

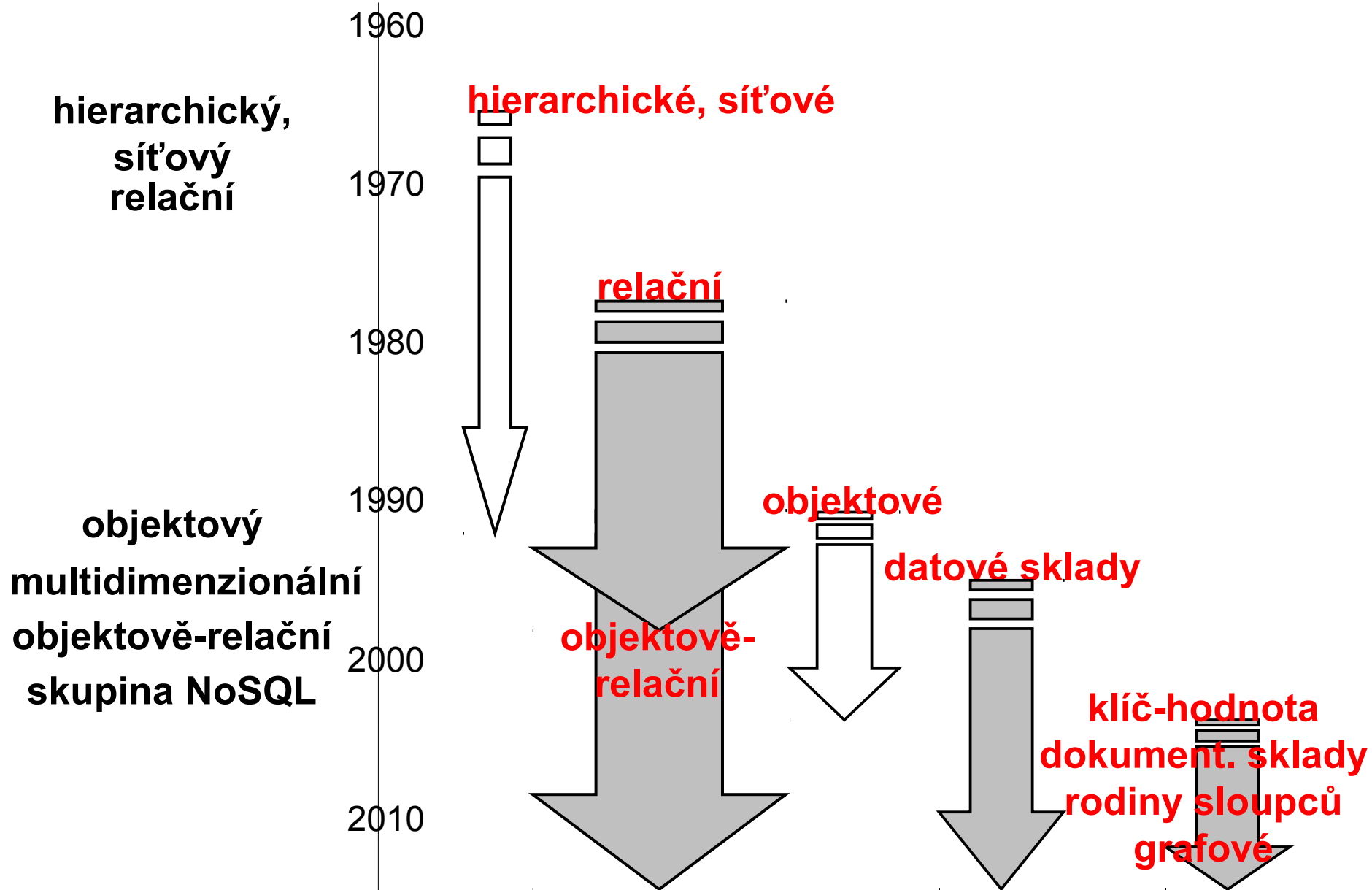
Nerelační databázové modely

Helena Palovská
palovska@vse.cz

Různé modely pro databázovou strukturu

databázové modely

SŘBD



Hierarchická organizace

- Vazby nadřízený záznam – podřízený záznam
- Žádný záznam nemá dva nadřízené
- Databáze je lesem stromů.

Příklady hierarchické organizace

- Fyzická pošta
- Územní správa
- Internetové domény
- Firemní organizace
- File system
- Textové dokumenty
- XML

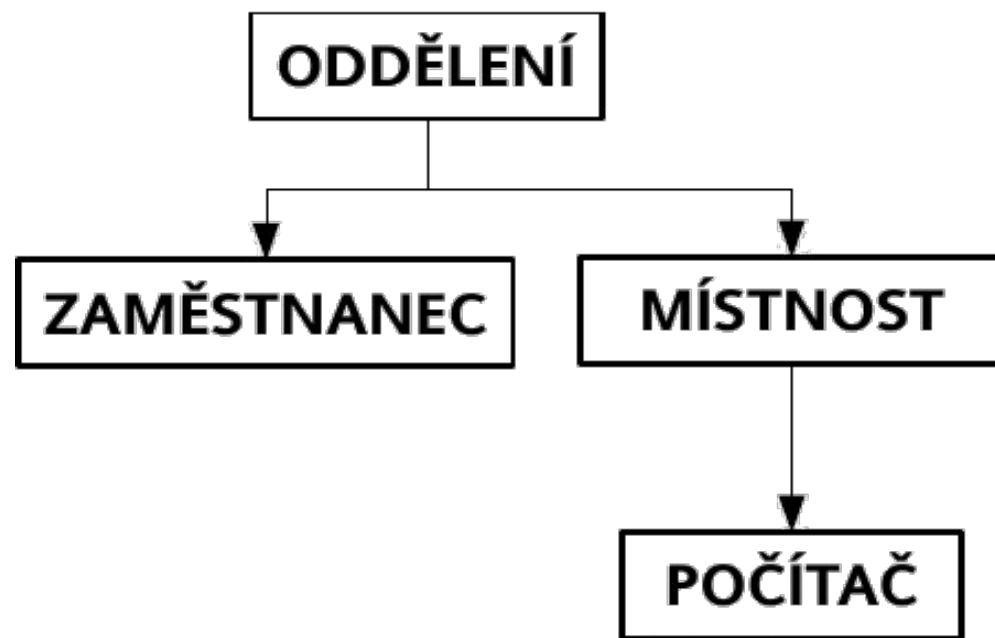
Příklad – objednávky

```
<objednávka číslo="2419">
  <zákazník>chralouh</zákazník>
  <datumOb>9.12.2009</datumOb>
  <doprava>PPL</doprava>
  <platba>dobírka</platba>
  <položky>
    <položka>
      <zboží>pxbd-21</zboží>
      <množství>3</množství>
      <cena měna="CZK">360</cena>
    </položka>
    <položka>
      <zboží>aha-57</zboží>
      <množství>2</množství>
      <cena měna="CZK">1480</cena>
    </položka>
  </položky>
</objednávka>
```

