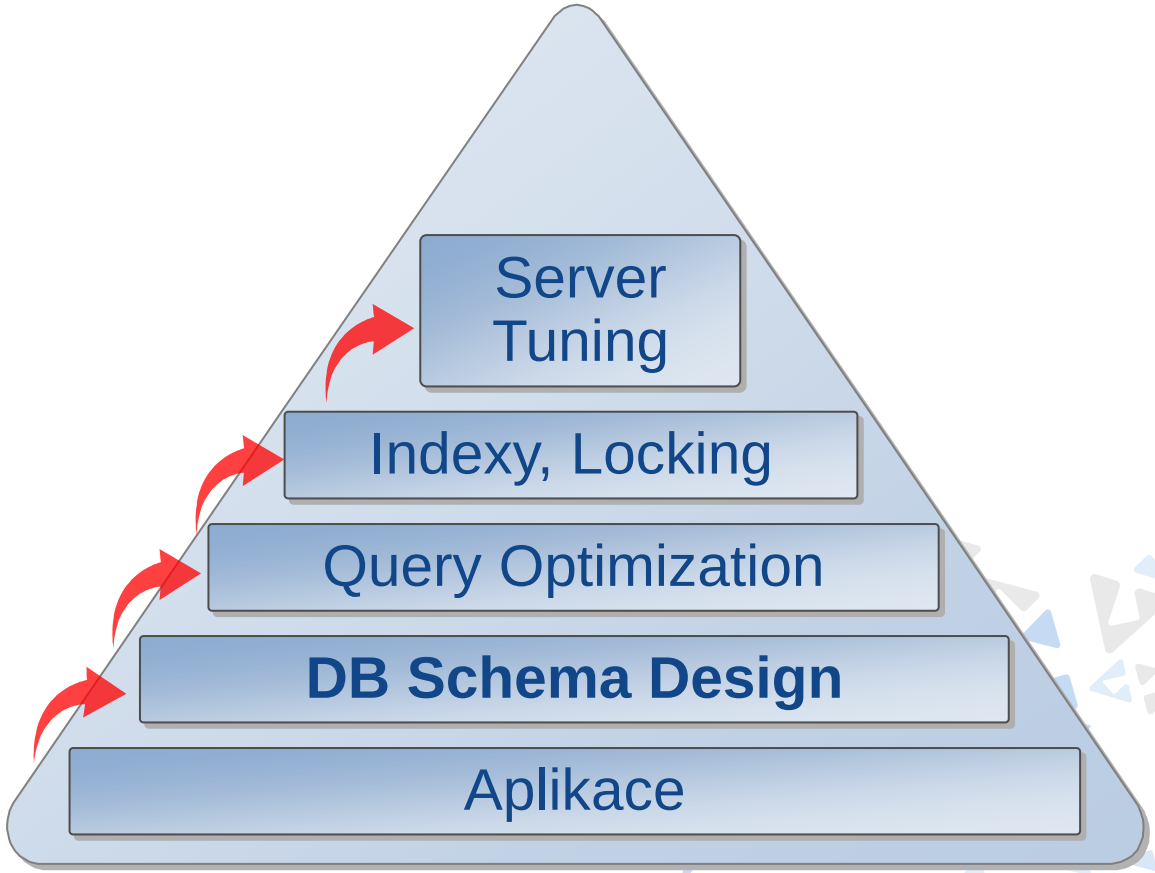
@RobertHaken

software & cloud architect, founder, HAVIT, s.r.o.

Návrh schématu databáze

Proč je schéma DB důležité?

- klíčový prvek designu aplikace
- zásadní vliv na výkon
  - rychlost
  - objem dat
- konzistence dat



