	PRIJMENI
Jan	Novák
Petr	Nový
Jan	Nováček
David	Vokurka

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra - projekce

Mějme relaci, která je popsána tabulkou **BYT**. Ta obsahuje sloupce:

- **CISLO** – evidenční číslo bytu
- **ADR_ULICE** – adresa: ulice, kde se byt nachází
- **ADR_CISLO** – číslo domu
- **ADR_MESTO** – město, ve kterém se byt nachází
- **ADR_PSC** – poštovní směrovací číslo
- **PODLAZI** – číslo podlaží
- **BALKON** – A/N, zda-li má byt balkon
- **SKLEP** – A/N, zda-li má byt sklep

Nechť v tabulce **BYT** je uloženo celkem 200 záznamů. To znamená, že naši relaci **BYT** splňuje celkem 200 různých bytů.

Zodpovězme nyní následující dotazy:

- *Čísla všech evidovaných bytů:* [redacted]
- *Seznam všech adres (ulice a město) kde jsou byty evidovány:* [redacted]
- *Čísla všech podlaží, na kterých se evidované byty nachází:* [redacted]
- *Seznam všech měst, ve kterých se nachází evidované byty:* [redacted]

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra - projekce

Mějme relaci, která je popsána tabulkou **BYT**. Ta obsahuje sloupce:

- **CISLO** – evidenční číslo bytu
- **ADR_ULICE** – adresa: ulice, kde se byt nachází
- **ADR_CISLO** – číslo domu
- **ADR_MESTO** – město, ve kterém se byt nachází
- **ADR_PSC** – poštovní směrovací číslo
- **PODLAZI** – číslo podlaží
- **BALKON** – A/N, zda-li má byt balkon
- **SKLEP** – A/N, zda-li má byt sklep

Nechť v tabulce **BYT** je uloženo celkem 200 záznamů. To znamená, že naši relaci **BYT** splňuje celkem 200 různých bytů.

Zodpovězme nyní následující dotazy:

- *Čísla všech evidovaných bytů*: **BYT[CISLO]**
- *Seznam všech adres (ulice a město) kde jsou byty evidovány*: **BYT[ADR_ULICE,ADR_MESTO]**
- *Čísla všech podlaží, na kterých se evidované byty nachází*: **BYT[PODLAZI]**
- *Seznam všech měst, ve kterých se nachází evidované byty*: **BYT[ADR_MESTO]**

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra - restrikce

Tak jako nám projekce vybírá do výsledku určité sloupce tabulky, tak restrikce vybírá do výsledku určité řádky tabulky, které splňují zadanou podmínku.

Restrikce podle logické podmínky **ϕ** nad relací **R** je relace **R***, ve které jsou ponechány jen ty n-tice z původní relace, jejichž hodnoty splňují logickou podmínku **ϕ**. Logická podmínka **ϕ** je zadána booleovským výrazem (pomocí logických spojek **and**, **or** a **not**) složeného z atomických formulí tvaru: $\mathbf{t_1 \theta\, t_2}$, kde $\boldsymbol{\theta} = \{<, >, =, \neq, \leq, \geq\}$. Přičemž $\mathbf{t_i}$ je buď nějaká konstanta, nebo název existujícího atributu. Logickou podmínku restrikce zapisujeme do kulatých závorek **(,)**.

Výše uvedený teoretický zápis je obecným zápisem například pro jednoduché podmínky **JMENO = Jan**, **PRIJMENI = Novák**, **CISLO = 2**, **PODLAZI ≠ 1**, apod. Jednotlivé podmínky můžeme spolu kombinovat pomocí závorek a logických spojek, např: **JMENO = Jan and PRIJMENI = Novak**.

Tabulka: ZAMESTNANEC

CISLO	JMENO	PRIJMENI	DAT_NAROZ	PLAT	SMLOUVA_OD
1	Jan	Novák	15.10.1975	15000	1.1.2000
2	Petr	Nový	1.4.1978	21500	12.5.1999
3	Jan	Nováček	6.9.1965	17500	7.7.1998
4	David	Vokurka	5.12.1973	28000	1.10.2002

Zkusme zformulovat několik jednoduchých dotazů, na kterých je dobře vidět formulování logické podmínky pro restrikci. U každého příkladu je rovněž uvedeno, kolik záznamů z tabulky (nebo-li kolik n-tic z relace) danou restrikci splňuje – snadno se ověří.