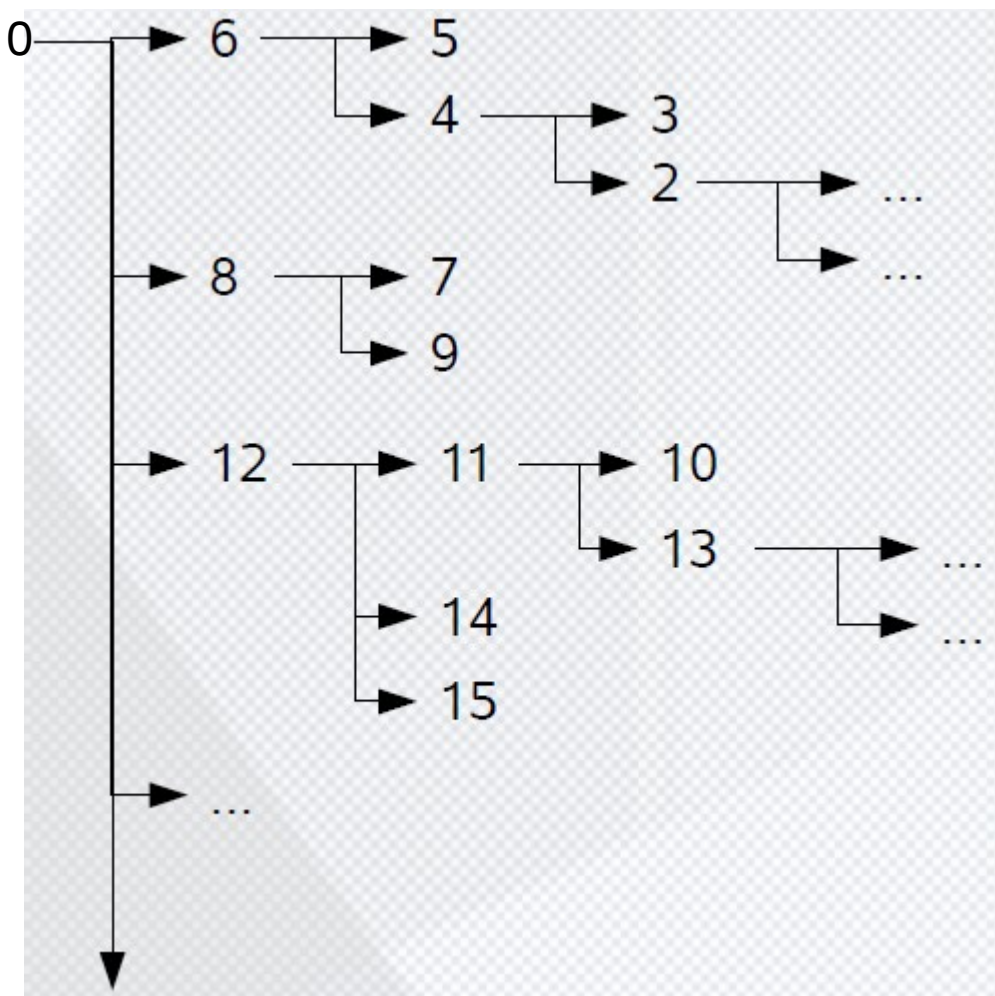


Příklad – oddělení ve firmě

```
<oddělení cisodd=2>  
  <šéf>Jan Novák</šéf>  
  <zaměstnanci>  
    <zaměstnanec>  
      <jméno>Petr Novotný</jméno>  
      <od>14.5.1999</od>  
    </zaměstnanec>  
    <zaměstnanec>  
      <jméno>Alena Černá</jméno>  
      <od>3.6.2004</od>  
    </zaměstnanec>  
  </zaměstnanci>  
  <místnosti>  
    <místnost číslo="45">  
      <počítače>  
        <počítač idpoč="24893">  
          <monitor idmonit="5563">  
            ...  
          </monitor>  
        ...  
      </počítače>  
    </místnost číslo="45">  
    ...  
  </místnosti>  
</oddělení>
```

