DATABÁZOVÝ DESIGN

Podpora výuky databázových systémů na SOŠ, založené na technologiích společnosti ORACLE.

Publikace vznikla v rámci projektu CZ.1.07/1.1.07/02.007, Podpora výuky databázových systémů na středních odborných školách, založené na technologiích společnosti ORACLE.

Studijní kapitoly jsou synchronizovány s mezinárodním vzdělávacím programem Oracle Academy. Více informací na academy.oracle.com nebo na portálu ucimedatabaze.cz.

Manager projektu: Mgr. Richard Štěpán
Překlad: Oracle Czech, Bc. Tomáš Romanovský, Mgr. Dana Mikesková, Mgr. Markéta Kytková
Metodik: Bc. Tomáš Romanovský
Jazyková korektura: Mgr. Pavlína Chovancová
Sazba: Bc. Tomáš Romanovský
Obálka: Bc. Tomáš Romanovský
Tisk: Reprografické studio LWR GRAPHIC

Zvláštní poděkování patří společnosti Oracle Czech za dlouholetou podporu vzdělávání v oblasti databázových technologií a za spolupráci při vytváření této publikace.

Autoři projektu

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Obsah

1.oddíl ... 3
Úvod k databázovým technologiím ... 3
Data versus informace ... 4
Historie databází ... 5
2.oddíl ... 8
Konceptuální & fyzický model ... 8
Entita, instance, atributy a identifikátory ... 10
Modelování entit a vztahů (ERD) ... 12
3.oddíl ... 14
Identifikace vztahů ... 14
Konvence ER Diagramu ... 17
Vyjádření ERDish a kreslení vztahů v diagramu ... 18
4.oddíl ... 20
Supertypy a podtypy ... 20
Dokumentování „podnikových“ pravidel ... 23
5.oddíl ... 25
Typy vztahů ... 25
Řešení M:N vztahu ... 27
6.oddíl ... 29
Umělé, složené a sekundární UID
(unikátní identifikátory) ... 30
Normalizace databáze ... 33
7.oddíl ... 36
Hierarchické a rekurzivní vztahy ... 36
Modelování historických dat ... 38
8.oddíl ... 41
Konvence pro tvorbu diagramů pro lepší čitelnost ... 41
9.oddíl ... 43
Úvod do relačních databází ... 43
Základní mapování:
Proces transformace ERD do RMD ... 46
Mapování vztahů: ... 51

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

1. ODDÍL

Obsah oddílu

- Úvod k databázovým technologiím
- Data versus informace
- Historie databází

Úvod k databázovým technologiím

Lekce 01

S příchodem mikropočítačů a jejich vstupem do života každého z nás se zdá, že vzrůstá zájem o počítačově zpracovávaná data, a to ne pouze ve smyslu něco si vypočítat, ale především ve smyslu něco se dozvědět. Vysvětlení je zcela pochopitelné, uvážíme-li, že i ty nejdokonalejší počítačové hry časem omrzí a že programovat v Basicu výpočty typu řešení soustavy dvou rovnic o dvou neznámých nepřináší až tak velké uspokojení.

Člověk by si rád uspořádal a vyhledával některé informace, které často používá a které se v průběhu používání mění. Za předpokladu vhodných prostředků pro ukládání takových informací ve formě dat by byl ochoten prosedět několik večerů u klávesnice a příslušná data si sám vyrobit, koupit nebo vyměnit s jiným podobným nadšencem. Příkladem může být knihovna, úřad, zpracovávání letenek, městské muzeum, nemocnice, podnik apod.

Hotový objekt popsaného typu můžeme nazvat **informačním systémem (IS).** IS je tedy soubor prvků ve vzájemných informačních a procesních vztazích (informační procesy), které zpracovávají data a zabezpečují komunikaci informací mezi prvky. Z uživatelského hlediska by měl mít IS takové vlastnosti, aby manipulace s ním byla co nejjednodušší (vstup dat, formulace dotazů, použití aplikačních programů), a současně, aby funkce, které poskytuje, byly dostatečně silné a rychlé. Předložit uživateli pružný „prázdný“ IS lze dnes poměrně rychle, avšak s dostatečnou mírou znalostí.

Naším cílem bude ukázat si tzv. databázovou technologii zpracování dat. Databázová technologie je soubor pojmů, prostředků a technik sloužící pro vytváření informačních systémů (IS). Na nejzákladnější úrovni si můžeme představit architekturu IS s databází takto: data jsou organizována v **databázi (DB)**, jsou řízena balíkem programů, který se nazývá **systém řízení bází dat (SŘBD)**. Databáze a SŘBD tvoří tzv. **databázový systém (DBS)**. V naší terminologii můžeme jednoduše psát **DBS = DB + SŘBD**. IS využívá data z DBS buď přímo, nebo je zpracovává aplikačními programy.

Data versus informace

Lekce 02 dd_S01_02

Co se v této lekci naučíte:

- rozlišovat mezi daty a informacemi a uvést příklady dat a informací
- popsat příklad toho, jak se z dat stává informace

Proč se to učit?

- Všechny druhy informací (záznamy školy, záznamy volání z mobilních telefonů, nákupy v obchodě s potravinami) jsou uloženy v databázích. Jsme ve styku s databází každý den, a to vědomě, či nevědomě. Je důležité pochopit, co jsou data uložená v databázi, a co z nich lze získat.

Slova Data, Informace se často používají jako synonyma. Nicméně, mají různé významy.
Data: materiál, z kterého vyvozujeme závěry. Fakta, z nichž dedukujeme nová fakta.

Informace: znalosti, inteligence, příslušná (určitá) data speciálního významu nebo funkce. Informace je často výsledkem spojování, porovnávání a počítání s daty.

Kdykoliv student, učitel, administrátor (nebo kterákoliv osoba používající počítač) pracuje na webu, sbírají se data. GUI může být unikátní pro danou školu nebo společnost, ale co se děje „za oponou“?

Představte si výsledky testu. Jestliže každý student dostane číselný výsledek, pomocí něhož můžete vypočítat průměr třídy. Pak pomocí průměrů tříd můžete určit průměr školy. Oracle databáze převede nahraná / uložená data a statistické údaje na použitelnou informaci.

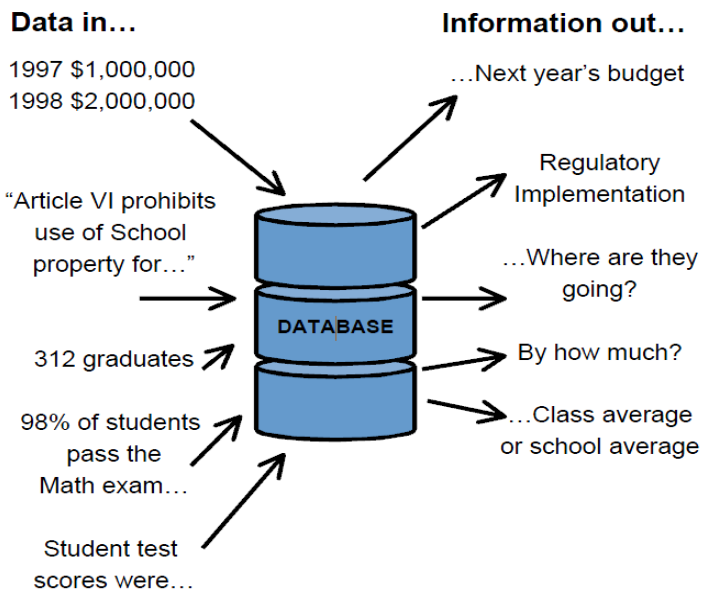
Data: Výsledek testu každého studenta jsou data.

Informace: Průměrný výsledek třídy nebo školy.

Co je databáze?

Databáze je centralizovaná a strukturovaná množina dat uložená v počítači. Poskytuje vybavení na získávání, přidávání, změnu a mazání dat dle potřeby. Taktéž poskytuje vybavení pro transformaci dat na užitečnou informaci.

Databáze je obvykle spravována **databázovým administrátorem** (DBA).



Dokumenty, obrázky, video a zvuk

U nejmodernějších databází můžete získávat a ukládat širokou škálu dat a dokumentů. Uvnitř databáze se data ukládají v „surové“ podobě. Dotázaná nebo získaná surová data jsou převedena na užitečnější výstupy nebo informace.

Otázka: Co má databáze co do činění s mým každodenním životem?

Odpověď: Víc, než si myslíte … Spousta webů, které navštívíte, jsou řízeny databází.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Historie databází

Lekce 03 dd_S01_03

Co se v této lekci naučíte:

- popsat vývoj databáze a uvést příklad jeho role v podnikatelském světě
- důležité historické inovace ve vývoji a konstrukci databáze
- popsat proces rozvoje databáze

Proč se to učit?

- Historie poskytuje pohled na to, kde jsme dnes v oblasti informačních technologií. Až budete příště používat počítač, herní systém nebo PDA, uvědomíte si, jak daleko jsme v ICT došli a co všechno předcházelo tomu, abychom se dostali do tohoto bodu.
- Datové modelování je prvním krokem v rozvoji databází. Tato lekce zahrnuje přehled toho, o čem pojednává zbytek studia databází.

Historie databáze na časové ose

1960: Počítače se staly nákladově efektivní pro soukromé společnosti spolu se zvýšením kapacity uložených dat v letech 1970 — 1972: EF Codd navrhuje relační model databáze, který umožňuje oddělení logického uspořádání od fyzického uložení dat.

1976: P. Chen představuje konceptuální model (ERM) pro návrh databáze.

Začátek 80. let: První komerčně dostupný relační databázový systém se objevil na začátku roku 1980 – Oracle verze 2.

Polovina 80. let: SQL (Structured Query Language) je standardizován.

Polovina 90. let: Objevuje se použitelný Internet / World Wide Web (WWW). To vede k obrovským aktivitám umožňující vzdálený přístup k počítačovému systému se staršími daty.

Konec 90. let: velké investice do internetových firem pomáhají vytvářet nástroje pro tržní propojení Web / Internet / DB.

Počátek 21. století: Pokračuje souvislý nárůst používání DB.

Příklady: Komerční internetové stránky (yahoo.com, amazon.com), vládní systémy (Úřad státního občanství a imigrační služby, Úřadu pro sčítání lidu), umění, muzea, nemocnice, školy atd.

Otázka: Co má společného databáze s modelováním dat? Datové modelování je první část procesu vývoje databáze.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost

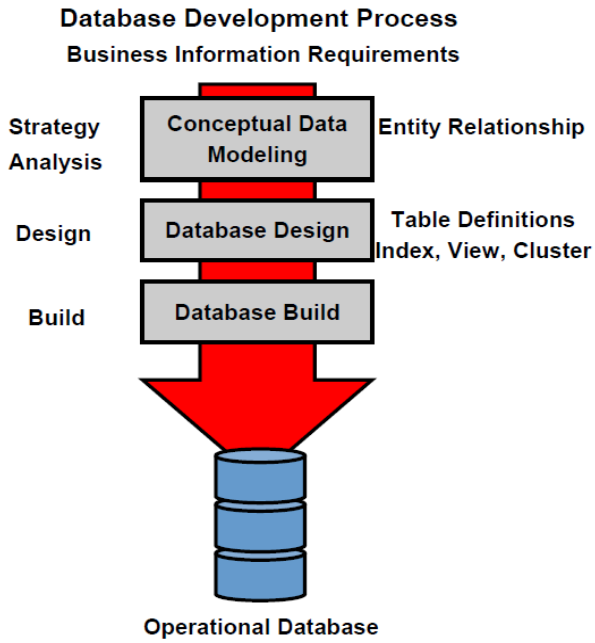
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Proces vývoje databáze začíná požadavky na podnikové informace

PŘÍKLAD

Zde je soubor požadavků na informace:

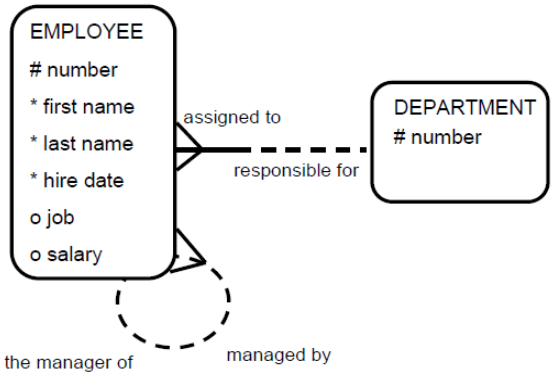
„Řídím oddělení lidských zdrojů velké společnosti. Potřebujeme uchovávat informace o každém z našich zaměstnanců. U každého zaměstnance musíme sledovat jméno, příjmení, pracovní zařazení nebo funkci, datum nástupu a plat. U všech zaměstnanců musíme také sledovat jejich potenciální provizi. Každý zaměstnanec má přiřazeno unikátní číslo zaměstnance. Naše společnost je rozdělena na oddělení. Každý zaměstnanec je přiřazen oddělení – například účetnictví, prodej nebo vývoj. Potřebujeme vědět, který zaměstnanec je zodpovědný za dané oddělení a také umístění oddělení. Každé oddělení má jedinečné číslo. Někteří zaměstnanci jsou manažeři. Potřebujeme vědět, kdo je vedoucím každého zaměstnance.“



Proces vývoje databáze

ERD („entitně relační model“) model by měl přesně modelovat informace, které potřebuje organizace, a podporovat funkčnost podniku.

PŘÍKLAD



Následující konceptuální model (ERD) představuje požadavky na informace v oblasti lidských zdrojů.

V návrhu databáze se požadavky na informace projeví v modelu jako entity a vztahy a jsou převedeny do relační databáze použitím tabulky instancí.

Tabulka má tyto části:

- Název tabulky.
- Názvy sloupců.
- Klíče: primární klíč (PK) je jedinečný identifikátor pro každý řádek dat, cizí klíč (FK) spojuje data jedné tabulky s daty jiné tabulky.
- Null: uveďte, zda sloupce musí mít hodnotu (povinná hodnota).
- Unikátnost: označujeme, pokud je hodnota ve sloupci unikátní v rámci tabulky.
- Datový typ: týká se formátu a definice dat v každém sloupci.

K vytvoření fyzické databáze jsou používány SQL příkazy (DDL — „Data Definition Language“)

Ukázka tvorby databáze:

```
SQL>CREATE TABLE DEPARTMENT
DEPTNO NUMBER(5) NOT NULL PRIMARY KEY, NAME VARCHAR2(25) NOT NULL,
LOC VARCHAR2(30) NOT NULL);

SQL>CREATE TABLE EMPLOYEES
(EMPNO NUMBER(9) NOT NULL PRIMARY KEY, FNAME VARCHAR2(15) NOT NULL,
LNAME VARCHAR2(20) NOT NULL,
JOB VARCHAR2(15),
HIREDT DATE NOT NULL,
SAL NUMBER(9,2),
COMM NUMBER(9,2),
MGR NUMBER(2) REFERENCES EMPLOYEES DEPTNO NUMBER(5) REFERENCES
DEPARTMENT);
```

2. ODDÍL

Obsah oddílu

- Konceptuální & fyzický model
- Entita, instance, atributy a identifikátory
- Modelování entit a vztahů (ERD)

Konceptuální & fyzický model

LEKCE 01 dd_S02_L01

V této lekci se naučíte:

- popsat důležitost požadovaných informací
- rozlišovat mezi konceptuálním modelem a fyzickou implementací
- seznam 5 důvodů pro výstavbu konceptuálního datového modelu
- seznam příkladů konceptuálních a fyzických modelů

Proč se to učit?

- Když budete schopni rozlišit a analyzovat informace, lépe pochopíte, jak věci fungují a potenciálně je vylepšíte.
- Např.:
jak zrychlit řadu u prodejního pultíku,
jak v obchodě něco úspěšně vyměnit,
jak zorganizovat a sledovat rostoucí sbírku CD nosičů.
- Rozpoznání a analýza informací pomáhá zamezit chybám a nedorozuměním. To je důležité v podniku, jelikož se šetří čas a peníze.

Co je to Konceptuální model?

