

Databázové systémy

Vladislav Novák

1. cvičenie

Obsah cvičení počas semestra

1. polovica semestra - návrh databáz, dátový model
princípy platné pri tvorbe každého softvéru, nie len pri tom, čo bežne voláme databázy
2. polovica semestra - programovací jazyk SQL
používaný v relačných databázach

Web stránka ku cvičeniam v 1. polovici semestra

- <https://dbs.useobjects.net>
- pracovná verzia učebného textu (môže sa meniť)

Moodle

- Testy počas 1. polovice semestra (každý týždeň)
- Literatúra z predchádzajúcich rokov
- <https://elearn.elf.stuba.sk/moodle/>
- Postup prihlásenia:
 - v pravo hore zvolte *Prihlásiť sa*, použite prihlasovacie údaje do AIS (nevytvárajte si nové konto)
 - v časti *Kategórie kurzov* zvolte: *Aplikovaná informatika* a v nej kurz *Databázové systémy*
 - prihláste sa pomocou prihlasovacieho kľúča (bol zaslaný emailom)

Domáce úlohy - testy v moodle

- v testoch základy, na písomkách budú náročnejšie úlohy
- v prvej polovici semestra
- každý týžden
- na vypracovanie približne týžden
- každý test je potrebné vypracovať do utorka 23:59 v nasledujúcom týždni
- každá úloha v teste bude za 0,05 bodu
 - spolu zo všetkých testov minimálne 7 bodov (nad 7 bodov bonus)
- test môžete opakovať ľubovoľný počet krát
 - do hodnotenia sa bude brať najlepšie dosiahnuté hodnotenie
- počet úloh môže byť každý týždeň iný

Testy (domáce úlohy)

 1. test (DÚ)

Bodové hodnotenie

- Testy v Moodle - konceptuálny a logický model relačnej databázy
 - 7 bodov (v skutočnosti okolo 7,5 bodu)
- Písomka - konceptuálny model, logický model relačnej databázy
 - 43 bodov
 - v polovici semestra
- Písomka – SQL
 - 30 bodov
- Písomka – princípy implementácie relačných databáz
 - 20 bodov
- Minimum na zápočet = minimum na absolvovanie predmetu = 56 zo 100 bodov

Cvičenia (1. polovica semestra)

- MS Teams
- Videozáznamy ukladané v časti *Files* (komunikačného kanálu *Cvičenia*)

Konzultácie (1. polovica semestra)

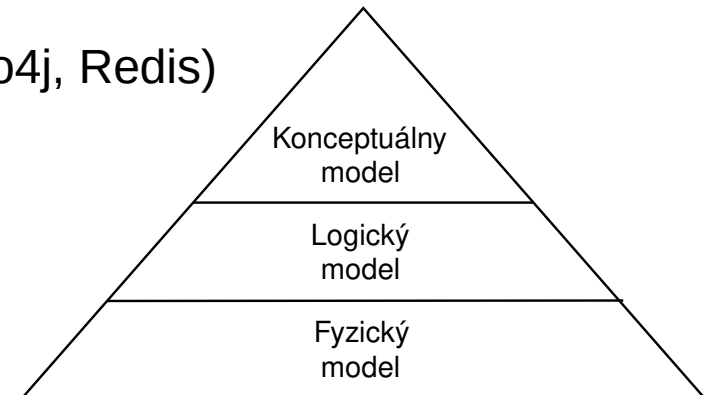
- po cvičení
- úloha na riešenie
- možnosť konzultovať riešenie zadanej úlohy počas jej vypracovávania
- možnosť konzultovať aj ďalšie preberané učivo na predmete
- vypracovanie testu
- dobrovoľné

Hierarchia dátových modelov

- konceptuálny návrh
 - nezávislý na type databázy
- logický model
 - závislý od typu databázy (na akom princípe pracuje)
 - (relačná, grafová, dokumentová, ..., databáza)
- fyzický model
 - závislý od špecifických vlastností implementácie
 - (Oracle, Postgresql, MySQL, SQLite, MongoDB, Neo4j, Redis)

1. až 3. cvičenie

4. až 6. cvičenie



Dátový model

- diagram
- textový popis obsahuje to, čo sa nedá zachytiť v diagrame

Konceptuálny model

- entitno-relačný model
 - základné prvky
 - grafický zápis: entitno-relačný diagram
 - angl.: entity-relationship model/diagram
 - skratky: ER model (ERM) / ER diagram (ERD)
- rozšírený entitno-relačný model
 - obsahuje viaceré rozšírenia
 - grafický zápis: rozšírený entitno-relačným diagram
 - angl.: enhanced entity-relationship model/diagram
 - skratky: EER model (EERM) / EER diagram (EERD)

(E)ER model
nie je štandardizovaný,
v rôznej literatúre
rôzne grafické značky
ale aj význam

Na predmete
budeme používať
zvolené značenie
a sémantiku (význam)

Príklad 1.1 – stavebná firma

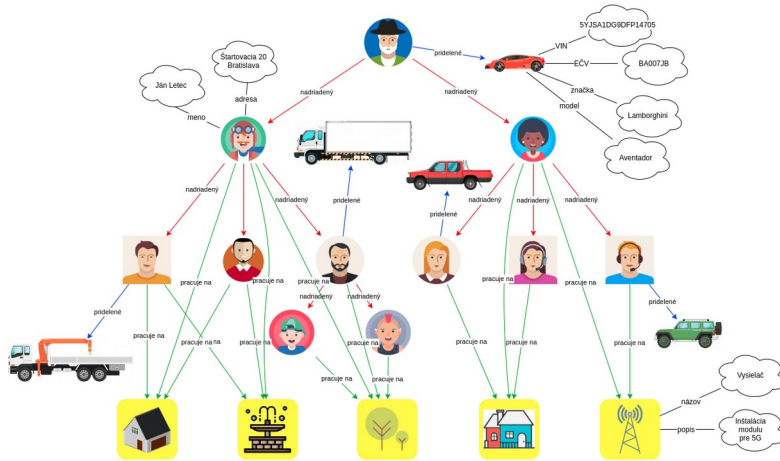
Vytvorte dátový model firmy, pre informačný systém, ktorý bude evidovať

- hierarchiu zamestnancov,
- projekty na ktorých firma a jej zamestnanci pracujú,
- pridelenie áut zamestnancom.

