



**RNDr. Jakub Lokoč, Ph.D.**  
**RNDr. Michal Kopecký, Ph.D.**  
Katedra softwarového inženýrství  
Matematicko-Fyzikální fakulta  
Univerzita Karlova v Praze

# Příklady na cvičení z DBS

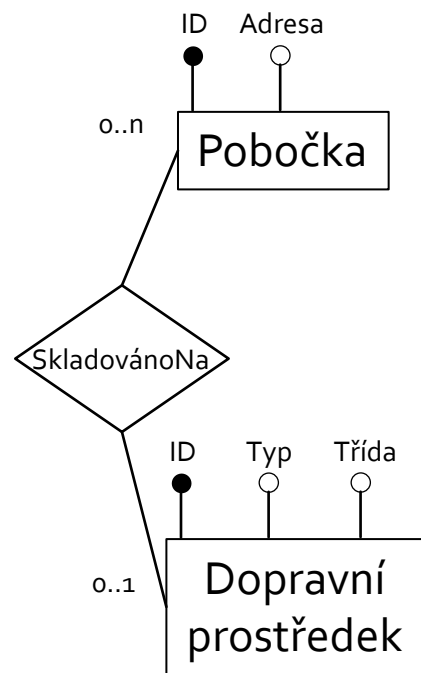
# ER-modelování

Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.

Jedná se o systém jedné půjčovny, budeme tedy evidovat pouze pobočky. V zadání nejsou zmíněné specifické atributy, proto budeme uvažovat pouze běžné atributy jako jsou ID a Adresa.

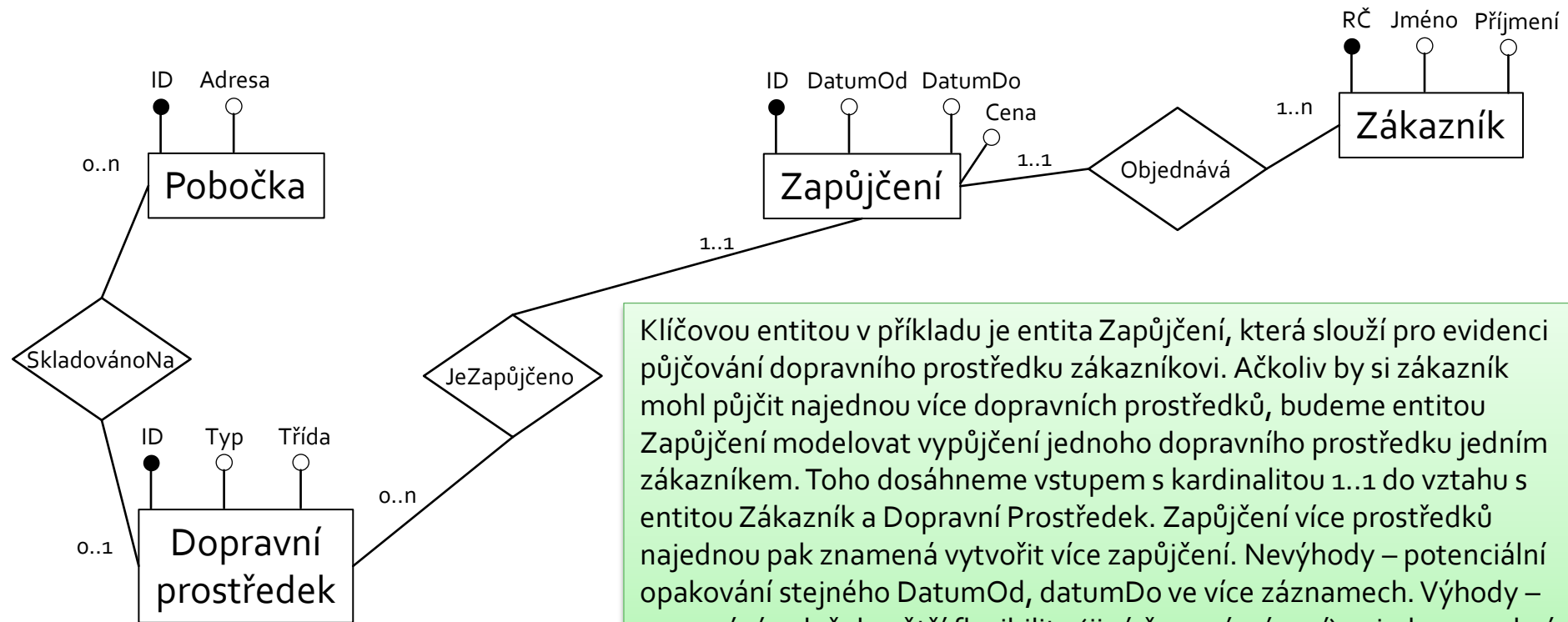
Dále budeme uvažovat dopravní prostředky, auta i kola budeme chápat jako instance obecné entity a odlišíme je atributem typ. V tomto případě atribut typ stačí, protože zadání se zabývá pouze evidencí půjčování a nevyžaduje specifické atributy pro auta a kola. Pokud by byly datové záznamy pro půjčování kol jiné než u aut, tak by bylo vhodnější vytvořit dvě entity. Poznámka – nebyla použita ISA-hierarchie z důvodu snadnějšího následného převodu do Relačního modelu.

Nad rámec zadání ještě evidujeme vztah mezi pobočkou a dopravním prostředkem. Tento vztah říká, na které pobočce je aktuálně kolo skladováno. Nula v dolní hranici intervalu kardinality vztahu u dopravního prostředku znamená, že dopravní prostředek může být půjčený, 1 v horní hranici znamená, že je skladovaný na maximálně jedné pobočce. Na pobočce může být 0..n dopravních prostředků (v zadání není shora omezeno).