Konceptuální model je model, který zakresluje funkční a informační potřeby podniku.

Na základě aktuálních potřeb může odrážet budoucí potřeby. Zabývá se pouze potřebami podniku a nezabývá se problémem jejich implementace.

Nazývá se Model entit a vztahů („Entity Relationship Model"). Zobrazuje se jako Diagram entit a vztahů ("Entity Relationship Diagram").

Firmy využívají data k růstu nebo ke snížení svých nákladů. V procesu vytváření datového modelu podniku představuje právě konceptuální model data podniku.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Účelem konceptuálního modelu je:

- Přesně popsat potřeby podniku na informace.
- Usnadnit diskuzi.
- Předejít chybám a nedorozuměním.
- Zformovat důležité „ideální systémovou“ dokumentaci vytvořit základ pro fyzický návrhu databáze.
- Dokumentovat procesy podniků (to je také známo jako „business rules“).
- Vzít v úvahu předpisy a zákony, které upravují dané výrobní odvětví .

Zhruba od poloviny 70. let se začalo prosazovat tzv. **konceptuální schéma (KS),** které formalizovaně, ale přitom dostatečně srozumitelně, popisuje reálný svět a ne pouze data v počítači. Konceptuálním schématem popisujeme jednak strukturu světa uživatelské aplikace, jednak jistá fakta, která se v čase nemění, např. Materiál je jednoznačně určen číslem materiálu z číselníku. Tato tvrzení představují **integritní omezení (IO)** na konceptuální úrovni.

Objekty, vyhovující konceptuálnímu schématu, tvoří **informační bázi**. (Je to velmi abstraktní pojem – není „nikde vidět“.)

Implementací informační báze je **databáze.**

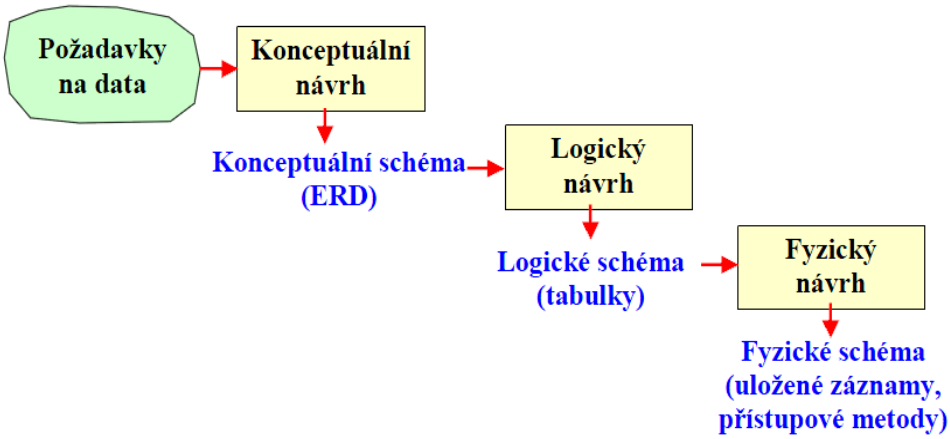
KS je implementačně nezávislé a slouží jako společný základ pro: pochopení objektů aplikace uživateli, projektanty atd. (společný pohled na věc různými skupinami zúčastněných), integraci uživatelských pohledů a návrh implementace (vnášení kompromisu do různých pohledů na tutéž realitu. Odstraňují se zde např. synonyma či homonyma).

KS je tedy výsledkem analýzy a návrhu IS. Jde o popis obsahující pro IS všechny důležité objekty reálného světa a tvoří dokumentaci budoucího IS.

Prostředky, kterými se KS vyjadřuje, se nazývají konceptuální či sémantické modely.

Chování aplikace se analyzuje funkční analýzou (většinou málo integrovanou s datovou analýzou).

Fáze návrhu databáze



Shrnutí:

Konceptuální modely jsou nástrojem pro vytvoření popisu dat v databázi, tj. **konceptuálního schématu**, nezávisle na logické úrovni i fyzickém uložení databáze. Tento popis by měl co nejvěrněji vystihnout konceptuální pohled člověka na danou část reálného světa.

Konceptuální modely slouží obvykle k vytvoření schémat s následnou transformací na databázové schéma (fyzické schéma).

Konceptuální a fyzikální modely jsou tedy uměním plánování, vývoje a komunikace, které produkuje požadovaný výsledek.

Datové modelování zahrnuje získávání požadavků, jejichž popis je zachycen pomocí diagramu. Tento diagram se stane plánem pro navrhování aktuálních požadavků.

Klientovo zadání (konceptuální model) se stane souborem fyzických objektů (fyzický model).

Entita, instance, atributy a identifikátory

LEKCE 02 dd_S02_L02

Co se naučíte:

- definovat a poskytnout příklad entit
- rozlišovat mezi entitou a instancí entity
- pojmenovat a popsat atributy určených entit
- rozlišit mezi atributem a jeho hodnotou
- rozlišovat mezi povinnými a volitelnými atributy a mezi nestálými a stálými
- dokážete vybrat a obhájit jedinečné identifikátory entit

Proč se to učit?

- Znalost, jak organizovat a klasifikovat data, umožňuje vyvozovat použitelné závěry o zdánlivě náhodných faktech. Náš svět, bohatý na technologie, vytváří obrovské množství faktů, které potřebují organizovat a řadit.
- Je důležité učit se o entitách, protože o nich ukládáme data. Příklad: Škola potřebuje uložit data (minimálně) o studentech, učitelích, hodinách, třídách a známkách. Je důležité učit se o atributech, protože informují o entitách a atributy pomáhají specifikovat, která data potřebujeme sledovat.

Např.:

- V restauraci potřebujete seznam různých položek seřazených tak abyste věděli, kolik požadovat financí.
- Když pro oddělení vytváříte různé zprávy, musíte "vypíchnout" příslušné zprávy ze seznamu zpráv.

A co jedinečné identifikátory?

- Je důležité rozumět unikátním identifikátorům, protože rozlišují jednu instanci od druhé.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Např.:

- Ve třídě potřebuješ odlišit jednu osobu od druhé.
- Při klasifikaci sbírky CD potřebuješ najít konkrétní CD.
- Ve finančním výpisu seznamu transakcí potřebujete rozlišit mezi následnými transakcemi z jednoho dne.

Projektant se musí podrobně seznámit s modelovanou realitou podniku, specifikovat požadavky na IS a v rámci datové analýzy:

- Identifikuje **entity** (entity types) jako množiny objektů stejného typu (STUDENT, UČITEL, KNIHA, ABONENT, ZAMĚSTNANEC). V objektové terminologii bude tento pojem většinou splývat s pojmem třída.
- Identifikuje typy **vztahů** (relationship types), do kterých mohou entity vstupovat (UČÍ, MÁ_PŮJČEN,...).
- Na základě přiměřené úrovně abstrakce přiřadí entitám **popisné atributy** (PŘÍJMENÍ, DATUM_VÝPŮJČKY).
- Formuluje různá **integritní omezení** – pravidla vyjadřující (s větší nebo menší přesností) soulad schématu s modelovanou realitou.

Proč termín „omezení“? Protože většinou jde skutečně o omezení v reálném světě – modelované databázi. IO nám tedy také mimo jiné zjednodušují návrh (např. máme-li IO formulované: „Každý klient má v naší bance nejvýše jednu půjčku“ – jistě povede na jednodušší řešení, než kdybychom museli řešit situaci, že klient může mít libovolný počet půjček); a integritní proto, že integrita je obecně soulad dat s realitou.

Konstrukty E-R modelu:

Entita … objekt reálného světa, který je:

- schopen nezávislé existence,
- je jednoznačně odlišitelný od jiných objektů.
- Entity mají instance. Instance je jeden výskyt entity. Např. (student NOVÁK JOSEF, r.č. 771201005).

Vztah ...vazba mezi dvěma nebo více entitami, např. student NOVÁK JOSEF může být ve vztahu „studuje“ k entitě předmět INFORMAČNÍ A KOMUNIKAČNÍ TECHNOLOGIE.

Atribut ... údaje přiřazující entitám či vztahům hodnotu (popisného typu), určující některou podstatnou vlastnost entity, např. datum výpůjčky dané knihy daným abonentem. Hodnota atributu může být číslo, řetězec, datum, obraz, zvuk atd. Jedná se o tzv. „datové typy“ nebo „formáty“. Každý atribut má určitý typ dat.

Atributy jsou atomické, tzn. že každý atribut může mít pouze jednu hodnotu (v každém okamžiku) pro každou instanci subjektu.

Vymezení těchto pojmů je volné. Klasifikace dat jako ENTITA nebo VZTAH závisí na pohledu analytika.

Pravidla pro pojmenování: entity převážně vyjadřovány podstatnými jmény; vztahy převážně vyjadřovány slovesy; nepoužíváme diakritiku a mezeru v názvu.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Modelování entit a vztahů (ERD)

LEKCE 03 dd_S02_L03

Co se naučíte:

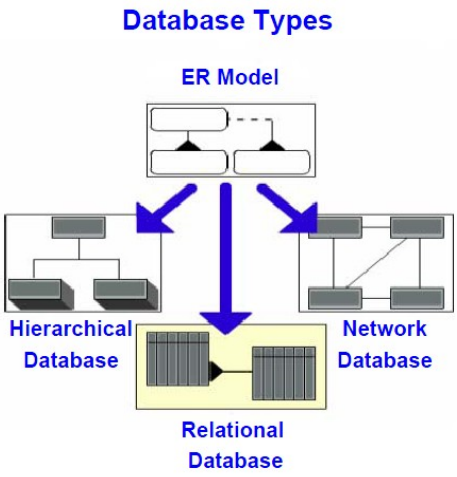
- definovat význam volná implementace (implementation-free) tak, jak se vztahuje k implementaci datových modelů a návrhu databáze
- seznam 4 cílů modelování entit a vztahů
- identifikovat ERD

Proč se to učit?

- Diagram entit a vztahů (ERD) je konzistentní nástroj používaný k reprezentaci datových požadavků bez ohledu na použitý typ databáze, dokonce bez samotného použití databáze.

Bez-implementační modely (volné)

- Dobrý konceptuální datový model se nemění při změně typu databáze, na níž je nakonec systém vystavěn. A proto říkáme, že model je bez implementace.
- Datový model by měl zůstat stejný, i když není použita žádná databáze, např. Data jsou uložena jako kartičky v kartotéce.



Co je ERD?

Seznam všech entit a atributů, rovněž seznam vztahů mezi důležitými entitami. Poskytuje podklady jako popis entit, datové typy a omezení dat. Pozn. Model nemusí obsahovat diagram, ale typicky je velmi užitečný.

Cíle modelování ERD:

Máme celkem 4 cíle modelování ERD:

- podchytit všechny potřebné informace,
- zajistit, aby se každá informace objevila pouze jedenkrát,
- vymodelovat informaci, která by se dala odvodit z jiné modelované informace,
- umístit informaci na očekávaném logickém místě.

Představte si školní záznamy. Od prvního dne se o vás sbírají data. Pravděpodobně jsou o vás zaznamenány informace o absenci, chování, rozvrhu a známkách.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

SCÉNÁŘ DJ ON DEMAND

Přečtěte si celý obchodní scénář firmy DJ on Demand, níže uvedený, a poté ověřte vytvořený ERD.

Obchod jsme rozjeli jako skupina přátel, kteří organizovali večírky, kde jsme používali vlastní hudbu. Pak jsme si řekli, že založíme firmu, abychom sledovali své zájmy a vydělali více peněz. Nazvali jsme se DJ on Demand („Žádaní DJ“).

Všichni zde pracující jsme partneři. Každý máme určitou zodpovědnost. Projektový manager domlouvá první schůzku s klientem, kde se projednává typ akce. Jde o narozeniny? Svatbu, výročí, promoce? Kdy akce proběhne?

Po dohodě následuje plánovač akce, který s klientem probere konkrétní místa, občerstvení, výzdobu a další podrobnosti.

DJ s klientem probere požadovanou hudbu. Projektový manager dohlíží na DJ a plánovače. Taktéž podepisuje výdaje spojené s projektem. Máme velkou sbírku CD. Každé CD obsahuje několik písní a stejná píseň se může objevit na několika CD. Rozdělujeme písně podle typu (hip-hop, salsa, rock, pop, classic ...).

Můžeme nabídnout vzorový seznam písní, upravených podle typu akce. Samozřejmě i klient si může vyžádat určité písně. Seznam klientů nám roste. Naši klienti se k nám vracejí, protože mají rádi naši práci, a připravujeme pro ně další akce. Máme hodně aktivní zákazníky, pro které pořádáme i několik akcí najednou.

Akce dělíme tematicky. Např. tropická svatba, karnevalový večírek, výročí ve stylu 60. let. To nám pomáhá vybrat místo a vytvořit si představu, jak by měl být DJ a další hudebníci oblečení. Někteří partneři mají specializace a kvalifikace – takže nám téma pomůže odhadnout správnou osobu pro danou akci.

Akce konáme ve veřejných budovách i v soukromých domech. Manažer akce navštíví veřejné místo nebo soukromý dům. Manažer akce domlouvá podmínky, jak s nájemcem veřejné budovy, tak i s vlastníkem domu. Jelikož několik partnerů pracuje na akci a několik partnerů akci řídí, míváme zaznamenáno, kdo pracuje na jaké akci, jakou práci vykonal a kdy.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

3. ODDÍL

Obsah oddílu

- Identifikace vztahů
- Konvence ER Diagramu
- Vyjádření ERDish a kreslení vztahů v diagramu

Identifikace vztahů

LEKCE 01 dd_S03_L01

V této lekci se naučíte:

- interpretovat a popisovat volitelnost vztahu (členství ve vztahu)
- interpretovat a popisovat poměr ve vztahu
- nastavit vztah mezi entitami aplikováním pravidla poměru a volitelnosti vztahu

Proč se to učit?

- Tím, že jste schopni porozumět identifikaci vztahů mezi entitami, lépe chápete vztahy mezi různými daty.
- Díky této schopnosti rozeznáte, jak se navzájem ovlivňují odlišné části systému. Např.: Entity STUDENT a KURZ mají mezi sebou určitý vzájemný vztah.
- Abychom správně vymodelovali zadání, vztahy jsou stejně důležité jako entity.

Vztahy v datových modelech:

Vztahy:

- reprezentují něco důležitého v zadání,
- ukazují, v jakém vztahu jsou entity,
- vždy existují mezi dvěma entitami,
- vždy mají dvě strany,
- jsou pojmenovány na obou koncích,
- mají volitelnost (členství ve vztahu),
- mají stupeň poměru ve vztahu.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Co je volitelnost ve vztahu?

Vztahy jsou buď povinné, nebo nepovinné. Vycházíme-li z toho, co známe o instancích entit, můžeme určit volitelnost, když odpovíme na 4 otázky: (příklad „firma“)

- Musí mít každý zaměstnanec práci? Jinými slovy: je toto povinné, nebo nepovinné členství ve vztahu zaměstnance k práci?
- Mohou mít zaměstnanci více než jednu práci?
- Musí být každá práce prováděna nějakým zaměstnancem? Jinými slovy, má práce povinný nebo nepovinný vztah k zaměstnanci?
- Může být práce vykonávána více než jedním zaměstnancem?

Co je poměr ve vztahu?

Poměr ve vztahu zahrnuje stupeň ve vztahu. Odpovídá na otázku „kolik“.

Příklad: Kolik má zaměstnanec zaměstnání? Jedno nebo více? Pouze jedno? Kolik zaměstnanců má dané zaměstnání? Jeden nebo více?

Volitelnost a poměr ve vztahu

PŘÍKLAD

Každý zaměstnanec musí mít pouze jedno zaměstnání.

Každé zaměstnání může být obsazeno jedním nebo více zaměstnanci.

Každý produkt musí být klasifikován pouze jedním typem produktu.

Každý typ produktu může klasifikovat jeden nebo více produktů.

