

E-R model

Jiří Zaccpal

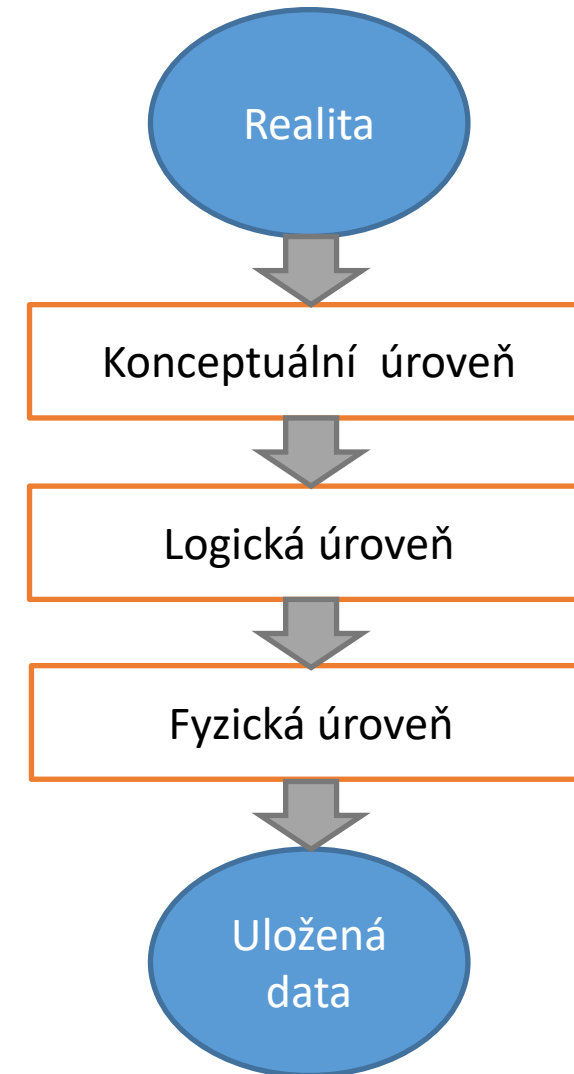


KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/DATAB Databáze

Abstrakce pohledu na data

- **Konceptuální** úroveň – zabývá se modelováním reality (ER model)
- **Logická** úroveň – vztahuje se ke konkrétnímu datovému modelu (relační model)
- **Fyzická** úroveň – řeší fyzické uložení dat



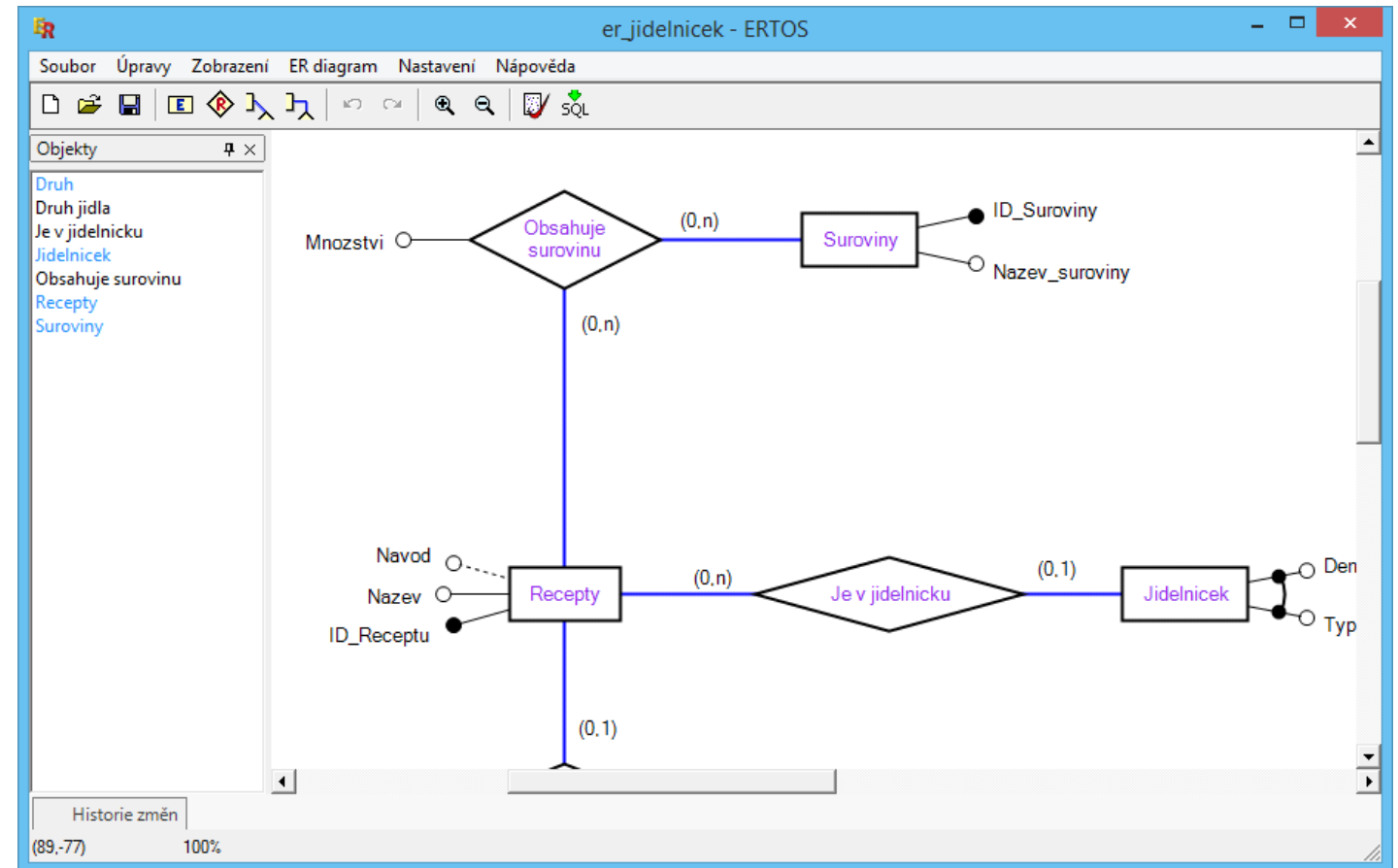
Principy konceptuálních modelů



- oddělení konceptuální a interní úrovně
- orientace na **objekty**, **entity** ne na záznamy a soubory
- bohatší koncept, v relačním modelu jsou relace využívány na „všechno“, reprezentují entity, vícehodnotové atributy, asociace, agregace, dědičnost, ...
- možnost využít úrovně abstrakce v komplexních objektech k zakrytí detailů, možnost modelovat přímo aplikační objekty.
- funkcionální podstata vztahů (atribut nebo funkce je jediným konstruktem)
- ISA hierarchie (práce s nadtypy a podtypy)
- hierarchický mechanismus (objekty lze konstruovat z jiných objektů, formou agregace, seskupováním do množin, tříd)