Ukázka cílové podoby DB schématu

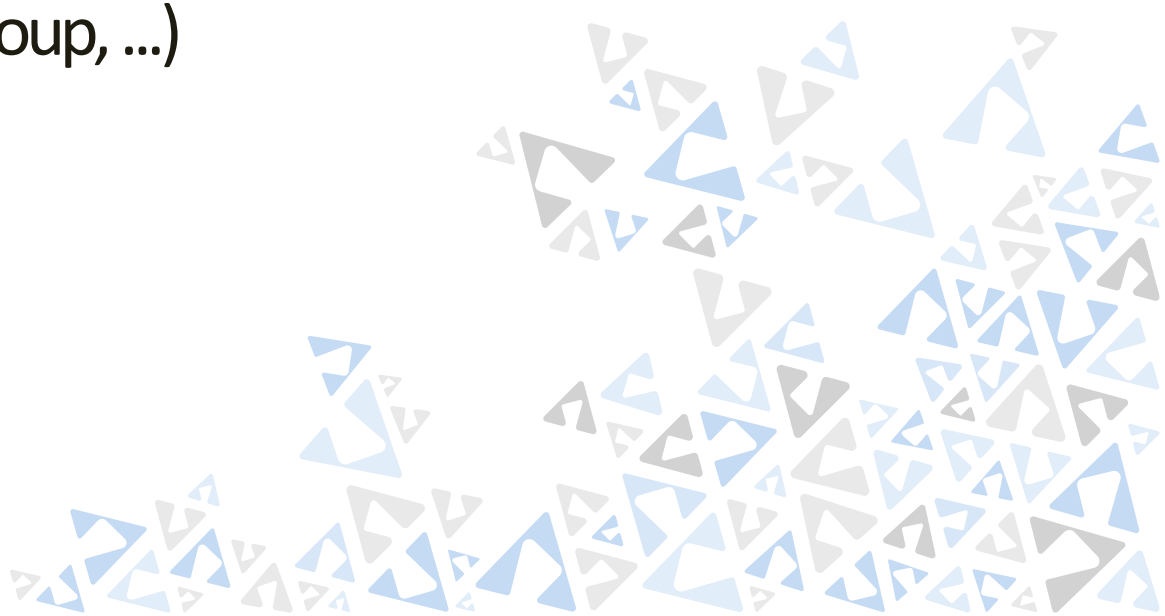
DEMO

Hlavní zásady tvorby schématu

- Konvence a standardy
- Přirozená míra normalizace
- Respektujte scénáře použití
- Standardní návrhové vzory
- Vlastní návrhové vzory
- Self-documented + přehlednost

Konvence – Naming Conventions

- název tabulky jednotným číslem
- výstižná podstatná a přídavná jména
- PascalCase
- bez národních znaků, speciálních znaků, mezer, atp.
- raději nepřekrývat s keywords (Order, Text, Group, ...)
- zkratky lowercase (např. SazbaDph, HtmlText)
- jednotná terminologie



Naming conventions II.

- suffix ID/Id pro všechny PK i FK sloupce (InvoiceId)
- PK vždy v podobě Id/TableNameId
- FK jako TargetTableNameId, pokud není potřeba rozlišit
  - InvoiceItem.InvoiceId
  - Invoice.DeliveryAddressId, Employee.BossId
- Varianta konvencí – trigram prefixy
  - tabulka Invoice, sloupce invID, invNumber, ...

Standardy – Výchozí schéma, PK

- každá tabulka jednoduchý umělý PK typu int
  - s výjimkou čistě vazebních tabulek M:N (PK = 2xFK)
- PK – primární klíč
  - umělý jednoduchý typu int, clustered
  - auto-increment (pokud není tabulka read-only)
- výhody
  - umělý = nemění se
  - velmi rychlé SQL operace (JOIN atp.)
  - monotónní růst = vhodný pro clustered index
  - zajišťuje unikátnost klíčů pro nové záznamy
- nevýhoda - nejednoznačný v distribuovaných systémech => GUID

Přirozená míra normalizace

- přiměřená atomicita atributů
  - Employee.Name
  - Employee.FirstName, Employee.Surname
  - vždy záleží na nejzazším **dohledném** scénáři
- NE vícehodnotové sloupce (...nebo někdy ano?)

EmployeeId	Name	PhoneNumbers
156	Josef Novák	608 123 456, 602 123 456
157	Marta Nová	601 123 456

Přirozená míra normalizace II.

- NIKDY vícenásobné sloupce

EmployeeId	Name	Phone1	Phone2
156	Josef Novák	608 123 456	602 123 456
157	Marta Nová	601 123 456	*NULL*

- správná závislost na klíči (PK) je rozhodující

EmployeeId	Name	Department	DepartmentManagerID
156	Josef Novák	IT	158 *(Karel Malý)*
157	Marta Nová	IT	158 *(Karel Malý)*
180	Helena Malá	HR	180 *(Jan Bílý)*

  - směřujte pozornost na sebemenší **redundance**
  - každá duplicita dat je velmi podezřelá !!!

Denormalizace

- denormalizace pouze, pokud požadovaného výkonu v klíčových scénářích nejsme schopni dosáhnout normalizovaným schématem !!!
- vždy něco ztrácíme, abychom jinde získali
- obtížnější údržba (nejlépe automatizace)
- potenciální nekonzistence

Standardy - další doporučení

- procenta – ukládat jako koeficient 100% => 1.00
- stabilizace významu a používání pojmů
  - např. marže / přirážka / rabat / sleva / koeficient / ...
  - User / Login / LoginAccount / Uzivatel / ...
- využívejte Database Diagrams (DB First)
  - Komentujte/Dokumentujte – např. xp attribute MS_Description
  - např. DB>doc generátor do HTML/XML/WikiPlex
- využívejte O/R Mappery (Code First)
  - DDD - Domain Driven Development
  - lze „anemický model“