Vztahy

- Každé sedadlo může být prodáno jednomu nebo více cestujícím.
- Každý pasažér si může koupit jedno sedadlo.
- Sedadlo je prodáno cestujícímu (nebo cestujícím, proto dojde ke změně rezervace).
- Pasažér si koupí nebo rezervuje jedno sedadlo.

OBCHODNÍ SCÉNÁŘ 1:

Jaké jsou vztahy v následujícím scénáři?

„Rádi rozdělujeme veškerou hudbu podle typu – každou písničku, každý soudtrack. Různé typy jsou jazz, country, pop …. Podle potřeby můžeme přidat nové typy, vlastně jsme teď přidali nový typ rapová hudba a také víme, že píseň může být zařazena do různých typů, ale pro naše potřeby jsme si vybrali pro každou písničku jeden typ.“

volitelnost = „musí“ nebo „může“?

poměr = kolik?

Píseň má typ: volitelnost a poměr

Každé **písni** je přiřazen jeden (pouze jeden) **typ**.

Každý **typ** může klasifikovat jednu nebo více **písní**.

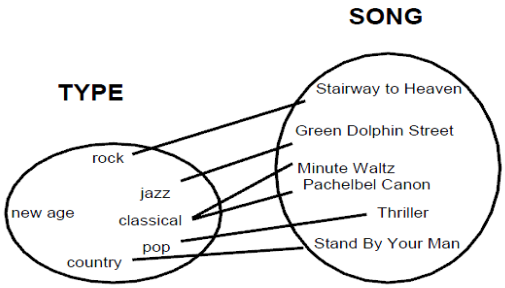
... co když neexistuje pro píseň typ?

Uvádějí obchodní (podniková) pravidla, že každá píseň má typ? Pokud ano, pak je potřeba přidat typ navíc.

Může existovat typ bez písně?

Proč bys měl mít typ bez písně?

Pod kolik typů může píseň patřit?



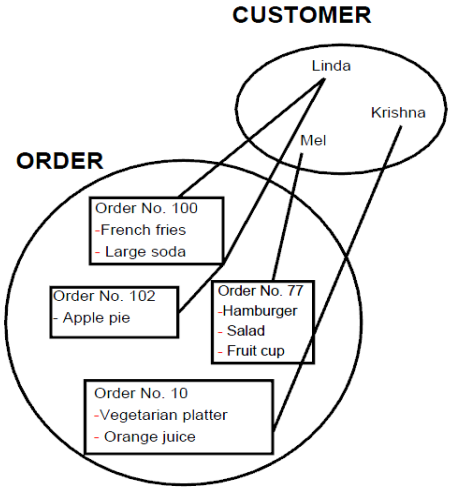
Pravidla určují poměr. Ve vztahu.

Pokud písnička může patřit pod více než jeden typ, pak by měl být poměr určen jako: Každou píseň musí klasifikovat jeden nebo více typů.

OBCHODNÍ SCÉNÁŘ 2:

Jaké jsou vztahy v následujícím scénáři?

„U nás v restauraci přijde host k baru a objedná si. Zákazník si může objednat jen pro sebe nebo pro celou skupinu lidí. Např.: Jestliže matka objedná pro sebe a děti, pak bereme matku za zákazníka, který je vlastníkem objednávky a je zodpovědný za platbu. Zákazník samozřejmě může učinit tolik objednávek, kolik chce. “



Zákazník má objednávky: volitelnost a poměr

volitelnost = musí nebo může?

Poměr ve vztahu = kolik?

Každá objednávka musí být obdržena jedním (pouze jedním) zákazníkem.

Každý zákazník musí dát jednu nebo více objednávek.

OBCHODNÍ SCÉNÁŘ 3:

Vztahy v rámci stejné entity.

Vyzkoušejte si následující scénář:

„Potřebujeme vést záznam o zaměstnancích a jejich managerech. Každý zaměstnanec má jednoho manažera, včetně generálního ředitele, který řídí sám sebe. Každý manažer může řídit více zaměstnanců. Jelikož jsou manažeři také zaměstnanci, existuje zde jen jedna entita – zaměstnanec“

Vztahy:

Zaměstnanec řídí zaměstnance.

Zaměstnanec je řízen jedním zaměstnancem.

Konvence ER Diagramu

LEKCE 02 dd_S03_L02

V této lekci se naučíte:

- vytvářet komponenty (konstrukty) ER Diagramu, které představují entity a atributy, podle konvencí diagramu

Proč se to učit?

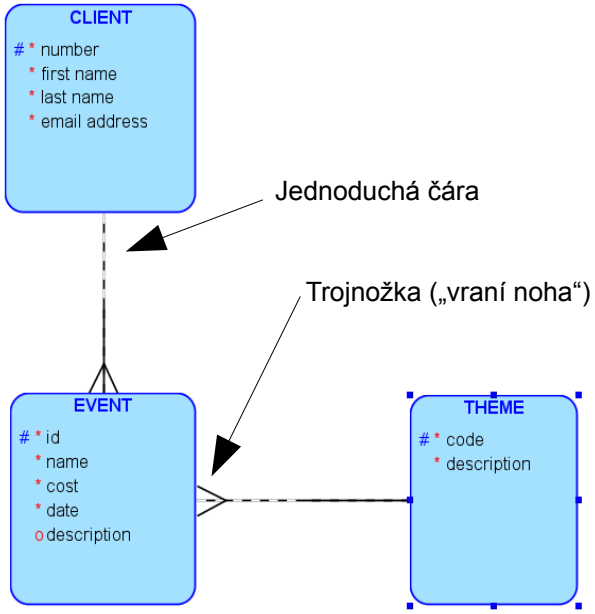
- Lidé po celém světě hovoří různými jazyky, ale všichni rozumí dopravním značkám.
- Je efektivní zpracovávat informaci způsobem, kterému rozumí mnoho lidí. ERD fungují v tomto významu. Můžete popisovat nebo napsat věci odlišně, protože mluvíte jinak (přízvuk je jiný), ale každý maluje diagramy podle stejných principů.

DJ on Demand: Klienti, akce a typy

Popis reálného světa – viz „Scénář DJ on DEMAND“ v lekci Modelování entit a vztahů.

Konvence kreslení ERD

- Entity jsou reprezentovány rámečky.
- Jména entit zapisujeme do rámečků.
- Názvy entit jsou vždy v jednotném čísle a velkými písmeny.



Principy kreslení:

- Seznam Atributů umísťujeme pod jméno entity.
- Povinné atributy označujeme hvězdičkou: „*“.
- Nepovinné atributy označujeme kroužkem: „○“.
- Jedinečné identifikátory označujeme mřížkou: „#“.
- Vztahy jsou čáry, které propojí entity.
- Tyto čáry jsou buď plné, nebo čárkované. Čáry končí jednoduchou čarou nebo trojnožkou (vraní noha).

Více se o významu vztahových čar dozvíte dál.

esf evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Vyjádření ERDish a kreslení vztahů v diagramu

LEKCE 03 dd_S03_L03

V této lekci se naučíte:

- přesně popsat (slovy) vztahy mezi entitami (schéma ERDish), správně kreslit a označit vztahy v ERD

Proč se to učit?

- Většina činností má unikátní terminologii – slova, která mají speciální význam v rámci dané činnosti – kterou lidé používají, aby předali a zpracovali informaci.
- Modelování dat má také jedinečnou terminologii, kterou nazýváme ERDish. Když se naučíte, jak vytvořit diagramy a jak hovořit ERDish, získáte běžnou terminologii pro komunikaci s klienty a administrátory, kteří implementují váš návrh.

ERDish je jazyk, kterým vyjadřujeme vztahy mezi entitami.

Už jsme tímto jazykem hovořili a psali, když jsme určovali vztahy a specifikovali volitelnost a poměr ve vztahu. Prostě jednoduše rozložíme každou ERDish větu na jednotlivé části.

Části ERDish:

1. EACH („každá“)
2. ENTITA A
3. OPTIONALITY („musí“/ „může“)
4. RELATIONSHIP NAME (jméno vztahu)
5. CARDINALITY („jedna a pouze jedna“/ „jedna nebo více“)
6. ENTITA B

1. EACH
2. **EMPLOYEE** (entity A)
3. **MUST BE** (optionality, solid line)
4. **WORKING IN** (relationship name)
5. **ONE** (cardinality, single toe)
6. **DEPARTMENT** (entity B)

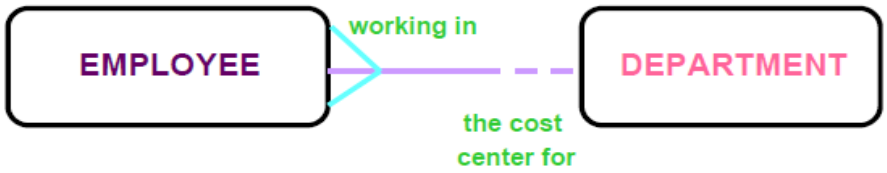
1. EACH
2. **DEPARTMENT** (entity B)
3. **MAY BE** (optionality, dotted line)
4. **THE COST CENTER FOR** (relationship name)
5. **ONE OR MORE** (cardinality, crow's foot)
6. **EMPLOYEE** (entity A)

Jelikož má vztah dvě strany, čteme nejprve jednu stranu – zleva doprava.

Potom čteme vztah zprava doleva.

esf evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Nyní spojíme obě vyjádření dohromady:



1. EACH
2. **EMPLOYEE** (entity A)
3. **MUST BE** (optionality, solid line)
4. **WORKING IN** (relationship name)
5. **ONE AND ONLY ONE** (cardinality, single toe)
6. **DEPARTMENT** (entity B)

1. EACH
2. **DEPARTMENT** (entity B)
3. **MAY BE** (optionality, dotted line)
4. **THE COST CENTER FOR** (relationship name)
5. **ONE OR MORE** (cardinality, crow's foot)
6. **EMPLOYEE** (entity B)

4. ODDÍL

Obsah oddílu

- Supertypy a podtypy
- Dokumentování „podnikových“ pravidel

Supertypy a podtypy

LEKCE 01 dd_S04_L01

V této lekci se naučíte:

- definovat a uvést příklad podtypu
- definovat a uvést příklad supertypu
- uvést pravidla týkající se entit a podtypů, uvést jejich příklady
- použít pravidla pro supertypy a podtypy vyhodnocením přesnosti ER diagramů, které je reprezentují
- použít pravidla pro supertypy a podtypy a zahrnout je vhodně do diagramu

Proč se to učit?

- Supertypy a podtypy se často vyskytují v reálném světě – objednávky jídla (konzumace na místě, s sebou), nákupní tašky (papírové, plastové), typy platby (šek, hotovost, úvěr).
- Chápání příkladů z reálného světa nám pomáhá pochopit, jak a kdy je modelovat.

Některé instance entity mají často atributy a/nebo vztahy, které se v jiných instancích nevyskytují. Představte si podnik, který potřebuje sledovat platby od zákazníků. Zákazníci mohou platit v hotovosti, šekem nebo kreditní kartou.

Všechny platby mají některé společné atributy: datum platby, výše platby atd.. Ale pouze kreditní karty budou mít atribut "číslo karty".

U kreditních karet a plateb šekem budeme chtít vědět, který ZÁKAZNÍK platbu provedl, přičemž u plateb v hotovosti to není potřeba.

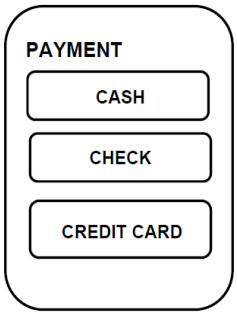
Měli bychom vytvořit jedinou platební entitu nebo tři samostatné entity HOTOVOST, ŠEK, KREDITNÍ KARTA? A co se stane, pokud v budoucnu zavedeme čtvrtou metodu platby?

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Někdy má smysl rozdělit entitu do podtypů. Např. v případě, kdy má skupina instancí zvláštní vlastnosti, jako jsou atributy nebo vztahy, které existují pouze pro tuto skupinu. V tomto případě se entita nazývá "supertyp" a každá skupina se nazývá podtyp.

Podtyp:

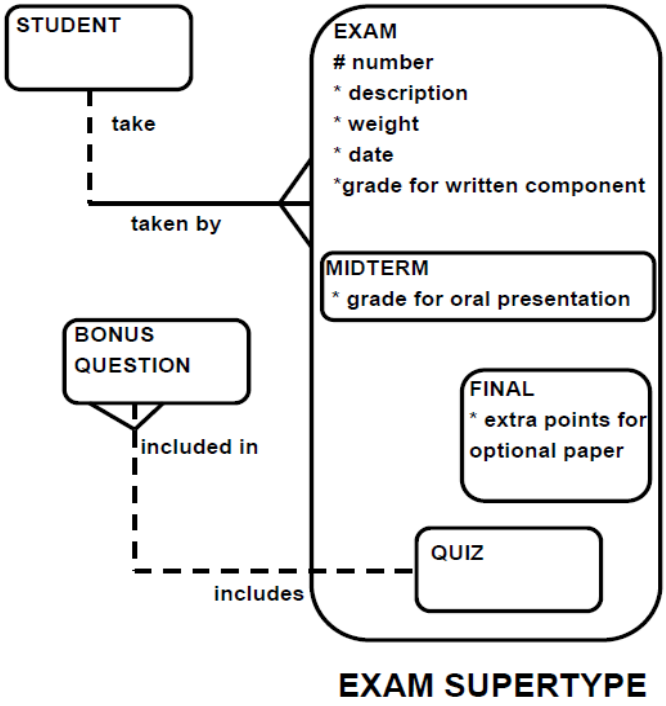
- přebírá všechny atributy supertypu
- přebírá všechny vztahy mezi supertypy
- má obvykle své vlastní atributy nebo vztahy
- je vybrán v rámci supertypu
- nikdy neexistuje sám o sobě
- může mít vlastní podtypy
- také se mu říká "podentita“



PŘÍKLAD:

ZKOUŠKA je supertyp zahrnující KVÍZ, POLOLETNÍ ZKOUŠKA a ZÁVĚREČNÁ ZKOUŠKA.

Podtypy mají několik společných atributů. Tyto společné atributy jsou uvedeny na úrovni supertypu. Totéž platí pro vztahy. Podtypy přebírají všechny atributy a vztahy entity typu supertyp.



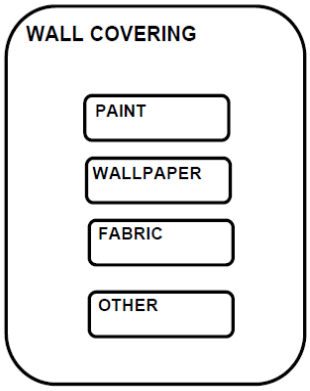
Vždy více než jeden podtyp

Když je ER model kompletní, nikdy nemá samostatné podtypy. Jinak řečeno, pokud má entita podtyp, měl by vždy existovat alespoň jeden další podtyp. Dává to smysl. K čemu by bylo rozlišovat mezi entitou a jediným podtypem?

Z této myšlenky plynou dvě pravidla pro podtypy:

- **Úplné:** Každá instance supertypu je zároveň instancí jednoho z podtypů.
- **Vzájemně se vylučující:** Každá instance supertypu je instancí pouze jednoho podtypu.

POVRCHOVÁ ÚPRAVA STĚN



Ve fázi koncepčního modelování je dobrým zvykem zařadit podtypu OSTATNÍ, aby bylo jisté, že je váš seznam podtypů vyčerpávající – že jste ošetřili každou instanci supertypu.

Podtypy vždy existují

Každé entitě lze vždy přiřadit podtypy. Vždy můžete vytvořit pravidlo, jak rozdělit instance do skupin.

Ale to není to hlavní. Důvodem pro tvorbu podtypů by vždy měla být potřeba ukazovat zároveň podobnosti a rozdíly.

Správná identifikace podtypů

Při zvažování supertypů a podtypů si můžete položit tři otázky pro ověření, zda jste podtyp správně identifikovali:

(1) Je tento podtyp druhu supertyp?