# ER-modelování

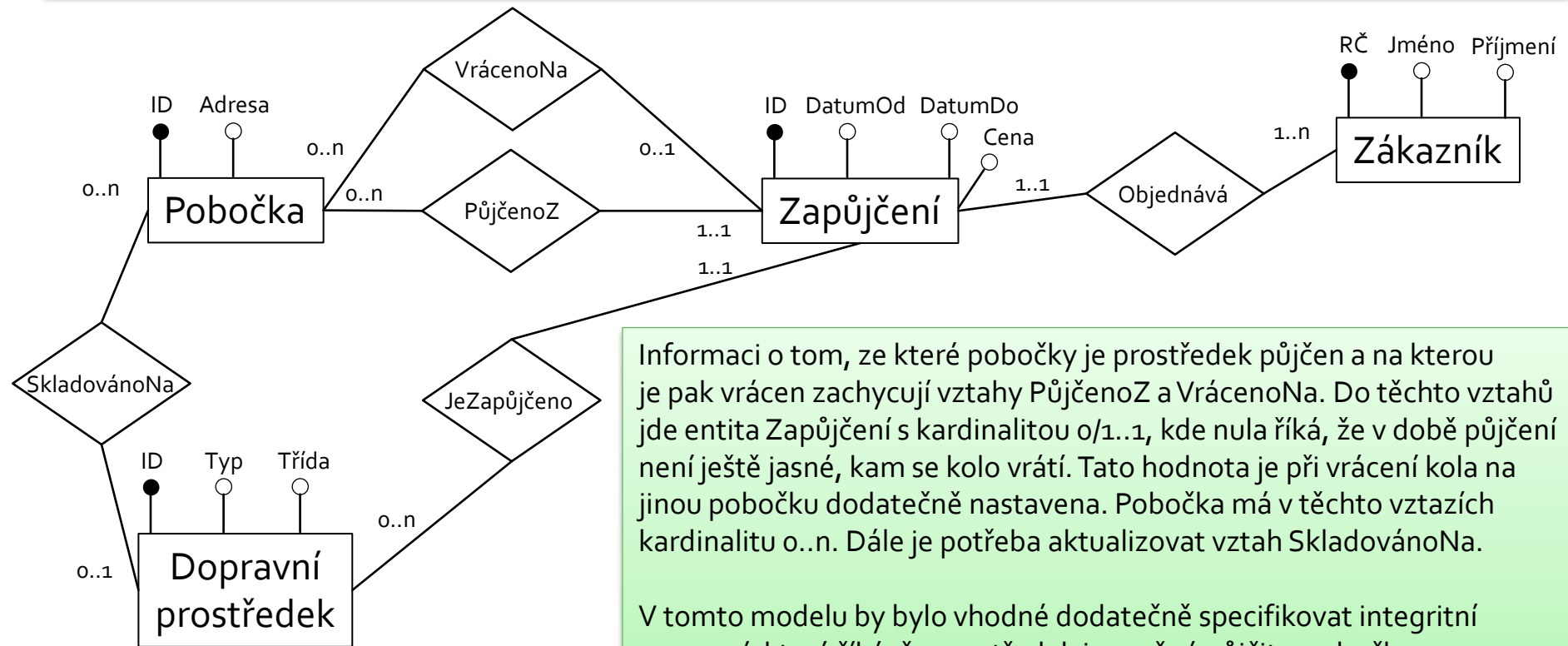
Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.



Klíčovou entitou v příkladu je entita Zapůjčení, která slouží pro evidenci půjčování dopravního prostředku zákazníkovi. Ačkoliv by si zákazník mohl půjčit najednou více dopravních prostředků, budeme entitou Zapůjčení modelovat vypůjčení jednoho dopravního prostředku jedním zákazníkem. Toho dosáhneme vstupem s kardinalitou 1..1 do vztahu s entitou Zákazník a Dopravní Prostředek. Zapůjčení více prostředků najednou pak znamená vytvořit více zapůjčení. Nevýhody – potenciální opakování stejného DatumOd, datumDo ve více záznamech. Výhody – rozepsání položek, větší flexibilita (jiné časy návratu), o jednu vazební tabulku méně (viz. převod do RM).

# ER-modelování

Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.

