

Datové a procesní modely

[Relační databáze]

Přednáška 1

Marian Kamenický

Synteia software group a.s.
marian.kamenicky@synteia.cz

MFFUK Praha

2019/20

Datové a procesní modely

Relační databáze

XML

LDAP [adresřové služby]

NoSql





a co trochu historie ???

Charles Babbage

matematik, filozof, vynálezce, inženýr

1835 nápad sestavit programovatelný počítač

mechanizace počítání

Babbageovi počítačí stroje - nikdy dokončeny

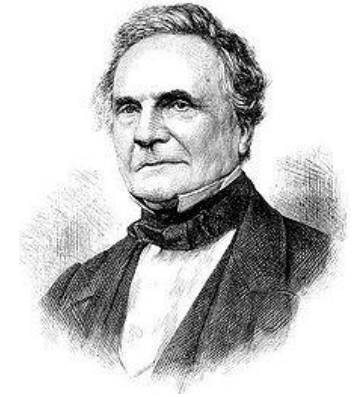
oddělena paměť datová a programová

paměť - sklad

procesor - mlýnice

hádky s řemeslníkem vyrábějícím ozubená kolečka

petice proti flašinetářům



26. prosince 1791

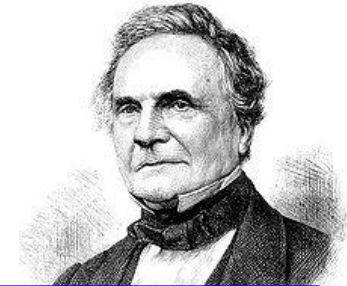
† 18. října 1871



Charles Babbage

matematik, filozof, vynálezce, inženýr

1835 nápad sestavit programovatelný počítač



- [v 19 století]
- vymyslel **základní principy** fungování stroje
- pro řešení složitých výpočtů
- !!!

hádky s řemeslníkem vyrábějícím ozubená kolečka

petice proti flašinetářům



Ada Lovelace

- dcera lorda Byrona
- **Augusta Ada King, hraběnka z Lovelace**
- první programátor - ka na světě
- financovala vývoj stroje **Babbage**
- navrhla, jak počítač naprogramovat pomocí děrných štítků
- základy algoritmizace - zavedla pojmy 10. prosince 1815
- podmíněný + nepodmíněný skok 27. listopad 1952
- cyklus
- podprogram ...
- programování označovala jako ***poetickou vědu***

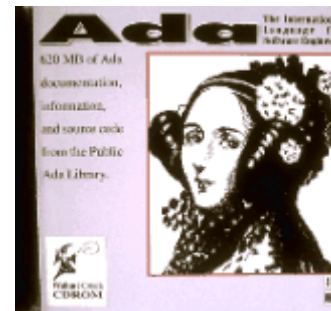


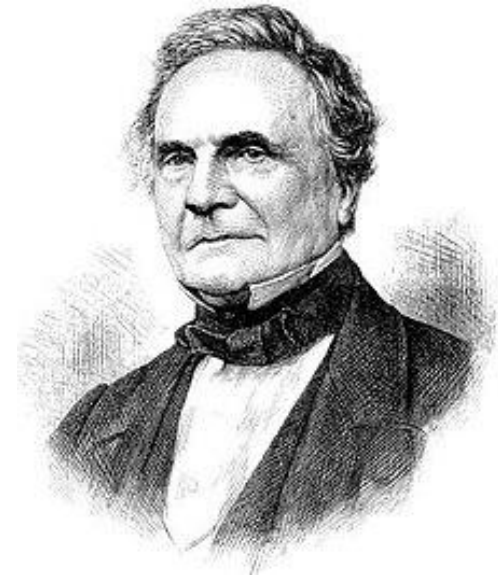
myšlenky o budoucnosti počítačů

- řešení komplikovaných matematických úloh
- komponování hudby
- malba obrazů
- na její počest programovací jazyk **ADA**



Ada
The Language For A Complex World





26.prosince 1791 Londýn

18.října 1871 Londýn (ve věku 79 let)

Zaměstnavatel Univerzita v Cambridgi [profesor matematiky]

Ocenění člen Královské společnosti (1816)

1824 Zlatá medaile Královské astronomické společnosti

Po něm pojmenován kráter Babbage na přivrácené straně Měsíce

_2002 Sestrojili jej po 182 letech - Vystaveno v Kalifornském muzeu

Tvoří jej **5 tun koleček**

Točí se **ručně klikou**

Někdy se zasekne a je třeba výpočet přerušit,
součástky odblokovat a začít znovu.

Je to opravdový [a vlastně jediný] **HARDWARE**



1888 Herman Hollerith děrnoštitkový stroj - patent

1890 konkurz na mechanizaci sčítání lidu USA

(předchozí **1880** trvalo 7 let)

sčítání trvalo 3 měsíce

uspořilo 5 mil. Dolarů

1890 Rakousko-Uhersko sčítání lidu

(naši dědečkové, pradědečkové sčítání hollerity)

1896 Tabulating Machine Company [a.s.]

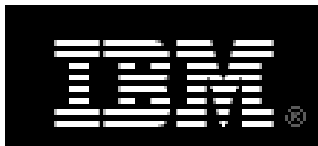
1911 Fuze s Computing-Tabulating-Recording Company

1924 změna jména International Business Machines (IBM)



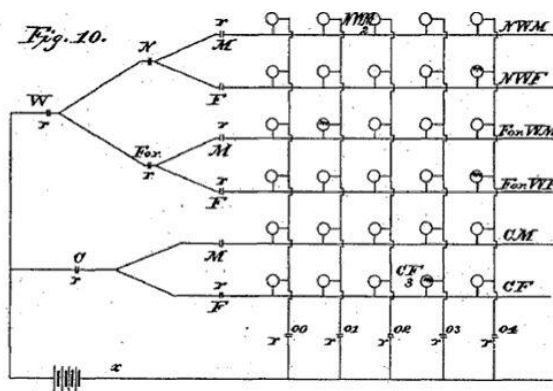
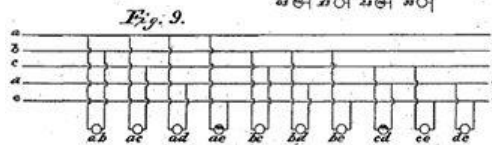
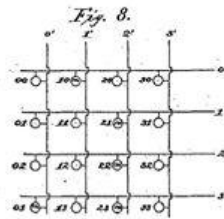
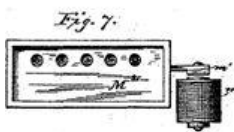
29.února 1860

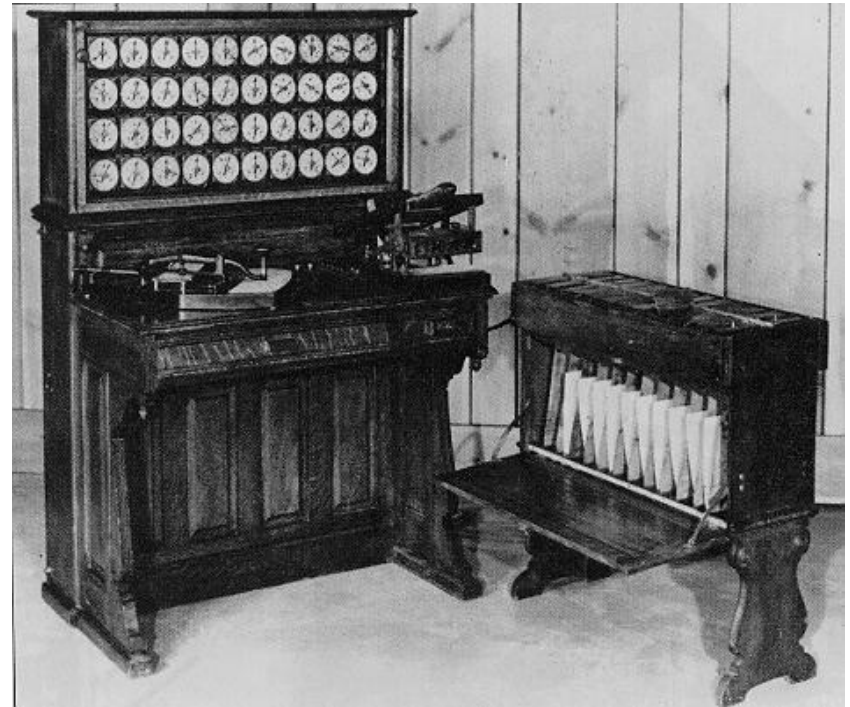
17.listopadu 1929



Co bylo na Holerritovi jedinečné

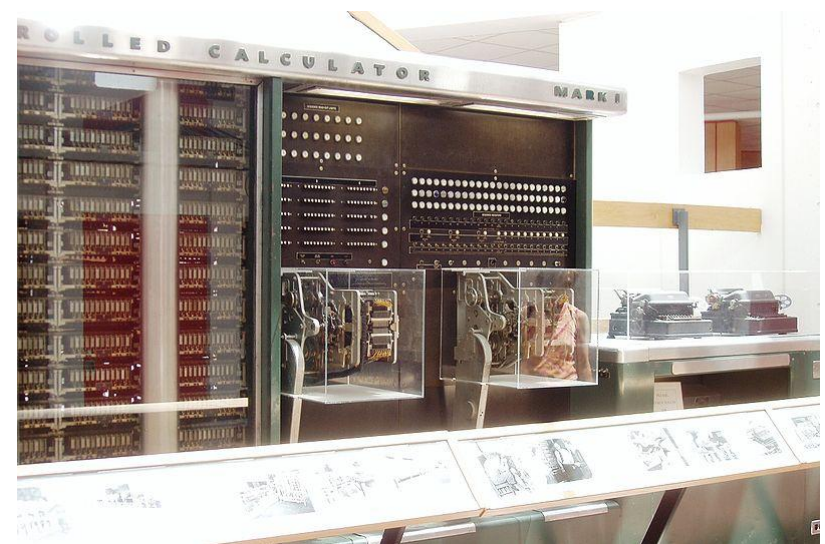
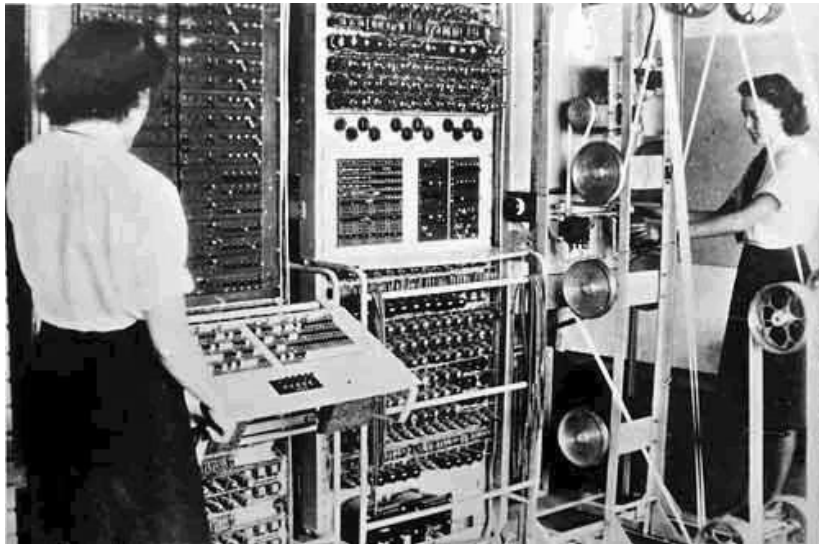
- děrný štítek použil jako nosiče dat
 - před tím program pro tkalcovské stavy
- přístup k informacím
 - každá informace byla uložena jako číslo



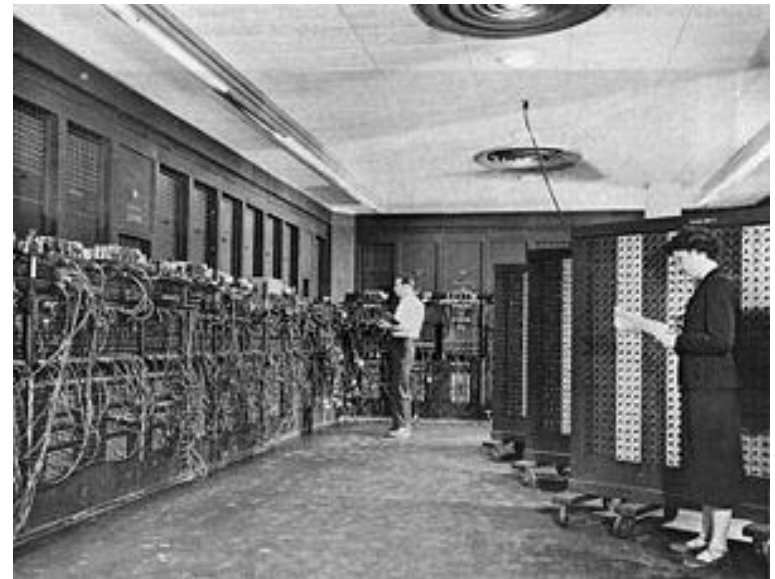
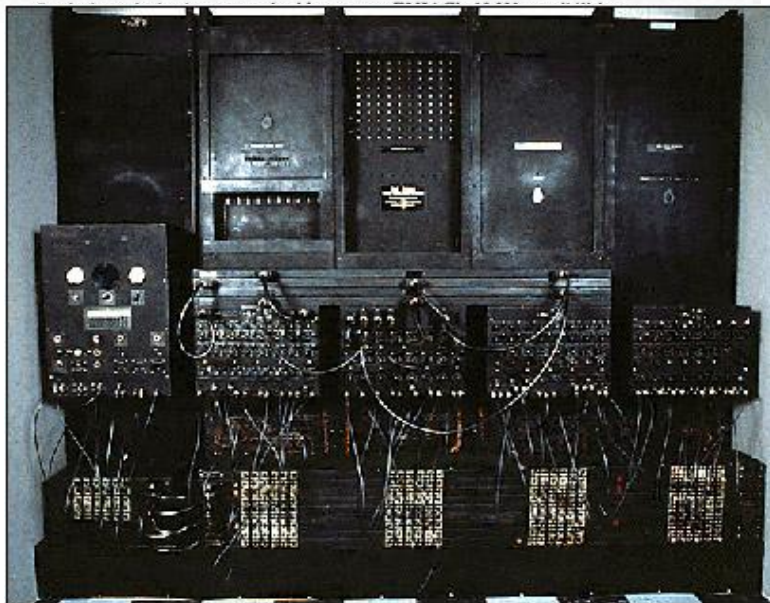
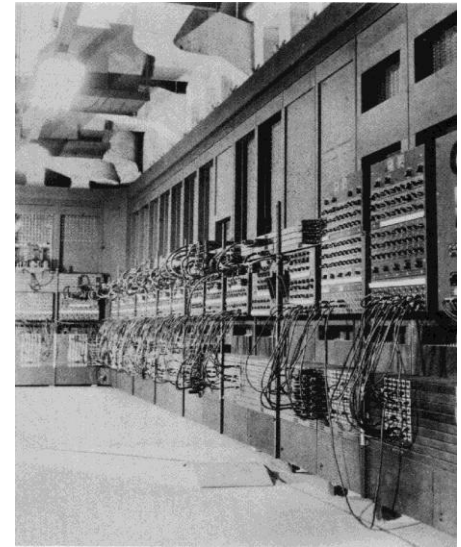
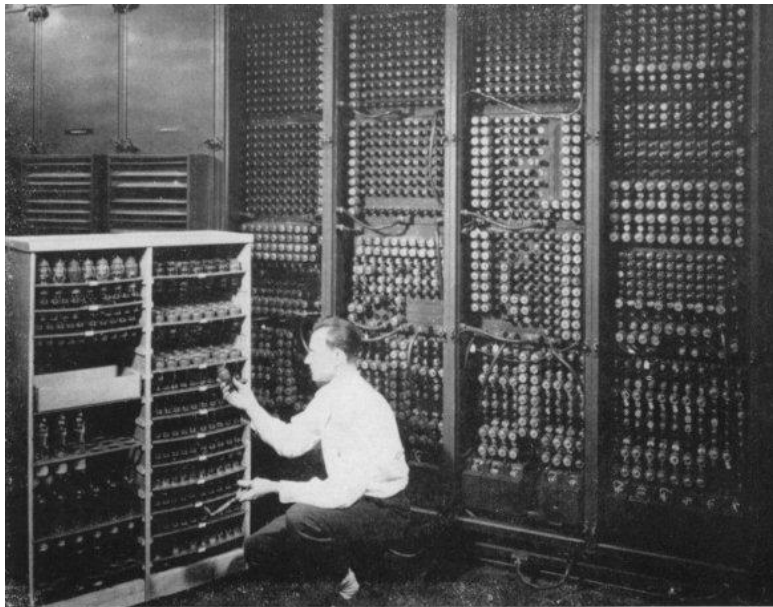


Howard Aiken

- + IBM
- 15m dlouhé monstrum
- 4 počítače, 3 manželky
- [Mark I, Mark II, Mark III, Mark IV]



- Electronic Numerical Integrator And Computer
- první elektronický počítač
- 5 000 operací / vteřinu
- 17 000 elektronek, 10 000 kondenzátorů, ...
- 150 m² , 30 tun
- poruchový velmi - praskaly elektronky
- zapínání+vypínání ==> nepřetržitý provoz
- zapnutí ==> pouliční světla Philadelfie blikala
- vlastní elektrárna - vedlejší budova
- komplikované programování - nastavování přepínačů

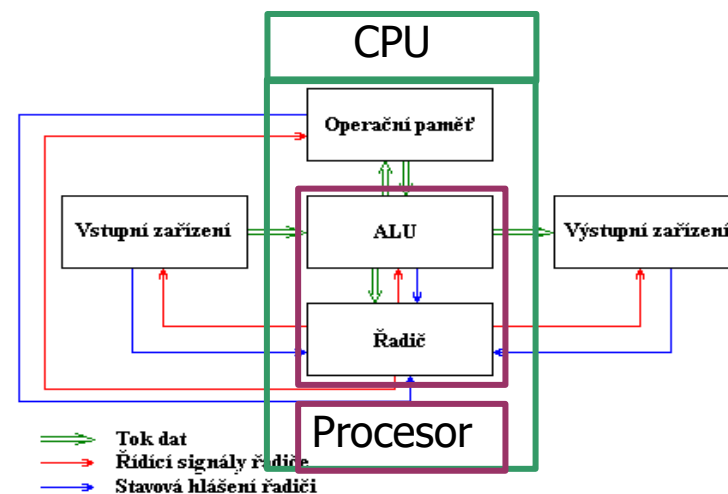


- Mechanical And Numerical Integrator And Computer
- vypočet vodíkové pumpy
- John von Neumann
- John Presper Eckert
- John W. Masuchly

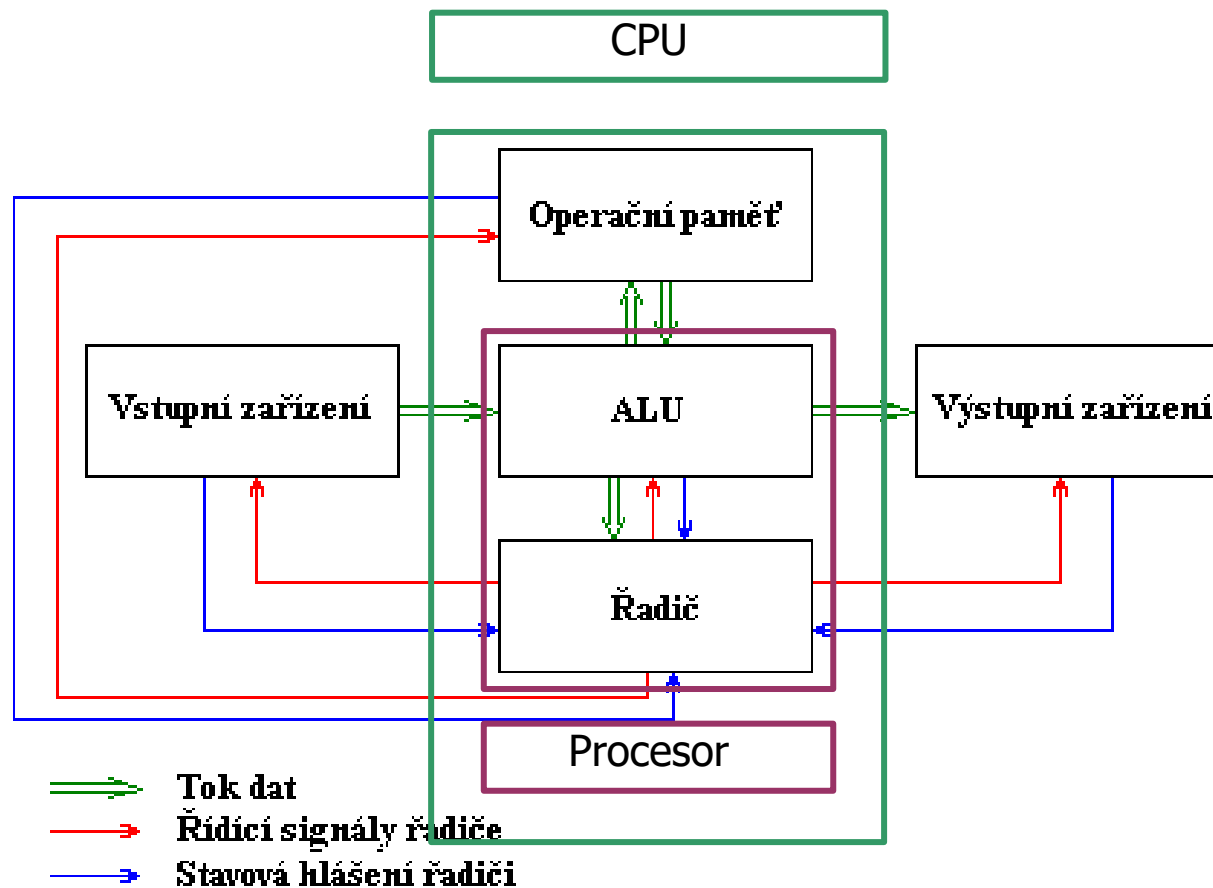


John von Neumann

- zcela nová koncepce počítače
- skládá z několika základních funkčních částí
- program se ukládá do paměti
- spojil paměť pro data i program
- vykonává se postupně tak,
- jak byl uložen
- prosadil
- dvojkovou soustavu



John von Neumann





- maďarský matematik židovského původu
- kvantová fyzika, funkcionální analýza, teorie množin,
- ekonomika, numerická analýza, informatika,
- hydrodynamika, statistika
- od malička znaky geniality
- jazykové nadání, neobyčejná paměť
- zpaměti uměl dělit 8-mi místnými čísly
- ve 12 soukromá výuka univerzitními profesory
- v 17 první vědecká práce
- na univerzitě - chemické inženýrství - se nudil
 - udělal bokem doktorát z matematiky





co takhle malou odbočku

sčítání lidu

- kdo, kdy zavedl sčítání lidu u nás ???
- náznaky 1058 - Spytihněv II
 - soupis majetku litoměřického kostela

- Marie Terezie Valpurga Amálie Kristýna
- 1753 patent
- roční sčítání od roku 1754
- současně, jednotně na celém území
- vyjimečný počín





- 12. srpna 1981



- Control Program/ Monitor

19.května 1942

- **Gary Kildall**

11.července 1994



- Control Program/ Monitor
- **Gary Kildall**



MS-DOS



Windows





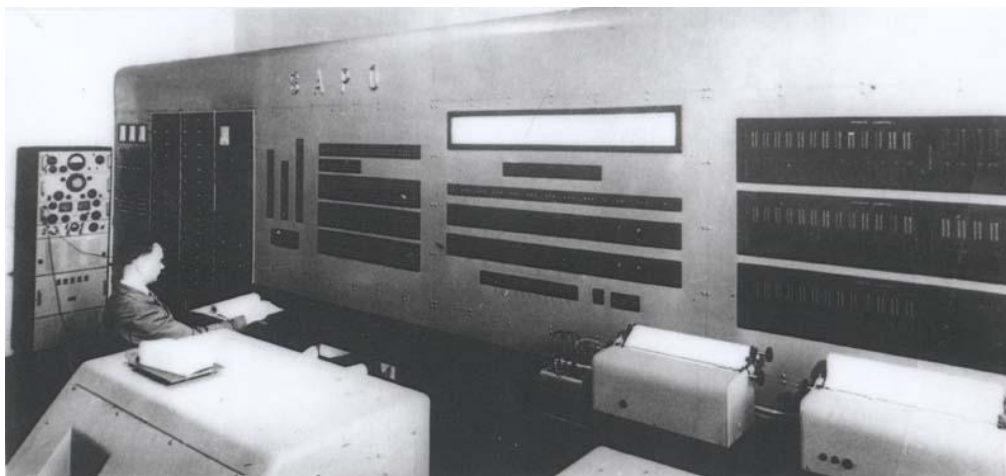
???

konec sázení

- A. bída, nouze 100 bodů
- B. průměrná 100 bodů
- C. světová 100 bodů

SAPO 1951-1957

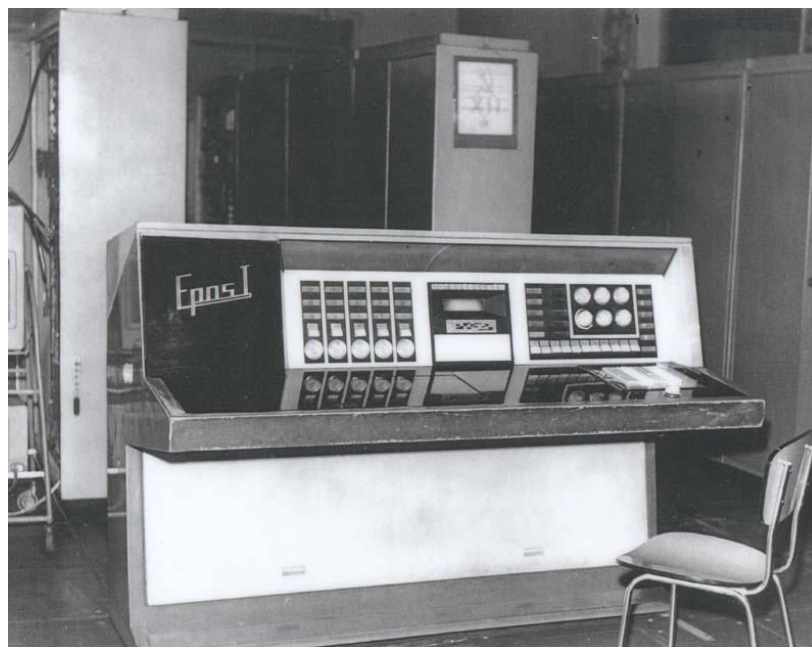
- 7000 relé a 400 elektronek
 - 5 adresový (2 operandy, výsledek, skoky Ano,Ne)
 - 3 shodné procesory pracují současně
 - výsledky se porovnávají a hlasuje se o něm
-
- 1960 3 roky po vytvoření shořel



EPOS1 Elektronický POčítačový Stroj

1960

- čistě dekadický
- multiprogramování
 - výhradně hardwarové prostředky
- důkladné zajištění vůči chybám paměti



EPOS1

Elektronický **PO**čítačový **S**troj

1960

- čistě

-

A kdo to v Česku všechno vymyslel a zorganizoval ?