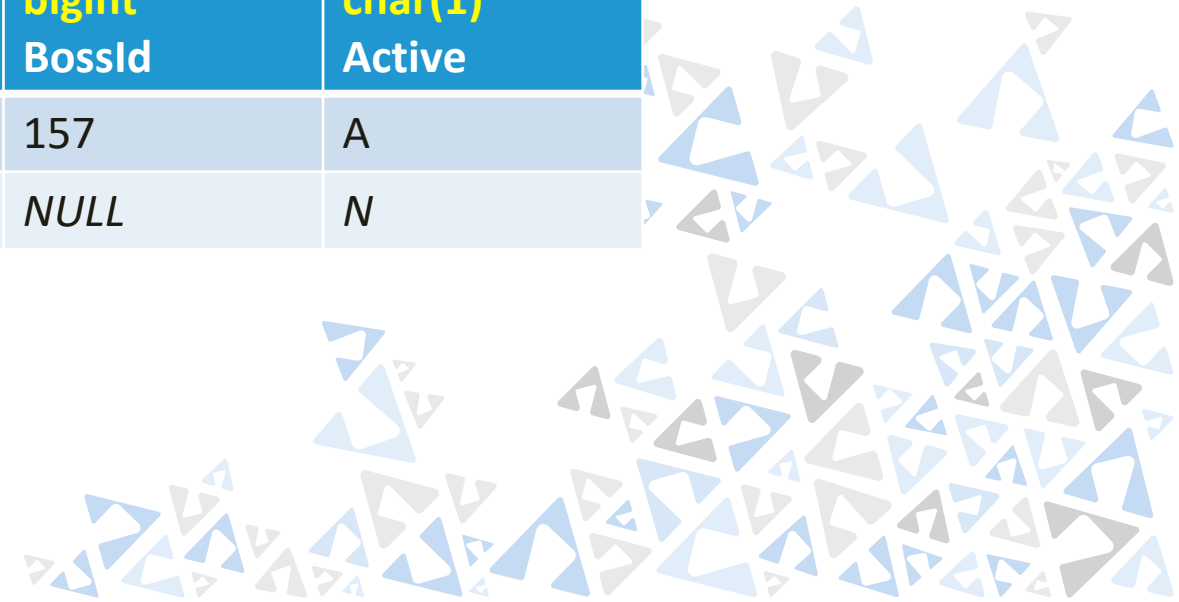
Schema Killers

- hodnoty v názvech sloupců

EmployeeId	Name	Salary2010	Salary2011
156	Josef Novák	123 456,38	348 260,30
157	Marta Nová	2 650 456,42	123 128,40

- chybné datové typy

int EmployeeId	varchar(20) StartDate	bigint BossId	char(1) Active
156	20.3.2011	157	A
157	17.6.1989	*NULL*	*N*



Schema Killers II.

- nedefinovaný PK tabulky
- chybějící reference mezi tabulkami (FK)
- vazba na jiný sloupec, než PK

DirectoryEntryID	EntryCode	ParentEntryCode
123	{123-XYZ-....-ABC-123}	{124-XYZ-....-ABC-124}
234	{124-XYZ-....-ABC-124}	*NULL*

Bad Practices

- podřizování schématu lidské čitelnosti dat
  - přirozené PK/FK (Username, číslo faktury, ...)
    - PROTI: nejsou stabilní
    - PROTI: nejsou výkonné pro typické operace (JOIN atp.)
    - PROTI: nejsou obv. monotónně rostoucí (CLUSTERED)
  - chybějící číselníky, hodnoty přímo místo FK
- magic numbers = chybějící enum-tabulky
  - PROTI: chybí validace hodnot, konzistence
  - VÝJIMKA: O/R mappery

Bad Practices II.

- hromadné číselníky (popř. složený PK)

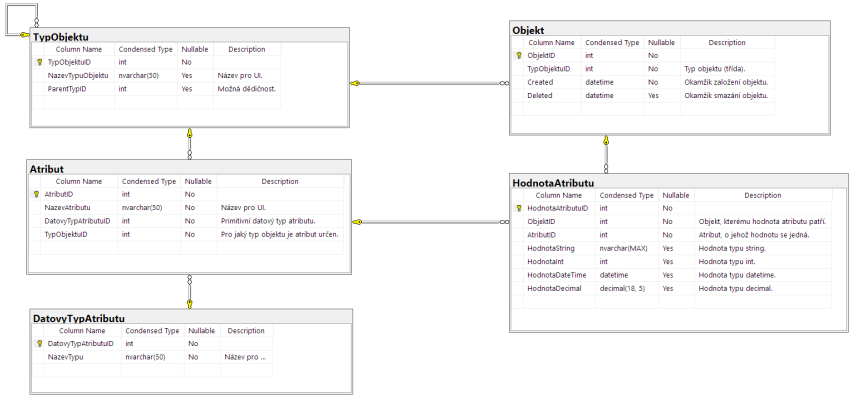
CodebookItemId	CodebookId	Value
1	1	aktivní
2	1	neaktivní
3	2	hnědá
4	2	červená
5	2	modrá

  - PROTI: obtížná rozšiřitelnost o další vlastnosti
  - PROTI: komplikované FK odkazy, validace konzistence
  - komplikovanější indexování (složené/filtrované)
  - PRO: unifikované použití/editace z aplikace