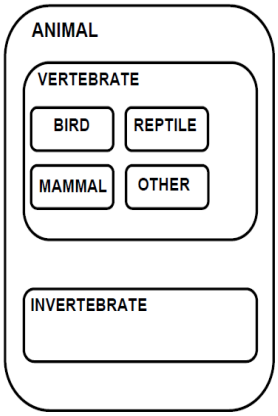
(2) Ošetřil jsem všechny možné případy? (úplný)

(3) Je možné daný příklad zařadit pouze pod jeden podtyp? (vzájemně se vylučující)

Vnořené podtypy

Podtypy můžete vnořit. Pro snadné čtení -- "čitelnost" – obvykle ukazujete podtypy pouze do dvou úrovní, ale neexistuje žádné pravidlo, které by vám bránilo jít i do dalších úrovní.

VNOŘENÝ SUPERTYP ŽIVOČICH



esf evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Dokumentování „podnikových“ pravidel

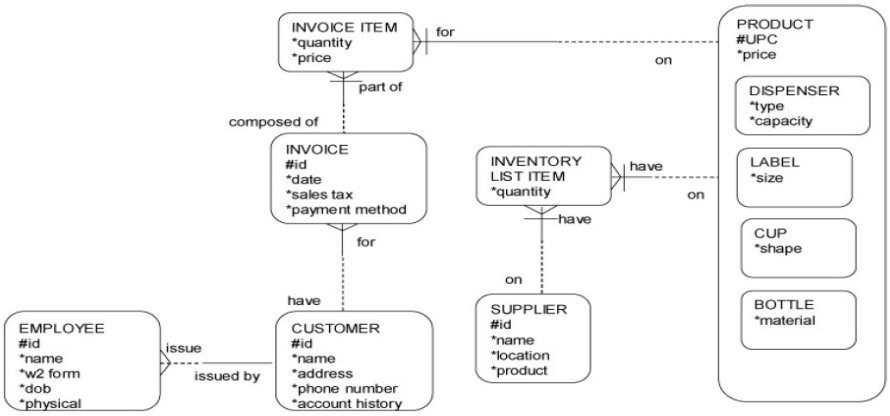
LEKCE 02 DD_S04_L02

V této lekci se naučíte:

- definovat a tvořit strukturální obchodní pravidla
- definovat a tvořit procesní obchodní pravidla
- rozpoznat, že některá obchodní pravidla vyžadují programování
- vytvořit diagram obchodních pravidel, je-li možné je vyjádřit ER modelem

Proč se to učit?

- Dokážete identifikovat tuto firmu?
- Jedním z hlavních cílů datového modelování je zajistit, že všechny informace, které jsou potřeba pro běh podniku, jsou identifikovány.
- Identifikace a dokumentování obchodních pravidel jsou klíčová pro ověření přesnosti a úplnosti datového modelu.
- Je důležité si uvědomit, že ne všechna obchodní pravidla lze vyjádřit ve formě ER diagramu. Některá obchodní pravidla je nutné naprogramovat.



Dva typy obchodních pravidel: strukturální a procesní

Strukturální obchodní pravidla uvádějí typy informací, které se mají ukládat, a jak jsou elementy těchto informací vzájemně propojeny.

Procesní pravidla souvisejí s workflow nebo obchodním procesem.

Strukturální obchodní pravidla lze téměř vždy vyjádřit ER diagramem. Některá procesní obchodní pravidla nelze vyjádřit diagramem, ale i tak je nutné je zdokumentovat.

Mnoho procesních pravidel souvisí s časem: událost A musí stát dříve než událost B.

Strukturální obchodní pravidla uvádějí typy informací, které se mají ukládat, a jak jsou elementy těchto informací vzájemně propojeny.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Příklady strukturálních obchodních pravidel:

Všechny objednávky v restauraci musí zpracovávat zaměstnanec (konkrétně, pracovník přijímající objednávky).

Neexistuje žádný samoobslužný systém objednávek.

Všichni učitelé na naší škole musí vlastnit platnou akreditaci.

DISKUSE:

Jaké druhy pravidel u vašeho zaměstnavatele se týkají vás?

- Každá směna, kterou odpracuji, se musí zadokumentovat na záznamové kartě zaměstnance.
- Každá směna musí probíhat pod dohledem nadřízeného.

Naše škola má mnoho obchodních pravidel:

- Je rozumné/efektivní, aby třída neměla přiděleného třídního učitele?
- Je rozumné/efektivní, aby dva studenti měli stejné studentské ID číslo nebo vůbec žádné ID číslo?
- Je rozumné naplánovat učiteli výuku ve třídě, pokud do ní není zapsán žádný student?
- Je rozumné dovolit studentům chodit do školy, pokud nejsou zapsáni do žádné (jedné nebo více) třídy?

Procesní pravidla souvisí s workflow nebo obchodním procesem.

Příklady procesních obchodních pravidel:

Prvotní kontakt s klientem DJs on Demand musí provést projektový manažer.

Schválení cestovní žádostí na konkrétní událost musí podepsat projektový manažer pro tuto událost.

DISKUSE

- Studenti musí absolvovat algebru a geometrii, aby se mohli zapsat na trigonometrii. Dokážete to znázornit ER diagramem?
- Jak byste to naprogramovali?
- Pokud student absolvoval dané předměty, napadá vás další obchodní pravidlo, které by škola v tomto scénáři mohla chtít?

Při vytváření koncepčního datového modelu nelze vždy namodelovat všechna obchodní pravidla. Některá pravidla, jako ta níže uvedená, je nutné naprogramovat jako procesy, které interagují s daty:

každý zaměstnanec, který má více než 10 přesčasových hodin týdně, musí dostat 1,5 násobek hodinové mzdy.

Nebo:

zákazníkům, jejichž závazky na účtu jsou 90 dní po splatnosti, nebude umožněno zadávat další objednávky.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

5. ODDÍL

Obsah oddílu

- Typy vztahů
- Řešení M:N vztahů

Typy vztahů

LEKCE 02 dd_S05_L02

V této lekci se naučíte:

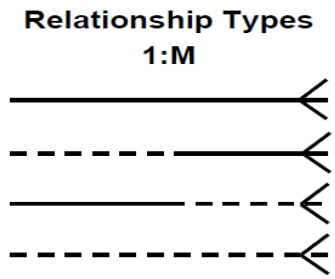
- rozlišovat a vytvářet příklady vztahu jedna ku jedné (1:1)
- rozlišovat a vytvářet příklady vztahu jedna ku více (1:N)
- rozlišovat a vytvářet příklady vztahu více ku více (M:N)
- rozpoznávat redundantní (vícenásobné) vztahy a odstranit je z ERD

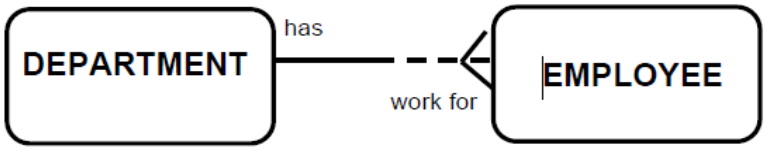
Proč se to učit?

- Může jedna osoba vlastnit více CD nebo pouze jedno? Může jedno CD vlastnit více osob?
- Tak jak dolaďujeme svůj model, chceme si být jisti, že vztahy mezi entitami řádně modelují pravidla naší činnosti. **Nezapomeňte!** Čím dříve budete promýšlet návrh ERD do detailů, tím lépe se vyhnete drahým chybám v budoucnosti.

Jedna ku více (1:N)

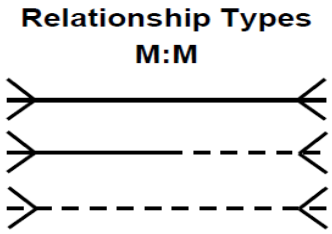
V ER modelu jsou nejčastější typy vztahů jedna ku více. Už jste viděli několik příkladů.

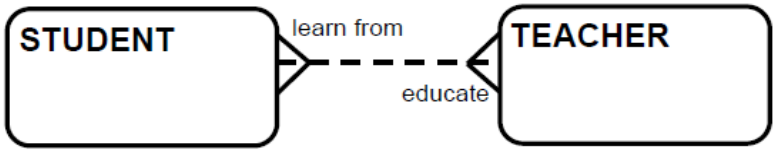




Více ku více (M:M), (M:N)

Různé typy M:N vztahu jsou běžné zvláště v prvních verzích ERD. V dalším stádiu procesu modelování se většina vztahů M:N (takřka všechny) rozloží.

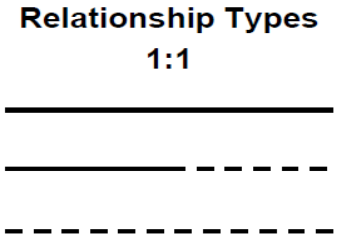




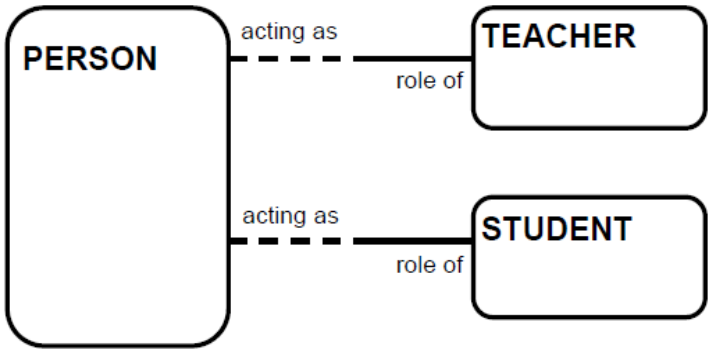
Jedna ku jedné 1:1

V každém ERD obvykle najdete pouze pár typů vztahu jedna ku jedné.

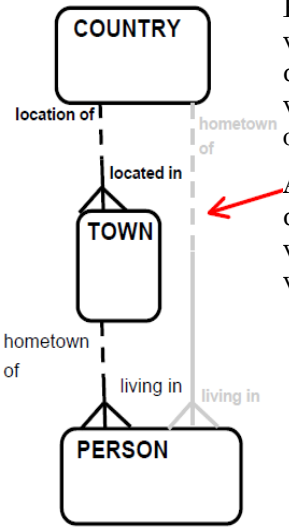
Obvykle se objevují na jednom konci vztahu.



1:1 vztah (všechny 3 varianty) se také objevují, když některé z entit reprezentují různá stádia procesu

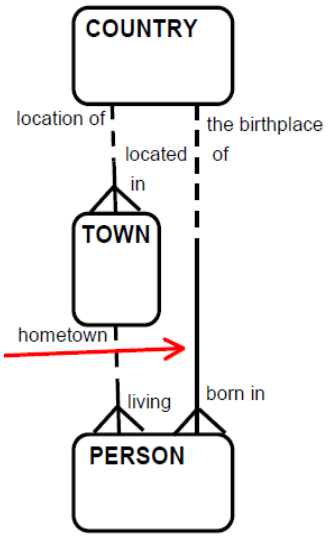


Redundantní vztah



Redundantní vztah můžeme odvodit z jiného vztahu v modelu. Příklad nalevo – můžete odvodit vztah osoby k zemi a tento vztah můžete odvodit ze dvou ostatních vztahů a mohli byste jej odstranit z modelu, tak jak je znázorněno vlevo.

Avšak, pozor na vyvozování, že vztah je redundantní pouze na struktuře. Pro kontrolu čtěte vztahy. ERD napravo neodráží redundantní vztah.



Řešení M:N vztahu

LEKCE 02 dd_S05_L03

V této lekci se naučíte:

- identifikovat atributy, které patří do M:N vztahu
- uvést kroky vedoucí k vyřešení M:N vztahu s použitím průnikové entity
- určit UID průnikové entity a aplikovat jej v entitě relačního diagramu

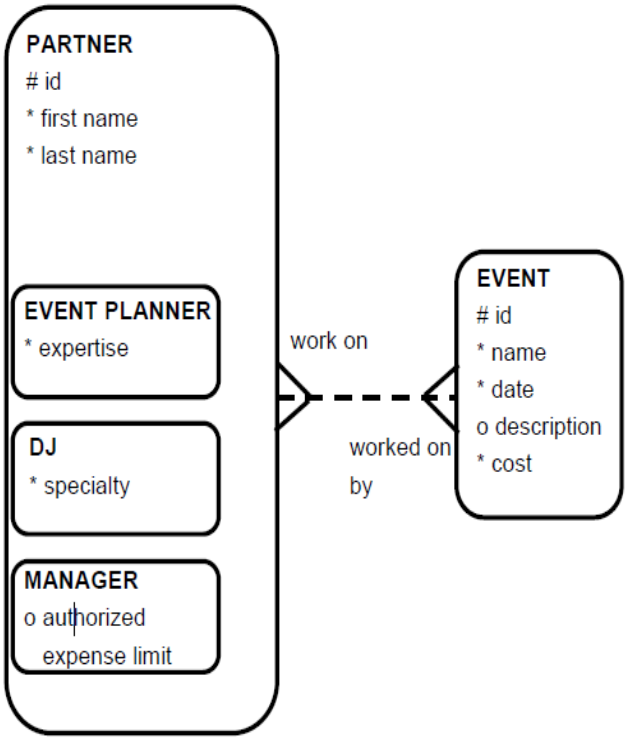
Proč se to učit?

- Tato lekce vám pomůže dokončit model – možná budete potřebovat vytvořit nové entity nebo nové vztahy založené na podnikovém zadání.
- Taktéž pomůže určit rozsah datového modelu – modelujete pouze důležité ze zadání.

Vztah skrývající atribut – skrytý atribut

Podle podnikového zadání DJ on Demand může být každému partnerovi určena práce na jedné nebo více akcích. Každá akce může být prací pro jednoho nebo více partnerů. Když EVENT_PLANNER nebo PROJEKT_MANAGER pracují na jedné akci, chceme, aby zaznamenávali STAV své práce.

Ke které entitě by patřil atribut STAV?

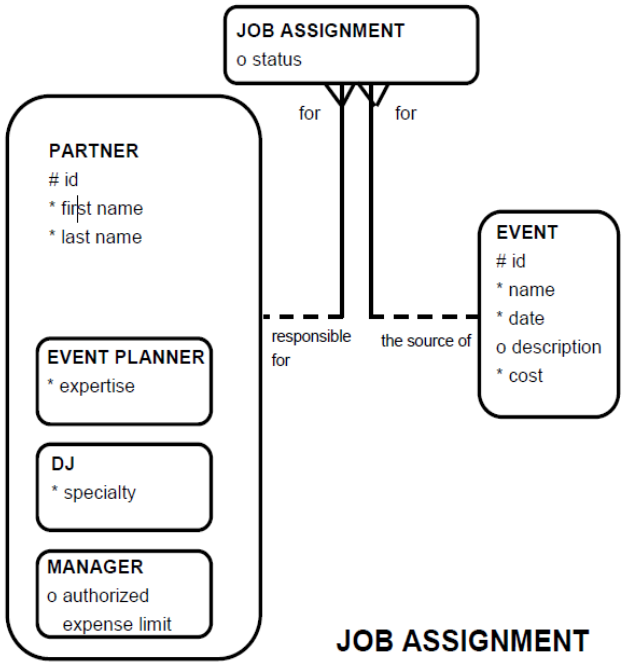


Řešení vztahu M:N

Pro vyřešení vztahu M:N potřebujeme třetí entitu. **Ta se nazývá „průniková“ entita.**

Průniková entita – JOB ASSIGNMENT (PRACOVNÍ ZAŘAZENÍ) – byla přidána včetně atributu stav. Z původního M:N vztahu vznikly dva 1:N vztahy.

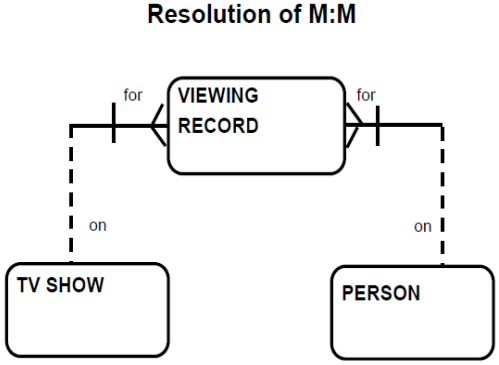
Který atribut by mohl být UID pro průnikovou entitu?



