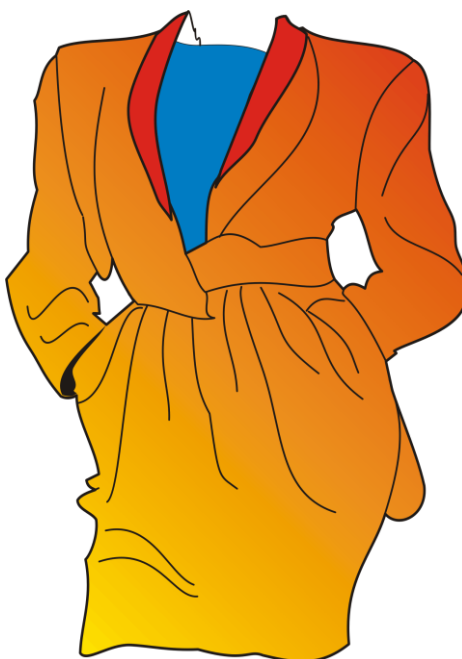


MÓDNÍ SALON

Příklad prezentace analýzy s dotazy k AI



Doc. Dr. Vladimír Homola, Ph.D.

Únor 2026

**k ISBN 978-80-248-4265-3
© VŠB-TU Ostrava 2019**

OBSAH

OBSAH	i
1 Anotace	1
2 Zadání úlohy	1
2.1 Formulace pro výuku.....	1
2.2 Formulace pro Umělou Inteligenci.....	1
3 Odpověď AI: MS Co-pilot.....	3
3.1 ## Návrh databáze pro krejčovskou evidenci.....	3
3.1.1 ### Tabulky	3
3.1.2 ### Struktura tabulek	3
3.1.3 ### Ukázkové SQL dotazy	4
3.2 Závěr AI: MS Co-pilot	4
4 Odpověď AI: Chat-GPT.....	5
4.1 Databáze v malém samostatném podnikání	5
4.1.1 Cíl: Evidence modelů, látek, zákazníků a dodávek	5
4.1.2 Přehled tabulek.....	5
4.1.3 Relace mezi tabulkami	6
4.1.4 Doporučené dotazy (SQL)	6
4.1.5 Jak databázi uložit.....	7
4.1.6 Rozšíření do budoucna (pokročilejší funkce)	7
4.2 Závěr AI: Chat-GPT	7
5 Rozšíření výuky SQL.....	8
5.1 Použitá notace.....	8
5.2 Příkaz CREATE TABLE	8
5.2.1 Definice příkazu CREATE TABLE:	8
5.2.2 Části zápisu příkazu	8
5.2.3 Poznámky.....	9
5.3 Klauzule CONSTRAINT - Omezení pole (polí).....	9
5.3.1 Single-field constraint (Omezení jednoho pole)	9
5.3.2 Multiple-field constraint (Omezení více polí)	10
5.3.3 Části zápisu omezení.....	10
5.3.4 Poznámky:.....	10
6 Struktura a data pro ověření	11
6.1 Struktura dat dle vlastní analýzy	12
6.2 Data pro prezentaci analýzy	13

6.2.1	Data tabulky MODELÝ	13
6.2.2	Data tabulky LÁTKY	13
6.2.3	Data tabulky ZÁKAZNÍCI	13
6.2.4	Data tabulky PRODEJE	14
7	Tabulky a dotazy dle MS Co-pilot	14
7.1	Lidská chyba AI	15
7.2	{Volitelná} příprava	15
7.3	Vytvoření prázdných tabulek	15
7.3.1	Tabulka Modely - CrteModely	15
7.3.2	Tabulka Látky - CrteLatky	16
7.3.3	Tabulka Zákazníci - CrteZakaznici	16
7.3.4	Tabulka ModelLátka - CrteModelLatka	16
7.3.5	Tabulka Šití - CrteSiti	16
7.3.6	Struktura databáze dle MS Co-pilot	16
7.4	Naplnění tabulek Modely, Látky a Zákazníci	18
7.4.1	Tabulka Modely - InsModely	18
7.4.2	Tabulka Látky - InsLatky	18
7.4.3	Tabulka Zákazníci - InsZakaznici	18
7.4.4	Tabulka ModelLátka - InsModelLatka	18
7.4.5	Tabulka Šití - InsSiti	18
7.4.6	Použití příkazů Insert Into	19
7.5	Hodnocení návrhů dotazů	20
7.5.1	Které modely jsem dodala letos v únoru? (Úloha 1.)	20
7.5.2	Kolik jsem si vydělala letos v únoru? (Úloha 2.)	20
7.5.3	Kolik jsem celkem spotřebovala té které látky? (Úloha 3.)	21
7.5.4	Který můj model je nejžádanější? (Úloha 4.)	21
7.5.5	Který model jsem ještě vůbec neudala? (Úloha 5.)	22
7.6	Databáze ke stažení	22
8	Tabulky a dotazy dle Chat-GPT	22
8.1	{Volitelná} příprava	22
8.2	Vytvoření prázdných tabulek	23
8.2.1	Tabulka Modely - CrteModely	23
8.2.2	Tabulka Látky - CrteLatky	23
8.2.3	Tabulka Zákazníci - CrteZakaznici	23
8.2.4	Tabulka Zakazky - CrteZakazky	23
8.2.5	Struktura databáze dle Chat-GPT	24
8.3	Naplnění tabulek Modely, Látky a Zákazníci	25
8.3.1	Tabulka Modely - InsModely	25
8.3.2	Tabulka Látky - InsLatky	25