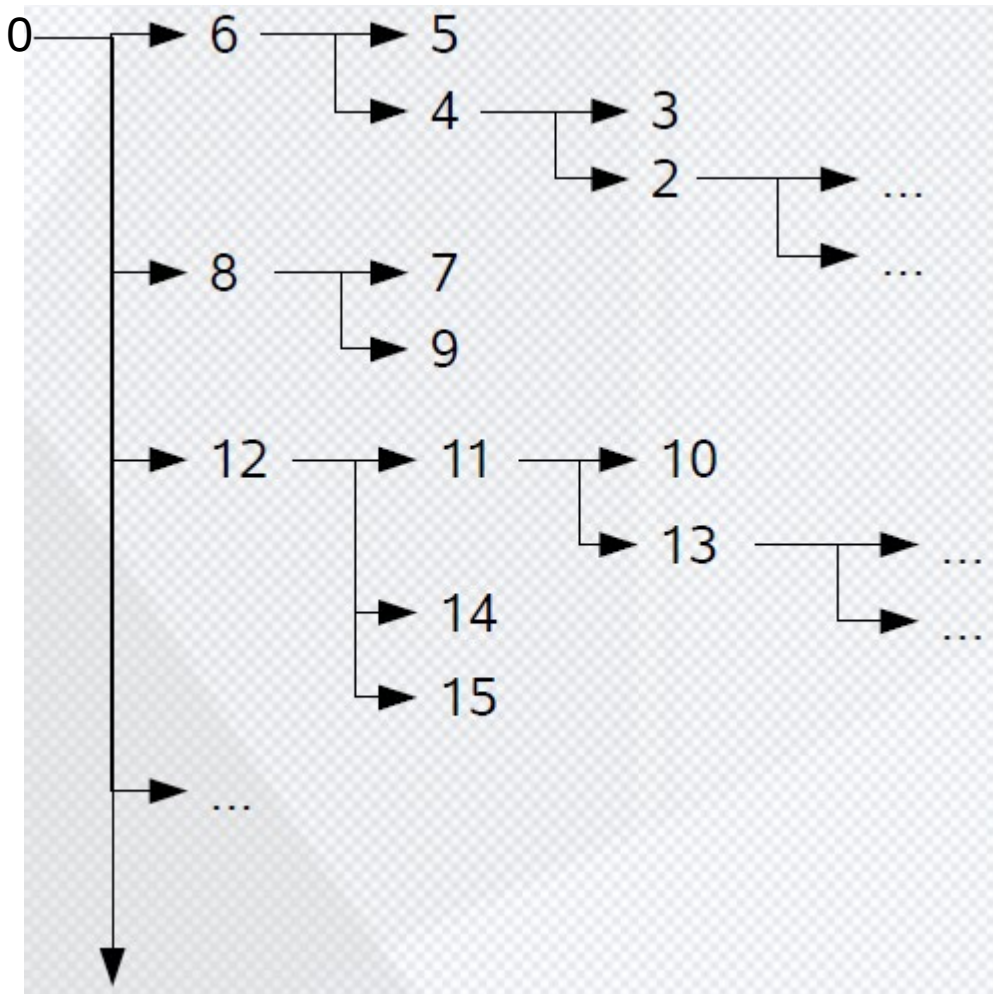


Hierarchická identifikace & serializace



0 | 0.6 | 0.6.5 | 0.6.4 | 0.6.4.3 | 0.6.4.2 | 0.6.4.2... |
0.8 | 0.8.7 | 0.8.9 | 0.12 | 0.12.11 | 0.12.11.10 |
0.12.11.13 | 0.12.11.13.... | 0.12.14 | 0.12.15 | 0....

Porovnání s relačním řešením hierarchie



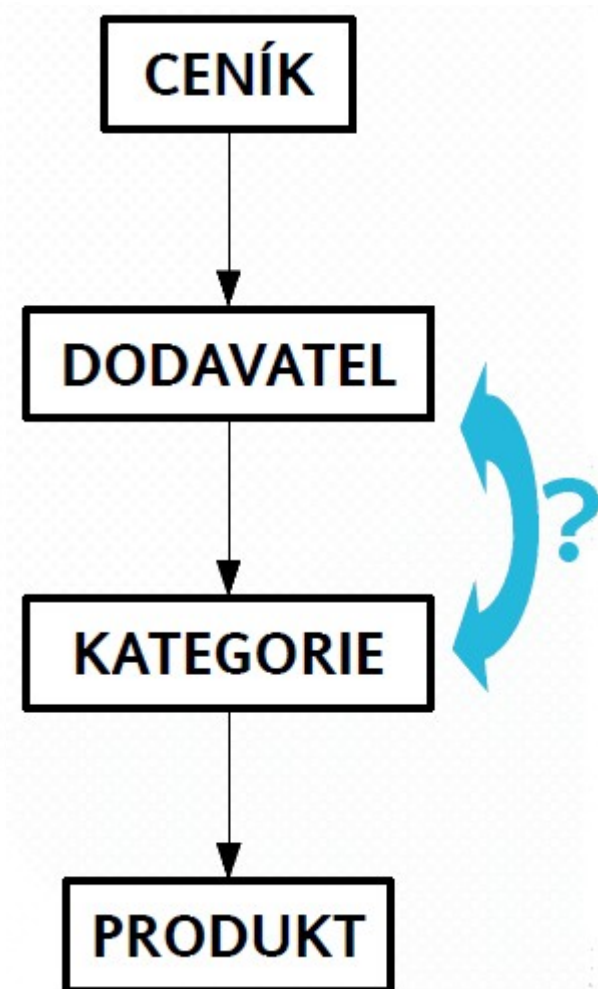
ID	data	nadr
0	...	-
6	...	0
8	...	0
12	...	0
5	...	6
4	...	6
7	...	8
9	...	8
3	...	4
2	...	4
11	...	12
14	...	12
15	...	12
10	...	11
13	...	11
...	...	...

Vlastnosti hierarchického modelu

- Přednosti:
 - Lze efektivně serializovat (co patří k sobě je u sebe)
- Přístupové cesty:
 - Nadřazený prvek zná cesty k jemu podřízeným, ne vždy lze efektivně obráceně
- Zpracování:
 - Dle přístupových cest a pravidel v daném typu uzlu
- Omezení modelu:
 - Vztahy N:M způsobují redundanci (někdy lze řešit pomocí konstrukce odkazů)
 - Upřednostnění jednoho pohledu na data (příklad ceník →)

Příklad - ceník

```
<ceník>
  <název>Počítačové komponenty</název>
  <platnost>
    <od>1.1.2009</od>
    <do>31.3.2009</do>
  </platnost>
  <dodavatel>
    <název>První hardwarová, s.r.o.</název>
    <adresa>
      <ulice>Průmyslová</ulice>
      <město>Praha 10</město>
      <psč>100 00</psč>
      <email>info@prhw.cz</email>
    </adresa>
    <nabídka>
      <kategorie>
        <název>polohovací zařízení</název>
        <produkt kód="pxbd-21">
          <název>Hyperoptická digitální myš</název>
          <cena měna="CZK">850</cena>
        </produkt>
      </kategorie>
    </nabídka>
  </dodavatel>
</ceník>
```



