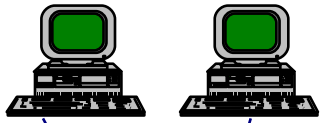
Různé úrovně pohledu na data



vnější pohled

vnější pohled

vnější pohled

konceptuální schéma

Databázové schéma

úložiště jako množina souborů

úložiště jako množina BOIS bloků

Úroveň analytických konceptů

Úroveň implementačních konceptů

Fyzická úroveň

Konceptuální, logická, fyzická uroveň

- Konceptuální – zabývá se modelováním reality, není ovlivněna budoucími prostředky řešení (E-R model, Class Diagram, ...).
- Logická – vztahuje se ke konkrétnímu datovému modelu a používá jeho konstrukční dotazovací a manipulační prostředky (relační, objektová, síťová, hierarchická, XML, ...).
- Fyzická – jde o fyzické uložení dat (sekvenční soubory, indexy, clustery, ...). Programátor je od ní typicky odstíněn vrstvou SŘBD.

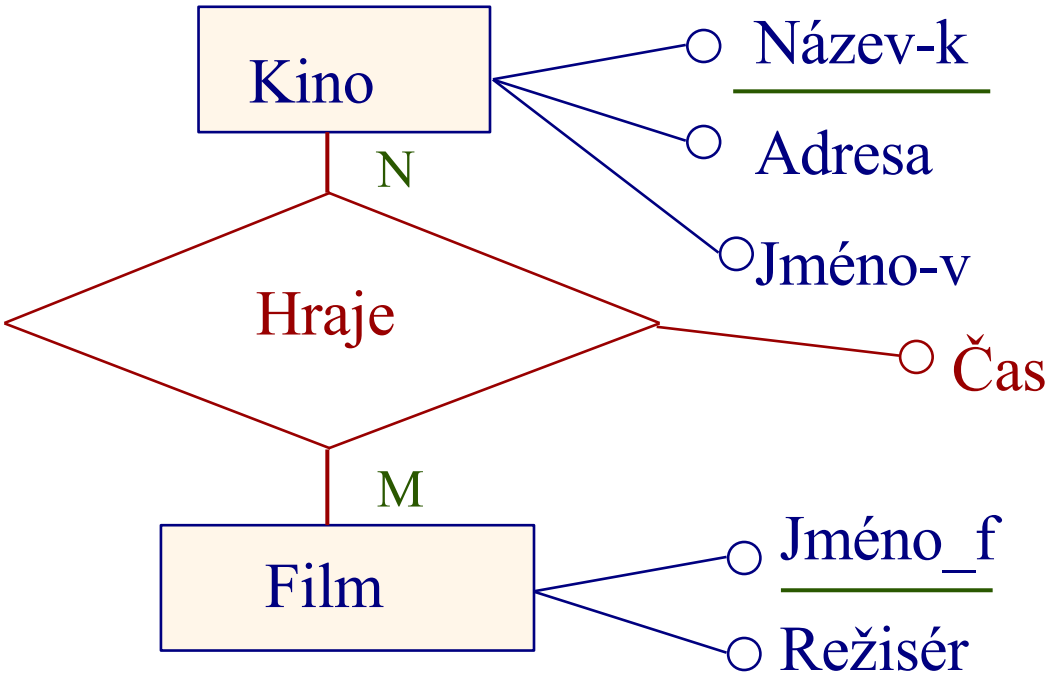
Konceptuální modelování

- v polovině 70. let - konceptuální modely
  - společné chápání objektů aplikace uživateli, projektanty, …
  - integrace různých uživatelských pohledů
  - Výsledek je vstupem pro návrh implementace
- důvody:

  nízká úroveň pohledu na data $\Rightarrow$ obtížná komunikace nad schématem se zákazníkem

Konceptuální modelování

- de facto standard:
  E-R diagramy (1976)
- entitní množiny
  - Atributy entit
- vztahové typy
  - účast ve vztahu
  - Atributy vztahů
- integritní omezení
  - Identifikátory
  - Násobnost účasti (kardinalita a parcialita vztahu)