- *Všichni zaměstnanci, kteří pracují ve firmě déle než 3 roky*:
- *Všichni zaměstnanci, kterým je více než 30 let*:
- *Všichni zaměstnanci, jejichž křestní jméno je Jan*:
- *Všichni zaměstnanci, kteří mají plat vyšší než 25000 Kč*:
- *Všichni zaměstnanci, kteří nejsou ve zkušební době a pobírají plat více jak 20000 Kč*:
- *Všichni zaměstnanci, kteří jsou starší 30ti let a pobírají plat maximálně 18000 Kč*:

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra - restrikce

Tak jako nám projekce vybírá do výsledku určité sloupce tabulky, tak restrikce vybírá do výsledku určité řádky tabulky, které splňují zadanou podmínku.

Restrikce podle logické podmínky **ϕ** nad relací **R** je relace **R***, ve které jsou ponechány jen ty n-tice z původní relace, jejichž hodnoty splňují logickou podmínku **ϕ**. Logická podmínka **ϕ** je zadána booleovským výrazem (pomocí logických spojek **and**, **or** a **not**) složeného z atomických formulí tvaru: $t_1 \theta t_2$, kde $\theta = \{<, >, =, \neq, \leq, \geq\}$. Přičemž t_i je buď nějaká konstanta, nebo název existujícího atributu. Logickou podmínku restrikce zapisujeme do kulatých závorek `(, )`.

Výše uvedený teoretický zápis je obecným zápisem například pro jednoduché podmínky **JMENO = Jan**, **PRIJMENI = Novák**, **CISLO = 2**, **PODLAZI ≠ 1**, apod. Jednotlivé podmínky můžeme spolu kombinovat pomocí závorek a logických spojek, např: **JMENO = Jan and PRIJMENI = Novak**.

Tabulka: ZAMESTNANEC

CISLO	JMENO	PRIJMENI	DAT_NAROZ	PLAT	SMLOUVA_OD
1	Jan	Novák	15.10.1975	15000	1.1.2000
2	Petr	Nový	1.4.1978	21500	12.5.1999
3	Jan	Nováček	6.9.1965	17500	7.7.1998
4	David	Vokurka	5.12.1973	28000	1.10.2002

Zkusme zformulovat několik jednoduchých dotazů, na kterých je dobře vidět formulování logické podmínky pro restrikci. U každého příkladu je rovněž uvedeno, kolik záznamů z tabulky (nebo-li kolik n-tic z relace) danou restrikci splňuje – snadno se ověří.

- *Všichni zaměstnanci, kteří pracují ve firmě déle než 3 roky*:
  `ZAMESTNANEC(SMLOUVA_OD <= 1.1.2003)` – splňují 4 záznamy
- *Všichni zaměstnanci, kterým je více než 30 let*:
  `ZAMESTNANEC(DAT_NAROZ < 31.12.1975)` – splňují 2 záznamy
- *Všichni zaměstnanci, jejichž křestní jméno je Jan*:
  `ZAMESTNANEC(JMENO = Jan)` – splňují 2 záznamy
- *Všichni zaměstnanci, kteří mají plat vyšší než 25000 Kč*:
  `ZAMESTNANEC(PLAT > 25000)` – splňuje 1 záznam
- *Všichni zaměstnanci, kteří nejsou ve zkušební době a pobírají plat více jak 20000 Kč*:
  `ZAMESTNANEC(SMLOUVA_OD < 1.1.2006 and PLAT > 20000)` – splňují 2 záznamy
- *Všichni zaměstnanci, kteří jsou starší 30ti let a pobírají plat maximálně 18000 Kč*:
  `ZAMESTNANEC(DAT_NAROZ < 31.12.1975 and PLAT >= 18000)` – splňuje 1 záznam

Najdete někde chybu, které se autor dopustil?

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – restrikce + projekce

Již víme, že výsledkem operace restrikce na relaci je opět relace. Nic nám nebrání v tom, abychom na výsledek operace restrikce aplikovali další operaci a to projekci. Všimněte si, že v předchozích uvedených příkladech restrikce jsme vždy z tabulky získali celé záznamy (celé n-tice). Co kdybychom ale měli dotaz: *Jména a příjmení všech zaměstnanců (další jejich údaje nás nezajímají), kteří začali pro naši firmu pracovat před rokem 2001?* Takový dotaz bychom zkonstruovali ve dvou krocích:

- Nejprve provedeme restrikci pro „*začali pracovat před rokem 2001*“ – výsledná „mezi“-relace vypadá následovně:

Tabulka: ZAMESTNANEC'

CISLO	JMENO	PRIJMENI	DAT_NAROZ	PLAT	SMLOUVA_OD
1	Jan	Novák	15.10.1975	15000	1.1.2000
2	Petr	Nový	1.4.1978	21500	12.5.1999
3	Jan	Nováček	6.9.1965	17500	7.7.1998

- A po té provedeme projekci na „*Jména a příjmení*“, s výsledkem:

Tabulka: ZAMESTNANEC*

JMENO	PRIJMENI
Jan	Novák
Petr	Nový
Jan	Nováček

Uvedený dotaz v relační algebře zapíšeme následovně:

```
ZAMESTNANEC(SMLOUVA_OD<1.1.2001)[JMENO,PRIJMENI]
```

Formulujme další jednoduché dotazy s využitím projekce a restrikce:

- *Kdy se narodil Jan Novák?*
- *Jaký plat pobírá a odkdy pracuje ve firmě David Vokurka?*
- *Kdy nastoupil do firmy pan Nováček?*

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – restrikce + projekce

Již víme, že výsledkem operace restrikce na relaci je opět relace. Nic nám nebrání v tom, abychom na výsledek operace restrikce aplikovali další operaci a to projekci. Všimněte si, že v předchozích uvedených příkladech restrikce jsme vždy z tabulky získali celé záznamy (celé n-tice). Co kdybychom ale měli dotaz: *Jména a příjmení všech zaměstnanců (další jejich údaje nás nezajímají), kteří začali pro naši firmu pracovat před rokem 2001?* Takový dotaz bychom zkonstruovali ve dvou krocích:

- Nejprve provedeme restrikci pro „*začali pracovat před rokem 2001*“ – výsledná „mezi“-relace vypadá následovně:

Tabulka: ZAMESTNANEC'

CISLO	JMENO	PRIJMENI	DAT_NAROZ	PLAT	SMLOUVA_OD
1	Jan	Novák	15.10.1975	15000	1.1.2000
2	Petr	Nový	1.4.1978	21500	12.5.1999
3	Jan	Nováček	6.9.1965	17500	7.7.1998

- A po té provedeme projekci na „*Jména a příjmení*“, s výsledkem:

Tabulka: ZAMESTNANEC*

JMENO	PRIJMENI
Jan	Novák
Petr	Nový
Jan	Nováček

Uvedený dotaz v relační algebře zapíšeme následovně:

```
ZAMESTNANEC(SMLOUVA_OD<1.1.2001)[JMENO,PRIJMENI]
```

Formulujme další jednoduché dotazy s využitím projekce a restrikce:

- *Kdy se narodil Jan Novák?*
  `ZAMESTNANEC(JMENO=Jan and PRIJMENI=Novák)[DAT_NAROZ]`
- *Jaký plat pobírá a odkdy pracuje ve firmě David Vokurka?*
  `ZAMESTNANEC(JMENO=David and PRIJMENI=Vokurka)[PLAT, SMLOUVA_OD]`
- *Kdy nastoupil do firmy pan Nováček?*
  `ZAMESTNANEC(PRIJMENI=Novacek)[SMLOUVA_OD]`

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – přirozené spojení

- Z definice uvedených tabulek víme, že sloupec ID_FUNKCE v tabulce ZAMESTNANEC je cizím klíčem do tabulky FUNKCE na její sloupec ID. Tato vazba primární-cizí klíč je důležitá a právě skrz tuto vazbu se realizuje přirozené spojení dvou tabulek (nebo-li relací)

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ID_FUNKCE
1	Jan	Novák	15.10.1975	1
2	Petr	Nový	1.4.1978	2
3	Jan	Nováček	6.9.1965	3
4	David	Vokurka	5.12.1973	4

Tabulka: FUNKCE

ID	NAZEV	PLAT
1	uklízeč	15000
2	programátor	21500
3	manažer	17500
4	ředitel	28000

Přirozené spojení vezme dvě původní relace **R** nad atributy z množiny **A**, tj. **R(A)** a relace **S** nad atributy z množiny **B**, tj. **S(B)**, které do něj vstupují a vytvoří jednu (souhrnnou) relaci **R*S** a to takovým způsobem, že atributy této nové relace budou všechny atributy jak z původní množiny atributů **A** tak i z **B**, přičemž jednotlivé řádky (n-tice) budou „svázány“ tak, aby si jednotlivé n-tice z původních tabulek „odpovídaly“. Nebo-li budou k sobě navázány taková n-tice z první relace s takovou n-ticí z druhé relace, že hodnota příslušného primárního klíče v první relaci se rovná hodnotě příslušného cizího klíče ve druhé relaci. V našem uvedeném příkladě relací **ZAMESTNANEC** a **FUNKCE** bude výsledek přirozeného spojení vypadat následovně:

Tabulka: ZAMESTNANEC*FUNKCE

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ID_FUNKCE	ID	NAZEV	PLAT
1	Jan	Novák	15.10.1975	1	1	uklízeč	15000
2	Petr	Nový	1.4.1978	2	2	programátor	21500
3	Jan	Nováček	6.9.1965	3	3	manažer	17500
4	David	Vokurka	5.12.1973	4	4	ředitel	28000