Jedinečný identifikátor (UID) často pochází z původních vztahů a je znázorněn mřížkou. V tomto případě se vztahy průnikové entity, vzniklé z původních entit, nazývají blokované.

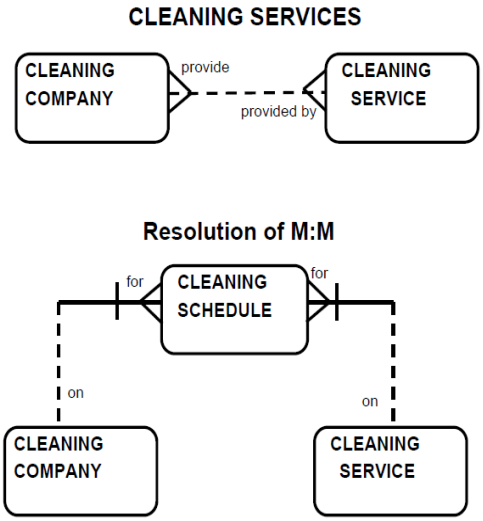
PŘÍKLAD ŘEŠENÍ M:M: POŘADY V TELEVIZI

Každý pořad může být sledován jednou nebo více osobami. Každá osoba může sledovat jeden nebo více pořadů.



PŘÍKLAD ŘEŠENÍ M:N: ÚKLIDOVÁ SLUŽBA

Každá společnost může poskytnou jednu nebo více úklidových služeb. Každá úklidová služba může být poskytnuta jednou nebo více společnostmi.



6. ODDÍL

Obsah oddílu

- Umělé, složené a sekundární UID (unikátní identifikátory)
- Normalizace databáze
- Normální formy (NF)

Umělé, složené a sekundární UID (unikátní identifikátory)

LEKCE 01 DD_S06_L01

V této lekci se naučíte:

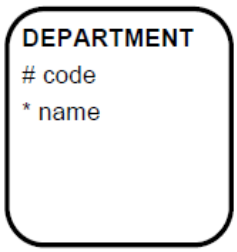
- definovat různé typy unikátních identifikátorů (UID)
- definovat kandidátní UID a vysvětlit, proč může mít entita někdy více než jeden kandidátní UID
- analyzovat obchodní pravidla a zvolit z kandidátů to nejvhodnější primární UID
- rozpoznat a diskutovat o otázkách identifikace v reálném světě

Proč se to učit?

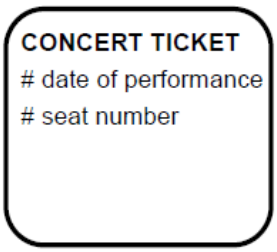
- Unikátní identifikátor (UID) je v relačních databázích velmi důležitý. Je to hodnota nebo kombinace hodnot, které umožní uživateli nalézt tuto unikátní položku mezi všemi ostatními. Identifikace toho pravého atributu, kombinace atributů a/nebo vztahů je dovednost, kterou si musí osvojit každý tvůrce databází. Unikátní identifikátor umožňuje najít svůj záznam v souboru, určitou kartu v balíčku karet, váš balíček ve skladu a konkrétní část dat v databázi.

Jednoduchý UID versus složený UID

UID, který má formu jediného atributu, označujeme jako jednoduchý UID. Někdy však jeden atribut nestačí k unikátní identifikaci instance entity. Pokud je UID kombinací atributů, říkáme mu složený UID.



Simple Unique Identifier



Composite Unique Identifier

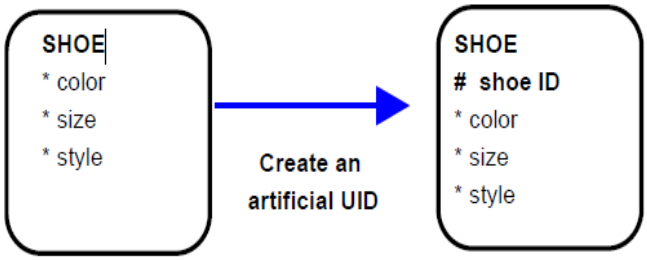
Umělý UID

Umělé UID jsou ty, které se nevyskytují v přirozeném světě, ale jsou vytvořeny pro účely identifikace v systému.

Lidé se nerodí s "čísly", ale mnoho systémů přiřazuje unikátní čísla pro identifikaci osob: ID studenta, ID zákazníka atd.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PŘÍKLAD

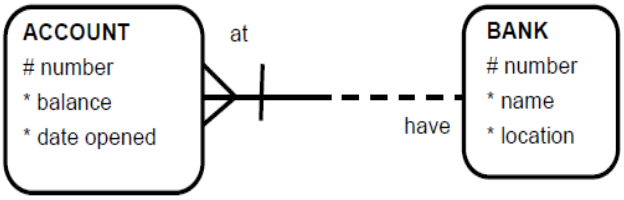


Bota má svou barvu, velikost, styl, ale žádné opravdu popisné "číslo". Obchod s obuví ale přiřadí unikátní čísla každému páru bot, takže je mohou jednoznačně identifikovat.

UID z blokovaných vztahů

Někdy je UID kombinace atributu a vztahu. Jaký je UID ÚČTU? Je umělý? Je složený?

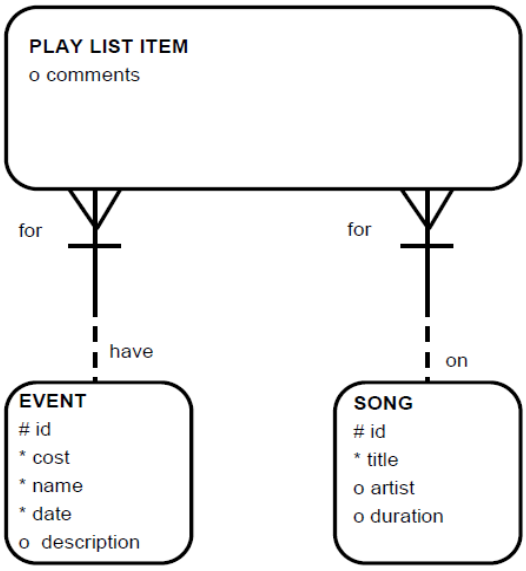
Dva lidé mohou mít stejné číslo bankovního účtu, ale u různých bank. Mezibankovní převody vždy vyžadují kód banky kromě čísla bankovního účtu.



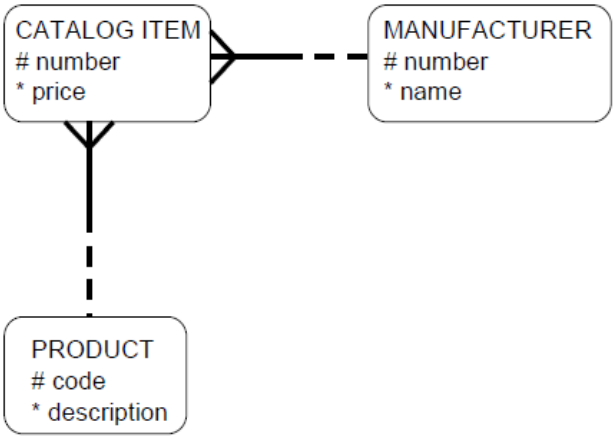
UID z blokovaných vztahů: průnikové entity

Jak jsme již viděli dříve, řešení M:M vztahu často vede k blokovaným vztahům od průsečíkové entity k těm původním.

V tomto příkladě vychází UID PLAY LIST ITEM z EVENT a SONG. Poznáte to podle bloků u vztahů.



Je možné, aby průsečíková entita použila umělý atribut jako UID místo blokovaných vztahů k původním entitám.



Každý VÝROBCE může vyrábět jeden nebo více PRODUKTŮ (boty, košile, džíny atd.). Každý PRODUKT může být vyroben jedním nebo více VÝROBCI (boty Nike, boty Adidas, džíny Levi´s atd.).

KATALOGOVÁ POLOŽKA řeší tento vztah typu více ku více. Položka v katalogu se dá jednoznačně identifikovat podle čísla výrobce a kódu produktu. Vztahy nejsou blokovány, protože se místo toho vytvořilo umělé UID – katalogové číslo

Kandidátské UID

Někdy je možný více než jeden UID.

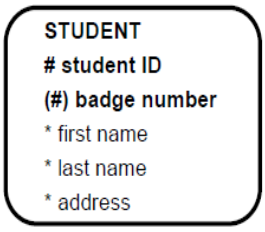
Například když si objednáte produkt z komerční webové stránky, obvykle vám přidělí unikátní kód zákazníka a také jste vyzváni k zadání vaší emailové adresy.

Každá z těchto položek vás jednoznačně identifikuje a dá se použít jako UID. Obě jsou tedy kandidátské UID.

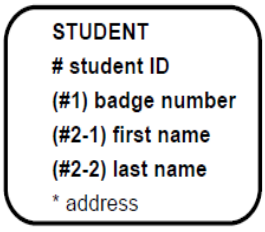
Pouze jeden z kandidátských UID se vybere jako skutečné UID. Tomu se říká primární UID. Ostatní kandidáti se nazývají sekundární UID.

ID studenta se zvolilo jako primární UID u obou těchto entit STUDENT.

První entita má jeden sekundární UID, druhá má dva sekundární UID (z nichž jeden je složený).



One Primary UID
One Secondary UID



One Primary UID
Two Secondary UIDs

Identifikace: databáze versus reálný svět

Unikátní identifikátory nám umožňují odlišit jednu instanci entity od jiné. Jak uvidíte později, stávají se primárními klíči v databázi. Primární klíč umožňuje přístup ke konkrétnímu záznamu v databázi.

V reálném světě ale není vždy tak snadné odlišit jednu věc od jiné.

Normalizace databáze

LEKCE 02 – 04 dd_S06_L02-04

Normalizace je proces organizace dat v databázi. Zahrnuje vytvoření tabulek, definici jejich struktur a nastavení vazeb mezi tabulkami podle pravidel ochrany dat a maximální flexibility databáze. Tato pravidla se snaží vyloučit dva faktory:

1. **redundance (vícenásobné uložení stejných dat)**

Několikanásobně uložené údaje jednak plýtvají prostorem na disku, jednak komplikují údržbu. Např. adresa firmy – lépe centrálně umístit v tabulce ADRESAR_FIREM než pokaždé znovu v evidenci objednávek, faktur, dodacích listů … ,

2. **nekonzistentní závislost dat**

Nekonzistentní závislostí dat rozumíme např. zaznamenávání platu zaměstnance v adresáři firem u firmy, ve které je zaměstnán. Plat zaměstnance je závislý na zaměstnanci a patří do tabulky zaměstnanců a ne do adresáře firem. Nekonzistentní závislost ztěžuje dostupnost k údajům.

Pro odstranění těchto faktorů bylo definováno několik pravidel. Každé pravidlo je nazýváno „**forma normalizace**“. O datech, která byla upravena daným pravidlem (formou normalizace), říkáme, že jsou v dané „**normální formě**“. Jsou-li dodrženy první tři formy normalizace, říkáme, že databáze je normalizována třetí formou normalizace nebo že databáze je ve třetí normální formě (3NF).

První forma normalizace (1NF)

Cíl: Neopakovat skupiny v tabulkách.

Jak?

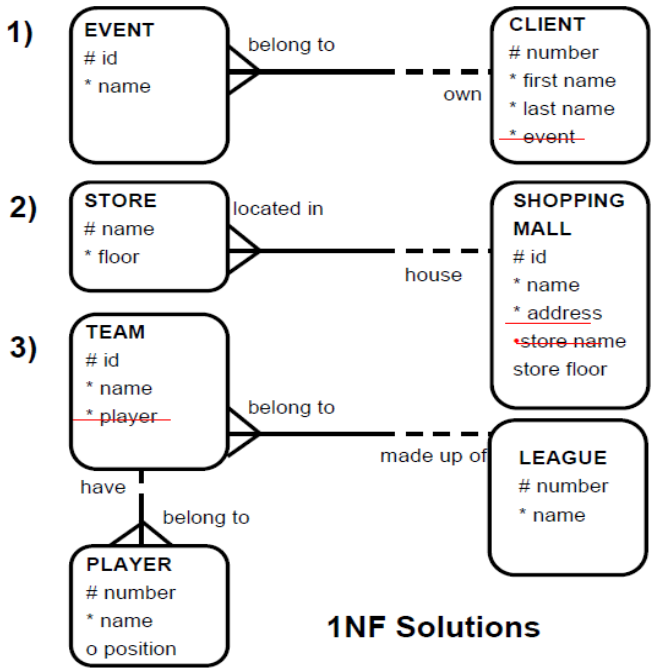
- zrušit opakující se skupiny dat v individuálních tabulkách
- vytvořit samostatnou tabulku pro každou množinu svázaných údajů
- pro každou samostatnou množinu definovat primární klíč
- nepoužívat více položek pro uložení stejných dat

PŘÍKLAD:

Odstranění vícehodnotového atributu *DETI(jmeno, RC):multi* u entitního typu ZAMESTANEC.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PŘÍKLAD



Pokud jsou všechny atributy v entitě jednoduché, pak říkáme, že entita je v 1NF.

Druhá forma normalizace (2NF)

Cíl: Vyřadit redundantní údaje.

Jak?

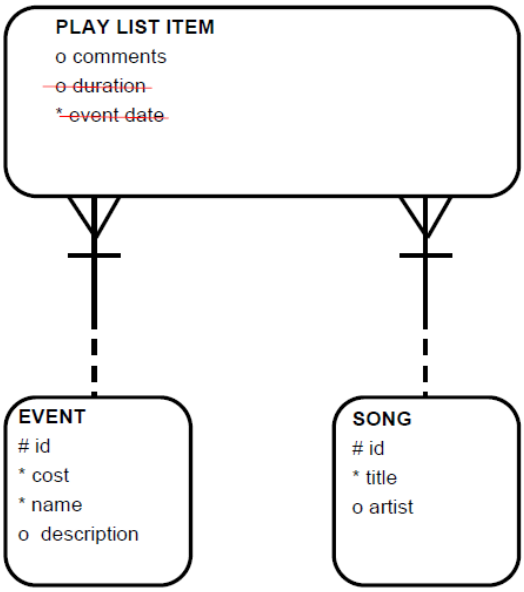
- vytvořit samostatnou tabulku pro množiny hodnot, které jsou použity ve více záznamech
- spojit tabulky pomocí klíčů

PŘÍKLAD:

Adresa firmy – nebude opakovaně uvedena v mnoha evidencích (objednávka, faktura …), ale bude uvedena v jediné tabulce "adresář firem" a připojena k jiné tabulce pomocí klíče.

PŘÍKLAD:

Pojďme se podívat na mírně upravený ERD podnikání DJ. Co je špatného na tomto diagramu? Odpověď: atributy doba trvání (duration) a datum události (event date) jsou chybně umístěné. Délka závisí pouze na SONG (píseň) a datum události závisí pouze na EVENT (událost).



Třetí forma normalizace (3NF)

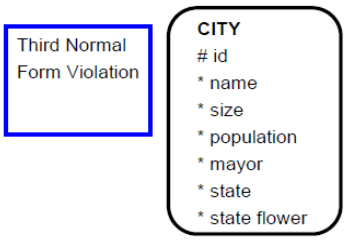
Cíl: Vyřadit data nezávislá na klíči.

Jak?

- vyřadit položky, které nezávisí na primárním klíči

PŘÍKLAD:

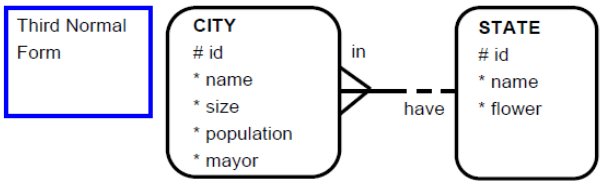
V evidenci zaměstnanců (tab. ZAMESTANCI) potřebuji evidovat informace o univerzitě, na které získal zaměstnanec vzdělání. Nejenom název univerzity, ale i adresu. Adresa univerzity je již závislá na univerzitě a ne na pracovníkovi, proto vytvořím samostatnou tabulku UNIVERZITY a spojím se ZAMESTNANCI pomocí klíče.