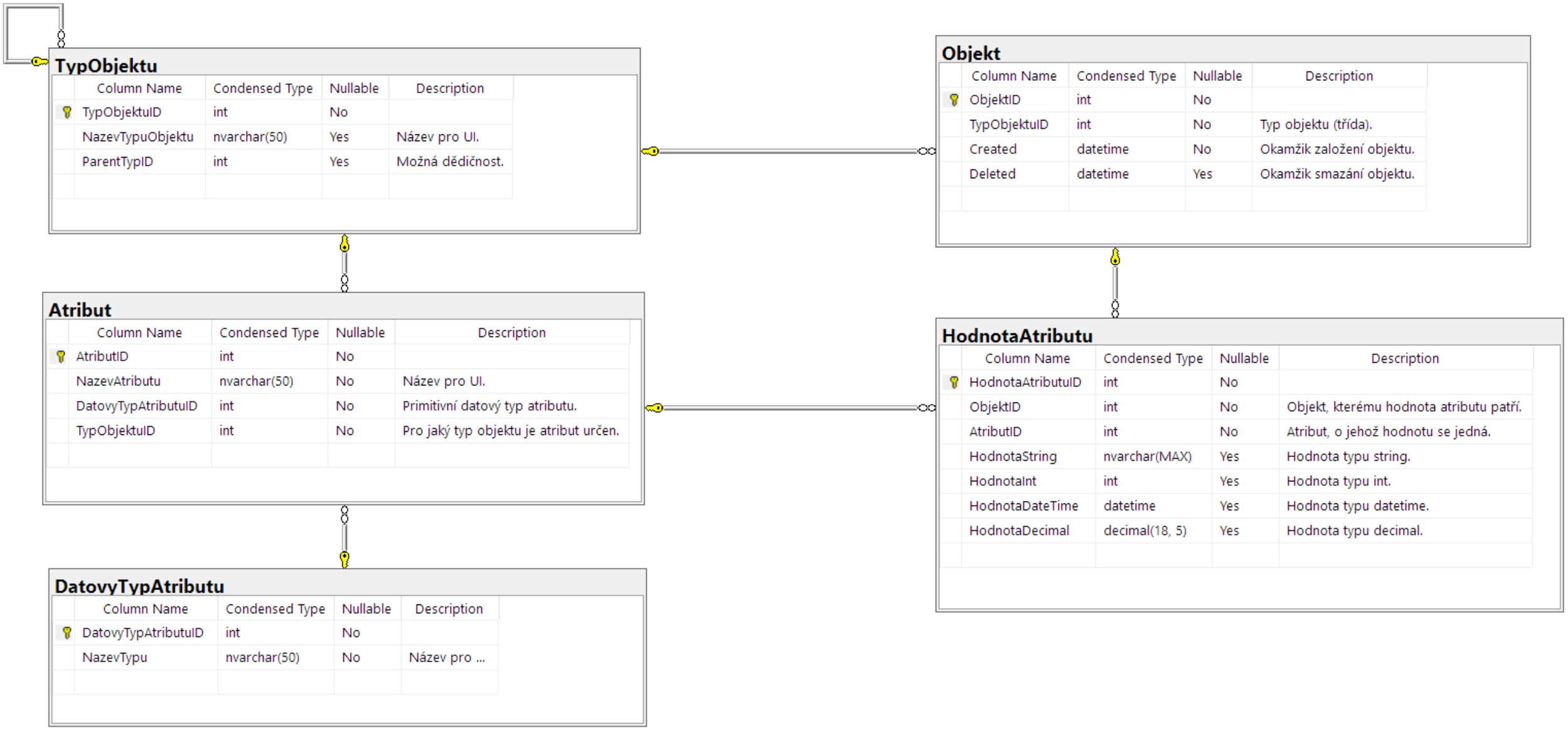
Patterns – Dynamické datové schéma

Atributová databáze („Attrida“)

- uložení všech objektů v cca 5 tabulkách
- PRO: snadná dynamická změna schématu za provozu
- PRO: relativně slušný výkon
- PRO: snadná implementace dědičnosti / interface
- PROTI: komplikované validace FK
- PROTI: komplikovanější dotazování, výkonové ztráty



Attrida - schéma



TypObjektu

Column Name	Condensed Type	Nullable	Description
TypObjektuID	int	No	
NazevTypuObjektu	nvarchar(50)	Yes	Název pro UI.
ParentTypID	int	Yes	Možná dědičnost.

Objekt

Column Name	Condensed Type	Nullable	Description
ObjektID	int	No	
TypObjektuID	int	No	Typ objektu (třída).
Created	datetime	No	Okamžik založení objektu.
Deleted	datetime	Yes	Okamžik smazání objektu.

Atribut

Column Name	Condensed Type	Nullable	Description
AtributID	int	No	
NazevAtributu	nvarchar(50)	No	Název pro UI.
DatovyTypAtributuID	int	No	Primitivní datový typ atributu.
TypObjektuID	int	No	Pro jaký typ objektu je atribut určen.

HodnotaAtributu

Column Name	Condensed Type	Nullable	Description
HodnotaAtributuID	int	No	
ObjektID	int	No	Objekt, kterému hodnota atributu patří.
AtributID	int	No	Atribut, o jehož hodnotu se jedná.
HodnotaString	nvarchar(MAX)	Yes	Hodnota typu string.
HodnotaInt	int	Yes	Hodnota typu int.
HodnotaDateTime	datetime	Yes	Hodnota typu datetime.
HodnotaDecimal	decimal(18, 5)	Yes	Hodnota typu decimal.

DatovyTypAtributu

Column Name	Condensed Type	Nullable	Description
DatovyTypAtributuID	int	No	
NazevTypu	nvarchar(50)	No	Název pro ...