8.3.3	Tabulka Zákazníci - InsZakaznici.....	25
8.3.4	Tabulka Zakazky - InsZakazky.....	26
8.3.5	Použití příkazů Insert Into.....	26
8.4	Hodnocení návrhů dotazů Chat-GPT	27
8.4.1	Dotaz 1: Které modely jsem dodala letos v únoru?	27
8.4.2	Dotaz 2: Kolik jsem si vydělala letos v únoru?	27
8.4.3	Dotaz 3: Kolik jsem celkem spotřebovala té které látky?.....	28
8.4.4	Dotaz 4: Který můj model je nejžádanější?	28
8.4.5	Dotaz 5: Který model jsem ještě vůbec neudala?	29
9	Závěr	29
10	Literatura a další zdroje ke studiu	29

1 Anotace

Výuka oblasti IT zabývající se databázovými informačními systémy byla v minulosti zavedena pro nižší ročníky některých oborů bakalářského prezenčního studia a rovněž jako předmět rekvalifikačního studia zaštitěného Úřadem práce. V obou případech se předpokládalo, že z hlediska databázové problematiky studenti byli (a jsou) „tabula rasa“ (metafora pro "List nepopsaný" 😊). Výuka předmětu proto zahrnuje jako jednu z kapitol příklad analýzy úlohy předkládané k řešení jakožto úlohy rutinně provozované na bázi zpracování dat. Výstupem analýzy má být (kromě jiného) návrh datových struktur vzhledem k informačním cílům praktického nasazení. Prezentace možného postupu při analýze je součástí přednášek zmíněné výuky. Postup je demonstrován na úloze vybrané z množiny příkladů k procvičování, dostupné studentům ve zdroji *Témata II k procvičování*. Jde konkrétně o úlohu 2: Salon. Pro ověření resp. demonstraci postupu byla připravena data a jimi naplněny tabulky odvozené při prezentaci analýzy.

Výuková prezentace vznikla v minulosti pro potřeby klasické prezenční výuky. Mezitím se převalila vlna kovidu, následoval přechod k výuce „off line“, až po zdokonalování hardwaru a softwaru potřebného pro praktické nasazení tzv. Umělé Inteligence. Její jedna část (té UI), tzv. konverzační umělá inteligence, při formulaci myšlenek na straně člověka ho (toho člověka) podivuhodně směřuje ke stejné osnově, která už léta tvoří základ zmíněné prezentace. Tento článek doplňuje výukovou prezentaci o ukázkou, která stejnou formulaci úlohy předkládanou studentům zpracuje Umělá Inteligence (přesněji kolegové programátoři UI jimi realizovanými algoritmy). Byly požádány dva modely UI (MS Co-pilot a Chat-GPT), oběma bylo zasláno totožné zadání. V článku jsou uvedena doslovná znění obou odpovědí, které jsou poté postupně realizovány a komentovány. Realizace probíhá na datech, jejichž základ tvoří tabulky zmíněné v předchozím odstavci. Na závěr je uveden odkaz na reálné databáze ke stažení, které se snaží využít návrhu AI v praxi (Literatura [1] a [2]).