PŘÍKLAD

Přemýšlejte o systému, který sleduje informace o městech – velikost, počet obyvatel, starosta atd. První model ukazuje entitu CITY, která obsahuje informace o zemi (state). Ačkoli je ZEMĚ atributem města, státní symbol je opravdu atributem země.

Druhý model, s novou entitou STATE (země), je ve třetí normální formě.



7. ODDÍL

Obsah oddílu

- Hierarchické a rekurzivní vztahy
- Modelování historických dat

Hierarchické a rekurzivní vztahy

LEKCE 02 dd_S07_L02

V této lekci se naučíte:

- definovat a uvést příklad hierarchického vztahu
- identifikovat UID v hierarchickém modelu
- definovat a uvést příklad rekurzivního vztahu
- znázornit rekurzivní vztah v ERD podle podnikového scénáře
- sestavit model s využitím rekurzí a hierarchií s cílem vyjádřit stejný konceptuální význam

Proč se to učit?

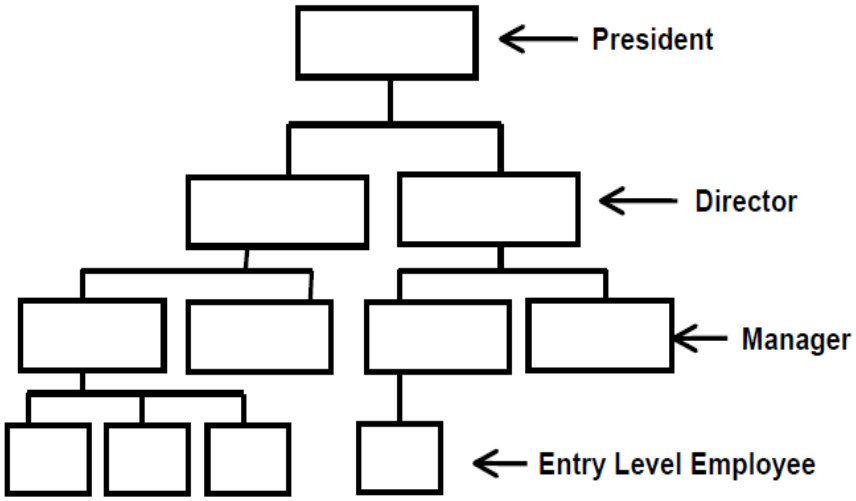
Role jsou často organizovány podle hierarchie – v práci (manažer, vedoucí skupiny, úředník, úředník, pracovník, který připravuje pokrmy) nebo ve škole (ředitel, zástupce ředitele, učitelé, další zaměstnanci). Hierarchická data jsou velmi častá. Jejich pochopení vám pomůže lépe modelovat:

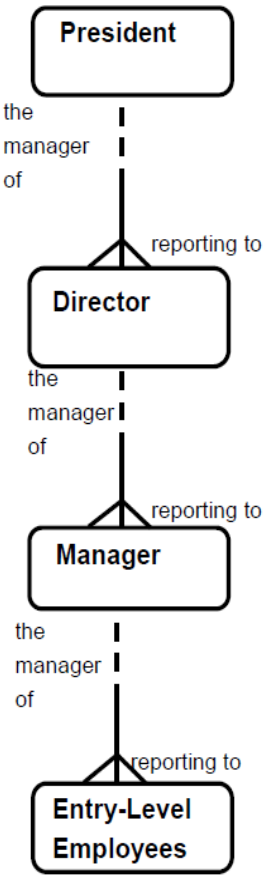
- organizační schémata firmy,
- stavební konstrukce,
- rodinné stromy,
- … a mnoho dalších hierarchií, které nalezneme v reálném světě.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

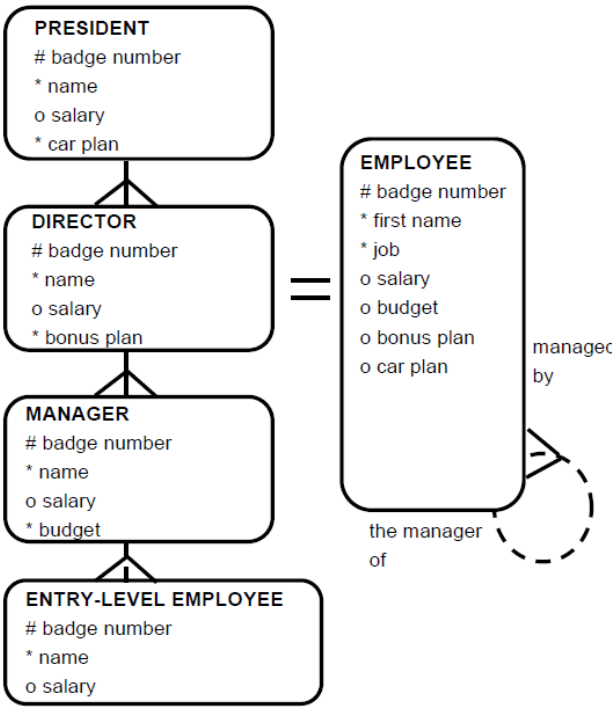
Vztahy v organizačním schématu

Organizační schéma můžeme vyjádřit tímto datovým modelem. Jaké UID jsou zde pro každou entitu?





Hierarchie versus rekurzivní vztah



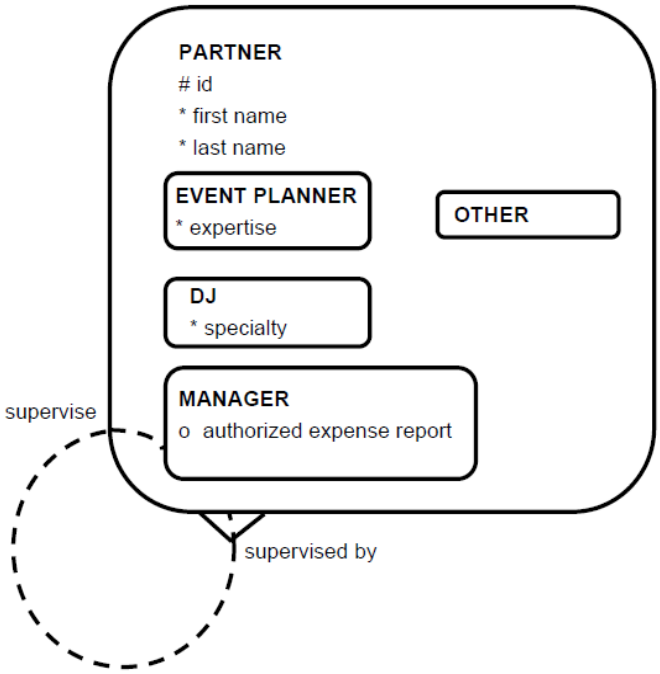
Oba tyto modely reprezentují všechny zaměstnance.

Ten vlevo představuje **hierarchickou strukturu**. Ten vpravo pracuje s **rekurzivním vztahem**.

Který z nich je podle vás lepší?

PŘÍKLAD: DJS ON DEMAND (PODNIKOVÝ SCÉNÁŘ)

V modelu DJs má manažer projektu celkovou odpovědnost za událost a řídí ostatní zaměstnance (plánovač událostí, DJ) pracující na dané události. Rozhodli jsme se zobrazit tuto hierarchii pomocí rekurzivního vztahu.



PŘÍKLAD: OBCHODNÍ SCÉNÁŘ PRO AUTOMOBILOVOU VÝROBU

U výrobce automobilů považujeme všechny základní díly, montážní podčásti a části, sestavy a produkty za instance entity nazvané KOMPONENT. Model můžeme vytvořit jako jednoduchý rekurzivní vztah.

Namodelujte data z "kusovníku" jako rekurzivní vztah typu více ku více:

- každý komponent může být součástí jednoho nebo více KOMPONENTŮ,
- každý komponent se může skládat z jednoho nebo více KOMPONENTŮ.

Modelování historických dat

LEKCE 02 dd_S07_L03

V této lekci se naučíte:

- identifikovat potřebu sledovat data, která se mění v čase
- sestavit modely ERD, které obsahují elementy "dat v čase"
- identifikovat UID entity, která ukládá historická data, a vysvětlit a zdůvodnit výběr UID
- sestavit konceptuální model založený na daném obchodním scénáři
- aplikovat pravidla pro tvorbu E-R diagramů (entita-vztah) a vytvořit tak ERD, který odráží obchodní pravidla

evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- prezentovat a interpretovat datový model posluchačům
- vytvořit písemnou dokumentaci doprovázející ústní prezentaci a ERD

Proč se to učit?

- Kolik jsi měřil/a v 5ti letech? Kolik jsi měřil/a v 10ti letech?
- Kolik měříš nyní? Pokud vaši rodiče tyto míry během vašeho dětství zapisovali, zaznamenávali tím historická data. Většina podniků potřebuje sledovat některá historická data. Pomáhá jim to identifikovat trendy a vzorce, které mohou být základem pro obchodní inovace nebo zlepšení procesů.
- Historie výpůjček filmů je užitečná pro prodejnu filmů. Vedoucí prodejny může zjistit, které filmy jsou populární a které by měly být přesunuty do zadních regálů.
- Napadá vás použití pro farmaceutickou společnost, potraviny nebo pekárnu nebo továrnu na zpracování mořských plodů?
- Jaká historická data budou chtít sledovat a proč?

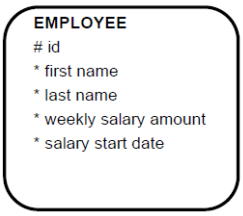
Kdy je to nutné modelovat data v průběhu času?

Zeptejte se svého klienta:

- Je nutný záznam auditu?
- Mohou se hodnoty atributů měnit v čase?
- Mohou se vztahy měnit v čase?
- Potřebujete generovat sestavy ze starších dat?
- Potřebujete uchovávat předchozí verze dat? Pokud ano, po jakou dobu?

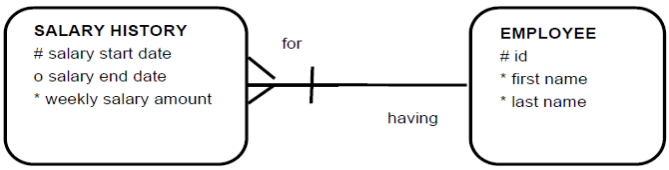
Organizace musí uchovávat údaje o platech zaměstnanců. Všichni zaměstnanci jsou vypláceni týdně.

Původně byla namodelována následující entita ZAMĚSTNANEC.



Dodatečné požadavky upřesňují, že organizace potřebuje uchovávat historická data o tom, jak a kdy se změnily platy zaměstnanců během jejich pracovního poměru.

Chcete-li modelovat změny platu v průběhu času, přidejte entitu SALARY HISTORY (HISTORIE PLATU).



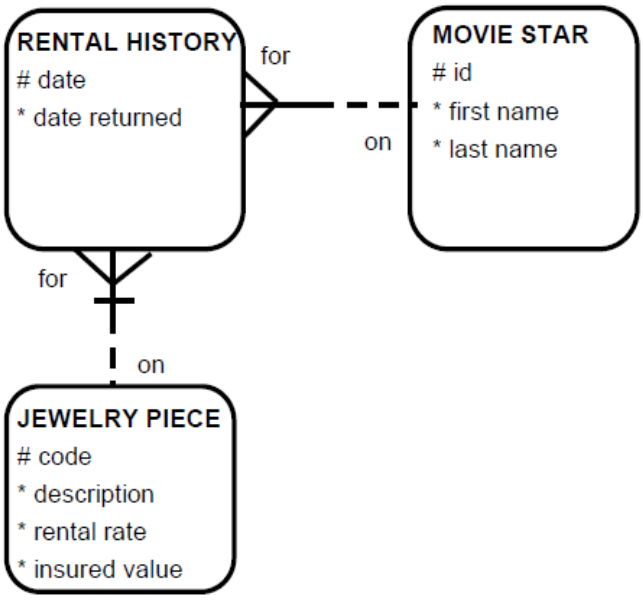
evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

UID entity HISTORIE PLATU je související ID ZAMĚSTNANCE a datum zahájení výplaty mzdy.

PŘÍKLAD

Klenotnictví půjčuje filmovým hvězdám šperky (náhrdelníky, náramky atd.) na zvláštní příležitosti, jako je udílení cen nebo premiéry filmů. Rádi by sledovali historii pronájmů jednotlivých šperků. Následující ER model sleduje pouze aktuálního nájemce daného šperku.

Jak byste vztah upravili, aby se sledovala historie?



Vztah mezi ŠPERK a FILMOVÁ HVĚZDA by se měl změnit na vztah M:M, který pak řeší průsečíková entita HISTORIE PRONÁJMU.

Jaký UID má HISTORIE PRONÁJMU?

UID HISTORIE PRONÁJMU je datum pronájmu a UID ŠPERKu (znázorněno blokovaným vztahem).

8. ODDÍL

Konvence pro tvorbu diagramů pro lepší čitelnost

LEKCE 01 dd_S10_L01

V této lekci se naučíte:

- aplikovat pravidla Oracle pro tvorbu diagramů datových modelů
- identifikovat velkoobjemové entity v diagramu datového modelu a vysvětlit jejich významu pro firmu
- Překreslit daný diagram datového modelu a zvýšit tak jeho přehlednost a čitelnost
- Rozpoznat užitečnost dělení složitého ERD do více funkčních subdiagramů

Proč se to učit?

- Co kdyby si všichni výrobci obuvi zavedli své vlastní velikosti?
- Co kdyby každý architekt používal jiný systém k zákresu plánu budovy?
- Dodržování stejných konvencí usnadňuje práci v týmu. Jak se ERD zvětšuje a komplikuje, je stále těžší vyjádřit jej v jasném a čitelném formátu.

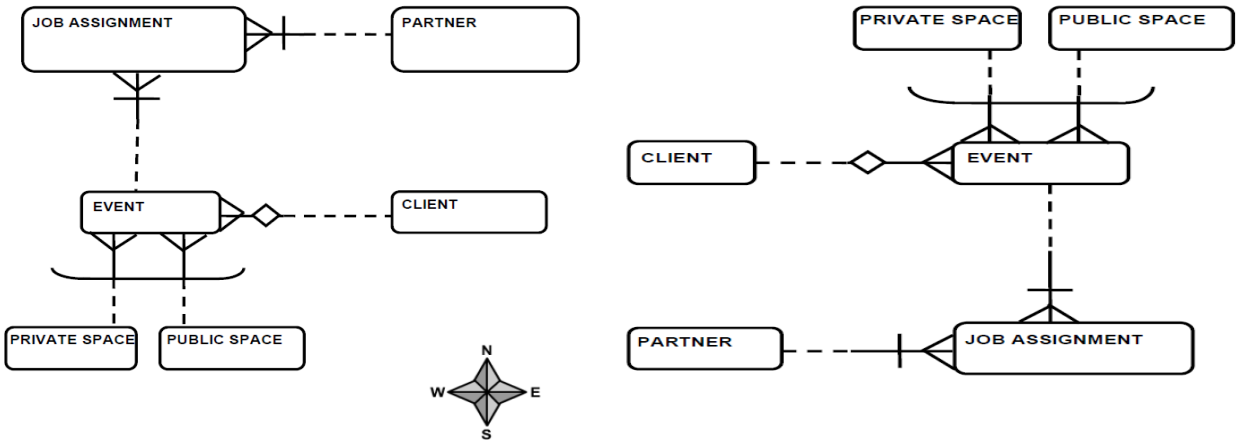
Dodržováním konvence, že "vrány létají k jihu a na východ" se velkoobjemové entity dostanou do levé horní části ERD.

Velkoobjemové entity jsou ty, které mají největší počet instancí ve srovnání s jinými entitami.

Dodržováním konvence, že "vrány létají na sever a na západ" se velkoobjemové entity dostanou do spodní pravé části ERD.