Logické (databázové) modely

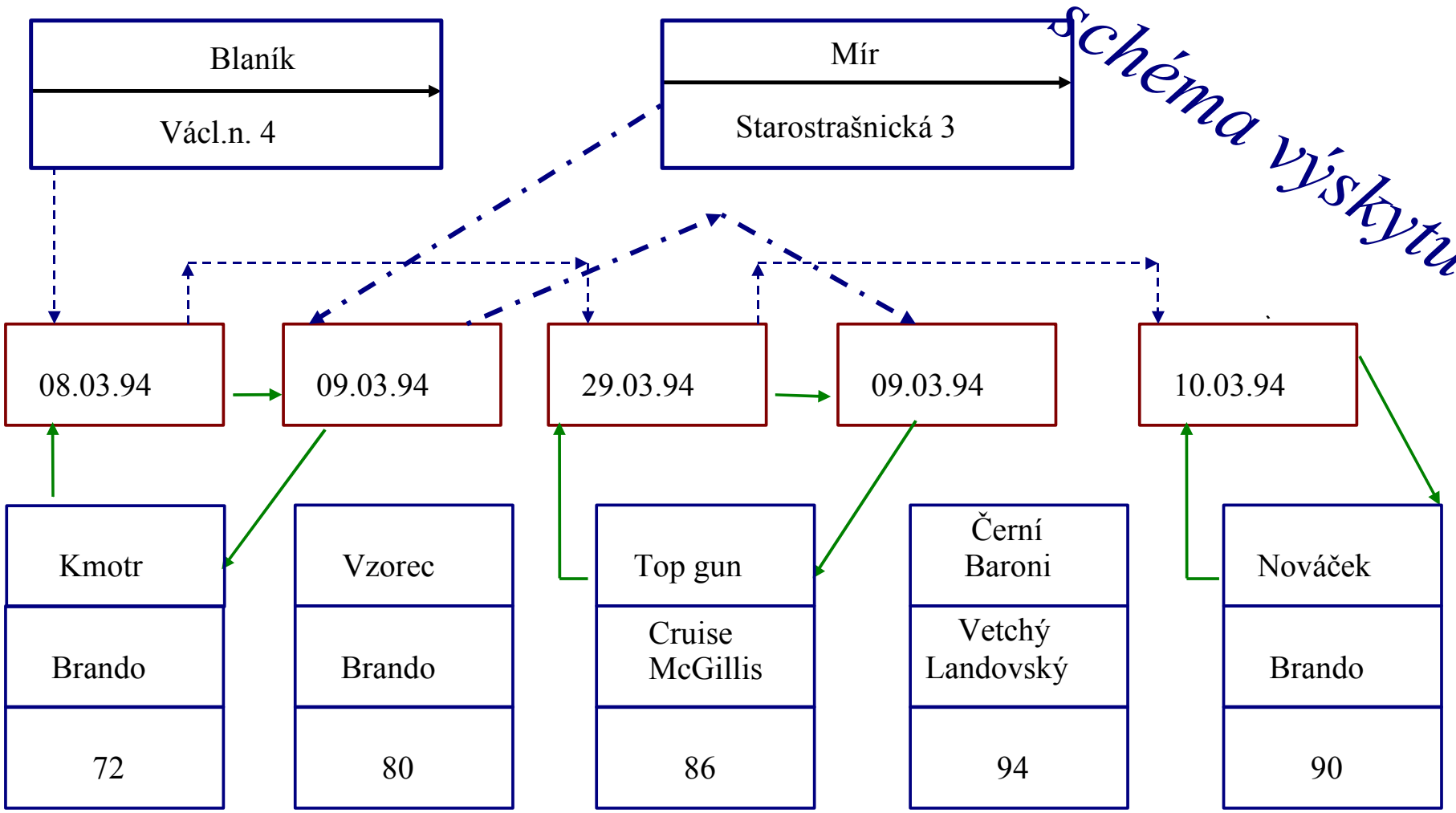
- Síťový model (60. léta 20. století)
- Hierarchický (konec 60. let lze chápat jako specializaci síťového modelu)
- Relační model (začátek 70. let)
- Objektový model (80. léta, lze chápat jako rozšíření síťového modelu)
- Objektově relační model (90. léta – komerčně úspěšný “kočkopes”)
- XML model (konec 90. let, mnoho prvků hierarchického modelu)

Logické (databázové modely)