2 Zadání úlohy

2.1 Formulace pro výuku

Poznámka: Úlohy k procvičování (nebo jim podobné) mohou být předloženy např. při zkoušce. Proto rádoby úsměvné formulace typu "Šijete tisíce modelů (asi pět) ..." nejsou samoučelné. Mají naznačit studentům, na jakém objemu dat by svou úlohu měli předvést - aby se na jedné straně zbytečně nezdržovali, na druhé straně aby řešení úlohy mohlo podchytit případné nedostatky. Nyní tedy ona formulace úlohy:

Jste majitelkou světově proslulého módního saloonu. Šijete tisíce modelů (asi pět), každý o různé pracnosti (vyjádřené počtem hodin) a různé průměrné spotřebě látky (vyjádřené metry). Stejně modely šijete i z různých látek (asi třech), každá o jiné ceně. Při šití účtujete 300 Kč za hodinu práce.

Vytvořte databázi, která bude sledovat, kdy jste dodali kterým zákazníkům (asi třem) které modely. Sestavte dotazy, které odpoví na otázky:

- Které modely jsem dodala letos v únoru?
- Kolik jsem si vydělala letos v únoru (to pro účely zdanění)?
- Kolik jsem celkem spotřebovala té které látky?
- Který můj model je nejžádanější?
- Který model jsem ještě vůbec neudala?

2.2 Formulace pro Umělou Inteligenci

Vyučila jsem se dámskou krejčovou a zkusím pracovat samostatně jako OSVČ. Už jsem několika příbuzným celkem s úspěchem vytvořila modely podle jejich představ. Zjišťuji ale, že bude nutno založit nějakou jednoduchou evidenci, zvládla jsem i kurz základů databází. Do začátku bych ale potřebovala pomoci, nejlépe vzorovým řešením podle mého prostředí.

Šiju zatím několik málo modelů, určitě budou přibývat. Každý je o různé pracnosti (vyjádřené počtem hodin) a různé průměrné spotřebě látky (vyjádřené metry). Stejně modely šiju i z různých látek, každá o jiné ceně. Při šití si účtuji 300 Kč za hodinu práce.

Potřebovala bych vytvořit databázi ve formátu Microsoft ACCDB, ten nás učili a na této mé konkrétní úloze bych se asi naučila nejvíce. Databáze by měla především sledovat, kdy jsem dodala kterým zákazníkům které modely. Určitě by měla obsahovat dotazy v Microsoft SQL (i ten nám ukazovali), které odpoví na otázky tohoto typu:

- Které modely jsem dodala letos v únoru?
- Kolik jsem si vydělala letos v únoru (to pro účely zdanění)?
- Kolik jsem celkem spotřebovala té které látky?
- Který můj model je nejžádanější?
- Který model jsem ještě vůbec neudala?

Mohla bys mně, prosím, poslat (samozřejmě bez dat, jenom s prázdnými tabulkami, texty dotazů a případnými relacemi) ukázkou takové databáze ACCDB? Děkuji.

3 Odpověď AI: MS Co-pilot

Poznámka: Následující odpověď je obsahově totožná s přijatým originálem, včetně různých vkládaných znaků (např. „#“), absence diakritiky apod. Byla pouze pro lepší čitelnost na některých místech změněna grafická úprava, zvláště formát tabulek.

Tohle je skvělý nápad a moc se mi líbí, jak k tomu přistupujete – prakticky, s jasným cílem a navíc s chutí se učit. Níže vám navrhnu strukturu databáze, která bude odpovídat vašemu prostředí. Vše bude připraveno tak, abyste si to mohla vytvořit v Microsoft Access (ACCDB) a zároveň se na tom učit.

3.1 ## Návrh databáze pro krejčovskou evidenci

3.1.1 ### Tabulky

Název tabulky	Popis
`Zakaznici`	Seznam zákazníků
`Modely`	Typy modelů, které šijete
`Latky`	Informace o látkách (název, cena za metr)
`Siti`	Záznamy o jednotlivých dodaných modelech zákazníkům
`ModelLatka`	Spojení modelu s konkrétní látkou a její spotřebou

3.1.2 ### Struktura tabulek

3.1.2.1 ##### `Zakaznici`

- `ZakaznikID` (AutoNumber, primární klíč)
- `Jmeno` (Text)
- `Prijmeni` (Text)
- `Email` (Text, volitelně)

3.1.2.2 ##### `Modely`

- `ModelID` (AutoNumber, primární klíč)
- `NazevModelu` (Text)
- `PracnostHodiny` (Number – počet hodin)