Velkoobjemové entity jsou často "centrální" nebo významnější entity v ERD. Budou mít nejvyšší počet vztahů k jiným entitám a většina obchodních funkcí bude mít vliv na data uložená v této entitě.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Často budete mít mix podle velikosti prostoru a vlastních preferencí.

Jasnost a čitelnost jsou hlavní kritéria.

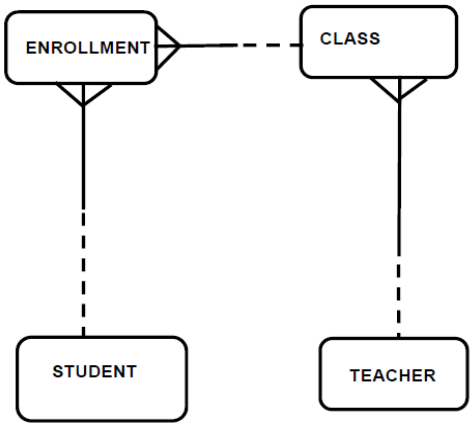
Velkoobjemové entity nejsou vždy ty nejdůležitější.

Která z těchto entit bude mít nejvyšší objem?

Která entita je nejdůležitější?

Opět platí, že jasnost a čitelnost jsou hlavní kritéria.

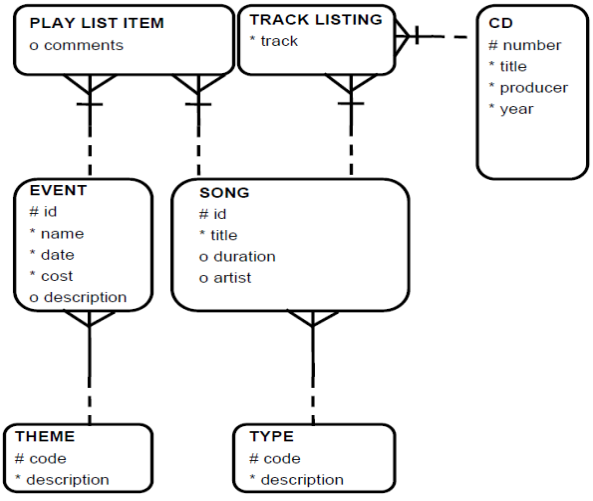
Čitelnost zabírá prostor a podléhá vkusu. Použití bílého prostoru pomáhá vyjasnit ERD.



Když máte velmi velký diagram, může pomoci rozdělit jej na menší diagramy funkčně souvisejících entit. Můžete použít menší subdiagramy pro prezentaci pro různé skupiny v rámci organizace zákazníka.

Často se stává, že více vývojářů vytváří aplikace, které používají stejnou databázi. Každý aplikační vývojář by mohl použít menší diagram, který obsahuje ty entity, pro které bude vytvářet obrazovky, formuláře a sestavy.

Toto jsou entity, které budou nejvíce zajímat DJs nebo někoho, kdo pro ně vytváří aplikaci.



9. ODDÍL (OA S11)

Obsah oddílu

- Úvod do relačních databází
- Základní mapování: Proces transformace ERD do RMD
- Mapování vztahů

Úvod do relačních databází

LEKCE 01 DD_S11_01

V této lekci se naučíte:

- definovat primární klíč
- definovat cizí klíč
- definovat pravidlo sloupcové integrity
- definovat řádek, sloupec, primární klíč, unikátní klíč a cizí klíč podle diagramu tabulky obsahující tyto elementy
- identifikovat porušení pravidel integrity dat

Proč se to učit?

- Konceptuální datový model bude transformován do relační databáze. To znamená, že naše entity, atributy, vztahy a unikátní identifikátory budou převedeny do objektů v relační databázi.
- Proto je nutné pochopit strukturu těchto objektů.

Relační databáze je databáze, kterou uživatel vnímá jako soubor dvourozměrných tabulek. Níže uvedená tabulka obsahuje údaje zaměstnanců.

EMPLOYEES (table name)

Row →

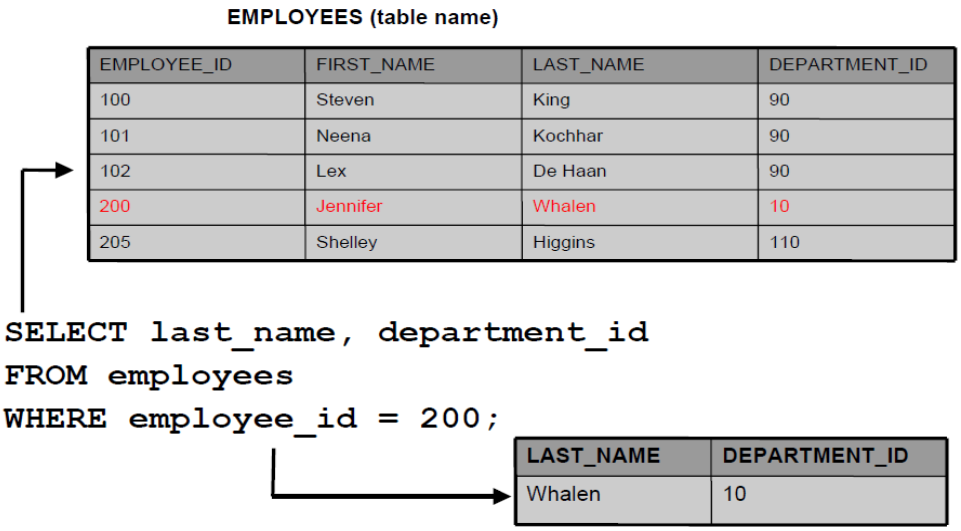
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
200	Jennifer	Whalen	10
205	Shelley	Higgins	110

↑ Column

Strukturovaný dotazovací jazyk nám umožňuje, abychom se efektivním způsobem dostali k datům v relačních databázích. Místo abychom procházeli každý řádek a hledali záznam o zaměstnanci 200, použijeme následující příkaz

```
SELECT last_name, department_id
FROM employees
WHERE employee_id = 200;
```

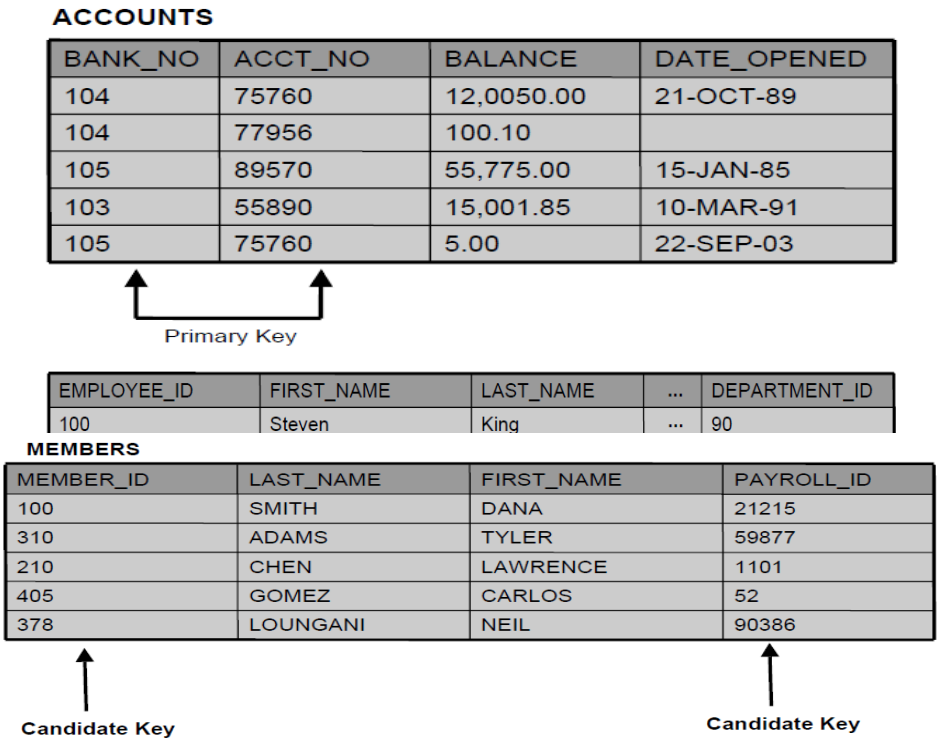
Výsledek příkazu:



Primární klíč

Primární klíč (dále PK) je buď sloupec, nebo množina sloupců, které jednoznačně identifikují každý řádek v tabulce.

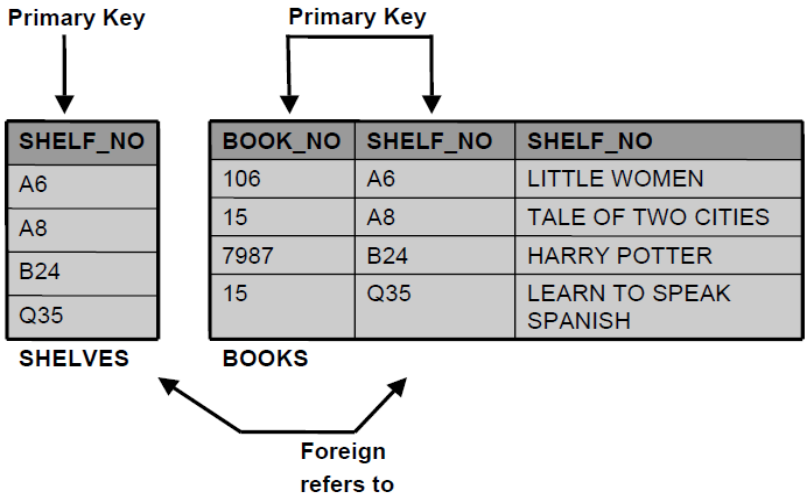
Každá tabulka by měla mít primární klíč a ten musí být jedinečný. Žádná část primárního klíče nesmí být prázdná (null).



Může mít více než jeden sloupec nebo kombinaci sloupců, které by mohly být použity jako primární klíč tabulky. Každý z nich se nazývá „kandidátní klíč“.

Cizí klíč

Cizí klíč je buď sloupec, nebo množina sloupců v jedné tabulce, které se vztahují k primárnímu klíči téže nebo jiné tabulky.



Pokud je cizí klíč součástí primárního klíče, nesmí být prázdný.

Sloupcová integrita

Sloupec musí obsahovat pouze konzistentní hodnoty s definovaným formátem sloupce

ACCOUNTS

BANK_NO	ACCT_NO	BALANCE	DATE_OPENED
104	75760	12,0050.00	21-OCT-89
104	77956	100.10	
105	89570	55,775.00	15-JAN-85
103	55890	15,001.85	10-MAR-91
105	75760	5.00	22-SEP-03

ACCOUNTS Table Definition

Column Name	Data Type	Optionality
BANK_NO	Number (5)	Not null
ACCT_NO	Number (8)	Not null
BALANCE	Number (12,2)	Not null
DATE_OPENED	Date	

Shrnutí pravidel datové integrity:

Pravidla datové integrity, taktéž známá jako omezení (constraints), definují relačně korektní stav databáze.

Pravidla datové integrity zajišťují, že uživatel může provádět pouze ty operace, které zachovávají databázi v konzistentním stavu.

Typ omezení	Vysvětlení	Příklad
Integrita entit	Primární klíč musí být jedinečný a žádná část primárního klíče nemůže být prázdná (null)	V tabulce EMPLOYEES nemůže být sloupec emp_no prázdný.
Referenční integrita	Cizí klíč musí patřit k hodnotě existujícího primárního klíče (nebo musí být prázdný).	Hodnota ve sloupci dept_no tabulky EMPLOYEES musí odpovídat k hodnotě sloupce dept_no v tabulce DEPARTMENTS.
Sloupcová integrita	Sloupec musí obsahovat pouze hodnoty v souladu s definovaným formátem dat ve sloupci.	Hodnota ve sloupci zůstatek tabulky ACCOUNT musí být číslo.
Uživatelem definovaná integrita	Údaje uložené v databázi musí být v souladu s pravidly zadání.	Pokud je hodnota ve sloupci zůstatek tabulky ACCOUNT nižší než 1,00, musíme poslat dopis vlastníkovi účtu (bude potřeba pravidlo zvlášť naprogramovat).

Základní mapování:
Proces transformace ERD do RMD

LEKCE 02 DD_S11_L02

V této lekci se naučíte:

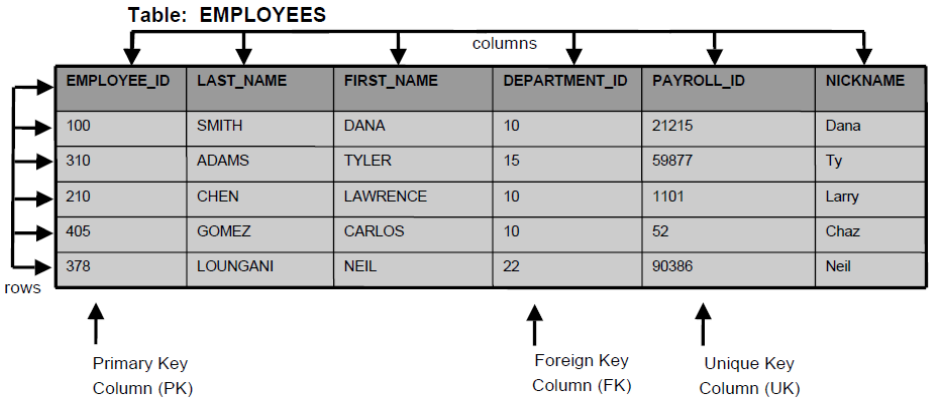
- rozlišovat modely typu E-R (entita-vztah) od databázových modelů
- popsat mapování terminologie mezi konceptuálním modelem a modelem relační databáze
- pochopit a aplikovat pojmenovávací konvence Oracle pro tabulky a sloupce použitých v relačních modelech
- transformovat entitu do tabulkového diagramu

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Proč se to učit?

- Když navrhujete dům, rádi byste jej nakonec viděli postavený. Dokonce i když vy sám nestavíte, budete muset pochopit terminologii používanou stavaři, abyste jim mohl pomoci převést váš návrh do reality.
- Původní návrh databáze lze použít pro další diskusi mezi designéry, správci databází a vývojáři aplikací.

Opakování relačních tabulek:



Tabulka je jednoduchá struktura, ve které jsou organizována a ukládána data. Ve výše uvedeném příkladu se tabulka ZAMĚTNANCI (EMPLOYEES) používá pro ukládání dat o zaměstnancích.

Tabulky mají sloupce a řádky. V příkladu každý řádek popisuje zaměstnance. Každý sloupec slouží k ukládání specifických hodnot, jako je číslo zaměstnance, příjmení, jméno.

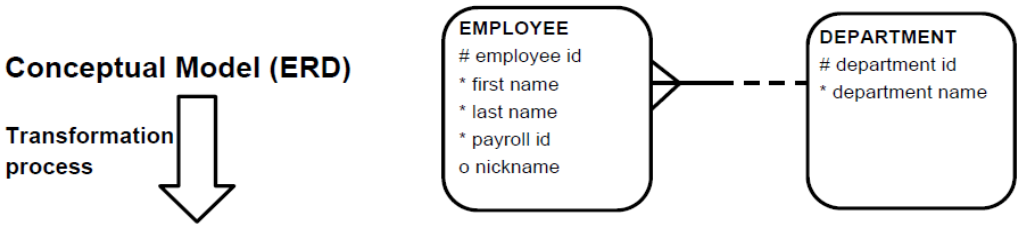
Sloupec employee_id je primární klíč. Každý zaměstnanec má v této tabulce jednoznačné identifikační číslo. Hodnota ve sloupci primárního klíče od sebe odlišuje jednotlivé řádky.

Payroll_id je jednoznačný (unikátní) klíč. To znamená, že systém nedovoluje dvěma řádkům mít stejné payroll_id.

Sloupec cizího klíče odkazuje na řádek v jiné tabulce. V příkladu číslo oddělení odkazuje na řádek v tabulce oddělení.