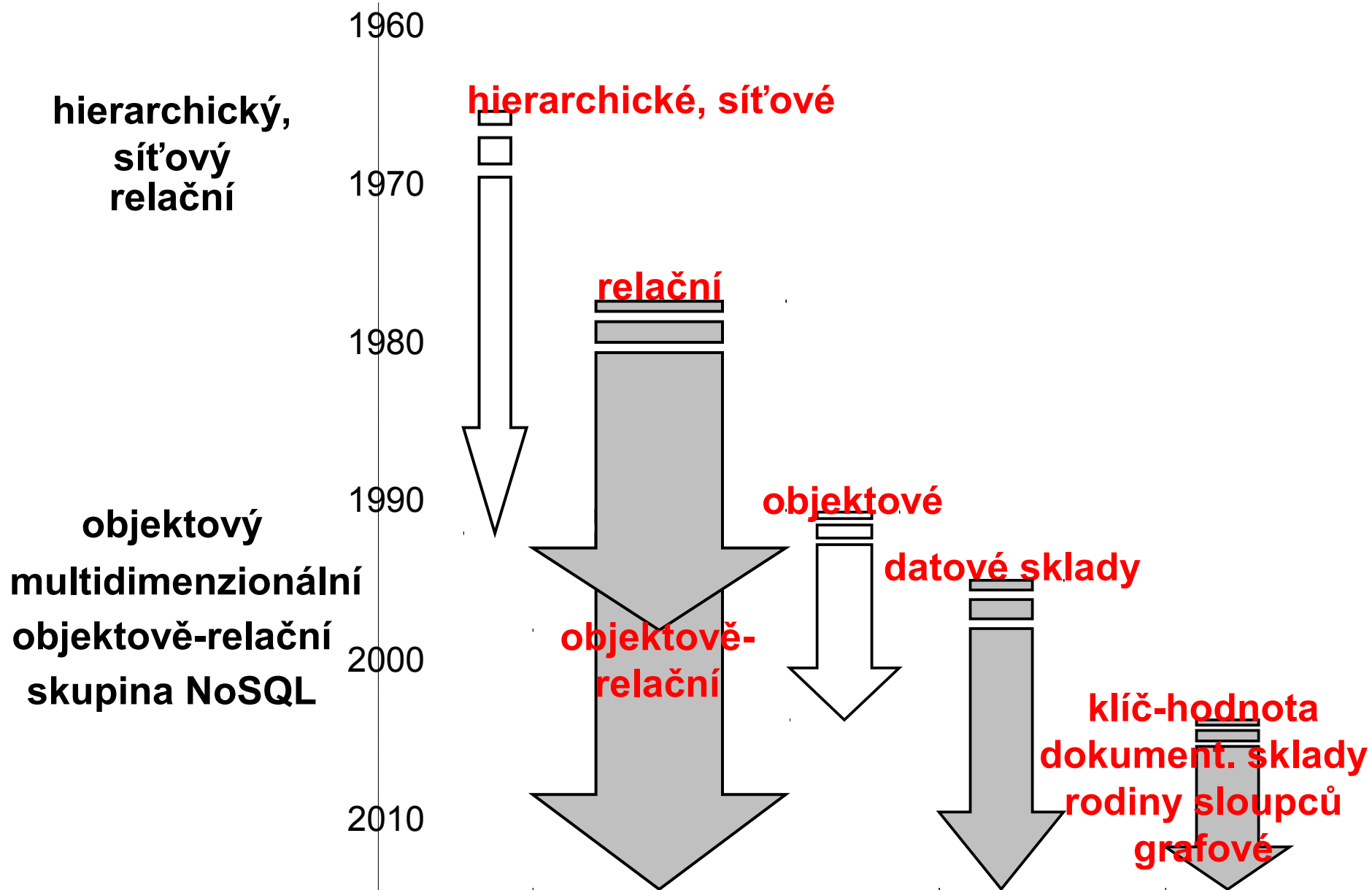
Hierarchické databáze

- Příklady
 - IBM IMS
 - Caché

Různé modely pro databázovou strukturu

databázové modely

SŘBD

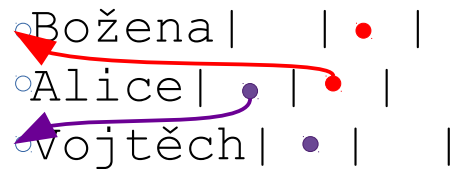


Sít'ové databáze

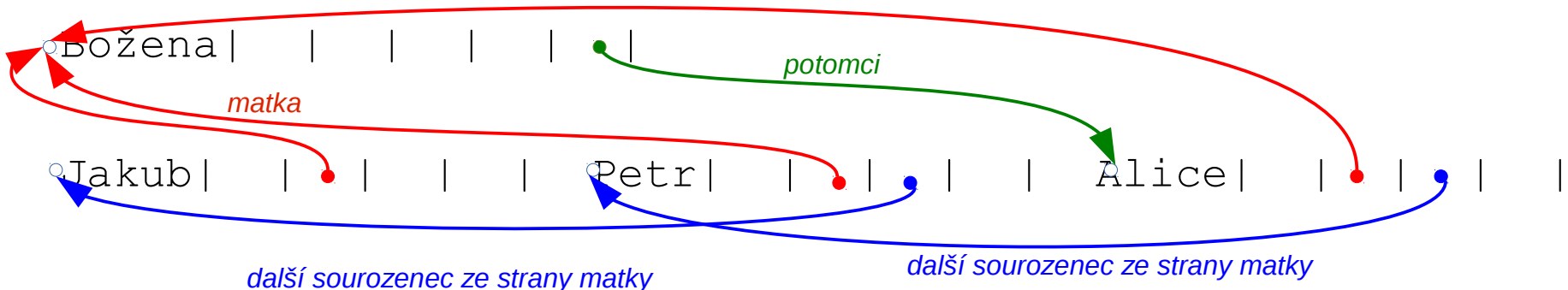
- Záznam může mít více předchůdců,
- ale v daném typu vztahu jen jednoho.
 - Například otce a matku.

Fyzická realizace

- Každý vztah realizován směrničky v rámci záznamů.