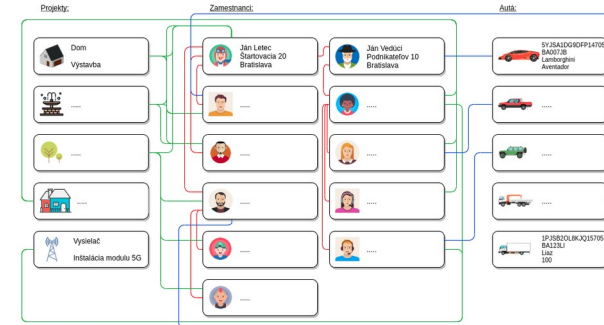
- Identifikujte typy entít a vzťahov medzi nimi. V prípade potreby aj role vo vzťahoch. Identifikujte, aké údaje o entitách a vzťahoch bude systém obsahovať. Doplňte model o atribúty.
- Naznačte možnosti implementácie v C++ a v tabuľkovom editore.
- Rozložte zložené atribúty. Identifikujte viachodnotové atribúty.
- Doplňte evidenciu počtu odpracovaných hodín zamestnancov pre jednotlivé projekty a celkový počet hodín odpracovaných na každom projekte.
- Kedy použiť entitu a kedy atribút?

Príklad 1.1 – stavebná firma – riešenie a)

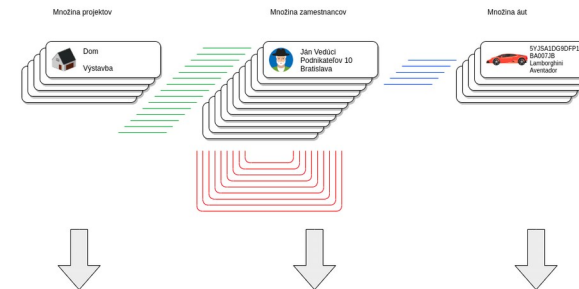
Inštanície v reálnom svete



Inštanície ako údaje zapísané na papierových kartách/listkoch



množiny entít a množiny vzťahov (množiny inštancií)



model: typy entít a vzťahov medzi nimi, role vo vzťahu



Príklad 1.1 – stavebná firma

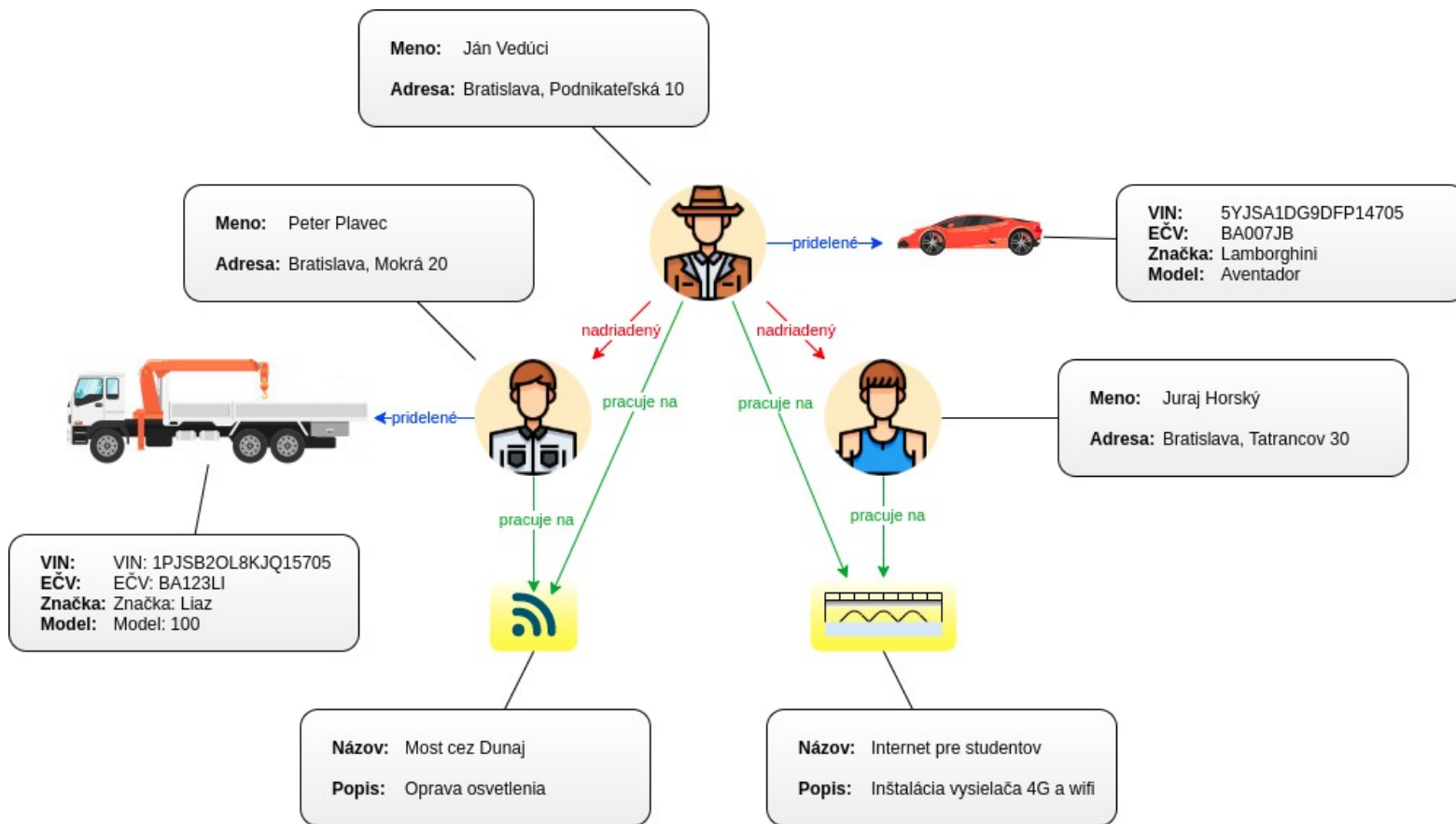
Vytvorte dátový model firmy, pre informačný systém, ktorý bude evidovať