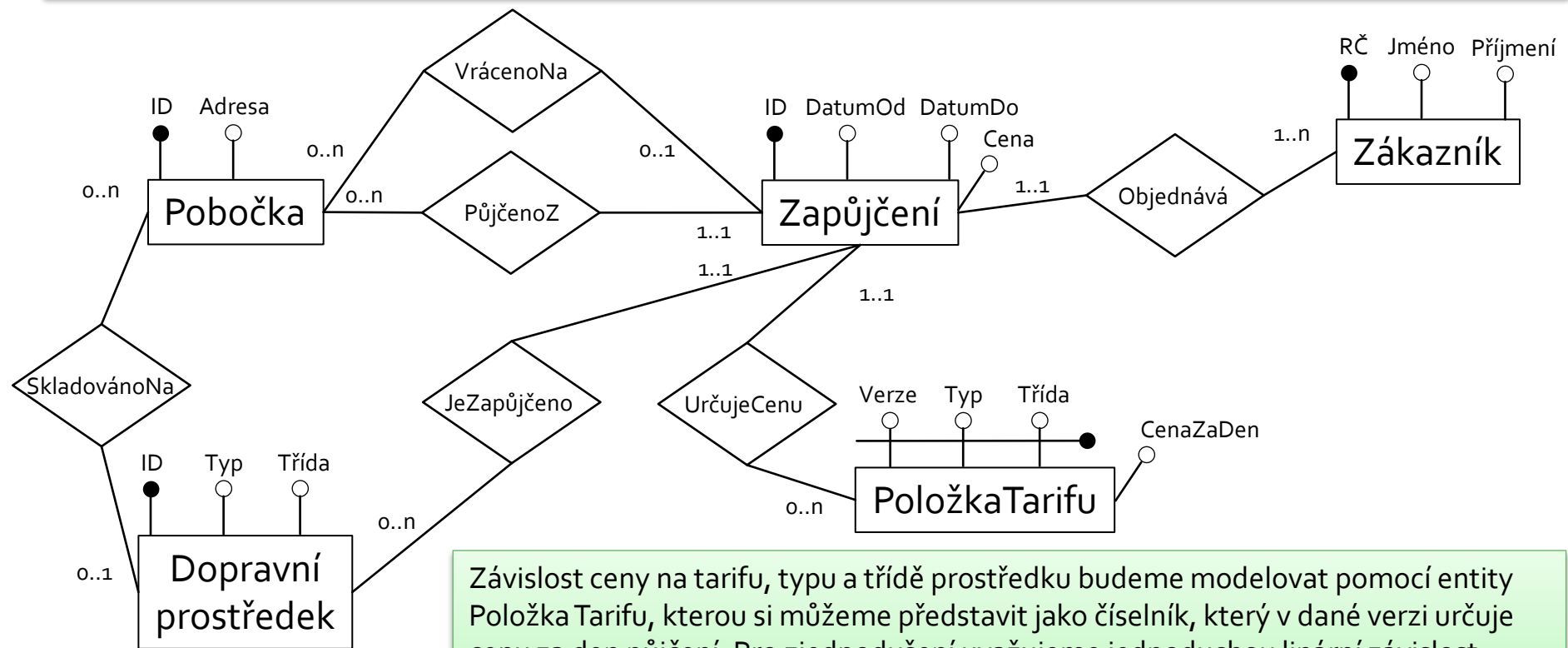


Informaci o tom, ze které pobočky je prostředek půjčen a na kterou je pak vrácen zachycují vztahy PůjčenoZ a VrácenoNa. Do těchto vztahů jde entita Zapůjčení s kardinalitou 0/1..1, kde nula říká, že v době půjčení není ještě jasné, kam se kolo vrátí. Tato hodnota je při vrácení kola na jinou pobočku dodatečně nastavena. Pobočka má v těchto vztazích kardinalitu 0..n. Dále je potřeba aktualizovat vztah SkladovánoNa.

V tomto modelu by bylo vhodné dodatečně specifikovat integritní omezení, které říká, že prostředek je možné půjčit z pobočky pouze tehdy, pokud je na pobočce skladován.

# ER-modelování

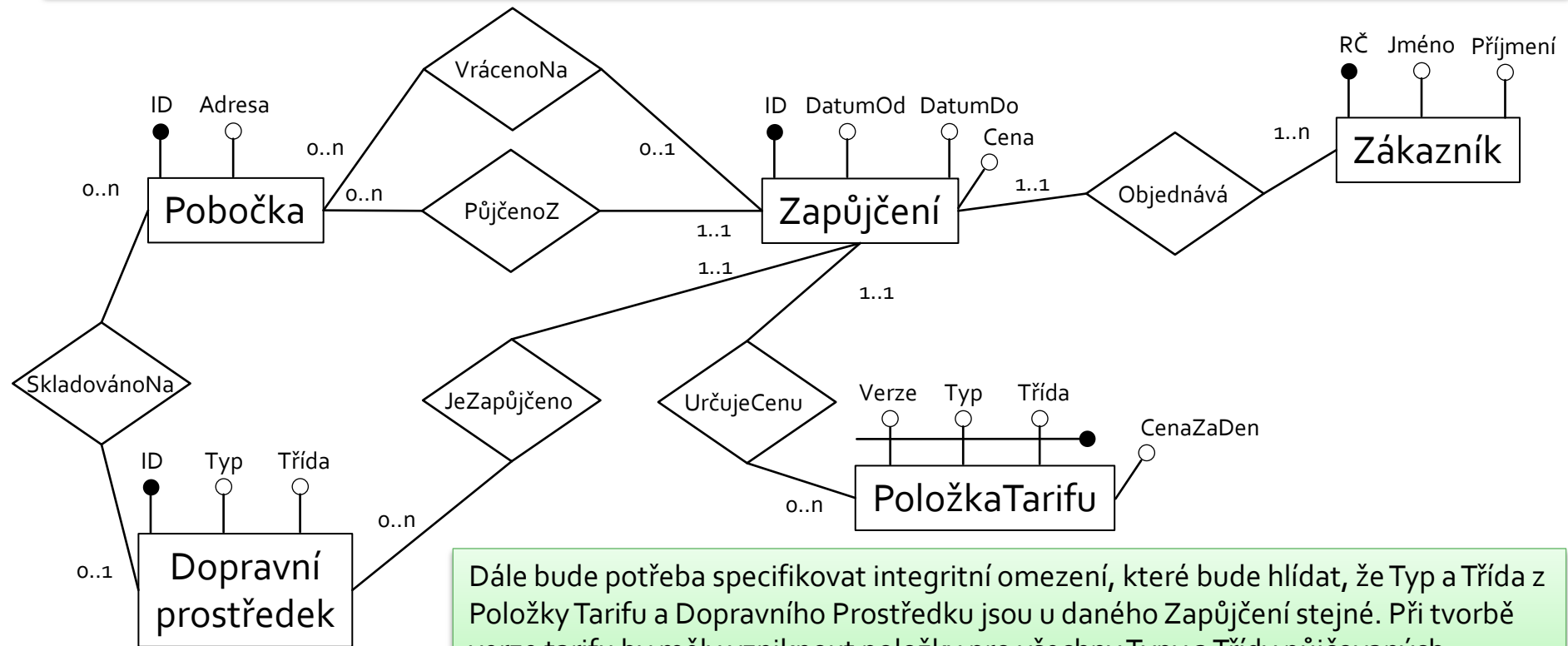
Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.



Závislost ceny na tarifu, typu a třídě prostředku budeme modelovat pomocí entity Položka Tarifu, kterou si můžeme představit jako číselník, který v dané verzi určuje cenu za den půjčení. Pro zjednodušení uvažujeme jednoduchou lineární závislost CenyZaDen na času půjčení. Atribut Cena u Zapůjčení se pak spočítá dynamicky jako počet dnů krát příslušná CenaZaDen.

# ER-modelování

Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.