Program Ertos

- Spuštění
- Práce s programem



Vytvoření ER modelu

- Vytvořit databázi pro knihovnu. Databáze by měla splňovat tyto požadavky:
 - Evidovat knihy
 - Evidovat čtenáře
 - Evidovat výpůjčky

Entitní typy (množiny)

Entitní typy (množiny)

- Kniha
- Autor
- Vztah
 - JeAutorem

- **Entity** odpovídají objektům reálného světa (osoba, věc, ...), popsány pomocí hodnot svých vlastností.
- Entita musí být rozlišitelná od ostatních entit a existovat nezávisle na nich.
- **Silný entitní typ** – Entitní typ existenčně nezávislý na jiném entitním typu.



Silný entitní typ

- **Slabý entitní typ** – Někdy nejsou dvě instance jednoho entitního typu rozlišitelné pomocí svých atributů, jsou rozlišitelné až pomocí toho, že jsou povinně v identifikačním vztahu k další entitě jiného typu (silné, regulární).



Slabý entitní typ

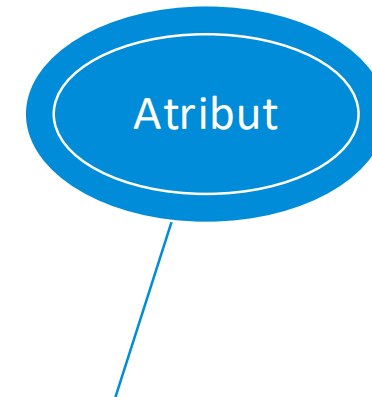
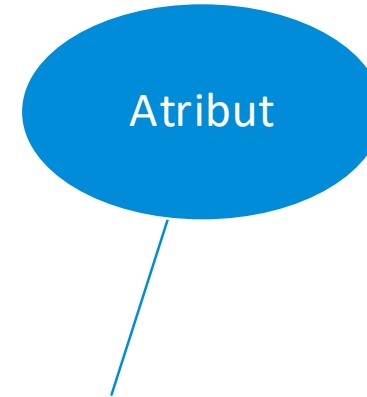


Kniha

Výtisk

Atributy

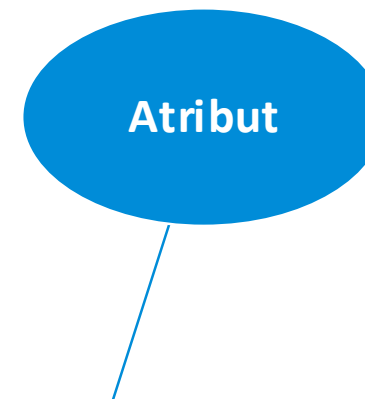
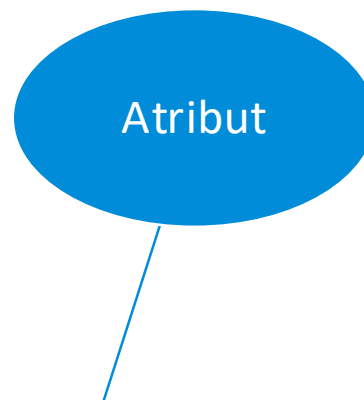
- **atribut** je funkce, která přiřazuje entitám nebo vztahům hodnotu vlastnosti, doménu atributu tvoří přípustné hodnoty atributu
 - jednoduché (jedna atomická hodnota)
- skupinové (strukturované, kompozitní, složené)