- hierarchiu zamestnancov,
- projekty na ktorých firma a jej zamestnanci pracujú,
- pridelenie áut zamestnancom.

- Identifikujte typy entít a vzťahov medzi nimi. V prípade potreby aj role vo vzťahoch. Identifikujte, aké údaje o entitách a vzťahoch bude systém obsahovať. Doplňte model o atribúty.
- Naznačte možnosti implementácie v C++ a v tabuľkovom editore.
- Rozložte zložené atribúty. Identifikujte viachodnotové atribúty.
- Doplňte evidenciu počtu odpracovaných hodín zamestnancov pre jednotlivé projekty a celkový počet hodín odpracovaných na každom projekte.
- Kedy použiť entitu a kedy atribút?

Príklad 1.1 – stavebná firma – riešenie b)

Pre prehľadnosť bude v implementácii menej údajov



Príklad 1.1 – stavebná firma – riešenie b1) – C++

```
struct Zamestnanec {
    string meno;
    string adresa;
    Zamestnanec *nadriadeny;
    // Auto *prideleneAuto;
    // list<Projekt*> projekty;
};

struct Auto {
    string vin;
    string ecv;
    string znacka;
    string model;
    Zamestnanec *zamestnanec;
};

struct Projekt {
    string nazov;
    string popis;
    list<Zamestnanec*> zamestnanci;
};
```

```
Zamestnanec jan;
Zamestnanec peter;
Zamestnanec juraj;
Auto sportiak;
Auto nakladiak;
Projekt most;
Projekt vysielac;
```

premenne
napríklad
vo funkcii main

Príklad 1.1 – stavebná firma – riešenie b1) – C++

```
struct Zamestnanec {
    string meno;
    string adresa;
    Zamestnanec *nadriadeny;
    // Auto *prideleneAuto;
    // list<Projekt*> projekty;
};

struct Auto {
    string vin;
    string ecv;
    string znacka;
    string model;
    Zamestnanec *zamestnanec;
};

struct Projekt {
    string nazov;
    string popis;
    list<Zamestnanec*> zamestnanci;
};
```

```
Zamestnanec jan {
    .meno = "Jan Hlavny",
    .adresa = "Bratislava, Podnikatelska 10",
    .nadriadeny = nullptr
};

Zamestnanec peter {
    .meno = "Peter Plavec",
    .adresa = "Bratislava, Mokra 20",
    .nadriadeny = &jan
};

Zamestnanec juraj {
    .meno = "Juraj Horsky",
    .adresa = "Bratislava, Tatrancov 30",
    .nadriadeny = &jan
};
```