-

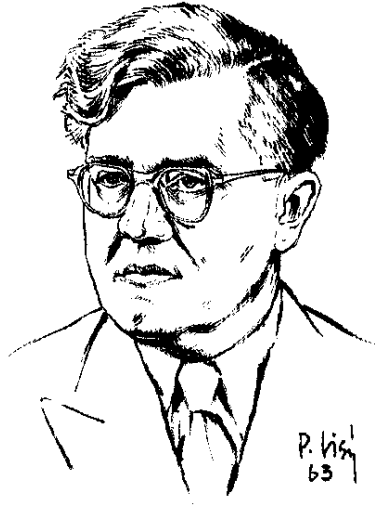


Gigant a zakladatel
české počítačové vědy
profesor MFF UK
a světový vědec

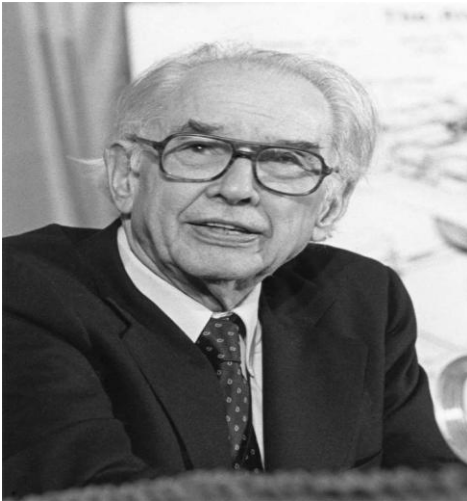


- 1936-9 zaměřovač pro protiletadlové dělostřelectvo
- 1943 Radiation laboratory - Massachusettská technika Boston
 - von Neumann, Norbert Wiener,
 - Claude Shannon

- prestižní ***Naval Ordnance Development Award***
- 1946/8 ***Computing mechanisms and linkages***
 - nestačí na habilitaci v Praze
- 1950 Utav matematických strojů
 - 1951 SAPO (1957 druhý v Evropě)
- 1955 Výzkumný ústav matematických strojů VÚMS
- 1963 EOPS1, později EPOS2



- před 2 měsíci Václav Trojan objevil že
- první počítač objevil a sestavil americký bulhar
- protože Trojan má ženu bulharku



John Vincent Atanasoff

1903 - 1995

nějaká konstansta Helia

postavil si počítač za 650 dolarů

když grant skončil musel vrátit peníz

požádal o patetizaci—skončil v archivu

1985 to v archivu v Iově

obejevili a znovu podali

bylo nakonec uznáno



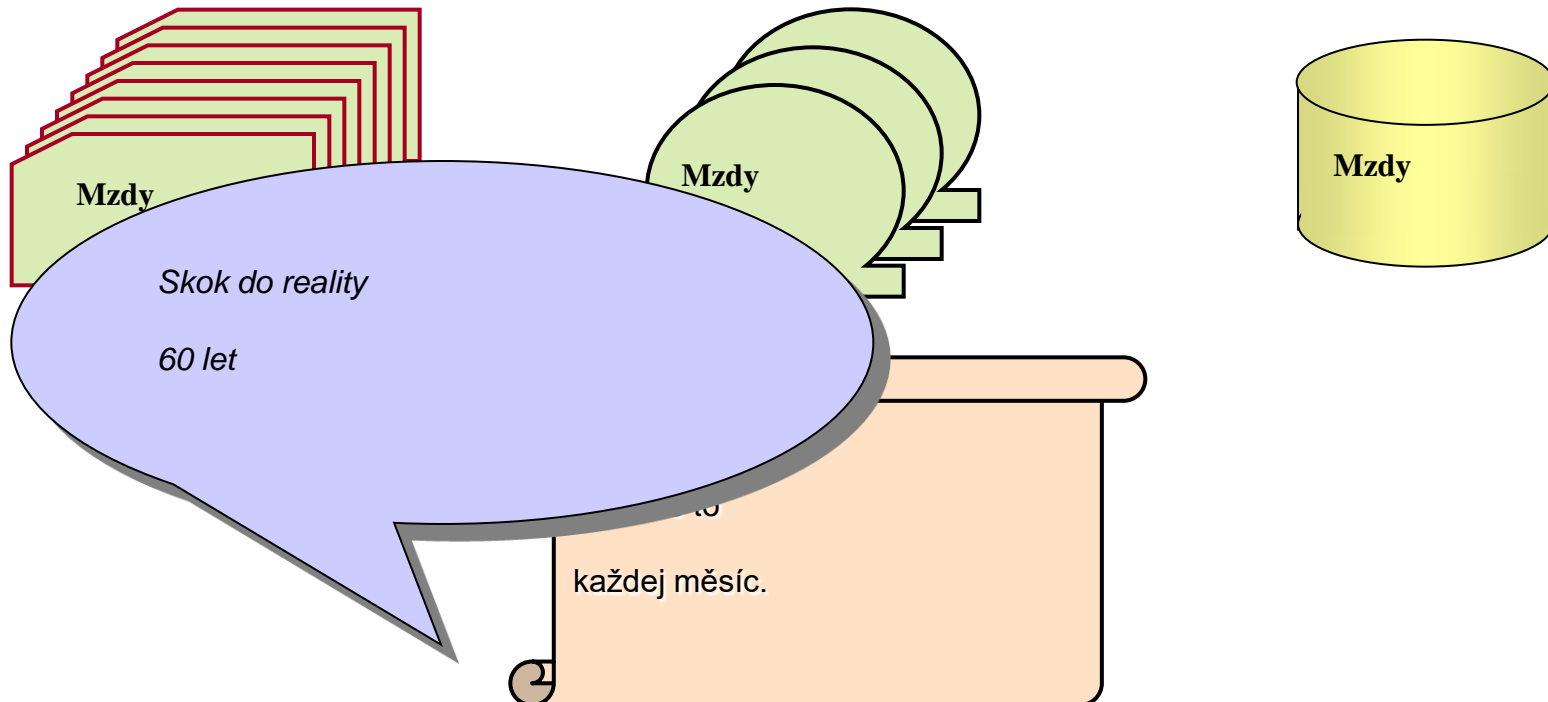
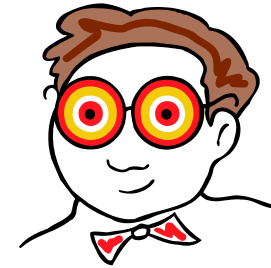


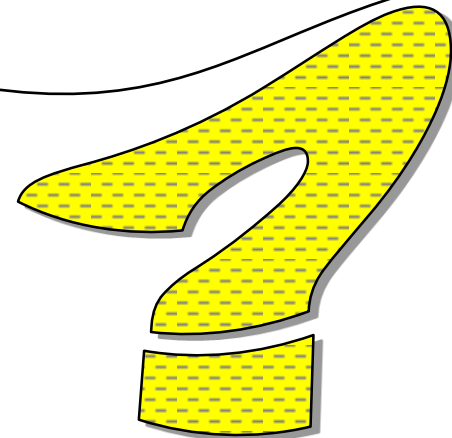
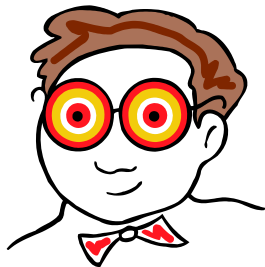
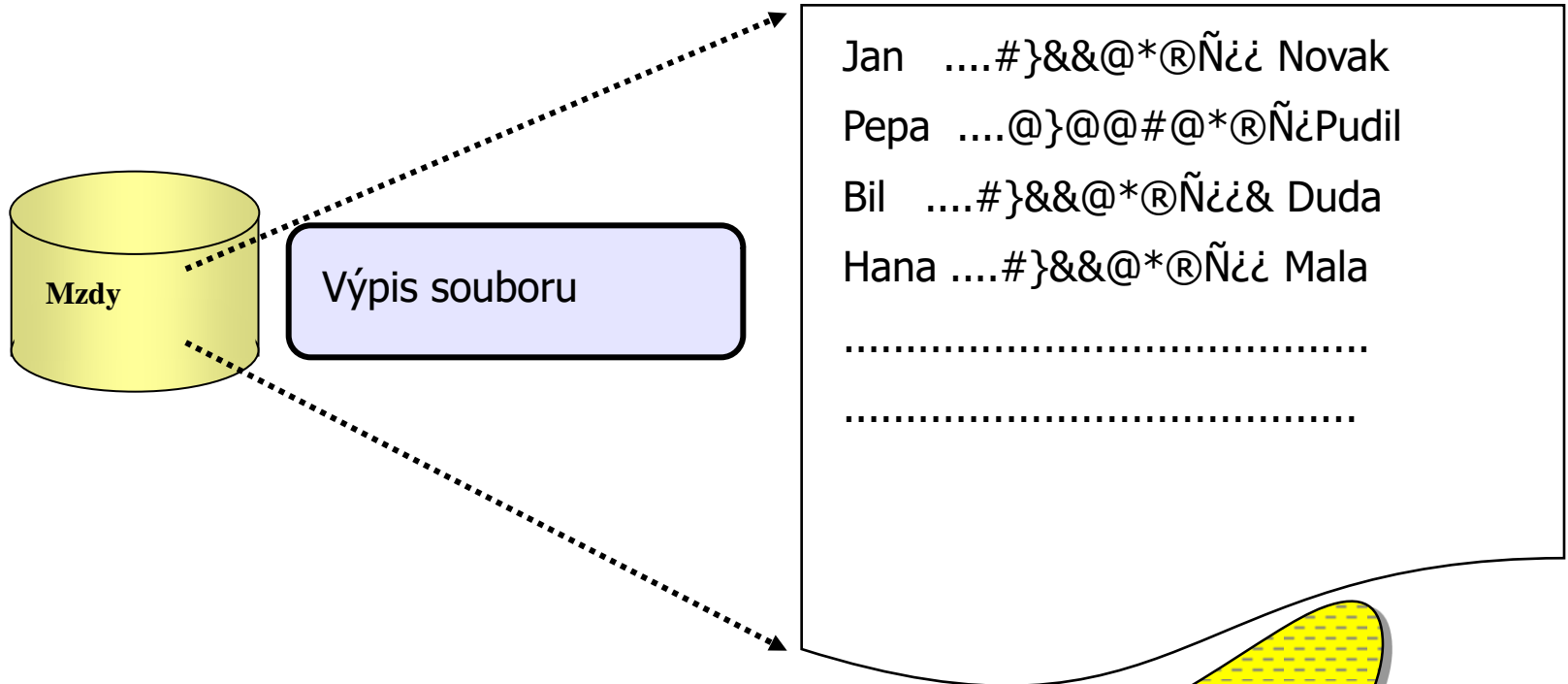
historie nám končí

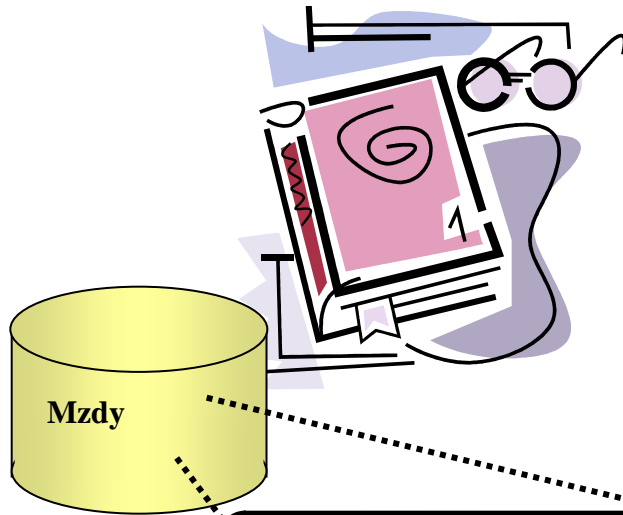
a jiná začíná



- převezmi agendu Mzdy
- udělej s tím, co umíš







DOKUMENTACE MEZD

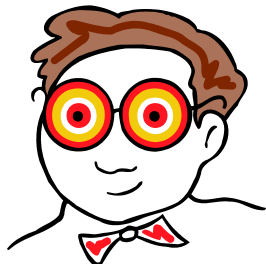
1..10	Jmeno	znakově
11..18	Datum Narozeni	pakovaně
19..23	Plat	integer
.....		

Výpis souboru

Jan	04/65/0736	60
Pepa	88/18/1233	4
Bil	00/00/1000	222222
Hana	44/44/4444	444444
.....		
.....		

Parametry Výpisu

(1,10,CH),(11,8,PD),(19,4,I)



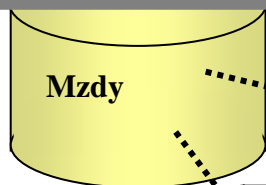
▪ sakra !

DOKUMENTACE MEZD

1..10 Jmeno
11..18 Datum Narozeni
19..23 Plat



a co změny ?



Výpis souboru

Parametry Výpisu

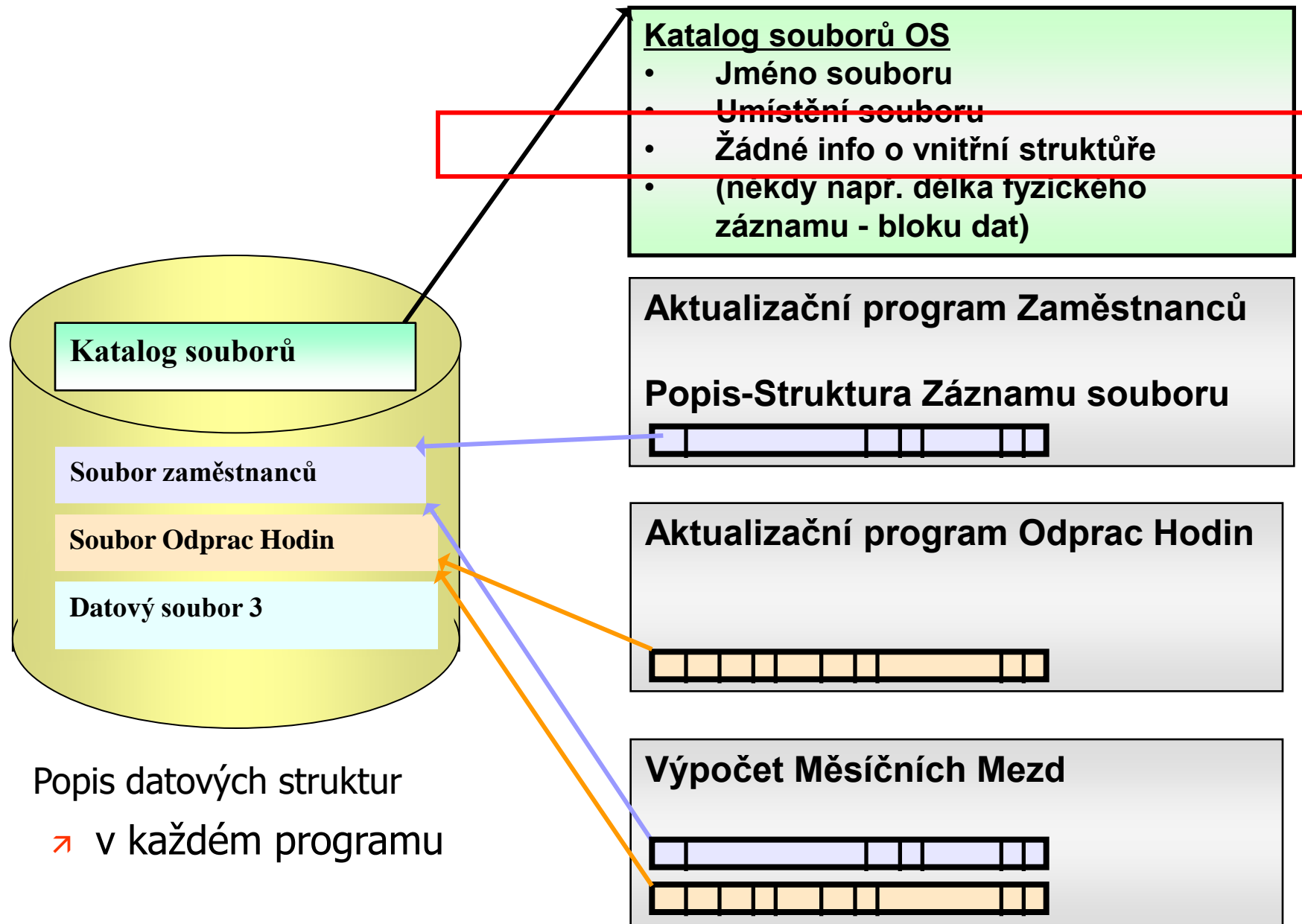
(1,12,CH),(13,8,PD),(21,4,I)

DOKUMENTACE MEZD [ZMěNA]

1..12 Jmeno znakově
13..19 Datum Narozeni pakovaně
21..25 Plat integer

.....

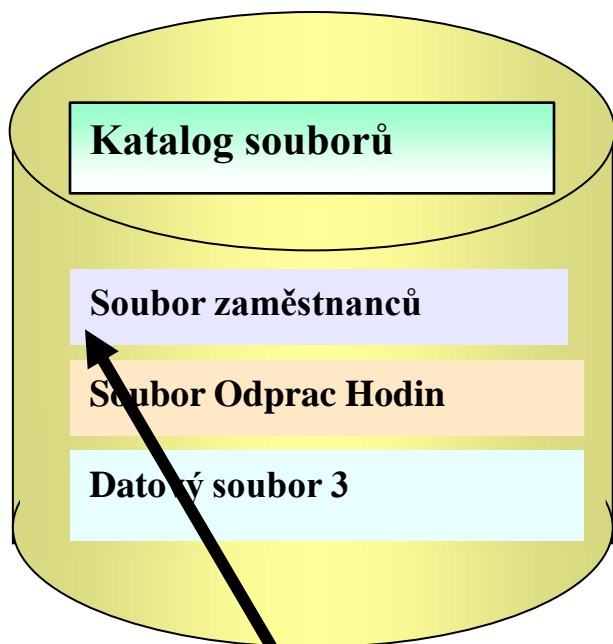
Jan	23/11/1950	2 500
Pepa	13/04/1962	4 200
Bil	01/04/1968	2 222
Hana	08/08/1959	1 600
.....		
.....		



- Popis datových struktur
 - v každém programu

Katalog souborů OS

- Jméno souboru
- Umístění souboru
- **Žádné info o vnitřní struktúře**
- (někdy např. délka fyzického záznamu - bloku dat)



Aktualizační program Zaměstnanců

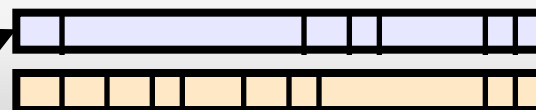
Popis-Struktura Záznamu souboru



Aktualizační program Odprac Hodin



Výpočet Měsíčních Mezd



Změna struktury zaměstnanců

➤ změna dat

➤ změna programu 1

➤ změna programu 2

```
----Struktura zamestnancu---  
DECLARE  
1 VETA_ZAM,  
  2 JMENO CHAR(10),  
  2 PRIJM CHAR(20),  
  2 PLAT DECIMAL (6,2),  
.....
```

Aktualizační program Zaměstnanců

Popis-Struktura Záznamu souboru



```
DECLARE  
1 VETA,  
  2 A CHAR(10),  
  2 B CHAR(20),  
  2 C DECIMAL (6,2),  
.....
```

Aktualizační program Odprac Hodin



Výpočet Měsíčních Mezd



- manuální obsluha sálových počítačů
- zálohování souborů
- havárie úlohy
 - obnova souborů z kopie
 - opakování úlohy

- programy
 - každý program obsahuje vše
 - včetně popisu datových struktur
 - při změně datových struktur
 - třeba opravit všechny programy
 - případná dokumentace v papírové formě
 - ne vždy aktualizovaná

- <1959 veškerý vývoj SW - výrobci počítačů (IBM, ...)
- 1959 Conference on Data Systems Languages CODASYL
 - ustavené ministerstvem obrany USA
 - konference zástupců firem
 - požadavek definice univerzálního jazyka
- 1960 CODASYL publikuje COBOL
- nnnn využití organizační codasylské struktury
 - pro kultivaci a "normalizaci" **datařiny**

- a s čím prosím v tom čurbesu začali ?

- tot' i má otázka



- soubory [mag.páskym disky, děrné pásky, ...]

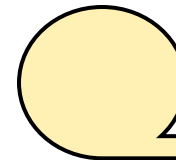
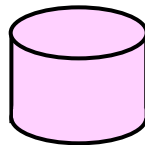
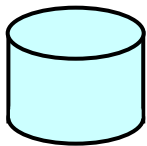
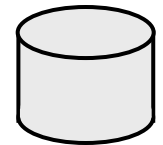
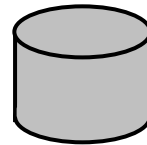
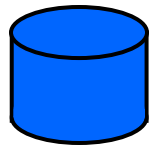
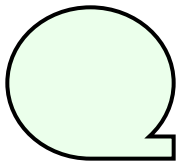
- pro určitou informační "agendu"

 - roztroušené

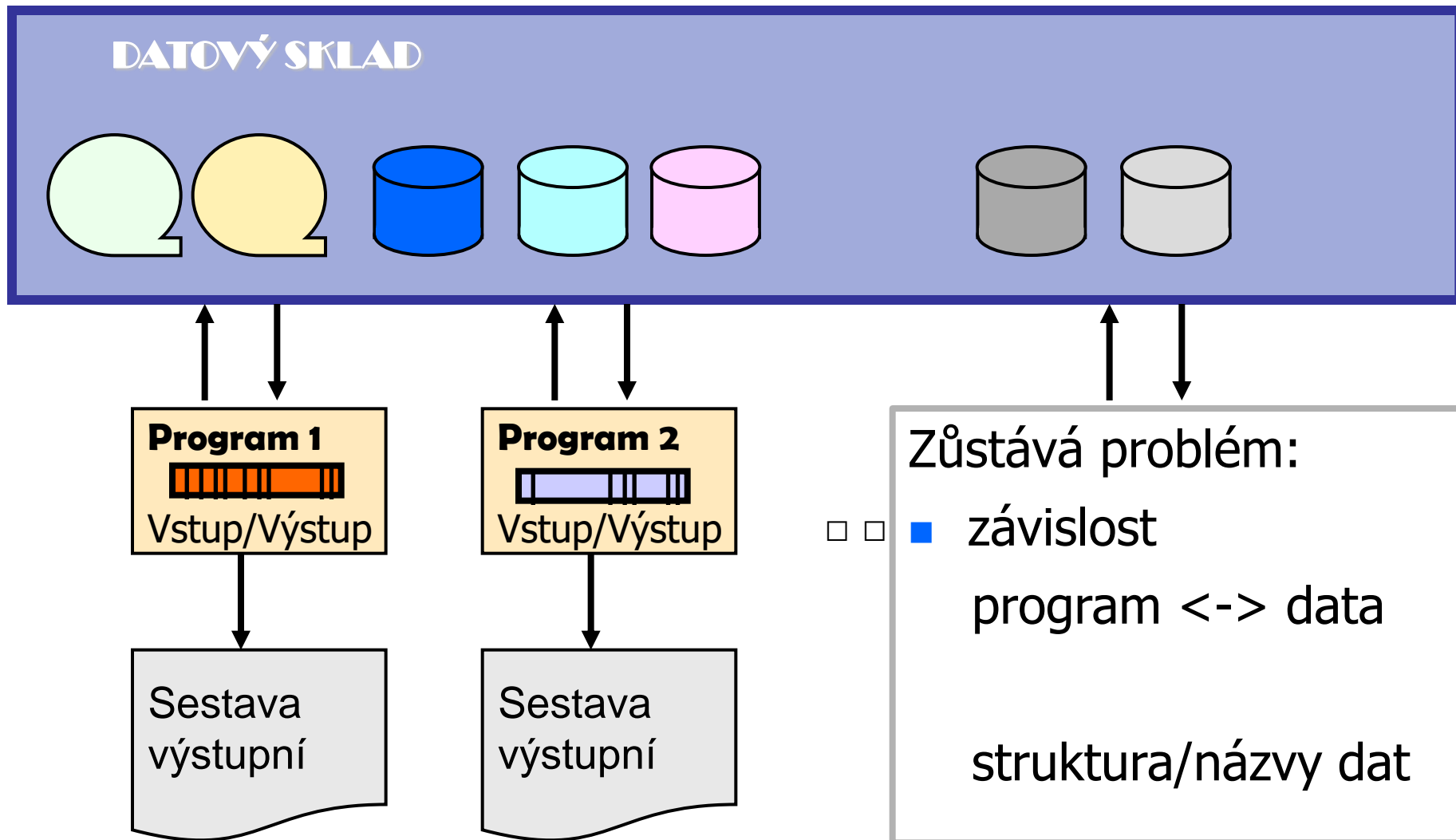
 - bezprizorné

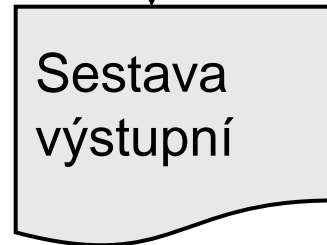
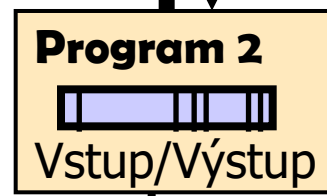
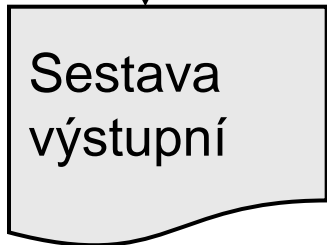
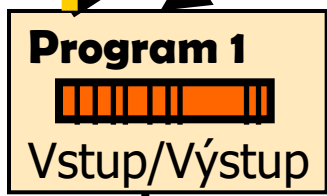
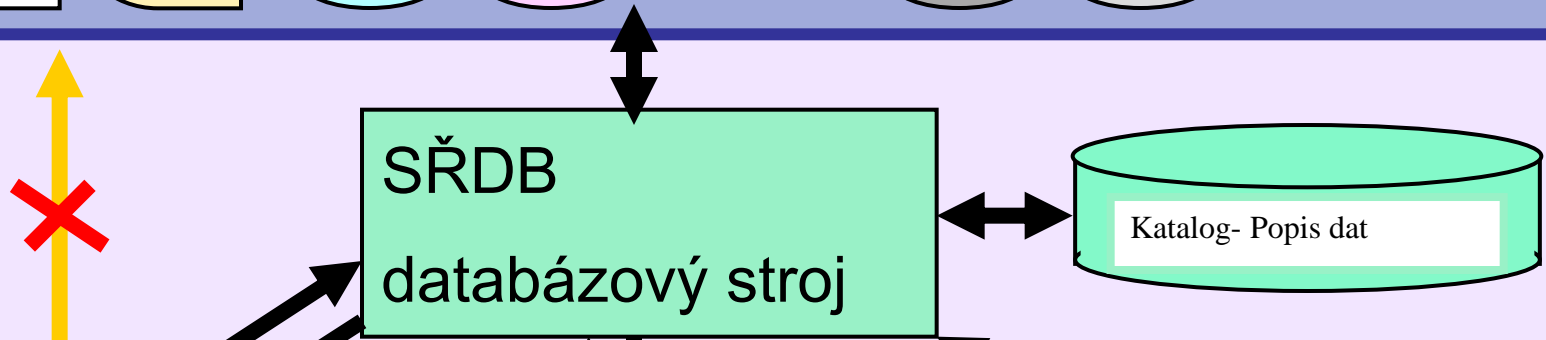
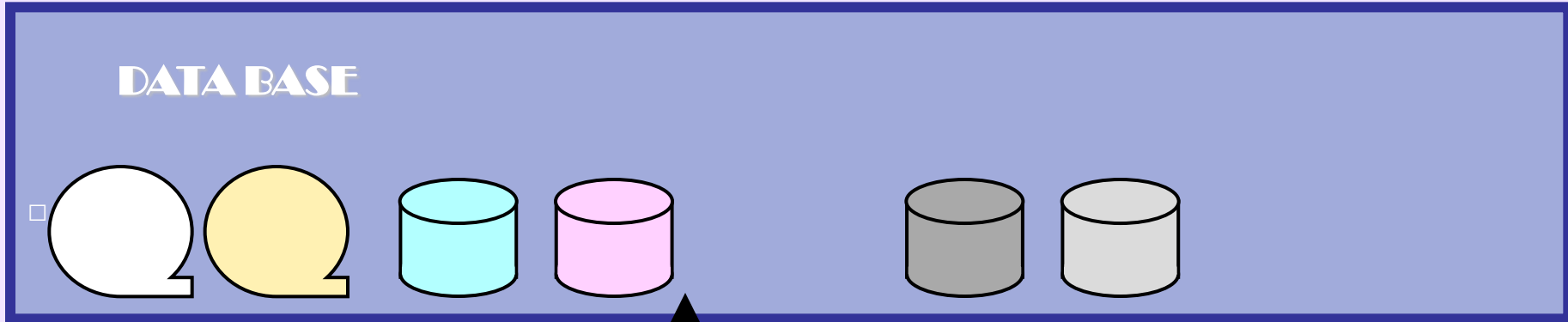
DATOVÝ SKLAD

□ □ □

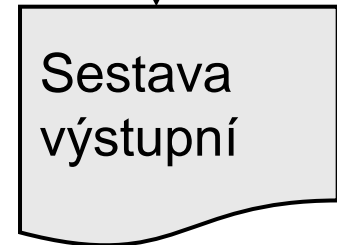
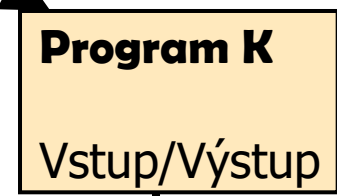


- vytvoření jednotného datového skladu
- všechny aplikační programy [jedné agendy] jej užívají

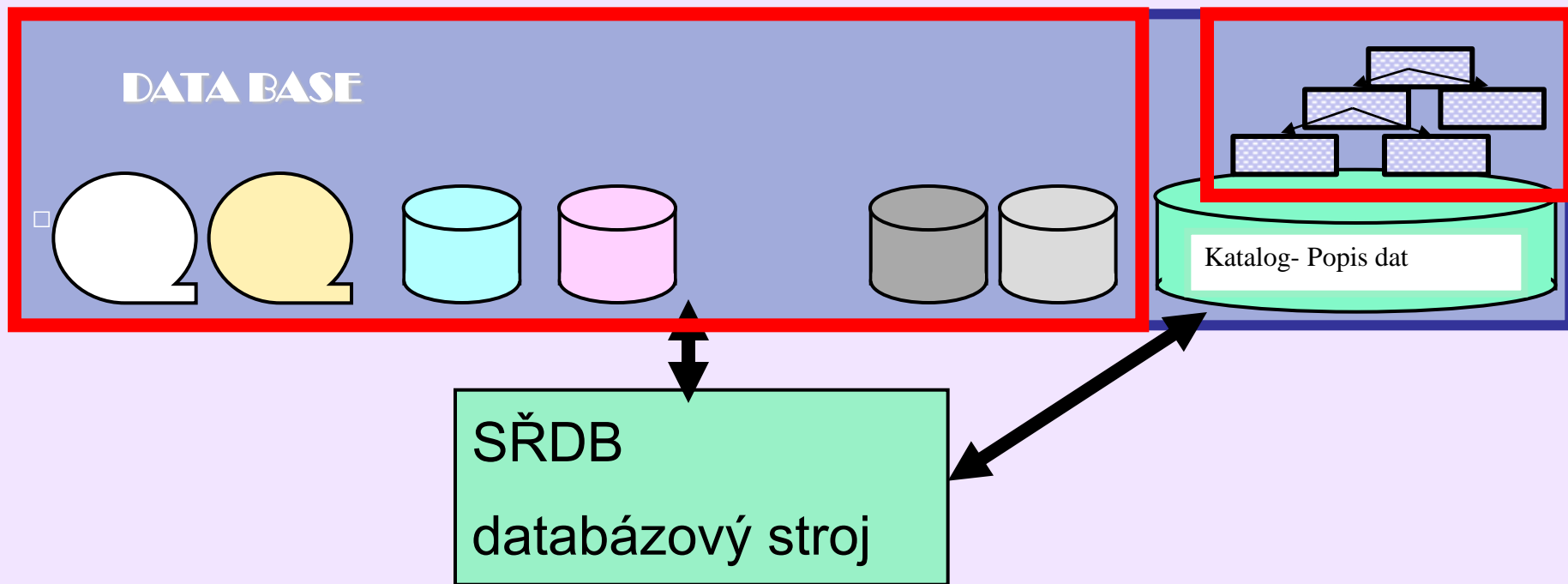




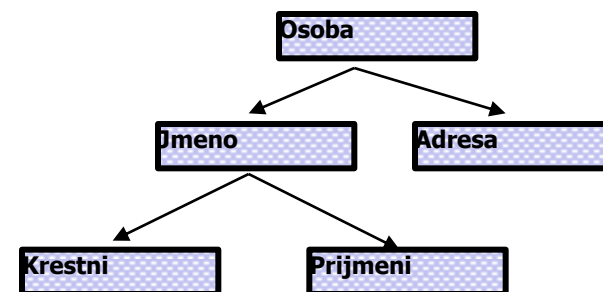
□ □ □

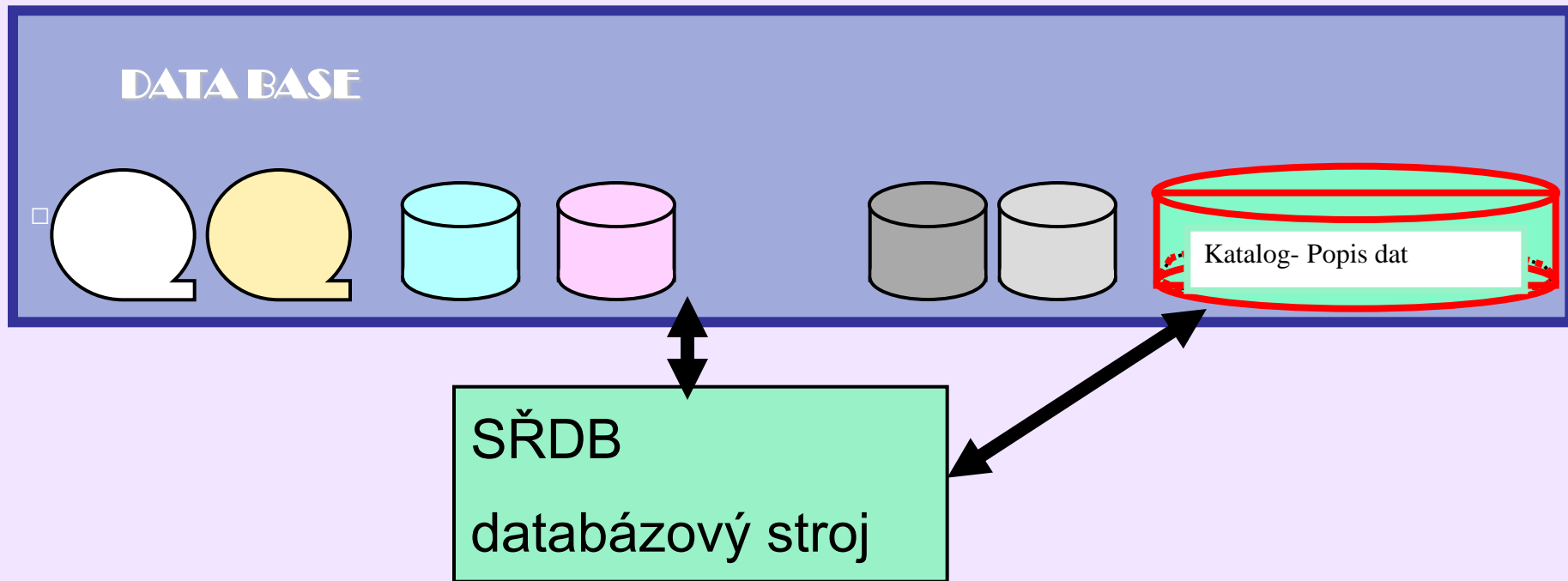


Čemu říkááme databáze ?