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – přirozené spojení

- *Jakou funkci vykonává pan Nováček?*
- *Jaký plat pobírají lidé v naši firmě starší 30 let?*
- *Seznam jmen, příjmení a funkcí, které vykonávají všichni zaměstnanci ve firmě:*
- *Seznam příjmení, datumů narození a výše platů všech zaměstnanců ve firmě:*

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ID_FUNKCE
1	Jan	Novák	15.10.1975	1
2	Petr	Nový	1.4.1978	2
3	Jan	Nováček	6.9.1965	3
4	David	Vokurka	5.12.1973	4

Tabulka: FUNKCE

ID	NAZEV	PLAT
1	uklízeč	15000
2	programátor	21500
3	manažer	17500
4	ředitel	28000

- Seznam jmen, příjmení a funkcí všech zaměstnanců ve firmě, kteří mají plat vyšší než 20000 Kč:
- Seznam příjmení a platů všech uklízečů ve firmě:
- Seznam jmen, příjmení všech zaměstnanců, kterým není ještě 30 let nebo mají plat nižší než 18000 Kč:

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – přirozené spojení

- *Jakou funkci vykonává pan Nováček?*
  ZAMESTNANEC*FUNKCE(PRIJMENI=Nováček)[NAZEV]
- *Jaký plat pobírají lidé v naši firmě starší 30 let?*
  ZAMESTNANEC*FUNKCE(DAT_NAROZ<1.1.1976)[PLAT]
- *Seznam jmen, příjmení a funkcí, které vykonávají všichni zaměstnanci ve firmě:*
  ZAMESTNANEC*FUNKCE[JMENO,PRIJMENI,NAZEV]
- *Seznam příjmení, datumů narození a výše platů všech zaměstnanců ve firmě:*
  ZAMESTNANEC*FUNKCE[PRIJMENI,DAT_NAROZ,PLAT]

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ID_FUNKCE
1	Jan	Novák	15.10.1975	1
2	Petr	Nový	1.4.1978	2
3	Jan	Nováček	6.9.1965	3
4	David	Vokurka	5.12.1973	4

Tabulka: FUNKCE

ID	NAZEV	PLAT
1	uklízeč	15000
2	programátor	21500
3	manažer	17500
4	ředitel	28000

- Seznam jmen, příjmení a funkcí všech zaměstnanců ve firmě, kteří mají plat vyšší než 20000 Kč:
  ZAMESTNANEC*FUNKCE(PLAT>20000)[JMENO,PRIJMENI,NAZEV]
- Seznam příjmení a platů všech uklízečů ve firmě:
  ZAMESTNANEC*FUNKCE(NAZEV="uklízeč")[PRIJMENI,PLAT]
- Seznam jmen, příjmení všech zaměstnanců, kterým není ještě 30 let nebo mají plat nižší než 18000 Kč:
  ZAMESTNANEC*FUNKCE(DAT_NAROZ>1.1.1976 or PLAT<18000)[JMENO, PRIJMENI]

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – kartézský součin

- Do operace kartézský součin budou vstupovat opět dvě relace.
- Jediný rozdíl oproti jednoduchým množinám bude ten, že již relace samy o sobě obsahují jako prvky n-tice.
- Řekněme že relace R nad množinou atribut A obsahuje trojice (tj. 3 atributy) a relace S nad množinou atribut B obsahuje dvojice (2 atributy). Kartézský souičn bude v tomto případě obsahovat pětice (3+2).
- V řeči tabulek si lze kartézský součin představit jako „sloučení“ dvou tabulek.
- Výsledná tabulka bude obsahovat záznamy z obou původních tabulek, které vzniknou tak, že zkombinujeme jednotlivé záznamy z původních tabulek metodou „každý s každým“.
- Je jasné, že pokud jedna tabulka obsahuje 10 záznam a druhá 5 záznam, výsledná tabulka bude mít celkem 50 záznam (10x5 kombinací).

Kartézský součin dvou relací **R** a **S** se zapisuje jako **R×S**.