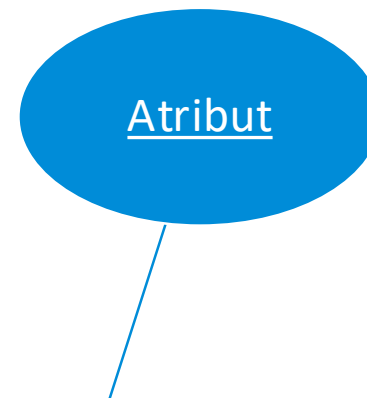
Atributy

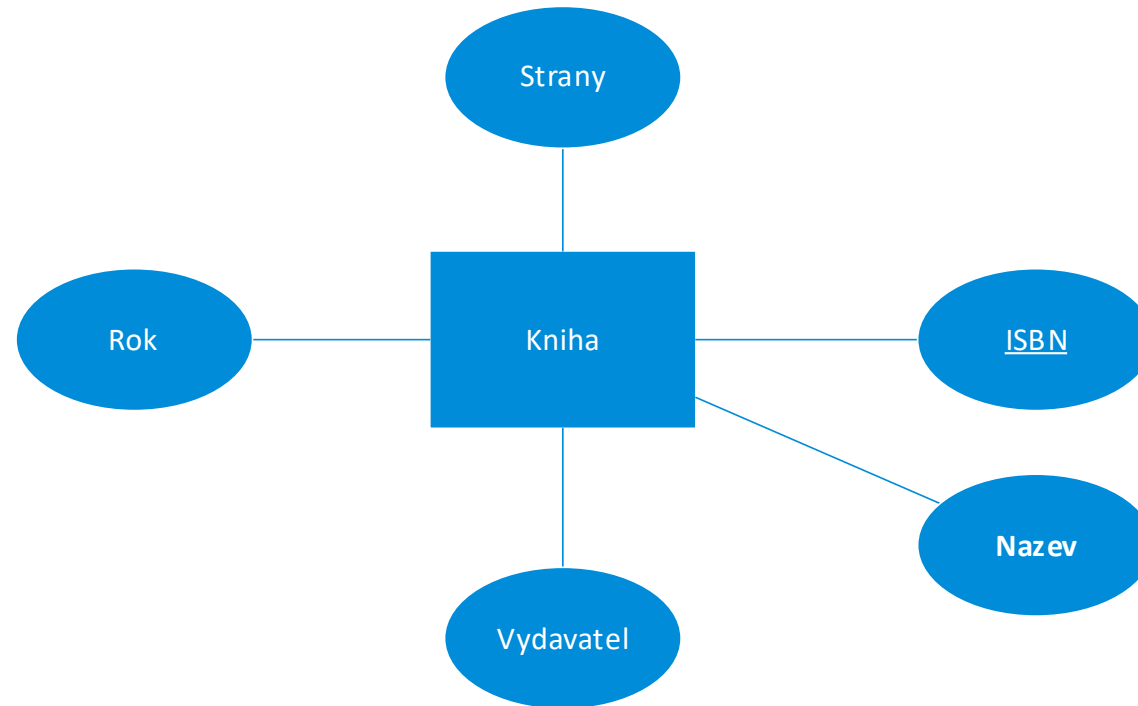


- nepovinný - název není tučný
- povinný – název tučným písmem



- primární klíč





Entitní množina Kniha

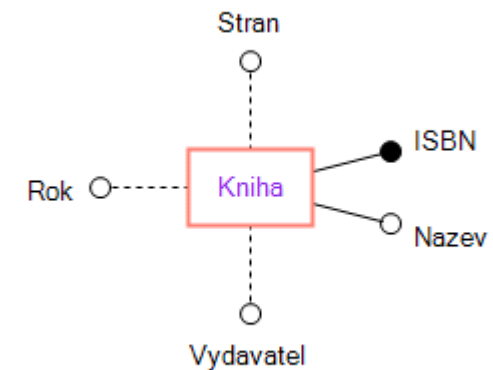


Entita

Název: Název pro databázi:

Atributy							
	PK	Název	Datový typ	Unique	Not NULL	Defaultní hodnota	Několikanásobný
▶	<input checked="" type="checkbox"/>	ISBN	Varchar(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	Nazev	Varchar(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	Vydavatel	Varchar(20)	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	Rok	Integer	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	Stran	Integer	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

Entita je identifikována těmito vztahy:



Vztahy

- **vztah**

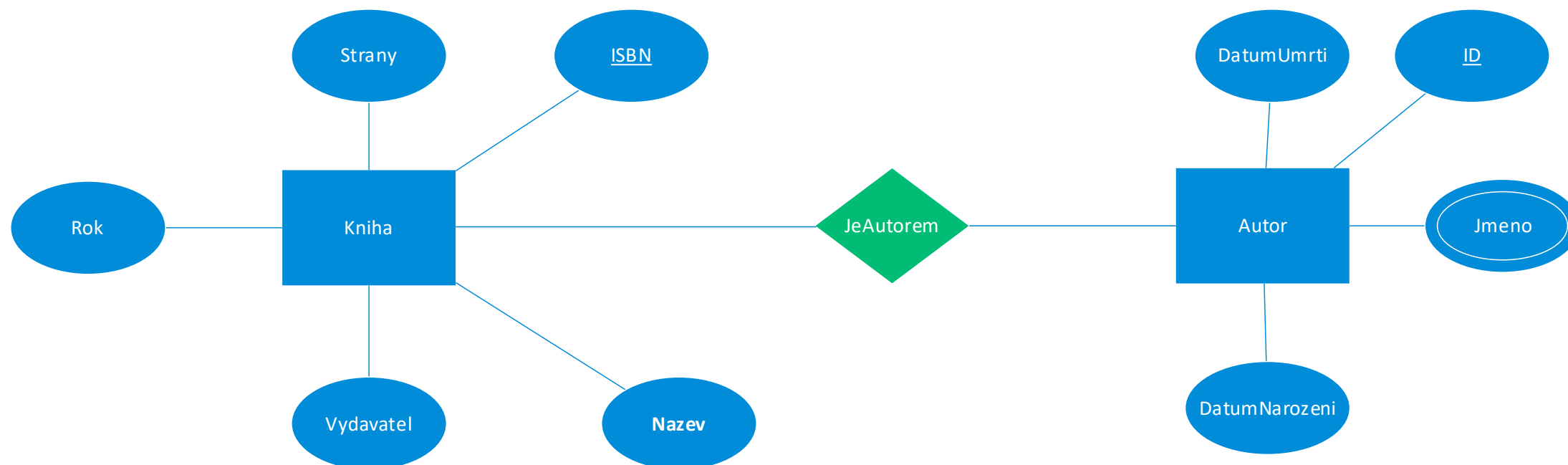
- vazba mezi dvěma nebo více entitami.
- množina smysluplných asociací mezi entitními typy v informačním systému