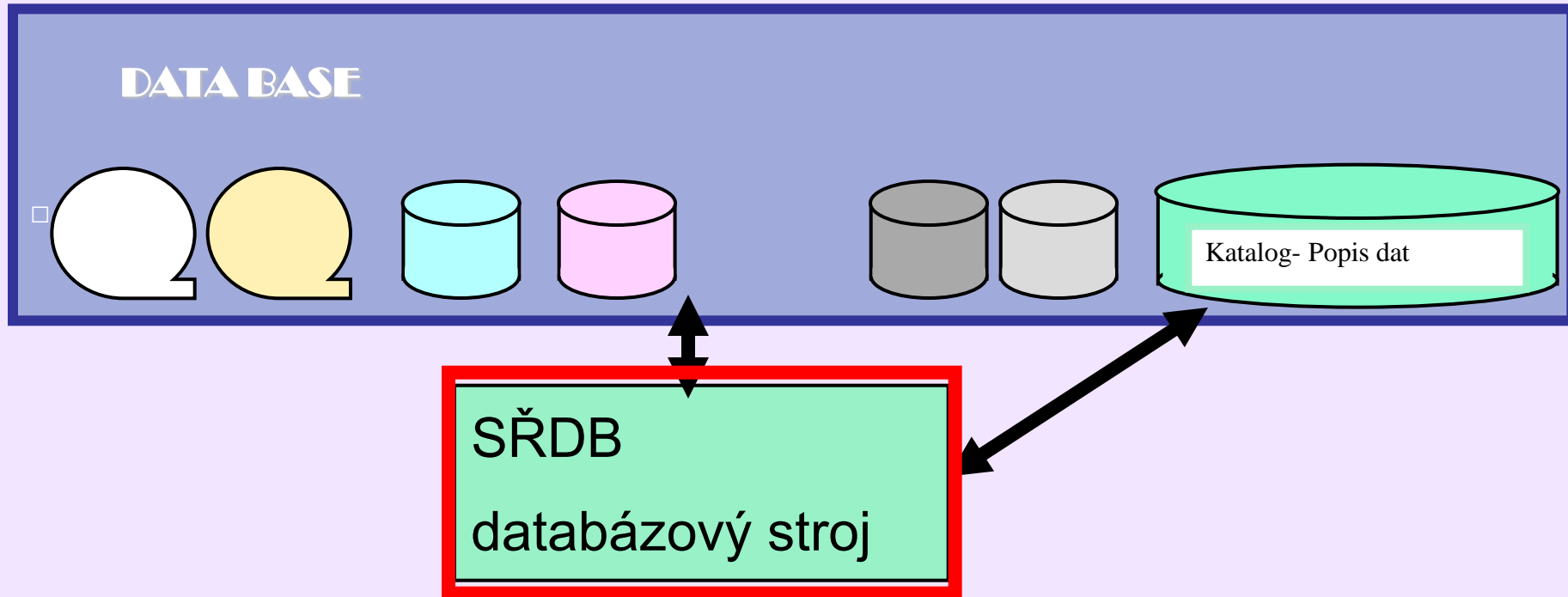


- Vlastním datům (údajům)
- Modelu dat
 - definovaný popisem dat





- Vlastním datům (údajům)
- Modelu dat
 - definovaný popisem dat



- Vlastním datům (údajům)
- Modelu dat
 - definovaný popisem dat
- SŘDB - DB stroji [program]
- Všemuhromady

- nástroj pro shromáždění a uspořádání
 - (vzájemně souvisejících) informací
- se kterými lze pracovat jako s **kompaktní jednotkou**
 - [data spojená se svým popisem]
- velké uložení dat
 - jako jedinné - data užívána **současně** mnoha uživateli
- **sebepopisné** uložení integrovaných záznamů
- strukturovaný soubor trvalých informací uložených počítačovým programem
- **prosím, nešlo by to jednodušeji ?**



- uložště [dat]
- rozsáhlé [velmi]
- celistvé [kompaktní - ne kousek tady, kousek tam]
- sebepopisné [nezávislost dat]
- vztahy [vzájemně souvislá data]
- konkurence [mnoho uživatelů]

- **to je hezké, ale**

- **co vlastně je v databázi uloženo ???**



Informační systém

???

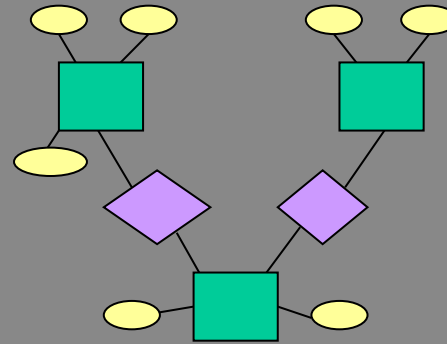
Databáze



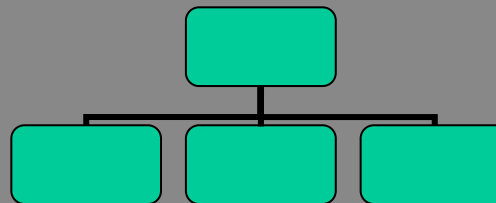
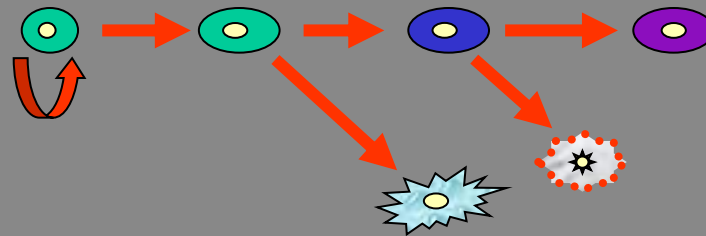
digitální svět

virtuální svět

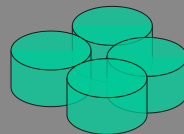
- datové struktury
- procesy
- soft.moduly



digitální svět



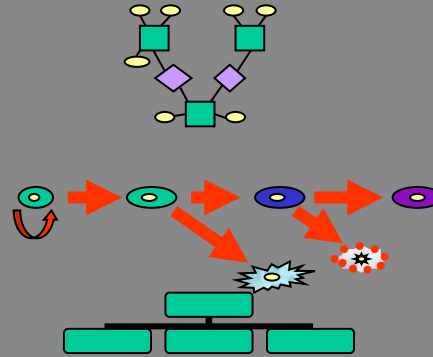
-



virtuální svět

- datové struktury
- procesy
- soft.moduly

digitální svět

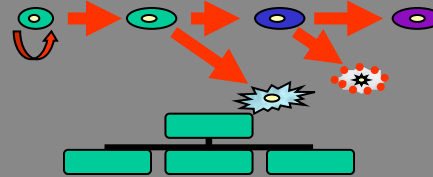
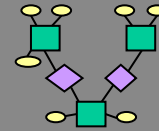


reálný svět



virtuální svět

- datové struktury
- procesy
- soft.moduly
-



digitální svět

reálný svět



virtuální svět

- datové struktury
- procesy
- soft.moduly
-

obraz reálného světa

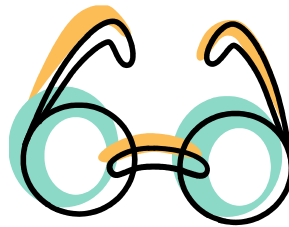


digitální svět



modelování

- zjednodušení
- metodika
 - jak popsat reálný svět
 - E-R model



transformace

reálný svět



Co to je ?

- (zjedodušená definice reálného světa)
- datový model
- který se dívá na reálný svět jako na množinu objektů
- a vztahů mezi nimi


Co je jeho cílem ?

- primárním cílem je návrh databáze (konceptuální)

Existují jiné modely než datové ???

- procesní modely
- funkční modely
- ...

- základní prvek modelu
 - základní prvek definice reálného světa (objekt)

- objekt reálného světa
- rozlišitelný a identifikovatelný
- je předmětem našeho zájmu 
- má smysl o něm podávat informace

něco

čemu stojí za
to dát jméno



- základní prvek modelu
 - základní prvek definice reálného světa (objekt)

- **4** objekt reálného světa
- **b** rozlišitelný a identifikovatelný
- **o** je předmětem našeho zájmu
- **d**
- **y** má smysl o něm podávat informace

něco

**čemu stojí za
to dát jméno**



Entity jsou (lidé, věci, místa, ...)

o kterých chceme uchovávat informace

konkrétní

- lidé



- zvířata



- věci hmotné - předměty

**abstraktní**

- věci nehmotné

Konto**Ohodnocení**

- události

Objednávka**Faktura****Dodávka**

- pojmy

Norma**Spotřeba materiálu**

konkrétní**Reálný svět lze kategorizovat různě**

- lidé



- zvířata



- věci hmotné - předměty



- **místa**



- ■ ■ ■

abstraktní**Abstraktní svět lze kategorizovat různě**

- věci nehmotné

Konto

Ohodnocení

- události

Objednávka

Faktura

Dodávka

- pojmy

Norma

Spotřeba materiálu

Cosí [ještě nemám název]

Zobecnění

Entita

- objekt reálného světa
- jednoznačná identifikace
- rozlišení od ostatních cosi

- množina cosi téhož typu
- (či si podobných)
- majících stejné vlastnosti



Bob



Franta



Dana



Alice

Abstrakce



Anton



Galileo



Pepa



Osoba



Osoba

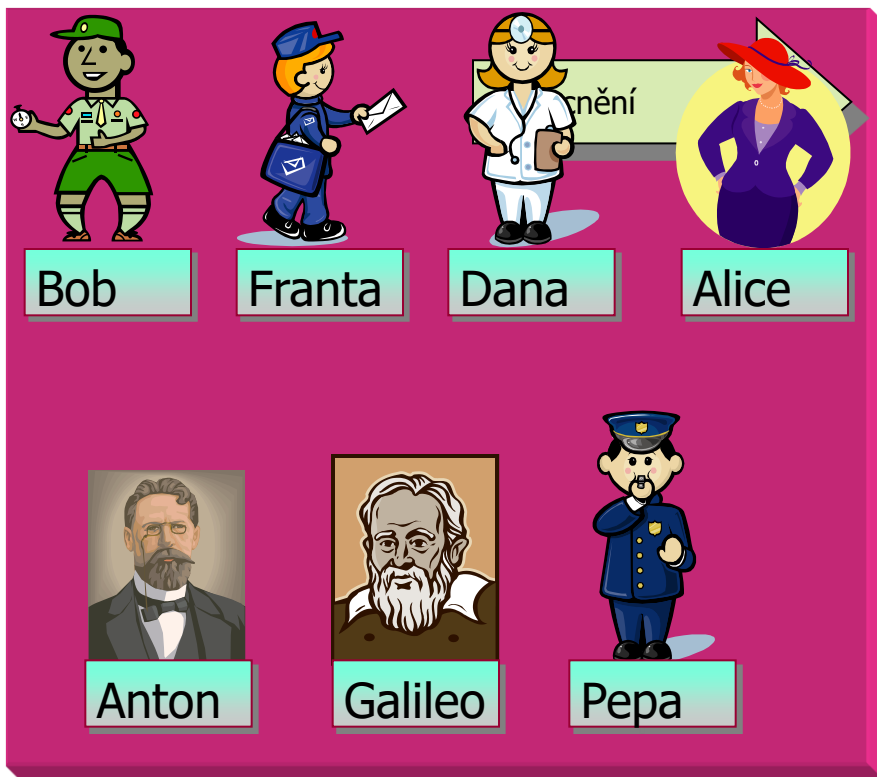
Cosi

Zobecnění

Entita

- objekt reálného světa
- jednoznačná identifikace
- rozlišení od ostatních entit

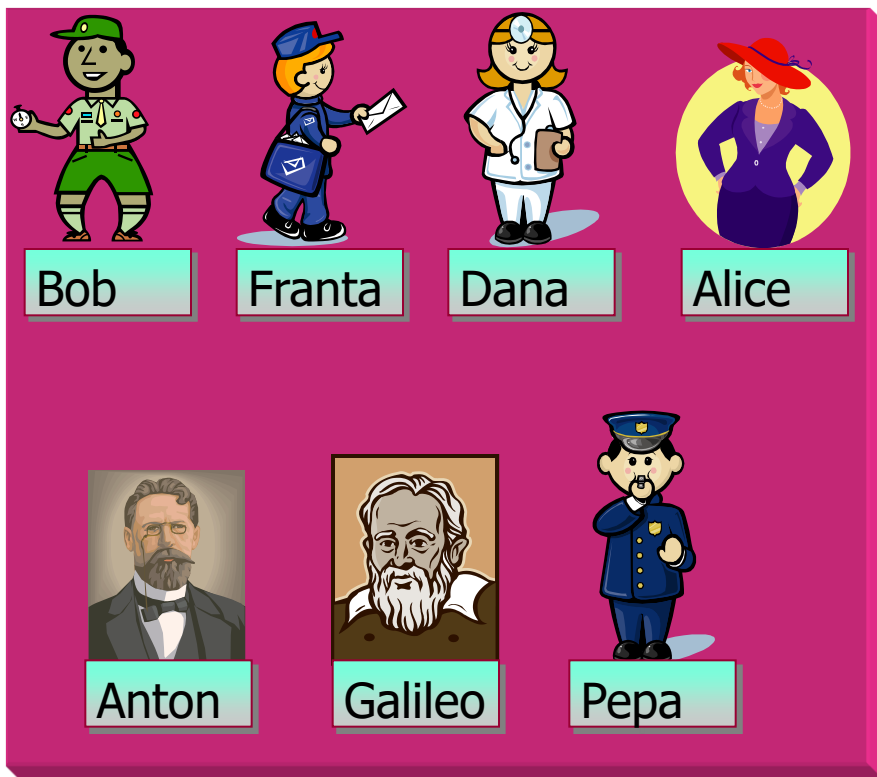
- množina cosi téhož typu
- (či si podobných)
- majících stejné vlastnosti



Instance - výskyt

Zobecnění

- objekt reálného světa
- jednoznačná identifikace
- rozlišení od ostatních entit

**Entitní množina**

- množina entit téhož typu
- (či si podobných)
- majících stejné vlastnosti

Zobecnění



Osoba

Abstrakce



Osoba



pane učiteli

zapoměl jste na diagramy

- vytváření E-R modelu - malůvky
 - ER diagram
- entita
 - **obdélník** s názvem entity
 - název entity - **podstatné jméno**

E-R model

Osoba

Auto

Budova

Kniha

Místnost

Mesto

Ulice

Nástroj

~~A~~~~ABC~~~~XYZ~~~~Alfa~~

proč ?

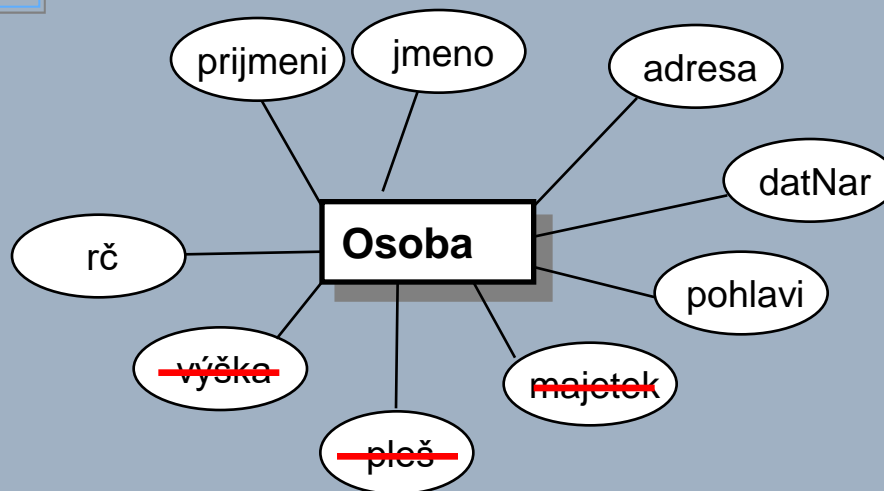
- entita - [množina stejných] objektů
 - má smysl podávat o nich informace
- nutno - skladovat hodnoty podstatných vlastností [entity]

Atributy

- podstatné vlastnosti entit a vztahů
 - nabývající hodnoty
- formálně jsou to funkce
- přiřazující entitám či vztahům
 - hodnotu nějaké podstatné vlastnosti
- název atributu - **podstatné jméno**

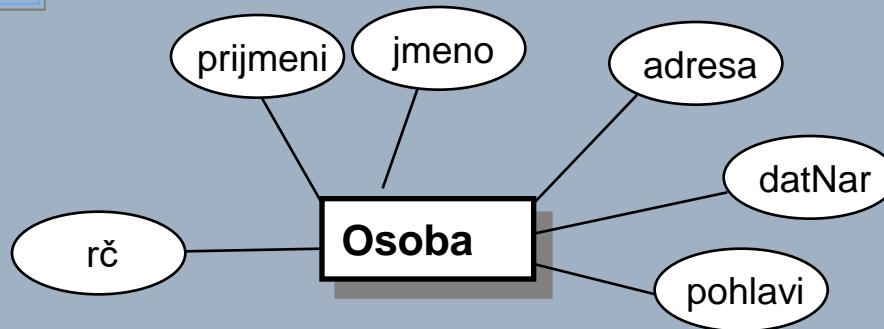
- pojmenovaný **ovál** spojený s entitou

E-R model



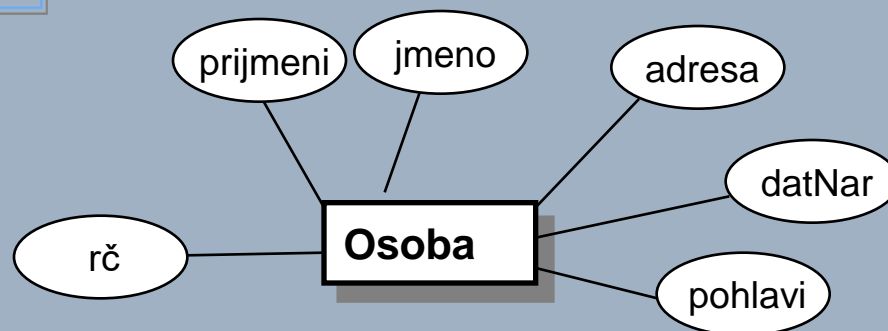
- pojmenovaný **ovál** spojený s entitou

E-R model



- každý výskyt entity musí být rozlišitelný - identifikovatelný
- identifikaci určuje Identifikátor - **Primární klíč**

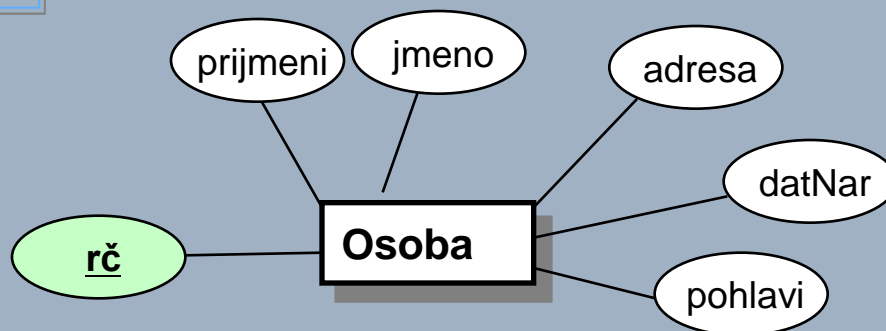
E-R model



- **primární klíč**- podtržený atribut

- každý výskyt entity musí být rozlišitelný - identifikovatelný
- identifikaci určuje Identifikátor - **Primární klíč**

E-R model



- **primární klíč**- podtržený atribut

A jak hodnotit **kvalitu** vlastností ???



- tedy
- jaké vlastnosti mají mít vlastnosti



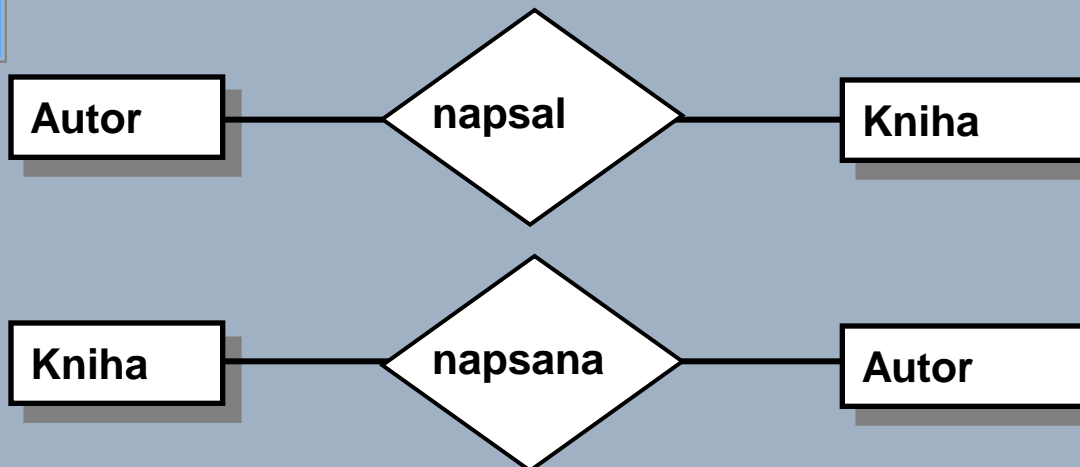
A jak hodnotit **kvalitu** vlastností ???



- **existence** (volitelnost)
 - 1 - povinný - totální (u všech instancí entity)
 - 0 - nepovinný - [ne u všech instancí,
ne v každém čase]
- **kardinalita** (kolik hodnot může vlastnost nabývat)
 - 1, 3, N [0, 1, 3, N]
- **doména**
 - obor hodnot
 - textové, číselné, datumové, ...

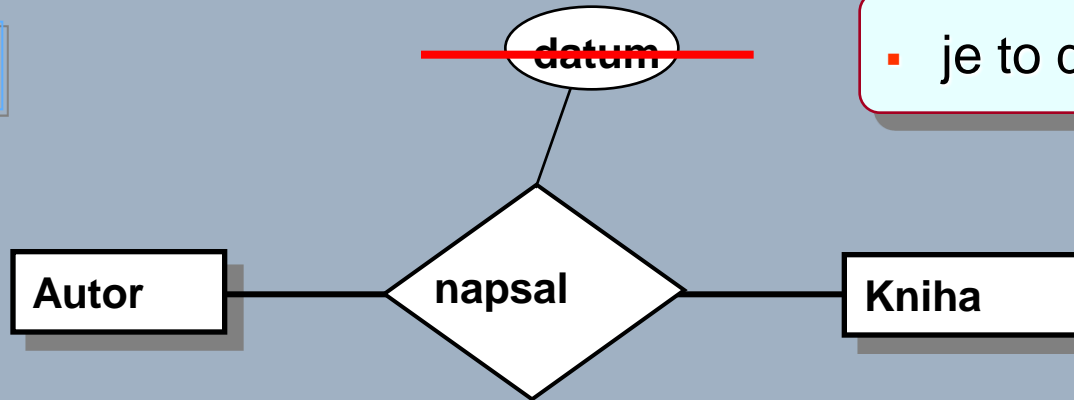
- mezi entitami mohou existovat vztahy
- vztahy je nutné v modelu rozeznat a označit
- vztah v diagramu
 - kosočtverec
 - názvem je sloveso / v trpném rodu

E-R model



- vztah může mít vlastnost - atribut

E-R model

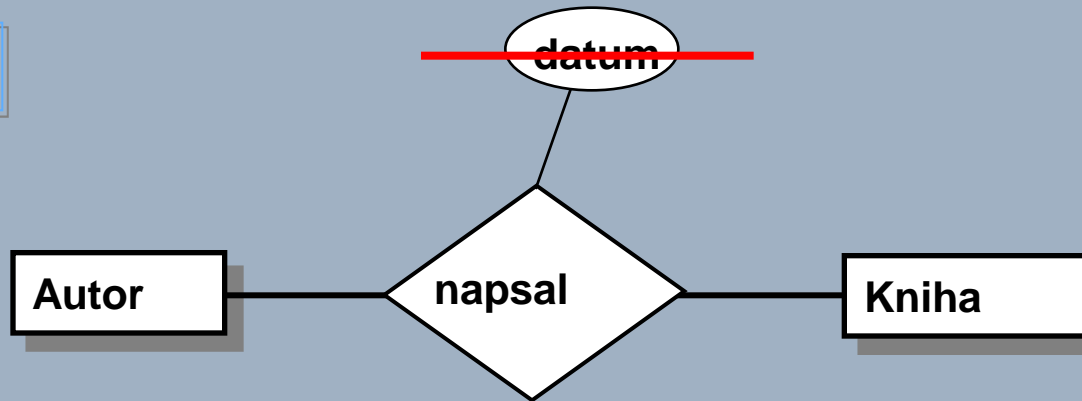


- je to dobře ?