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – kartézský součin

Tabulka: USECKA_A

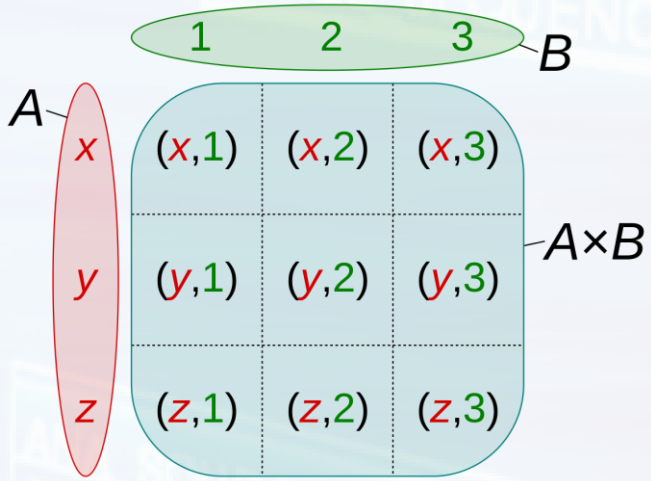
CISLO	DELKA
10	1
11	2
12	3

Tabulka: USECKA_B

CISLO	DELKA
13	1
14	2
15	3

Tabulka: USECKA_AxUSECKA_B

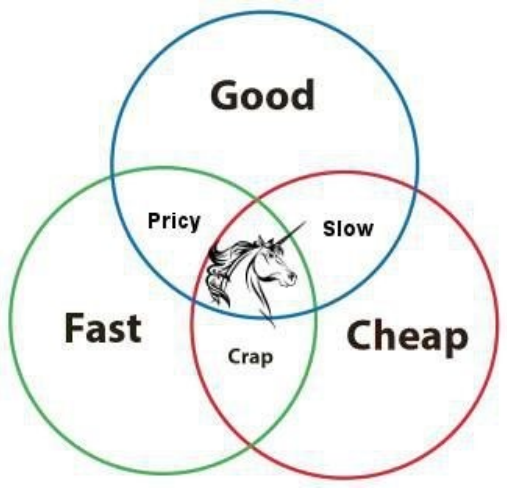
USECKA_A.CISLO	USECKA_A.DELKA	USECKA_B.CISLO	USECKA_B.DELKA
10	1	13	1
10	1	14	2
10	1	15	3
11	2	13	1
11	2	14	2
11	2	15	3
12	3	13	1
12	3	14	2
12	3	15	3



Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – sjednocení, průnik, rozdíl

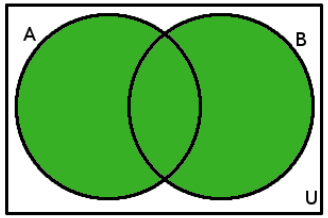
- Jednou z hlavních omezujících podmínek pro tyto množinové operace je tzv. „kompatibilita“ relací (tabulek). To znamená, že do množinové operace mohou vstupovat jen ty relace, které jsou kompatibilní. Kompatibilitou se rozumí:
  - Shodný počet sloupců (atribut)
  - Shodné typy dat v jednotlivých sloupcích



Nemůžeme míchat „hrušky s jabkama“

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – sjednocení



$A \cup B = \{x \in U: x \in A \vee x \in B\}$

Vezměme si náš příklad relací **ZAMESTNANEC** a **FUNKCE** a mějme následující dotaz: kteří lidé ve firmě (zajímají nás jejich jména a příjmení) pracují jako programátor nebo jako analytik? Takový dotaz se dá zapsat v relační algebře s využitím sjednocení.

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ID_FUNKCE
1	Jan	Novák	15.10.1975	1
2	Petr	Nový	1.4.1978	2
3	Jan	Nováček	6.9.1965	3
4	David	Vokurka	5.12.1973	4

Tabulka: FUNKCE

ID	NAZEV	PLAT
1	uklízeč	15000
2	programátor	21500
3	manažer	17500
4	ředitel	28000

Nejprve zkonstruujeme „pod“-dotaz, kde se zeptáme na všechny programátory a po té druhý „pod“-dotaz, kde totéž provedeme na analytiky. Nakonec provedeme projekci na **JMENO** a **PRIJMENI**. Výsledkem bude pak sjednocení lidí, kteří pracují buď jako programátor nebo jako analytik:

```
(ZAMESTNANEC*FUNKCE(NAZEV=programátor)∪
ZAMESTNANEC*FUNKCE(NAZEV=analytik))[JMENO,PRIJMENI]
```

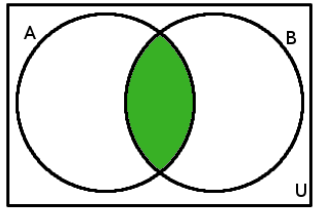
Samozřejmě dotaz „kteří lidé ve firmě pracují jako programátoři nebo jako analytici“ můžeme zapsat i bez použití operace sjednocení. Bude nám stačit přirozené spojení, restrikce a projekce:

```
ZAMESTNANEC*FUNKCE(NAZEV=programátor and NAZEV=analytik)[JMENO,
PRIJMENI]
```

Na tomto příkladu je vidět, že většinou existuje více způsobů jak daný dotaz zapsat. Každý způsob se bude zpracovávat pravděpodobně odlišně, nicméně výsledek musí vracet stejný. Jaký způsob zvolit je pak úlohou optimalizace dotazů pro konkrétní databázový systém, tím se zde zabývat nebudeme.

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – průnik



$A \cap B = \{x \in U;\ x \in A \wedge x \in B\}$

Do průniku vstupují dvě kompatibilní relace, výsledkem je pak jedna relace, která vznikne průnikem dvou původních relací. Zapisujeme **R∩S**.

