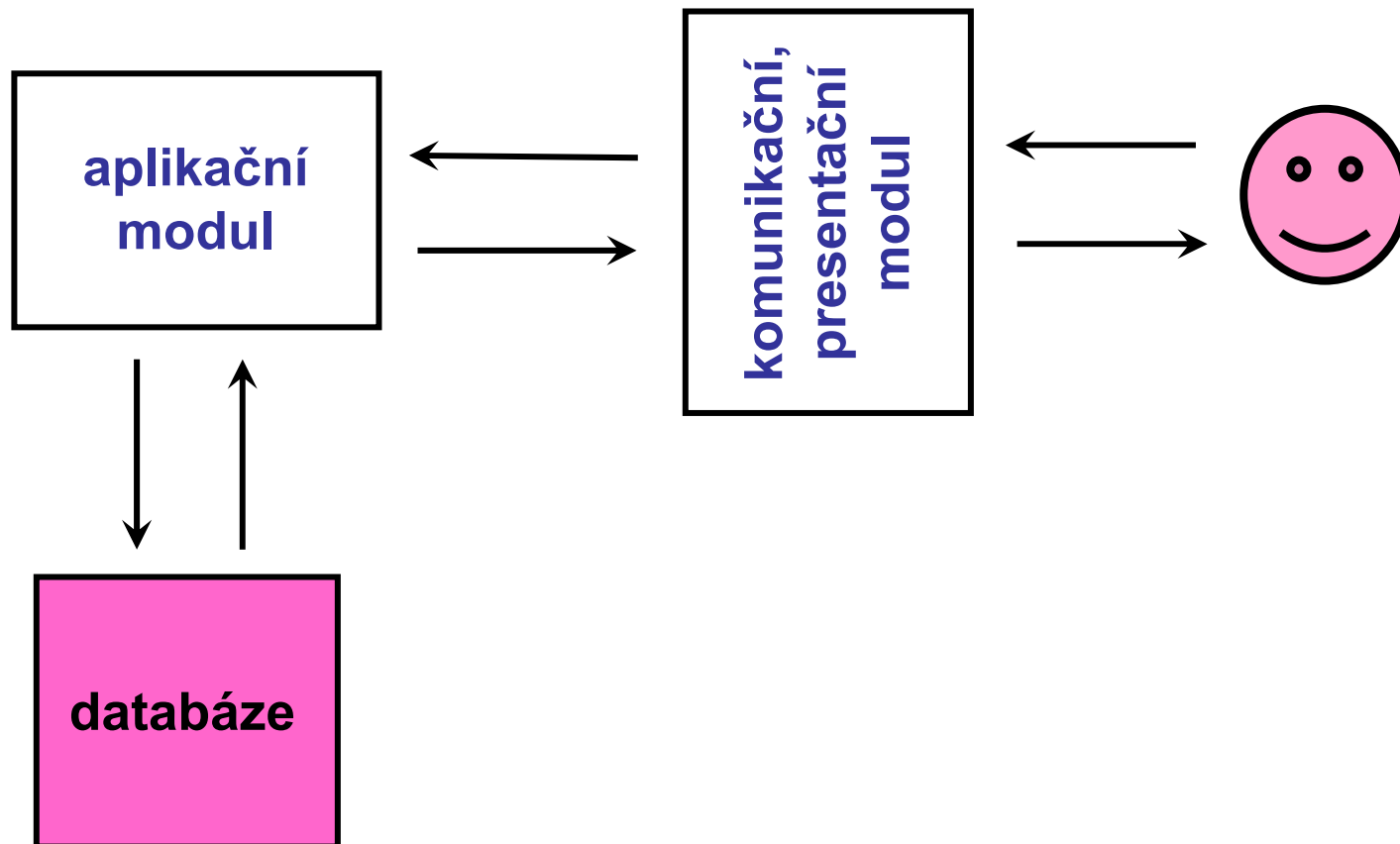


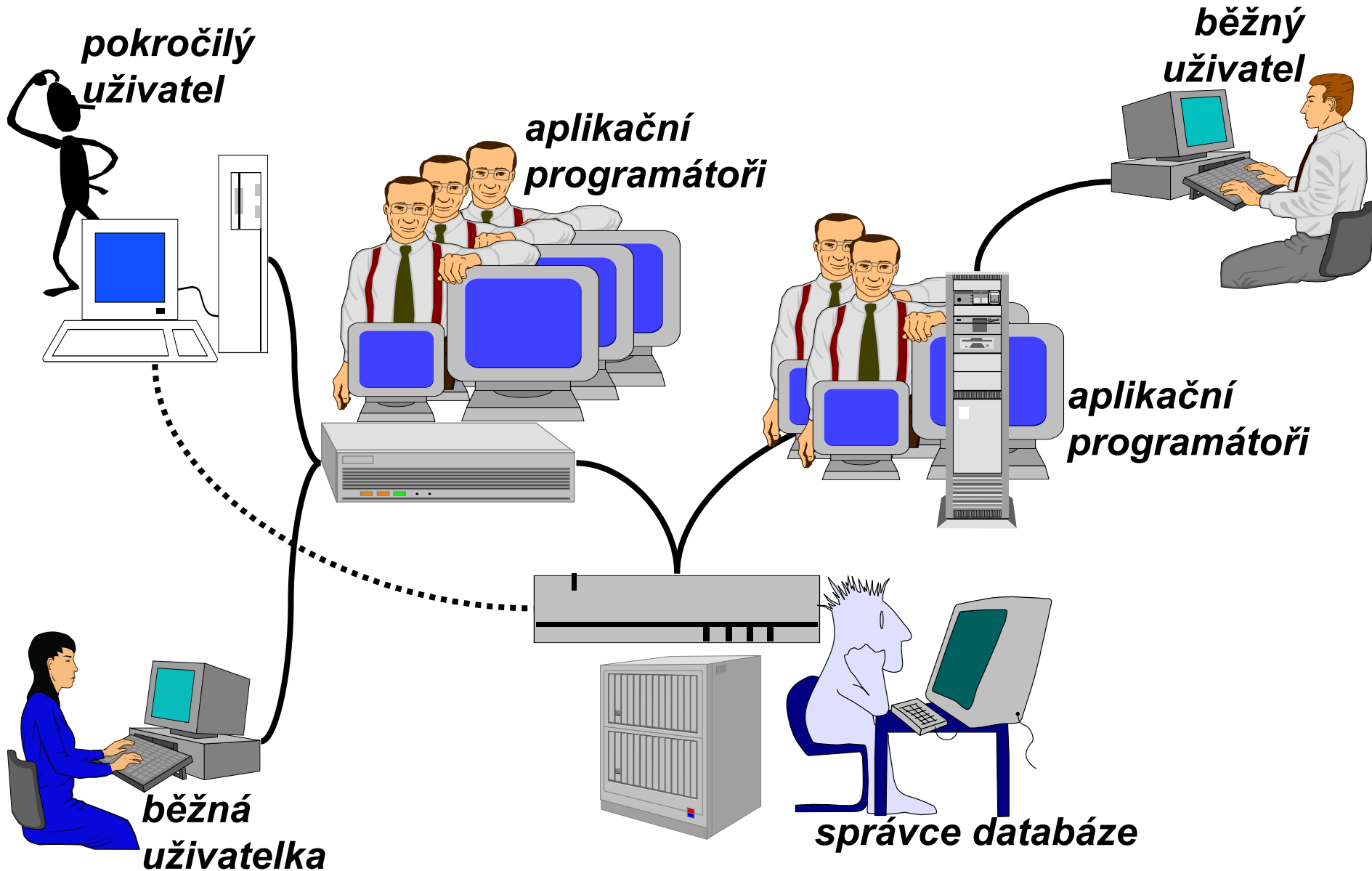
Databázové systémy

Helena Palovská
palovska@vse.cz

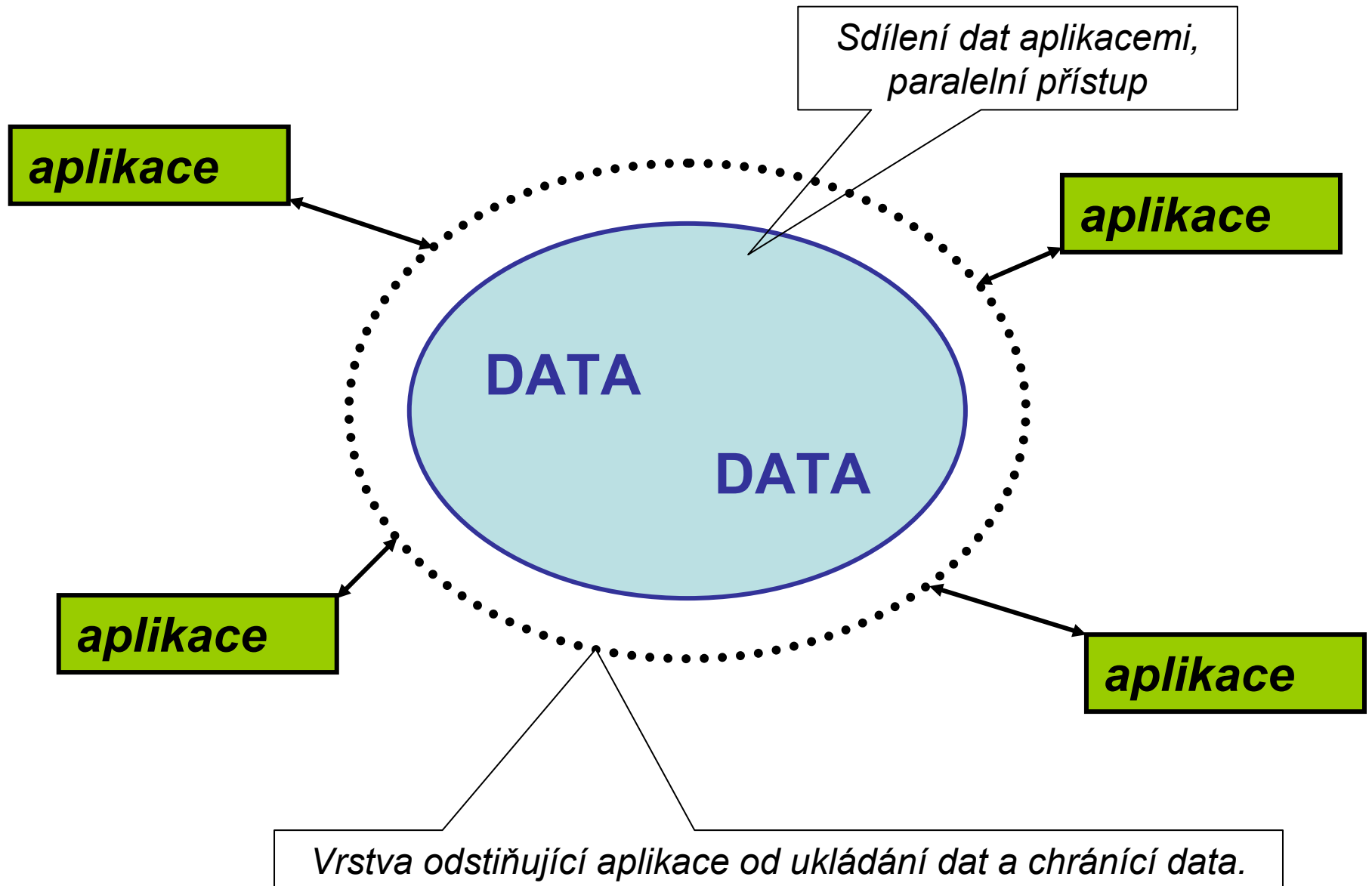
Třivrstvá architektura



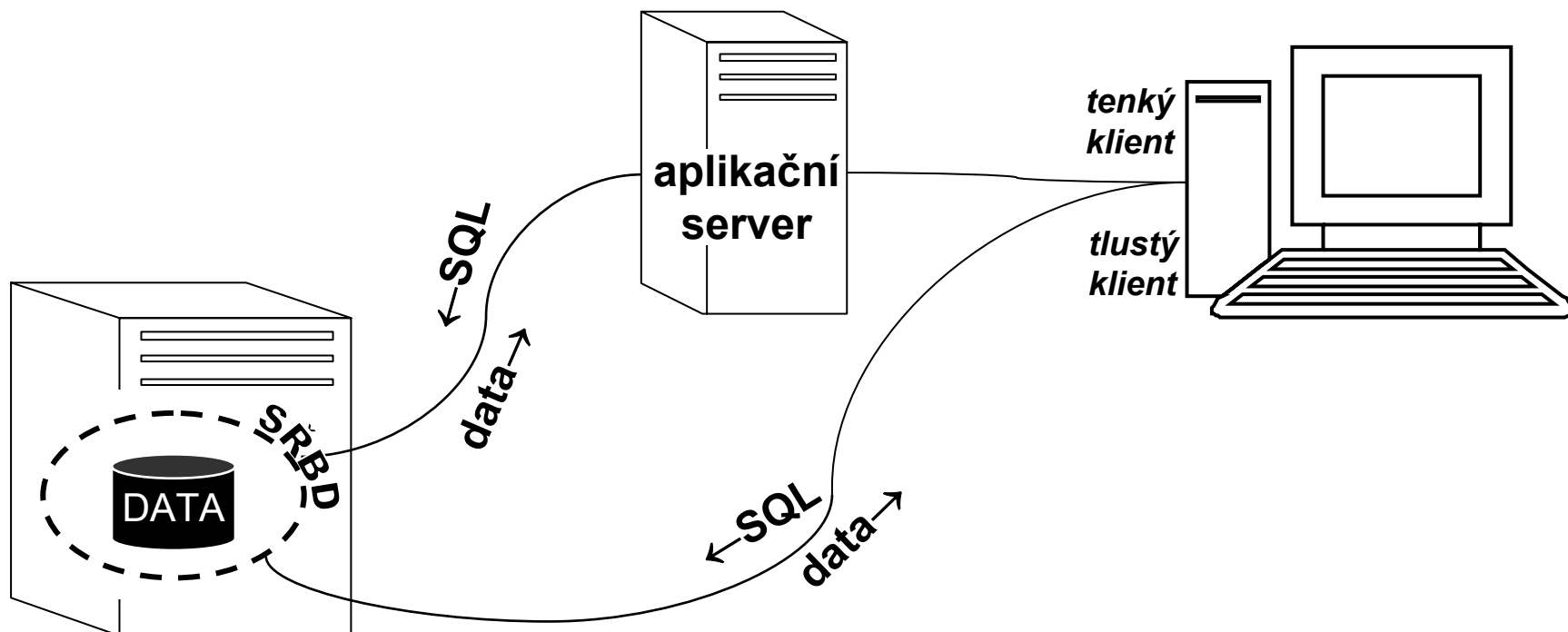
Role



Přístup k datům



Technologická architektura

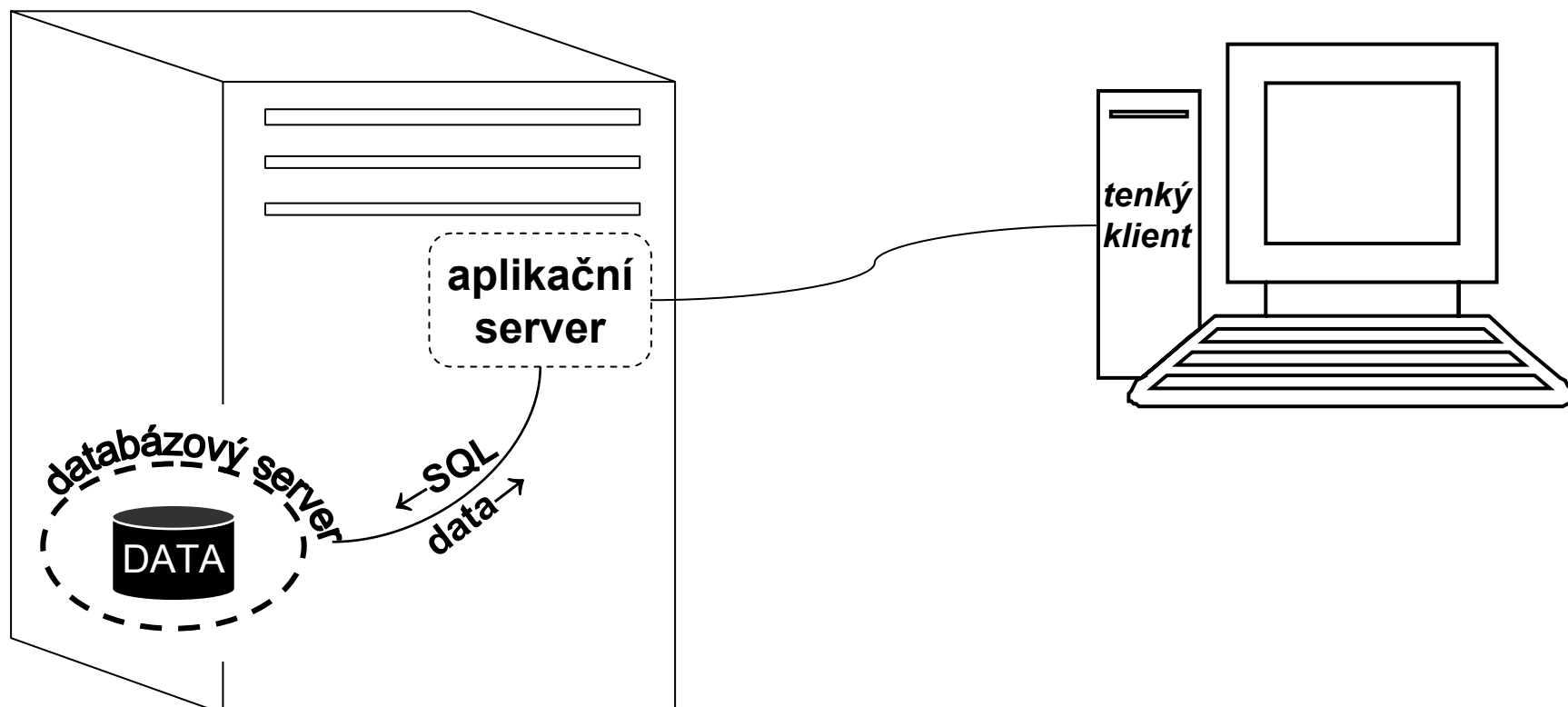


SŘBD (DBMS) – systém řízení báze dat (database management system)

– říká se mu databázový server

– např. IBM DB2, Oracle, Informix, MS SQL Server, Sybase, ...,

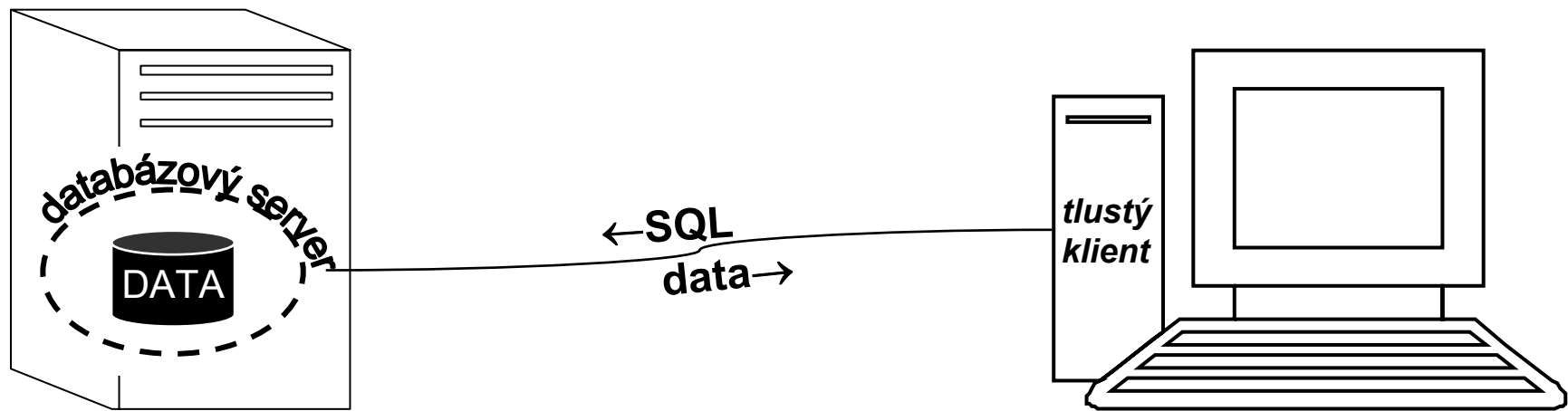
MySQL, PostgreSQL, Firebird ...