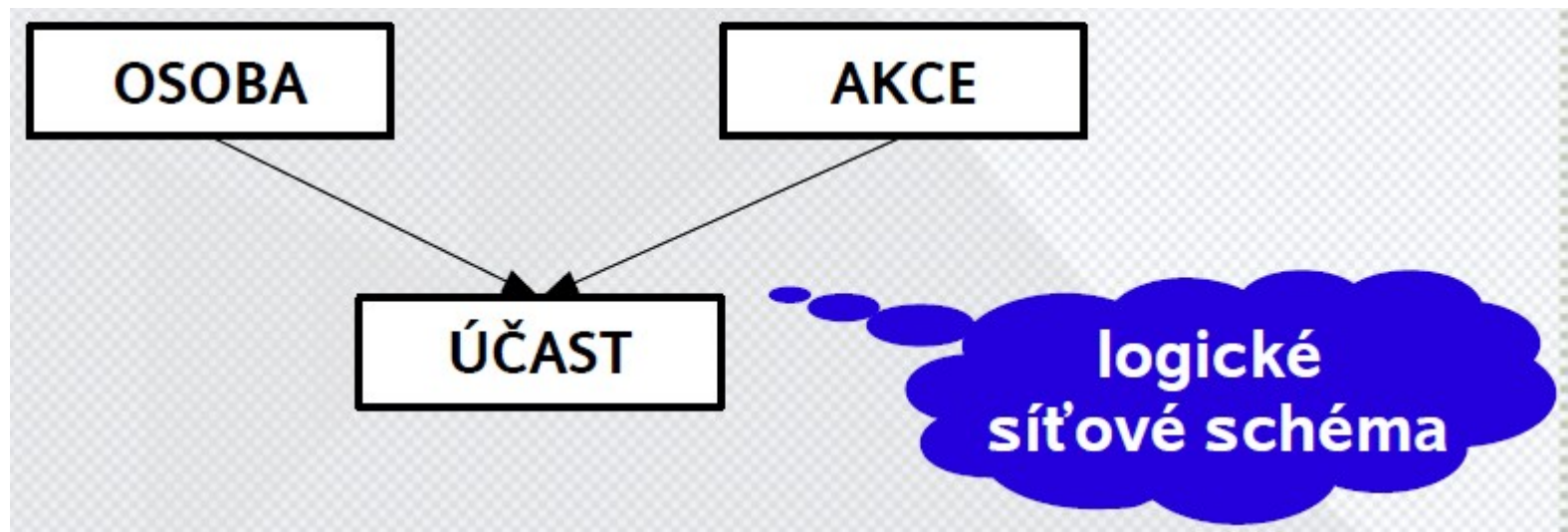


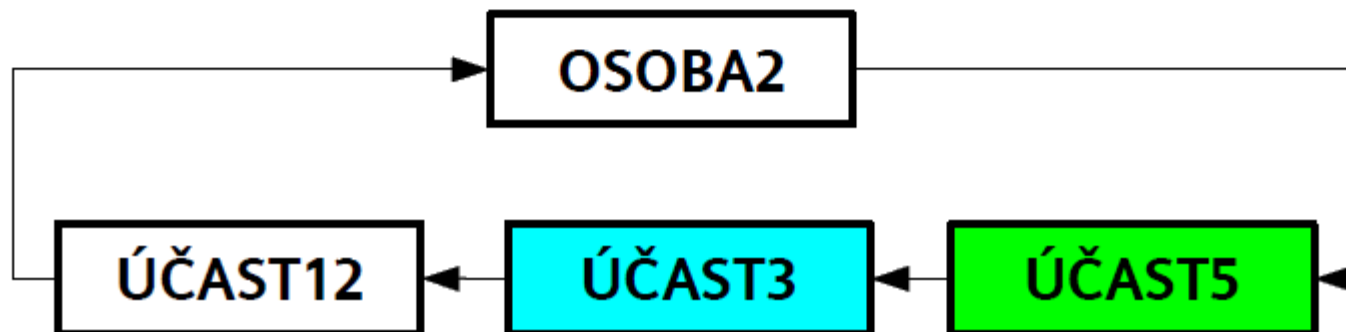
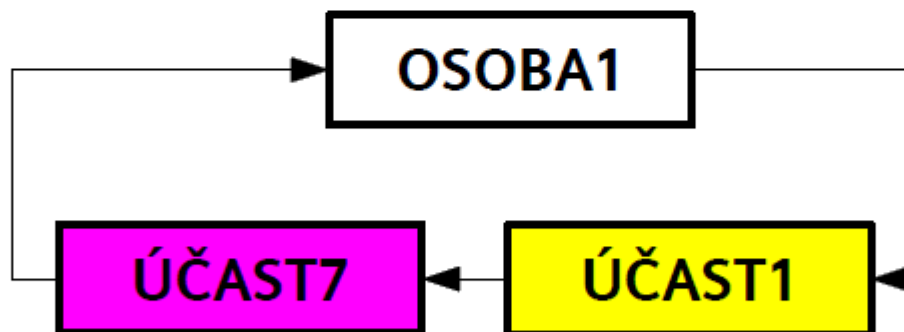
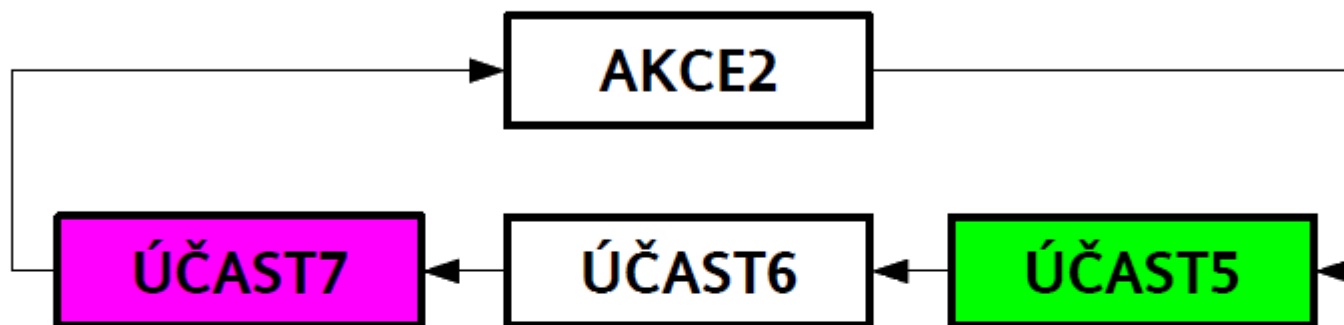
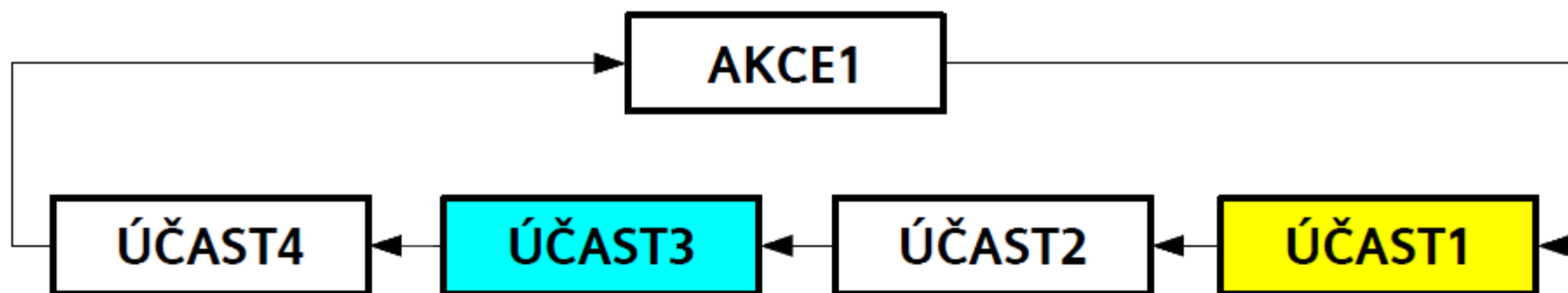
- Pro možnost procházení všech potomků v daném vztahu jsou potomci propojeni do spojového seznamu, začínajícího jejich předchůdcem v tomto vztahu.