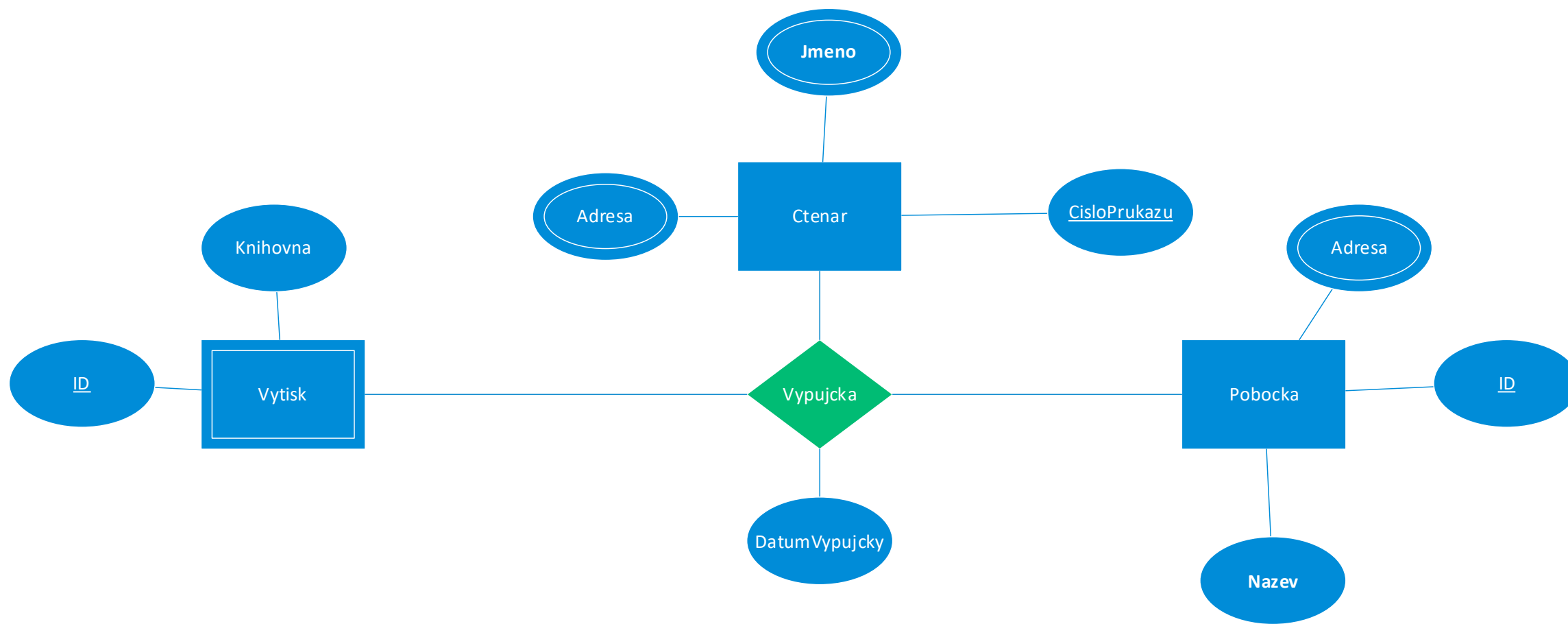
- vztahová množina R je určena:

$$R \subset E_1 \times \cdots \times E_n = \{(e_1, \dots, e_n) | e_1 \in E_1 \dots e_n \in E_n\},$$

- e_i je entita,
- E_i je entitní typ,
- n je stupeň vztahu.

Příklad – Vztah kniha autor





Integritní omezení

Integritní omezení

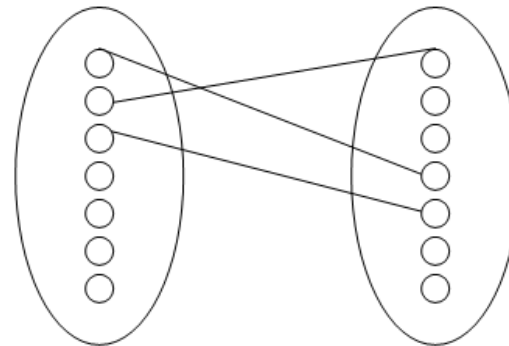


- jsou logická omezení na typy a hodnoty atributů, entit a vazeb taková, aby konceptuální schéma co nejlépe a nerozporně odpovídalo zobrazované realitě.
- **kardinalita vztahu** – udává kolikrát se entita může účastnit vztahu (1,...,N)
- **parcialita vztahu** – udává, zda se entita musí účastnit vztahu

Kardinalita vztahu

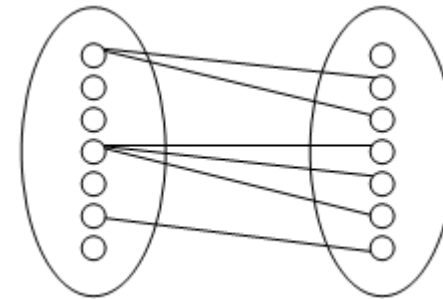
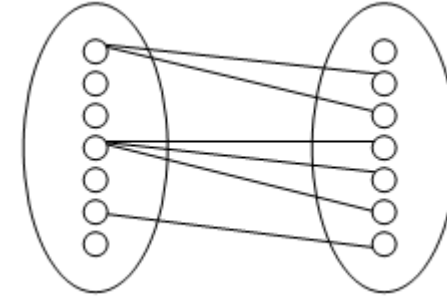


- Binární vztah typů entit E_1 a E_2 může mít jeden ze tří poměrů:
 - **1:1** - jedné $e_1 \in E_1$ odpovídá ve vztahu nejvýše jedna $e_2 \in E_2$ a naopak, jedné $e_2 \in E_2$ odpovídá nejvýše jedna $e_1 \in E_1$



Kardinalita vztahu

- **1:N** - jedné $e_1 \in E_1$ odpovídá ve vztahu obecně několik $e_2 \in E_2$, ale jedna $e_2 \in E_2$ má vztah pouze k jedné $e_1 \in E_1$
- **M:N** - jedné $e_1 \in E_1$ odpovídá ve vztahu obecně několik $e_2 \in E_2$, a naopak jedna $e_2 \in E_2$ má vztah k několika entitám $e_1 \in E_1$