Tenký klient : např. web browser

Aplikační server: např. Apache s PHP

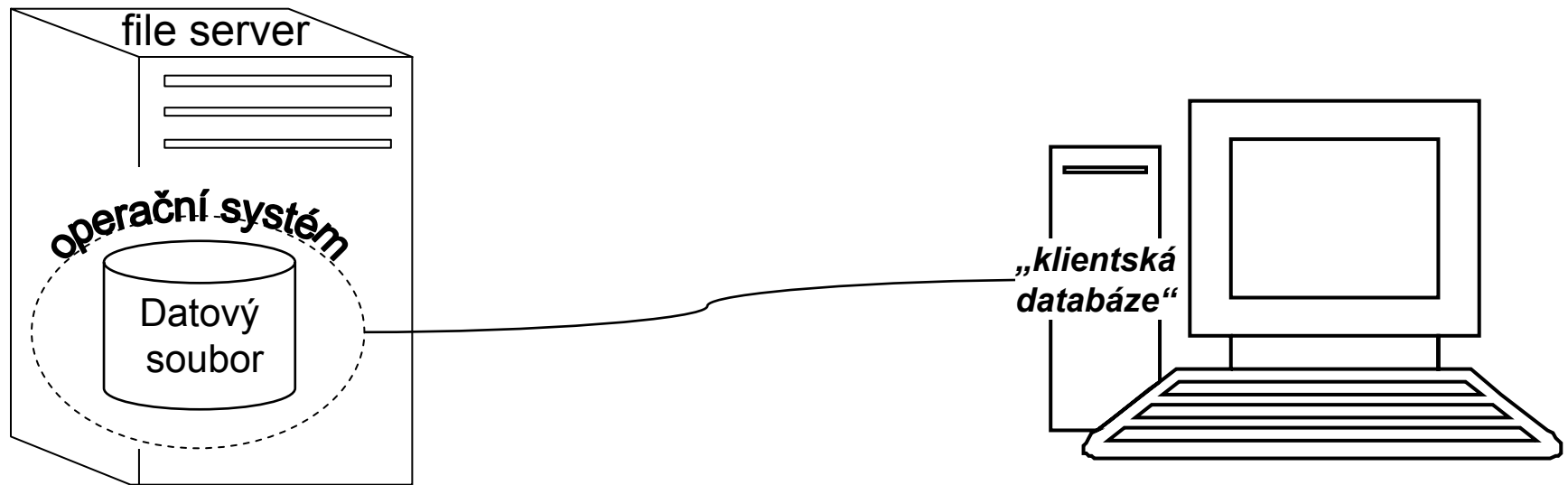
Databázový server: např. MySQL



Aplikační logika je v tlustém klientovi:

–programován v C, Javě ...,
nebo v proprietárním prostředí příslušném k db serveru,
nebo to může být i např. aplikace v MS Access.

Databázový server může být např. Oracle...



„Klientská databáze“ např. MS Access, Paradox, DBase

Databázový server vs. file server

- Přístup k jednotlivým datovým položkám záznamů, nikoli k celému souboru
 - názvy pro různé objekty databázové struktury
 - na strukturu je možno se dotázat...
- Specifická přístupová práva k jednotlivým typům záznamů pro různé uživatele
 - v rámci OS má k datovému souboru přístup pouze SŘBD

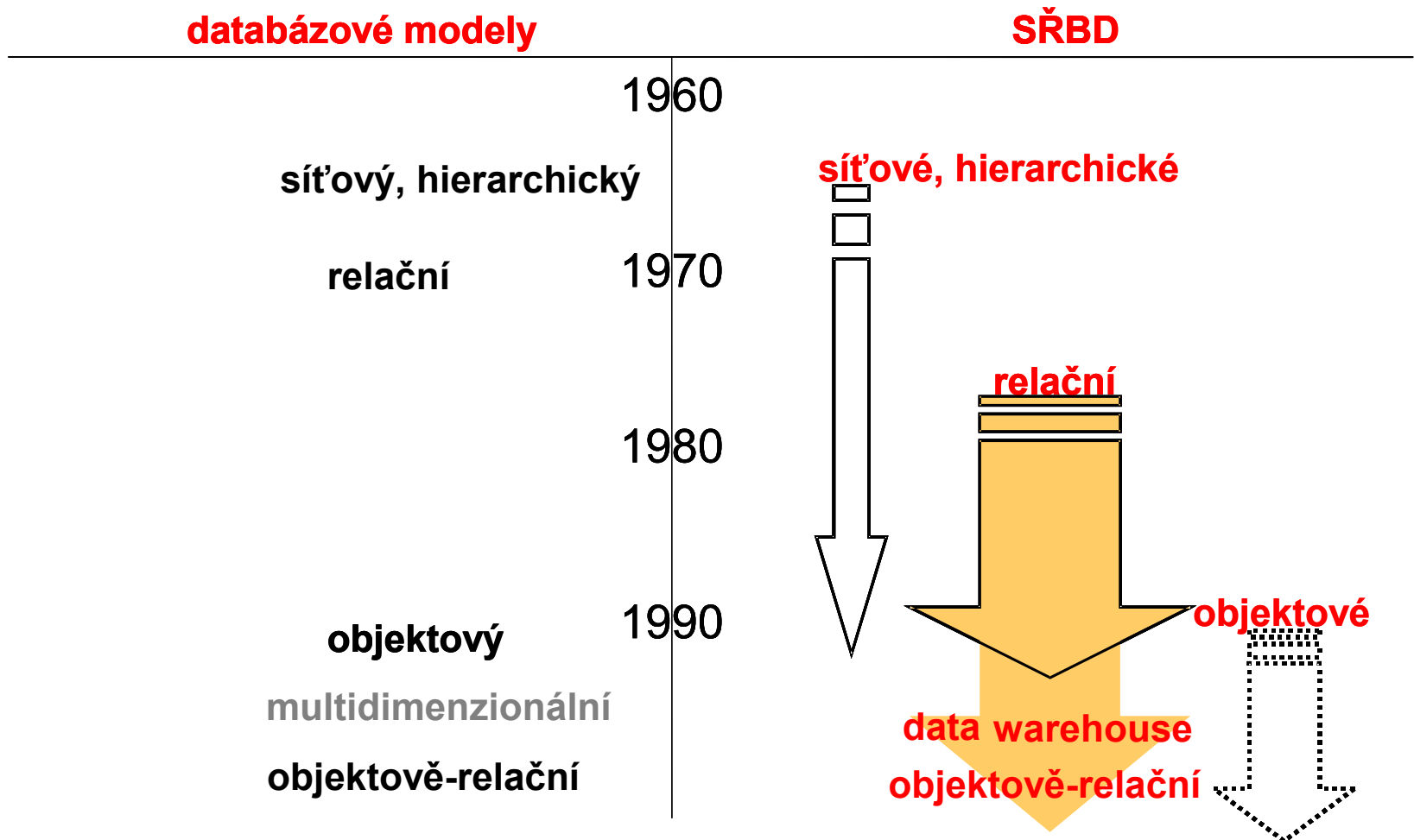
Databázový server vs. file server

- Zajišťování integrity a konzistence dat
 - Validace vstupu
 - Obnova po poruchách až k poslednímu konzistentnímu stavu
- Efektivní správa velkého objemu dat

Databázový systém

- (maximální) požadavky
 - sdílení dat, paralelní přístup
 - nezávislost aplikací na fyzickém uložení dat, logický přístup k datům, centrální popis dat
 - ochrana před neoprávněným přístupem před poruchami
 - kontrola konzistence dat
 - velké objemy dat

Různé modely pro databázovou strukturu



Databázová struktura obecně

- Definují se **typy záznamů**
 - výčet položek, jejich typů (domén) a jmen
- V typu záznamu může být určen **primární klíč**
 - skupina položek k jednoznačné identifikaci záznamu
 - užití v relačních a objektových databázích
- Záznamy mohou být **vyhledávány navigací** podle odkazů mezi záznamy
 - v síťových a objektových databázích
- Vyhledávat se může také **sekvenčně**
 - hierarchických databázích
 - když to nejde jinak

Indexy

- Jsou pomocné databázové struktury sloužící k **urychlení vyhledávání záznamů**
 - vyhledávání podle daných příznaků
 - podle primárního klíče
 - podle hodnot v dané položce či skupině položek
- SŘBD je udržuje v aktuálním stavu

Indexy

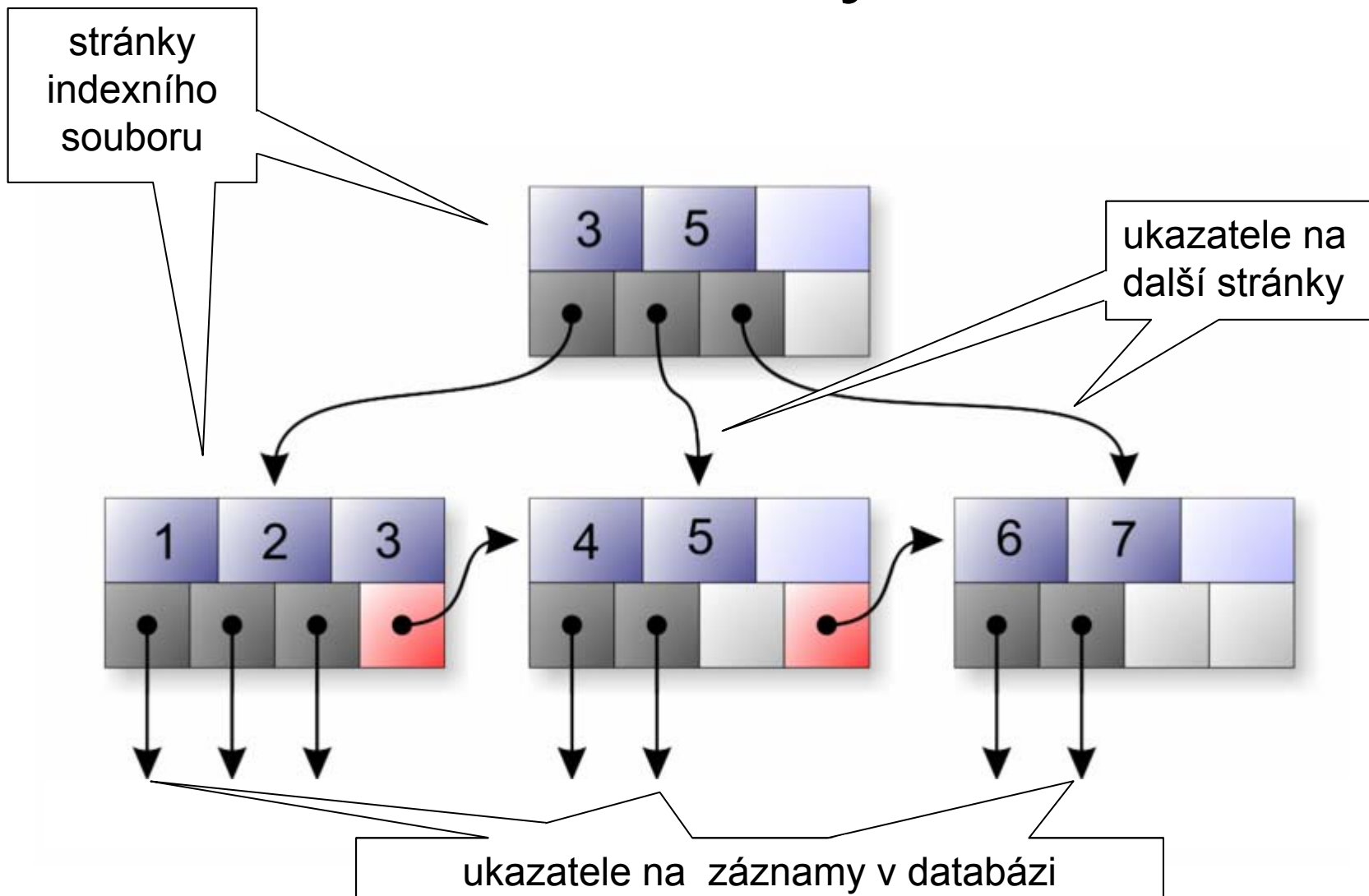
- Nejčastější způsoby organizace:
 - B-stromy
 - Bitové indexy
 - Hashované/clusterované indexy

B-stromy

- Jsou optimalizovány pro čtení stránek* při vyhledávání
- Mají nejlepší dynamické vlastnosti
 - při reakcích na aktualizace v databázi

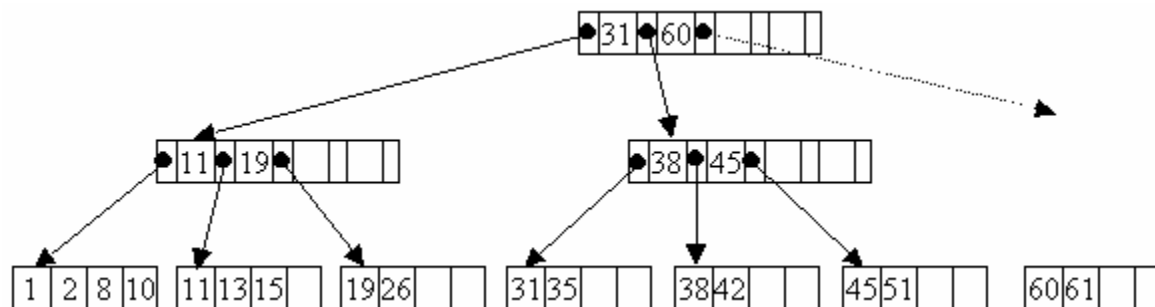
*) stránka je jednotkou čtení/zápisu z/do vnější paměti

B-stromy



B-stromy

- Na 1 stránce mohou být stovky klíčů
stačí 3 až 4 úrovně stromu
~ 3 až 4 čtení k nalezení záznamu



- Listy sekvenčně propojeny
⇒ usprádaní záznamů podle ind. klíče

Bitové indexy

- Vhodné pro malou doménu (množinu hodnot) vyhledávacího klíče
- Pro každou hodnotu vytvořen bitový vektor
 - pozicím odpovídají jednotlivé záznamy
- Možná i vhodná komprimace vektorů
 - rozbalování „od začátku“

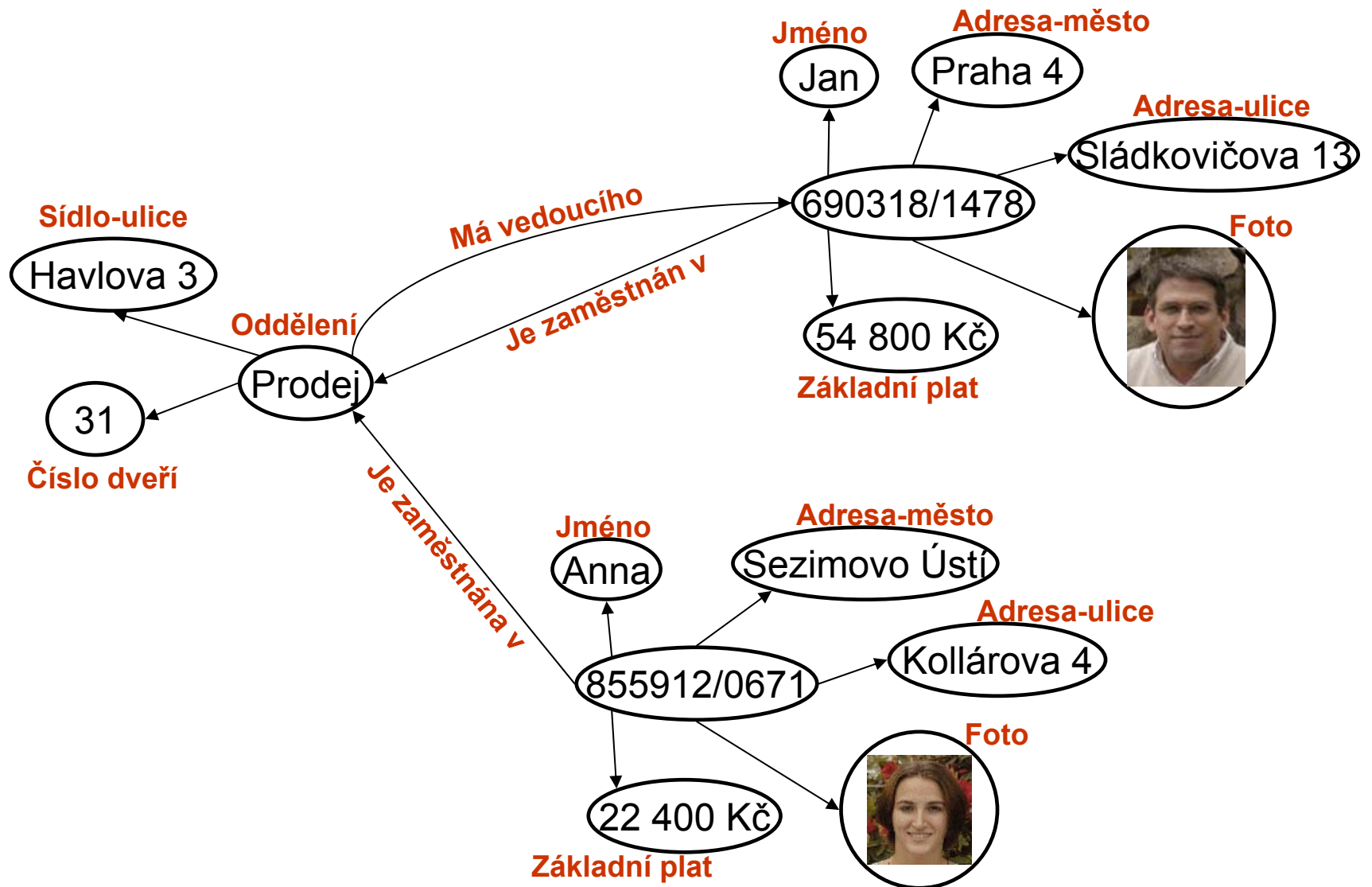
Hash indexy

- Hash funkce vypočítá adresu stránky uložení záznamu
 - Nutné značné rezervy pro ukládání záznamů
- Nejrychlejší „index“
- Můžeme mít nejvýše jeden hash index (clustrovaný index)
protože je to fyzické uložení záznamu

Databázové modely

- specifické způsoby organizace dat

Data a vztahy mezi nimi



Relační databáze

Zaměstnavec

RČ	Jméno	Adresa-město	Adresa-ulice	Základní plat	Foto	Oddělení
690318/1478	Jan	Praha 4	Sládkovičova 13	54 800 Kč		Prodej
855912/0671	Anna	Sezimovo Ústí	Kollárova 4	22 400 Kč		Prodej

Oddělení

Název	Sídlo-ulice	Číslo dveří	Vedoucí
Prodej	Havlova 3	31	690318/1478

*Relace,
relační tabulky*

Primární klíče

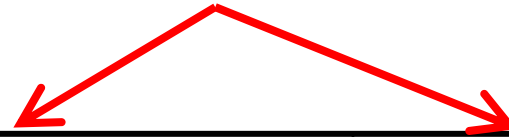
Cizí klíče

Relační databáze

- Data jsou organizována do tabulek
 - Jeden záznam = jeden řádek nějaké tabulky
 - Sloupce tabulky tvoří pole/položky záznamů
- Položky jsou atomické
 - nejsou složeny z částí, které by nesly nějaký význam (ve světě aplikace)
 - není v nich více údajů

Co jsou atomické položky?

Nejsou atomické:



Vysoká škola ekonomická	W.Churchilla 4, 130 00 Praha 3	224-095-111
Vysoká škola finanční a správní	Estonská 500, 101 00 Praha 10	210-088-800 271-741-597

Vnitřní struktura významů.

Více hodnot se stejným významem.