Attrida - dotazy

```sql
-- Vypiš seznam všech objektů a jejich typů
SELECT
    Objekt.ObjektID,
    TypObjektu.NazevTypuObjektu
  FROM Objekt
    INNER JOIN TypObjektu ON (Objekt.TypObjektuID = TypObjektu.TypObjektuID)
```

	ObjektID	NazevTypuObjektu
1	1	Faktura
2	2	Faktura
3	3	Subjekt
4	4	Subjekt
5	5	Subjekt

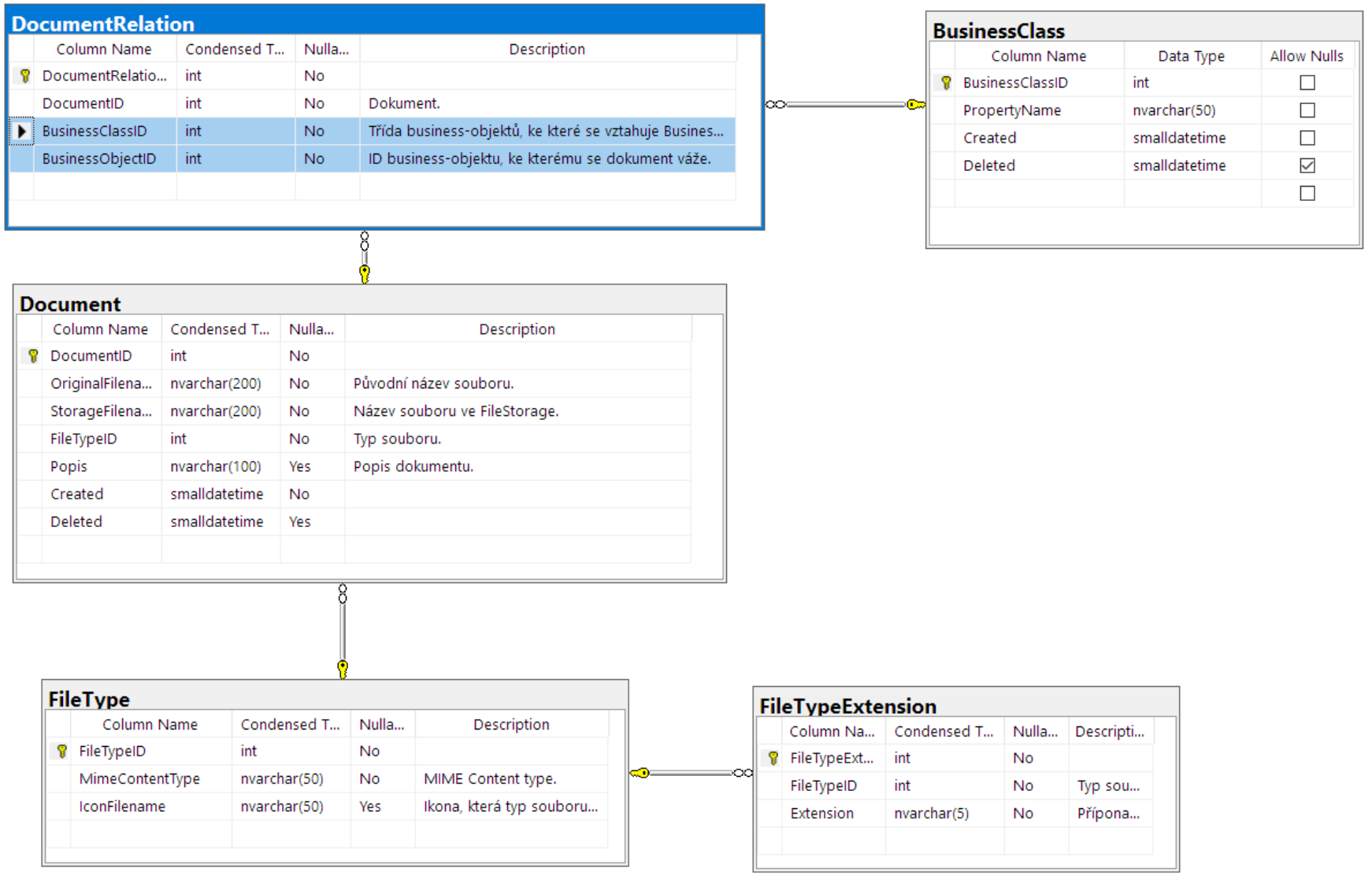
```sql
-- Vypiš hodnoty atributů určitého objektu
SELECT
    Atribut.NazevAtributu,
    HodnotaAtributu.*
  FROM HodnotaAtributu
    INNER JOIN Atribut ON (HodnotaAtributu.AtributID = Atribut.AtributID)
  WHERE
    HodnotaAtributu.ObjektID = 3 /* ObjektID objektu, který nás zajímá */
```

	NazevAtributu	HodnotaAtributuID	ObjektID	AtributID	HodnotaString	HodnotaInt	HodnotaDateTime	HodnotaDecimal
1	Název subjektu	7	3	1	HAVIT	NULL	NULL	NULL
2	Ulice	8	3	2	Zelený pruh 99	NULL	NULL	NULL
3	Počet zaměstnanců	9	3	3	6	NULL	NULL	NULL