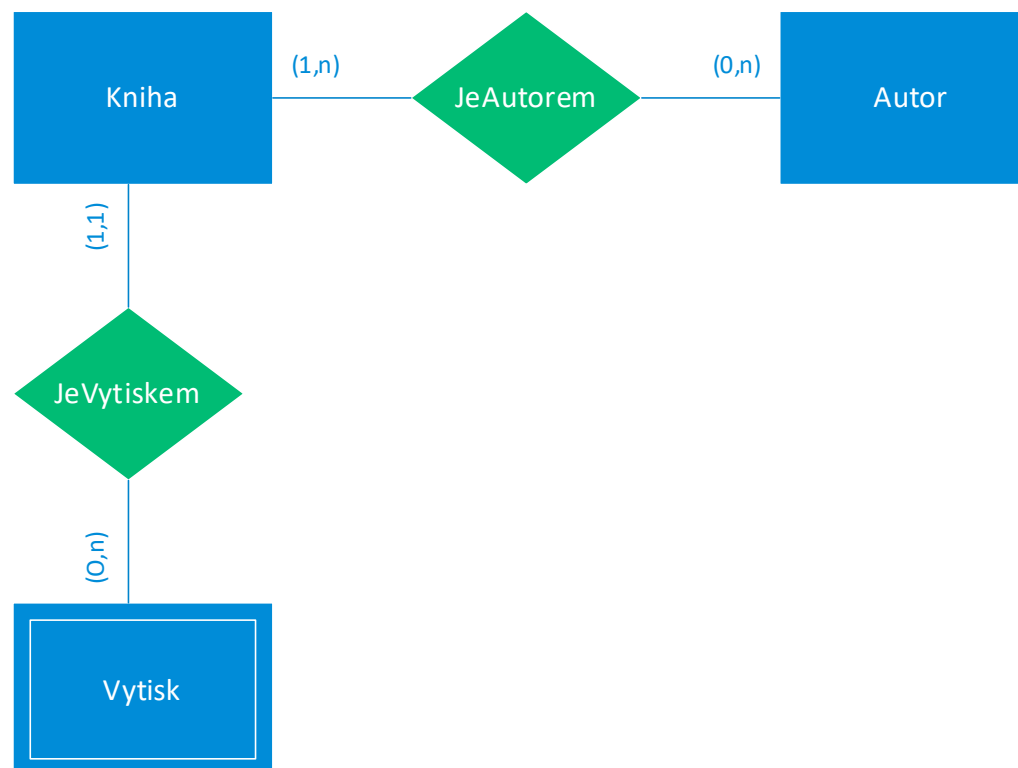
Parcialita (členství) ve vztahu



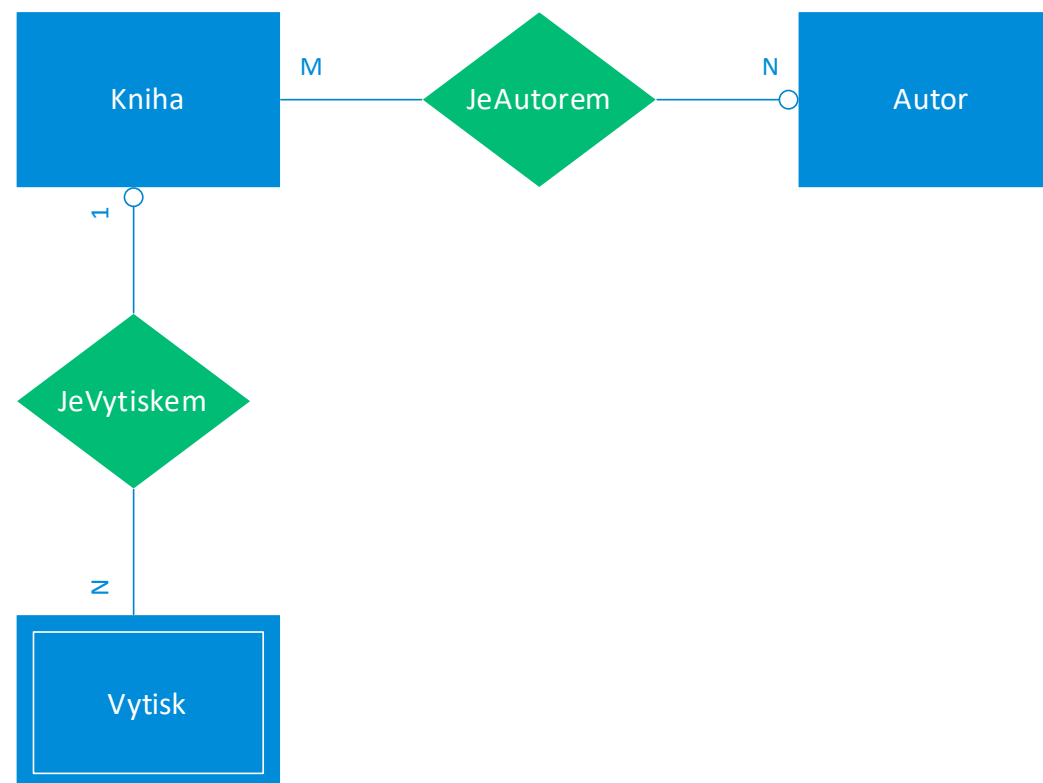
- vyjadřuje možnost samostatné existence entity (nepovinné, fakultativní členství)
- případ, kdy jsou entity existenčně svázány ve vztahu (povinné, obligatorní členství)
- přitom může mít jedna entita povinné členství, druhá nepovinné.
- alternativně zaznamenáváme graficky v ER diagramu značkou (např. plným kroužkem na straně příslušného entitního typu, nepovinnost prázdným kroužkem),
- nebo společně s kardinalitou

(E1:(min,max),E2:(min,max)), nebo
(min,max):(min,max)

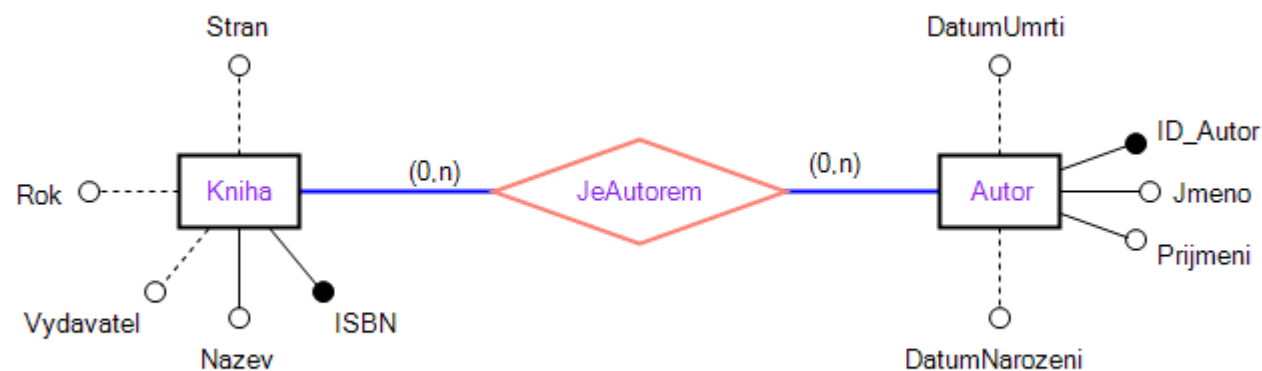
Příklad – Vztah mezi autorem, knihou a výtiskem