inicializované
premenné

```
Projekt most {
    .nazov = "Most cez Dunaj",
    .popis = "Oprava .....",
    .zamestnanci = {&jan, &peter}
};

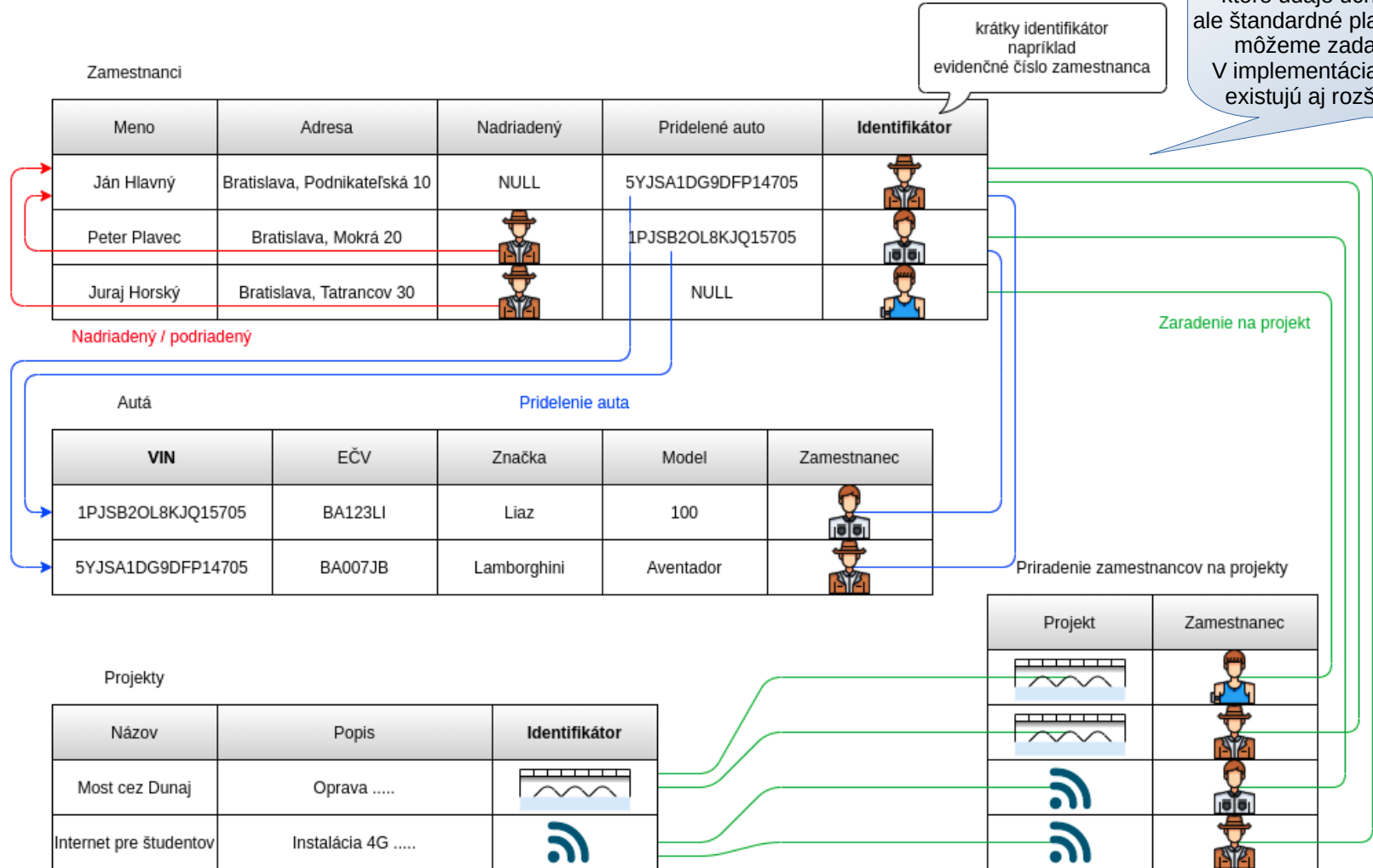
Projekt vysielac {
    .nazov = "Internet pre studentov",
    .popis = "Instalacia 4G .....",
    .zamestnanci = {&jan, &juraj}
};

Auto sportiak {
    .vin = "5YJSA1DG9DFP14705",
    .ecv = "BA007JB",
    .znacka = "Lamborghini",
    .model = "Aventador",
    .zamestnanec = &jan
};

Auto nakladiak {
    .vin = "1PJSB2OL8KJQ15705",
    .ecv = "BA123LI",
    .znacka = "Liaz",
    .model = "100",
    .zamestnanec = &peter
};
```

Príklad 1.1 – stavebná firma – riešenie b)

V tabuľkovom editore určenom pre kancelárske účely môžeme do bunky v tabuľke napísať ľubovoľný text. V relačných databázach, ktoré údaje uchovávajú v tabuľkách, ale štandardne platí pravidlo, že do bunky môžeme zadať len jednu hodnotu. V implementáciach relačných databáz existujú aj rozšírenia, kde to neplatí.



Príklad 1.1 – stavebná firma

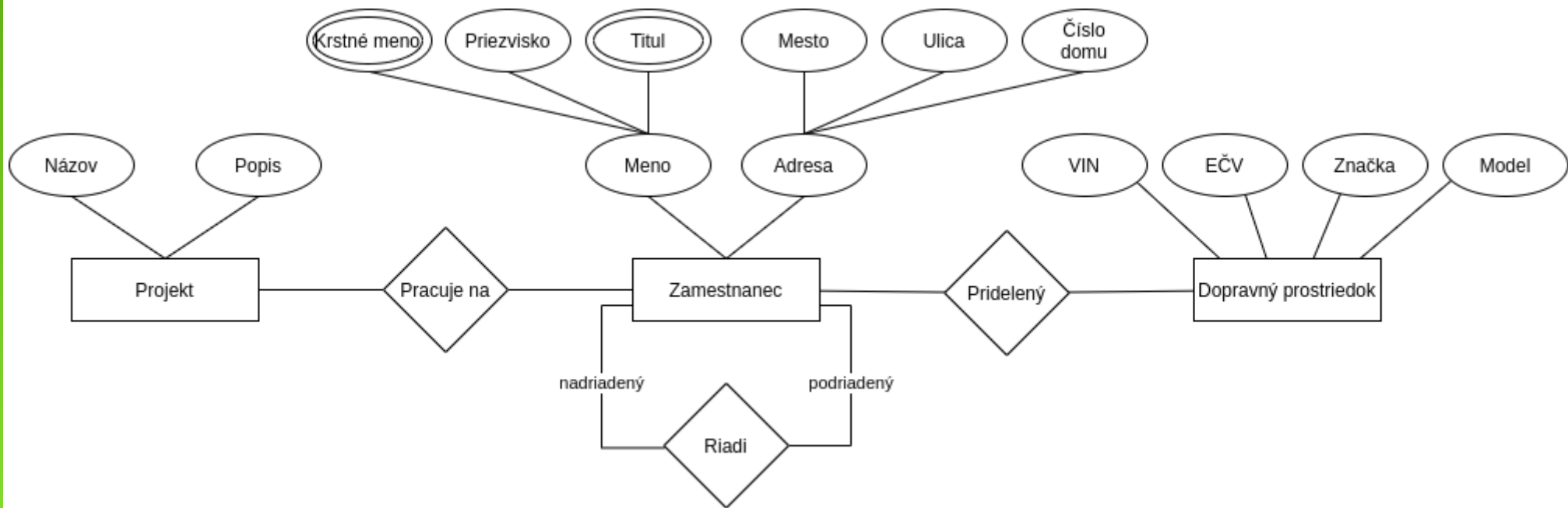
Vytvorte dátový model firmy, pre informačný systém, ktorý bude evidovať

- hierarchiu zamestnancov,
- projekty na ktorých firma a jej zamestnanci pracujú,
- pridelenie áut zamestnancom.

