



Samostatná práce z předmětu KIV/DBM1

ZOO – DOKUMENTACE

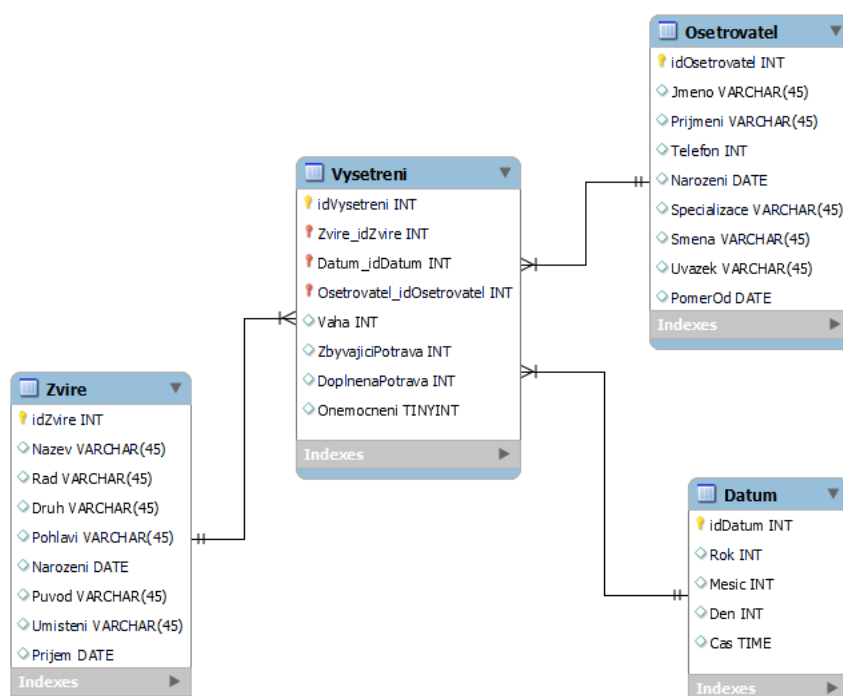
PETR CHRZ

Úvod

Pro moji samostatnou práci jsem již využil podklady samostatné práce z předmětu KIV/ZIS, kdy jsem navrhoval databázový model na téma Zoo. V původní datové sadě jsem měl tabulky *Zvire*, následně *Biom* a *Kontinent*, vazby 1:N. Dále tabulky *Prispevek* a *Sponzor*, kde bylo cílem evidovat finanční podporu pro jednotlivé živočichy. Nakonec tabulku *Vysetreni* a následně tabulku *Osetrovatel*, přičemž u každého živočicha byla možná evidence jeho vyšetření a osoby, která vyšetření prováděla. Z těchto posledních dvou tabulek (*Vysetreni*, *Osetrovatel*) jsem si vzal inspiraci a použil je pro moji samostatnou práci z tohoto předmětu, k tabulkám jsem ještě přiložil tabulku *Zvire* a *Datum* tak, aby bylo možné se dívat z různých úhlů pohledu, ať už z pohledu živočichů nebo ošetřovatelů, data.

Popis modelu

Jak již bylo naznačeno jako faktovou tabulku jsem zvolil *Vysetreni*, jednotlivé dimenze pak představují tabulky *Zvire*, *Osetrovatel* a *Datum*. Faktová tabulka má tyto specifické atributy: *Vaha*, *ZbyvajiciPotrava*, *DoplnenaPotrava* ve smyslu potřeby evidovat vývoj váhy u konkrétních živočichů a jejich zbývající a nově doplněnou potravu. Zoologická zahrada zaměstnává různé osoby, kteří mohou péči zanedbávat a následně zvířata mohou v důsledku toho strádat, proto tyto atributy. Posledním atributem je *Onemocneni* (Boolean), který plně rozšiřuje myšlenku. Dále tabulka samozřejmě obsahuje primární klíč a cizí klíče odkazující na jednotlivé dimenze. Samotné dimenze jsou pak tabulky *Zvire*, *Osetrovatel* a *Datum*, které pouze popisují jednotlivé entity viz. ER – diagram.