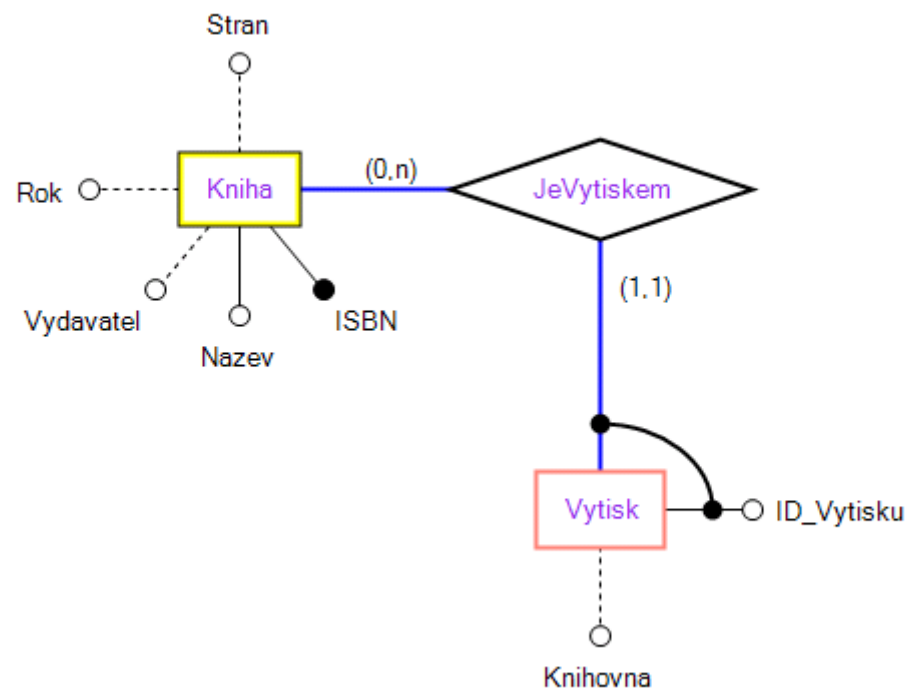
Příklad – Vztah mezi autorem, knihou a výtiskem



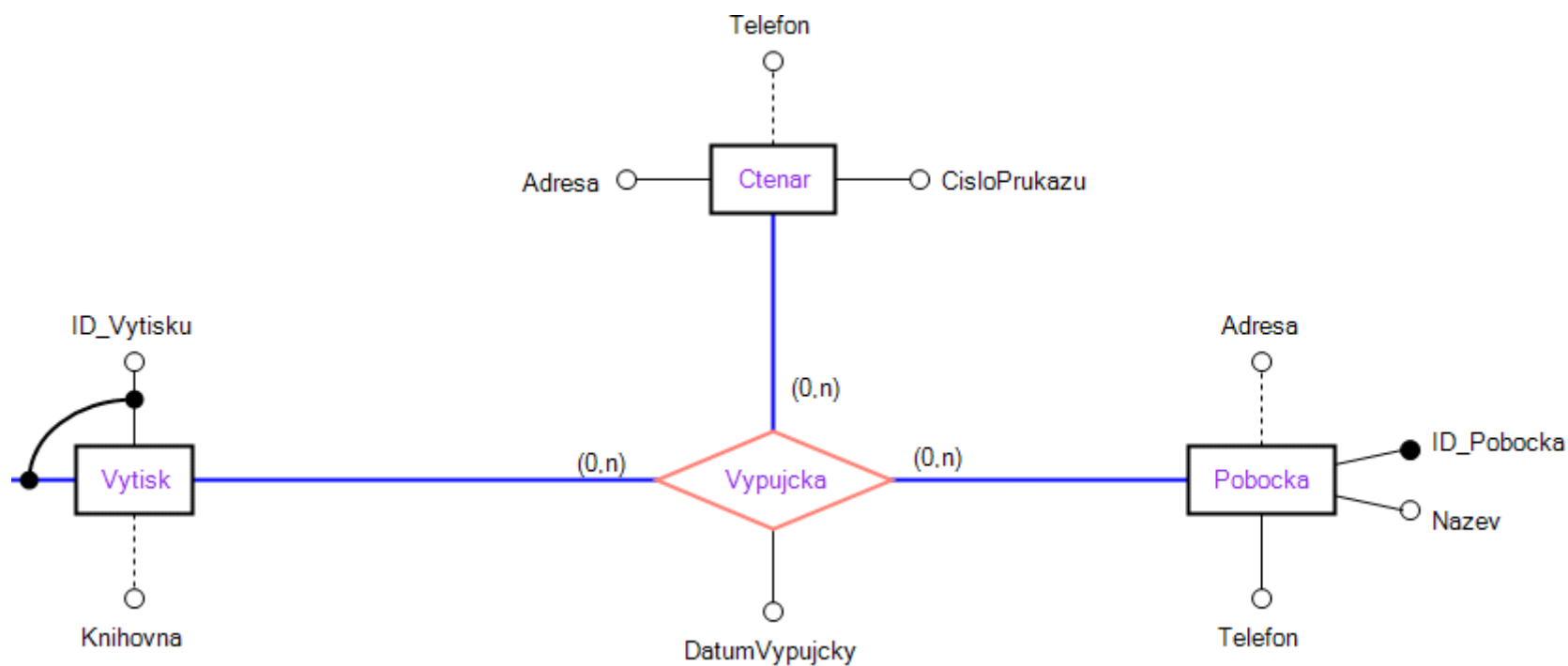
Vztah JeAutorem



Slabá entitní množina Vytisk



Ternární vztah



Postup vytvoření diagramu

Postup sestavení diagramu případů užití



- Jako příklad vezmeme **ER diagram pro vysokou školu**.
- Diagram by měl obsahovat tyto entity:
 - Student.
 - Kurz.
 - Rozvrh.
 - Vyucujici.