- Identifikujte typy entít a vzťahov medzi nimi. V prípade potreby aj role vo vzťahoch. Identifikujte, aké údaje o entitách a vzťahoch bude systém obsahovať. Doplňte model o atribúty.
- Naznačte možnosti implementácie v C++ a v tabuľkovom editore.
- Rozložte zložené atribúty. Identifikujte viachodnotové atribúty.
- Doplňte evidenciu počtu odpracovaných hodín zamestnancov pre jednotlivé projekty a celkový počet hodín odpracovaných na každom projekte.
- Kedy použiť entitu a kedy atribút?

Príklad 1.1 – stavebná firma – riešenie c)

Riešenie – zložené a viachodnotové atribúty



Príklad 1.1 – stavebná firma

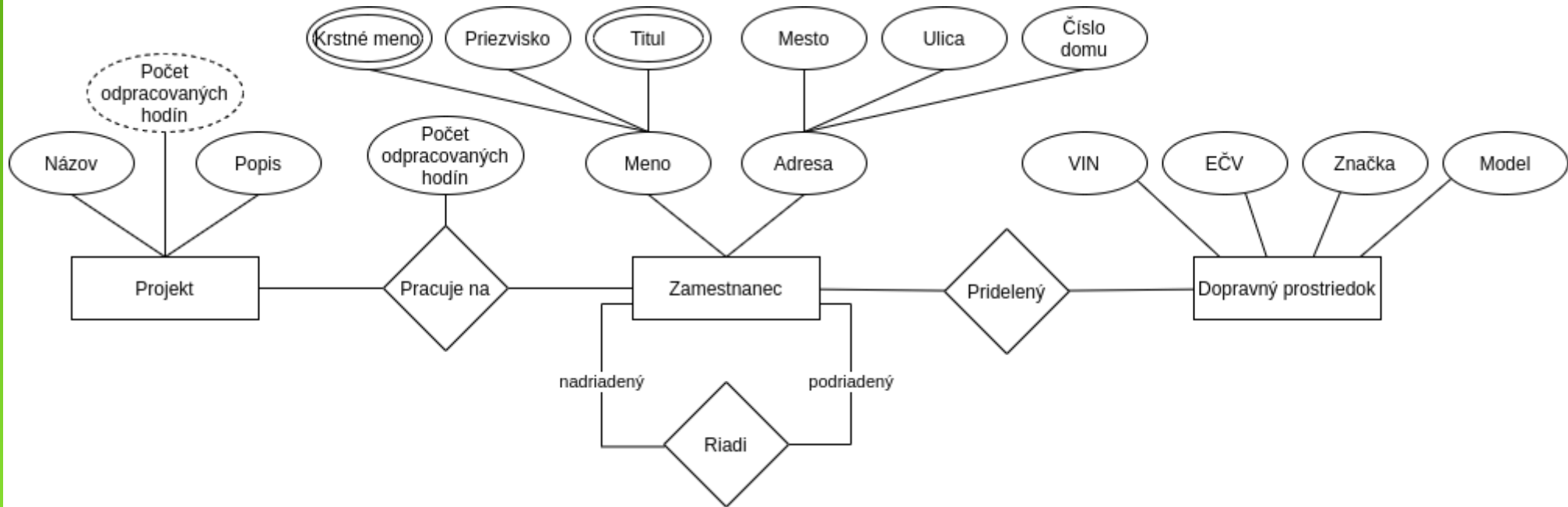
Vytvorte dátový model firmy, pre informačný systém, ktorý bude evidovať

- hierarchiu zamestnancov,
- projekty na ktorých firma a jej zamestnanci pracujú,
- pridelenie áut zamestnancom.

- Identifikujte typy entít a vzťahov medzi nimi. V prípade potreby aj role vo vzťahoch. Identifikujte, aké údaje o entitách a vzťahoch bude systém obsahovať. Doplňte model o atribúty.
- Naznačte možnosti implementácie v C++ a v tabuľkovom editore.
- Rozložte zložené atribúty. Identifikujte viachodnotové atribúty.
- Doplňte evidenciu počtu odpracovaných hodín zamestnancov pre jednotlivé projekty a celkový počet hodín odpracovaných na každom projekte.
- Kedy použiť entitu a kedy atribút?

Príklad 1.1 – stavebná firma – riešenie d)