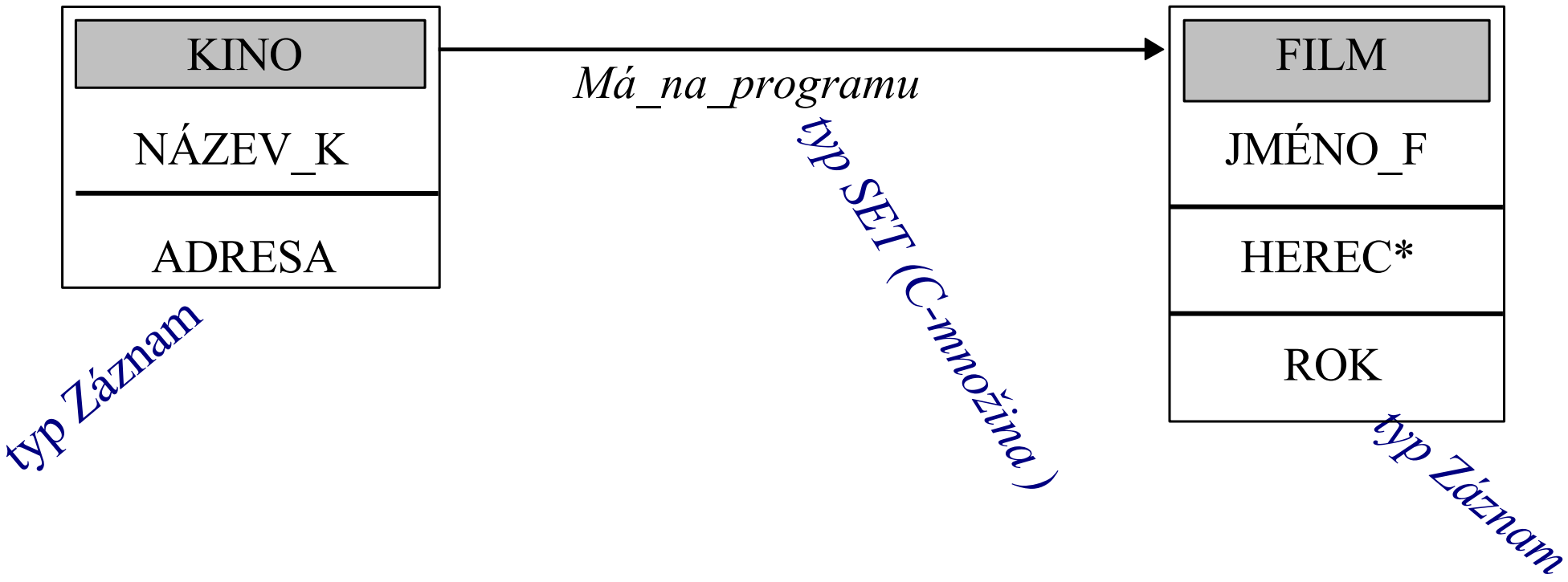
Volba databázového modelu typicky určuje také prostředky pro vytváření struktury databáze (DDL), prostředky pro tvorbu aplikací (DML, dotazovací jazyk, TCL, DCL)

Příklady:
Relační model – SQL
Objektový model – OQL
XML model – Xpath, XQuery

Databázové modely síťový model



Síťový model, diagram na úrovni typů (Bachman)