Student

Kurz

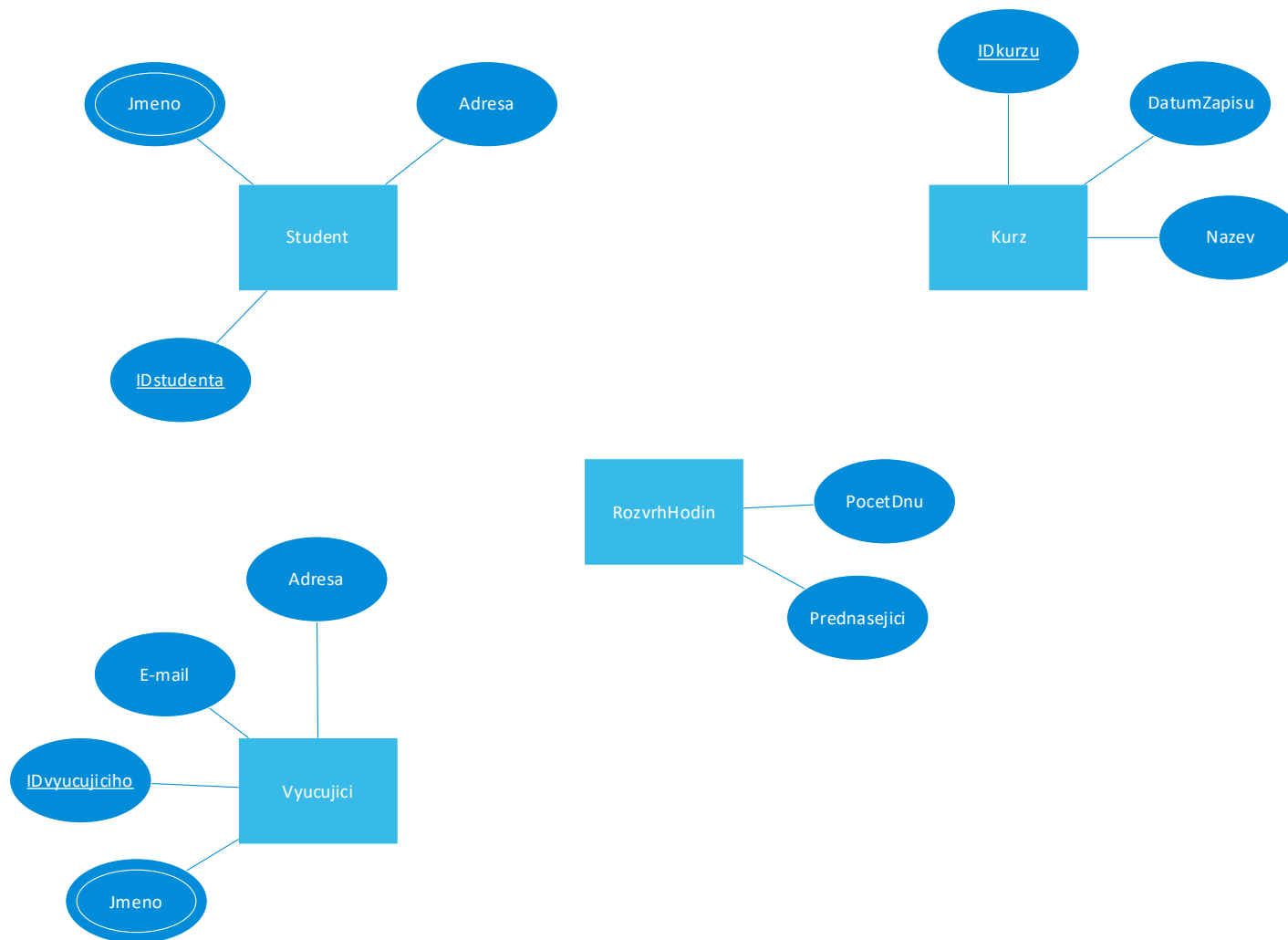
RozvrhHodin

Vyucujici

Postup sestavení diagramu případů užití



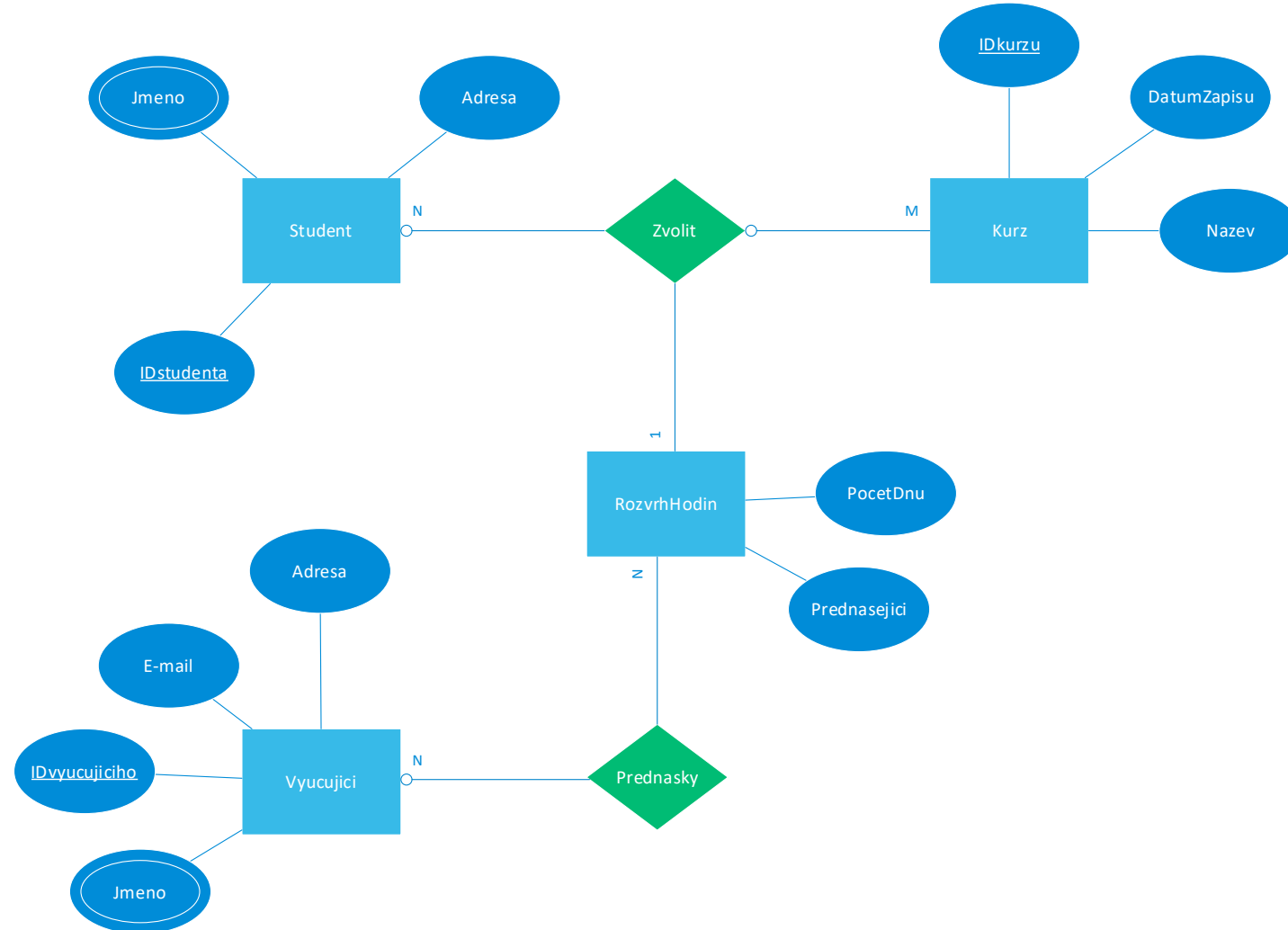
- Dalším krokem je přidání atributů k jednotlivým entitním typům.