Síťové databáze

- Pro vztah N:M tvořeny „průnikové“ typy záznamů
 - například pro vztah OSOBA se *účastní* AKCE je potřeba typ záznamu ÚČAST





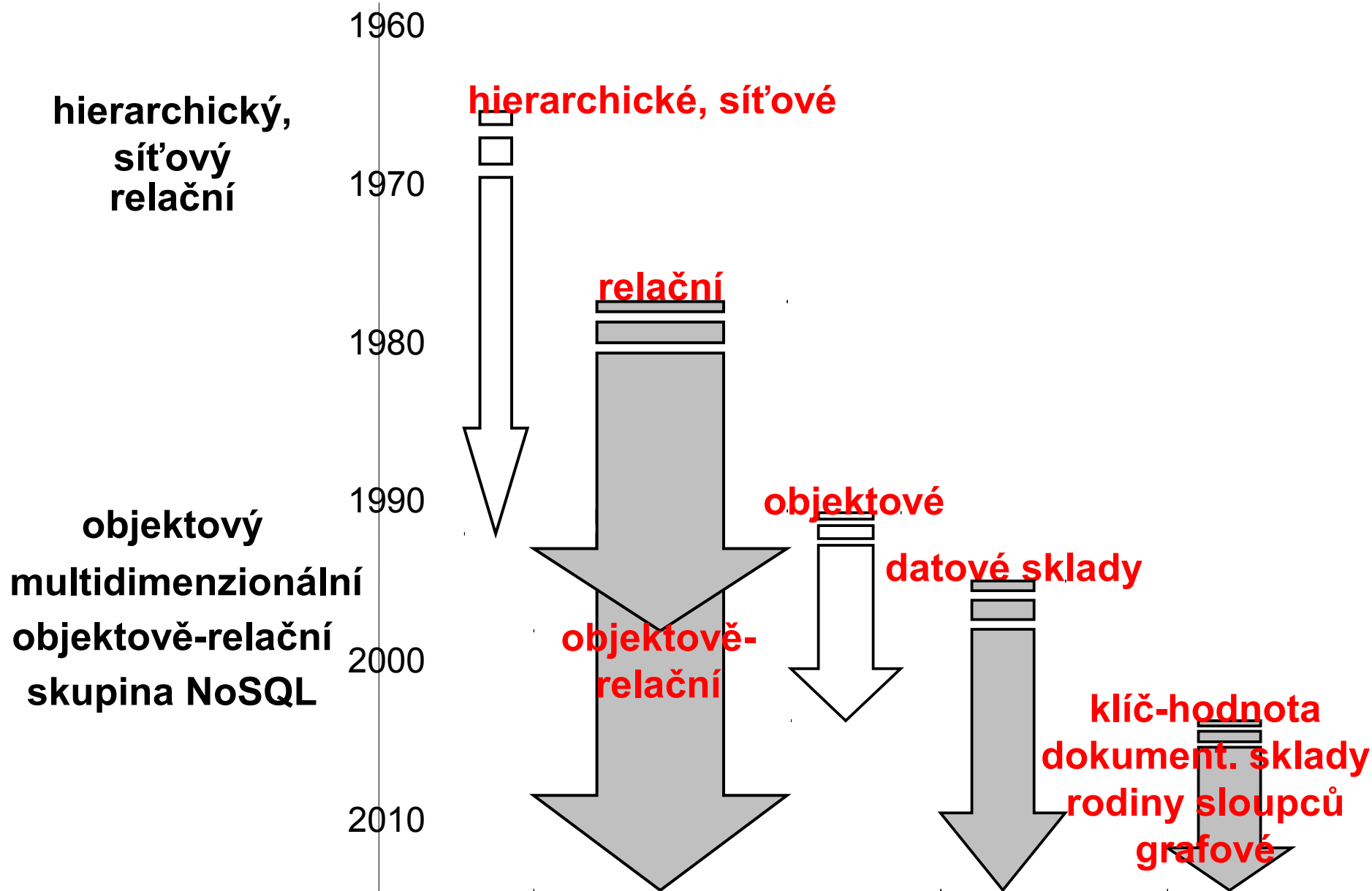
Vlastnosti síťového modelu

- Přednost
 - Rychlost provádění navigačních dotazů
- Problémy:
 - Obtížnost udržení integrity pro vztahy N:M
 - Obtížně realizovaná dědičnost
 - Obtížně realizovatelné vztahy 1:1
 - Nedostupnost nepropojených záznamů
- Dotazy a manipulace
 - pouze procedurální
podle předem definovaných vazeb

Různé modely pro databázovou strukturu

databázové modely

SŘBD



Objektový databázový model

- Objektový model podle OMG
 - objekty stejné třídy mají stejné chování a stejnou množinu stavů
 - interpretace odkazů na objekty jako vztahů
- Standardy
 - ODMG 3.0 (2001) (standard nedefinuje syntaxi jazyka, ale jen sémantiku)
 - JDO
 - 2006 ODBT WG
- Dotazovací jazyk OQL

Speciální konstrukce ODMG ODL

- SET
- BAG
- ARRAY
- STRUCT
- LIST

Příklad ODMG ODL

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

Také metody,
konstruktory, destruktory

Referenční vztahy,
integrita

1:1

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

pro 1:n, n:m, n:1

Dotazovací jazyk OQL - příklady

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

...výsledkem je Bag

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

...výsledkem je struktura

- Do SELECT možno i zahrnování struktur, i metody
- V ad-hoc dotazech se připouští porušení zapouzdření