Svět aplikace

Plátce: Richter Supermarkt
Grenzacherweg 237
Ženeva 1203
Švýcarsko

Číslo obj.:	Zákazník:	Prodejce:	Objednáno:	Dodat dne:
11075	RICSU	Laura Callahan	6.5.1998	3.6.1998

Č. výrobku:	Název výrobku:	Množství:	Cena:	Sleva:	Výsledná cena:
2	Chang	10	475,00 Kč	15%	4 037,50 Kč
46	Spegesild	30	300,00 Kč	15%	7 650,00 Kč
76	Lakkalikööri	2	450,00 Kč	15%	765,00 Kč

Celkem : 12 452,50 Kč

Číslo objednávky	Zákazník	Zaměstnanec	Datum objednávky	Dodat dne
11074	SIMOB	King, Robert	06.05.1998	03.06.1998
11075	RICSU	Callahan, Laura	06.05.1998	03.06.1998
11076	BONAP	Peacock, Margaret	06.05.1998	03.06.1998

databázová struktura

Číslo objednávky	Výrobek	Jednotková cena	Množství	Sleva
11074	16	436,25 Kč	14	5,00%
11075	2	475,00 Kč	10	15,00%
11075	46	300,00 Kč	30	15,00%
11075	76	450,00 Kč	2	15,00%
11076	6	625,00 Kč	20	25,00%
11076	14	581,25 Kč	20	25,00%
11076	19	230,00 Kč	10	25,00%

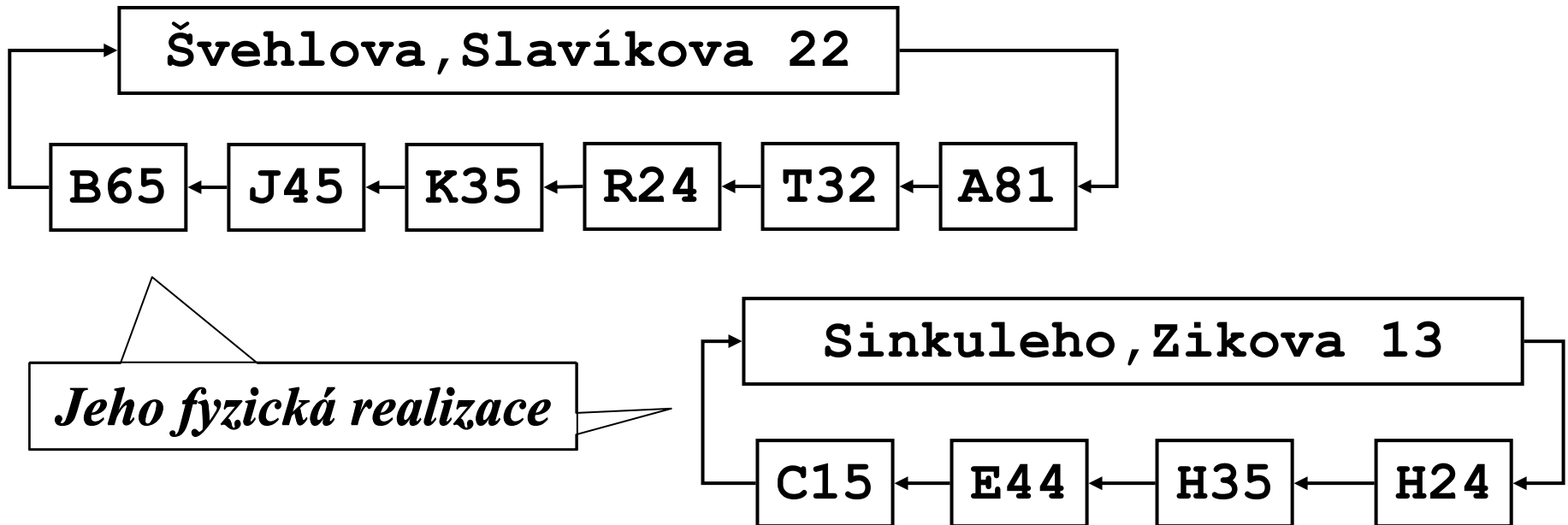
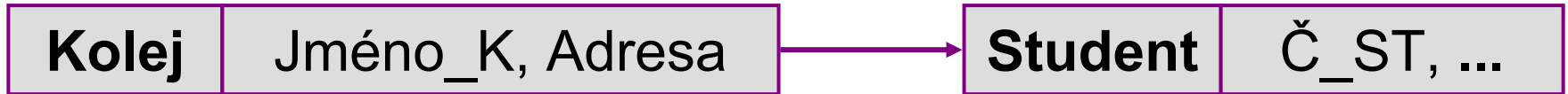
Číslo výrobku	Název výrobku	Dodavatel
1	Chai	Aux joyeux ecclésiastiques
2	Chang	Exotic Liquids
...		
46	Spegesild	Lyngbysild
...		
76	Lakkalikööri	Karkki Oy

Kód zákazníka	Firma	Adresa	Město	Země
RICAR	Ricardo Adocicados	Av. Copacabana, 267	Rio de Janeiro	Brazílie
RICSU	Richter Supermarkt	Grenzacherweg 237	Ženeva	Švýcarsko
ROMEY	Romero y tomillo	Gran Vía, 1	Madrid	Španělsko

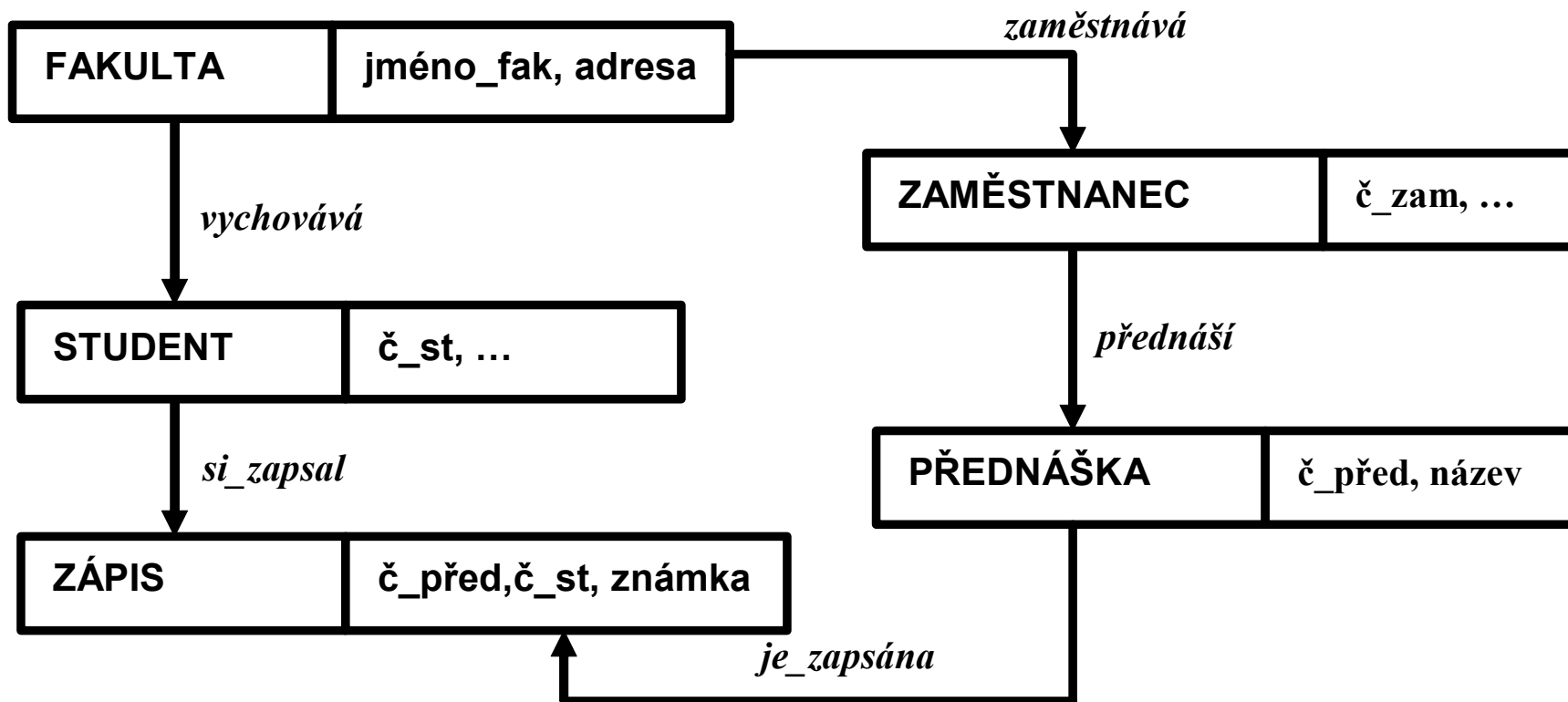
**relační
databáze**

Síťové databáze

Typ vztahu ve schématu:

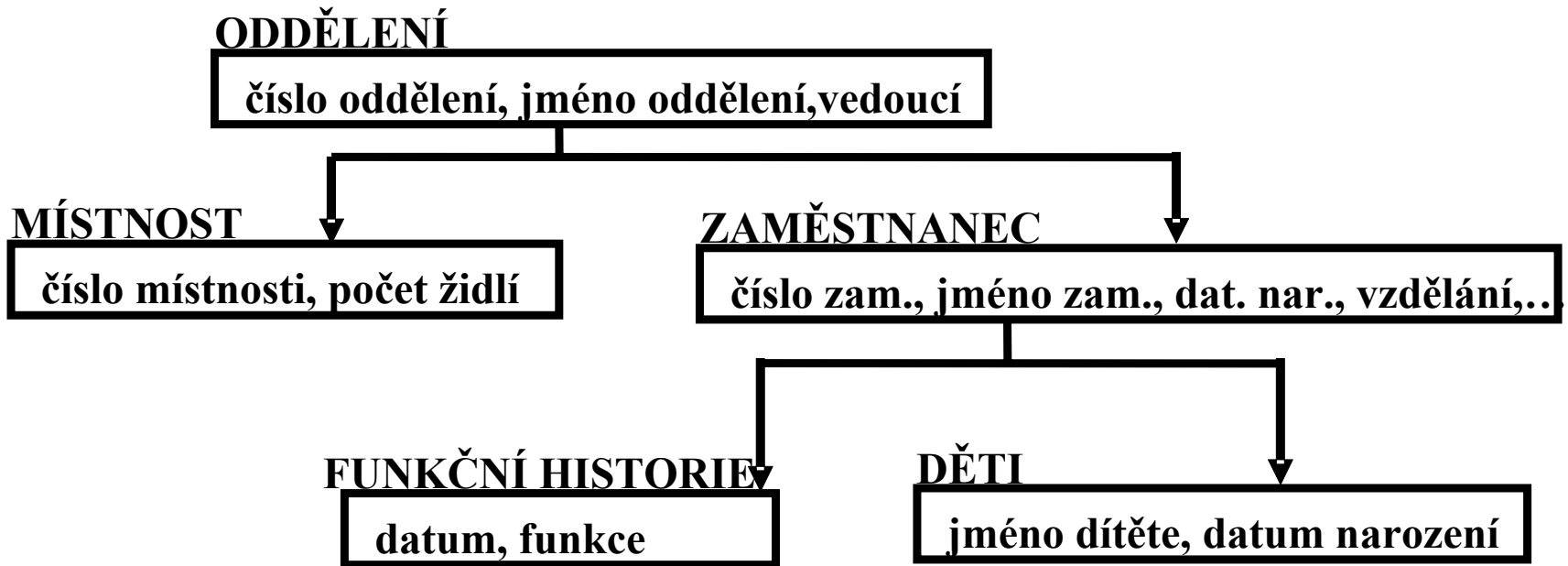


Příklad schématu síťové databáze



Pro každý typ vztahu jsou vytvořeny kruhové seznamy od master recordů, takže např od každé fakulty máme dva kruhové sezamy.

Hierarchické databáze



03	PROGRAMOVÁNÍ	Novák Jiří	
	324	4	
	322	10	
	321	5	
	1321	Albl Antonín 25.1.1953	vš
		1.1.1983 programátor	
		Jiří 28.4.1981	
		Eva 6.2.1980	
		Jan 5.1.1978	
	1239	Kusý Jan 31.12.1938	SŠ
		1.8.1983 programátor	
	1234	Novák Jiří 27.5.1950	vš
		1.1.1984 vedoucí	
		15.5.1983 programátor	
		1.7.1982 operátor	
		Eva 12.8.1982	

Takto vypadají záznamy:

Objektové databáze

- ODMG 3.0 norma (2001):
 - Adoptován je objektový model OMG
 - objekt, typ/třída - objekty stejné třídy mají mít stejné **chování** a stejnou **množinu stavů**
 - metody, atributy, přidány jsou **vztahy** -vazby na jiný objekt
 - Je definován **ODL**
 - Je definován **OQL**
- Jsou vytvářeny „**bindingy**“ pro programovací jazyky
 - příkazy pro persistenci dat nativní v příslušném jazyce

ODL

```
CLASS Person
{
  SRING Name,
  DATE Birthdate,
  ...
  INT Age,
  REF <Apartment> Lives_in
      INVERSE Apartment::IS_used_by
}
```

také metody,
konstruktor

Referenční vztahy,
integrita

1:1

- jiné konstrukty SET, BAG, ARRAY, STRUCT, LIST

```
SET<REF<Apartment>>
```

1:n, n:m

OQL

```
SELECT c.Lives_in.Building.Address  
FROM p IN Person, c IN p.Children
```

...výsledkem je Bag

porušení zapouzdření

```
SELECT STRUCT( me:p.Name,  
my_address:p.lives_in.building.address)  
FROM p IN Person  
WHERE ...
```

porušení zapouzdření

...výsledkem je struktura

- Do `SELECT` možno i Zahnízdování struktur, i metody...
- **Porušení zapouzdření** možno v **ad-hoc** dotazech.

Co je O-R mapping

- Vrstvy mezi OO aplikací a SQL databází
 - podpora myšlení v objektech
 - ev. cache objektů, zodpovědné za persistenci
- Co potřebujeme:
 - INSERT,UPDATE,DELETE – metody **proxy objektů**
 - SELECTy
 - Typové
 - Navigační na malém počtu objektů
 - Přehledové/rozsáhlé - **lépe SQL přímo**
 - Ad-hoc - **lépe SQL přímo**

Porovnání R,OO,O-R

- Relační db – výkon na tradičních datech, výkon na rozsáhlých dotazech
- O-R mapping –
 - programátorské pohodlí,
rychlý, udržitelný vývoj aplikací
- OO db – výkon na netradičních datech
 - slabší v „databázových“ rysech

Objektový přístup v databázích

- Objektová identita
- Složité objekty
- Abstraktní datové typy
- Chování dat

Objektová identita

- Záznamy mají/mohou mít OID
 - v relační db mohou být použity „jako primární klíče“
 - k odkazům z jiných záznamů
- V jiných záznamech : atribut typu REF – odkaz, ukazatel
 - jeho obsahem je OID nějakého záznamu
 - nelze s ním manipulovat jako s hodnotou, ale jako s odkazem

Referenční integrita a OID

- Slave records
 - INSERT (u REF není třeba hlídat)
 - UPDATE (u REF není třeba hlídat)
- Master records
 - UPDATE (OID není možno měnit)
 - DELETE
 - Nemaže se
 - Maže se ale není to kritické – použití utilit na řešení odkazů doprázdna
 - Maže se a je to kritické – nutno definovat vztah FOREIGN KEY

Jediný důvod pro hlídání referenční integrity

Reference a dereference v SQL

navigační dotazování

Tabulka Ucty má pole
Vlastnik typu REF do tabulky Klient,
tabulka Klient má pole
Jmeno
a pole Adresa strukturovaného typu:

```
SELECT U.Vlastnik.Jmeno  
FROM Ucty U  
WHERE U.Vlastnik.Adresa.Mesto="Pardubice"  
AND U.Zustatek>100000;
```

Objektové typy v SQL databázi

- Složité objekty

id	contact		lineitems		
	name	phone			
			item	qty	unit_price

možno použít jako řádkový typ při CREATE nové tabulky,
lineitems mohou být logicky dostupné v „nested table“ (ORACLE)

Objektové typy v SQL databázi

- Metody příslušného objektového typu v definici typu
 - Těla metod v uložených procedurách a funkcích

```
SELECT Formalni(Jmeno)
FROM Ctenar
WHERE Vek(Ctenar)>=40
```

je nejistá optimalizace dotazů

Datové typy

- Validace vstupu
- Operace, funkce, metody
- Význam, užití
- Objem, čas
 - uložení
 - zpracování operací
 - vyhledávání, indexace

Základní datové typy

číslo
množství, počet

přesná – pevná desetinná čárka

měna

nepřesná – plovoucí desetinná čárka

výpočty

text dané maximální délky
název, jméno, kód

zobrazení, porovnání, slepení
manipulace po písmenech

datum a čas
kdy

co bylo/je/bude dříve, výpočet intervalu
Year, Month, Day

logická hodnota
ano/ne

logické výrazy

typ vyjmenovaných hodnot
zařazení do nějaké kategorie

vyhledání, třídění

Datové typy v SQL

Základní datové typy

- číslo
- textový údaj
- datum a čas
- ano/ne (logický)

Nové datové typy

- bitové řetězce
- volný text
- obrázky, audio, multimedia
- prostorová data
- časové intervaly
- ...

+ „uživatelsky“ definované typy

UDTs and UDFs...

- Existují **balíky**, které je možno koupit
 - DataBlades
 - Cartridges
 - Extenders
- Jejich výhodou je **indexace** takových typů v SŘBD

Odlišující typy v SQL

- Emulace domén
 - přínos prostřednictvím strong typing

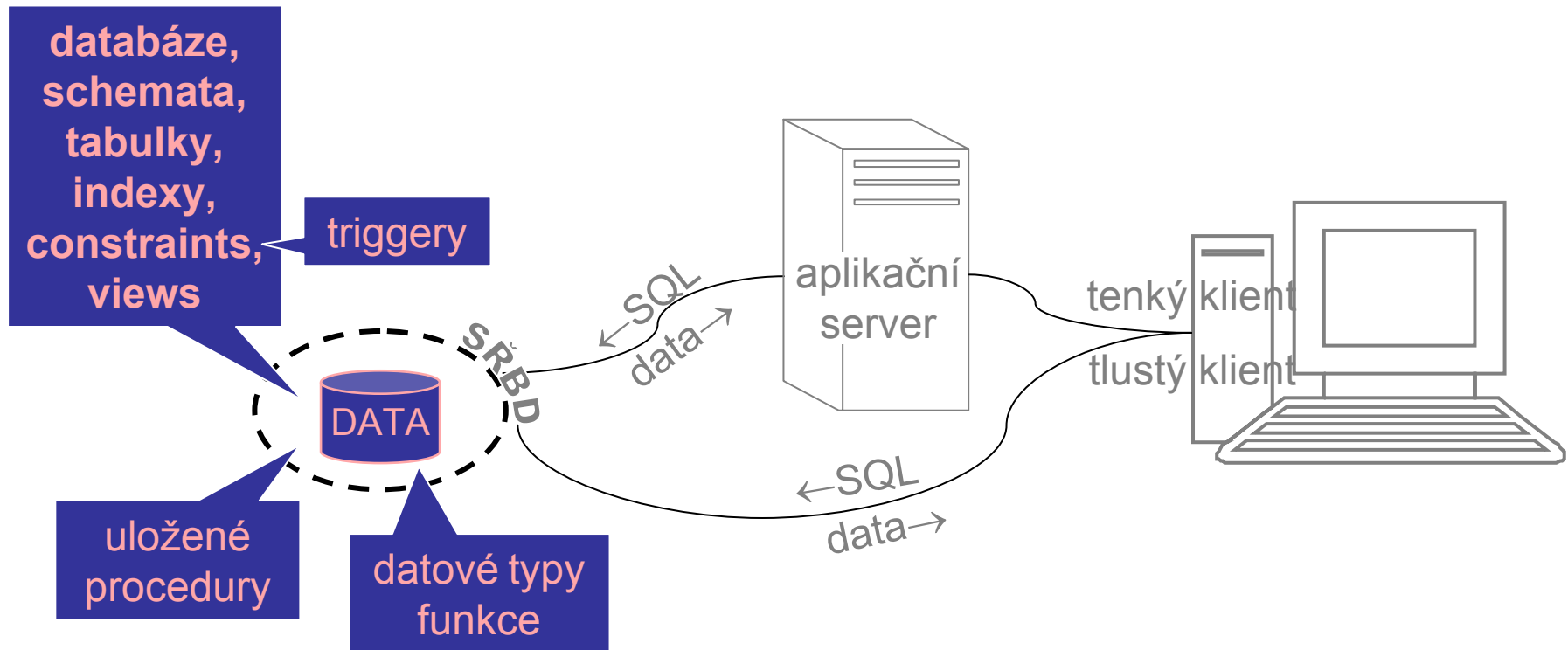
Chování dat v SQL databázi

Idea:

„Databáze = data + aplikace datům vlastní“

- K tomu slouží
 - Triggery
 - Uložené procedury

Objekty SQL databáze



Triggery

Účel:

- Integritní omezení
- Odvozené datové objekty
 - Repliky, archivy, vypočtená pole
- Akce ven
 - E-mail, zaslání objednávky...

Množství možných triggerů na jednu tabulku
je omezené !