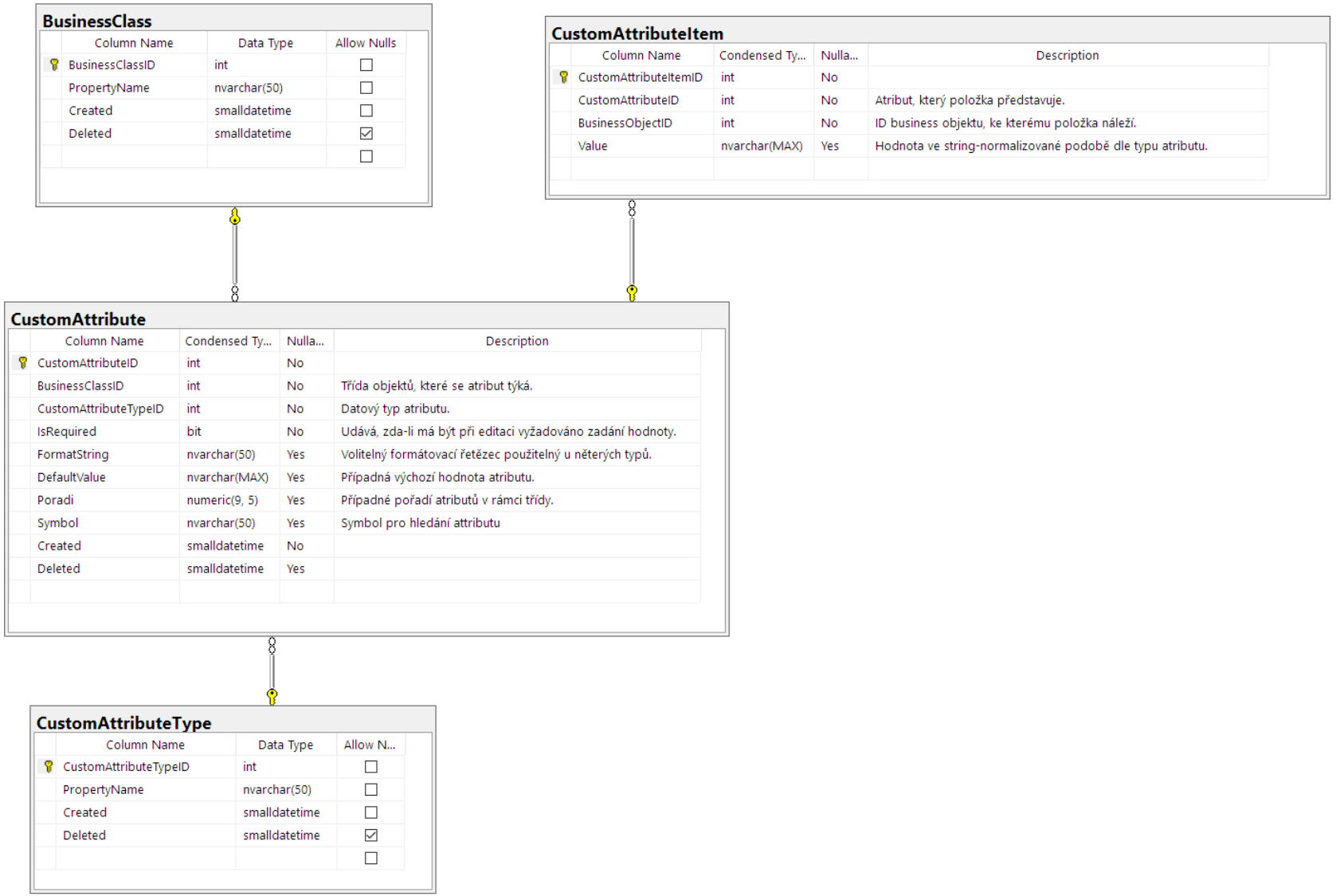
```sql
-- Vypiš čísla faktur a datum jejich vystavení, kde odběratelem je HAVIT
SELECT
    cisloFaktury.HodnotaInt AS CisloFaktury,
    datumVystaveni.HodnotaDateTime AS DatumVystaveni
  FROM Objekt faktura
    INNER JOIN HodnotaAtributu cisloFaktury ON ((faktura.ObjektID = cisloFaktury.ObjektID) AND (cisloFaktury.AtributID = 4 /* Číslo faktury */))
    INNER JOIN HodnotaAtributu odberatel ON ((faktura.ObjektID = odberatel.ObjektID) AND (odberatel.AtributID = 5 /* Odběratel */))
    LEFT JOIN HodnotaAtributu datumVystaveni ON ((faktura.ObjektID = datumVystaveni.ObjektID) AND (datumVystaveni.AtributID = 6 /* Datum vystavení */))
  WHERE
    (odberatel.HodnotaInt = 3 /* ObjektID firmy, jejíž faktury hledáme */)
```

	CisloFaktury	DatumVystaveni
1	40256	2010-04-23 00:00:00.000
2	40257	2010-04-24 00:00:00.000

Attrida – rozšíření statického schématu V1 - dokumenty



Attrida – rozšíření statického schématu V1 - atributy



BusinessClass

Column Name	Data Type	Allow Nulls
BusinessClassID	int	☐
PropertyName	nvarchar(50)	☐
Created	smalldatetime	☐
Deleted	smalldatetime	☑
		☐

CustomAttributeItem

Column Name	Condensed Ty...	Nulla...	Description
CustomAttributeItemID	int	No	
CustomAttributeID	int	No	Atribut, který položka představuje.
BusinessObjectID	int	No	ID business objektu, ke kterému položka náleží.
Value	nvarchar(MAX)	Yes	Hodnota ve string-normalizované podobě dle typu atributu.