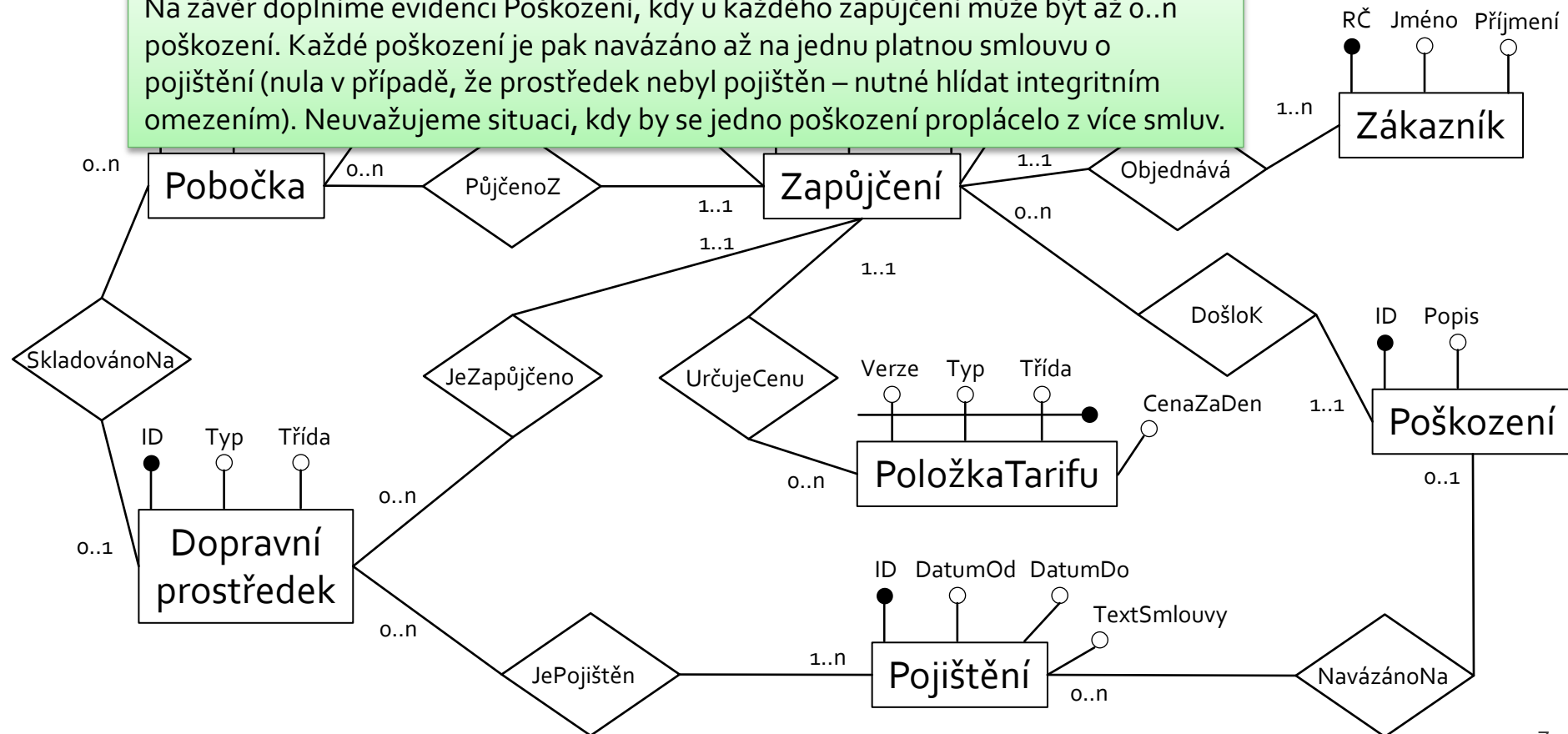


Dále bude potřeba specifikovat integritní omezení, které bude hlídat, že Typ a Třída z Položky Tarifu a Dopravního Prostředku jsou u daného Zapůjčení stejné. Při tvorbě verze tarifu by měly vzniknout položky pro všechny Typy a Třídy půjčovaných prostředků. Jejich existenci by mělo kontrolovat další integritní omezení.

# ER-modelování

Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.