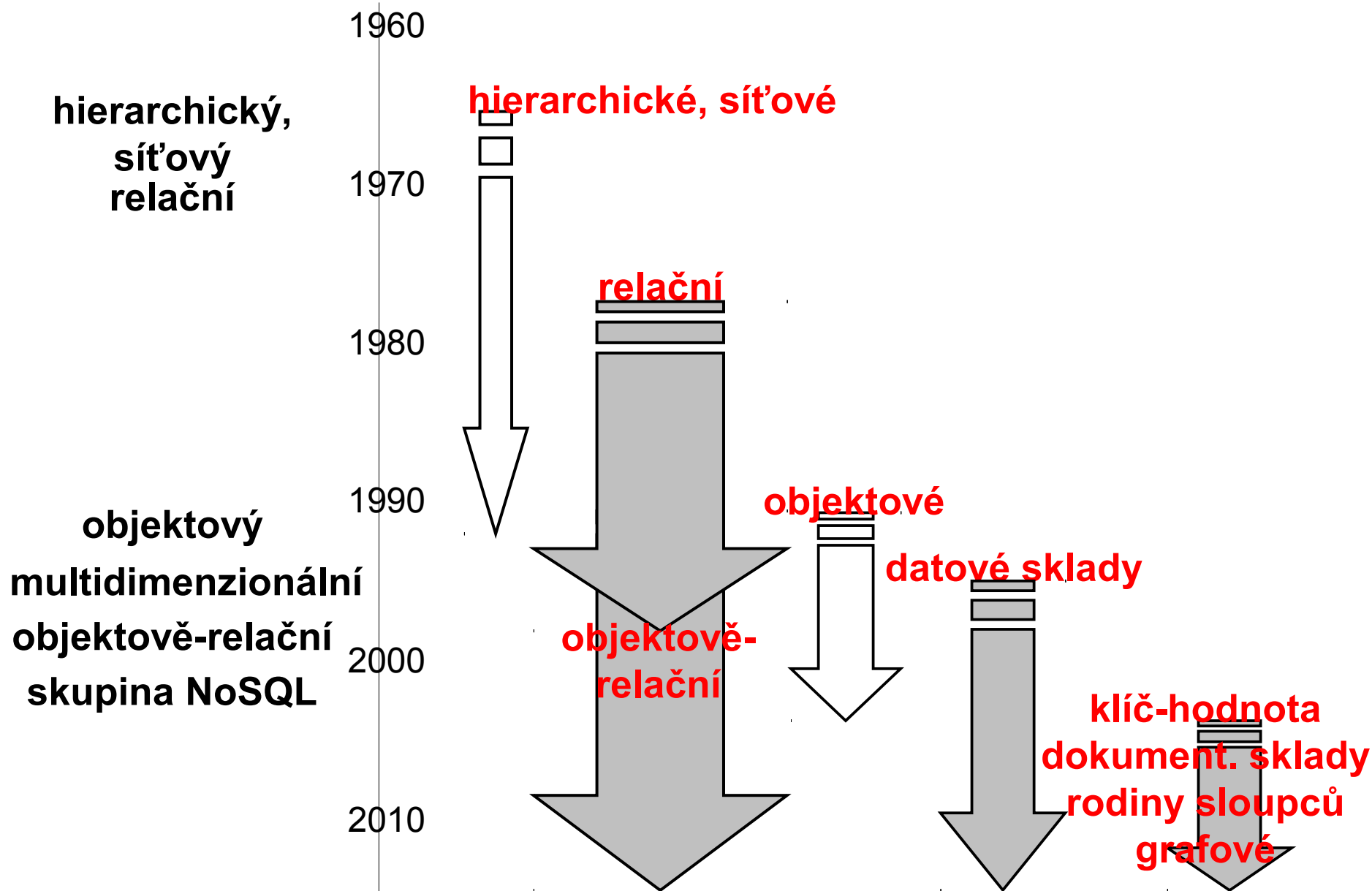
Vlastnosti objektových databází

- Jazyk manipulace dat není samostatně definován, jsou definována propojení s objektovými programovacími jazyky
- Standard ODMG není implementován, jednotlivé SŘBD nejsou interoperabilní
 - Gemstone, Caché, Objectivity/DB, ObjectStore, Versant...
- JDO
- Rozšiřování OO jazyků o perzistenci

Různé modely pro databázovou strukturu

databázové modely

SŘBD



Objektově-relační model

- Rozšíření relačního modelu, SQL 1999
- Obsahuje:
 - Objektovou identitu a reference
 - Funkce, uložené procedury, trigger
 - Složité datové struktury
 - Pole, hnížděné tabulky, ...
 - Strukturované hodnoty
 - Dědičnost
 - Rozšiřitelnost datových typů
 - DML s objektovými rysy

Složité datové struktury

id	contact		lineitems		
	name	phone	item	qty	unit_price

porušení 1NF

JMD i nadále
deklarativní

- Možno i jako řádkový typ při CREATE TABLE
- "lineitems" mohou být logicky dostupné v "nested table" (ORACLE)

Objektová identita v SQL 1999

- Záznamy mohou mít OID
- Mohou existovat pole typu REF
 - nelze s nimi manipulovat jako s hodnotou, ale jen jako s odkazem

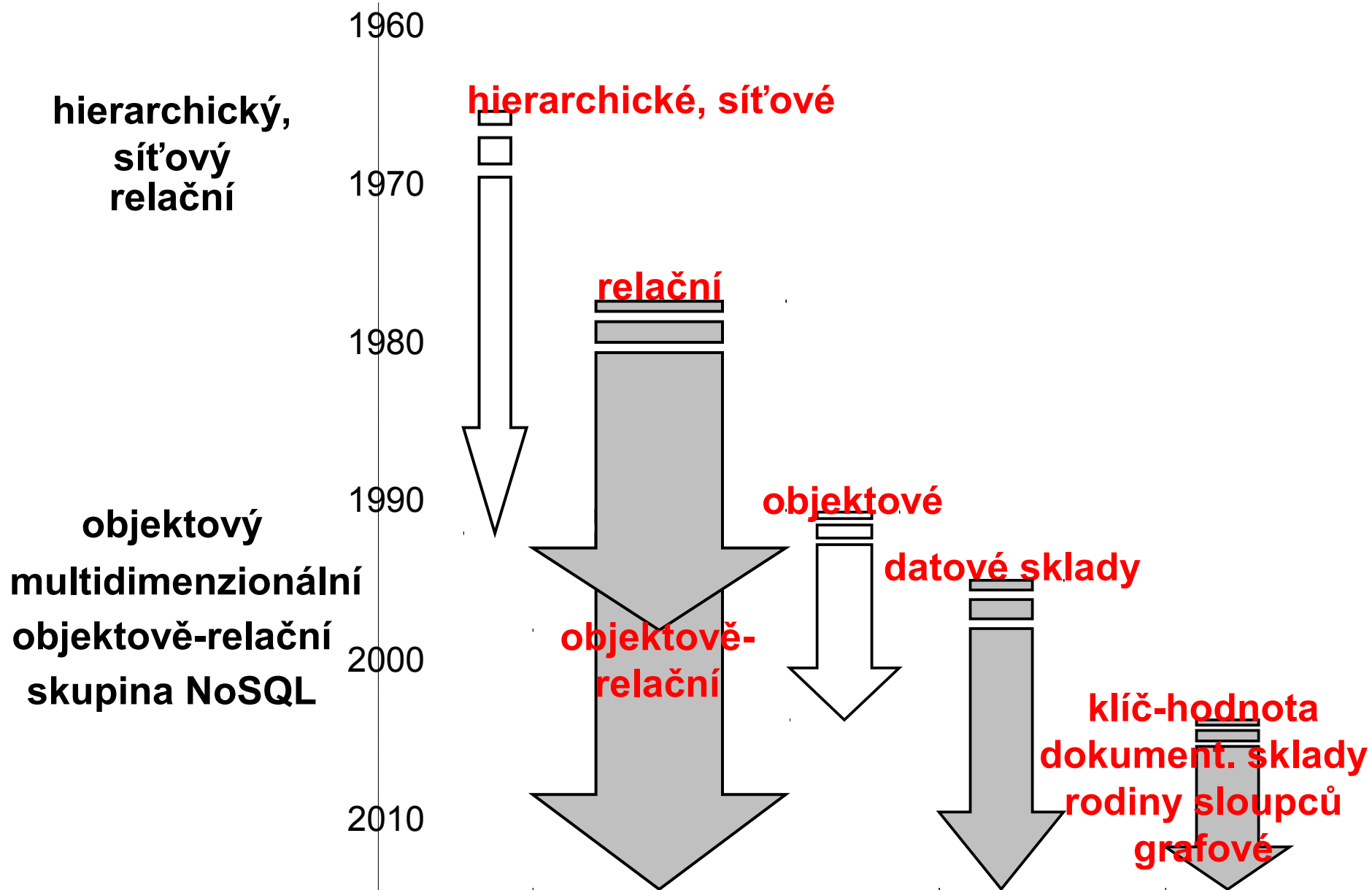
```
SELECT U.Vlastnik.Jmeno  
FROM Ucty U  
WHERE U.Vlastnik.Adresa.Mesto="Pardubice"  
AND U.Zustatek>100000;
```