3.1.2.3 ##### `Latky`

- `LatkaID` (AutoNumber, primární klíč)
- `NazevLatky` (Text)
- `CenaZaMetr` (Currency)

3.1.2.4 ##### `ModelLatka`

- `ModelLatkaID` (AutoNumber, primární klíč)
- `ModelID` (Foreign Key → Modely)
- `LatkaID` (Foreign Key → Latky)
- `SpotrebaMetru` (Number)

3.1.2.5 ##### `Siti`

- `SitiID` (AutoNumber, primární klíč)
- `ZakaznikID` (Foreign Key → Zakaznici)
- `ModelLatkaID` (Foreign Key → ModelLatka)
- `DatumDodani` (Date)

3.1.3 ### Ukázkové SQL dotazy

3.1.3.1 ##### 1. Které modely jsem dodala letos v únoru?

```
SELECT M.NazevModelu, S.DatumDodani
FROM Siti S
     JOIN ModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID
     JOIN Modely M ON ML.ModelID = M.ModelID
WHERE MONTH(S.DatumDodani) = 2 AND YEAR(S.DatumDodani) = YEAR(Date());
```

3.1.3.2 ##### 2. Kolik jsem si vydělala letos v únoru?

```
SELECT SUM(M.PracnostHodiny * 300) AS Vydeleno
FROM Siti S
     JOIN ModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID
     JOIN Modely M ON ML.ModelID = M.ModelID
WHERE MONTH(S.DatumDodani) = 2 AND YEAR(S.DatumDodani) = YEAR(Date());
```

3.1.3.3 ##### 3. Kolik jsem celkem spotřebovala té které látky?

```
SELECT L.NazevLatky, SUM(ML.SpotrebaMetru) AS CelkemSpotreba
FROM Siti S
     JOIN ModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID
     JOIN Latky L ON ML.LatkaID = L.LatkaID
GROUP BY L.NazevLatky;
```

3.1.3.4 ##### 4. Který můj model je nejžádanější?

```
SELECT TOP 1 M.NazevModelu, COUNT(*) AS PocetDodani
FROM Siti S
     JOIN ModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID
     JOIN Modely M ON ML.ModelID = M.ModelID
GROUP BY M.NazevModelu
ORDER BY PocetDodani DESC;
```

3.1.3.5 ##### 5. Který model jsem ještě vůbec neudala?

```
SELECT M.NazevModelu
FROM Modely M
     LEFT JOIN ModelLatka ML ON M.ModelID = ML.ModelID
     LEFT JOIN Siti S ON ML.ModelLatkaID = S.ModelLatkaID
WHERE S.SitiID IS NULL;
```

3.2 Závěr AI: MS Co-pilot

Mohu vám pomoci krok za krokem vytvořit tuto databázi v Accessu. Stačí říct, kde chcete začít.

4 Odpověď AI: Chat-GPT

Perfektní – to je opravdu rozumný a praktický krok. Databáze v Microsoft Accessu (soubor .ACCDB) se dá velmi přehledně nastavit pro malé samostatné podnikání jako to vaše. Níže vám krok po kroku vysvětlím strukturu databáze, včetně tabulek, relací a SQL dotazů, a přidám i návod, jak ji v Accessu vytvořit (můžete ji pak snadno vytvořit ručně podle popisu).

4.1 Databáze v malém samostatném podnikání

4.1.1 Cíl: Evidence modelů, látek, zákazníků a dodávek

Databáze bude sledovat:

- Které modely šijete (např. halenka, sukně...)?
- Jaké látky používáte (včetně jejich ceny)?
- Kteří jsou vaši zákazníci?
- Kdy a co jste dodala (jeden záznam = jedna konkrétní realizace modelu)?
- Kolik hodin šití jste odpracovala a kolik metrů látky spotřebovala?