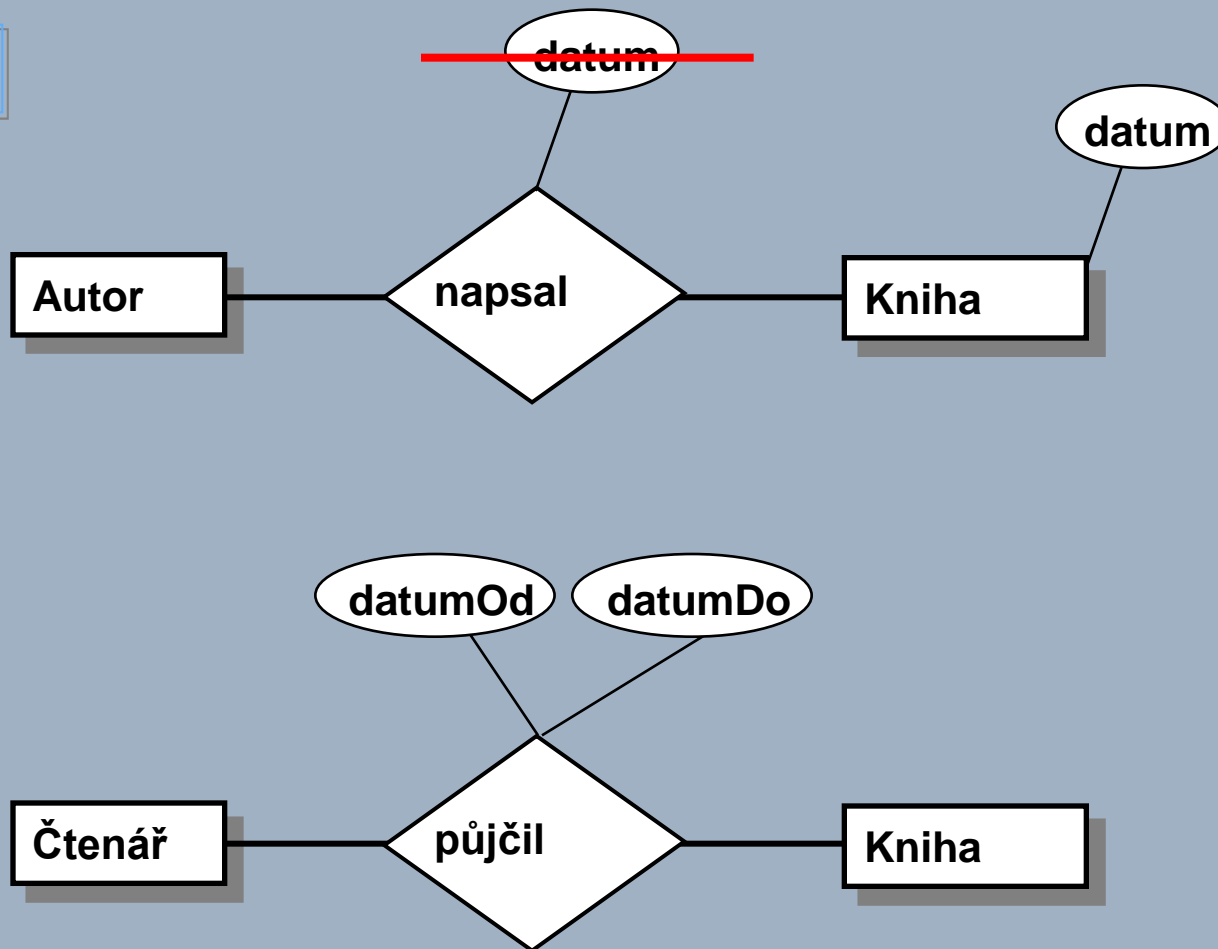
- vztah může mít vlastnost - atribut

E-R model



- vztah může mít vlastnost - atribut

E-R model



- existuje nezávisle na jiných entitách



Osoba



Osoba

- klíč je tvořen vždy pouze z vlastních atributů



Bob



Franta



Dana



Alice



Anton



Galileo

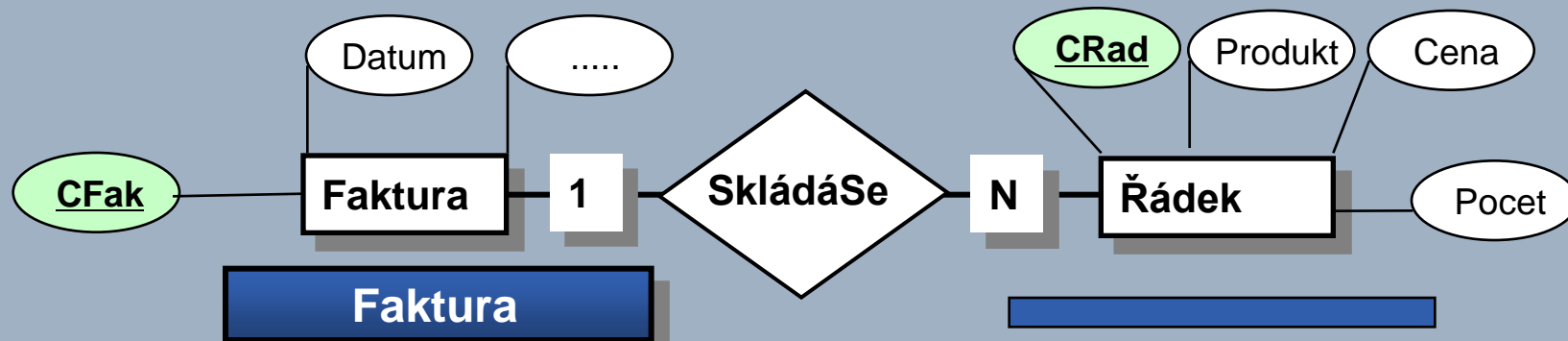


Pepa

- neexistuje beze své rodičovské entity
- pokud se ukončí rodičovská entita
 - ukončí se i slabá entita
 - příkladem budiž **doklad**

Faktura		

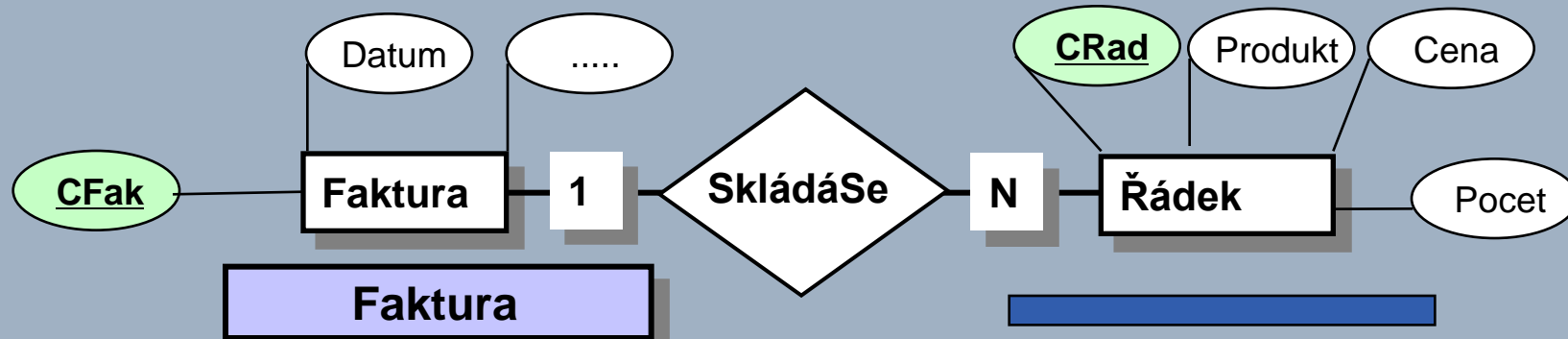
E-R model



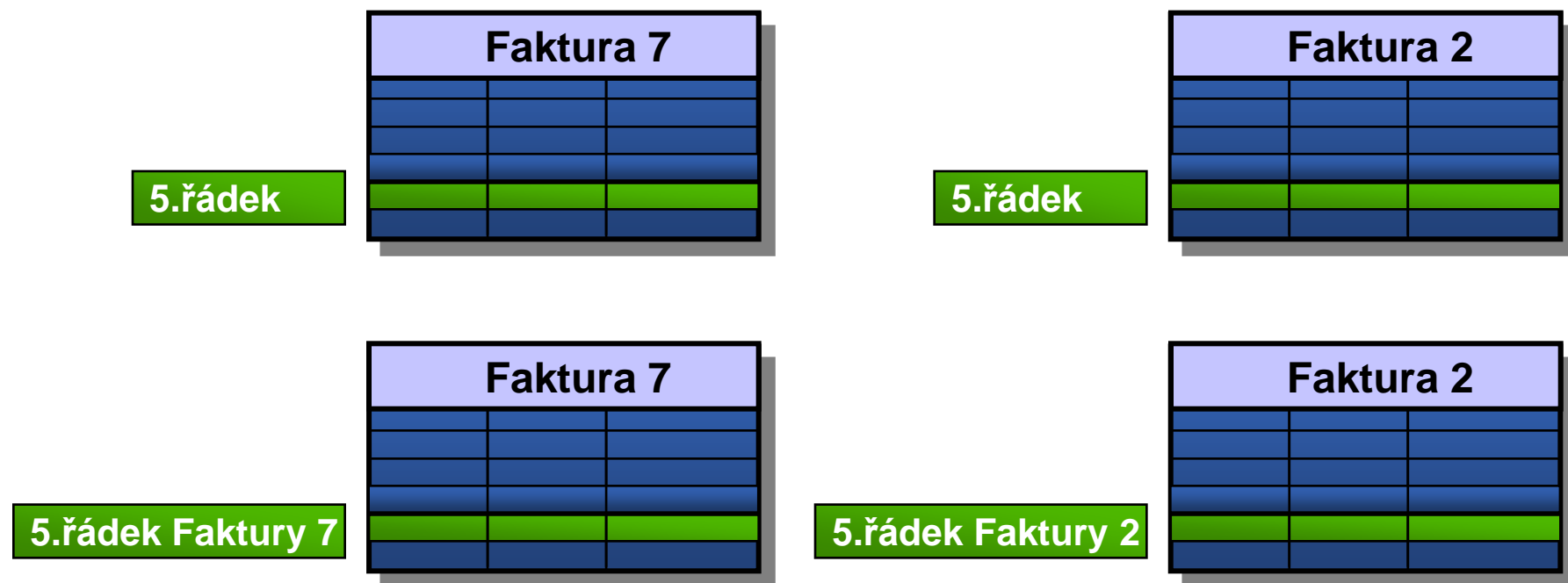
- neexistuje beze své rodičovské entity
- pokud se ukončí rodičovská entita
 - ukončí se i slabá entita
 - příkladem budiž **doklad**

Faktura		

E-R model



- nelze rozlišit dvě instance dle vlastních atributů
- rozlišuje se členstvím ve vztahu **identifikačním vlastníkem**



Primární klíč

=

dominantní_identifikátor

+

diskriminátor

identifikační vlastník

Faktura 7

5.řádek

- všechny vlastní atributy nestačí pro identifikaci instance
- primární klíč [řádku] je tvořen vždy jako složený
 - primární klíč rodičovské entity (identifikační vlastník)
 - unikátní identifikace uvnitř rodičovské entity

Primární klíč

=

CFak

+

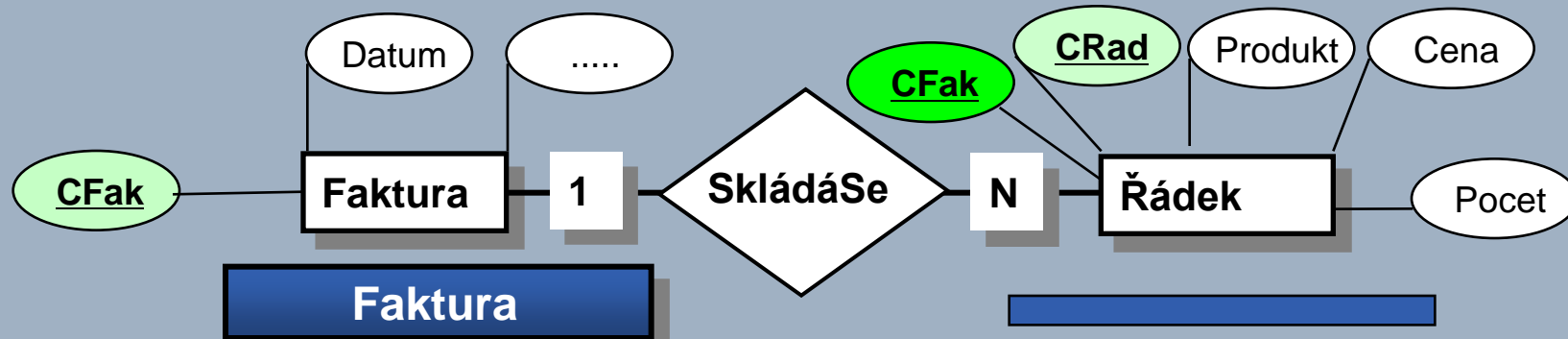
CRad

dominantní_identifikátor

+

diskriminátor

E-R model





A dejme úkol

- učiňte konceptuální model pro IS

■ Knihkupectví

- [taky mohou knihy trochu půjčovat]
- číňte

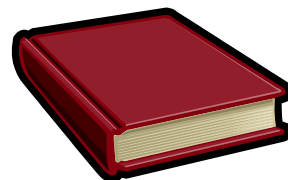
■ řešte

- pane učiteli
- já bysem si zamodeloval



- učiňte konceptuální model pro IS

■ Knihkupectví



- číňte



- řešte

- pane učitelé
- já bych si zaměřil

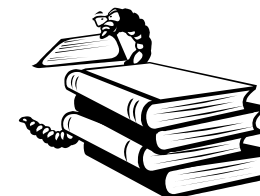
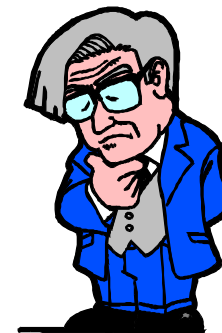


Milí studenti - studentky nevyjímaje
zatrenujeme si E-R modelování.

Navrhněte Entity a vztahy [a později i atributy] pro

Knihkupectví

[i malá půjčovna]





Knihkupectví



E-R model

Knihkupectví

Autor

NazevKnihy

Kniha

Vydání

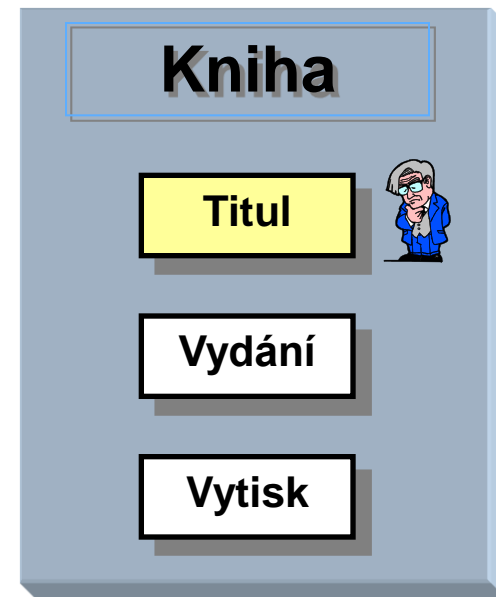
Výtisk

Vydavatel

~~ABC~~

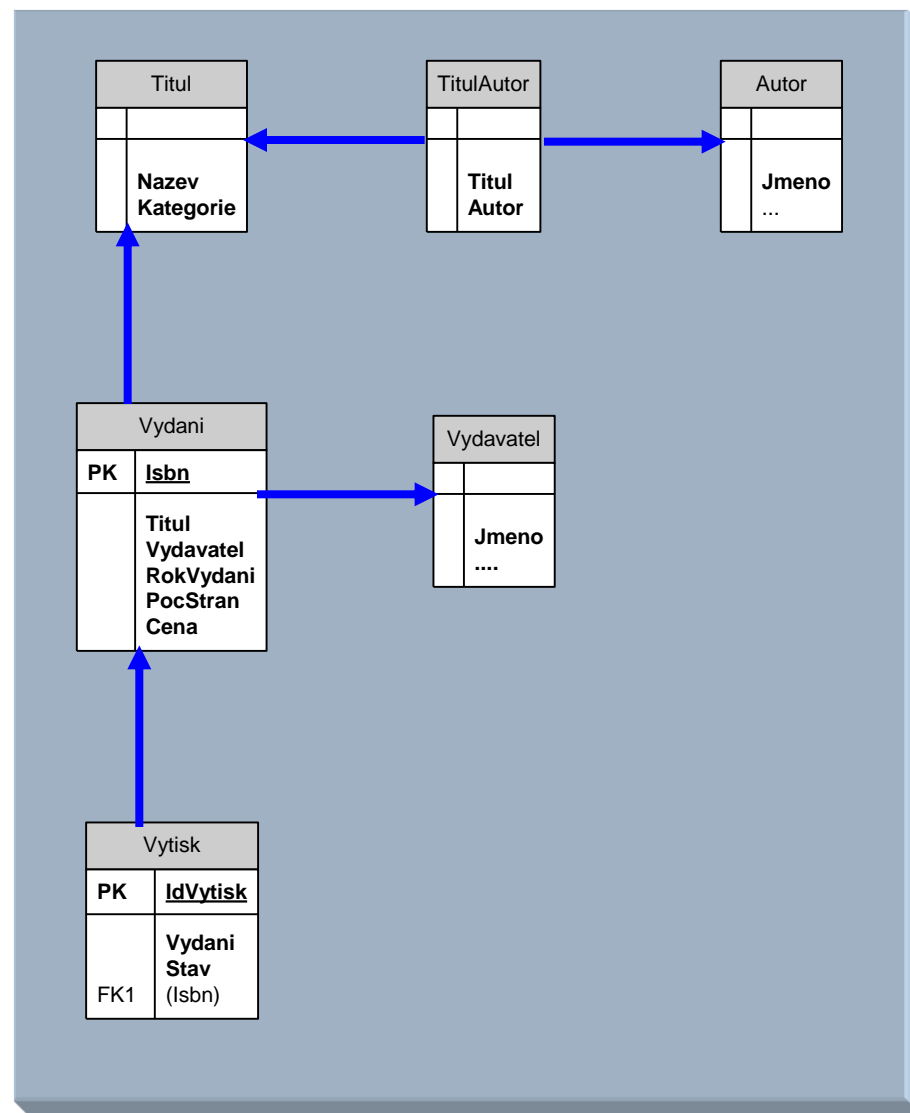


■ co je toto



- mírné tajemství projektování jest
- jednoduché
- věci správně pojmenovat





Knih**Titul**

Titul	
	Nazev Kategorie

TitulAutor	
	Titul Autor

Autor	
	Jmeno ...

- a co je to pane učiteli za tajnost

- ISBN ?



PK	<u>IdVytisk</u>
FK1	Vydani Stav (Isbn)



a zas něco jiného

- uspořádaná množina [v DB definovaných] informací
- spolu s SW programem pro přístup a manipulaci s daty

DBS =

Data + Program

[manipulace s daty]

- a lze ty DBS rozdělit
- podle nějakých typů
- [nějaký škatule DBS]



- uspořádaná množina [v DB definovaných] informací
- spolu s SW programem pro přístup a manipulaci s daty

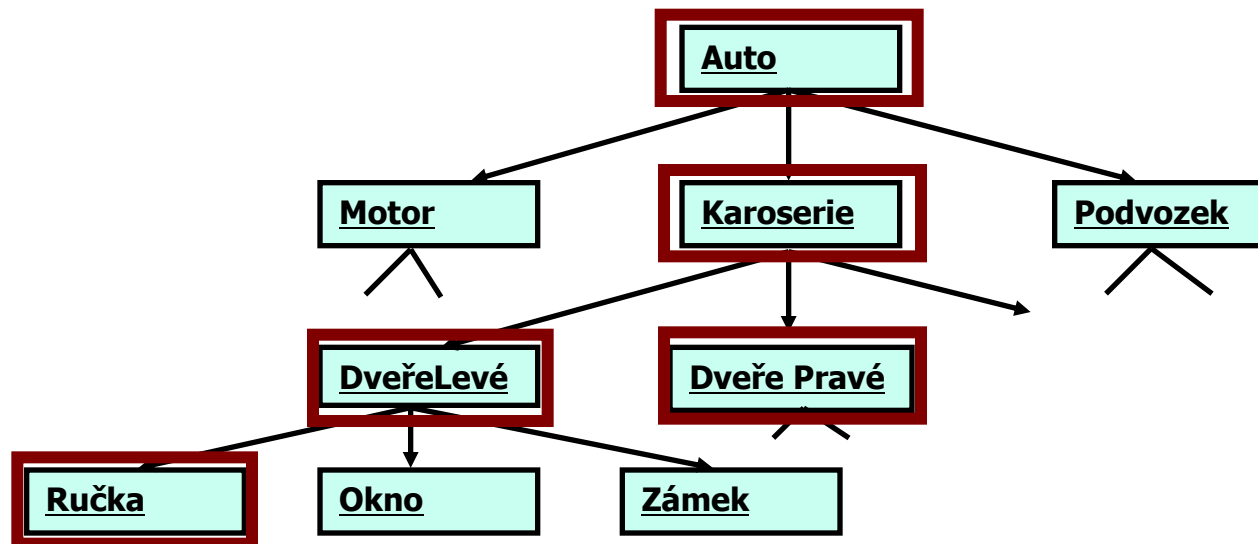
Struktura dat a technika přístupu k datům

- databázový model DB

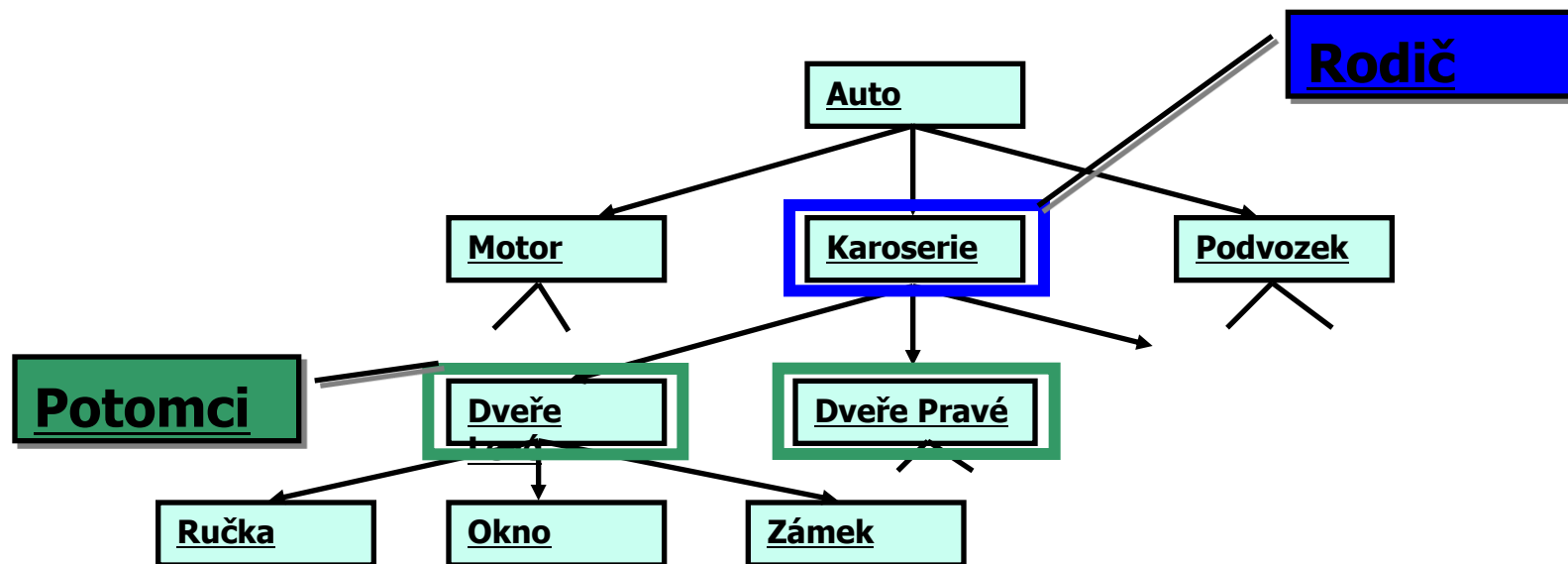
Databázové modely

[typy databází]

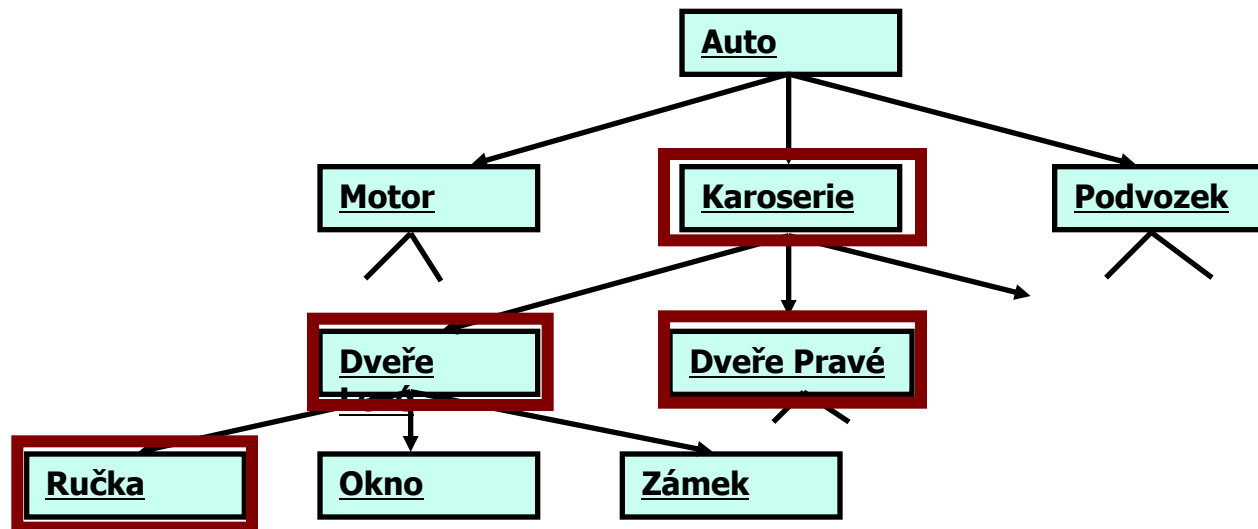
- Hierarchická
- Síťová
- **Relační**
- Objektová
- Objektově relační



- data jsou organizována do stromové struktury
- každý záznam je uzlem ve stromové struktuře
- vzájemný vztah mezi záznamy označujeme
 - rodič a potomek

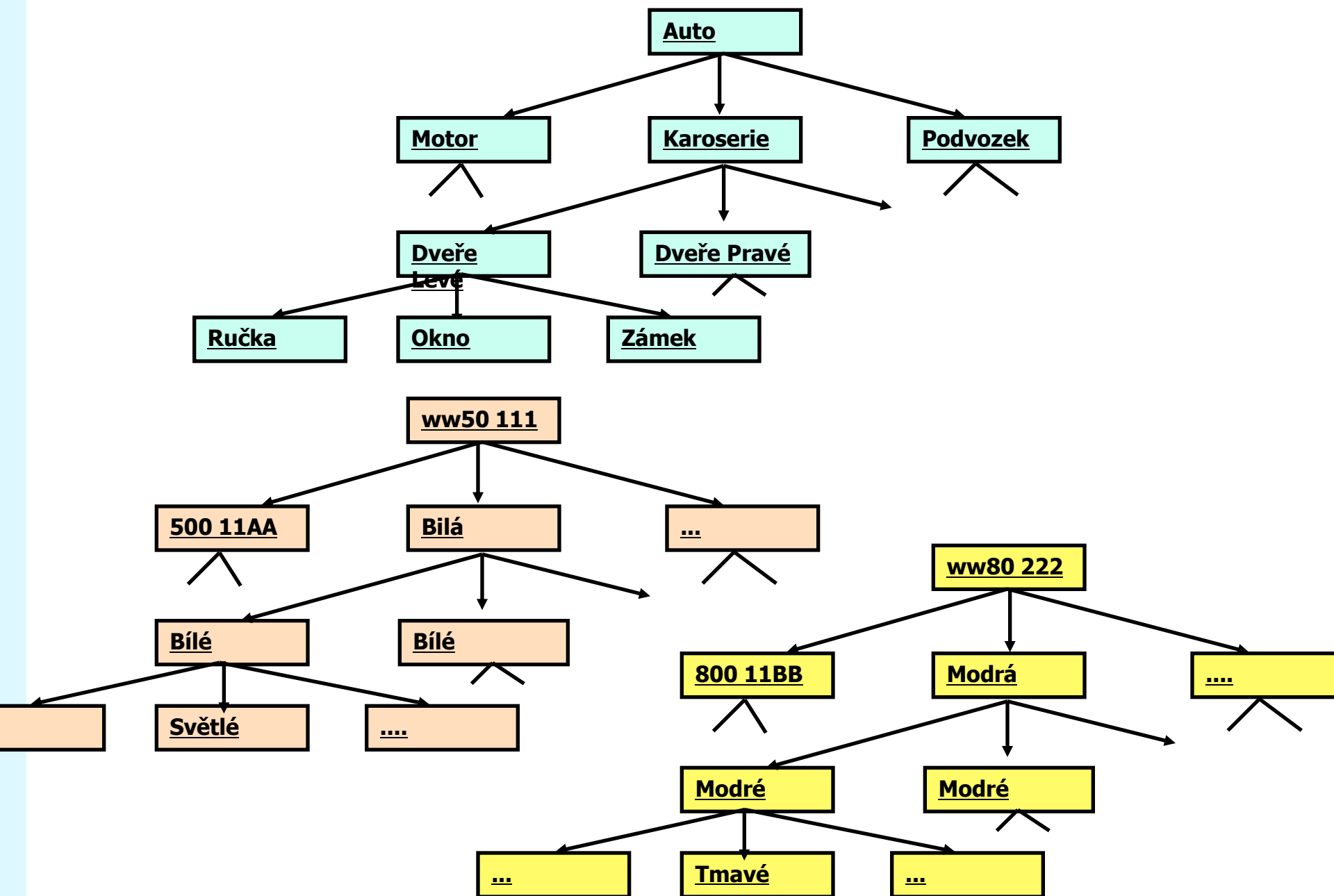


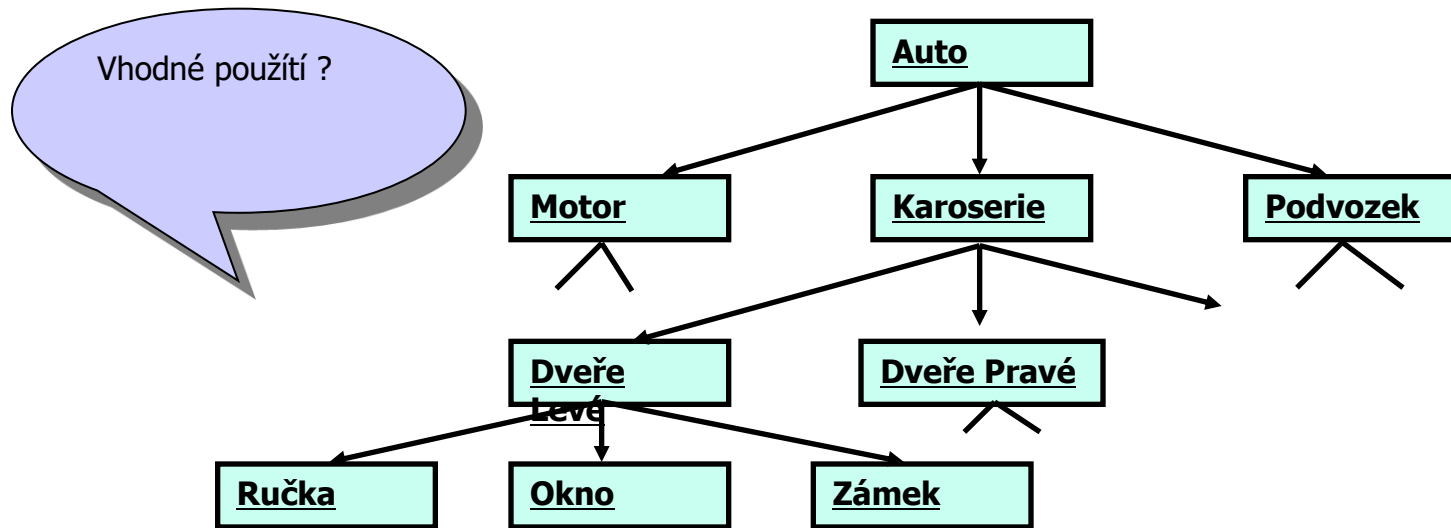
- data jsou organizována do stromové struktury
- každý záznam je uzlem ve stromové struktuře
- vzájemný vztah mezi záznamy označujeme
 - rodič a potomek
- **rodič** má 0 .. N potomků
- **potomek** má právě jednoho rodiče