Odpracované hodiny
- atribút vzťahu
- odvodený atribút



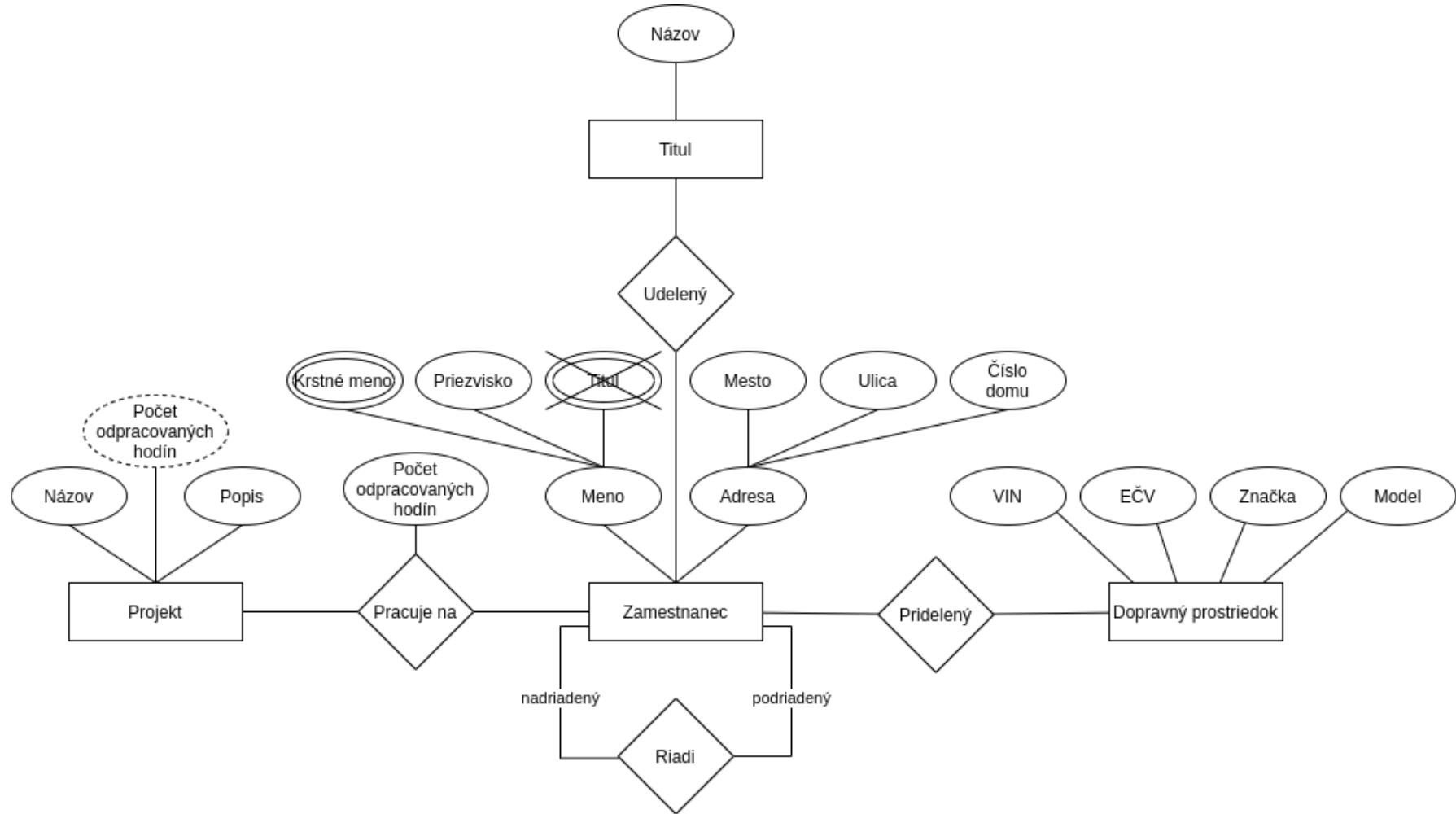
Príklad 1.1 – stavebná firma

Vytvorte dátový model firmy, pre informačný systém, ktorý bude evidovať

- hierarchiu zamestnancov,
- projekty na ktorých firma a jej zamestnanci pracujú,
- pridelenie áut zamestnancom.

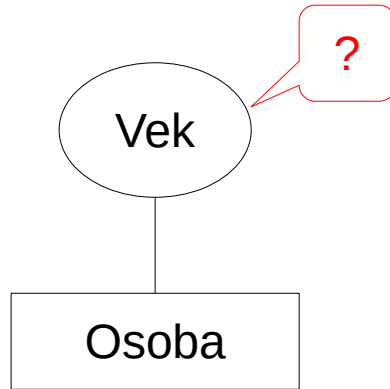
- Identifikujte typy entít a vzťahov medzi nimi. V prípade potreby aj role vo vzťahoch. Identifikujte, aké údaje o entitách a vzťahoch bude systém obsahovať. Doplňte model o atribúty.
- Naznačte možnosti implementácie v C++ a v tabuľkovom editore.
- Rozložte zložené atribúty. Identifikujte viachodnotové atribúty.
- Doplňte evidenciu počtu odpracovaných hodín zamestnancov pre jednotlivé projekty a celkový počet hodín odpracovaných na každom projekte.
- Kedy použiť entitu a kedy atribút?

Príklad 1.1 – stavebná firma – diskusia e)



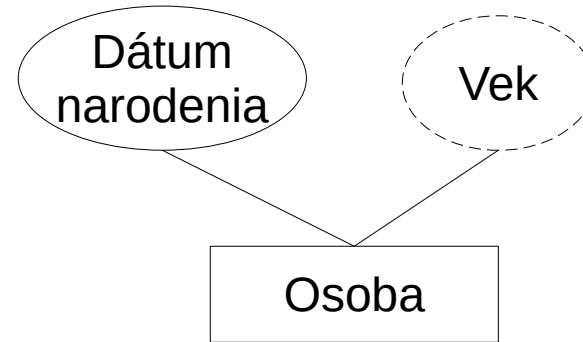
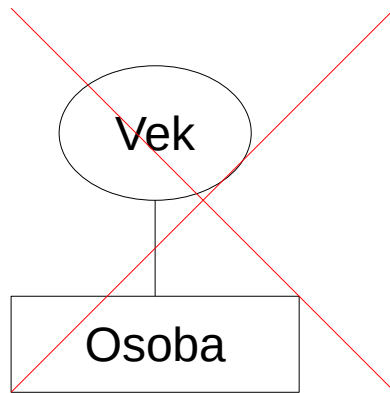
Príklad 1.2 – Vek osoby

Reprezentujte osobu a jej vek



Príklad 1.2 – Vek osoby

Reprezentujte osobu a jej vek



Príklad 1.3 – nedefinovaná hodnota atribútu

Hodnota atribútu nemusí byť v databáze vždy zaznamenaná.

Príklad 1.3 – nedefinovaná hodnota atribútu

Hodnota atribútu nemusí byť v databáze vždy zaznamenaná.

Uveďte príklad:

- neznámej hodnoty atribútu
- nedefinovanej hodnoty atribútu

Príklad 1.3 – nedefinovaná hodnota atribútu

Hodnota atribútu nemusí byť v databáze vždy zaznamenaná.