V tomto případě víme, že Dana Smith pracuje v oddělení 10. Kdybychom chtěli zjistit více o oddělení Dany Smith, podívali bychom se v tabulce oddělení na řádek, který má číslo oddělení 10.

Konceptuální model ERD je transformován do fyzického modelu. Fyzická implementace bude relační databáze.



Physical Implementation: Relational Database

EMPLOYEES (EPE)

Key type	Optionality	Column name
pk	*	employee_id
uk	*	payroll_id
	*	last_name
	*	first_name
	*	nickname
fk	*	department_id

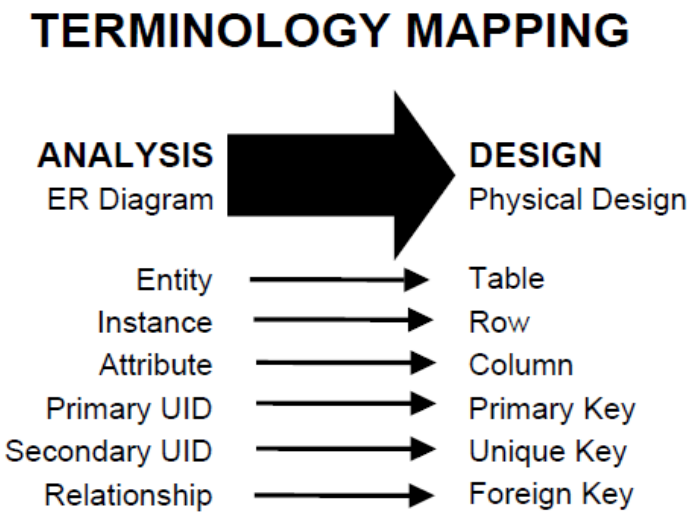
DEPARTMENTS (DPT)

Key type	Optionality	Column name
pk	*	department_id
	*	department_name

Terminologie mapování

Přechod od analýzy (konceptuální model) k návrhu (fyzická realizace) také znamená změnu terminologie:

- entita se stává tabulkou,
- instance se stává řádkem,
- atribut se stává sloupcem,
- primární jedinečný identifikátor se stává primárním klíčem,
- sekundární jedinečný identifikátor se stává unikátním klíčem,
- vztah se transformuje do sloupce s cizím klíčem a omezení cizího klíče

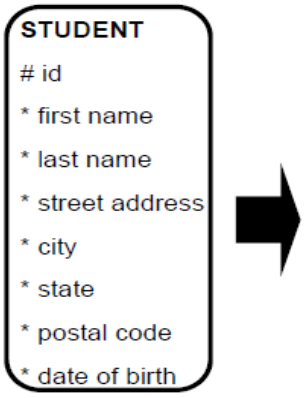


Notace tabulky

Sloupec "typ klíče" by měl obsahovat hodnoty "pk" pro primární klíč, "uk" pro unikátní klíč a "fk" pro cizí klíč. Pokud tento sloupec není součástí žádného klíče, bude prázdný.

Volitelnost musí obsahovat „*“, pokud jsou sloupce povinné, a „o“, pokud jsou nepovinné. Je to stejné jako v diagramu ERD.

Třetí sloupec je určen pro jméno sloupce.



STUDENTS

Key Type	Optionality	Column Name
pk	*	id
	*	first_name
	*	last_name
	*	str_addr
	*	city
	*	state
	*	p_code
	*	dob

Pravidla pro jména tabulek a sloupců

Název tabulky je množné číslo entity; příklad: Student se stává Studenti. Názvy sloupců jsou shodné s názvy atributů, kromě toho, že speciální znaky a mezery jsou nahrazeny podtržítky. Názvy sloupců bývají častěji zkratky než názvy atributů.

PŘÍKLAD:

First name (křestní jméno) se stává firtst_name nebo fname.

Konvence pro jména: krátká jména:

Unikátní krátké jméno pro každou tabulku je užitečné pro pojmenovávání sloupců s cizím klíčem.

Zkratky tvoříme podle následujících pravidel:

U názvů entity, skládající se z více než jednoho slova, vezměte:

- první znak prvního slova,
- první znak druhého slova,
- poslední znak posledního slova.

PŘÍKLAD

Pracovní pozice (JOB ASSIGNMENT) má zkratku JAT.

esf evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

U názvů entity, která se skládá z víceslabičného slova, vezměte:

- první znak první slabiky,
- první znak druhé slabiky,
- poslední znak poslední slabiky.

PŘÍKLAD

Zaměstnanec (EMPLOYEE) – EPM.

U jednoslabičných názvů entit s více než jedním znakem použijte :

- první znak,
- druhý znak,
- poslední znak.

Omezení v pojmenovávání v ORACLE

Názvy tabulek a sloupců:

- musí začínat písmenem,
- mohou obsahovat až 30 alfanumerických znaků,
- nemohou obsahovat mezery nebo speciální znaky jako „!“, ale $, # a „_“ jsou dovoleny
- názvy tabulek musí být unikátní v rámci jednoho uživatelského účtu v databázi ORACLE
- názvy sloupců musí být unikátní v rámci tabulky.

V ORACLE databázi a v jazyce SQL mají některá slova speciální význam. Říká se jim „rezervovaná“ (klíčová). Nepoužívejte tato slova pro názvy tabulek a sloupců. Zde jsou uvedeny některé příklady rezervovaných slov ORACLE:

- NUMBER,
- SEQUENCE,
- ORDER,
- VALUES,
- LEVEL,
- TYPE.

Kompletní seznam najdete na Technet. (otn.oracle.com)

Mapování vztahů:

LEKCE 03 dd_S11_L03

V této lekci se naučíte:

- aplikovat pravidlo mapování vztahů na správnou transformaci 1:M a „blokovaných („barred“) vztahů
- aplikovat pravidlo mapování vztahů na správnou transformaci M:M
- transformovat vztahy 1:1

Proč se to učit?

- Co kdybyste někomu stavěli dům? Koupíte veškerý materiál – dřevo, barvu, dveře, hřebíky … . Ale nevíte, jak poskládat jednotlivé části. Nevíte, kolik tam má být pokojů, kde by měla být okna, jak mají být orientovány dveře a jakou barvou vymalovat. Nějak byste dům postavili, ale pokud nemáte plán, který popisuje, jak jednotlivé části složit dohromady, finální produkt nemusí být tím domem, který měl zákazník na mysli.
- Vztahy jsou mapovány do cizích klíčů, které tabulkám umožňují se vzájemně provázat. Cizí klíče umožňují uživateli přístup k potřebným informacím v jiných tabulkách. Pokud nebudeme mapovat vztahy, budeme mít spoustu samostatných tabulek s informacemi, které nemůže využívat zbytek databáze.
- Stejně jako kopie slouží jako návrh domu. Mapování relací do struktur relační databáze je částí tvorby „first-cut“ návrhu databáze, který bude sloužit jako základ pro následnou diskuzi mezi analytiky, vývojáři a databázovými administrátory.

Pravidla vztahů:

Vztahy vytvářejí jeden nebo více cizích klíčů v tabulce.

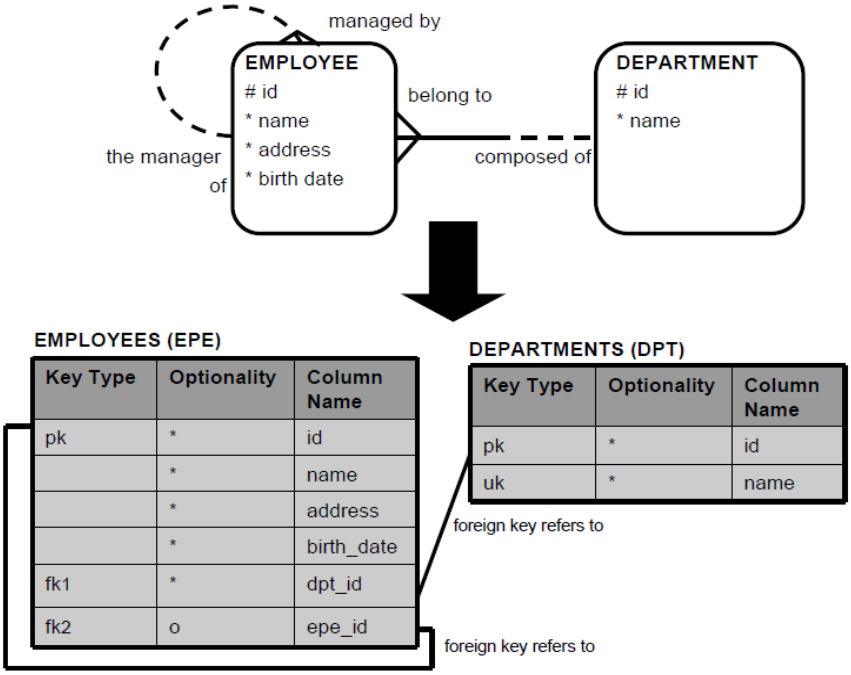
Do názvu cizího klíče používáme zkratku tabulky. Například sloupec cizího klíče v tabulce zaměstnanci je dpt_id pro vztah s oddělením a epe_id pro rekurzivní vazbu.

Cizí klíč je povinný nebo nepovinný podle toho, zda je vztah povinný nebo nepovinný.

evropský sociální fond v ČR EVROPSKÁ UNIE MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PŘÍKLAD

Vztahy mezi entitami ZAMĚSTNANEC a ODDĚLENÍ (v příkladu je dpt_id povinný a epe_id volitelný).



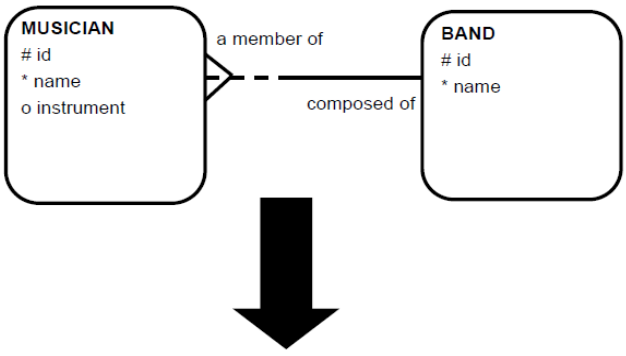
Key Type	Optionality	Column Name
pk	*	id
	*	name
	*	address
	*	birth_date
fk1	*	dpt_id
fk2	o	epe_id

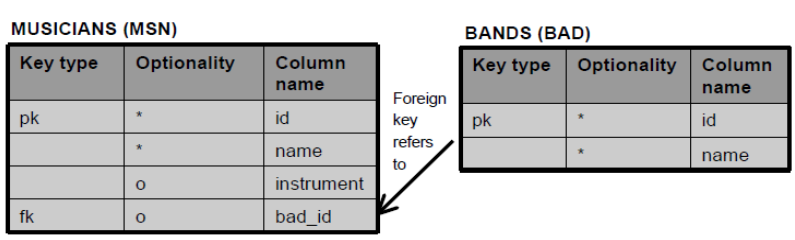
Key Type	Optionality	Column Name
pk	*	id
uk	*	name

Mapování povinného vztahu na jedné straně

Vztahy, které jsou povinné na jedné straně nebo na obou stranách, jsou mapovány úplně stejně jako vztah, který je volitelný na jedné straně. ERD je dostatečně bohatý na to, aby zachytil členství ve vztahu na obou koncích vztahu. Nicméně relační model je omezen v tom, že omezení cizího klíče vynucuje povinný vztah pouze pro konec „many“.

V následujícím příkladu si relační model nemůže vynutit, aby BAND (Skupina) musela obsahovat nejméně jednoho hudebníka. Volitelnost na jednom konci musí být implementována pomocí dalšího programu.





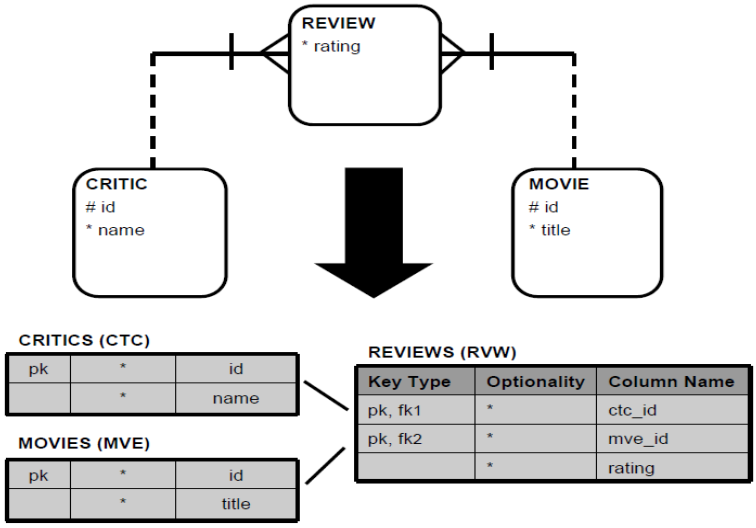
Key type	Optionality	Column name
pk	*	id
	*	name
	o	instrument
fk	o	bad_id

Key type	Optionality	Column name
pk	*	id
	*	name

Mapování M:N vztahu

M:N vztah se řeší pomocí průnikové entity, která se mapuje do průnikové tabulky. Tato tabulka bude obsahovat sloupce cizí klíče, které se budou odkazovat na původní tabulky.

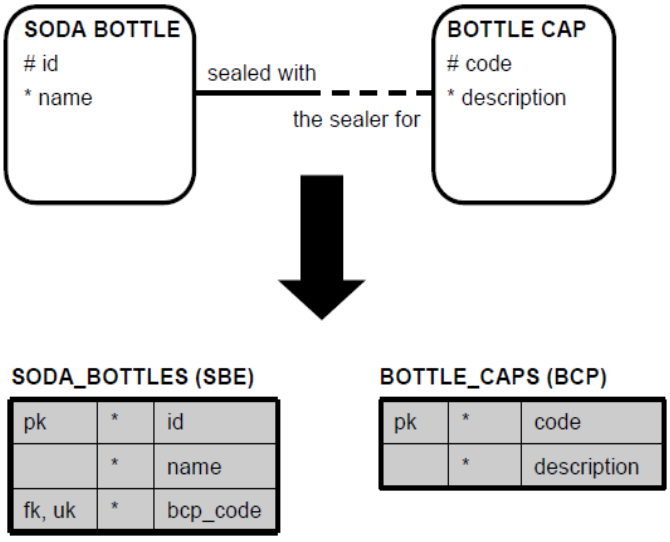
V následujícím příkladu REWIES obsahují všechny kombinace, které existují mezi kritikou a filmem.



Mapování 1:1 vztahu

Při transformaci vztahu 1:1 tvoříte cizí a unikátní klíč. Všechny sloupce cizího klíče jsou také součástí unikátního klíče.

Pokud je vztah povinný z jedné strany, cizí klíč se vytvoří v odpovídající tabulce. Např. bcp_code je sloupec cizího klíče SODA_BOTTLES (nealkoholické nápoje), který odkazuje na primární klíč tabulky BOTTLE_CAPS (víčka na nápoje). Bcp_code by byl také unikátním klíčem v rámci tabulky SODA_BOTTLES.



Mapování vztahu 1:1 – nepovinný z obou stran

Jestliže je vztah nepovinný z obou stran, můžete si vybrat, která tabulka získá cizí klíč. Zde neexistují žádná absolutní pravidla, ale nabízíme následující návod:

- implementujte cizí klíč do tabulky s méně řádky, abyste ušetřili prostor,
- implementujte cizí klíč, kde je to v rámci zadání nejlogičtější.

V příkladu: Autopůjčovna by se více zajímala o auta než o parkovací místo, takže je logičtější vložit cizí klíč do tabulky CARS. Nicméně u firmy, provozující parkoviště, je hlavním objektem parkovací místo. Proto by bylo smysluplnější dát cizí klíč do tabulky SPACES.

Mapování 1:1 – povinné na obou stranách

Jestliže je vztah povinný na obou koncích, existují v databázi stejné pravidla, jako vztah M:1, který je povinný z jedné strany. A proto budeme potřebovat další kód pro realizaci tohoto vztahu.