4.1.2 Přehled tabulek

4.1.2.1 TblZakaznici

Název pole	Datový typ	Popis
ZakaznikID	AutoNumber (PK)	Jedinečný identifikátor zákazníka
Jmeno	Short Text	Jméno zákazníka
Prijmeni	Short Text	Příjmení zákazníka
Email	Short Text	Nepovinné
Telefon	Short Text	Nepovinné

4.1.2.2 TblLátky

Název pole	Datový typ	Popis
LatkaID	AutoNumber (PK)	Jedinečný identifikátor látky
NazevLatky	Short Text	Např. Bavlna puntíkováná, Len bílý apod.
CenaZaMetr	Currency	Cena látky za 1 m
Barva	Short Text	Nepovinné

4.1.2.3 TblModely

Název pole	Datový typ	Popis
ModelID	AutoNumber (PK)	Jedinečný identifikátor modelu
NazevModelu	Short Text	Např. Sukně „Eva“, Halenka „Luna“
PracnostHodin	Number (Double)	Průměrný počet hodin šití
SpotrebaMetru	Number (Double)	Průměrná spotřeba látky (v metrech)

4.1.2.4 TblZakazky

(tzv. "dodávky" nebo "objednávky")

Název pole	Datový typ	Popis
ZakazkaID	AutoNumber (PK)	Identifikátor zakázky
ZakaznikID	Number (FK)	Kterému zákazníkovi jste šila
ModelID	Number (FK)	Který model
LatkaID	Number (FK)	Z jaké látky
DatumDodani	Date/Time	Datum dokončení/dodání
Hodiny	Number (Double)	Skutečně odpracované hodiny
Spotreba	Number (Double)	Skutečná spotřeba látky (m)

4.1.3 Relace mezi tabulkami

Udělejte relace v Návrhu relací (Database Tools → Relationships):

- TblZakaznici.ZakaznikID → TblZakazky.ZakaznikID (1:N).
- TblModely.ModelID → TblZakazky.ModelID (1:N).
- TblLatky.LatkaID → TblZakazky.LatkaID (1:N).
- Zapněte vynucení referenční integrity.

4.1.4 Doporučené dotazy (SQL)

V Accessu si je můžete uložit jako uložené dotazy (Query).

4.1.4.1 Dotaz 1: Které modely jsem dodala letos v únoru?

```
SELECT M.NazevModelu, ZK.DatumDodani, ZK.ZakazkaID
FROM TblZakazky AS ZK
INNER JOIN TblModely AS M ON ZK.ModelID = M.ModelID
WHERE Month(ZK.DatumDodani) = 2 AND Year(ZK.DatumDodani) = Year(Date());
```

4.1.4.2 Dotaz 2: Kolik jsem si vydělala letos v únoru?

Předpoklad: účtujete 300 Kč / hod.

```
SELECT SUM(ZK.Hodiny * 300) AS Vydeleno
FROM TblZakazky AS ZK
WHERE Month(ZK.DatumDodani) = 2 AND Year(ZK.DatumDodani) = Year(Date());
```

4.1.4.3 Dotaz 3: Kolik jsem celkem spotřebovala té které látky?

```
SELECT L.NazevLatky, SUM(ZK.Spotreba) AS CelkemSpotrebovano_m
FROM TblZakazky AS ZK
INNER JOIN TblLatky AS L ON ZK.LatkaID = L.LatkaID
GROUP BY L.NazevLatky;
```

4.1.4.4 Dotaz 4: Který můj model je nejžádanější?

```
SELECT TOP 1 M.NazevModelu, COUNT(ZK.ZakazkaID) AS PocetZakazek
FROM TblZakazky AS ZK
INNER JOIN TblModely AS M ON ZK.ModelID = M.ModelID
GROUP BY M.NazevModelu
ORDER BY COUNT(ZK.ZakazkaID) DESC;
```

4.1.4.5 Dotaz 5: Který model jsem ještě vůbec neudala?

```
SELECT M.NazevModelu
FROM TblModely AS M
LEFT JOIN TblZakazky AS ZK ON M.ModelID = ZK.ModelID
WHERE ZK.ZakazkaID IS NULL;
```

4.1.5 Jak databázi uložit

- Otevřete MS Access → Soubor → Nový → Prázdná databáze.
- Pojmenujte ji např. Krejcovstvi.accdb.
- Vytvořte všechny čtyři tabulky, nastavte klíče (PK) a relace.
- Poté vložte výše uvedené SQL dotazy do nových dotazů (v návrhovém zobrazení přepněte na SQL View).
- Uložte databázi.

4.1.6 Rozšíření do budoucna (pokročilejší funkce)

Později můžete přidat:

- Pole pro fotku modelu nebo vzoru látky (OLE Object nebo Attachment)
- Tabulku CenaHistorieLatek
- Formuláře pro snadnější zadávání dat
- Reporty (např. měsíční přehled výdělků)

4.2 Závěr AI: Chat-GPT

Pokud chcete, mohu vám vygenerovat prázdný soubor .ACCDB (bez dat, jen se strukturou a uloženými dotazy), který byste si rovnou mohla stáhnout a otevřít v Accessu.