–každému entitnímu typu odpovídá jeden typ Záznam

–každému vztahovému typu 1:N odpovídá jeden typ Set

Bachmanův diagram, síťový model

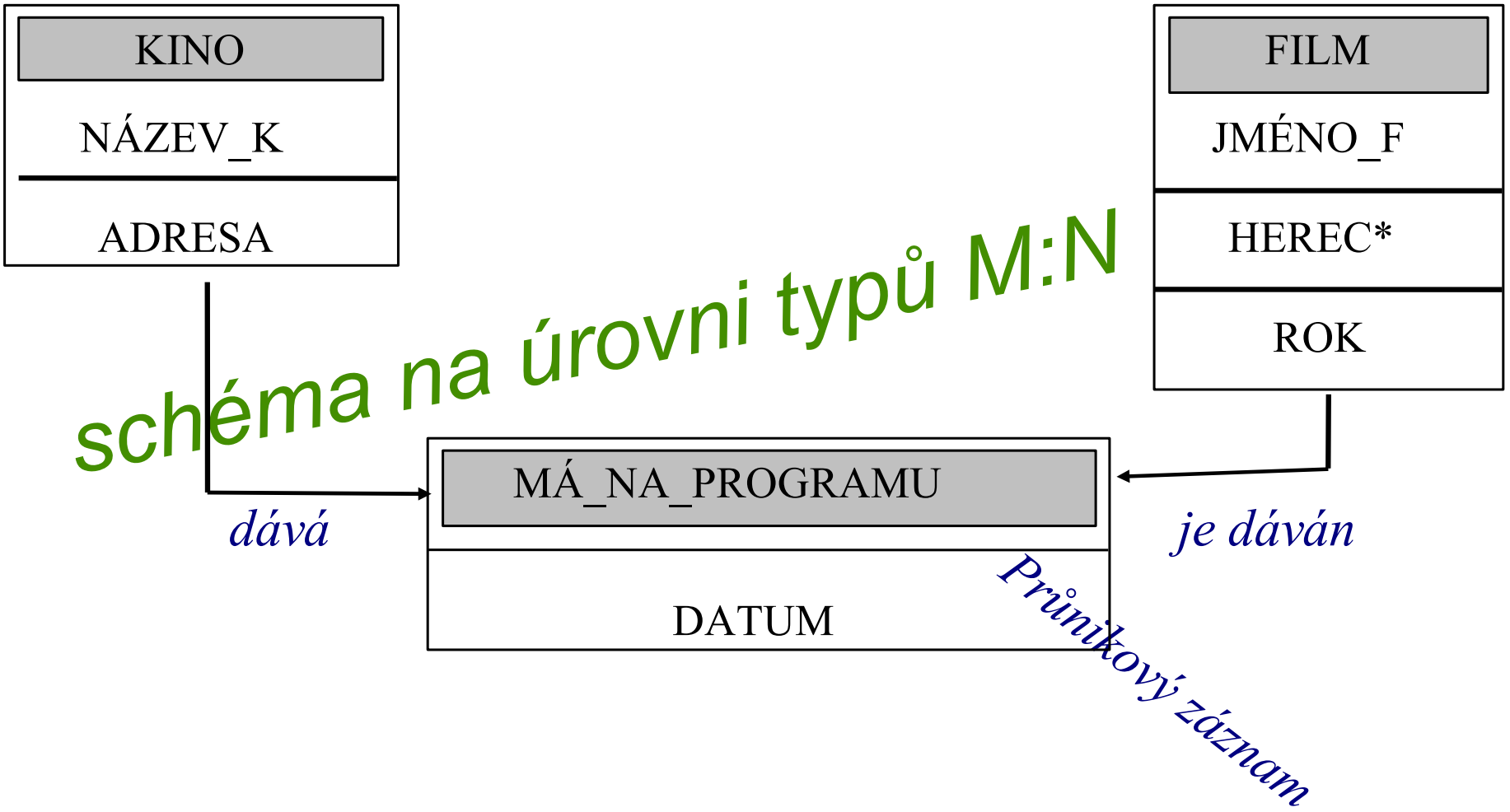
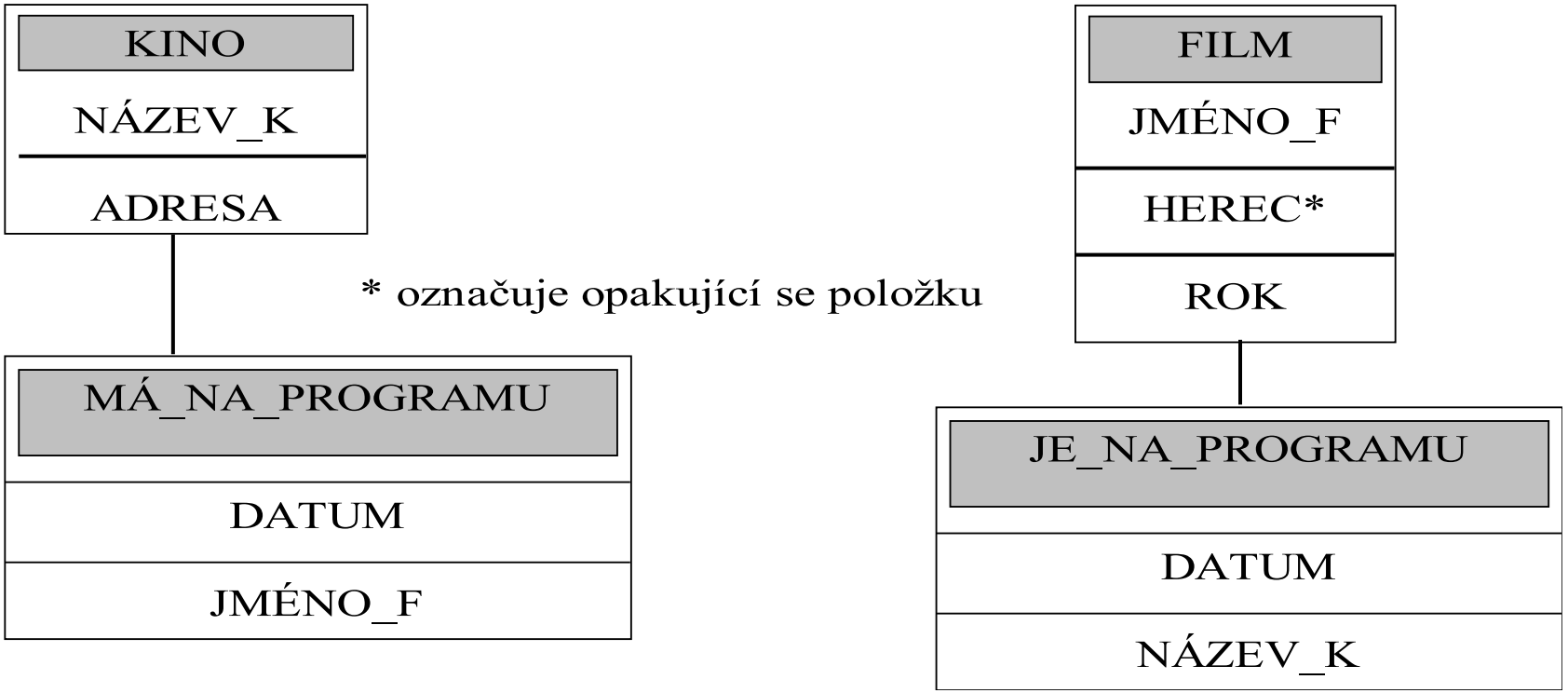