Možné operace

- najít určitou součástku **dle čísla** (levé dveře)
- posunout se směrem dolů na prvního potomka (ručka)
- posunout se směrem nahoru na rodiče (karoserie)
- posunout se směrem do boku na sourozence (dveře pravé)

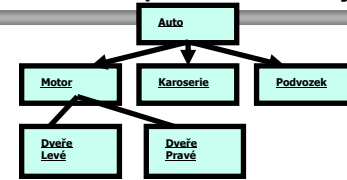




Kusovník produktu

- přirozená stromová struktura
- vyhledávání dat - navigace mezi záznamy
- po jednom na všechny strany
jednoduchá struktura
- výborný výkon
- uspořádání Rodič-Potomek

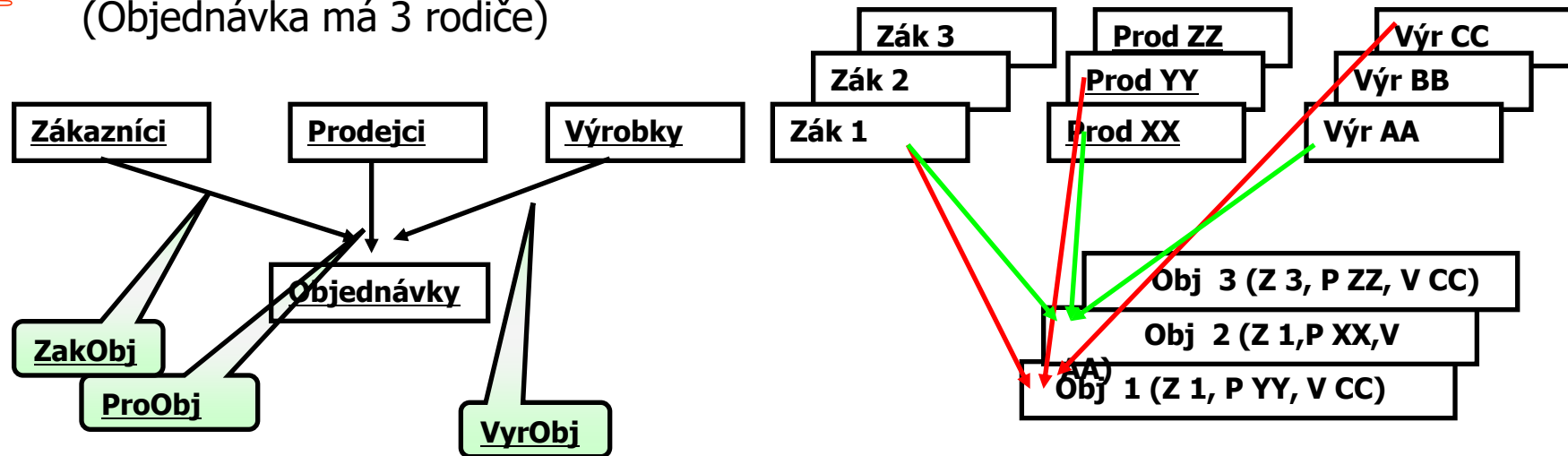
Hierarchický model



- data jsou organizována do stromové struktury
- každý záznam je uzlem ve stromové struktuře
- vzájemný vztah mezi záznamy označujeme
 - rodič a potomek
- struktura záznamů definována ukazateli (pointery)
- ukazatelem je fyzická adresa závislého záznamu
- záznam (nadřazený) může mít 1..více potomků
- záznam (podřízený) má jednoho rodiče
- neustaven **žádný standard**
- IMS (IBM) - nejznámější produkt

Hierarchická DB nevyhovuje:

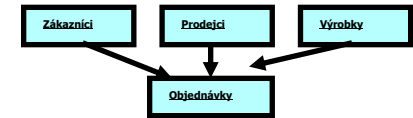
- 👉 když data mají složitější strukturu
- 👉 zvláště, když se jedná o t.zv. vatahy N:M
- 👉 př: spojení objednávky se zákazníkem, prodejcem a výrobkem.
- 👉 (Objednávka má 3 rodiče)



Objednávka má několik vztahů k rodičům - sady

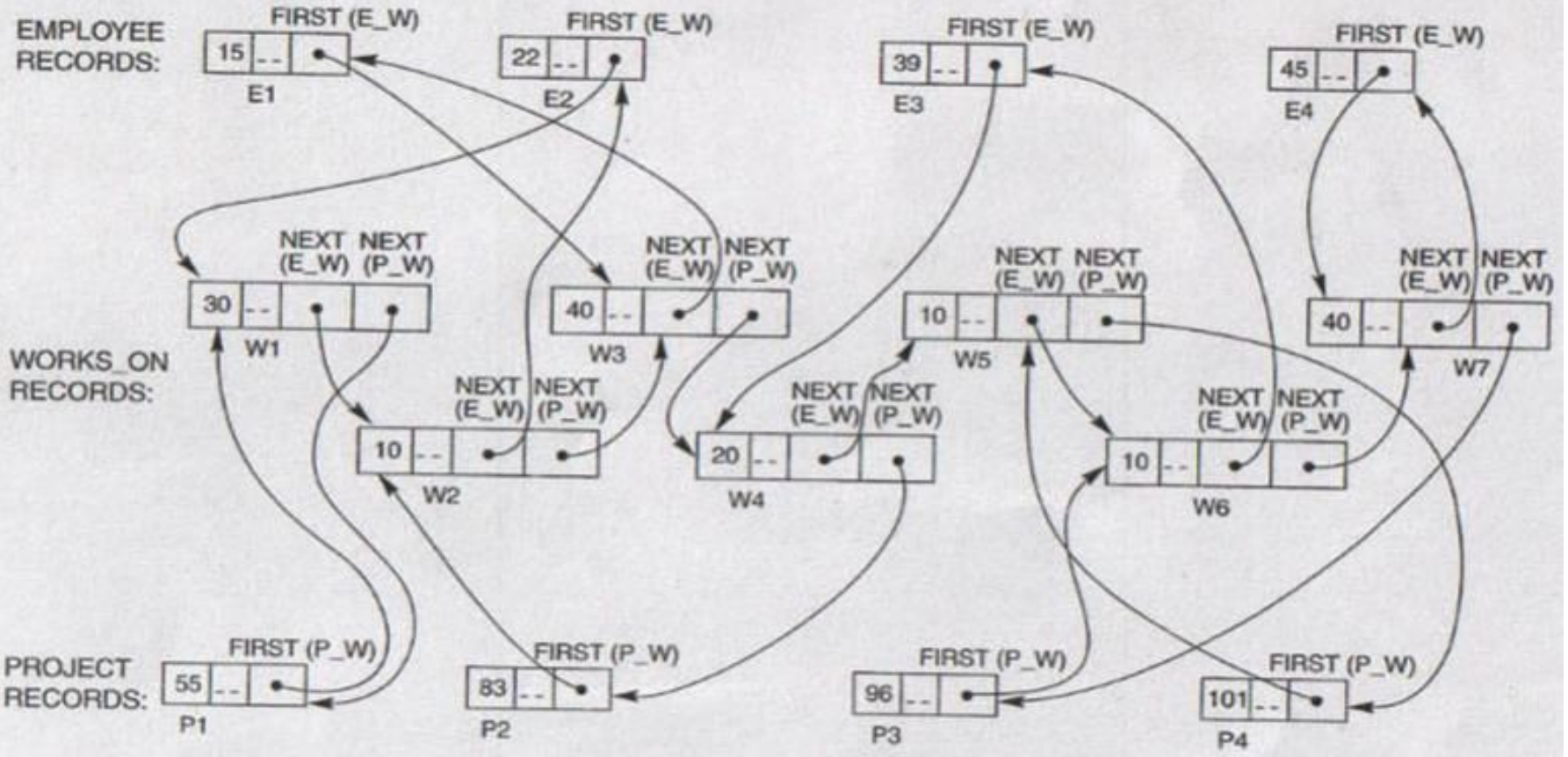
Možné operace:

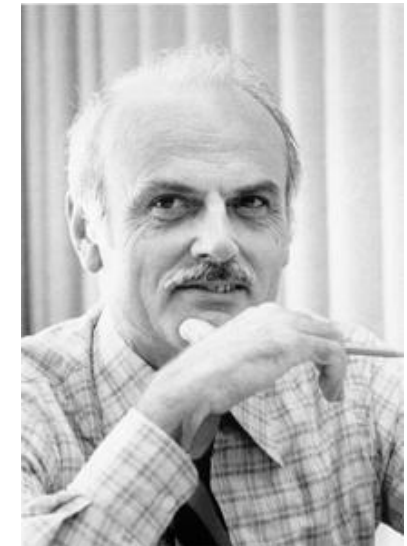
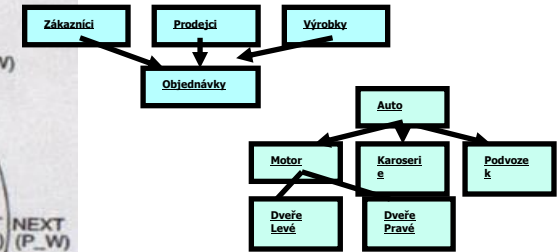
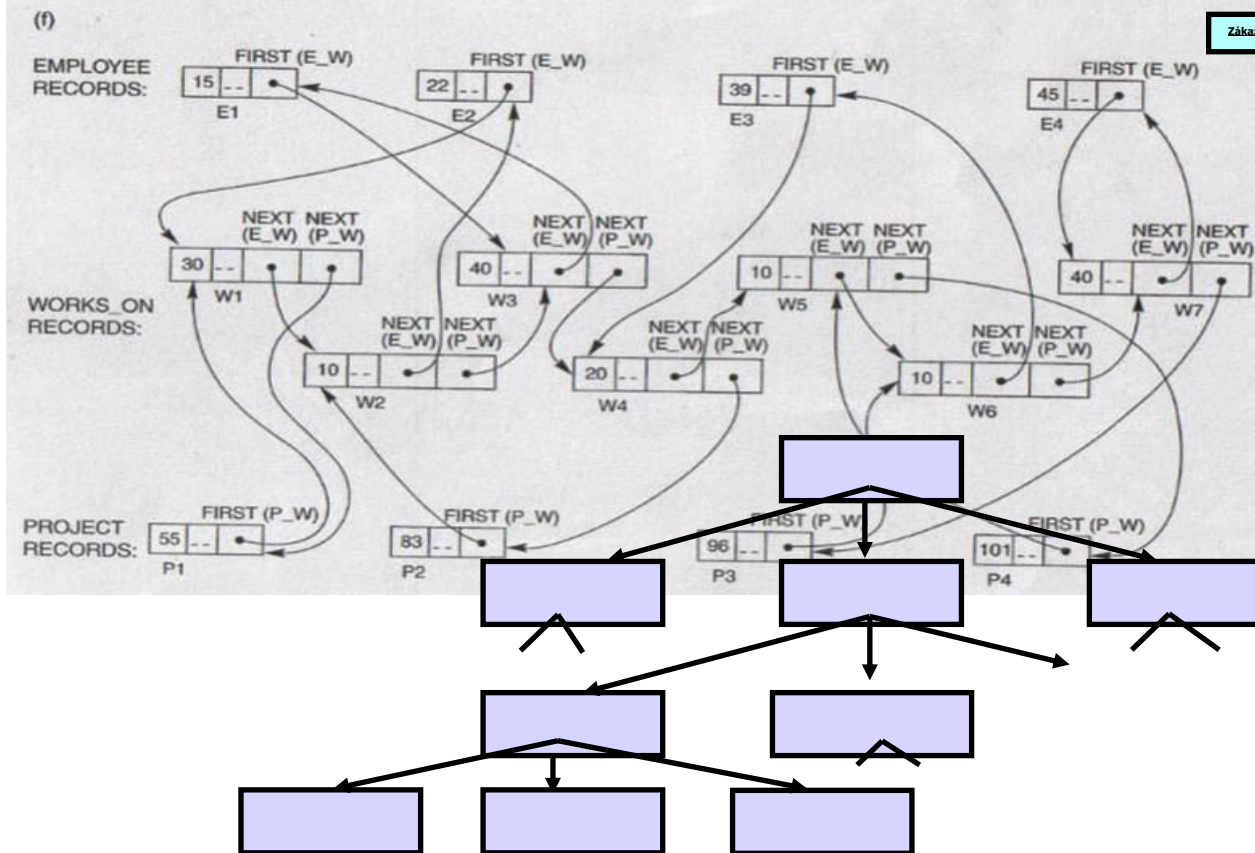
- 👉 najít určitý záznam přímo dle klíče (číslo zákazníka)
- 👉 posunout se směrem dolů na prvního potomka v sadě (Zak-Obj) tj první objednávku daného zákazníka
- 👉 posunout se v sadě stranou (druhá objednávka tohoto zákazníka)
- 👉 posunout se nahoru v jiné sadě (Prod-Obj) na prodejce této objednávky



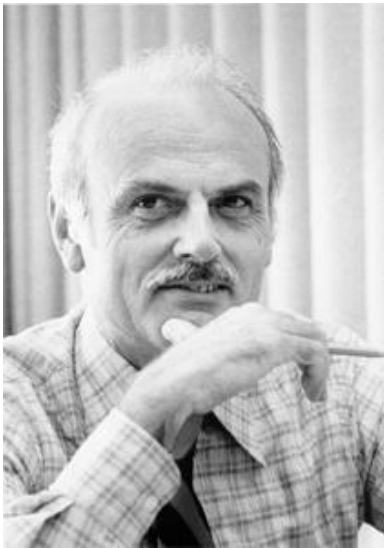
- zobecnění hierarchického modelu
- vztah mezi rodičovským záznamem a potomkem - ukazatel
- potomek může mít několik rodičů
 - účastní se několika vztahů
- flexibilnější (model)
- vyšší složitost
- velmi nákladná údržba modelu
- 1972 standardizován (DBTG) v rámci CODASYL
- IDMS - nejznámější produkt (na různých počítačích)

(f)





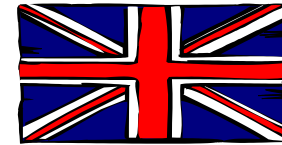
- teorie i technologie DB - programátoři, informatici
- do DB oblasti vstupuje genius
- algebraik **Ted Codd**



Narozen: 1923

Zemřel: 2003

Zaměstnán: IBM



1970 "A Relational Model of Data for Large Shared Data Banks"

1973 začátek projektu SYSTEM R

1985 **12 Coddových pravidel pro RDB**

- Zaměstnanec IBM
- IBM se přihodilo zlaté vejce
- nepochopila to [nevšimli si]
 - intelektuální kuriozita - nerealizovatelná
 - konkurence tehdejšího IBM DB - IMS
- zbohltli jiní (**Oracle**)



- 1970 Ted Codd

A Relational Model of Data for Large Shared Data Banks

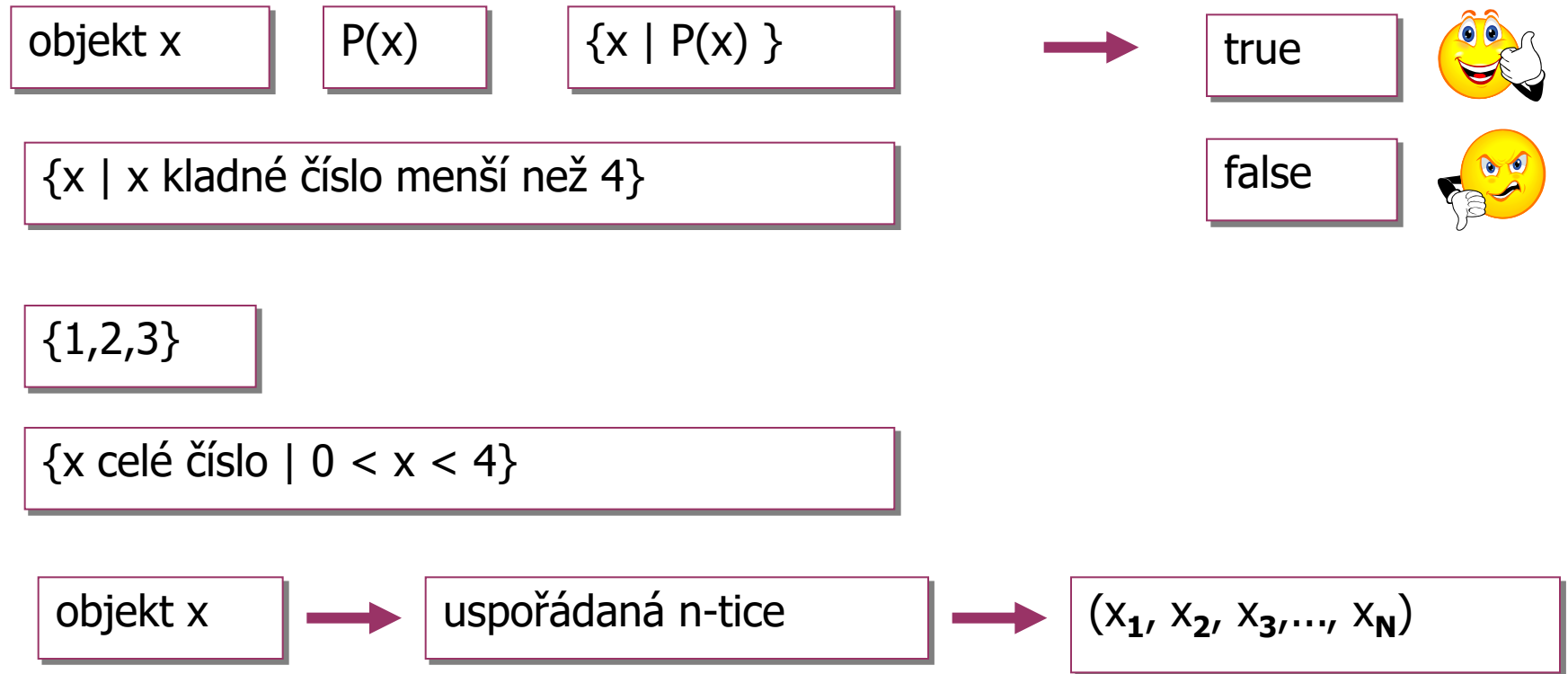
- 1972 University of California at Berkeley (INGRES) začíná
- 1974 IBM - začátek Systém/R
- 1974 První publikace dotazovacího jazyka SEQUEL
- 1978 Testy Systém/R
- 1979 první komerční systém ORACLE (Larry Ellison, System-R)
- 1981 INGRES (1996->PostgreSQL)
- 1981 IBM SQL/DS
- 1982 ANSI zahajuje standardizaci SQL - DB
- 1983 IBM ohlášení DB2

- **Coddův teorém**
- **Coddův relační kalkul**
 - matematický aparátu relačních množin
 - predikátová logiky
- množina
- potřeba množinu popsat
- uvedení seznamu jejích prvků
- určit vlastnost platnou pro všechny prvky - předpis

objekt x

$P(x)$

$\{x \mid P(x)\}$

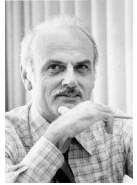


Pragmatická definice relační databáze

- databáze
- všechna data uspořádána do tabulek

Codd's Twelve Commandments

- 0. pravidlo
- 1. pravidlo
- 2. pravidlo
- 3. pravidlo
- 4. pravidlo
- 5. pravidlo
- 6. pravidlo
- 7. pravidlo
- 8. pravidlo
- 9. pravidlo
- 10. pravidlo
- 11. pravidlo
- 12. pravidlo



▪ to těch pravidel je však však 13 !!!!!!!!, pane učiteli



E-R model

Osoba

Auto

Budova

Kniha

Místnost

Mesto

Ulice

Nástroj

Osoba		

Auto		

Budova		

Kniha		

Místnost		

Mesto		

Ulice		

Nástroj		

E-R model

Osoba

Auto

Budova

Kniha

Místnost

Mesto

Ulice

Nástroj

Osoba		

Auto		

Budova		

Kniha		

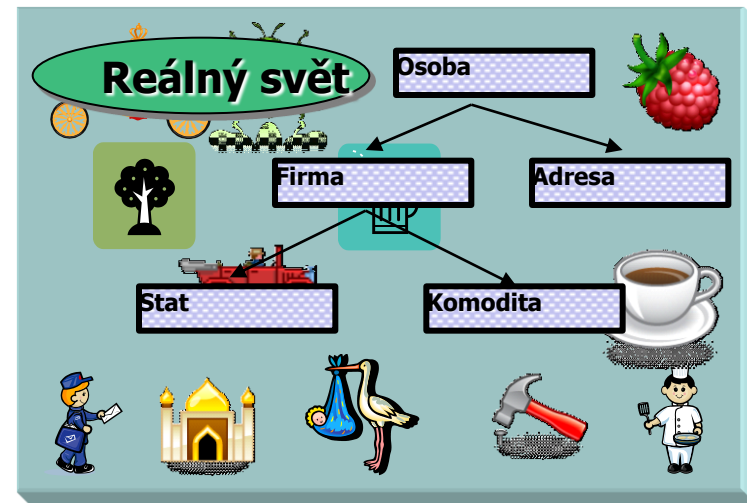
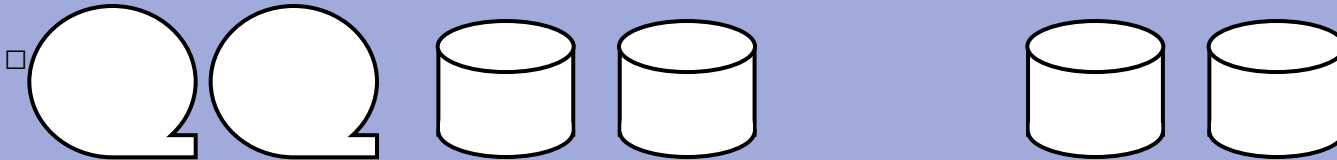
Místnost		