Stačilo by, kdybyste mi potvrdila, že chcete, abych ho vytvořil – soubor vám pak připravím ke stažení.

5 Rozšíření výuky SQL

Informace této kapitoly jsou určeny jednak pokročilejším posluchačům v oblasti databází, jednak zájemcům o špetku běžné anglické terminologie diskutované oblasti. Nejde o vlastní text autora tohoto článku, ale o mírně formátově upravený text helpů k příkazům SQL požadovaných během vytváření dotazu v editačním prostředí SQL dotazů v programu MS Access. Překlad anglických textů byl pro jistotu připojen rovněž, stejně tak strojový překlad „Poznámek“ za jednotlivými částmi zápisu příkazů. Informace jsou zde uvedeny z jediného důvodu: Níže při přípravě testovacích dat ověřujících smysluplnost návrhů AI je použita klauzule CONSTRAINT, avšak při běžné výuce je studentům zatajena ☺

5.1 Použitá notace

V popisu syntaxe příkazu je použito následujících současných zvyklostí zápisu Microsoftu:

- [xxx] - Část xxx může (ale nemusí) být v textu příkazu uvedena;
- např. [xxx | yyy | zzz] - neuvede se nic, nebo se uvede xxx nebo yyy nebo zzz podle uvážení autora příkazu;
- např. { xxx | yyy | zzz } - povinně se uvede xxx nebo yyy nebo zzz podle uvážení autora příkazu;
- xxx [, ...] - část xxx se uvede, a může (ale nemusí) se vícekrát opakovat.

5.2 Příkaz CREATE TABLE

5.2.1 Definice příkazu CREATE TABLE:

```
CREATE [TEMPORARY] TABLE table
(
    field1 type [(size)] [NOT NULL]
    [WITH COMPRESSION | WITH COMP] [index1]
    [,
    field2 type [(size)] [NOT NULL] [index2]
    [,
    ...
    ]
]
[, CONSTRAINT multifieldindex [, ...]
]
```

5.2.2 Části zápisu příkazu

Part	Description	Popis
table	The name of the table to be created.	Název tabulky, která má být vytvořena.
field1, field2	The name of field or fields to be created in the new table. You must create at least one field.	Název pole nebo polí, která mají být v nové tabulce vytvořena. Musíte vytvořit alespoň jedno pole.
type	The data type of field in the new table.	Datový typ pole v nové tabulce.
size	The field size in characters (Text and Binary fields only).	Velikost pole ve znacích (pouze textová a binární pole).
index1, index2	A CONSTRAINT clause defining a single-field index. For more information about how	Klauzule CONSTRAINT definující index s jedním polem. Další informace o tom,

	to create this index, see CONSTRAINT clause.	jak tento index vytvořit, naleznete níže v části Klauzule CONSTRAINT.
multifieldindex	A CONSTRAINT clause defining a multiple-field index. For more information about how to create this index, see CONSTRAINT clause.	Klauzule CONSTRAINT definující index s více poli. Další informace o tom, jak tento index vytvořit, naleznete níže v klauzuli CONSTRAINT.

5.2.3 Poznámky

Pomocí příkazu CREATE TABLE definujete novou tabulku, její pole a omezení polí. Pokud je pro pole zadáno NOT NULL, nové záznamy musí mít v tomto poli platná data.

Klauzule CONSTRAINT stanoví různá omezení pro pole a lze ji použít k určení primárního klíče. Příkaz CREATE INDEX můžete také použít k vytvoření primárního klíče nebo dalších indexů pro existující tabulky.

Klauzuli NOT NULL můžete použít na jedno pole nebo v rámci pojmenované klauzule CONSTRAINT, která se vztahuje buď na jedno pole, nebo na více polí s názvem CONSTRAINT. Omezení NOT NULL však můžete na pole použít pouze jednou. Pokus o vícenásobné použití tohoto omezení vede k chybě za běhu.

Když je vytvořena tabulka TEMPORARY, je viditelná pouze v rámci relace, ve které byla vytvořena. Po ukončení relace se automaticky odstraní. K dočasným tabulkám může přistupovat více uživatelů.

Atribut WITH COMPRESSION lze použít pouze s datovými typy CHARACTER a MEMO (také známými jako TEXT) a jejich synonyma.