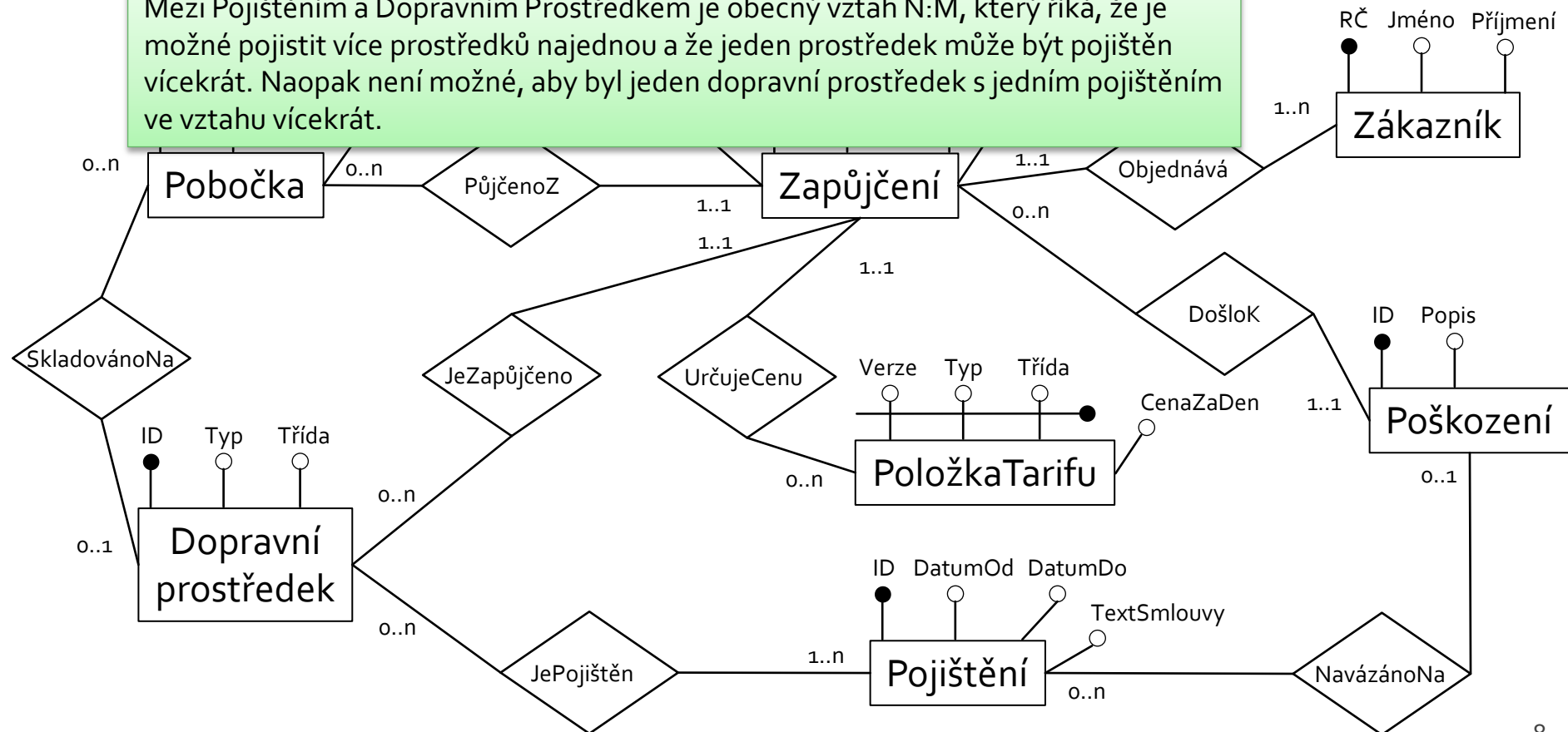
Na závěr doplníme evidenci Poškození, kdy u každého zapůjčení může být až o..n poškození. Každé poškození je pak navázáno až na jednu platnou smlouvu o pojištění (nula v případě, že prostředek nebyl pojištěn – nutné hlídat integritním omezením). Neuvažujeme situaci, kdy by se jedno poškození proplácelo z více smluv.



# ER-modelování

Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.

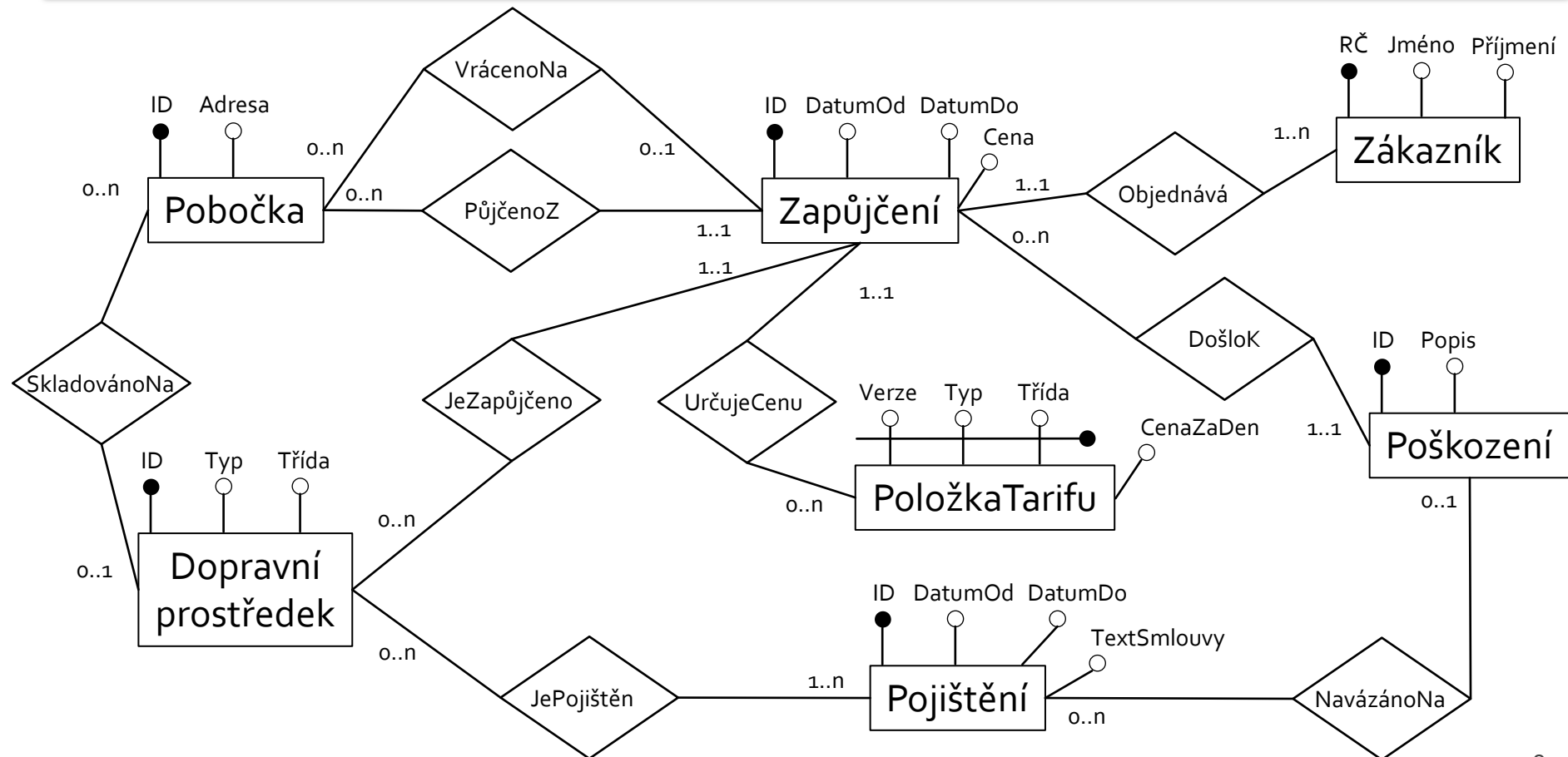
Mezi Pojištěním a Dopravním Prostředkem je obecný vztah N:M, který říká, že je možné pojistit více prostředků najednou a že jeden prostředek může být pojištěn vícekrát. Naopak není možné, aby byl jeden dopravní prostředek s jedním pojištěním ve vztahu vícekrát.