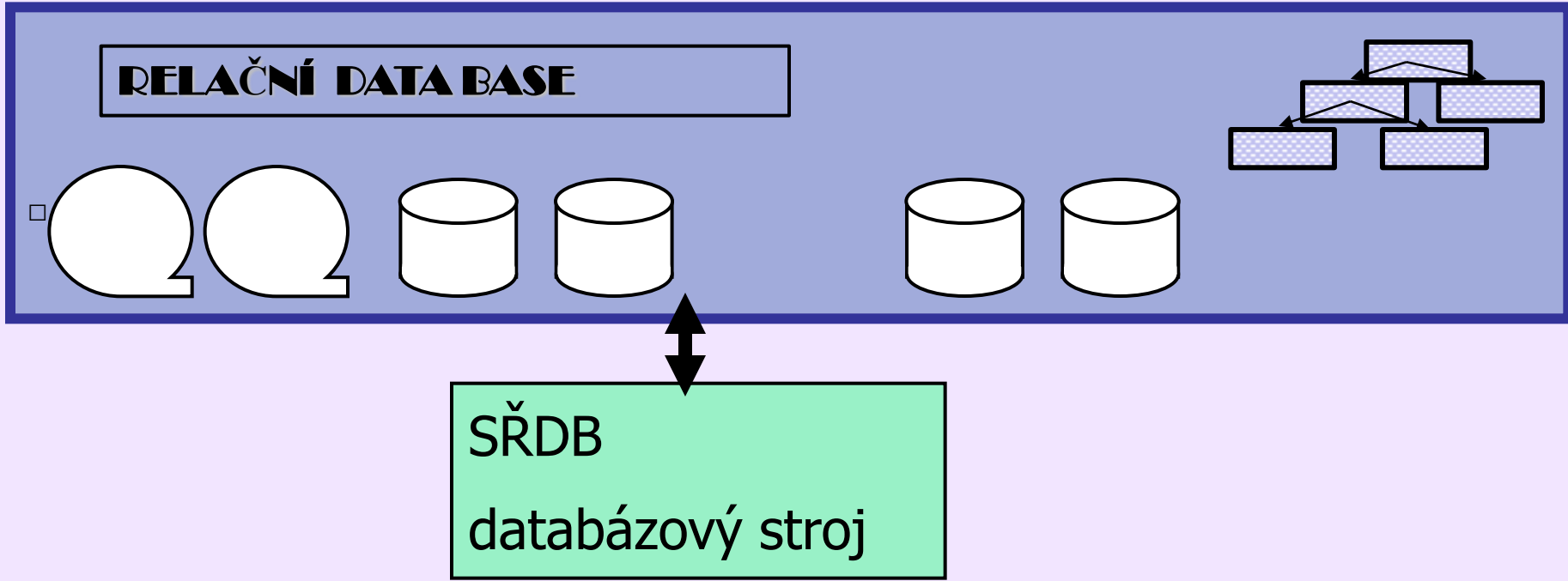
Mesto		

Ulice		

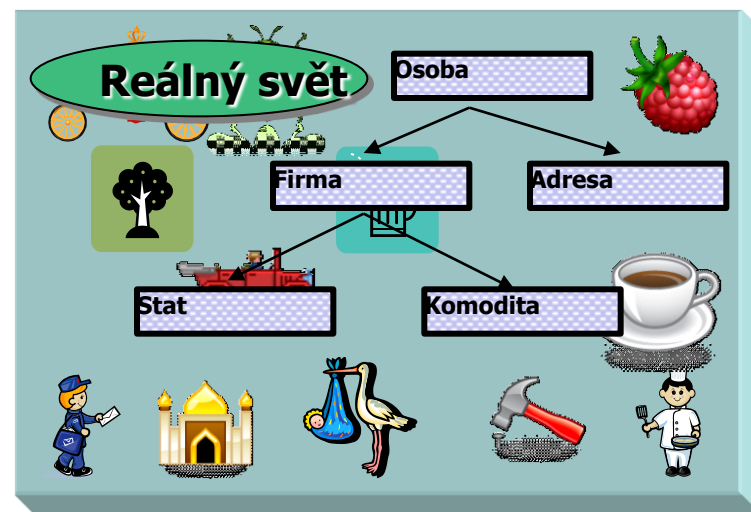
Nástroj		

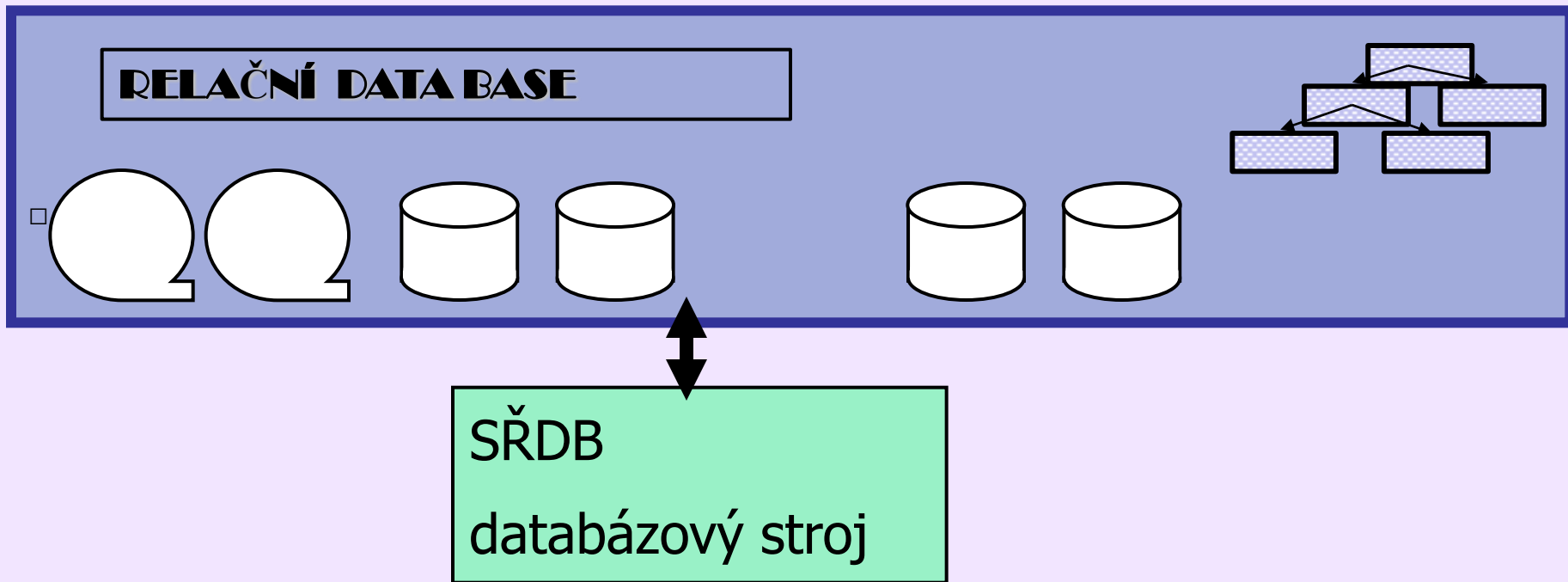
RELAČNÍ DATA BASE



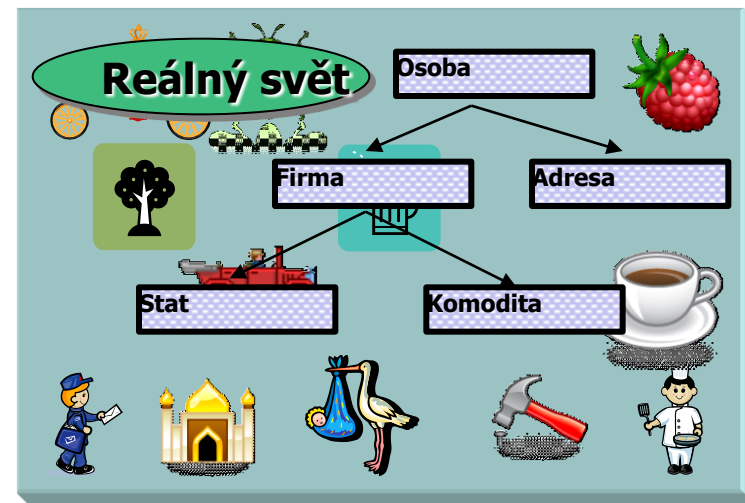


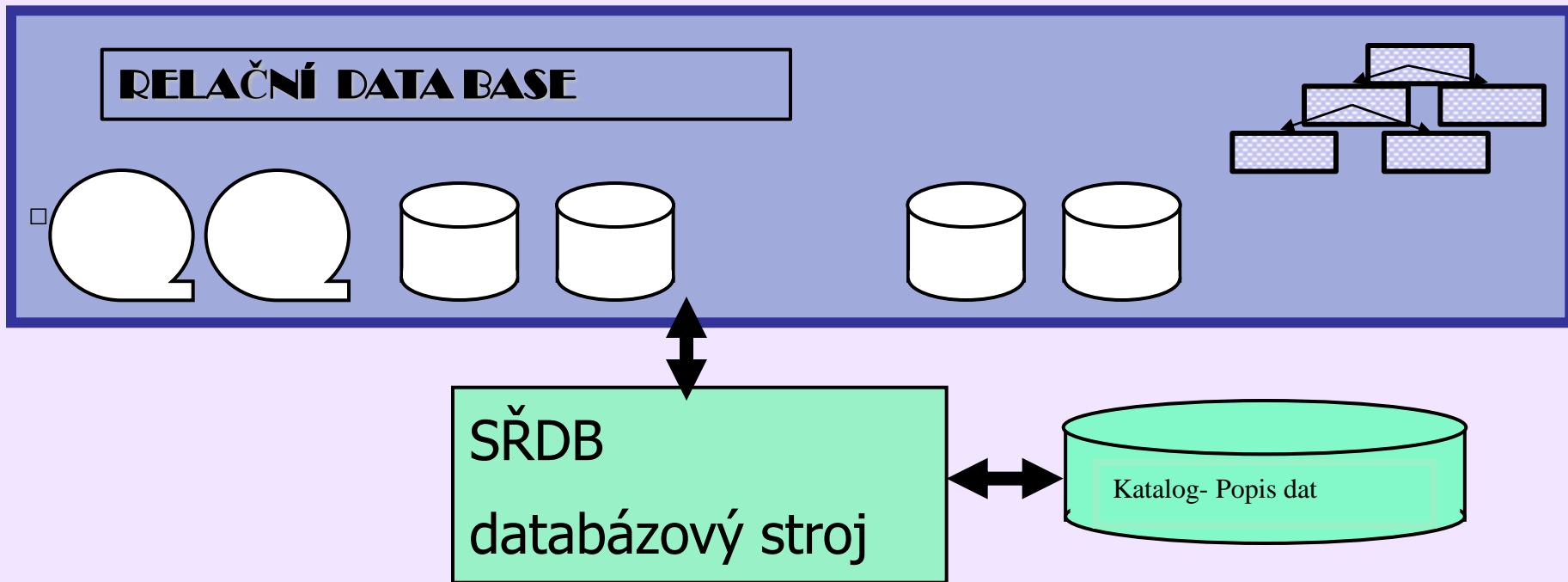
Relační DB



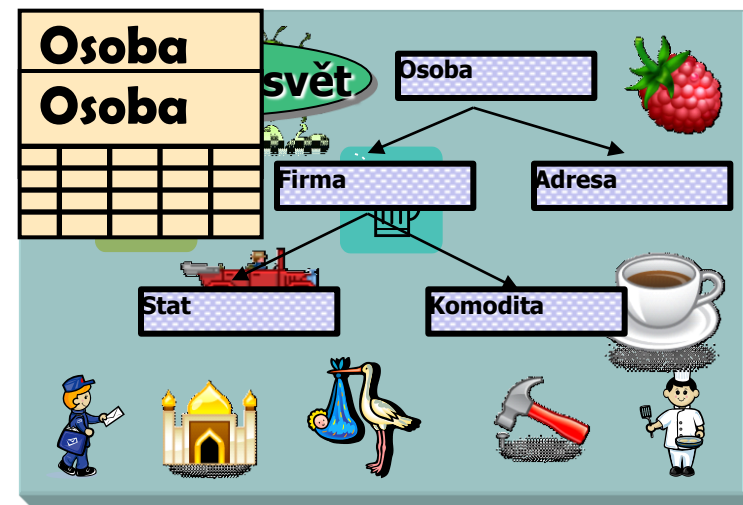


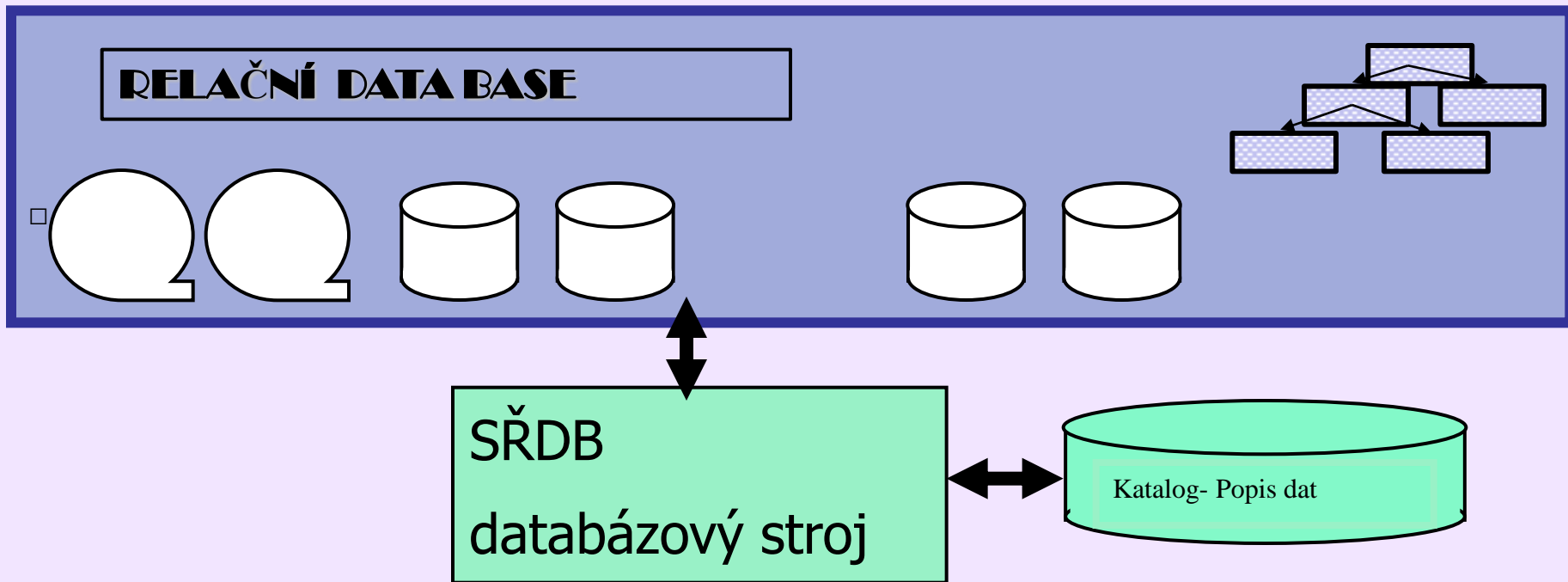
- každý objekt reálného světa
 - je modelován jako tabulka
- všechna data DB
 - jsou ve tvaru tabulek



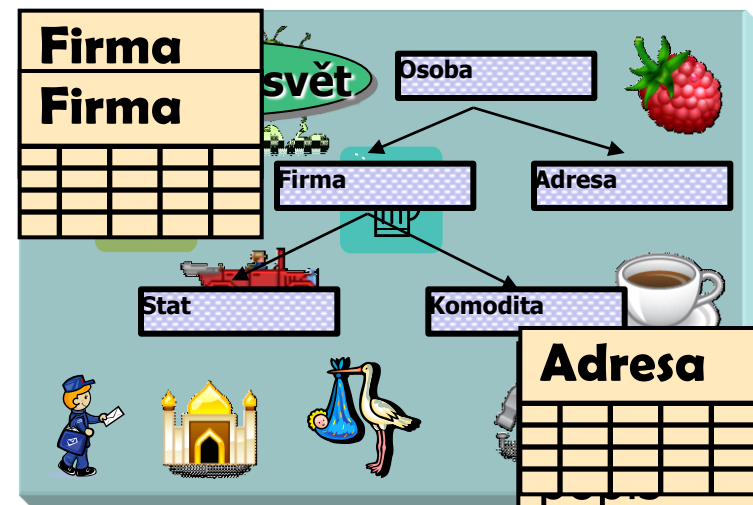


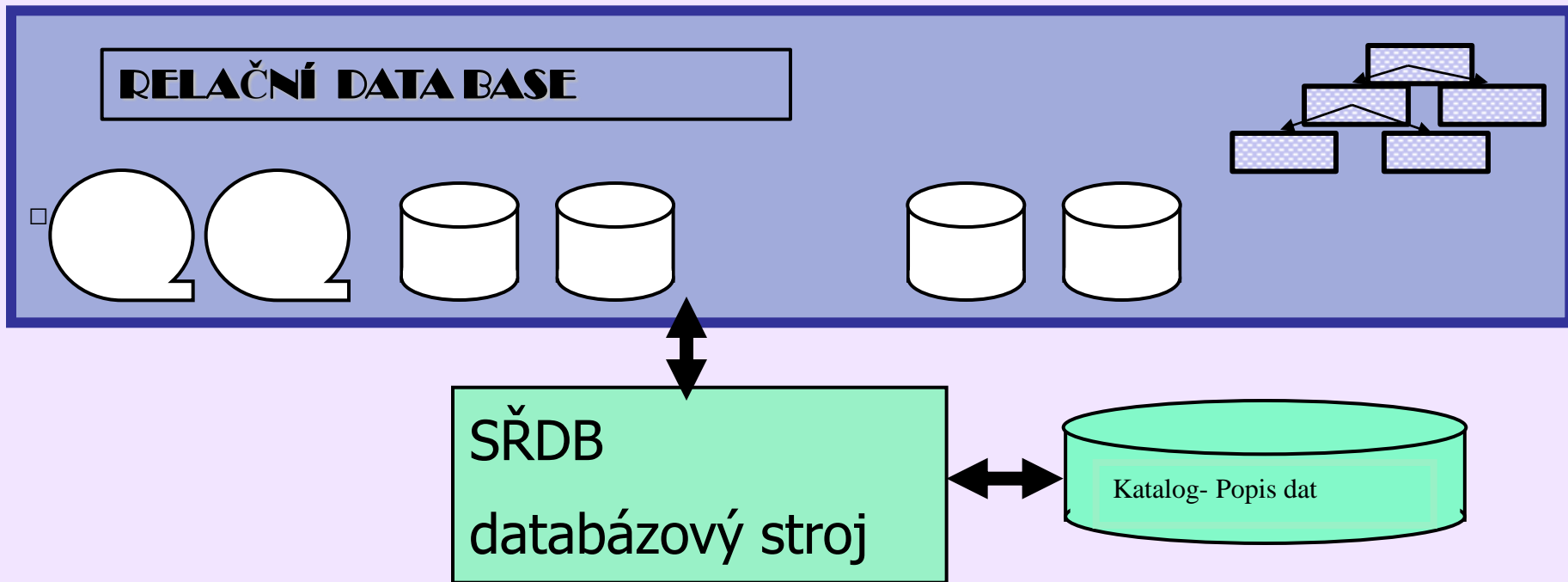
- každý objekt reálného světa
 - je modelován jako tabulka
- všechna data v DB
 - jsou ve tvaru tabulek



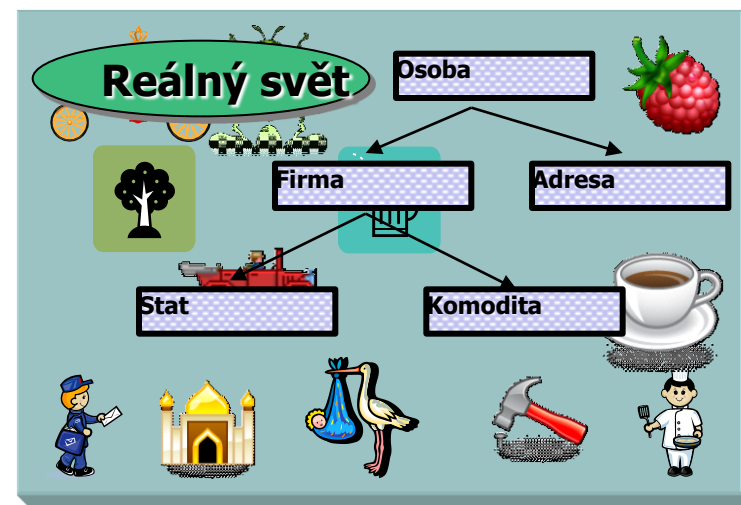


- každý objekt reálného světa
 - je modelován jako tabulka
- všechna data v DB
 - jsou ve tvaru tabulek





- jména tabulek musí být unikátní
- dvě různé tabulky nemohou mít stejné jméno



- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Sít'ové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Síťové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu

stromu

- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Síťové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu stromu

teoretický základ

žádný [všeobecně přijímaný]

- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Síťové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu stromu

teoretický základ

žádný [všeobecně přijímaný]

dotazovací [manipulační] jazyk

žádný

- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Síťové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu stromu

teoretický základ

žádný [všeobecně přijímaný]

dotazovací [manipulační] jazyk

žádný

povely [tudíž]

žádné

- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Síťové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu stromu

teoretický základ

žádný [všeobecně přijímaný]

dotazovací [manipulační] jazyk

žádný

povely [tudíž]

žádné

realizace [každého] dotazu

program

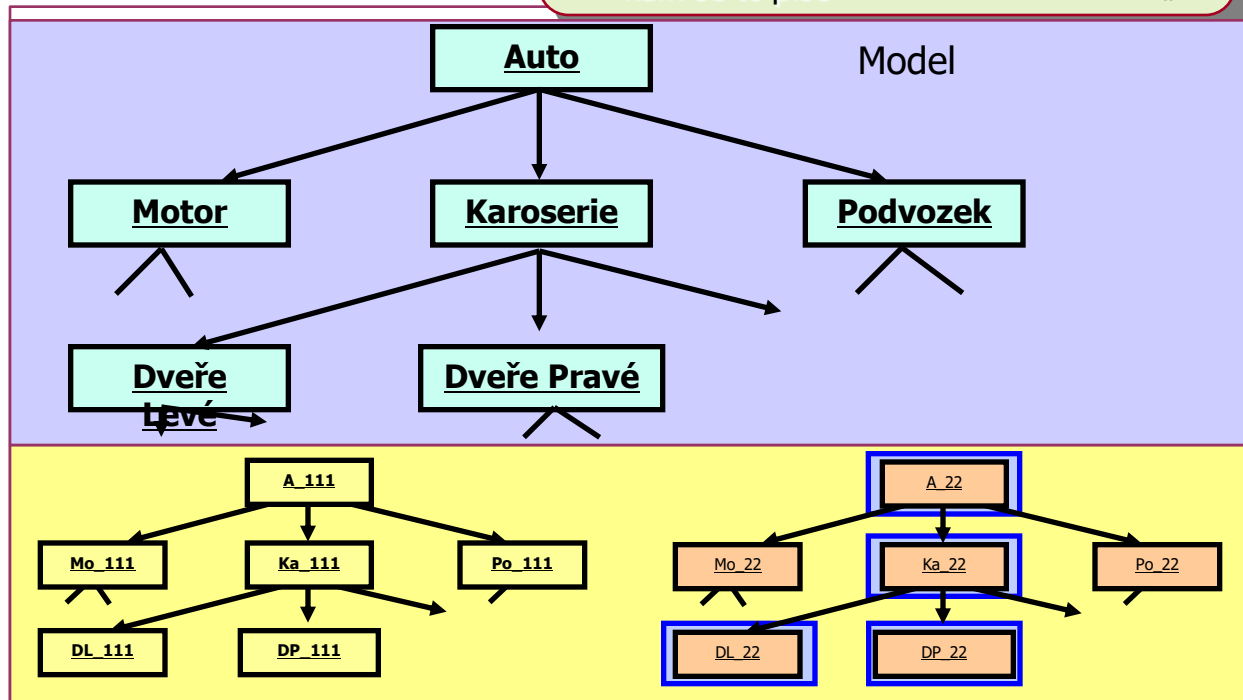
- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Sít'ové



DOTAZ-Zobraz

Karoserii [**Ka_22**]

s daty Auta,

Levých a Pravých

Dveří

Najdi Karoserii s Id= Ka_22

a nastav se ve stromu

Čti Rodiče [Karoserii -> Ka_22]

Čti Dítě LevéDveře [-> DL_22]

Čti Dítě PravéDveře [-> DP_22]

-- Dotaz Realizován

UF !!!

CALL (.....)

CALL (.....)

CALL (.....)

CALL (.....)

- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Sít'ové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu stromu

teoretický základ

žádný [všeobecně přijímaný]

dotazovací [manipulační] jazyk

žádný

povely [tudíž]

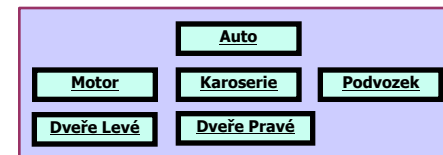
žádné

realizace [každého] dotazu

program

laik, amatér,

nebyl schopen s DB spolupracovat



- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Síťové

jednoduchý dotaz =

napsat složitý program

navigační přístup k jednomu záznamu

od nalezeného záznamu běhání po stromu

datová nezávislost

navigace v DB - zcela závislá na návrhu stromu

teoretický základ

žádný [všeobecně přijímaný]

dotazovací [manipulační] jazyk

žádný

povely [tudíž]

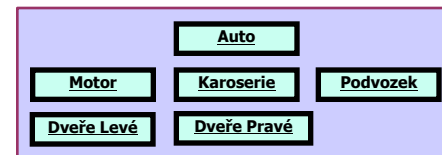
žádné

realizace [každého] dotazu

program

laik, amatér,

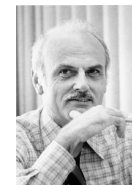
nebyl schopen s DB spolupracovat



- v začátcích relační koncepce
- opět přišel

Codd

a informatikům - programátorům zatrh tipec



- není určen pro matematiky nebo programátory
- jednoduchý jazyk, pochopitelný lajky
- jazyk v přirozené a jednoduché angličtině
- neprocedurální jazyk

SQL

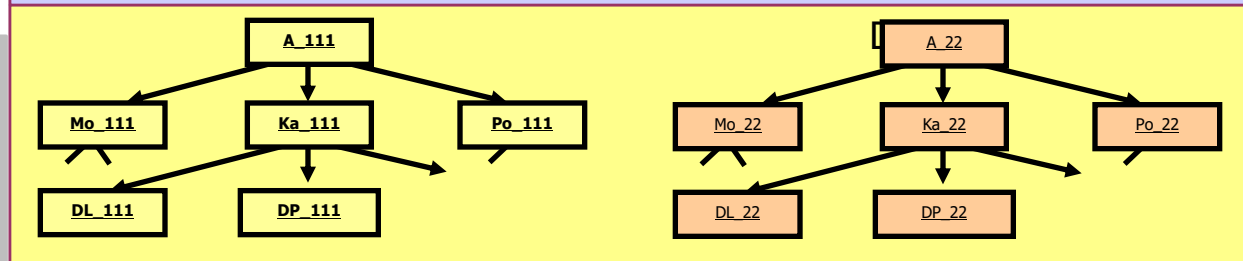
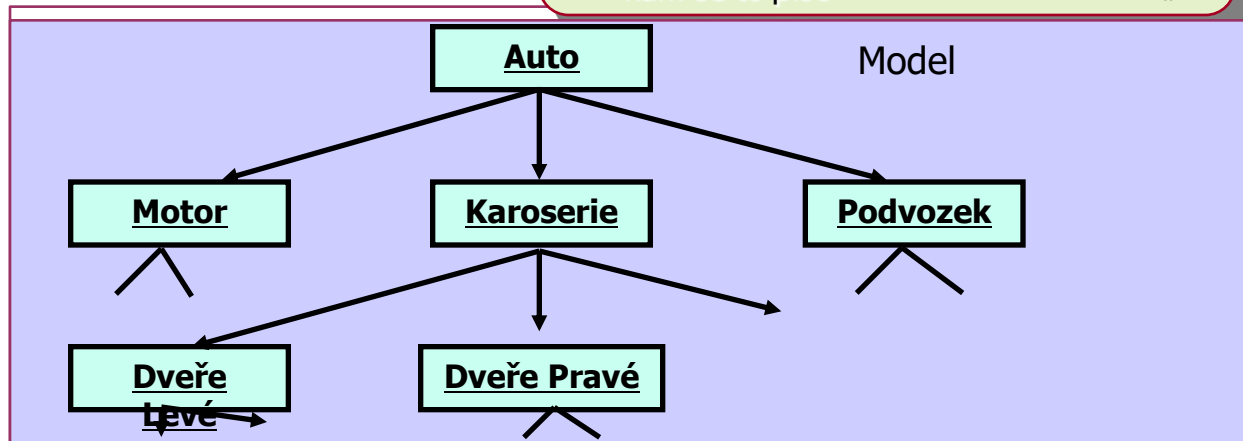
- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Sít'ové



SQL

```

SELECT
Auto.*,
Karoserie.*,
LDveře.*,
PDveře.*
WHERE
IdKaro='Ka_22'
AND .....;
    
```

Najdi Karoserii s Id= Ka_22
 a nastav se ve stromu
 Čti Rodiče [Karoserii -> Ka_22]
 Čti Dítě LevéDveře [-> DL_22]
 Čti Dítě PravéDveře [-> DP_22]
 -- Dotaz Realizován

```

CALL (.....)
CALL (.....)
CALL (.....)
CALL (.....)
    
```

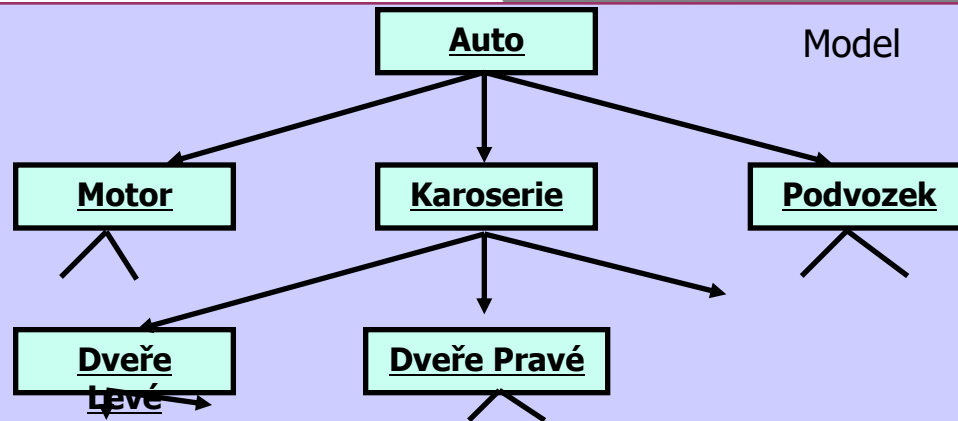
- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Sít'ové



SQL

SELECT

Auto.*,

Karoserie.*,

LDvere.*,

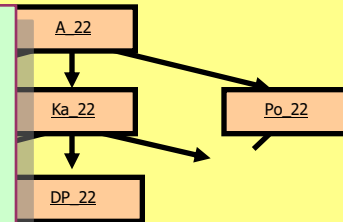
PDvere.*

WHERE

IdKaro='Ka_22'

AND;

- běhej si po DB **jak chceš**
- dělěj si to **jak chceš**
- čti nejdřív tohle a pak zas jinohle
- já do toho nebudu mluvit
- nechci se v tom vůbec vrtat
- klidně si to pobíhání **zoptimalizuj**
- **když mi dáš dobře**
- **to co chci**



CALL (.....)

CALL (.....)

CALL (.....)

CALL (.....)

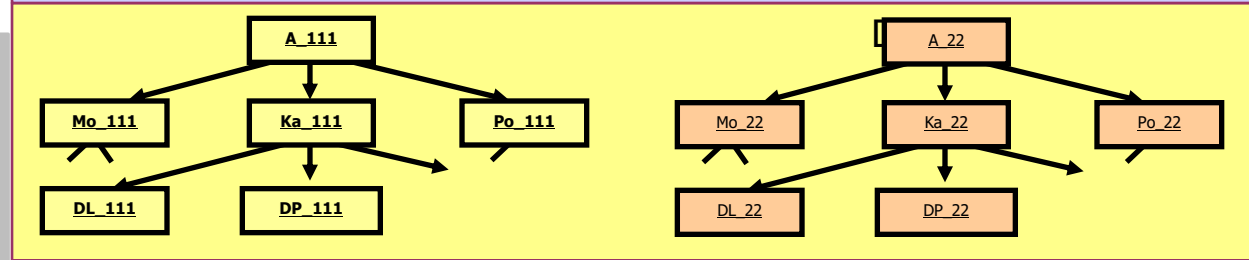
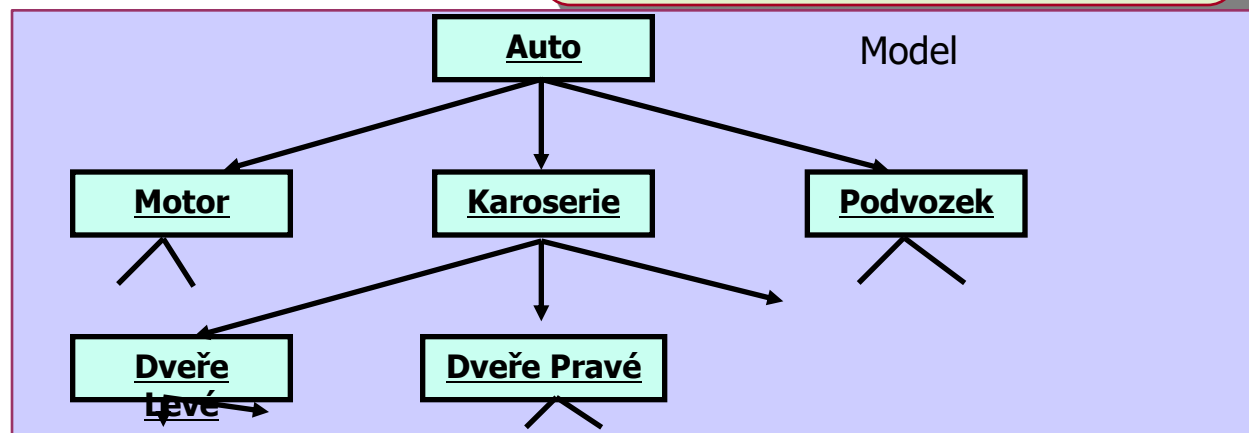
- ty povely
- CREATE, DROP, DELETE, INSERT atd
- co to je ??
- kam se to píše



PředRelační

Hierarchické

Sít'ové



SQL

```

SELECT
Auto.*,
Karoserie.*,
LDvere.*,
PDvere.*
WHERE
  IdKaro='Ka_22'
  AND .....;
  
```

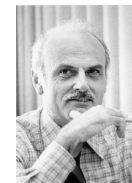
```

CALL (.....)
CALL (.....)
CALL (.....)
CALL (.....)
  
```

```

Call ...; IF ...
      ELSE
      END IF;
Call ...; IF ...
      ELSE
      END IF;
  
```

Codd



- není určen pro matematiky nebo programátory
- jednoduchý jazyk, pochopitelný lajky
- jazyk v přirozené a jednoduché angličtině
- neprocedurální jazyk

SQL

Sequel

- **Structured Query Language**
- skládá se z povelů
- základních povelů je pouze několik
- každý povel začíná slovesem

- požadavek SQL může obsahovat 1 nebo více povelů
- povely se oddělují středníkem
- samotný povel nemusí být ukončen středníkem

;

```
povel1
```

```
povel1;
```

```
povel1; povel2
```

```
povel1; povel2;
```

```
povel1; povel2; povel3.....; povelN
```

- nerozlišují se velká a malá písmena

```
povel1 osoba;
```

```
POVEL2 OSOBA;
```

```
PoVeL3 OsOba...;
```

```
POVeL_N osoBA
```

- a to prosím proč ?