Schéma na úrovni typů hierarchického modelu



konstrukty:
typ segment,
vztah daný typ segmentu je rodič jiného typu segmentu

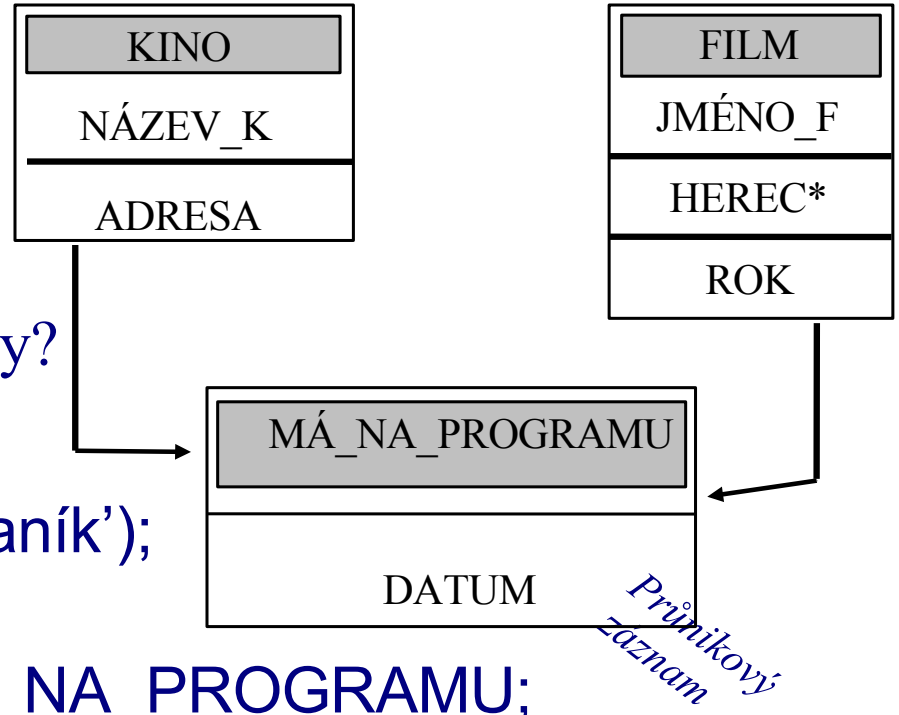
Síťový DB model, datové typy

- Datový typ Record (záznam), který se podobá pascalskému datovému typu *File of record*,
- Datový typ Set (C-množina, dvojice různých datových typů Record), který se podobá datovému typu *Seznam*
- Po transformaci E-R schématu do síťového
  - Každému entitnímu typu odpovídá jeden typ Record
  - Každému vztahovému typu 1:N odpovídá jeden typ Set