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

Různé modely pro databázovou strukturu

databázové modely

SŘBD



NoSQL databáze

- Motivace
 - Obrovské objemy dat (Google, Amazon, Facebook)
 - Vysoká dostupnost
 - Časté změny schématu
 - Specifické požadavky

Klíč – hodnota

(nejstarší NoSQL model)

- Již v r. 1979, jako utilitární modul
- Názvy modelu:
 - asociativní pole, mapa, dictionary, hash
- Struktura
 - klíč – unikátní identifikátor záznamů
 - hodnota – vlastní data záznamu (~blob)

Jediný přístup k datům je pomocí klíče

Obrovské objemy dat

- Distribuce do mnoha uzlů sítě serverů
 - často levné servery
- Opatření:
 - Sharding
 - Replikace

Sharding

- Dělení záznamů do uzlů dle hodnot klíče
- Cíl – rovnoměrná distribuce po uzlech
- Automaticky prováděná
- Lze definovat pravidla
 - např. pro lokální blízkost

Replikace

- Cíl - vysoká dostupnost
- 1 master, několik slave serverů v clusteru
- Čtení dat z replik, zápis dat na master
 - Role master může putovat mezi uzly clusteru
- Systém propaguje zápisy na repliky
 - „eventual consistency“ –
uživatel může nějakou dobu číst „zastaralá“ data,
nakonec se „to strovná“
 - ev. volitelně ACID

NoSQL modely

- Klíč – hodnota
 - základ všech ostatních modelů
- Vnitřní strukturovanost „hodnoty“
 - Dokumentově orientované (Document Stores)
 - „Rodiny“ sloupců (Column Families)
- Provázání záznamů
 - Grafové

Přehled NoSQL databází: <http://nosql-database.org/>

NoSQL modely

- Klíč – hodnota
 - základ všech ostatních modelů
- Vnitřní strukturovanost „hodnoty“
 - Dokumentově orientované (Document Stores)
 - „Rodiny“ sloupců (Column Families)
- Provázání záznamů
 - Grafové

Document Stores

- Záznam strukturován jako „datový agregát“
 - Atributy
 - Zanořování struktur
 - Vícehodnotovost
- Používané formáty:
 - XML
 - YAML
 - JSON
 - BSON

formát JSON

```
{
  "krestniJmeno": "Helena",
  "prijmeni": "Palovská",
  "tituly": {
    "predJmenem": "RNDr.",
    "zaJmenem": "PhD.",
  },
  "telefon": [
    {
      "typ": "kancelář",
      "cislo": "224095458"
    },
    {
      "typ": "mobilní",
      "cislo": ....
    },
  ]
}
```

Struktura záznamů NoSQL

- **Není stanovené schéma**
 - Každý „dokument“ může mít jinou strukturu
- **Někdy podporovány datové typy**
- **Někdy uchovávány verze záznamů,
dostupné dle časových razítek**

Přístupové cesty v Document Stores

- Základem je přístup pomocí klíče
- Datové agregáty je možno indexovat dle nějakých atributů
- Dle atributů je možno vyhledávat (filtrovat)
 - dotazovací jazyky podobné SQL
 - bez JOINů!

Column families

- Motivace:
 - různé části záznamů jsou vyhledávány a zapisovány nezávisle na sobě
 - tyto různé části mohou být nezávisle na sobě ukládány, i na různých uzlech sítě serverů
- Záznamy jsou rozvrženy do „rodin“ sloupců
- V „rodině“ sloupců mohou být
 - sloupce s různým typem informace (strukturovaná rodina sloupců)
 - více sloupců se stejným typem informace (vícehodnotová rodina sloupců)

Column Families

- V každém záznamu, v každé jeho rodině sloupců,
mohou být jiné sloupce.
- Rodiny sloupců tedy nemají pevné schéma.

Column Families

- Standardní, tj. „plochá“
 - obyčejné atributy – sloupce
 - může být určeno řazení sloupců v dané rodině
- Super Column Family
 - zanořuje několik rodin sloupců

- Příklad jednoho záznamu v Column Family databázi

Keyspace

ColumnFamily

Row key	Name 1	...	Name q
	Value 1		Value q
	time stamp		...

ColumnFamily 2

Row key	Name a	...	Name x
	Value a		Value x
	...		...

...

ColumnFamily k

Row key	Name s	...	Name z
	Value s		Value z
	...		...

SuperColumnFamily 1

Row key	Super column 1		
	Name 1		Name 5
	Value 1		Value 5
	time stamp		...
	...		
	Super column 16		
	Name 1		Name 6
	Value 1		Value 6
	...		...

...

SuperColumnFamily m

Row key	Super column 1		
	Name 1		Name 4
	Value 1		Value 4
	time stamp		...
	...		
	Super column 9		
	Name 1		Name w
	Value 1		Value w
	...		...

Zpracování dat v NoSQL

- Dle klíčů záznamů
- Verze záznamů
 - kromě jiného řeší aktualizací konflikty
- MapReduce
 - pro agregace

MapReduce

- Google technologie, zpřístupněna
- Distribuce úloh (Map i Reduce) do uzlů
- Funkce Map
 - vybírá potřebné klíč-hodnota páry z agregátů
- Funkce Reduce
 - postupně vypočítává agregaci