Vezměme si příklad o autorství knih, který jsme použili v kapitole věnované vazbě M:N v tabulkách:

Tabulka: KNIHA

ID	NÁZEV	JAZYK	ID_AUTOR
11	Oko	český	1
12	Bludiště	německý	4
13	Domek	český	4
14	Sauna a bazén	anglický	3

Tabulka: AUTORSTVÍ

ID_KNIHA	ID_AUTOR
11	1
11	5
14	3
12	4
13	4

Tabulka: AUTOR

ID	JMÉNO	PŘÍJMENÍ	DATUM_NAROZENÍ
1	Jan	Novák	15.10.1975
2	Petr	Nový	1.4.1978
3	Jan	Nováček	6.9.1965
4	David	Vokurka	5.12.1973
5	Vít	Mrkvička	30.1.1970

Jaké knihy (zajímají nás názvy) napsali autoři Jan Novák a Vít Mrkvička jako spoluautoři?

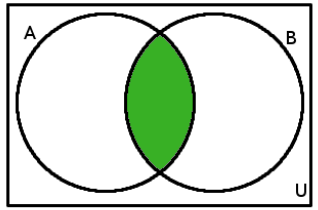
```
((KNIHA*AUTORSTVI)*AUTOR)((JMENO=Jan and PRIJMENI=Novák) and
(JMENO=Vít and PRIJMENI=Mrkvička))[NAZEV]
```

Projekci můžeme provádět buď až po skončení množinové operace (sjednocení, průnik, rozdíl) a nebo v každé „pod“-relaci vstupující do množinové operace. Výše uvedený dotaz s průnikem můžeme zapsat také takto:

Když se podíváme na obsah původních tabulek, zjistíme, že odpovědí na tento dotaz je kniha

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – průnik



$A \cap B = \{x \in U;\ x \in A \wedge x \in B\}$

Do průniku vstupují dvě kompatibilní relace, výsledkem je pak jedna relace, která vznikne průnikem dvou původních relací. Zapisujeme **R∩S**.

Vezměme si příklad o autorství knih, který jsme použili v kapitole věnované vazbě M:N v tabulkách:

Tabulka: KNIHA

ID	NÁZEV	JAZYK	ID_AUTOR
11	Oko	český	1
12	Bludiště	německý	4
13	Domek	český	4
14	Sauna a bazén	anglický	3

Tabulka: AUTORSTVÍ

ID_KNIHA	ID_AUTOR
11	1
11	5
14	3
12	4
13	4

Tabulka: AUTOR

ID	JMÉNO	PŘÍJMENÍ	DATUM_NAROZENÍ
1	Jan	Novák	15.10.1975
2	Petr	Nový	1.4.1978
3	Jan	Nováček	6.9.1965
4	David	Vokurka	5.12.1973
5	Vít	Mrkvička	30.1.1970

Jaké knihy (zajímají nás názvy) napsali autoři Jan Novák a Vít Mrkvička jako spoluautoři?

```
((KNIHA*AUTORSTVI)*AUTOR)((JMENO=Jan and PRIJMENI=Novák) and
(JMENO=Vít and PRIJMENI=Mrkvička))[NAZEV]
```

```
(((KNIHA*AUTORSTVI)*AUTOR)(JMENO=Jan and PRIJMENI=Novák)∩
((KNIHA*AUTORSTVI)*AUTOR)(JMENO=Vít and PRIJMENI=´Mrkvička))
[NAZEV]
```

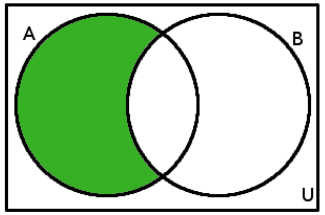
Projekci můžeme provádět buď až po skončení množinové operace (sjednocení, průnik, rozdíl) a nebo v každé „pod“-relaci vstupující do množinové operace. Výše uvedený dotaz s průnikem můžeme zapsat také takto:

```
((KNIHA*AUTORSTVI)*AUTOR(JMENO=Jan and PRIJMENI=Novák)[NAZEV])∩
((KNIHA*AUTORSTVI)*AUTOR(JMENO=Vít and PRIJMENI=Mrkvička)[NAZEV])
```

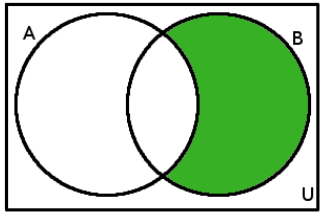
Když se podíváme na obsah původních tabulek, zjistíme, že odpovědí na tento dotaz je kniha **Oko**.