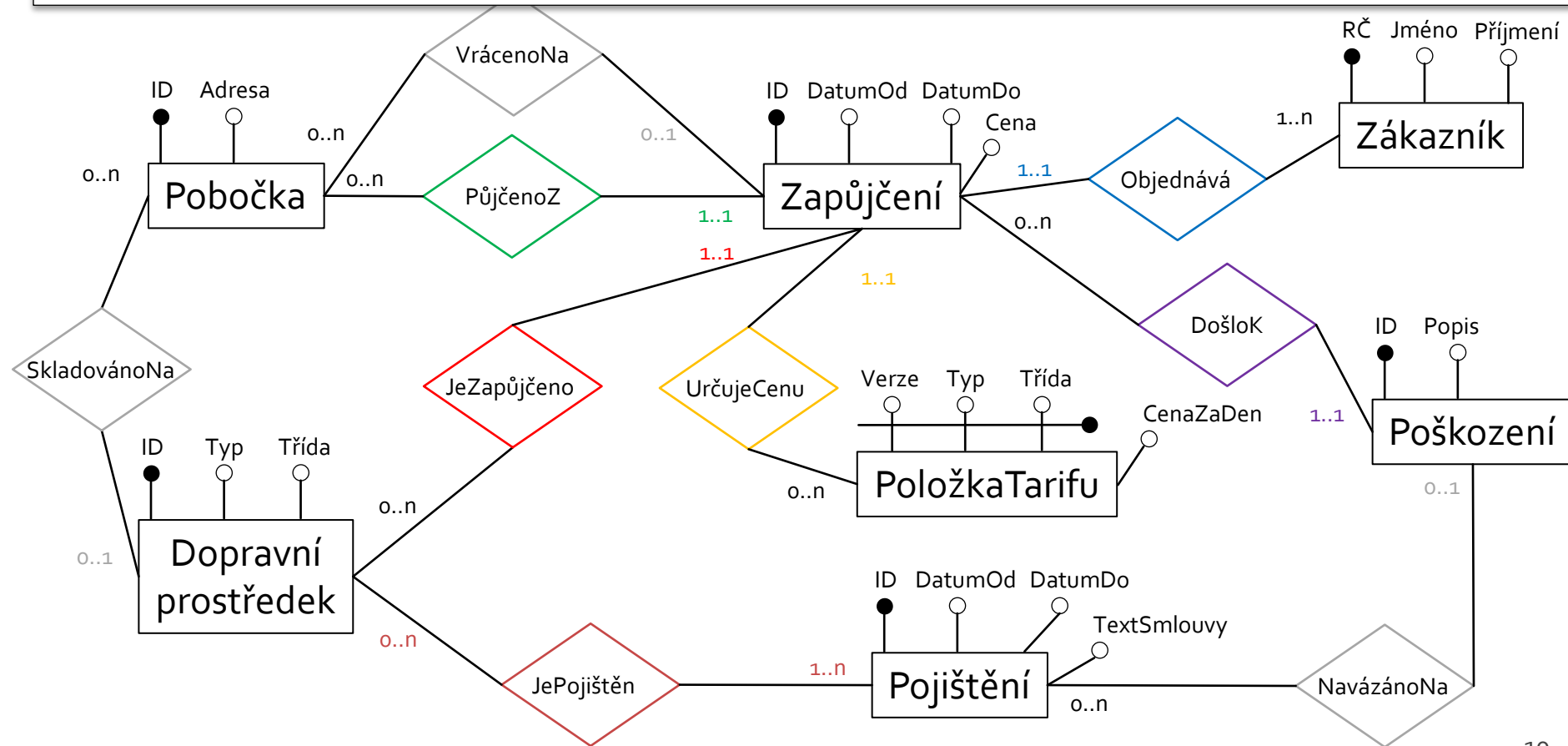
# ER-modelování

Vytvořte model půjčovny dopravních prostředků. Zákazník si může půjčit kolo nebo auto na daný časový interval dle platného tarifu. Cena závisí na verzi tarifu, třídě auta/kola a době půjčení. Dopravní prostředek je možné vrátit na kterékoliv pobočce. Při vrácení se evidují případná poškození, která se navazují na pojistky dopravních prostředků.