Struktura triggeru

událost INSERT|UPDATE|DELETE

před/po BEFORE/AFTER

tabulka ... (u UPDATE **výběr polí**)

...**jméno pro referenci** na
vkládaný/měněný/mazaný řádek

podmínka ...

akce ...

pro FOR EACH ROW | STATEMENT

Integritní omezení triggerem - příklad

Titul (ISBN, název, ...)

Svazek (signatura, titul → Titul.ISBN, ..., umístění)

Vypujcka (ctenar, svazek, od, do)

Rezervace (ctenar, titul, od, do)

Titul není možno rezervovat, když existuje volný svazek tohoto titulu.

pokračování příkladu

událost: INSERT do tabulky **Rezervace**

podmínka: „existuje volný svazek“

```
EXISTS SELECT * FROM Svazek s
LEFT OUTER JOIN Vypujcka v ON
(v.svazek=s.signatura AND v.do IS NULL)
WHERE s.titul=NEW.titul
AND v.svazek IS NULL
```

akce: vyvolat chybu, vrátit „nepovolit“

...málo efektivní

pokračování příkladu

Svazek (signatura, titul → Titul.ISBN,..., umístění,
je_volný:A/N)

podmínka: „existuje volný svazek“

```
EXISTS SELECT * FROM Svazek s WHERE  
s.titul=NEW.titul  
AND s.je_volný
```

...trigger na údržbu odvozeného pole je_volný

pokračování příkladu

událost: UPDATE pole do v tabulce **Vypujcky**

podmínka: -

akce: „u toho svazku napiš, že je volný“

```
UPDATE Svazek SET je_volny=T WHERE  
Svazek.signatura=NEW.svazek
```

pokračování příkladu

událost: INSERT do tabulky **Vypujcky**

podmínka: -

akce: „u toho svazku napiš, že není volný“

```
UPDATE Svazek SET je_volny=F WHERE  
Svazek.signatura=NEW.svazek
```

Uložené procedury

- Jsou spravované db serverem
 - Přístupová práva
- Zkompilované, prováděné db serverem
 - Rychlost, sdílení paměti
- Logika aplikace je na jediném místě
 - Snadná údržba (opravy, aktualizace)
 - Může programovat nejlepší programátor, jednou
- Lze omezit přístupová práva k datovým objektům
 - Bezpečnost, spolehlivost přístupu

Uložené procedury

- Zatěžují DB server
- Vhodné jsou „datově intenzivní procesy“

Uložené procedury

Užití:

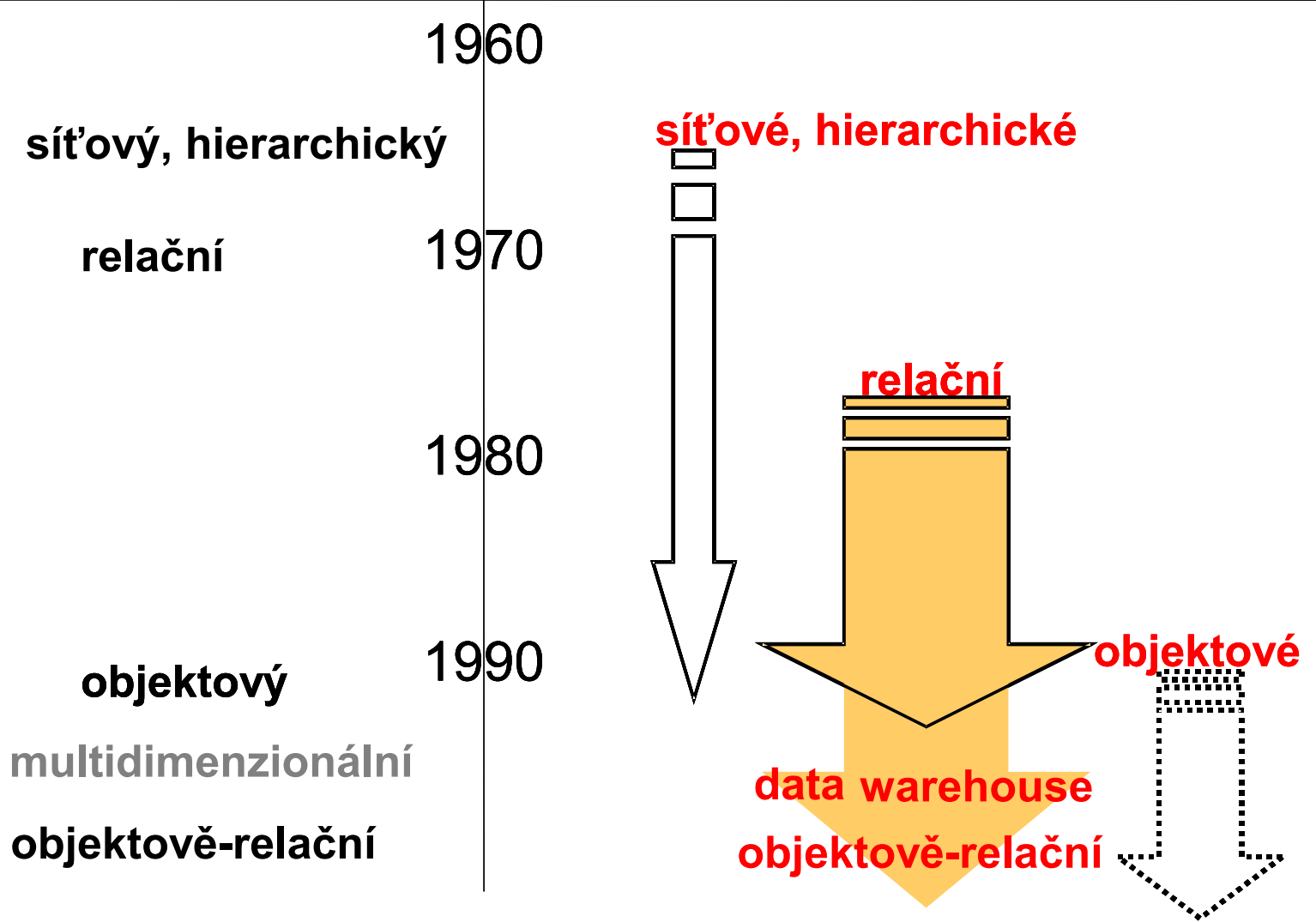
- Jednotlivé typové SELECTy
 - (proti nebezpečí neoptimalizovaných SELECTů)
- Těla triggerů
- Údržbové procedury

Uložené procedury

- Napsané v SQL
 - Od verze 1999 je SQL úplný programovací jazyk
 - Proměnné...
- Napsané v jiném programovacím jazyce
 - Db server musí mít kompilátor toho jazyka

databázové modely

SŘBD



Trendy v databázových modelech

- Fuzzy dotazování
- Prostorová data
- XML databáze
- Temporální databáze, verzování

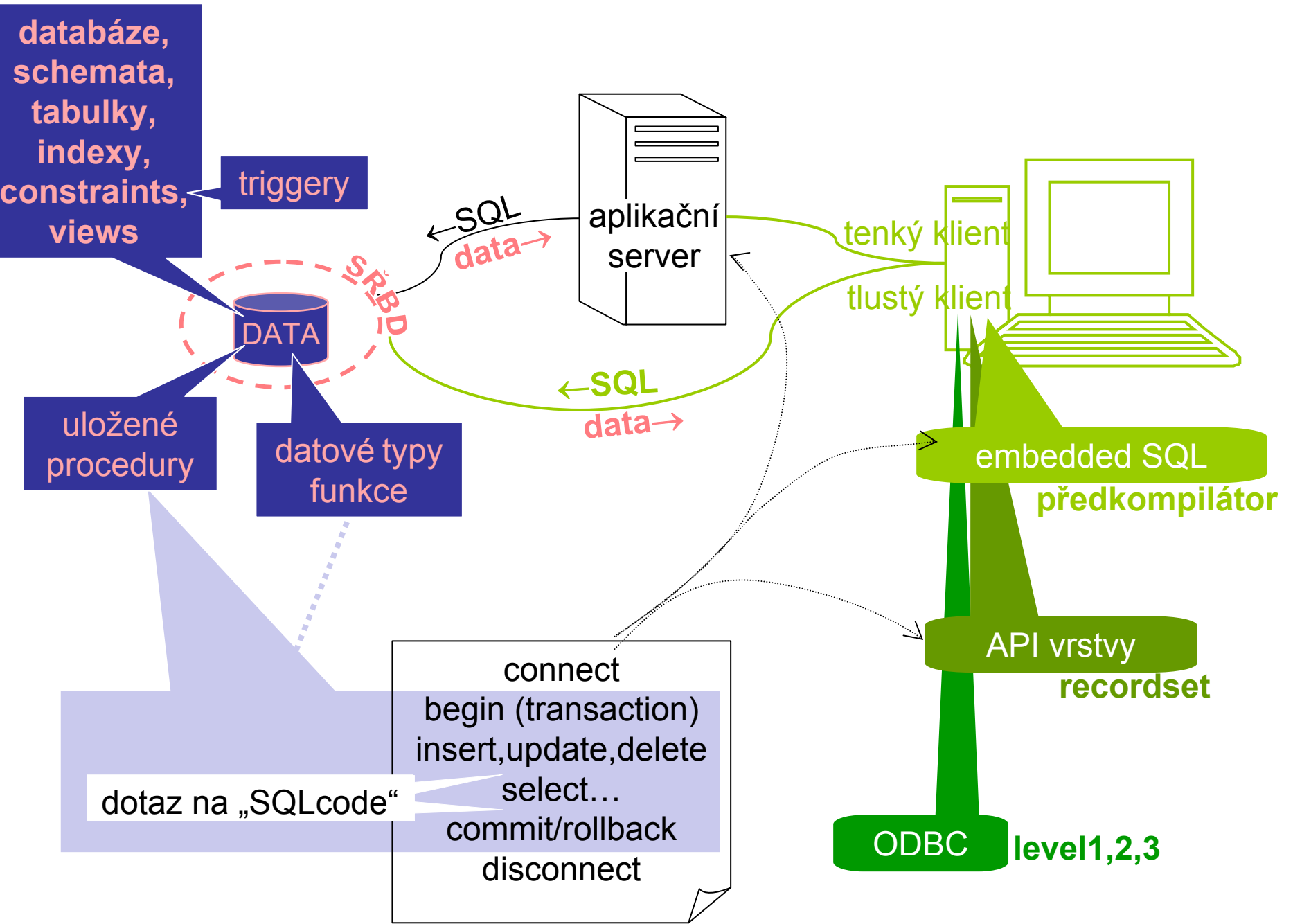
...DALŠÍ ČÁSTI SQL

- **Recursive SQL SELECT**

```
WITH RECURSIVE Reachable_From (Source, Destin) AS  
  (SELECT Source, Destin FROM Flights  
   UNION  
   SELECT in.Source, out.Destin  
   FROM Reachable_From in, Flights out  
   WHERE in.Destin=out.Source)  
SELECT * FROM Reachable_From
```

***Podobně:**
Nalezněte
nejlevnější
spojení z
jednoho města
do druhého!*

Architektura databázových aplikací



„SQLcode“ ~ informace o úspěšnosti příkazu

Příklad v PostgreSQL

```
dbtest=> begin;
```

```
BEGIN
```

```
dbtest=> insert into pokus values ('eee');
```

```
INSERT 38414 1
```

```
dbtest=> update ucty set stav=-100 where  
        cislouctu=1230;
```

```
ERROR:  new row for relation "ucty" violates  
        check constraint "$1"
```

```
dbtest=> insert into pokus values ('fff');
```

```
ERROR:  current transaction is aborted,  
        commands ignored until end of transaction  
        block
```

```
dbtest=> rollback;
```

```
ROLLBACK
```

```
dbtest=> select * from pokus;
```

```
 a
```

```
-----
```

```
aaaa
```

```
(1 row)
```


Deadlock

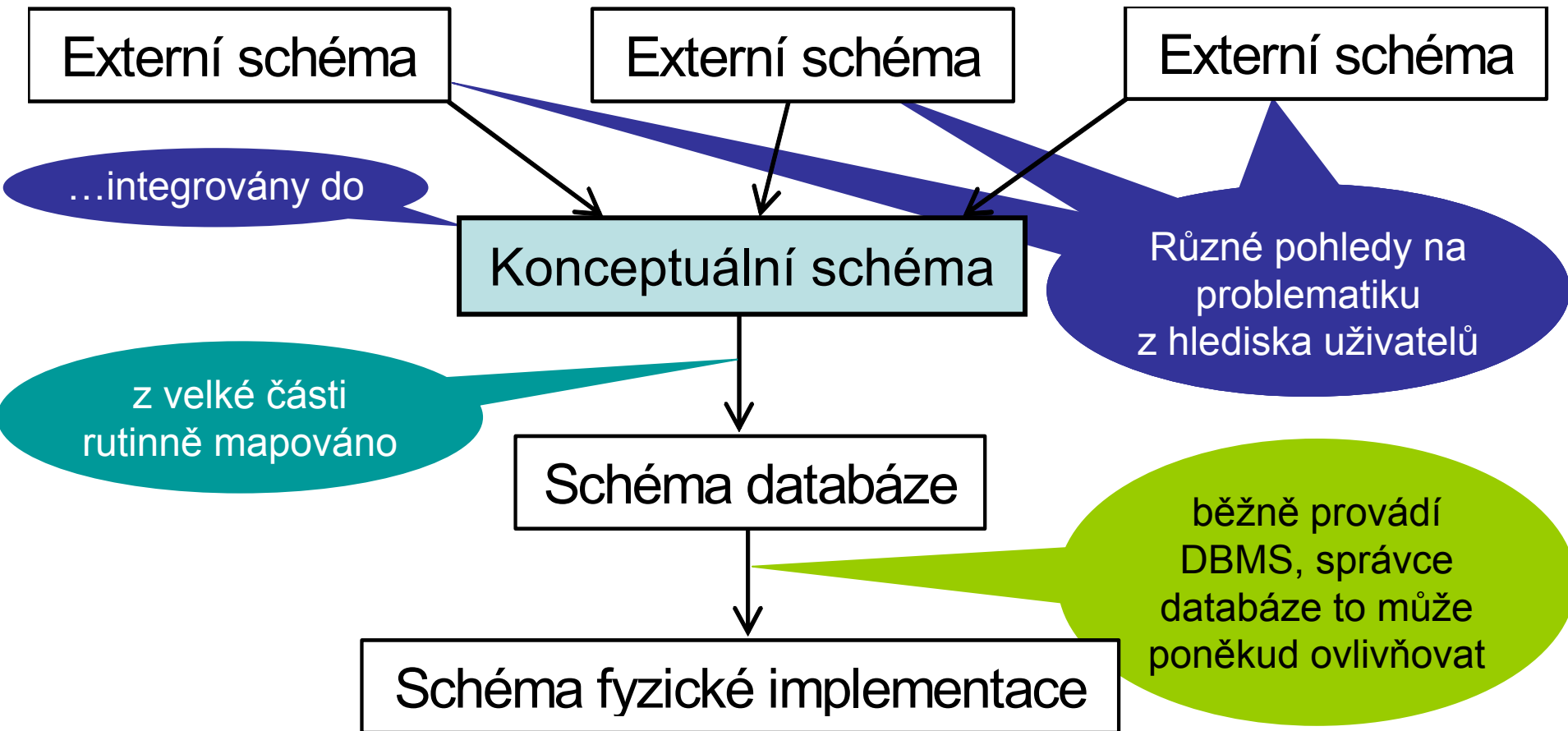
```
begin;  
update ucty  
  set stav=stav-100  
  where cislouctu=501;
```

```
update ucty  
  set stav=stav+100  
  where cislouctu=1230;
```

```
begin;  
update ucty set  
stav=stav+200 where  
cislouctu=1230;  
update ucty  
set stav=stav-200  
where cislouctu=501;
```