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – rozdíl



$A \setminus B = \{x \in U;\ x \in A \wedge x \notin B\}$



$B \setminus A = \{x \in U;\ x \notin A \wedge x \in B\}$

Poslední zmiňovanou množinovou operací je rozdíl. Vstupem jsou opět dvě kompatibilní relace, výstupem pak relace, která vznikne rozdílem dvou původních relací. Zapisujeme **R-S**.

Tabulka: KNIHA

ID	NÁZEV	JAZYK	ID_AUTOR
11	Oko	český	1
12	Bludiště	německý	4
13	Domek	český	4
14	Sauna a bazén	anglický	3

Tabulka: AUTORSTVÍ

ID_KNIHA	ID_AUTOR
11	1
11	5
14	3
12	4
12	3
13	4

Tabulka: AUTOR

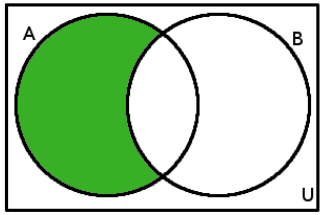
ID	JMÉNO	PŘÍJMENÍ	DATUM_NAROZENÍ
1	Jan	Novák	15.10.1975
2	Petr	Nový	1.4.1978
3	Jan	Nováček	6.9.1965
4	David	Vokurka	5.12.1973
5	Vít	Mrkvička	30.1.1970

A formulujme následující dotaz: *Jaké knihy napsal David Vokurka bez toho, aniž by mu pomáhal Jan Nováček jako spoluautor?* Jinými slovy, hledáme takové knihy, které napsal **David Vokurka** a zároveň je nenapsal **Jan Nováček**. Použijeme operaci rozdíl:

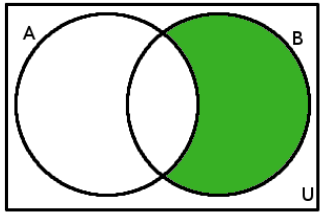
Opět zde máme možnost zapsat uvedený výraz i jiným způsobem:

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

3. Relační algebra – rozdíl



$A \setminus B = \{x \in U;\ x \in A \wedge x \notin B\}$



$B \setminus A = \{x \in U;\ x \notin A \wedge x \in B\}$

Poslední zmiňovanou množinovou operací je rozdíl. Vstupem jsou opět dvě kompatibilní relace, výstupem pak relace, která vznikne rozdílem dvou původních relací. Zapisujeme **R-S**.

Tabulka: KNIHA

ID	NÁZEV	JAZYK	ID_AUTOR
11	Oko	český	1
12	Bludiště	německý	4
13	Domek	český	4
14	Sauna a bazén	anglický	3

Tabulka: AUTORSTVÍ

ID_KNIHA	ID_AUTOR
11	1
11	5
14	3
12	4
12	3
13	4

Tabulka: AUTOR

ID	JMÉNO	PŘÍJMENÍ	DATUM_NAROZENÍ
1	Jan	Novák	15.10.1975
2	Petr	Nový	1.4.1978
3	Jan	Nováček	6.9.1965
4	David	Vokurka	5.12.1973
5	Vít	Mrkvička	30.1.1970

A formulujme následující dotaz: *Jaké knihy napsal David Vokurka bez toho, aniž by mu pomáhal Jan Nováček jako spoluautor?* Jinými slovy, hledáme takové knihy, které napsal **David Vokurka** a zároveň je nenapsal **Jan Nováček**. Použijeme operaci rozdíl:

```
((KNIHA*AUTORSTVI)*AUTOR(JMENO=David and PRIJMENI=Vokurka))-
((KNIHA*AUTORSTVI)*AUTOR(JMENO=Jan and PRIJMENI=Nováček))[NAZEV]
```

Opět zde máme možnost zapsat uvedený výraz i jiným způsobem:

```
((KNIHA*AUTORSTVI)*AUTOR(JMENO=David and PRIJMENI=Vokurka)
 [NAZEV])-
((KNIHA*AUTORSTVI)*AUTOR(JMENO=Jan and PRIJMENI=Nováček)[NAZEV])
```

Zdroj: https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

4. Základní datové typy

Datový typ definuje v programování druh nebo význam hodnot, kterých smí nabývat proměnná (nebo konstanta). Je určen oborem hodnot a zároveň výpočetními operacemi, které lze s hodnotami tohoto typu provádět. Nemůže být určen pouze oborem hodnot, protože existují i datové typy lišící se pouze v operacích, které je s nimi možné provádět.