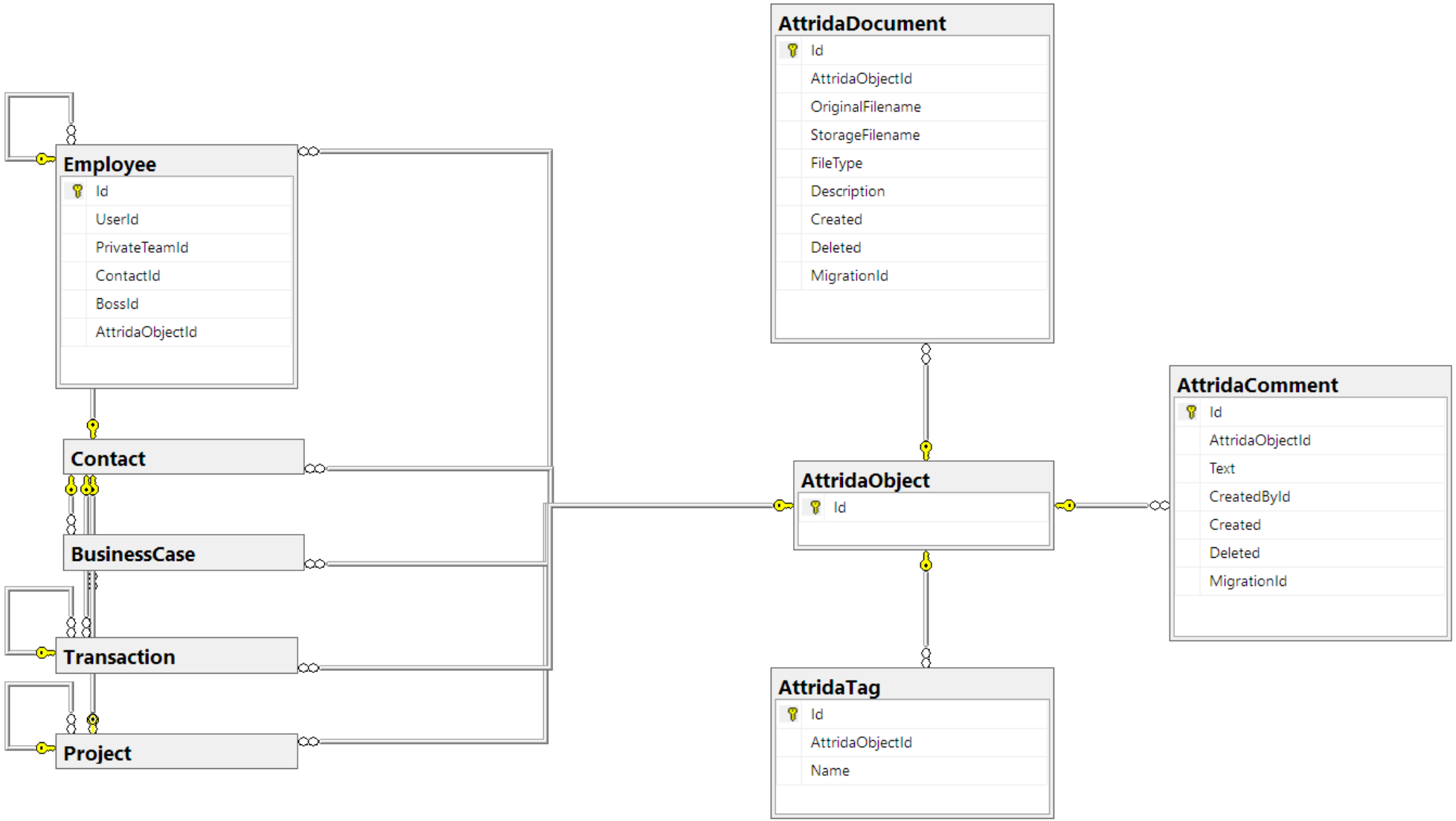
CustomAttribute

Column Name	Condensed Ty...	Nulla...	Description
CustomAttributeID	int	No	
BusinessClassID	int	No	Třída objektů, které se atribut týká.
CustomAttributeTypeID	int	No	Datový typ atributu.
IsRequired	bit	No	Udává, zda-li má být při editaci vyžadováno zadání hodnoty.
FormatString	nvarchar(50)	Yes	Volitelný formátovací řetězec použitelný u některých typů.
DefaultValue	nvarchar(MAX)	Yes	Případná výchozí hodnota atributu.
Poradi	numeric(9, 5)	Yes	Případné pořadí atributů v rámci třídy.
Symbol	nvarchar(50)	Yes	Symbol pro hledání attributu
Created	smalldatetime	No	
Deleted	smalldatetime	Yes	

CustomAttributeType

Column Name	Data Type	Allow N...
CustomAttributeTypeID	int	☐
PropertyName	nvarchar(50)	☐
Created	smalldatetime	☐
Deleted	smalldatetime	☑
		☐

Attrida – rozšíření statického schématu V2 - AttridaObject



Patterns – Embedded Object (JSON/XML)

sloupec s objektem serializovaným do JSON / XML

O/R mapping

datový typ XML + SQL funkce pro práci s JSON

DEMO

Patterns – Verzování dat v čase (historizace) – V1

Pracovnik *

	Column Name	Condensed Ty...	Nulla...	Description
🔑	PracovnikID	int	No	
	UzivatelID	int	Yes	Uživatelský účet pracovníka.
	PrivatniTeamID	int	No	Privátní team pracovník (jednočlenný team).
	Created	smalldatetime	No	
	Deleted	smalldatetime	Yes	