Obrázek 1 ER Diagram

Samotná data jsem vygeneroval pomocí umělé inteligence ChatGPT a BingChat, poskytl jsem atributy jednotlivých tabulek a požádal o vygenerování dat ve formátu CSV. Zde nastal menší problém s daty pro tabulku *Zvire*, ChatGPT generoval příliš málo rozmanitá data a až moc si vymýšlel. Např. po pár akceptovatelných řádcích se začali objevovat názvy zvířat jako „Pingvin“ nebo „Jehňátko“, dále téměř u všech byl uveden druh „Savec“ a řád „Selmy“, třeba i u orla. K tomuto lépe posloužil Bing Chat od Microsoftu, kterého jsem dotázel ať nejprve určí u jednotlivých zvířat druh a třídu, poté jsem vygenerované asociace poskytl opět chatbotu od OpenAI, který už data doplnil do požadované struktury, přesto jsou ale některé kombinace trochu úsměvné, např. Plameňák a řád „Plamenakovci“ nebo Tučňák a „Tucnakotvorni“, ale pro účely této práce to není podstatné. Nutno ještě zmínit, že používám neplacenou verzi ChatGPT, k dnešnímu datu (2.5.2024) verze GPT- 3.5.

```
22,Klokan,Savci,Vacnatci,Samice,2019-03-05,Australie,Q07,2023-06-22
23,Medved,Savci,Selmy,Samec,2018-11-10,SeverniAmerika,L02,2023-06-05
24,Orangutan,Savci,Primati,Samice,2016-03-02,Asie,L19,2023-03-10
25,Zebra,Savci,Sudokopytnici,Samice,2018-08-20,Afrika,B11,2023-02-03
26,Krajta,Plazi,Supinati,Samice,2017-06-10,JizniAmerika,V02,2023-07-10
27,Delfin,Savci,Kytovci,Samec,2018-06-15,Oceanie,O24,2023-06-15
28,Makak,Savci,Primati,Samice,2019-01-20,Asie,N18,2023-06-12
29,Slon,Savci,Chobotnatci,Samice,2016-07-12,Afrika,E09,2023-02-12
30,Krajta,Plazi,Supinati,Samec,2017-07-22,JizniAmerika,X22,2023-04-20
31,Zirafa,Savci,Sudokopytnici,Samec,2020-01-10,Afrika,C17,2023-02-05
32,Lev,Savci,Selmy,Samec,2019-05-15,Afrika,A25,2023-02-01
33,Plamenak,Ptaci,Plamenakovci,Samec,2018-01-04,JizniAmerika,E26,2023-05-12
34,Pstros,Ptaci,Pstrosotvarni,Samec,2019-02-14,Afrika,G06,2023-05-20
35,Levhart,Savci,Selmy,Samec,2017-03-19,Afrika,R21,2023-03-30
36,Tucnak,Ptaci,Tucnakotvarni,Samice,2019-06-09,Antarktida,D19,2023-05-10
```

Obrázek 2 Ukázka dat, tabulka *Zvire*

Transformace dat a samotná implementace řešení

Při samotné implementaci jsem si nejprve vytvořil pole (csvFiles), do kterého jsem uložil cesty k jednotlivým souborům. Dále pro ověření, že jsou data správně načítána, jsem si vytvořil třídu CSVReader. Pro pohodlí ještě vytvořil konstruktor, kterému jsem předal pole s cestami k souborům a spojení k databázi. Následně pomocí standartní implementace (metoda readCSVFiles), nechal obsahy vypsat do konzole, vše se vypsalo správně. Dále jsem tak vytvořil třídu CSVLoader, která již zajišťuje samotné vkládání dat do tabulek. Opět jsem vytvořil konstruktor a část z kódu ze třídy CSVReader pro načítání dat. Zde dále dobře posloužil opět ChatGPT, kterému jsem společně s mým konstruktorem poskytl metoda readCSVFiles a požádal ho o vytvoření části kódu pro vkládání dat. Tady opět nastal problém, při vkládání dat vnikal problém s konfliktem, který porušoval problém integritních omezení. V tomto případě stačilo dodat jednoduchou podmínku „ON CONFLICT DO NOTHING“, která