Celá čísla

Typ	Rozsah
TINYINT	8bitové číslo
SMALLINT	16bitové číslo
MEDIUMINT	24b číslo
INT	32b
BIGINT	64b

Texty

Typ	Rozsah
TINYTEXT	max. 255 B
TEXT	max. 64 KB (to je opravdu dlouhý text, třeba článku)
MEDIUMTEXT	max. 2^{24} B
LONGTEXT	max. 2^{32} B
VARCHAR (max. počet znaků)	počet znaků jaký si zvolíte, max. 64 KB
CHAR (počet znaků)	pevný počet znaků jaký si zvolíte, max. 255

Ostatní

Typ	Rozsah
DATE	Datum jako textový řetězec ve tvaru 'rrrr-mm-dd', rozsah od '1000-01-01' do '9999-12-31'
TIME	Čas jako textový řetězec ve tvaru 'hh:mm:ss'
DATETIME	Prakticky je to zřetězení data a času: 'rrrr-mm-dd hh:mm:ss'
TIMESTAMP	Navenek se chová stejně jako Datetime, ale interně je to dword s počtem sekund uplynulých od 1.1.1970. Rozsah mu končí v lednu 2038, takže tenhle typ moc nedoporučuji a uvádím ho jenom pro úplnost.
BLOB a spol.	Obecná data v binárním tvaru, použití analogické k typům TEXT. Umožňují do databáze ukládat např. obrázky nebo zvuky.

NULL = hodnota nenastavena

Praktické cvičení

- Procvičení relačních operací - HOTOVO

Na doma

Zkonstruujte pomocí znalostí o relační algebře alespoň 10 dotazů z níže uvedených:

Tabulka: ZAMESTNANEC

ČÍSLO	JMÉNO	PŘÍJMENÍ	DAT_NAROZ	ID_FUNKCE	ID_PRACOVISTE
1	Jan	Novák	15.10.1975	1	11
2	Petr	Nový	1.4.1978	2	21
3	Jan	Nováček	6.9.1965	3	31
4	David	Vokurka	5.12.1973	4	41

Tabulka: POVOLANI

ID	FUNKCE	PLAT
1	uklízeč	15000
2	programátor	21500
3	manažer	17500
4	ředitel	28000

Tabulka: PRACOVISTE

ID	ODDELENI
11	chodba
21	studovna
31	kancelář
41	ředitelna

- *Jakou funkci vykonává David Vokurka?*
- *Jaký plat pobírá programátor?*
- *Všichni zaměstnanci (jméno, příjmení) starší 20ti let*
- *Kteří zaměstnanci (jméno, příjmení) pracují na chodbě?*
- *Na kterých pracovištích (název) pracují programátoři?*

Tabulka: KNIHA

ID	NÁZEV	JAZYK
11	Oko	český
12	Bludiště	německý
13	Domek	český
14	Sauna a bazén	anglický

Tabulka: AUTORSTVÍ

ID_KNIHA	ID_AUTOR
11	1
11	5
14	3
12	4
13	4

Tabulka: AUTOR

ID	JMÉNO	PŘÍJMENÍ	DATUM_NAROZENÍ
1	Jan	Novák	15.10.1975
2	Petr	Nový	1.4.1978
3	Jan	Nováček	6.9.1965
4	David	Vokurka	5.12.1973
5	Vít	Mrkvička	30.1.1970

- *Jaké knihy (název) napsal Vít Mrkvička?*
- *Kdo napsal (jméno, příjmení) knihu „Bludiště“?*
- *V jakém jazyce byla napsána kniha „Sauna a bazén“?*
- *Jaké knihy (název) napsali Petr Nový a Jan Nováček jako spoluautoři?*
- *Kdy se narodil autor/autoři knihy „Domek“?*
- *V jakých jazycích napsal knihy Jan Novák?*
- *Jaké knihy (název) nenapsali Vít Mrkvička a David Vokurka jako spoluautoři?*

Tabulka: BYT

CISLO	ADR_ULICE	ADR_CISLO	ADR_MESTO	ADR_PSC	PODLAZI	BALKON	SKLEP
1	Novákova	123	Brno	61500	2	A	N
2	Jirkova	2	Praha	11400	1	A	A
3	Tulíkova	900	Zlín	57890	7	N	N
4	Prackova	7	Znojmo	52109	3	A	A
5	Holíkova	1	Brno	62100	2	N	A
6	U Boba	100	Zlín	57880	6	N	N
7	Krtkova	88	Praha	11100	-1	N	N

- *Ve kterém městě (název) leží ulice Tulíkova?*
- *Které byty (číslo) leží nejvýše ve 2. podlaží?*
- *Jaká je přesná adresa bytu č. 5?*
- *Seznam bytů (čísla) s balkonem.*
- *Seznam bytů (čísla) buď s balkonem a nebo se sklepem (ale ne dohromady).*

Děkuji za pozornost a těším se příště!

Prostor pro zpětnou vazbu

mail@mjakl.cz

Tel. 608 544 839

Web: mjakl.cz

Telegram: t.me/mikimikimikia