- i já se ptám



- každý povel začíná (anglickým) slovesem
- za slovesem se vyskytují různé klauzule daného povelu
- všech povelů je nemnoho (asi 40)
- 7 základních povelů
- bohatost jazyka je dána
 - různými [možnými] variantami klauzulí
- povely jsou děleny do několika skupin

- Structured Query Language
- nástroj pro organizování
správu
a získávání dat
 - uložených v počítačové databázi
- průmyslový standard pro počítačové DB jazyky
 - pokud je to standard, proč je tolik různých verzí ?
- nepatří žádné instituci ani jedinci
- nezávislý na prodejci (?)

- Structured Query Language
- přenositelný mezi počítačovými systémy (?)
- komplexní DB jazyk
- interaktivní dotazy "ad hoc"
- **neprocedurální jazyk**
- programový přístup k databázi

SQL**DDL**

Data Definition Language

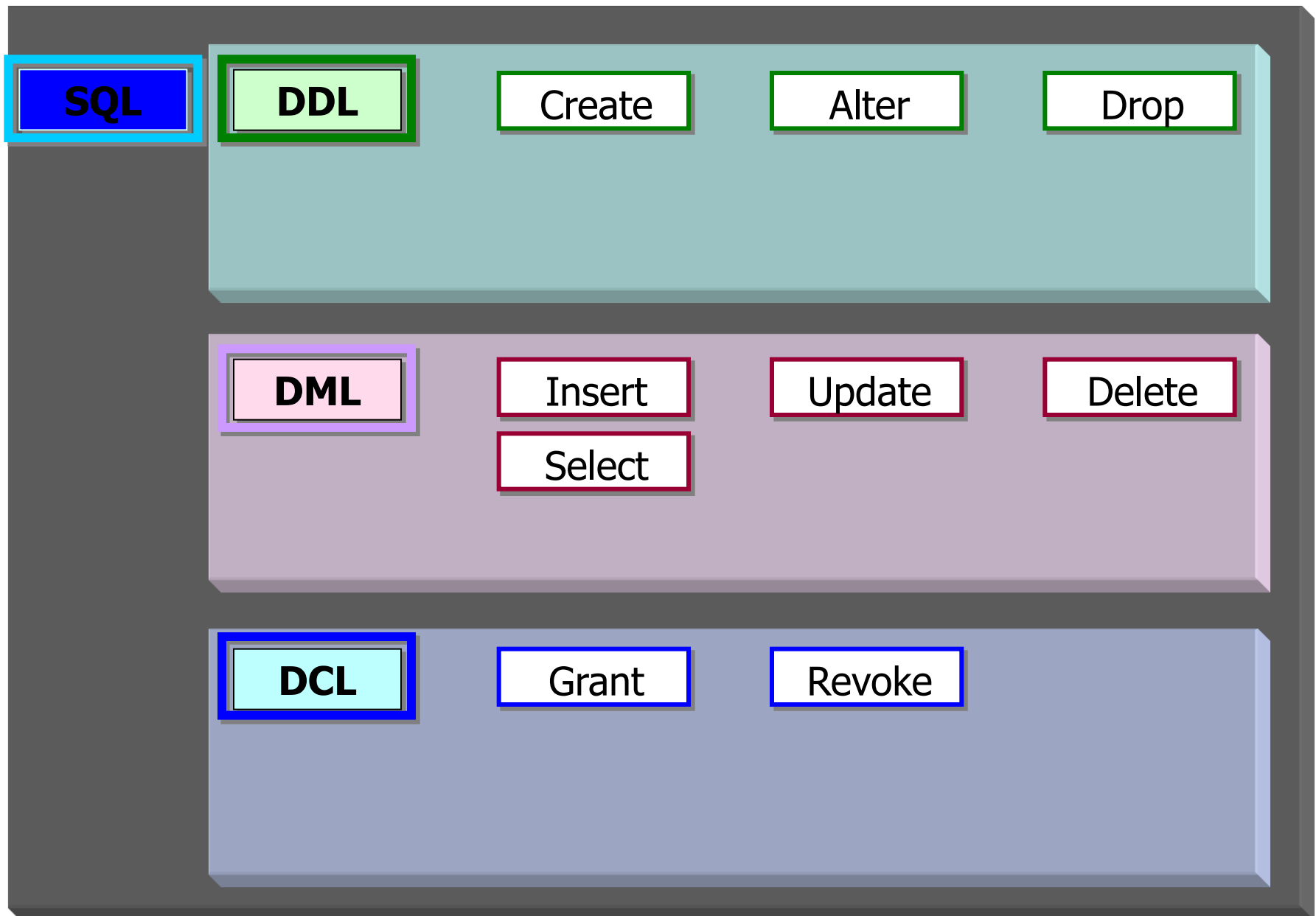
Definice a údržba DB objektů**DML**

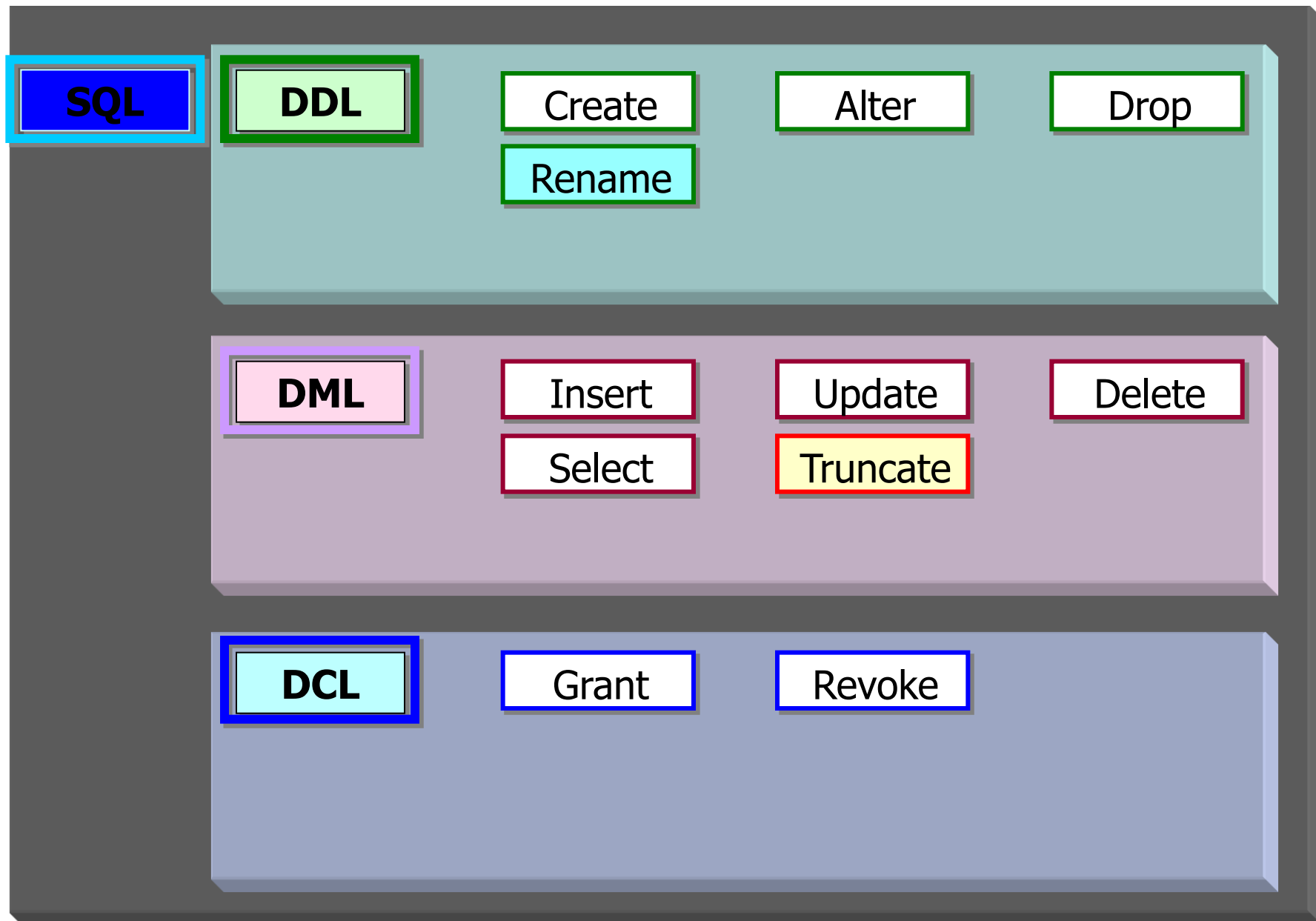
Data Manipulation Language

Manipulace s daty DB**DCL**

Data Control Language

Zadávání a údržba práv (oprávnění)





- ty povely SQL se posílaj poštou
- nebo jak se do te DB dostanou ?



- zásadně technologie KLIENT-SERVER
- aplikace rozdělena na spolupráci dvou počítačů
 - počítač SERVERu
 - na počítači pracuje program SERVERu [DB Serveru]
 - program poslouchá na určitém portu na zprávu [přerušení]
 - počítač Klienta
 - na počítači pracuje program KLIENTa[DB Klienta]



- uživatel [pomocí klienta] zadává povel SQL
- --> DB stroji [server]
- je-li povel špatně
- DB stroj označí chybu
- je-li povel dobře
- je proveden + uživatel dostane výsledky



povel je prověřen

- syntakticky

- odpovídá gramatice povelů ?

- semanticky

- oslovuje databázi známé objekty ?

- jsou jména takových objektů v katalogu DB ?



a zas něco jiného

Placené

- Oracle [Ellison 14]
- DB2 [IBM 11]
- SQL Server [2012] [MS]
- Sybase

Neplacené

- MySQL [5.1 5.5 6.0]
- PostgreSQL [Ingres 9.4]
- Firebird [Interbase Borland]
- [2.5 3.0-vyvoj]

- multiplatformní DBS
 - pokročilé možnosti zpracování dat
 - vysoký výkon
 - snadná škálovatelnost
- jednička mezi profesionálními placenými DBS
- Larry Ellison - jeden z nejbohatších - 1977

- www.oracle.com
- www.oracle.cz



- objektově-relační DBS
- původně pro sálové počítače [mainframe]
- mnoho různých platforem HW - OS
 - různé verze DB pro různé HW-OS - nepřehledné označování
 - Viper, Viper2, ...

- špičková inovace spolupráce DB2- XML
- pureXML
- www-01.ibm.com/software/data/db2
- www.db2.cz
- www.ibm.com/cz/cs/

MF SERVIS s. r. o. - Produkty DB2 9 - Mozilla Firefox

www.mfservis.cz/db2-edice/produkty-db2-9.html

MF SERVIS s.r.o.
MICRO FOCUS software
informační technologie

IBM :: O společnosti :: Produkty :: Služby :: Podpora :: Dokumentace :: Kontakt

IBM

- Produkty
 - Business Intelligence
 - DB2 verze 9
- Informace
 - Edice DB2 a jejich licencování
 - BI aplikace-výrobní linka IBM
 - DB2 9.7 - nové funkce
 - DB2 9.1 - nové funkce
- Micro Focus pro Windows
- Micro Focus pro UNIX, Linux
- Micro Focus pro System z

hledat...

IBM

MICRO FOCUS

Produkty IBM

DB2 - databáze pro relační data a XML	DB2 - Business Intelligence
Nové vlastnosti DB2 - verze 9 DB2 Everyplace (E) DB2 Personal Edition (PE) DB2 Express Edition (ExE) DB2 Express -C (EC) DB2 Workgroup Server Edition (WSE) DB2 Enterprise Server Edition (ESE) DB2 Enterprise Developer Edition (EDE) DB2 Advanced Enterprise Server Edition (AESE) DB2 Přehled nástrojů, vlastností a funkcí	DB2 Data Warehouse Starter Edition (DWSE) DB2 Data Warehouse Intermediate Edition (DWIE) DB2 Data Warehouse Advanced Edition (DWAE) DB2 Data Warehouse Enterprise Base Edition (DWEBE) DB2 Data Warehouse Enterprise Edition (DWEE) DB2 Warehouse Developer's Edition (DWDE) DWE Design Studio DWE SQL Warehousing Tool DWE Administration Console DB2 Query Patroller (QP) OLAP Acceleration (drive Cube Views) DWE Intelligent Miner DB2 Alphablox (AB) Přehled komponent DWES WebSphere Information Analyzer (WIA) WebSphere Business Glossary (WBG) WebSphere DataStage (WDS) WebSphere QualityStage (WQS) Rational Data Architect (RDA)
DB2 ESE - placené nástroje, vlastnosti a funkce	DB2 WSE a ExE - placené nástroje, vlastnosti a funkce
DB2 Storage Optimization Feature DB2 Advanced Access Control Feature DB2 Performance Optimization Feature pro ESE DB2 Database Partitioning Feature (DPF) DB2 Geodetic Data Management Feature IBM Homogeneous Federation Feature	DB2 High Availability Feature DB2 Performance Optimization Feature pro ExE a WSE DB2 Workload Management Feature pro ExE a WSE IBM Homogeneous Federation Feature
DB2 - připojení k host platformě	DB2 - další produkty
DB2 Connect Personal Edition (CPE) DB2 Connect Enterprise Edition (CEE) DB2 Connect Application Server Edition (CASE) DB2 Connect Unlimited Edition for zSeries (CUEz) DB2 Connect Unlimited Edition for iSeries (CUEi)	DB2 Net Search Extender DB2 Performance Expert DB2 Data Links Manager (DLM) DB2 Spatial Extender (SE) DB2 Recovery Expert (RE) DB2 Table Editor (TE) DB2 Web Query Tool (WQT)
WebSphere Information Integration	
WebSphere Federation Server WebSphere Replication Server WebSphere Data Event Publisher	

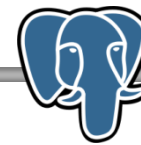
Find: | Next Previous Highlight all Match case

- relační a analytický DBS
- [1990-92] Microsoft Access a Microsoft FoxPro
 - tintítka, MS nemá žádný konkurenceschopný DBS
- 1992 SQL SERVER - spolu se **Sybase** - 4.2
- později přestane spolupráce se Sybase
- [SQL Server 4.2 , ... 7.0, 2000, 2005, 2008,2012,2014]

- SQL Server 2014

- <http://www.microsoft.com/cze/sqlserver2014/>

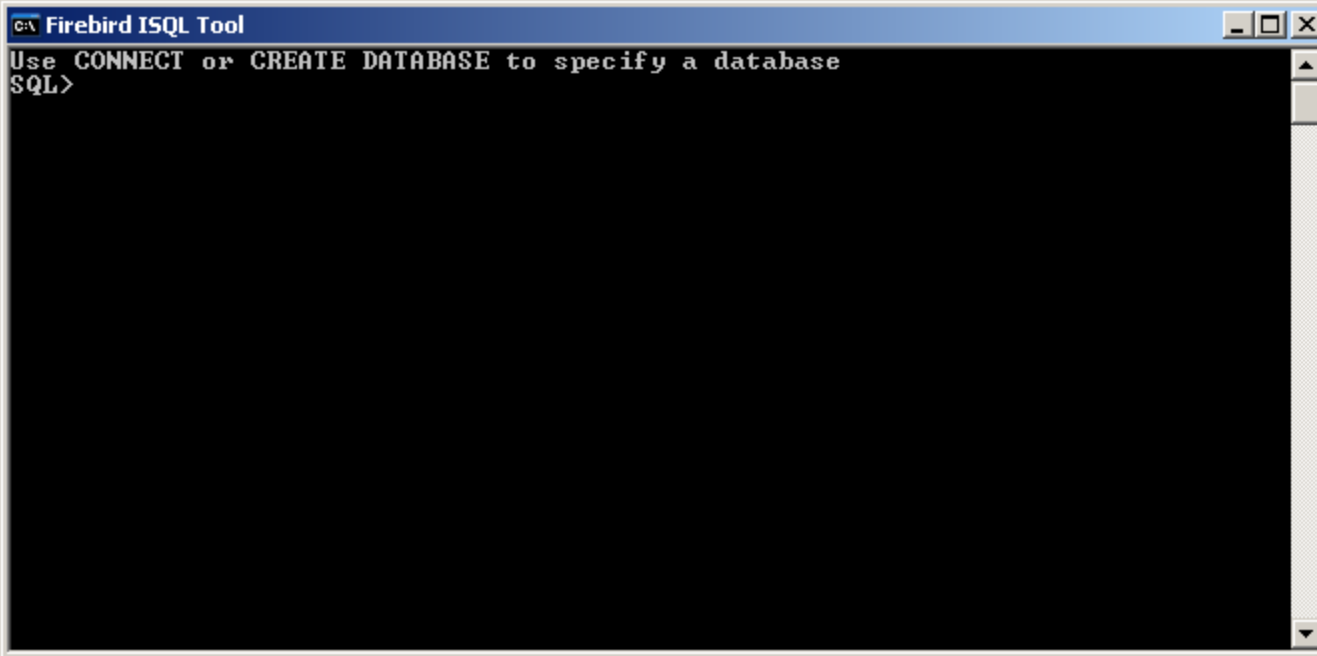
- nejrozšířenější neplacený [Open Source] relační DBS
- velmi rychlý pro čtení
- 1995 Švédská společnost MySQL AB
- 2008 akvizice Sun + MySQL [1 miliarda \$]
- 2010 leden Oracle koupil Sun Microsystems
- verze 12c 2013,2014
-



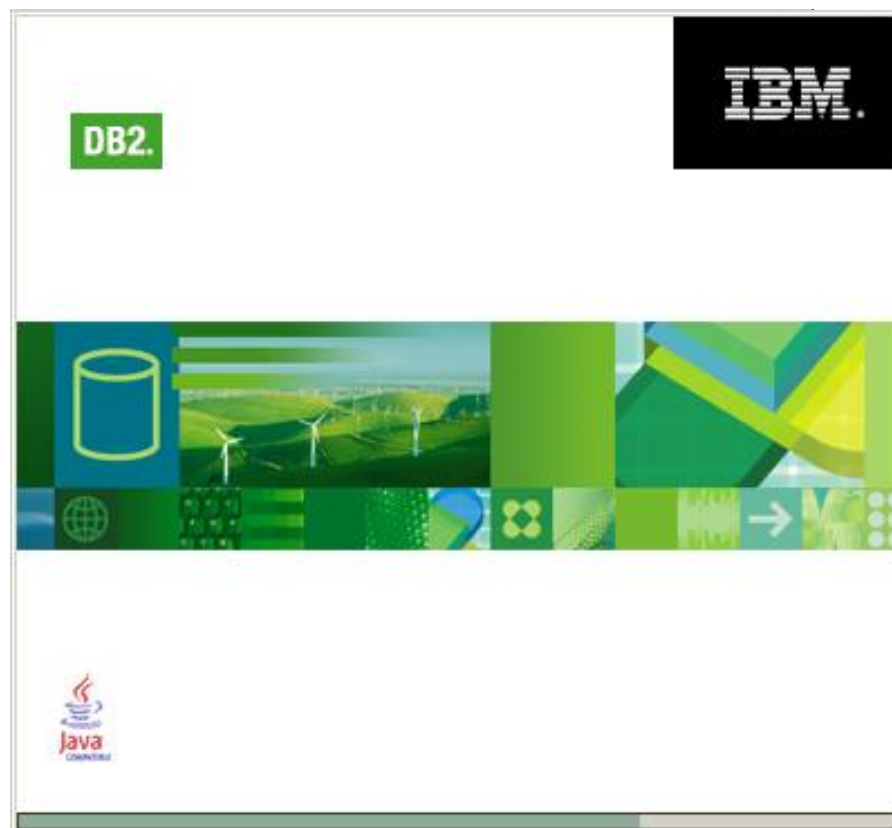
- objektově-relační DBS
- free a open source software
- globální komunita vývojářů a firem
- Linux, Windows
- **Ingres** - University of California, Berkeley [1882]
- postIngres
- mnoho ocenění [1999, 200, 2002, ..., 2008]
- verze 8.3, ... 9.5
- www.postgresql.org
- www.postgres.cz

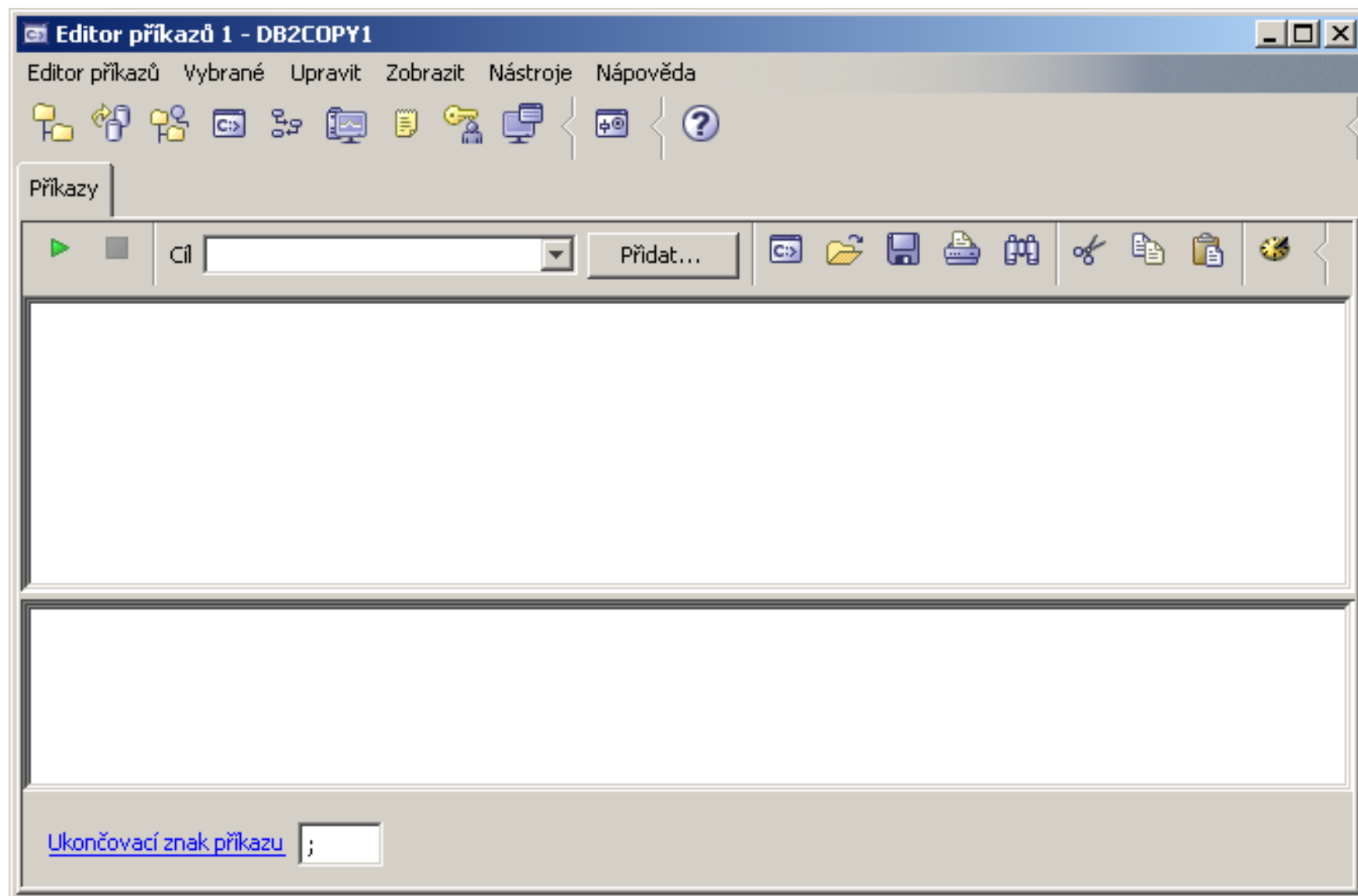


- multiplatformní relační databáze [linux, windows]
- 2000 Borland InterBase 6.0 Open Source uvolněn
- open source licence
- nevytvořila se Borland Open Komunita
- vyvíjí a spravuje **Firebird Foundation**
- prohlížeč Mozilla Firebird - Mozilla Firefox
- ocenění 2007, 2009
 - SourceForge Community Choice Award:
 - Best Project for enterprise
- www.firebirdsql.org
- groups.google.com/group/firebird_cz/topics

A screenshot of a Windows-style window titled "c:\ Firebird ISQL Tool". The window has a blue title bar and standard minimize, maximize, and close buttons. The main area is black with white text. The text reads: "Use CONNECT or CREATE DATABASE to specify a database" followed by "SQL>" on the next line. There are scroll bars on the right side of the window.

```
c:\ Firebird ISQL Tool
Use CONNECT or CREATE DATABASE to specify a database
SQL>
```





MySQL Query Browser 1.2.16

MySQL Query Browser

Connect to MySQL Server Instance

Stored Connection: local

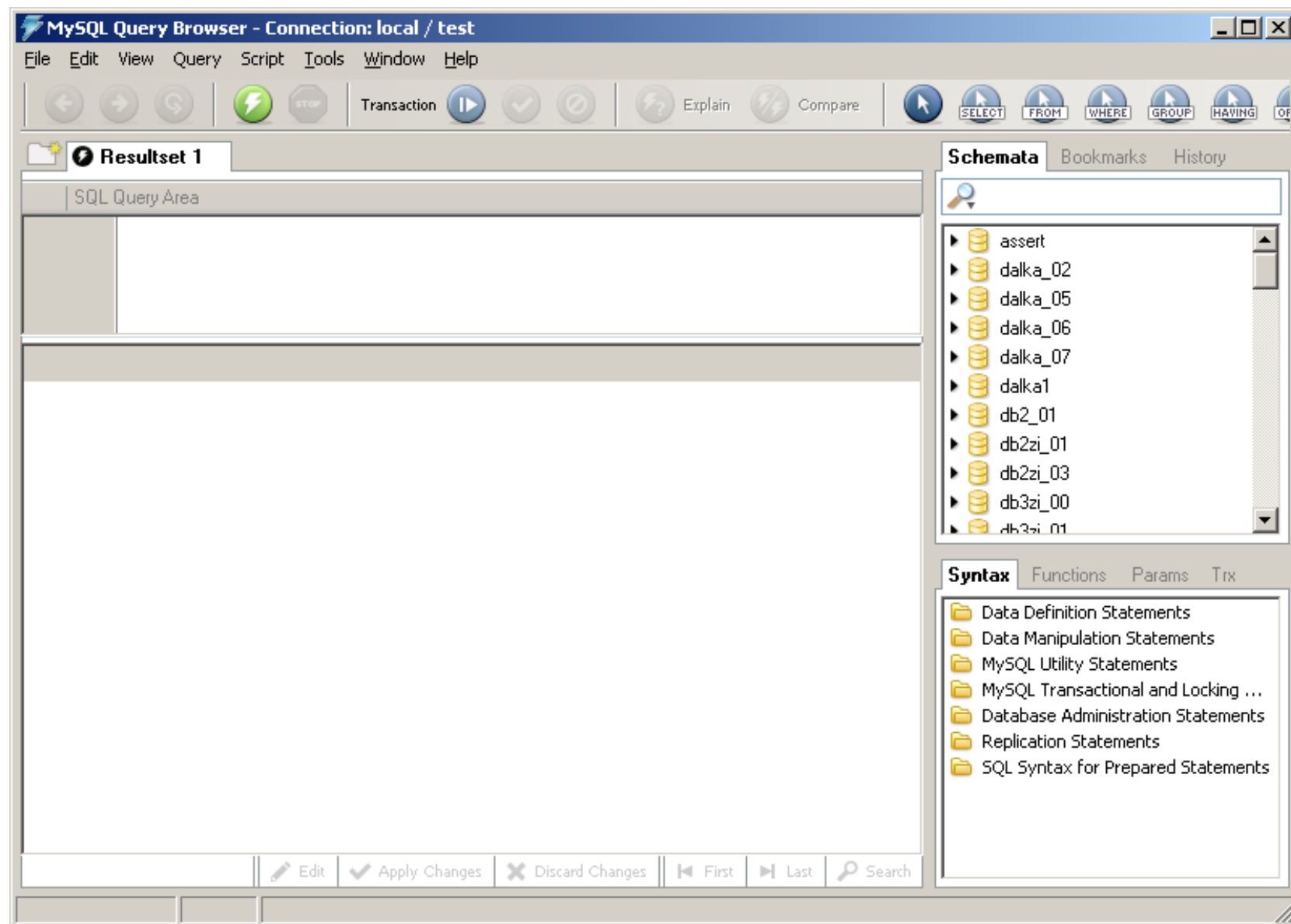
Server Host: localhost Port: 3306

Username: root

Password:

Default Schema: test

Details >> OK Clear Cancel



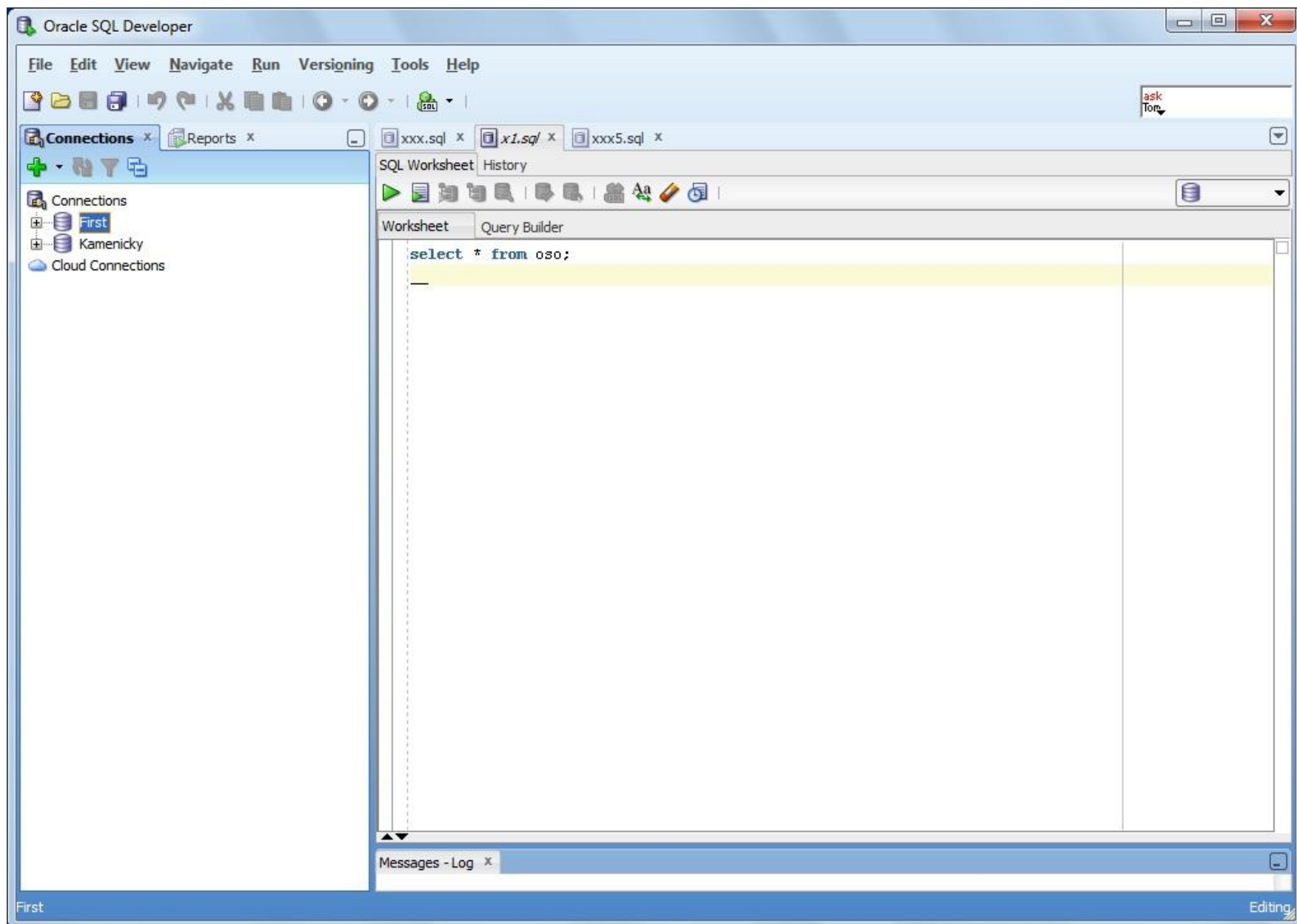
The screenshot displays the MySQL Workbench application window. At the top, the title bar reads 'MySQL Workbench'. Below it is a menu bar with options: File, Edit, View, Database, Plugins, Scripting, Community, and Help. The main interface is divided into two primary sections: 'Workbench Central' and 'Workspace'.