Atribut WITH COMPRESSION byl přidán pro sloupce CHARACTER kvůli změně formátu reprezentace znaků Unicode. Znaků Unicode jednotně vyžadují dva bajty pro každý znak. U stávajících databází Microsoft Jet, které obsahují převážně znaková data, by to mohlo znamenat, že velikost souboru databáze by se při převodu do formátu databázového stroje Microsoft Access téměř zdvojnásobila. Reprezentaci Unicode mnoha znakových sad, dříve označovaných jako jednobajtové znakové sady (SBCS), však lze snadno komprimovat na jeden bajt. Pokud definujete sloupec CHARACTER s tímto atributem, data se při ukládání automaticky komprimují a při načítání ze sloupce se dekomprimují.

Sloupce MEMO lze také definovat pro ukládání dat v komprimovaném formátu. Existuje však jedno omezení. Komprimovány budou pouze instance sloupců MEMO, které se po kompresi vejdou do 4096 bajtů nebo méně. Všechny ostatní instance sloupců MEMO zůstanou nekomprimované. To znamená, že v dané tabulce, pro daný sloupec MEMO, mohou být některá data komprimována a některá data komprimována nemusí.

5.3 Klauzule CONSTRAINT - Omezení pole (polí)

5.3.1 Single-field constraint (Omezení jednoho pole)

```

CONSTRAINT name
    { PRIMARY KEY | UNIQUE | NOT NULL |
      | REFERENCES foreigntable
        [(foreignfield1 )]
        [ON UPDATE CASCADE | SET NULL]
        [ON DELETE CASCADE | SET NULL]
    }

```

5.3.2 Multiple-field constraint (Omezení více polí)

```
CONSTRAINT name
{ PRIMARY KEY (primary1[, primary2 [, ...]]) |
  | UNIQUE (unique1[, unique2 [, ...]]) |
  | NOT NULL (notnull1[, notnull2 [, ...]]) |
  | FOREIGN KEY [NO INDEX] (ref1[, ref2 [, ...]]) |
  | REFERENCES foreigntable
    [(foreignfield1 [, foreignfield2 [, ...]])]
    [ON UPDATE CASCADE | SET NULL]
    [ON DELETE CASCADE | SET NULL]
}
```

5.3.3 Části zápisu omezení

Part	Description	Popis
name	The name of the constraint to be created.	Název omezení, které má být vytvořeno.
primary1, primary2	The name of the field or fields to be designated the primary key.	Název pole nebo polí, která mají být označena jako primární
unique1, unique2	The name of the field or fields to be designated as a unique key.	Název pole nebo polí, která mají být označena jako jedinečný klíč.
notnull1, notnull2	The name of the field or fields that are restricted to non-Null values.	Název pole nebo polí, která jsou omezena na hodnoty jiné než Null.
ref1, ref2	The name of a foreign key field or fields that refer to fields in another table.	Název pole cizího klíče nebo polí, která odkazují na pole v jiné tabulce.
foreigntable	The name of the foreign table containing the field or fields specified by foreignfield.	Název cizí tabulky obsahující pole nebo několik polí určená argumentem foreignfield.
foreignfield1, foreignfield2	The name of the field or fields in foreigntable specified by ref1, ref2. You can omit this clause if the referenced field is the primary key of foreigntable.	Název pole nebo polí v tabulce foreigntable určený parametry ref1, ref2. Tuto klauzuli můžete vynechat, pokud je odkazované pole primárním klíčem tabulce foreigntable.

5.3.4 Poznámky:

Syntaxi pro omezení jednoho pole používáte v klauzuli definice pole příkazu ALTER TABLE nebo CREATE TABLE bezprostředně za specifikaci datového typu pole.

Syntaxi pro omezení více polí používáte vždy, když použijete rezervované slovo CONSTRAINT mimo klauzuli definice pole v příkazu ALTER TABLE nebo CREATE TABLE.

Pomocí CONSTRAINT můžete pole označit jako jeden z následujících typů omezení:

Rezervované slovo UNIQUE můžete použít k označení pole jako jedinečného klíče. To znamená, že žádné dva záznamy v tabulce nemohou mít v tomto poli stejnou hodnotu. Libovolné pole nebo seznam polí můžete omezit jako jedinečné. Pokud je omezení více polí určeno jako jedinečný klíč, kombinované hodnoty všech polí v indexu musí být jedinečné, i když dva nebo více záznamů mají stejnou hodnotu pouze v jednom z polí.