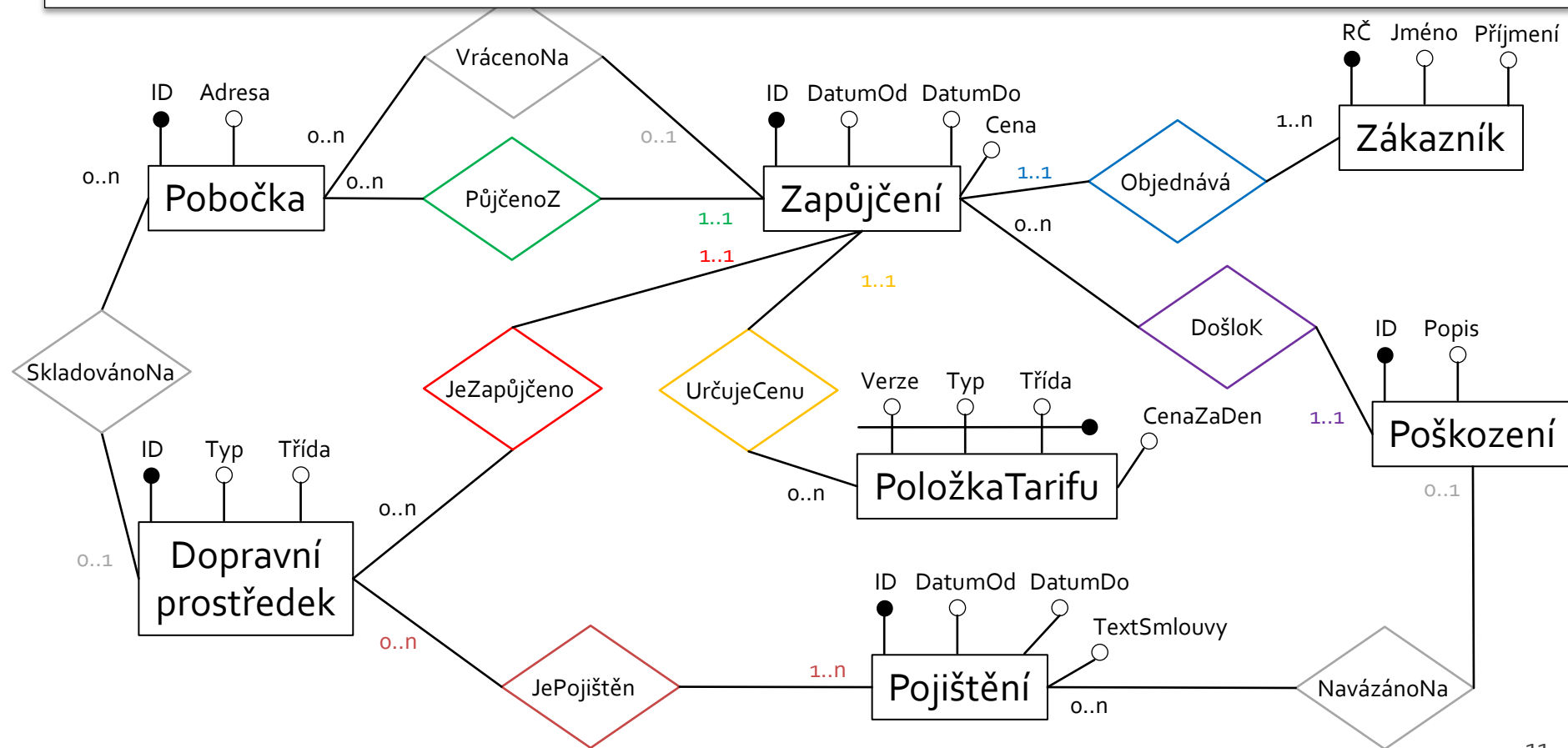
# ER-modelování a převod do RM

**Pobočka**(ID, Adresa), **DopravníProstředek**(ID, Typ, Třída, SkladovánoNaID), **Zákazník**(RČ, Jméno, Příjmení)  
**PoložkaTarifu**(Verze, Typ, Třída, CenaZaDen), **Poškození**(ID, Popis, DošloKID, NavázánoNaID)  
**Zapůjčení**(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, ObjednáváRČ, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
**Pojištění**(ID, DatumOd, DatumDo, TextSml.), **JePojištěn**(DPID, PojID)



# ER-modelování a převod do RM

Pokud je vztah typu  $o..n : o..1$ , tak je také možné vytvořit namísto atributu vazební tabulku. Tato vazební tabulka se vyplácí v případech, kdy ke vztahu dochází pouze velice vzácně. V našem příkladě je výhodnější pro modelování vztahu použít atribut. VrácenoNa – kolo bude nakonec vždy někam vráceno, NavázánoNa – předpokládáme většinou pojištěné prostředky, SkladovánoNa – hodnota se v čase mění, ale je neustále k dispozici u prostředku.



# SQL dotazování

**Pobočka**(ID, Adresa), **DopravníProstředek**(ID, Typ, Třída, SkladovánoNaID), **Zákazník**(RČ, Jméno, Příjmení)  
**PoložkaTarifu**(Verze, Typ, Třída, CenaZaDen), **Poškození**(ID, Popis, DošloKID, NavázánoNaID)  
**Zapůjčení**(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, **ObjednáváRČ**, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
**Pojištění**(ID, DatumOd, DatumDo, TextSml.), **JePojištěn**(DPID, PojID)

Jména zákazníků, kteří si půjčili dopravní prostředek od 1.1.2016 do 3.1.2016 ??

Pomocí vnitřního spojení

```
SELECT Jméno FROM Zákazník INNER JOIN Zapůjčení ON (RČ = ObjednáváRČ)
WHERE DatumOd = '1.1.2016' AND datumDo = '3.1.2016'
```

Pomocí vnořeného selektu

```
SELECT Jméno FROM Zákazník WHERE RČ IN (SELECT ObjednáváRČ FROM
    Zapůjčení WHERE DatumOd = '1.1.2016' AND datumDo = '3.1.2016')
```

# SQL dotazování

**Pobočka**(ID, Adresa), **DopravníProstředek**(ID, Typ, Třída, SkladovánoNaID), **Zákazník**(RČ, Jméno, Příjmení)  
**PoložkaTarifu**(Verze, Typ, Třída, CenaZaDen), **Poškození**(ID, Popis, DošloKID, NavázánoNaID)  
**Zapůjčení**(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, **ObjednáváRČ**, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
**Pojištění**(ID, DatumOd, DatumDo, TextSml.), **JePojištěn**(DPID, PojID)

Jména zákazníků, kteří si nepůjčili dopravní prostředek ??

Pomocí levého vnějšího spojení

```
SELECT Jméno FROM Zákazník LEFT OUTER JOIN Zapůjčení ON (RČ = ObjednáváRČ)
WHERE ID IS NULL
```

Pomocí vnořeného selektu

```
SELECT Jméno FROM Zákazník
WHERE RČ NOT IN (SELECT ObjednáváRČ FROM Zapůjčení)
```

# SQL dotazování

**Pobočka**(ID, Adresa), **DopravníProstředek**(ID, Typ, Třída, SkladovánoNaID), **Zákazník**(RČ, Jméno, Příjmení)  
**PoložkaTarifu**(Verze, Typ, Třída, CenaZaDen), **Poškození**(ID, Popis, DošloKID, NavázánoNaID)  
**Zapůjčení**(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, **ObjednáváRČ**, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
**Pojištění**(ID, DatumOd, DatumDo, TextSml.), **JePojištěn**(DPID, PojID)

Jména zákazníků, kteří si nepůjčili dopravní prostředek od 1.1.2016 do 3.1.2016 ??