zajišťuje, že v případě konfliktu bude operace vkládání ignorována. Dále při pokusu vkládání dat do tabulky Vysetreni a atributu Onemocneni nastal problém ten, že v databázi je atribut nastaven typu Boolean, ale java skript tento sloupec načítal jako String, bylo tedy nutné hodnoty konvertovat. Opět dobře posloužil ChatGPT, který to vyřešil pomocí větvení.

Popis analytických dotazů

Pro ověření funkčnosti databáze jsem zvolil 2 dotazy. První obsáhlejší SELECT, který spojuje více tabulek, s klauzulemi GROUP BY a ORDER BY. Druhý dotaz využívá rozšíření GROUP BY ROLLUP. Níže jsou jednotlivé dotazy popsány.

Dotaz č. 1

Tento dotaz slouží k analýze pracovní vytíženosti ošetřovatelů během měsíce dubna (4). Dotaz vybere jednotlivé atributy z tabulky Osetrovatel, následně spočítá celkový počet vyšetření provedeným každým ošetřovatelem v tomto období, kde je pak celkový počet seskupen pomocí klauzule GROUP BY dle jednotlivých ošetřovatelů. Nakonec jsou výsledky seřazeny dle celkové počtu vyšetření, sestupně.

```
SELECT Osetrovatel.idOsetrovatel,  
       Osetrovatel.Jmeno,  
       Osetrovatel.Prijmeni,  
       Osetrovatel.Telefon,  
       Osetrovatel.Smena,  
       Osetrovatel.Uvazek,  
       COUNT(Vysetreni.idVysetreni) AS PocetVysetreni  
FROM Osetrovatel  
JOIN Vysetreni ON Osetrovatel.idOsetrovatel = Vysetreni.Osetrovatel_idOsetrovatel  
JOIN Datum ON Vysetreni.Datum_idDatum = Datum.idDatum  
WHERE Datum.Mesic = 4  
GROUP BY Osetrovatel.*  
ORDER BY PocetVysetreni DESC;
```

Obrázek 3 Dotaz č. 1

Z výsledku tohoto dotazu si můžeme všimnout, že nejvíce vyšetření za měsíc duben provedli Petr Kovář a Marek Novotný, celkem 6 vyšetření, viz obrázek.

idOsetrovatel	Jmeno	Prijmeni	Telefon	Smena	Uvazek	PocetVysetreni
3	Petr	Kovar	555666777	Nocni	Zkraceny	6
8	Marek	Novotny	666555444	Odpoledni	Plny	6
15	Katerina	Sedlackova	777666555	Odpoledni	Plny	3
22	Barbora	Krejцова	777888999	Odpoledni	Plny	3
13	Alena	Hajkova	111222333	Odpoledni	Brigada	3
17	Adela	Kralova	666777888	Nocni	Zkraceny	3
1	Eva	Horakova	987654321	Odpoledni	Brigada	3
27	Anna	Pachova	666777888	Denni	Plny	3