Design databázového schématu

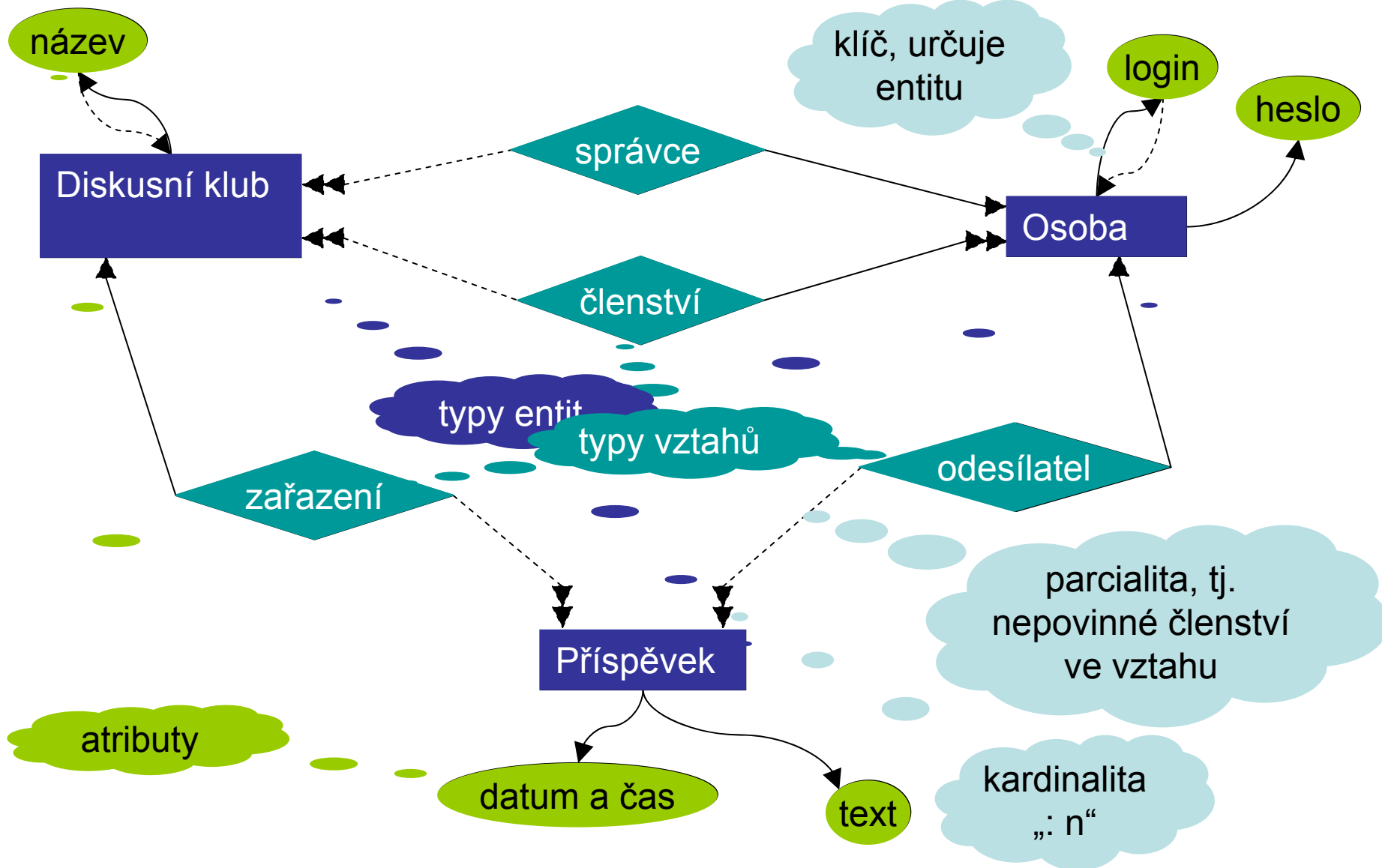
ANSI/SPAC 1975



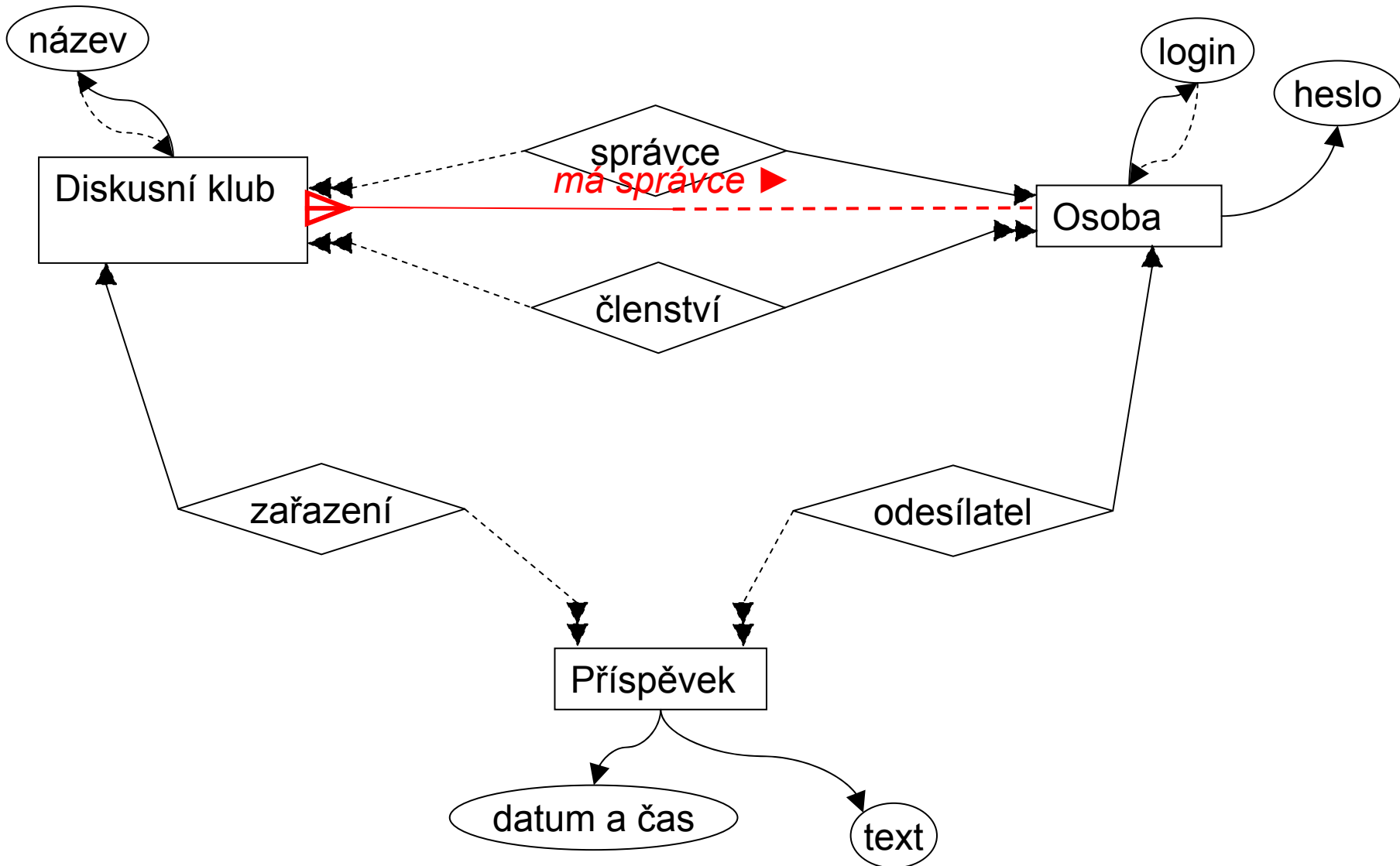
Proč konceptuální model

- Srozumitelný a věcně správný model, na jehož základě bude databáze navržena
 - *srozumitelný, přehledný*
 - *vše podstatné*
 - *věcně správný*
- Společný základ pro chápání objektů aplikace uživateli, analytikem, správcem databáze i programátory
- ✓ Dokumentace

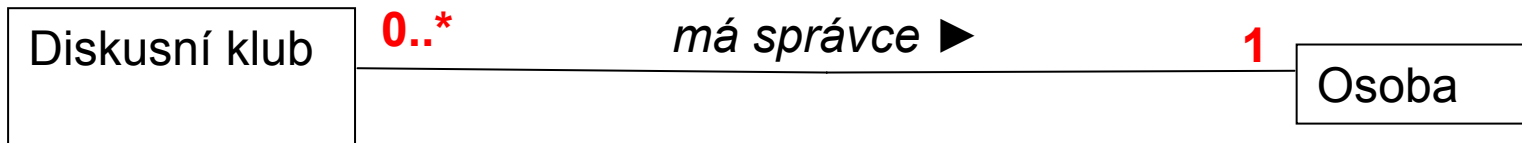
Značení P. P. Chena



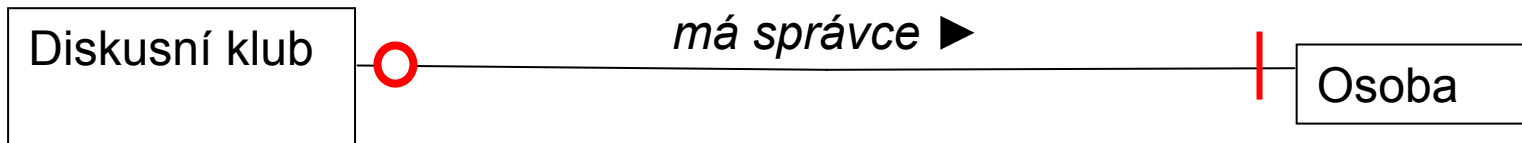
Jiná značení...



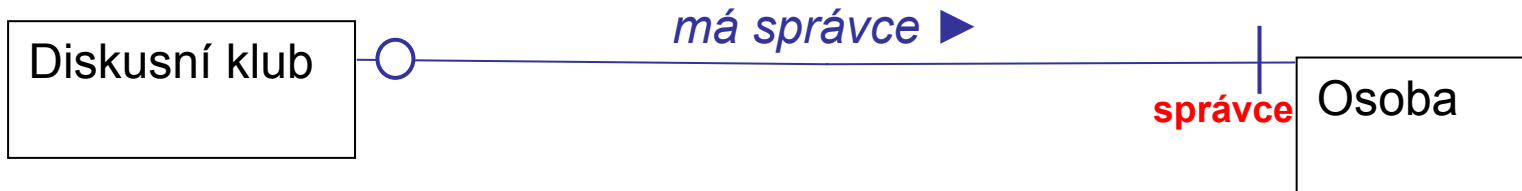
Jiná značení...



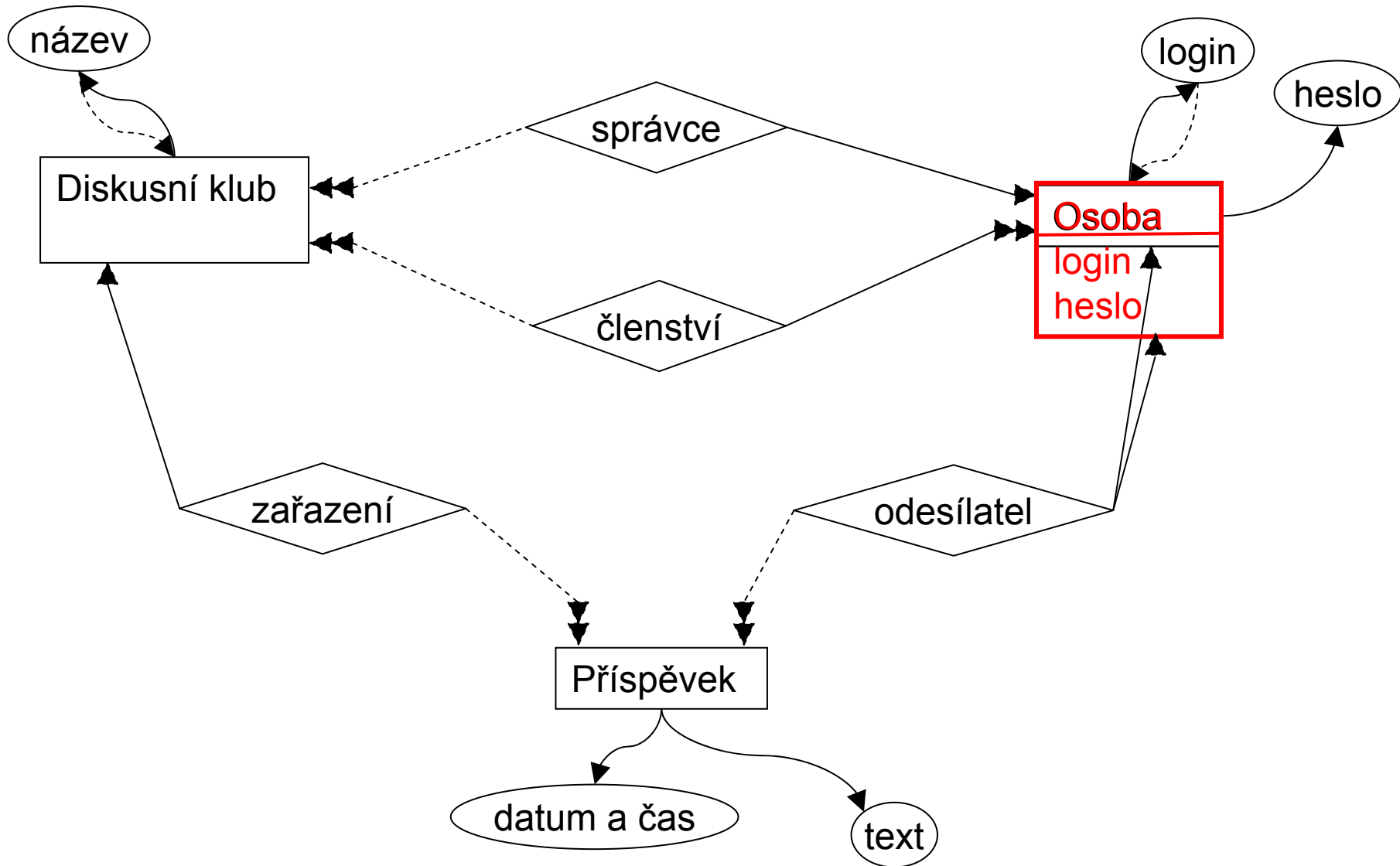
Jiná značení...



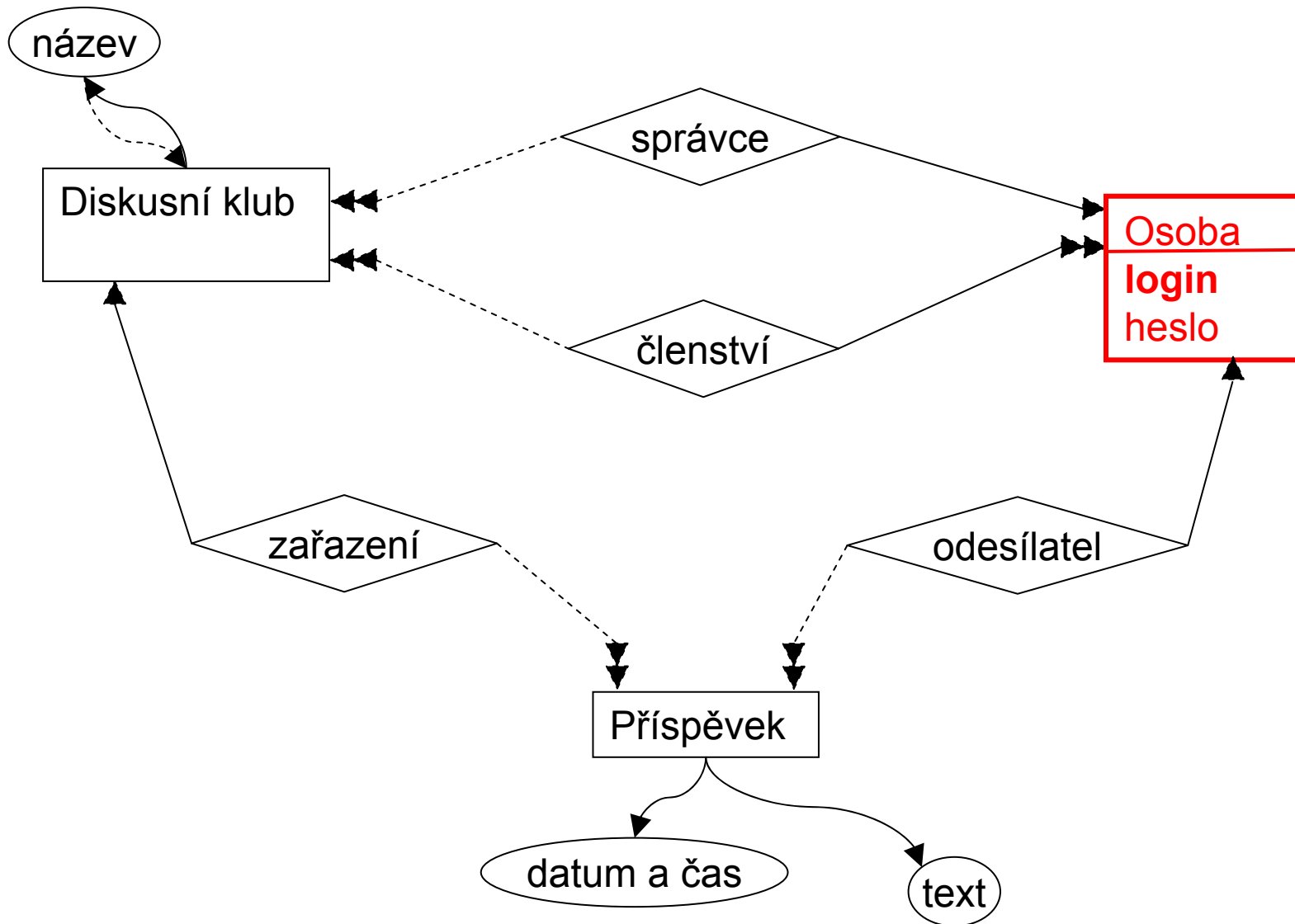
Jiná značení...



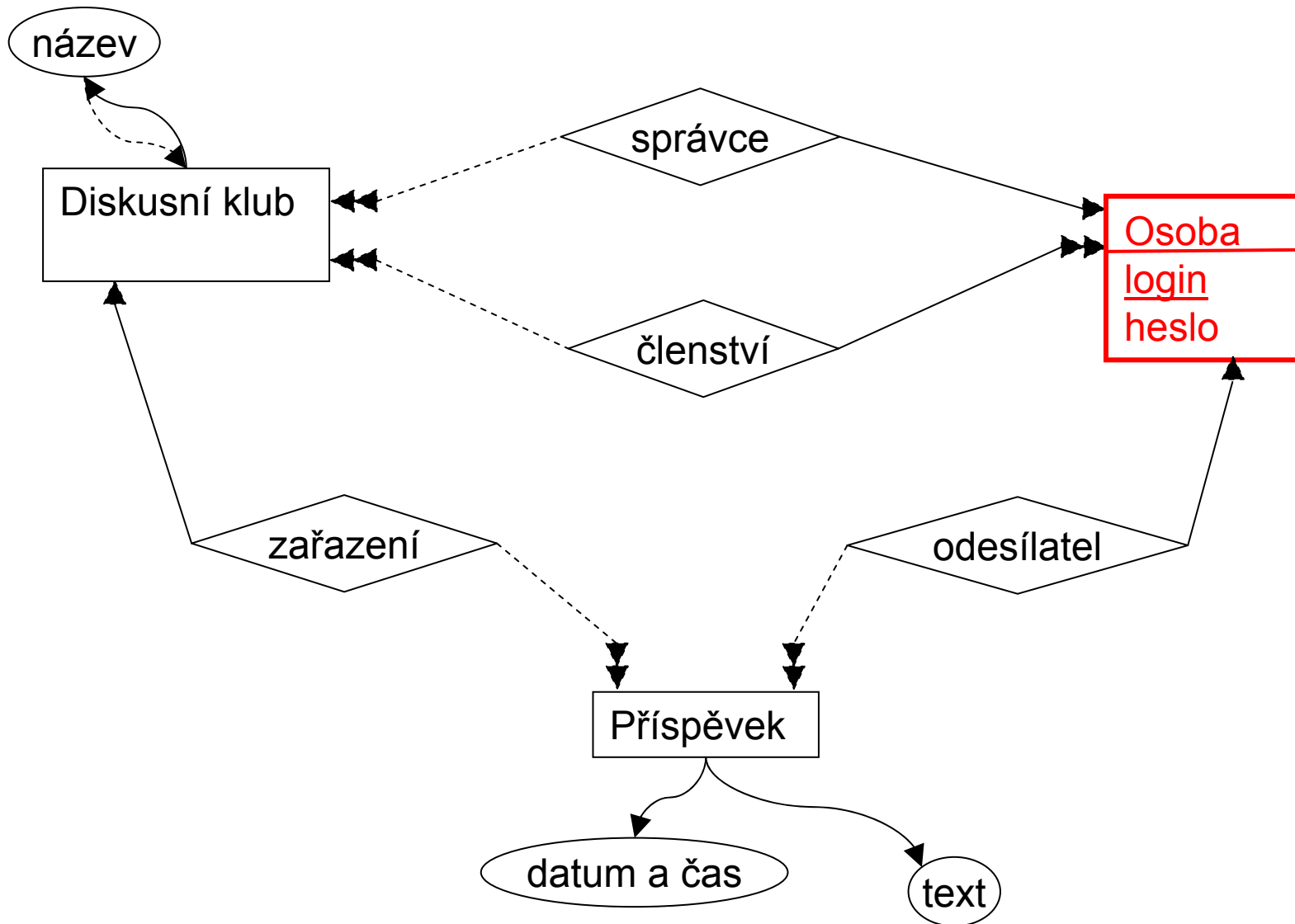
Jiná značení...



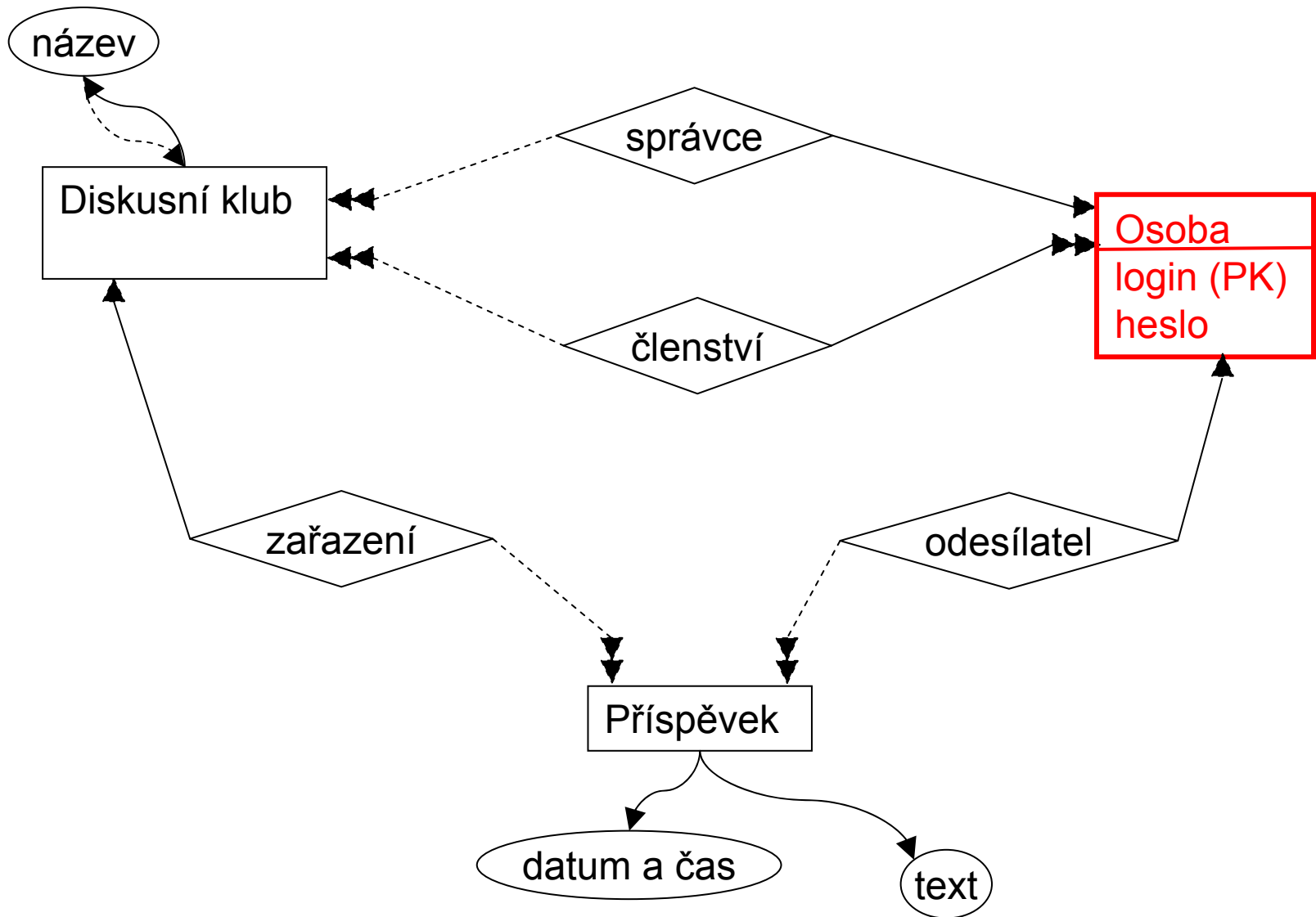
Jiná značení...