Síťový DB model, operace

- vytvoř nový záznam daného typu , zruš daný záznam, změň daný záznam,
- zařaď členský záznam do výskytu c-množiny daného **vlastníka**
- vyřaď daný **člen** z daného výskytu c-množiny
- najdi první **člen** ve výskytu c-množiny daného vlastníka,
- najdi následovníka ve výskytu c-množiny daného vlastníka,
- najdi vlastníka ve výskytu c-množiny známého člena

Navigační jazyk u síťového modelu



Které filmy jsou dávány v kině Blaník a kdy?

```
Begin
 Najdi KINO záznam (NAZEV='Blaník');
 Get KINO;
 Najdi prvního člena v dává – MÁ_NA_PROGRAMU;
 While Not EOL Do
    Get MÁ_NA_PROGRAMU into A;  Print (A.Datum);
    Najdi vlastníka k A v dáván;
    Get FILM into B;  Print (B.Nazev);
    Najdi následovníka v dává;
 End;
End;
```

Relační DB model

- jediný konstrukt - relace
  - schéma relace; jméno relace, jména atributů a specifikace domén atributů
  - prvky domén jsou atomické hodnoty (1. normální forma)
  - formální zápis: R(A1:D1,...,An:Dn)

  KINO(NAZEV_K:CHAR(15),ADRESA:CHAR(25))
- Integritní omezení: primární klíč, cizí klíč

Relační model

KINO	NÁZEV_K	ADRESA
	Blaník	Václ.n. 4
	Vesna	Olšiny 6
	Mír	Strašnická 3
	Domovina	V dvorcích

FILM	JMÉNO_F	HEREC	ROK
	Černí baroni	Vetchý	94
	Černí baroni	Landovský	94
	Top gun	Cruise	86
	Top gun	McGillis	86
	Kmotr	Brando	72
	Nováček	Brando	90
	Vzorec	Brando	80

MÁ_NA PROGRAMU	NÁZEV_K	JMÉNO_F	DATUM
	Blaník	Top gun	29.03.94
	Blaník	Kmotr	08.03.94
	Mír	Nováček	10.03.94
	Mír	Top gun	09.03.94
	Mír	Kmotr	08.03.94

schéma výskytů

Relační model, schéma relací

KINO(NAZEV_K, ADRESA)

FILM(JMENO_F, HEREC, ROK)

MA_NA_PROGRAMU(NAZEV_K, JMENO_F, DATUM)

IO:

- Primární klíče: NAZEV_K
  JMENO_F
  {NAZEV_K, JMENO_F}
- Cizí klíče: MA_NA_PROGRAMU.NAZEV_K
  MA_NA_PROGRAMU.JMENO_F

Relační DB model, operace

- vytvoř novou relaci (tabulku)
- přidej novou n-tici (řádek) do dané relace (tabulky)
- vymaž n-tice (řádky) zadaných vlastností
- vytvoř novou relaci (tabulku) ze zadané relace
  - výběrem n-tic (řádků) zadaných vlastností - **selekce**
  - výběrem zadaných atributů (sloupců) - **projekce**
- ve vybraných n-ticích (řádcích) zadané relace (tabulky) změň hodnoty zadaných prvků (polí)
- vytvoř novou relaci (tabulku) ze zadaných relací (tabulek) pomocí **množinových operací** sjednocení, průnik, rozdíl
- vytvoř novou relaci (tabulku) ze zadaných relací (tabulek) jako podmnožinu jejich kartézského součinu - **spojení**

Dotazování v relační databázi

Které filmy jsou dávány v kině Blaník a kdy?