Workbench Central: This section features a 'Welcome to MySQL Workbench' message with a 'What's New in This Release?' link. To the right, there are several utility icons: MySQL Doc Library, MySQL Bug Reporter, Workbench Team Blog, Planet MySQL, Workbench Forums, Scripting Shell, and MySQL Utilities.

Workspace: This section is organized into three main columns:

- SQL Development:** Includes a 'SQL Development' overview with a lightning bolt icon and a description: 'Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.' Below this is a 'New Connection' button and a list of actions: 'Open Connection to Start Querying', 'Edit Table Data', 'Edit SQL Script', and 'Manage Connections'.
- Data Modeling:** Includes a 'Data Modeling' overview with a diagram icon and a description: 'Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.' Below this is an 'Open Existing EER Model' button and a list of actions: 'Create New EER Model', 'Create EER Model From Existing Database', and 'Create EER Model From SQL Script'.
- Server Administration:** Includes a 'Server Administration' overview with a play button icon and a description: 'Configure your database server, setup user accounts, browse status variables and server logs.' Below this is a 'Server Administration' button and a list of actions: 'New Server Instance', 'Manage Import / Export', 'Manage Security', and 'Manage Server Instances'.

At the bottom left of the window, the status bar shows 'Ready.' and at the bottom right, there are window control icons.



Oracle SQL Developer : ormar

File Edit View Navigate Run Versigning Tools Help

Connecti... x OmarCosi.sql x oooo.sql x Alkfa_01.sql x aaa.sql x huhu.sql x yyy.sql x dd2.sql x Ctrvtek.sql x iii.sql x ormar x

Connections

- ormar
 - Tables (Filtered)
 - AATAB
 - ABC
 - BUDOVA
 - CENA
 - DALL
 - DATUMY
 - DD2
 - HOSPUDKA0
 - CH
 - CHLAP
 - MUZ
 - MUZ3
 - NASZAM
 - NASZAM2
 - ODD
 - ODDIL
 - ODEL
 - OS
 - OS2
 - OSOX
 - OSOY
 - OSUBKA
 - PO
 - POBO
 - PODEZRELI
 - PROBLEM0
 - PROD
 - PRODEJCI
 - PRODO

Worksheet Query Builder

```
select * from NasZam;
```

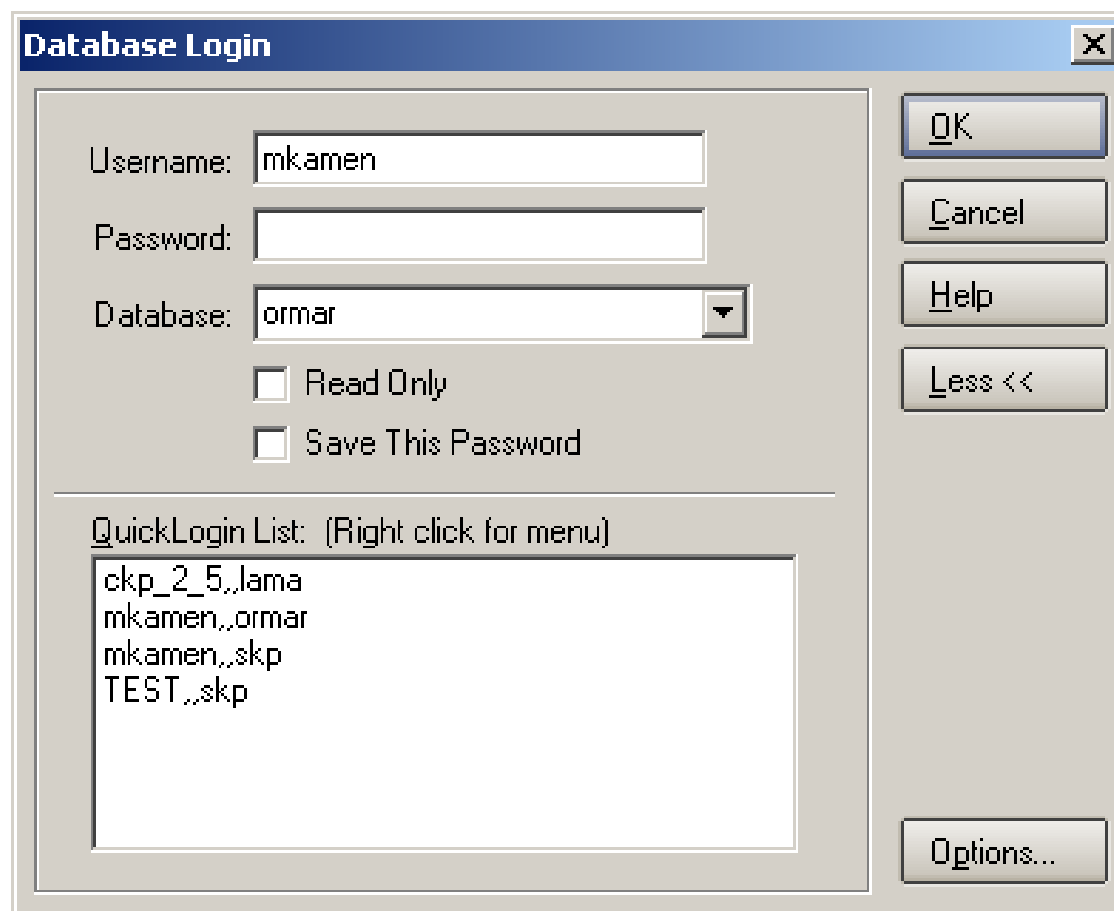
Query Result x

SQL | All Rows Fetched: 6 in 0,078 seconds

	IDZAM	JM	POH	TIT	ROKNAR	IDPOBO	IDODEL	PLAT
1	1	Ada	m	ing	1990	1	1	20000
2	2	Bob	m	(null)	1980	1	1	10000
3	3	Ema	z	mgr	1960	1	1	30000
4	4	Fan	z	(null)	1950	1	2	25000
5	5	Ivo	z	dr	1966	2	1	40000
6	6	Tom	m	(null)	1985	2	1	22000

Messages - Log x SQL History x

Table MKAMEN.null@ormar | Line 1 Column 10 | Insert | Modified | Windows: CR/LF Editing



The image shows a 'Database Login' dialog box with a blue title bar and a close button (X) in the top right corner. The main area contains several input fields and checkboxes. The 'Username' field is filled with 'mkamen'. The 'Password' field is empty. The 'Database' field is a dropdown menu currently showing 'ormar'. Below these are two unchecked checkboxes: 'Read Only' and 'Save This Password'. A horizontal line separates this section from a 'QuickLogin List' section. The list is titled '(Right click for menu)' and contains four entries: 'ckp_2_5,,lama', 'mkamen,,ormar', 'mkamen,,skp', and 'TEST,,skp'. On the right side of the dialog, there are four buttons stacked vertically: 'OK', 'Cancel', 'Help', and 'Less <<'. At the bottom right, there is an 'Options...' button.

Database Login [X]

Username:

Password:

Database: ▼

Read Only

Save This Password

QuickLogin List: (Right click for menu)

ckp_2_5,,lama
mkamen,,ormar
mkamen,,skp
TEST,,skp

[OK] [Cancel] [Help] [Less <<]

[Options...]

mkamen@ormar - Benthic Software: Golden

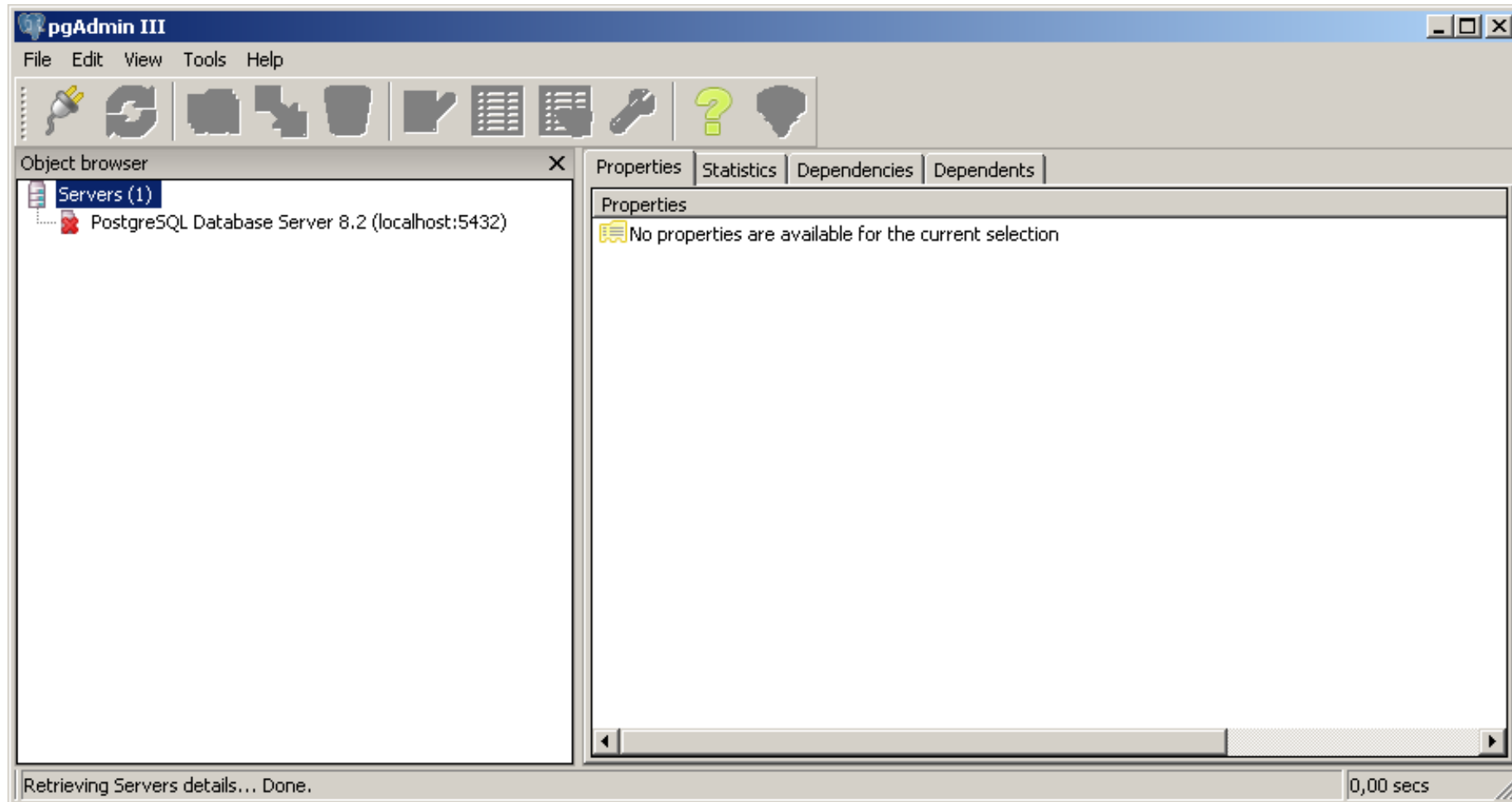
File Edit Script Results View Tools Help

Query1

```
Select *  
From NasZam;
```

#	IDZAM	JM	POH	TIT	ROKNAR	IDPOBO	IDODEL	PLAT
▶ 1	1	Ada	m	ing	1990	1	1	20000
2	2	Bob	m		1980	1	1	10000
3	3	Ema	z	mgr	1960	1	1	30000
4	4	Fan	z		1950	1	2	25000
5	5	Ivo	z	dr	1966	2	1	40000
6	6	Tom	m		1985	2	1	22000

Done, ran 1 of 1 statements. Selected 6 records Script: 0,015 Secs





a vzhůru dovnitř

SQL

- každý DB objekt je pojmenován
- existují 2 druhy SQL jmen

- **nonquoted identifier** [bez "
"]
- **quoted identifier** [delimited] [
"jméno"]

- nonquoted identifier

[bez " "]

- začíná anglickým písmenem A..Z, _, \$ [PostgreSQL ~~\$~~]
- A..Z , _ , \$, 0..9 [PostgreSQL ~~\$~~]
- nelze užít rezervované SQL slovo
- nelze užít českých [švédských, azbuka, ...] znaků
- nerozlišuje se malá a velká písmena
- malá písmena jsou převedena na velká

■ nonquoted identifier [bez " "]

Table		

Column		

Insert		

želva		

moje%tab		

....		
	Table	

Tables		

Zelva		

_Fakt		

TABLES		

ZELVA		

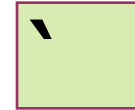
_FAKT		

- **quoted identifier** ["jméno "]
- začíná a končí " "
- uvnitř cokoliv mimo " [čeština ... # \$ % ^ ...]
- lze užít rezervovaná slova
- rozlišují se malá a velká písmena
- malá písmena **nejsou** převedena na velká
- **nikdy** to [radši] neužijte [až na pár komentovaných výjimek]
- toto užívání nechte DB Serveru [má to hlavně pro sebe]

- quoted identifier **MySQL**

[`jméno`]

- užívá místo uvozovek zborcený apostrof



`Table`		

`Column`		

`Insert Delete From`		

`želva i pštros`		

`moje%tab`		

`@#\$%^&`		
	`Table`	

- maximální délka DB jména
 - nejednotné
 - u některých DB různé u různých pojmenování
-
- | | | |
|--------------|------|--------------|
| ■ Oracle | ■ 30 | |
| ■ PostgreSQL | ■ 31 | [MAXLEN-1] |
| ■ MySQL | ■ 64 | |

SQL**DDL****Data Definition Language**

Definice a údržba DB objektů

DML**Data Manipulation Language**

Manipulace s daty DB

DCL**Data Control Language**

Zadávání a údržba práv (oprávnění)

SQL**DDL****Data Definition Language****Definice a údržba DB objektů**


```
Create TypObjektu JménoObjektu [ vlastnosti objektu ] ;  
Create User Novak;  
Create Schema Zbozi;  
Create Table Osoba (sloupec1 ... ,sloupec2 ... , ... );  
Create Table Osoba ( Jmeno Char (20),  
                    Prijmeni Char(20),  
                    DatNar Date );
```

```
Create TypObjektu JménoObjektu [ vlastnosti objektu ] ;
Create Schema Zbozi;
Create User Novak;
Create Table Osoba (sloupec1 ... ,sloupec2 ... , ... );
Create Table Osoba ( Jmeno Char (20),
                    Prijmeni Char(20),
                    DatNar Date );
```

```
Drop TypObjektu JménoObjektu [ vlastnosti zrušení ] ;
Drop Schema Zbozi;
Drop User Novak;
Drop Table Osoba;
```

```
Create Schema JmenoSchema [ Authorization JmenoUzivatele ];
```

```
Create Schema EviOsob; Err
```

```
Create Schema EviOsob Authorization Novak;
```

```
Create Schema If Not Exists EviOsob;
```

```
Drop Table Osoba; Err
```

```
Drop Table If Exists Osoba; Err
```

```
Drop Table .... JmenoTab [ Restrict Cascade ];
```

```
Drop Table If Exists Osoba Cascade ;
```

```
Create Table If Not Exists Osoba ( Jmeno Char (20),  
Prijmeni Char(20),  
DatNar Date );
```

```
Create Schema If Not Exists EviOsob;  
Drop Table If Exists Osoba Cascade ;  
Create Table Osoba ( Jmeno Char (20),  
                    Prijmeni Char(20),  
                    DatNar Date );  
Drop Table If Exists Tab2 Cascade ;  
Create Table Tab2 ( Sloup1 ...,  
                  .....  
                  SloupN ...);  
.....
```

Alter TypObjektu NázevObjektu změný.....;

Alter Table JménoTable změny;

Alter Table Osoba Rename To Clovek;

Alter Table Osoba Set Schema Evid; [Xxx.Osoba →Evid.Osoba]

Alter Table Osoba Add Column Vyska Int

Drop Column BarvaOci

Drop Column If Exists BarvaOci

Rename Column Jmeno To Jm

Alter Column Plat Type To Int

Owner To Alena

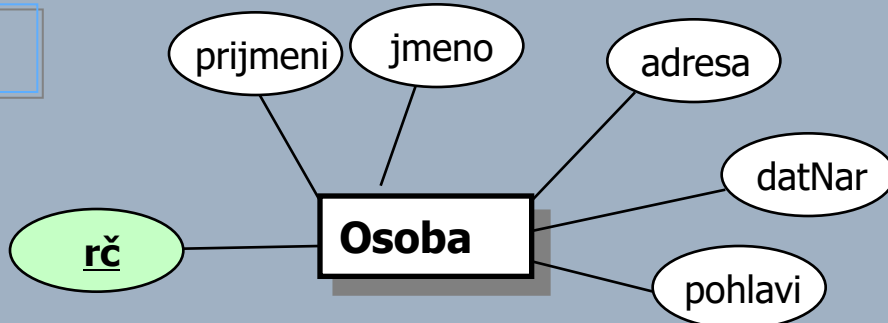
;

- přidat sloupec
- zrušit sloupec
- přejmenovat sloupec
- změnit typ dat sloupce
- přejmenovat tabulku
- přidat, změnit, zrušit jiné vlastnosti sloupců
- přidat, změnit, zrušit jiné vlastnosti tabulky [vlastníka]

- **Definice vlastností tabulky**
- **Definice sloupců as jejich vlastností**
- **Sloupec není DB objekt , ale podobjekt Tabulky**
- **resp. DB objekt vždy podřízený tabulce**

```
Create Table JménoTab (  
    JménoSloupce1   DatovýTyp [ další nepovinné vlastnosti ] ,  
    Sloupec2        DatovýTyp ,  
    .....  
);
```

E-R model



Osoba

Osoba					

numerické

znakové

časové

boolean

rozsáhlé

objektové

numerické

Int

Numeric [(p,s)]

Real - Float

Numeric

Decimal [(p,s)]

znakové

Char(n)

VarChar(n)

Text

časové

Date

Time

DateTime

Timestamp

boolean

rozsáhlé

objektové

-	textové	Char (n)	text pevné délky	< = 255
-	..	Varchar(n)	text proměnné délky	< = 255
-		Text	text proměnné délky	< = 65

535

-	číselné	Smallint	celočíslné - 32 nnn	+32 nnn
-		Int	32 bit celočíselné	
-		Float (p)	plovoucí	
-		Dec (p,z)	Dekadické [p-před,z-za]	
-	časové	Date	YYYY-MM-DD	
-		Time	HH:MI:SS	
-		Datetime	YYYY-MM-DD HH:MI:SS	
-	Logické	Boolean	TRUE nebo FALSE	

- různé typy dat [mají]
- různé vlastnosti

účel ???
správných

- vymezení operátorů a funkcí

- pro manipulaci
- pro prezentaci

SQL**DML**

Insert

Update

Delete

Select

Povel: Insert

Cíl : Vložení jednoho / či více řádek do jedné tabulky

```
Insert Into Tabulka |  
                Tabulka(Sloupec1, Sloupec2,...)  
                Values (Hodnota1, hodnota2,...) ;
```

```
Insert Into Tabulka | Tabulka(Sloupec1, Sloupec2,...)  
                Values (Hodnota1, hodnota2,...) ,           -- pro 1 radku  
                (Hodnota1, hodnota2,...) ,                 -- pro 2 radku  
                (Hodnota1, hodnota2,...) ,                 -- pro 3 radku  
                (Hodnota1, hodnota2,...) ;                 -- pro 4 radku
```

A	B	C

```

Insert Into T1 Values ( 1, 2, 3 );
Insert Into T1 Values ( 11, 22, 33 );
Insert Into T1 Values ( 111, 222,33 3 );
    
```

```

Insert Into T1 Values ( 1, 2, 3 ),
                    ( 11, 22, 33 ),
                    ( 111, 222,33 3 );
    
```

Oracle - nelze

A	B	C
1	2	3
11	22	33
111	222	333

Povel: Delete

Cíl : Výmaz jednoho, více , všech řádek tabulky

```
Delete
```

```
  From Tabulka
```

```
  [ Where PodminkaRušeníŘádku ]
```


A	B	C
1	2	3
11	22	33
111	222	333

```
Delete From T1;
```

A	B	C
---	---	---

A	B	C
1	2	3
11	22	33
111	222	333

```
Delete From T1  
Where B=22;
```

A	B	C
1	2	3
111	222	333

Povel: Update

Cíl : Změna jednoho či více sloupců
jednoho , více, či všech řádek tabulky

```
Update   Tabulka  
        Set     Sloupec1=výraz  
                [ , Sloupec2=Výraz, ..... ]  
[ Where  VýrazVýběruŘádek                       ]
```

A	B	C
1	2	3
11	22	33
111	222	333

```
Update T1 Set B=77;
```

A	B	C
1	77	3
11	77	33
111	77	333

A	B	C
1	2	3
11	22	33
111	222	333

```
Update T1 Set B=77  
Where B= 22;
```

A	B	C
1	2	3
11	77	33
111	222	333

A	B	C
1	2	3
11	22	33
111	222	333

```
Update T1 Set B=77  
Where B > 20;
```

A	B	C
1	2	3
11	77	33
111	77	333

Povel: Select

Cíl : Výpis údajů z tabulky/ tabulek

```
SELECT  Konstanta | Sloupec | Výraz | PodDotaz | * [položka2, položka3,....]
        From      Tabulka
[ Where  Výraz-VýběruŘádek                               ]
[Order By SloupecA | , SloupecB ,.....
         | 3 , 1 , čísloPoložkyDotazu , ...           ]
```

A	B	C
1	2	3
11	22	33
111	222	333

```
Select *  
From T1;
```

A	B	C
1	2	3
11	22	33
111	222	333

A	B	C
1	2	3
11	22	33
111	222	333

```
Select A ,C  
From T1;
```

A	C
1	3
11	33
111	333

A	B	C
1	2	3
11	22	33
111	222	333

```
Select 66 , 'Pepa', 7+3-5
From T1;
```

66	Pepa	7+3-5
66	Pepa	5
66	Pepa	5
66	Pepa	5

A	B	C
1	2	3
11	22	33
111	222	333

```
Select  B, 555 , A+20
From    T1;
```

B	555	A+20
2	555	21
22	555	31
222	555	131

A	B	C
1	2	3
11	22	33
111	222	333

```
Select  C, B, A
From    T1
Where   B < 100;
```

C	B	A
3	2	1
33	22	111

A	B	C
1	2	3
11	22	33
111	222	333

```
Select *  
From T1  
Order By C DESC
```

A	B	C
111	222	333
11	22	33
1	2	3

A	B	C
1	2	3
11	22	33
111	222	333

```
Select  5 + 8 + 3 * 4  
From    T1;
```

$5 + 8 + 3 * 4$
25
25
25

DUAL	
Dummy	
	X

```
Select  5 + 8 + 3 * 4  
From    Dual;
```

5 + 8 + 3 * 4	
	25

DUAL	
Dummy	
	X

```
Select  5 + 8 + 3 * 4 , 'Hola', 55  
From    Dual;
```

5 + 8 + 3 * 4	Hola	55
25	Hola	55


```
Select 5 + 8 + 3 * 4 , 'Hola', 55 ;
```

5 + 8 + 3 * 4	Hola	55
25	Hola	55

Každá položka dotazu může mít zádáno náhradní jméno pomocí **as**

```
SELECT  Konstanta | Sloupec | Výraz | PodDotaz | * [položka2, položka3,....]
        From    Tabulka
[ Where  Výraz-VýběruŘádek ]
[Order By SloupecA | , SloupecB ,....
         | 3 , 1 , čísloPoložkyDotazu , ... ]
```

```
Select  Jm as Jmeno, Pr as "Příjmení",
        Plat + 100 as "NovýPlat", 'Dnes', 12345
        From  Osoba;
Order By  Plat';
```

Osoba				
Jmeno	Příjmení	NovýPlat	Dnes	12345
Bobo	Velky	20 100	Dnes	12345
Pepa	Novak	30 100	Dnes	12345
Hana	Mala	40 100	Dnes	12345

Každá položka dotazu může mít zádáno náhradní jméno

```
SELECT  Konstanta | Sloupec | Výraz | PodDotaz | * [položka2, položka3,....]
        From      Tabulka
[ Where  Výraz-VýběruŘádek ]
[Order By SloupecA | , SloupecB ,....
         | 3 , 1 , čísloPoložkyDotazu , ... ]
```

```
Select  Jm  Jmeno, Pr  "Příjmení",
        Plat + 100  "NovýPlat", 'Dnes', 12345
        From      Osoba;
Order By Plat';
```

Osoba				
Jmeno	Příjmení	NovýPlat	Dnes	12345
Bobo	Velky	20 100	Dnes	12345
Pepa	Novak	30 100	Dnes	12345
Hana	Mala	40 100	Dnes	12345

Tabulka ve From může mít náhradní jméno

```
SELECT  Konstanta | Sloupec | Výraz | PodDotaz | * [položka2, položka3,....]
        From    Tabulka
[ Where  Výraz-VýběruŘádek ]
[Order By SloupecA | , SloupecB ,....
         | 3 , 1 , čísloPoložkyDotazu , ... ]
```

```
Select  Jm  Jmeno, Pr  "Příjmení",
        Plat + 100  "NovýPlat", 'Dnes', 12345

        From  Osoba as Oso;

Order By Plat;
```

Oso				
Jmeno	Příjmení	NovýPlat	Dnes	12345
Bobo	Velky	20 100	Dnes	12345
Pepa	Novak	30 100	Dnes	12345
Hana	Mala	40 100	Dnes	12345

Tabulka ve From může mít náhradní jméno

```

SELECT   Konstanta | Sloupec | Výraz | PodDotaz | * [položka2, položka3,....]
        From     Tabulka
[ Where   Výraz-VýběruŘádek                               ]
[Order By SloupecA | , SloupecB ,....
        | 3 , 1 , čísloPoložkyDotazu , ... ]
    
```

```

Select   Jm   Jmeno, Pr   "Příjmení",
          Plat + 100   "NovýPlat", 'Dnes', 12345

        From   Osoba   Oso;

Order By Plat';
    
```

Oso				
Jmeno	Příjmení	NovýPlat	Dnes	12345
Bobo	Velky	20 100	Dnes	12345
Pepa	Novak	30 100	Dnes	12345
Hana	Mala	40 100	Dnes	12345

```
SELECT  a+b, 13 + 5, a * c
From    Tab
```

A	B	C
1	2	3
10	2	4



Jak to, že není **E RRRRR O RRRR**

a+b	13+5	a * c
3	18	3
12	18	40

Názvy sloupců jsou syntakticky chybné.

a+b neplatný název sloupce

13+5 neplatný název sloupce

a * c neplatný název sloupce



```
SELECT a+b, 13 + 5, a * c
From Tab
```

A	B	C
1	2	3
10	2	4

Stroj SQL Vám vyšel vstříct
Váš povel upravil následovně
Přidal pro každý sloupec **ALIAS**

a+b	13+5	a * c
3	18	3
12	18	40

```
SELECT a+b as "a+b" , 13 + 5 as "13 + 5" , a * c as "a * c"
From Tab
```

A tudíž je to syntakticky OK.

Identifikátory v uvozovkách "xxxx" jsou tam pro Server , ne pro uživatele.

Uživatel je může použít – **ale téměř nikdy to nečiní**