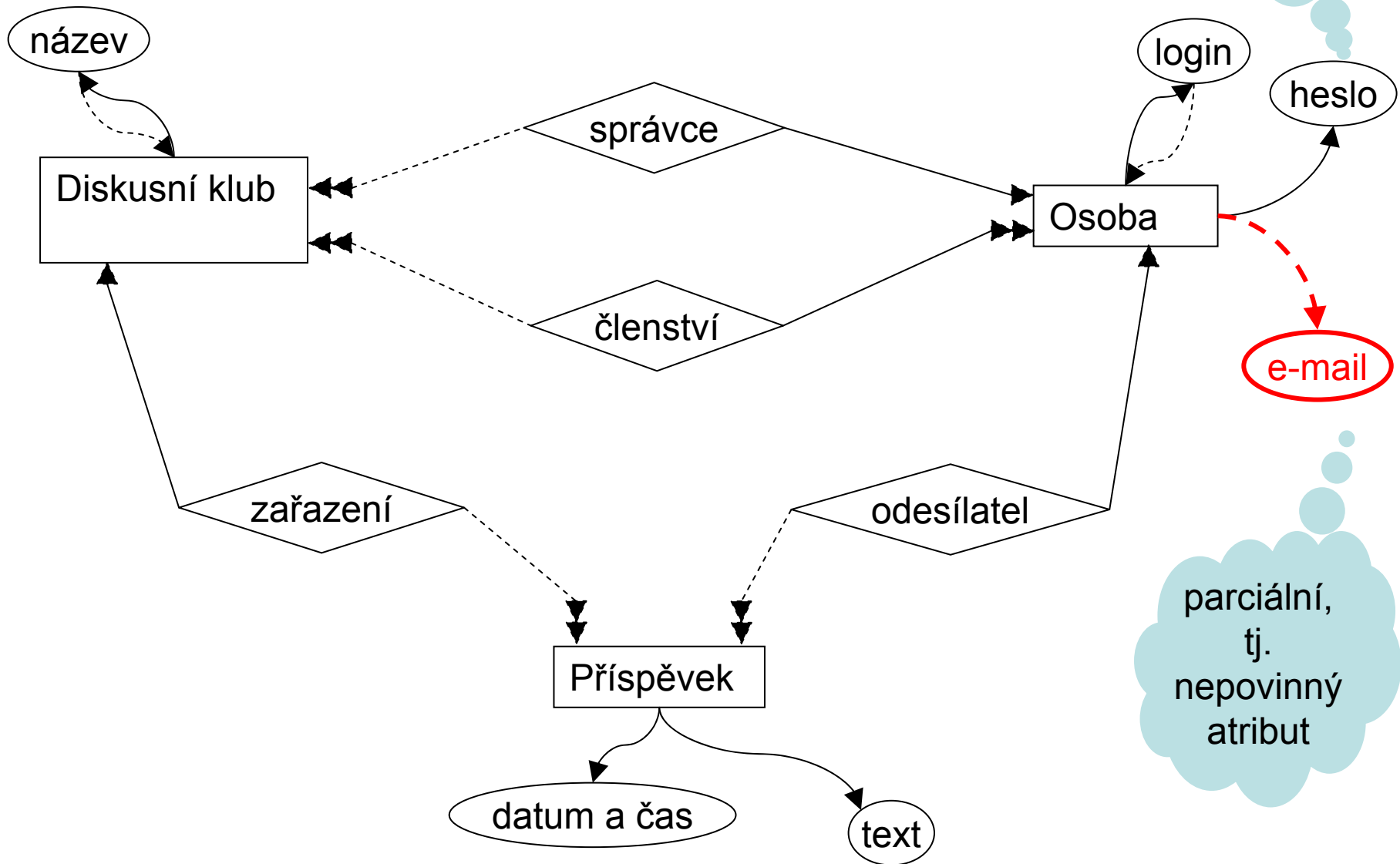
Jiná značení...



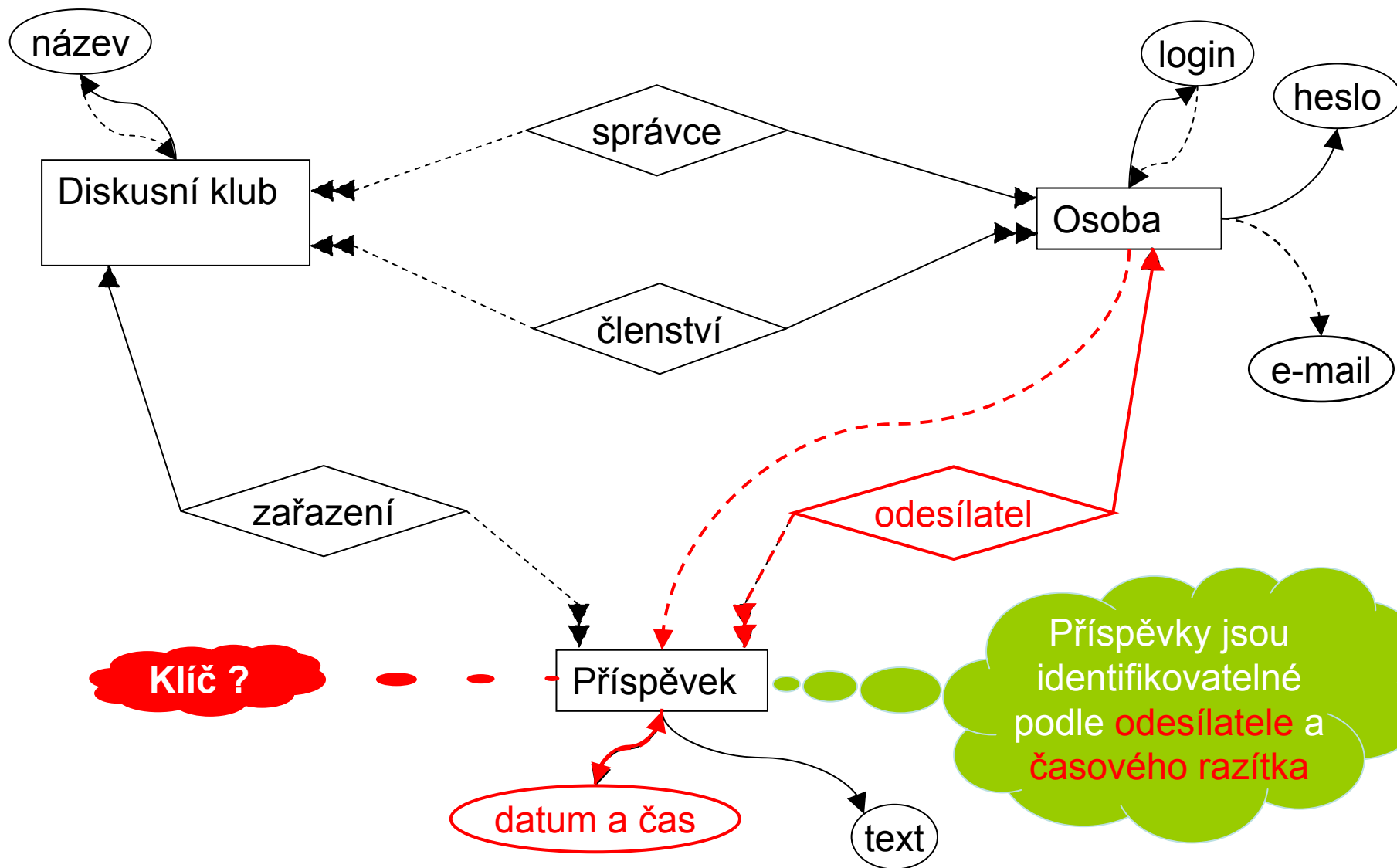
Jiná značení...



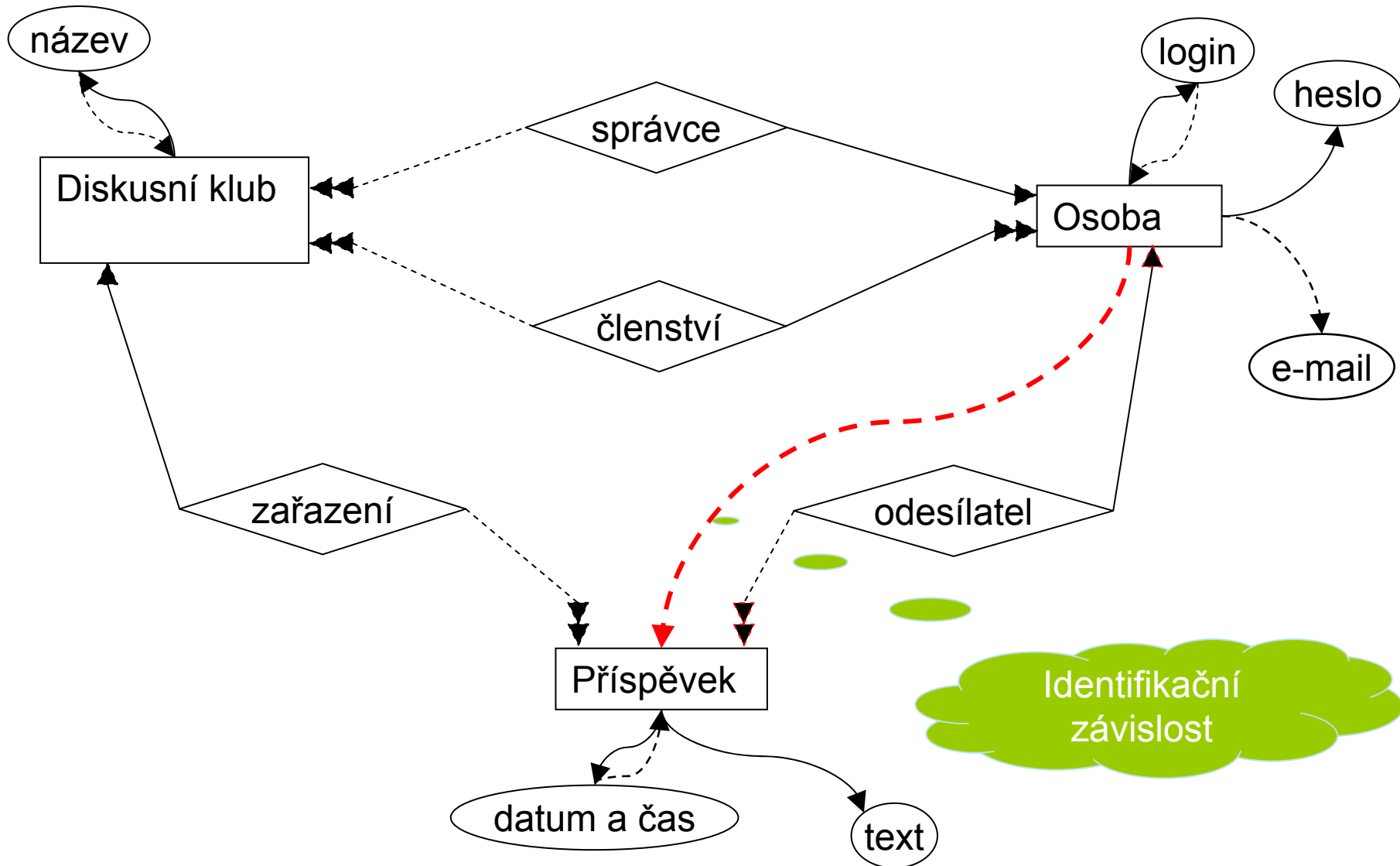
Další prvky ER(A)



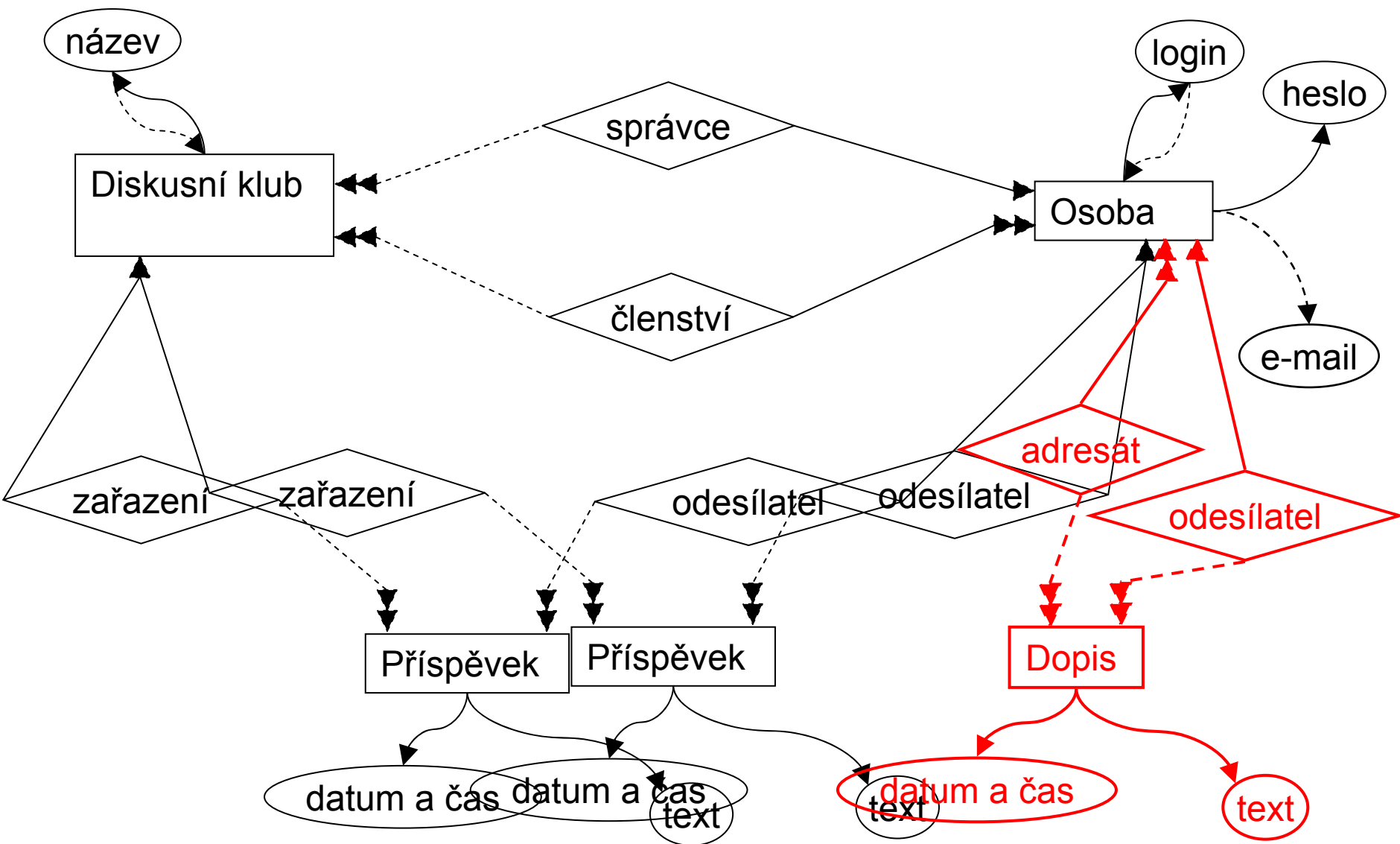
Další prvky ER(A)



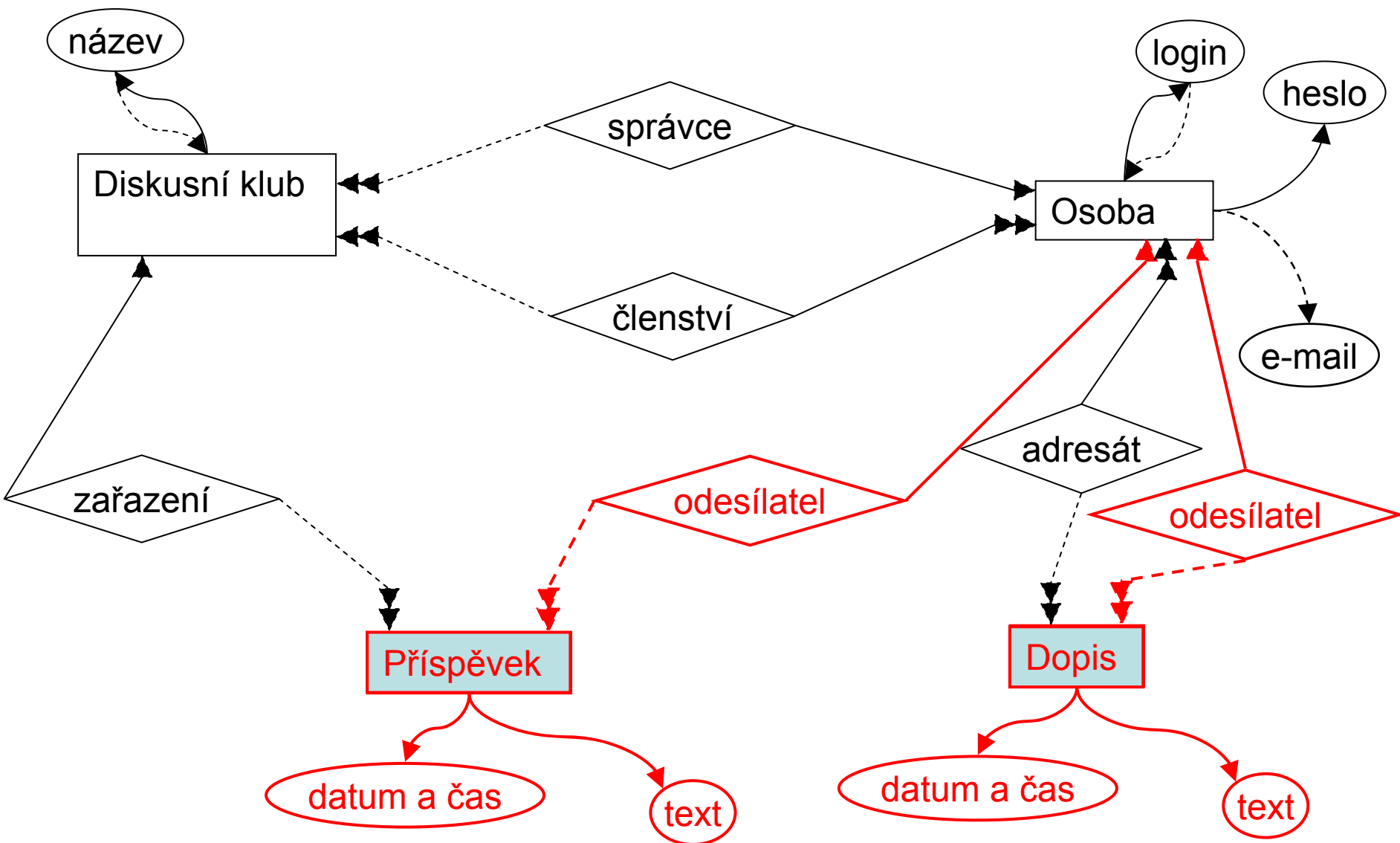
Další prvky ER(A)