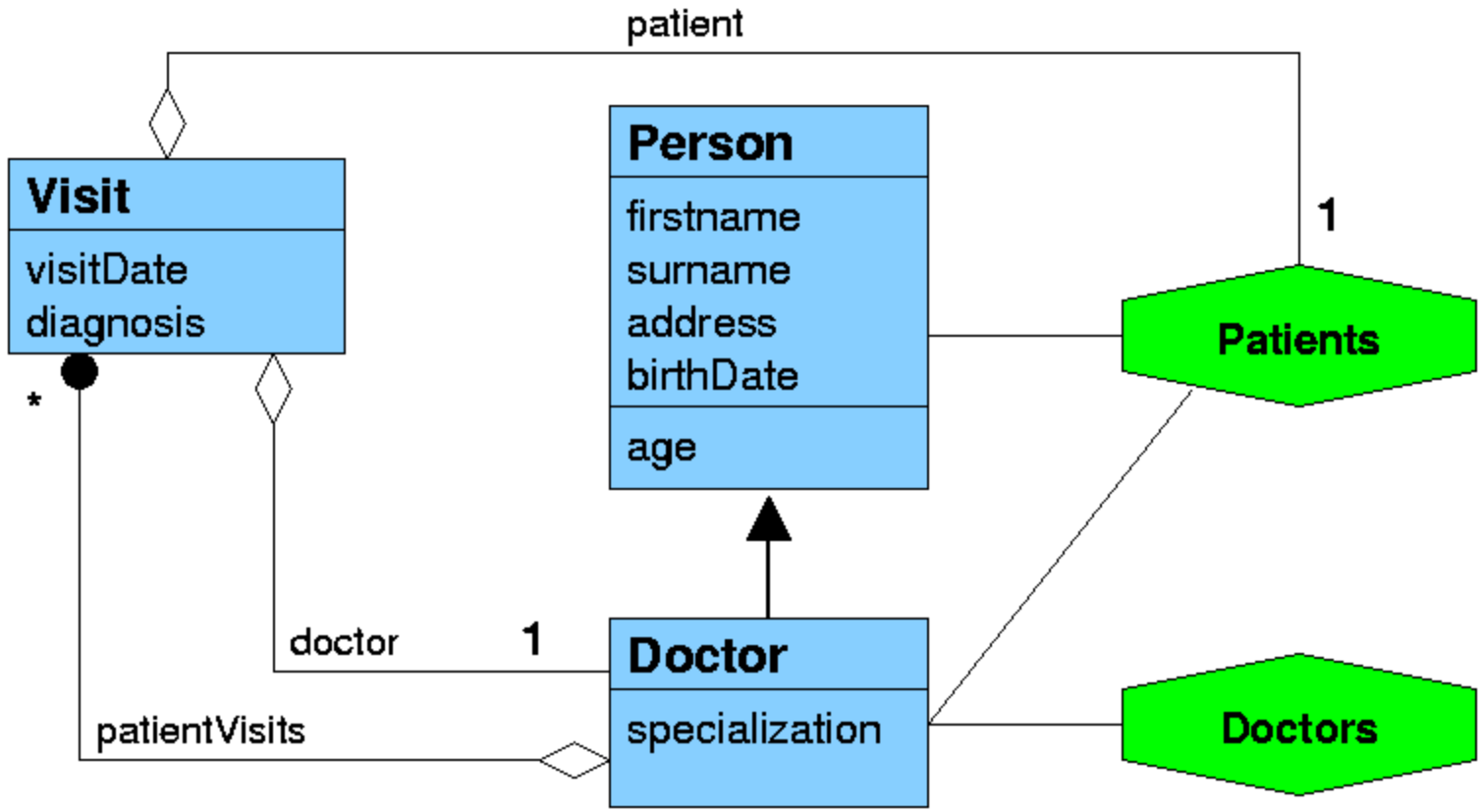
```
(KINO(NAZEV_K='Blaník')* MA_NA_PROGRAMU *
 FILM)[jmeno_f,datum]
```

```
Select Jmeno_F, Datum
From KINO K JOIN MA_NA_PROGRAMU MNP
            ON (K.NAZEV_K='Blaník' and
                K.NAZEV_K= MNP.NAZEV_K)
            JOIN FILM USING(Jmeno_F)
```

Integritní omezení (IO)

- IO jsou tvrzení vymezující korektnost DB
- definují se na konceptuální i databázové úrovni
  - název_k jednoznačně *určuje* řádky tabulky *Kina*
  - daný film si lze *rezervovat* v půjčovně, jen když jsou všechny jeho kopie *vypůjčeny*
  - čtenář si může *vypůjčit* nejvýše 6 filmů (kopií)
  - vypůjčující musí být v seznamu *zákazníků*

Objektový DB model - příklad



Objektový datový model - charakteristika

- Objekty = data + metody. Mezi objekty existuje skládání, dědění, závislost, klasifikace podle tříd, ... Strukturované informace není třeba rozdělovat jako v RDM.
- Protokol objektu je dán množinou přípustných zpráv (ne atributů jako v RDM).
- Jedna množina může s využitím polymorfismu obsahovat objekty s různou strukturou dat i metod.
- Je rozdíl mezi množinou objektů a třídou.
- Identita objektu je dána nejen vnitřními, ale i vnějšími vazbami. Klíče jsou interní záležitostí systému.