Uveďte príklad:

- neznámej hodnoty atribútu
- nedefinovanej hodnoty atribútu

Neznáma hodnota atribútu:

- Hmotnosť výrobku (existuje ale nemusíme ju poznať)

Nedefinovaná hodnota atribútu:

- Ulica v adrese, ak obec nemá ulice (neexistuje)

Príklad 1.3 – nedefinovaná hodnota atribútu

Hodnota atribútu nemusí byť v databáze vždy zaznamenaná.

Uveďte príklad:

- neznámej hodnoty atribútu
- nedefinovanej hodnoty atribútu

Neznáma hodnota atribútu:

- Hmotnosť výrobku (existuje ale nemusíme ju poznať)

Nedefinovaná hodnota atribútu:

- Ulica v adrese, ak obec nemá ulice (neexistuje)

Ako označujeme neznámu alebo nedefinovanú hodnotu atribútu?

Príklad 1.3 – nedefinovaná hodnota atribútu

Hodnota atribútu nemusí byť v databáze vždy zaznamenaná.

Uveďte príklad:

- neznámej hodnoty atribútu
- nedefinovanej hodnoty atribútu

Neznáma hodnota atribútu:

- Hmotnosť výrobku (existuje ale nemusíme ju poznať)

Nedefinovaná hodnota atribútu:

- Ulica v adrese, ak obec nemá ulice (neexistuje)

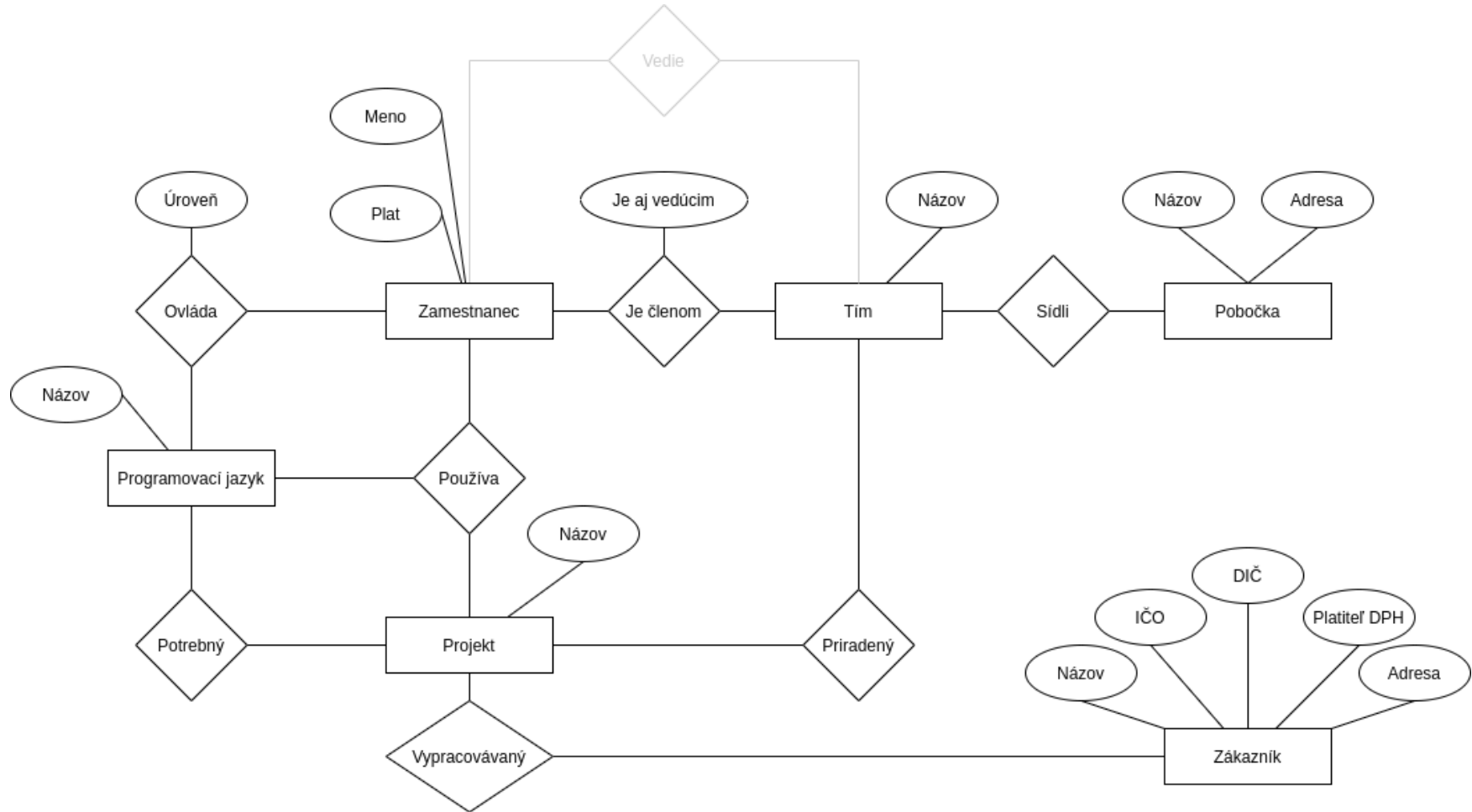
Ako označujeme neznámu alebo nedefinovanú hodnotu atribútu?

Neznáma alebo nedefinovaná hodnota atribútu je označovaná NULL.

Príklad 1.4 – softvérová firma

Navrhňte konceptuálny model databázy firmy. Databáza bude obsahovať informácie o zamestnancoch, ich zaradení do tímov a informácie o vedúcich tímov. Firma má viacero pobočiek. Každý tím pracuje na jednej z pobočiek. Každý tím môže pracovať na viacerých projektoch. Viaceré projekty môžu byť zadané tým istým zákazníkom. Databáza obsahuje informáciu o úrovni ovládania programovacích jazykov zamestnancami. Každém projekte sú pridelené aj programovacie jazyky, ktoré sú pre jeho realizáciu potrebné. Tiež obsahuje informáciu o tom, ktorý zamestnanec používa ktoré jazyky na ktorých projektoch.