Pomocí levého vnějšího spojení

```
SELECT Jméno FROM Zákazník LEFT OUTER JOIN Zapůjčení ON (RČ = ObjednáváRČ)
WHERE DatumOd != '1.1.2016' OR datumDo != '3.1.2016' OR ID IS NULL
```

Pomocí vnořeného selektu

```
SELECT Jméno FROM Zákazník WHERE RČ NOT IN (SELECT ObjednáváRČ FROM
Zapůjčení WHERE DatumOd = '1.1.2016' AND datumDo = '3.1.2016')
```

# SQL dotazování

**Pobočka**(ID, Adresa), **DopravníProstředek**(ID, Typ, Třída, SkladovánoNaID), **Zákazník**(RČ, Jméno, Příjmení)  
**PoložkaTarifu**(Verze, Typ, Třída, CenaZaDen), **Poškození**(ID, Popis, DošloKID, NavázánoNaID)  
**Zapůjčení**(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, ObjednáváRČ, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
**Pojištění**(ID, DatumOd, DatumDo, TextSml.), **JePojištěn**(DPID, PojID)

Data zapůjčení a skutečná cena za den ??

```
SELECT *, Cena / (1 + DAY(DatumDo) – DAY(DatumOd)) AS CenaZaDen FROM Zapůjčení
```

Zapůjčení, kde skutečná cena za den neodpovídá příslušné položce tarifu ??

```
SELECT *, Cena / (1 + DAY(DatumDo) – DAY(DatumOd)) AS CenaZaDen FROM Zapůjčení Z
WHERE CenaZaDen NOT IN
    (SELECT CenaZaDen FROM PoložkaTarifu PT
     WHERE Z.Verze = PT.Verze AND Z.Typ = PT.Typ AND Z.Třída = PT.Třída)
```

# SQL dotazování

Pobočka(ID, Adresa), DopravníProstředek(ID, Typ, Třída, SkladovánoNaID), Zákazník(RČ, Jméno, Příjmení)  
PoložkaTarifu(Verze, Typ, Třída, CenaZaDen), Poškození(ID, Popis, DošloKID, NavázánoNaID)  
Zapůjčení(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, ObjednáváRČ, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
Pojištění(ID, DatumOd, DatumDo, TextSml.), JePojištěn(DPID, PojID)

Nevyužitý tarif ??

```
SELECT * FROM PoložkaTarifu PT WHERE NOT EXISTS  
  (SELECT ID FROM Zapůjčení Z  
   WHERE Z.Verze = PT.Verze AND Z.Typ = PT.Typ AND Z.Třída = PT.Třída)
```

Kolik je verzí tarifů pro daný typ a třídu ?? Jaká je průměrná cena za den pro daný typ a třídu ??

```
SELECT Typ, Třída, Count(*) FROM PoložkaTarifu GROUP BY Typ, Třída
```

```
SELECT Typ, Třída, Avg(CenaZaDen) FROM PoložkaTarifu GROUP BY Typ, Třída
```



# SQL dotazování

Pobočka(ID, Adresa), DopravníProstředek(ID, Typ, Třída, SkladovánoNaID), Zákazník(RČ, Jméno, Příjmení)  
PoložkaTarifu(Verze, Typ, Třída, CenaZaDen), Poškození(ID, Popis, DošloKID, NavázánoNaID)  
Zapůjčení(ID, DatumOd, DatumDo, Cena, JeZapůjčenoID, ObjednáváRČ, PůjčenoZID, VracenoNaID, Verze, Typ, Třída)  
Pojištění(ID, DatumOd, DatumDo, TextSml.), JePojištěn(DPID, PojID)

Které prostředky byly vráceny na stejnou pobočku, ze které byly půjčeny ??

```
SELECT * FROM DopravníProstředek WHERE ID IN (SELECT JeZapůjčenoID FROM  
Zapůjčení WHERE PůjčenoZID = VracenoNaID)
```

Adresa půjčení a adresa vrácení všech zapůjčených prostředků ??

```
SELECT P1.Adresa AS AdresaPujceni, P2.Adresa AS AdresaVraceni FROM Pobočka P1  
INNER JOIN Zapůjčení Z ON (P1.ID = Z. PůjčenoZID)  
INNER JOIN Pobočka P2 ON (P2.ID = Z. VracenoNaID)
```

# SQL dotazování

**Pobočka**(ID, Adresa), **DopravníProstředek**(ID, Typ, Třída, SkladovánoNaID), **Zákazník**(RČ, Jméno, Příjmení)  
**PoložkaTarifu**(Verze, Typ, Třída, CenaZaDen), **Poškození**(ID, Popis, DošloKID, NavázánoNaID)  
**Zapůjčení**(ID, DatumOd, DatumDo, Cena, **JeZapůjčenoID**, ObjednáváRČ, **PůjčenoZID**, **VrácenoNaID**, Verze, Typ, Třída)  
**Pojištění**(ID, DatumOd, DatumDo, TextSml.), **JePojištěn**(DPID, PojID)

Položky tarifu s nejvyšší cenou za den ??

```
SELECT * FROM PoložkaTarifu WHERE CenaZaDen =  
      (SELECT Max(CenaZaDen) FROM PoložkaTarifu )
```

ID nejčastěji zapůjčených dopravních prostředků ??

```
SELECT JeZapůjčenoID FROM Zapůjčení GROUP BY JeZapůjčenoID  
HAVING Count(*) >= ALL (SELECT Count(*) FROM Zapůjčení GROUP BY JeZapůjčenoID)
```