XML databázový model

- Podobá se hierarchickému (XML dokument je chápán jako strom (DOM))
- Aplikační doména? (vhodnost použití)
- Datový model: elementy, atributy, PCDATA, zachování pořadí (document order). Někdy bohatší.
- Silné a standardizované dotazovací jazyky (XPath, XQuery)
- Monoho implementací a mnoho věcí stále ve vývoji (indexování, zamýkání, ...)

XML DB model – příklad datově or.

```
<addressbook>
<person>
        <name> Michal Valenta </name>
<tel> 2 2435 7313 </tel>
         <tel> 2 2435 7258 </tel>
<email> valenta@fel.cvut.cz </email>
</person>
<person>
        <name> Josef Švejk </name>
<email> svejk.josef@hasek.net </email>
</person>
</addressbook>
```

XML DB model – příklad dokum. or.

```
<?xml version="1.0" encoding="UTF-8"?>
 <slidesinfo>
  <title>XML Database Mnagement
 Systems</title>
  <author>MichalValenta</author>
 </slidesinfo>
 <foil>
  <title>XML DBMS "definition"</title>
  <orderedlist>
   <listitem>
    <para>Defines a (logical) model for an
    <emphasis>XML document</emphasis> and
    stores and retrieves documents according ...
```

XML DB model – příklad DTD str.

SCHEDULE (DATE?, STATION+)

STATION (NETWORK*, CALL_LETTERS*, CHANNEL*, SHOW+)

SHOW (NAME?,TITLE*, TYPE? , EPISODE_NUMBER? ,

START_TIME? , LENGTH? , AIR_DATE?,
ORIGINAL_AIR_DATE?,YEAR_MADE?, CLOSED_CAPTIONED?,
REPEAT? ,RATING*, STARS*, DESCRIPTION* , DIRECTOR*,
WRITER*, CAST*, PRODUCER*)

DIRECTOR (GIVEN_NAME* ,MIDDLE_INITIAL*, SURNAME*)+

WRITER (GIVEN_NAME* ,MIDDLE_INITIAL*, SURNAME*)+

CAST (ACTOR*)

ACTOR (GIVEN_NAME* ,MIDDLE_INITIAL*, SURNAME*)

PRODUCER (GIVEN_NAME*, MIDDLE_NAME*, SURNAME*)

<!-- Attributes -->

<!ATTLIST SCHEDULE id ID #IMPLIED>

XML DB model – příklady XPath

Všecny tituly v databázi

/BOOK/TITLE

Všichni autoři

//AUTHOR/

Kapitoly, které se skládají přesně ze dvou sekcí

//CHAPTER[count(SECTION) = 2]

Knihy z kategorie “technology” napsané autorem Scott.

/BOOK[@CLASS='technology' and AUTHOR='Scott']

XML DB model – příklady XQuery

Rok a názvy knih vydaných před rokem 2000.

```
for $book in /BOOKS/BOOK
where $book/@YEAR &lt; 2000
 return <BOOK>
            { $book/@YEAR, $book/TITLE}
        </BOOK>
```

Názvy knih seskupené podle autorů.

```
for $author in   distinct(/BOOKS/BOOK/AUTHOR)
return <AUTHOR NAME="{ $author }">
          { /BOOKS/BOOK[AUTHOR = $author]/TITLE}
       </AUTHOR>
```

V tomto předmětu se budeme dále věnovat ***výhradně relačnímu databázovému modelu.***

Je dobře si uvědomit, že:
1. RM není jediný, ze kterého si můžeme vybírat.
2. Pro určitý typ aplikace nebo aplikační doménu můžeme výběrem vhodného DB modelu mnoho ušetřit.
3. Volbu DB modelu je třeba dobře uvážit a zdůvodnit.