Príklad 1.4 – softvérová firma



Príklad 1.4 – softvérová firma

Množiny entít

Množina zamestnancov = {  }

Množina programovacích jazykov = {  ,  ,  ,  ,  ,  ,  ,  }

Množina tímov = {  }

Množina projektov = {  }

Na slajde nie sú všetky množiny definované modelom

Príklad obsahu databázy

Množiny vzťahov

binárny vzťah - množina dvojíc

Množina členstiev v tímoch = {  ,  } }

binárny vzťah - množina dvojíc

Zamestnanec ovláda = {  ,  } atď. }

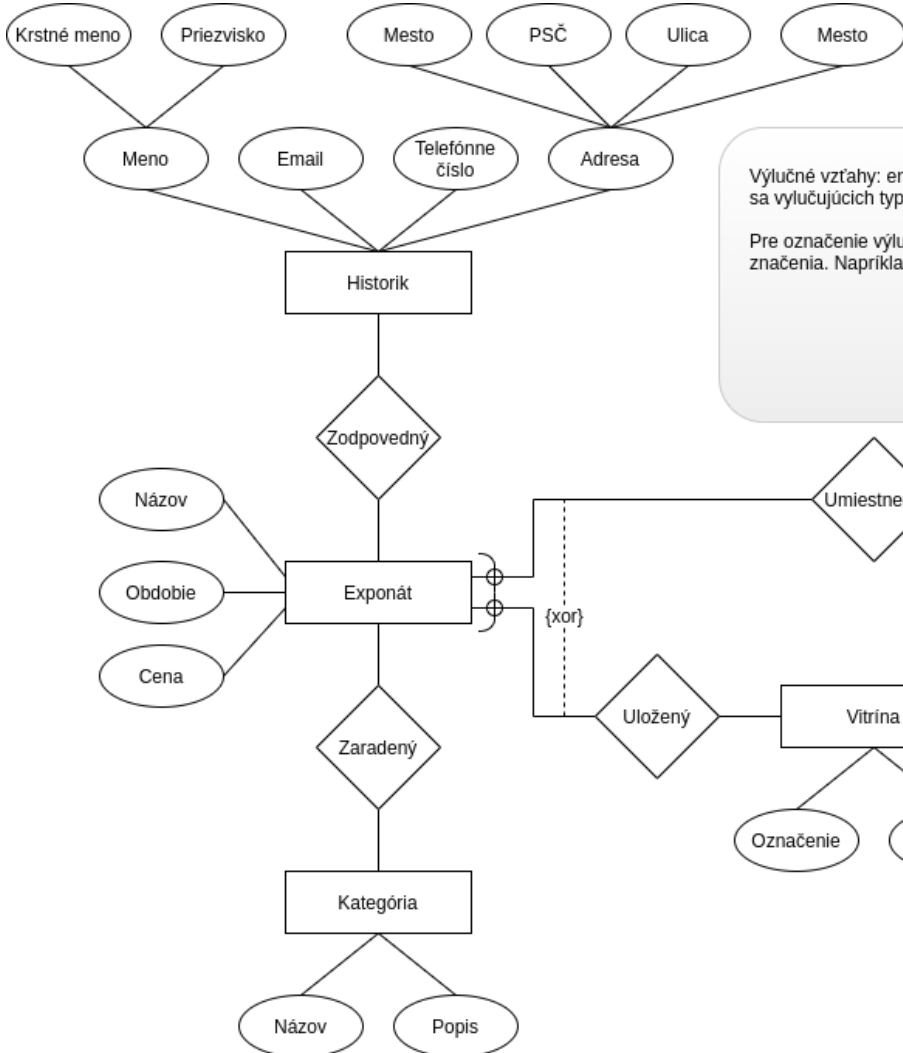
ternárny vzťah - množina trojíc

Zamestnanec používa na projekte = {  ,  ,  } atď. }

Príklad 1.5 - Múzeum

Navrhňte konceptuálny model databázy pre malé múzeum. Databáza musí obsahovať katalóg exponátov. Umožnite evidovať odhadovaný vek, cenu a druh exponátu. Navrhňte aj ďalšie informácie o exponátoch, ktoré bude databáza obsahovať. Malé exponáty sú uložené vo vitrínach, veľké priamo na zemy vo výstavných sálach. Múzeum má viacero výstavných sál a depozitár, kde sú uložené exponáty. V múzeu sú zamestnanci, ktorí sa starajú o exponáty. V databáze evidujte pridelenie exponátov zamestnancom. V návrhu doložte ďalšie potrebné údaje. Postačuje, ak databáza obsahuje aktuálny stav.

Príklad 1.5 - Múzeum



Výlučné vzťahy: entita môže byť len v jednom z navzájom sa vylučujúcich typov vzťahu.

Pre označenie výlučných typov vzťahu existujú rôzne značenia. Napríklad:

Dva druhy "alebo":

- obyčajné (aký je slovenský výraz?)
- anglicky: (inclusive) OR
- možnosť A alebo možnosť B alebo obe možnosti (A aj B)
- príklad: vyhral súťaž v matematike alebo v angličtine alebo obe súťaže
- vylučujúce
- anglicky: exclusive OR = XOR
- možnosť A alebo možnosť B, ale nemôžu nastať obe možnosti
- príklad: kúpil vstupenku v pokladni alebo na internete

Domáca úloha

Na web-stránke predmetu

konzultácie

- Vypracovať domácu úlohu
- Vypracovať test