Postup sestavení diagramu případů užití



- Posledním krokem je vytvoření vztahů mezi entitními typy.

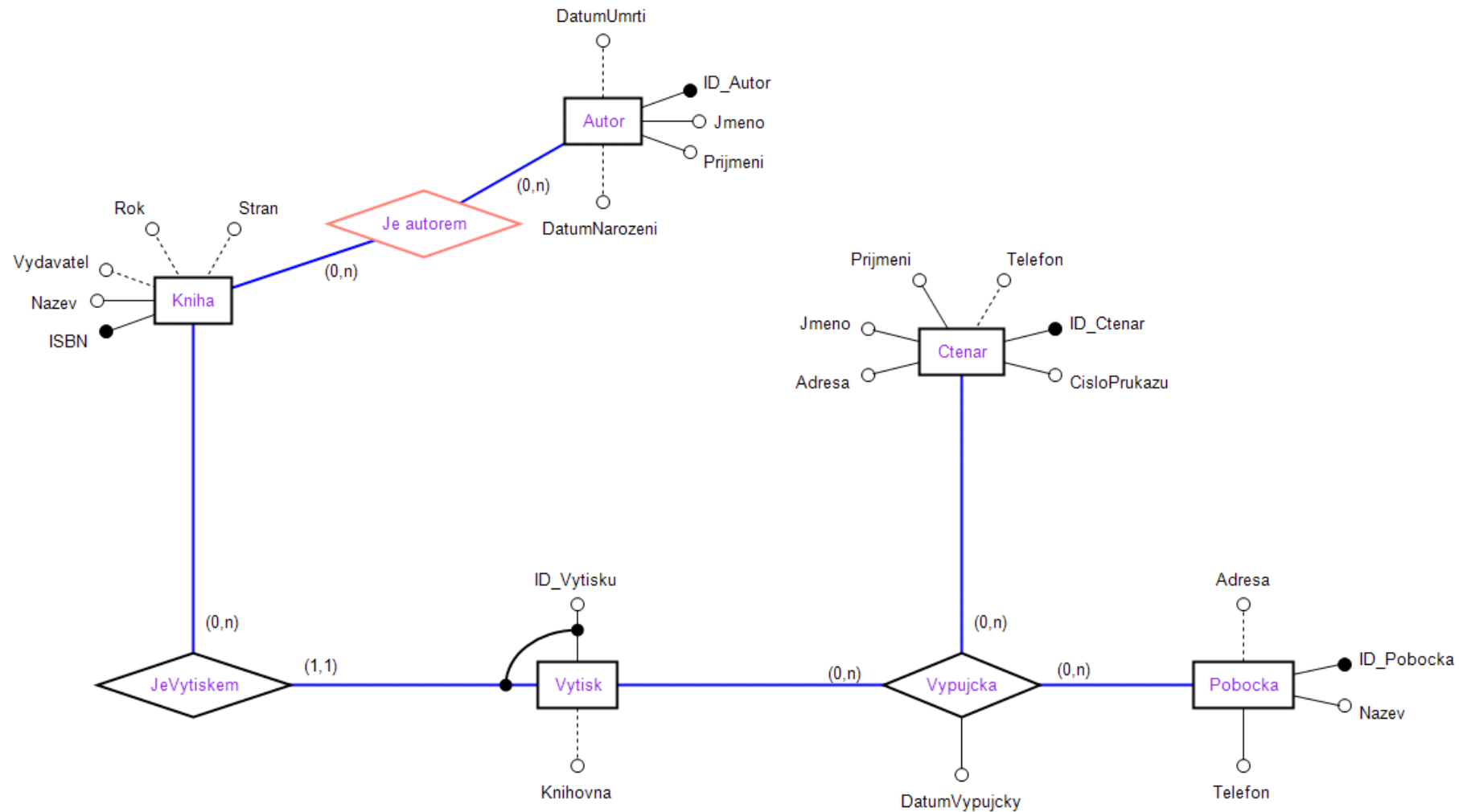


Převod ER modelu do relačního modelu

Postup převod

- Entitní typy (množiny) převedeme jako tabulky s atributy.
- Vztahy:
 - 1:1 – oba entitní typy sloučíme do jedné tabulky.
 - 1:N – do tabulky na straně N umístíme primární klíč z tabulky na straně 1.
 - M:N – vytvoříme novou tabulku, do které vložíme primární klíče z obou tabulek, které jsou ve vztahu.

Příklad



Příklad

```
CREATE TABLE Autor (  
  DatumNarozeni      Integer,  
  DatumUmrti         Integer,  
  Jmeno              Char      NOT NULL,  
  Prijmeni           Char      NOT NULL,  
  ID_Autor            Integer    NOT NULL,  
  CONSTRAINT pk_Autor PRIMARY KEY (ID_Autor));
```

```
CREATE TABLE Kniha (  
  ISBN               Varchar(10) NOT NULL,  
  Nazev              Varchar(20)  NOT NULL,  
  Vydavatel          Varchar(20),  
  Rok                Integer,  
  Stran              Integer,  
  CONSTRAINT pk_Kniha PRIMARY KEY (ISBN));
```

...