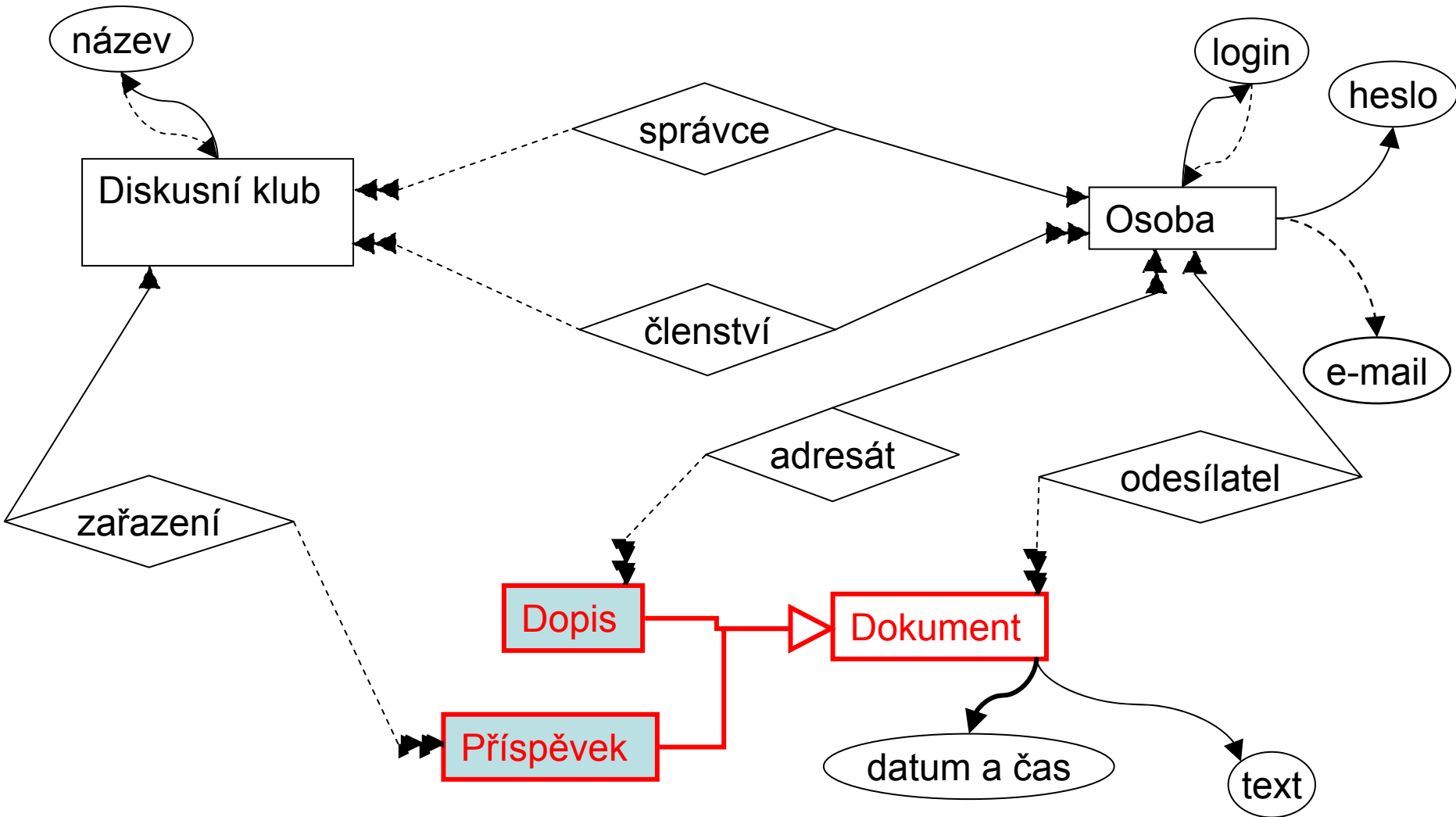
Další prvky ER(A)



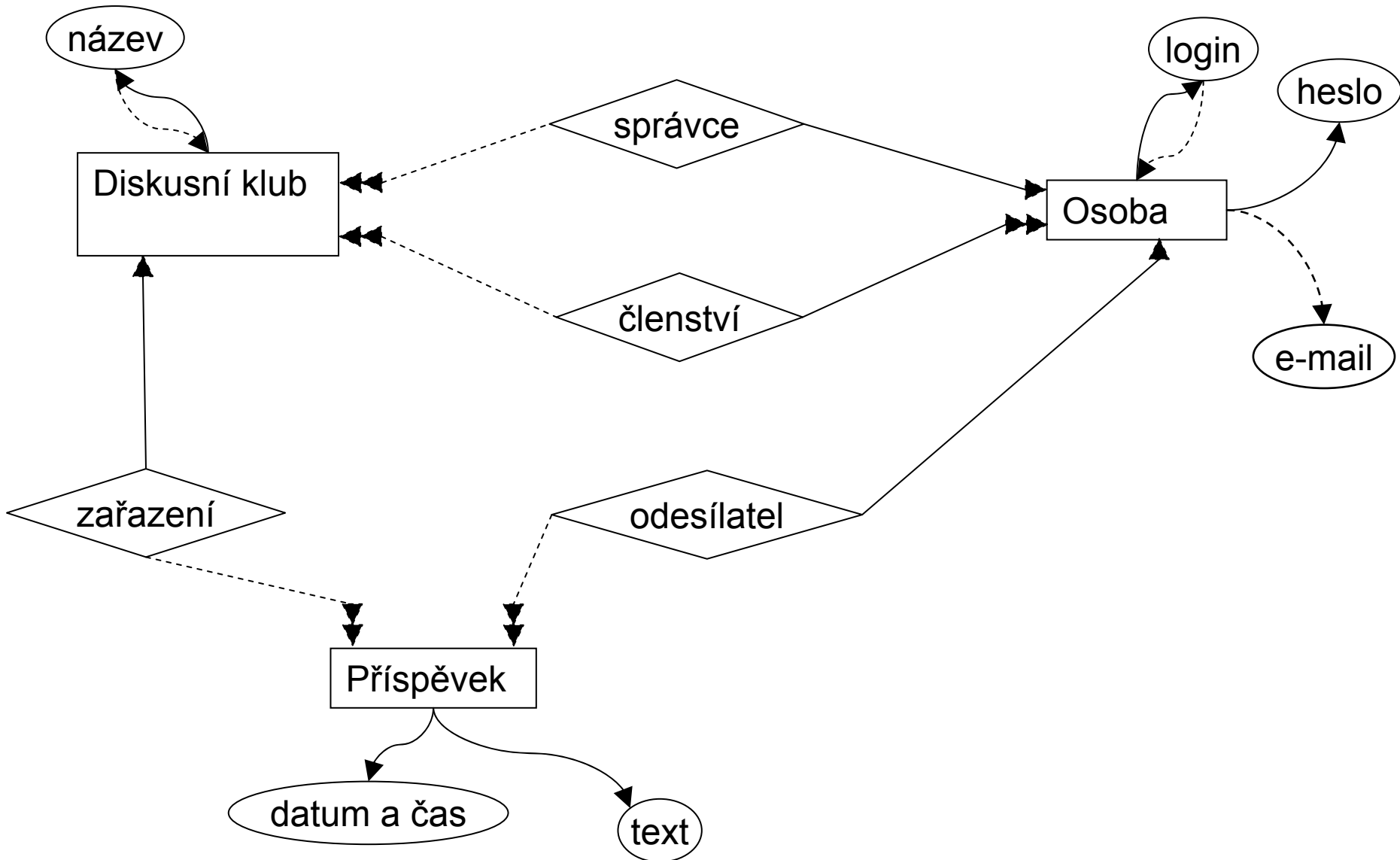
Další prvky ER(A)



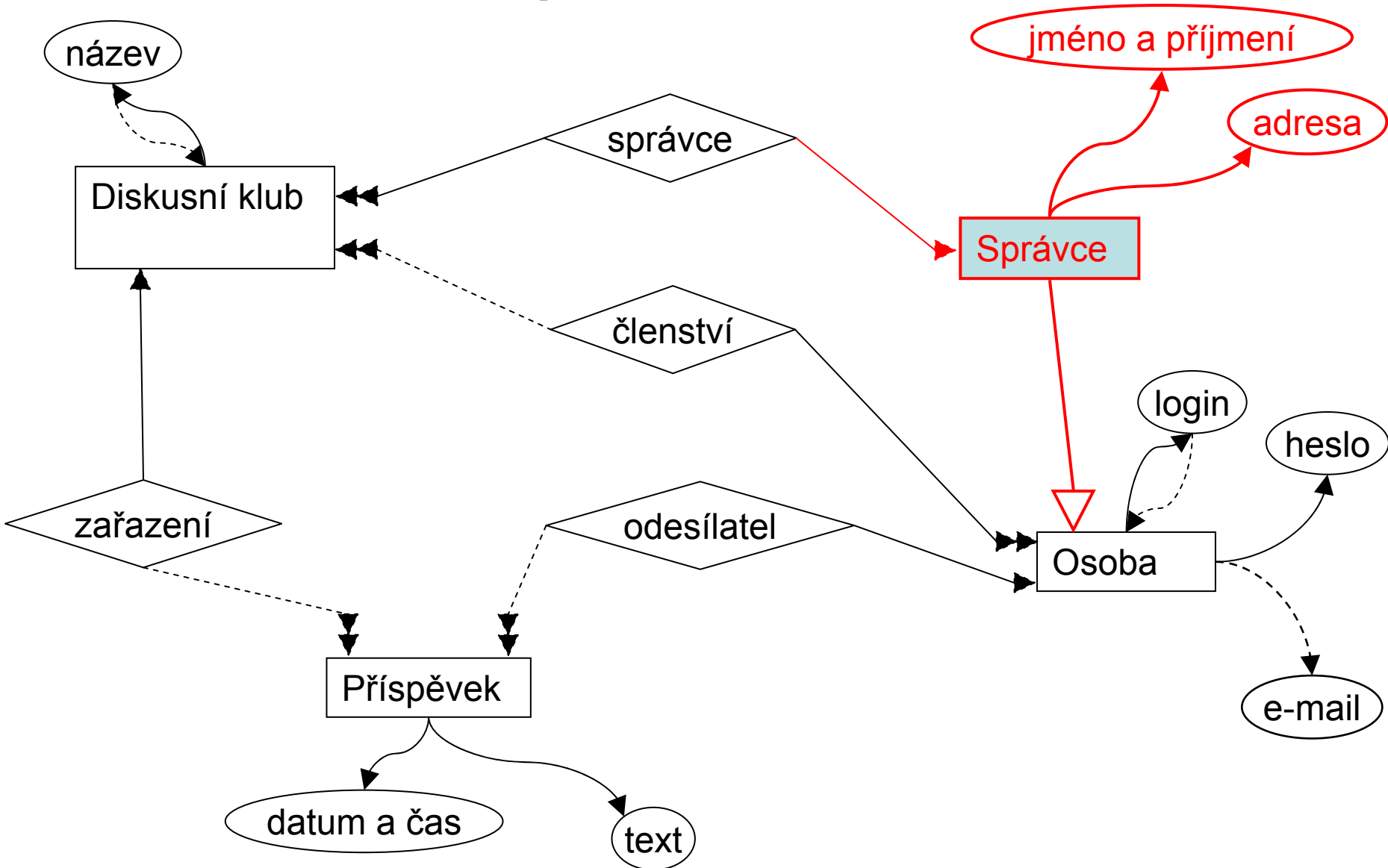
Generalizace



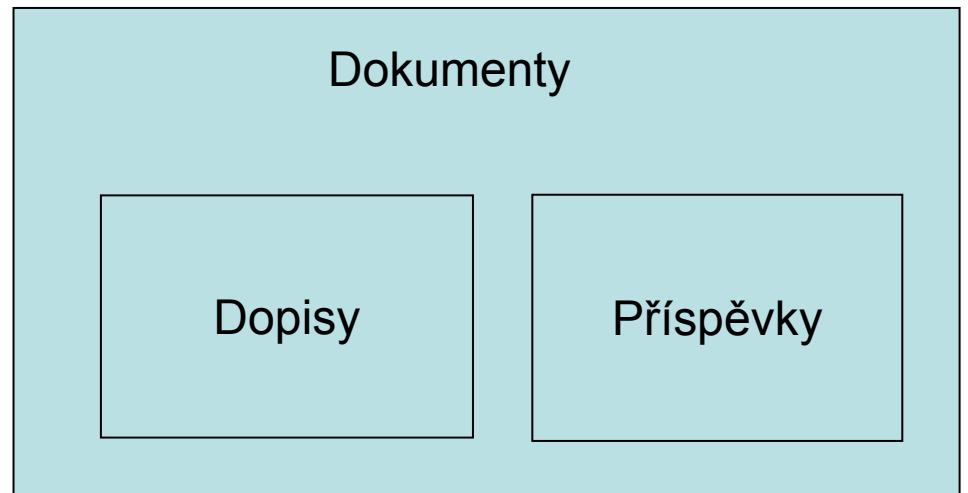
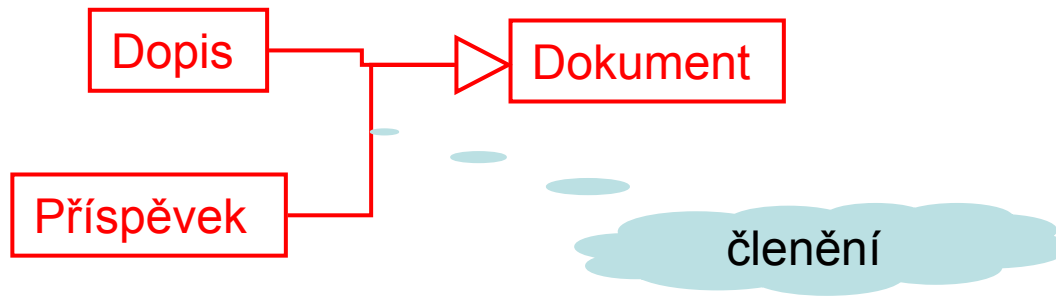
Specializace



Specializace

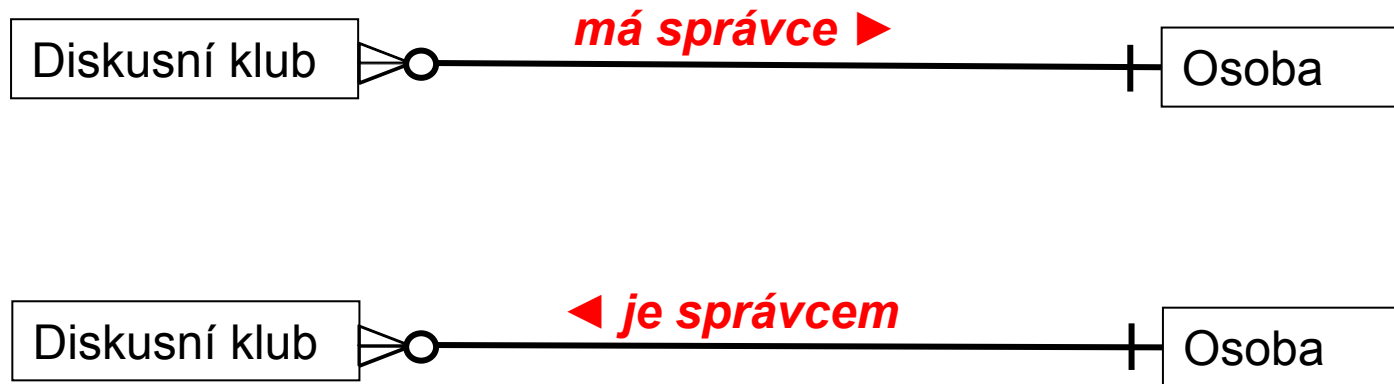


Generalizace/specializace



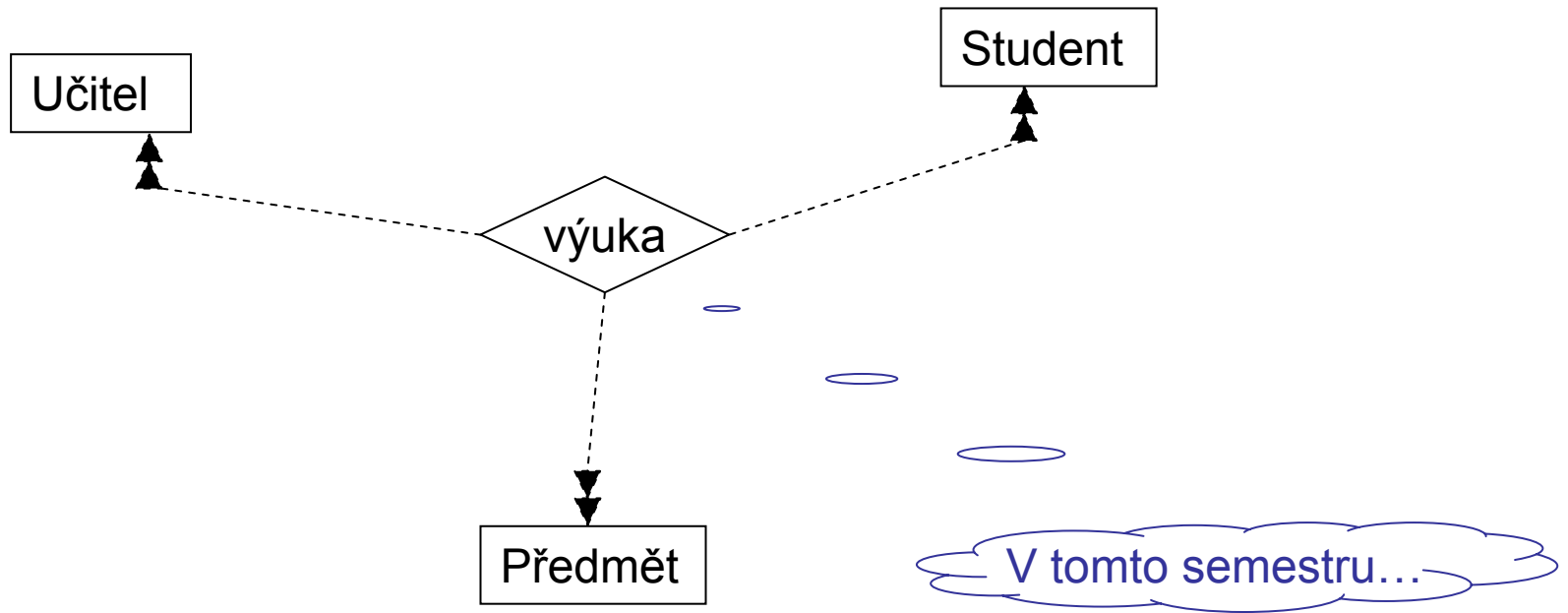
Vztahy

- Binární

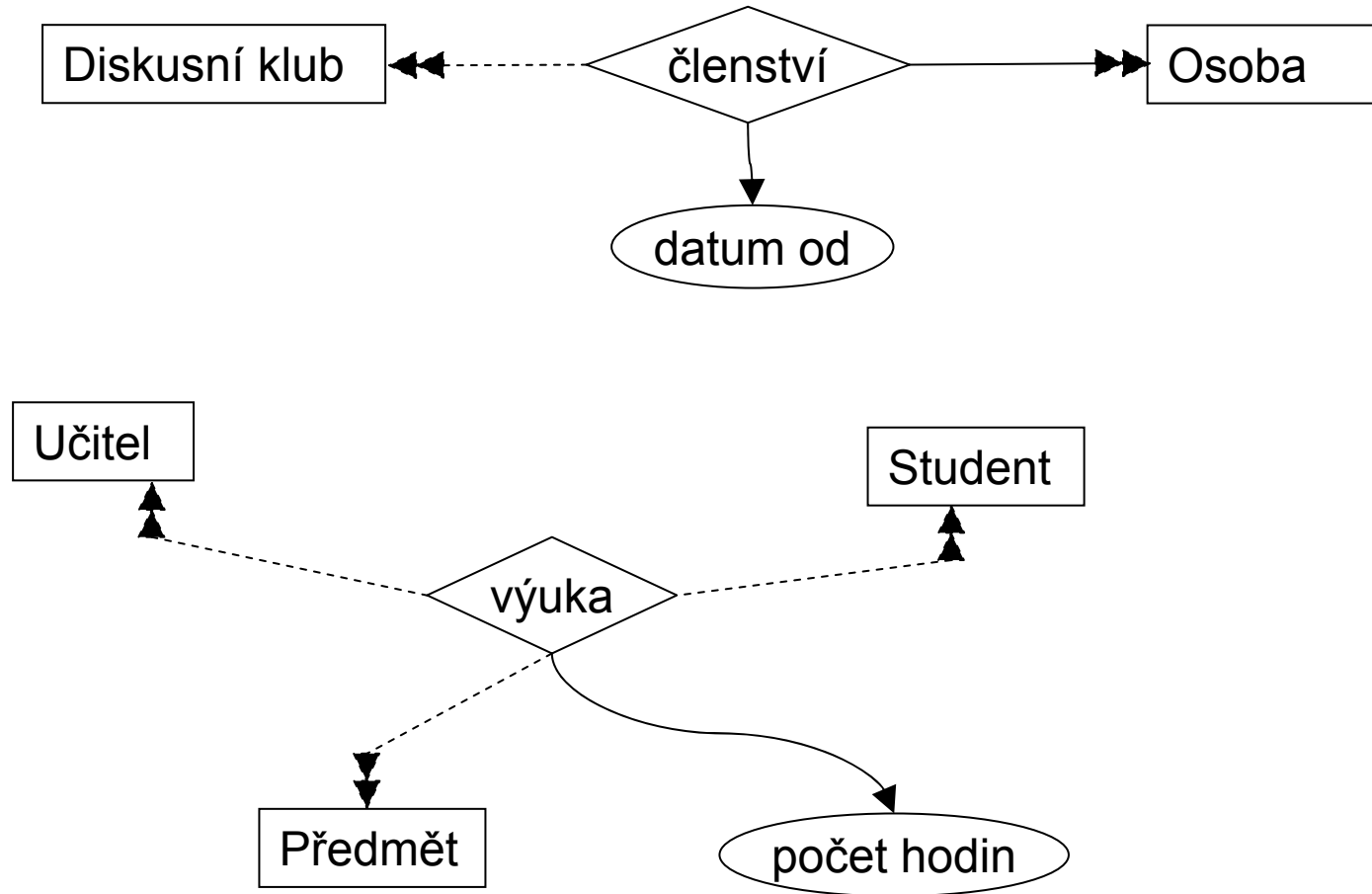


- více-ární...