Obrázek 4 Výsledek dotazu č. 1

Dotaz č. 2

U druhého dotazu jsem si kladl za cíl analyzovat, zda ošetřovatelé s různými úvazky provádějící stejně kvalitní o péči o zvířata, doplňují stejné množství potravy. Vybral jsem tedy atributy Druh, Řád, Úvazek a pomocí funkce AVG() spočítal průměrnou doplněnou potravu. Následně pomocí klauzule GROUP BY ROLLUP jsem zajistil vytvoření hierarchie v agregovaných datech. Dotaz tedy vrací jednotlivé mezisoučty a celkový součet doplněné potravy po úrovních (dle druhu, řádu a úvazků ošetřovatelů). Nakonec jsem ještě přidal podmínku, že počet provedených vyšetření bude alespoň 2. Data jsem ještě seřadil dle druhu a řádu.

```
SELECT z.Druh,
       z.Rad,
       o.Uvazek,
       AVG(v.DoplnenaPotrava) AS PrumernaDoplnenaPotrava
FROM Zvire z
JOIN Vysetreni v ON z.idZvire = v.Zvire_idZvire
JOIN Osetrovatel o ON v.Osetrovatel_idOsetrovatel = o.idOsetrovatel
GROUP BY ROLLUP(z.Druh, z.Rad, o.Uvazek)
HAVING COUNT(v.idVysetreni) > 2
ORDER BY z.Druh, z.Rad;
```

Obrázek 5 Dotaz č. 2 (GROUP BY ROLLUP)

Druh	Rad	Uvazek	PrumernaDoplnenaPotrava
Plazi	Supinati		4.5
Plazi			4.5
Ptaci	Dravci		5.222222222222222
Ptaci	Dravci	Plny	5.222222222222222
Ptaci	Plamenakovci	Plny	5.4
Ptaci	Plamenakovci		5.9
Ptaci	Plamenakovci	Zkraceny	7.0
Ptaci	Pstrosotvarni		4.125
Ptaci	Pstrosotvarni	Zkraceny	5.0
Ptaci	Pstrosotvarni	Brigada	3.0
Ptaci	Tucnakotvarni	Plny	4.5
Ptaci	Tucnakotvarni		5.0
Ptaci			5.125
Savci	Chobotnatci	Zkraceny	7.0
Savci	Chobotnatci		6.333333333333333
Savci	Kytovci		5.125
Savci	Kytovci	Plny	5.75
Savci	Kytovci	Zkraceny	5.0
Savci	Primati		4.266666666666667
Savci	Primati	Plny	4.25
Savci	Selmy		5.3076923076923075
Savci	Selmy	Plny	5.5
Savci	Selmy	Brigada	4.0
Savci	Selmy	Zkraceny	6.166666666666667
Savci	Sudokopytnici		4.833333333333333

Obrázek 6 Výsledek dotazu č. 2

Při zapojení troše fantazie, lze z výsledku tohoto dotazu dedukovat usuzovat závěr, že ošetřovatelé se zkráceným úvazkem doplňují živočichům více potravy než ostatní (vyjma zvířatům třídy kytovci), resp. mají vyšší průměrnou doplněnou potravu. Samozřejmě průměr je citlivý na extrémní hodnoty, bylo pak tedy nutné analyzovat, zda to není záležitost jednoho konkrétního ošetřovatele, který z nějakého důvodu doplňuje mnohem více potravy. Zda toto zjištění je pozitivní či negativní záleží pak na konkrétní situaci.

Závěr

Závěrem lze jen říci, že při tvorbě této práce, jsem se snažil postupovat co nejvíce efektivně, za účelem nejrychlejšího odhalení příčiny vzniklých problémů. Po vytvoření databáze, jednotlivých tabulek a vygenerování, jsem zkusil nejprve tabulky manuálně naplnit přímo v program DBEaver. Hned jsem narazil na problém, tabulky nešly naplnit. Podíval jsem se tedy na vygenerovaná data a odhalil, že některé cizí klíče jsou ve větším rozsahu, než je počet ošetřovatelů. Po optimalizaci dat, jsem tedy 100% jistotu, že jsou v pořádku a případný problém už bude pouze v samotném skriptu.

Následně např. po samotné implementaci ETL a neúspěšném pokusu o naplnění tabulek, jsem tedy vcelku rychle pochopil, že bude nutné v kódu ošetřit správné načítání hodnot z „true“ a „false“ ze souboru převodem na hodnoty typu Boolean. Nakonec tedy počáteční obavy z této semestrální práce byly tak trochu zbytečné.