Pracovnik

PracovnikHistorie

	Column Name	Condensed Ty...	Nulla...	Description
🔑	PracovnikHistorieID	int	No	
	PracovnikID	int	No	Pracovník, kterého se záznam týká.
	DatumOd	smalldatetime	No	Datum, od kdy je záznam účinný (včetně).
	PracovniPoziceID	int	No	Pracovní pozice (zařazení).
	PracovniVztahID	int	No	Pracovní vztah (HPP, VPP, DPP, DPČ, ičař, ...).
	SjednanaZakladniSazba	money	No	Sjednaná základní sazba mzdy/odměny v režimu dle typu ...
	TypUvazkuID	int	No	Typ úvazku (určuje, k čemu se vztahuje SjednanaZakladniS...
	HodinovaSazbaOsobnichNakla...	money	No	Hodinová sazba úplných přímých osobních nákladů (tj. vče...
	PlanovanePremie	numeric(6, 3)	No	Plánované prémie (odměny) v procentech [měřítko 100% ...
	KoeficientPrirazkyRezie	decimal(9, 4)	Yes	Koeficient režie pracovníka

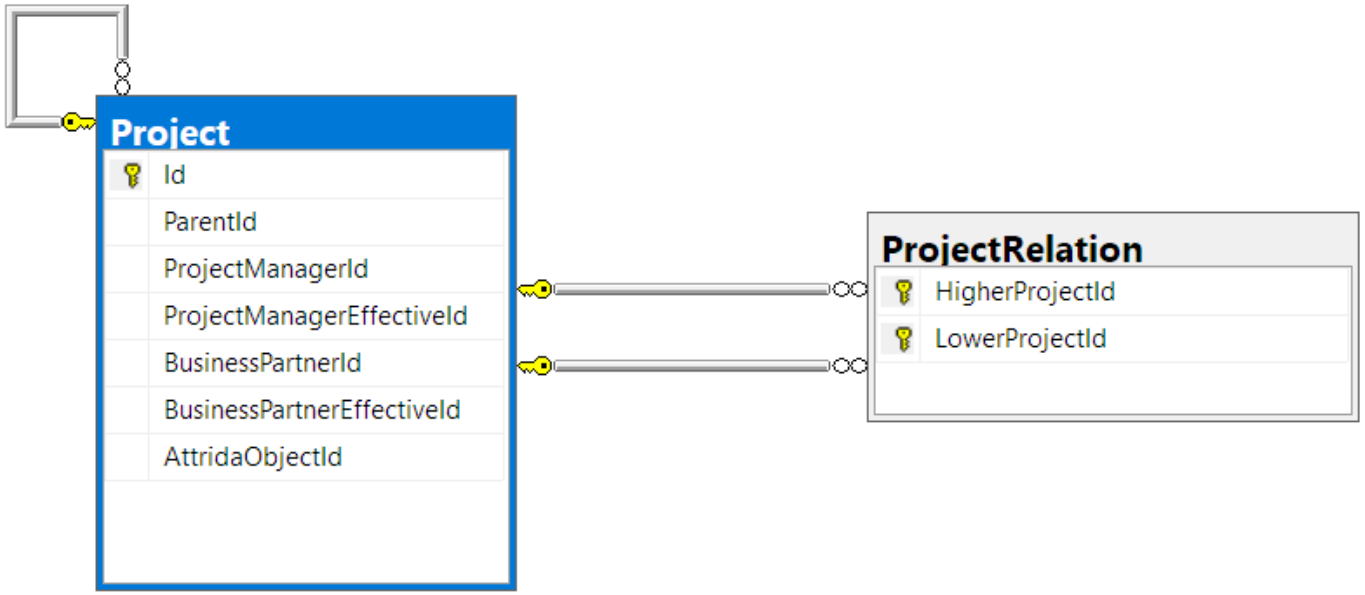
Patterns – Verzování dat v čase – SQL Temporal Tables

```
CREATE TABLE WebsiteUserInfo
(
    [UserID] int NOT NULL PRIMARY KEY CLUSTERED
  , [UserName] nvarchar(100) NOT NULL
  , [PagesVisited] int NOT NULL
  , [ValidFrom] datetime2 (0) GENERATED ALWAYS AS ROW START
  , [ValidTo] datetime2 (0) GENERATED ALWAYS AS ROW END
  , PERIOD FOR SYSTEM_TIME (ValidFrom, ValidTo)
 )
 WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.WebsiteUserInfoHistory));
```

DEMO

Patterns – Hierarchie a práce s ní

!redundance!



též viz SQL hierarchyid

Patterns – Dědičnost objektů

Class Table Inheritance

- tabulka pro každou třídu i abstraktní předky
- pouze atributy definované třídou (nutný JOIN)

Single Table Inheritance

- jedna tabulka pro všechny možné potomky

Concrete Table Inheritance

- tabulka pro každou neabstraktní třídu
- sloupce všech předků + třídy

References

Blog – HAVIT Knowledge Base - http://knowledge-base.havit.cz/ + .eu

Twitter - @RobertHaken

YouTube - https://www.youtube.com/user/HAVITcz