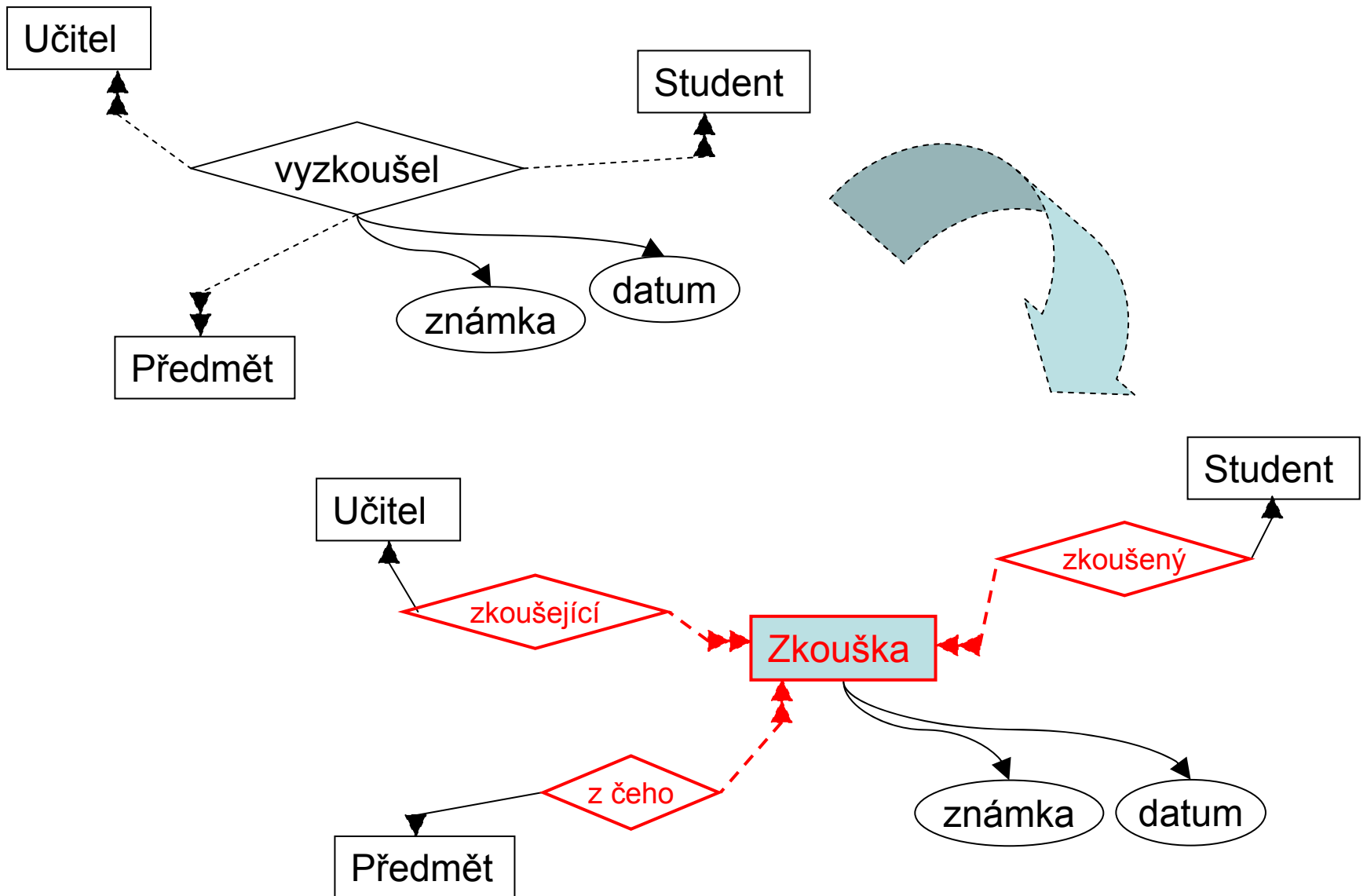
Více-ární vztahy



Atributy vztahů



Vazební entity

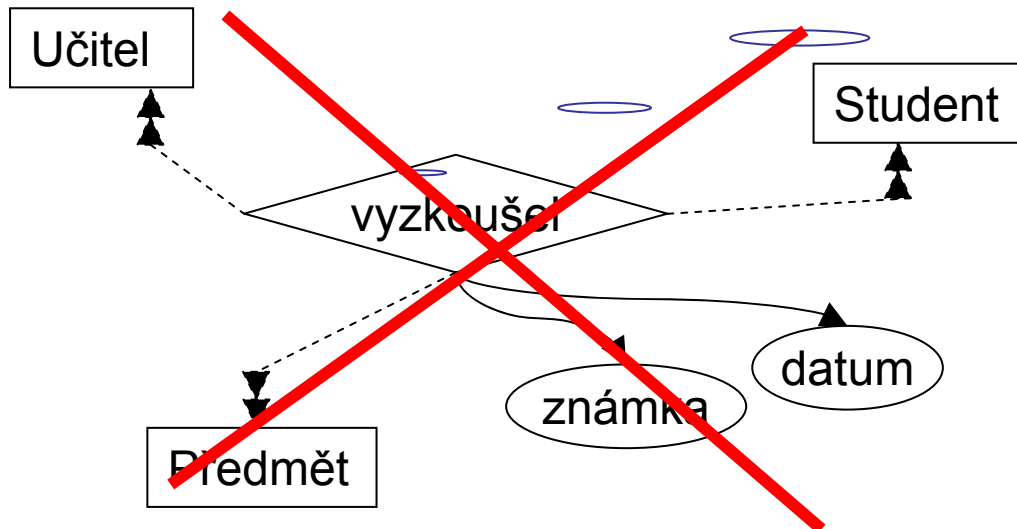


Vazební entity

- Binarizace více-árního vztahu
- Atributy vztahu
- Opakovaný vztah:

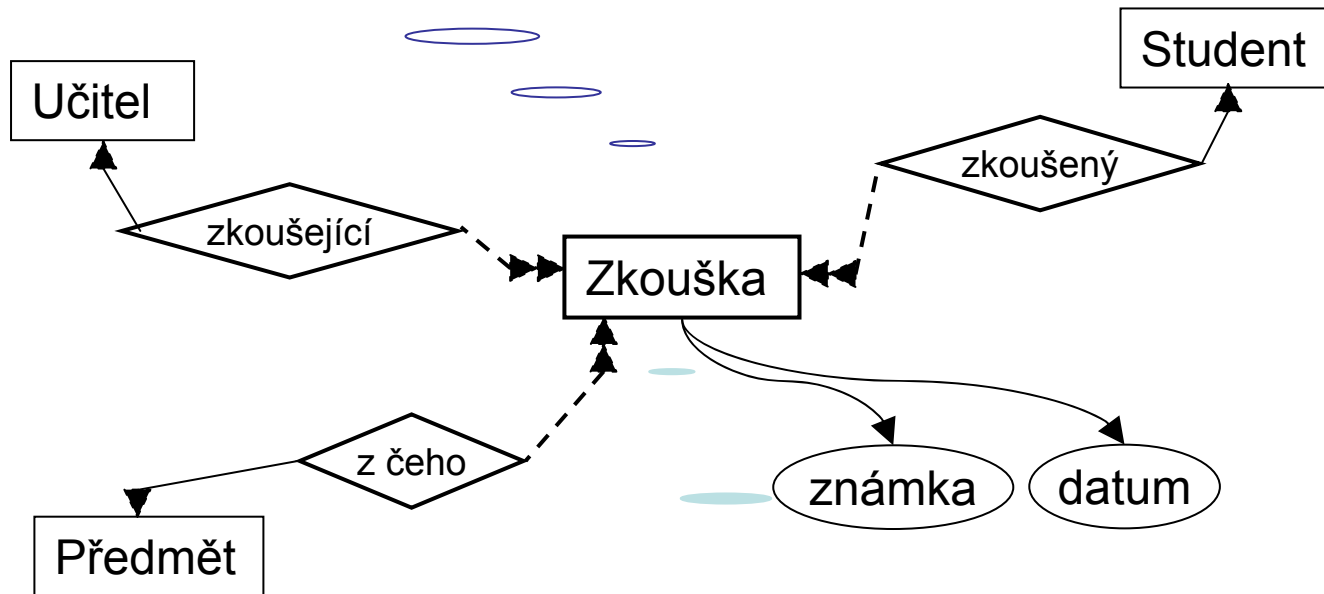
K tomu
můžeme být
přinuceni
použitou
technologií...

v celém průběhu studia...



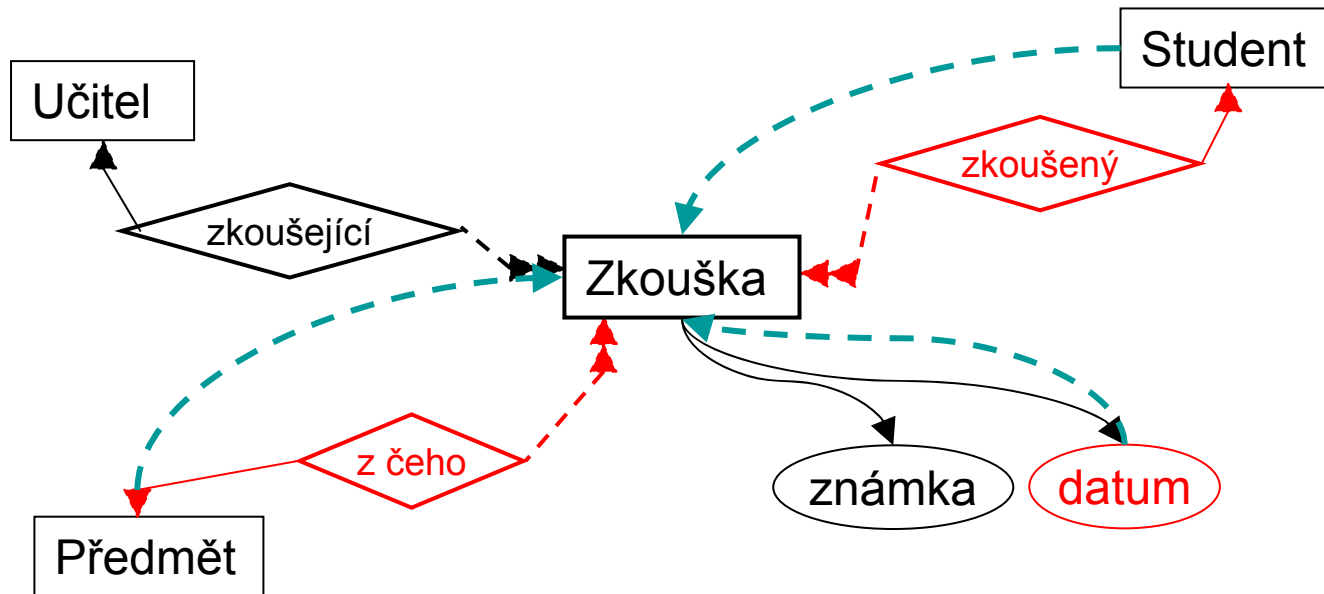
Opakovaný vztah

v celém průběhu studia !



Co je klíčem zkoušky?

Opakovaný vztah



Distribuované uložení dat

Distribuované databáze

- Zvnějšku se chovají stejně jako jiná databáze (centralizovaná)
 - něco trvá trochu déle
- Co musí řešit:
 - dlouhé (distribuované) transakce
 - replikace dat
 - distribuce správy dat
 - distribuce ochrany dat, včetně autorizace

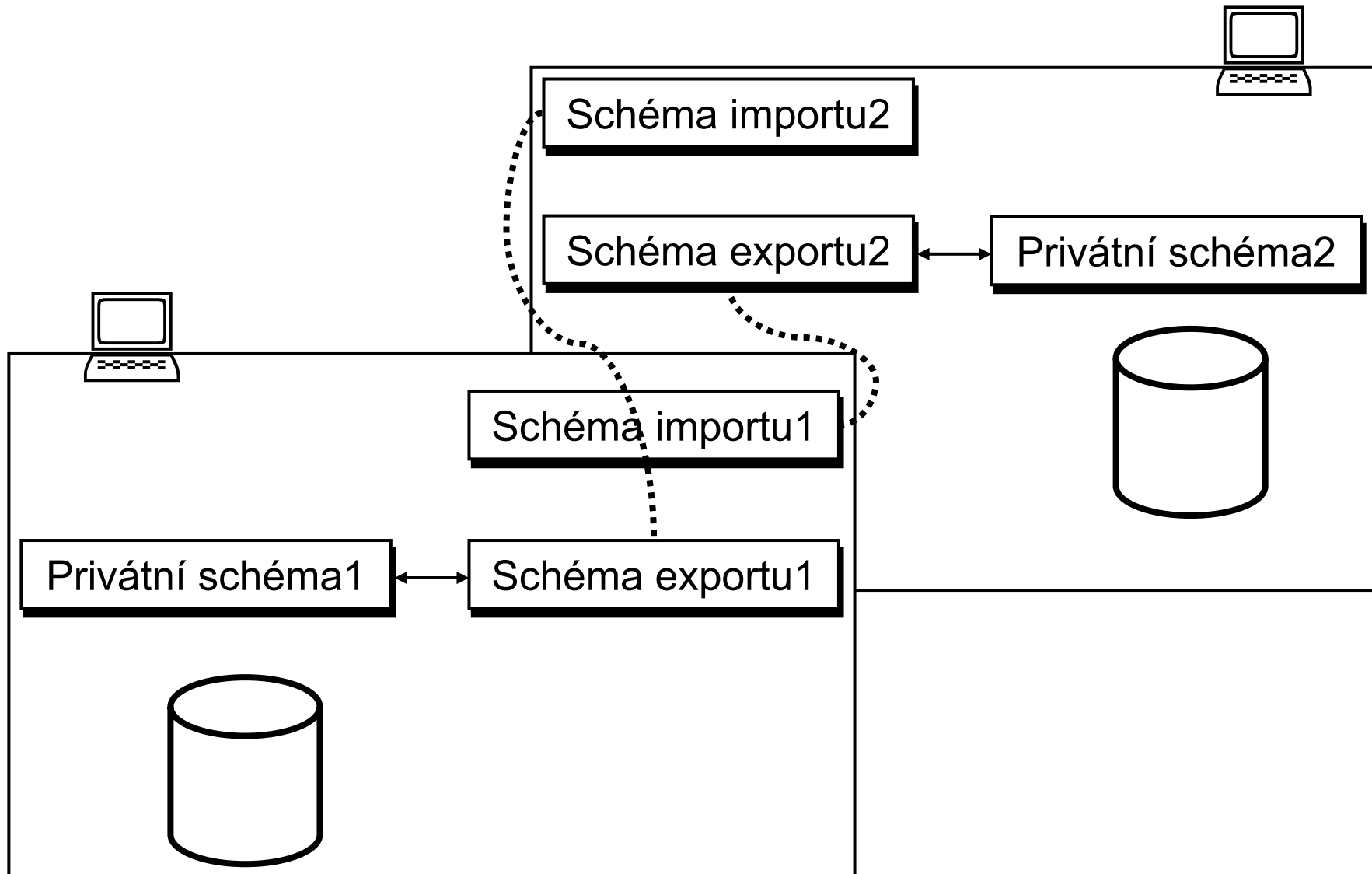
Federace databází

řízení integrity a transakcí

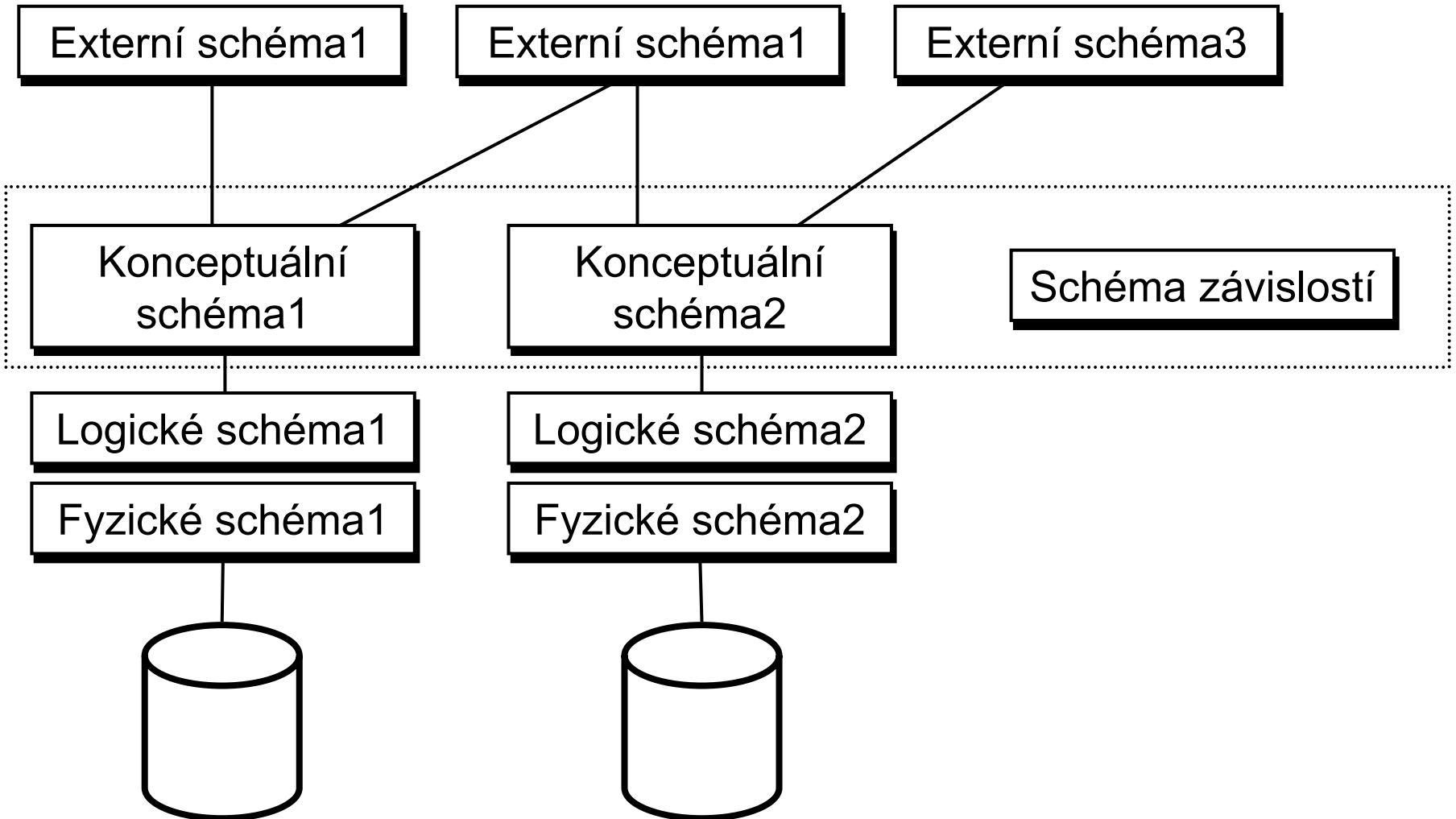
- **volně vázané**
 - schémata importu a exportu
 - multidatabáze (globální datový model = kompozice)
- **úzce vázané**
 - architektura 5ti schémat (integrované glob. schéma)

import/export	multidatabáze	5 schémat	
uživatel a lokální správce dat	uživatel	globální správce dat	zodpovědnost
z lokálního systému	globální rozhraní	globální systém	přístup
žádné	částečné	úplné	globální řízení

Schémata importu a exportu:



Multidatabáze:



Globální schéma:

*globální dotaz je formulován nad globálním konceptuálním
(event. externím) schématem...*