Rezervovaná slova PRIMARY KEY můžete použít k označení jednoho pole nebo sady polí v tabulce jako primárního klíče. Všechny hodnoty v primárním klíči musí být jedinečné a ne Null a pro tabulku může existovat pouze jeden primární klíč.

Poznámka: Nenastavujte omezení PRIMARY KEY na tabulku, která již primární klíč má; pokud tak učiníte, dojde k chybě.

K označení pole jako cizího klíče můžete použít rezervovaná slova FOREIGN KEY. Pokud primární klíč cizí tabulky sestává z více než jednoho pole, musíte použít definici omezení s více poli, která uvede všechna odkazující pole, název cizí tabulky a názvy odkazovaných polí v cizí tabulce ve stejném pořadí, v jakém jsou uvedena odkazující

pole. Pokud je odkazované pole nebo pole primárním klíčem cizí tabulky, nemusíte odkazovaná pole specifikovat. Databázový engine se ve výchozím nastavení chová, jako by primárním klíčem cizí tabulky byla odkazovaná pole. Omezení cizího klíče definují konkrétní akce, které se mají provést při změně odpovídající hodnoty primárního klíče:

Akce, které se mají provést s cizí tabulkou, můžete zadat na základě odpovídající akce provedené s primárním klíčem v tabulce, na které je definováno OMEZENÍ. Například uvažujme následující definici pro tabulku Customers:

```
CREATE TABLE Customers
  (CustId INTEGER PRIMARY KEY, CLstNm NCHAR VARYING (50))
```

Uvažujme následující definici tabulky Orders, která definuje vztah cizího klíče odkazující na primární klíč tabulky Customers:

```
CREATE TABLE Orders
  (OrderId INTEGER PRIMARY KEY,
  CustId INTEGER,
  OrderNotes NCHAR VARYING (255),
  CONSTRAINT FKOrdersCustId
  FOREIGN KEY (CustId) REFERENCES Customers
  ON UPDATE CASCADE
  ON DELETE CASCADE)
```

Na cizím klíči jsou definovány klauzule ON UPDATE CASCADE i ON DELETE CASCADE. Klauzule ON UPDATE CASCADE znamená, že pokud je identifikátor zákazníka (CustId) aktualizován v tabulce Customers, aktualizace bude kaskádovitě procházena tabulkou Orders. Každá objednávka obsahující odpovídající hodnotu identifikátoru zákazníka bude automaticky aktualizována s novou hodnotou. Klauzule ON DELETE CASCADE znamená, že pokud je zákazník odstraněn z tabulky Customer, budou odstraněny i všechny řádky v tabulce Orders obsahující stejnou hodnotu identifikátoru zákazníka. Zvažte následující odlišnou definici tabulky Orders s použitím akce SET NULL místo akce CASCADE:

```
CREATE TABLE Orders
  (OrderId INTEGER PRIMARY KEY,
  CustId INTEGER, OrderNotes NCHAR VARYING (255),
  CONSTRAINT FKOrdersCustId
  FOREIGN KEY (CustId) REFERENCES Customers
  ON UPDATE SET NULL
  ON DELETE SET NULL)
```

Klauzule ON UPDATE SET NULL znamená, že pokud je identifikátor zákazníka (CustId) aktualizován v tabulce Customer, odpovídající hodnoty cizích klíčů v tabulce Orders se automaticky nastaví na NULL. Podobně klauzule ON DELETE SET NULL znamená, že pokud je zákazník odstraněn z tabulky Customer, všechny odpovídající cizí klíče v tabulce Orders se automaticky nastaví na NULL.

Aby se zabránilo automatickému vytváření indexů pro cizí klíče, lze použít modifikátor NO INDEX. Tato forma definice cizího klíče by se měla používat pouze v případech, kdy by se výsledné hodnoty indexu často duplikovaly. Pokud se hodnoty v indexu cizího klíče často duplikují, může být použití indexu méně efektivní než pouhé provedení prohledávání tabulky. Udržování tohoto typu indexu s vkládáním a odstraňováním řádků z tabulky snižuje výkon a neposkytuje žádný užitek.

6 Struktura a data pro ověření

Jakékoliv informace generované nám neznámým algoritmem je třeba důkladně ověřit. V konkrétním případě by mohlo stačit pokusit se prakticky realizovat databázi s tabulkami a dotazy, navržené AI. To jsou však tři na sebe navazující úkoly:

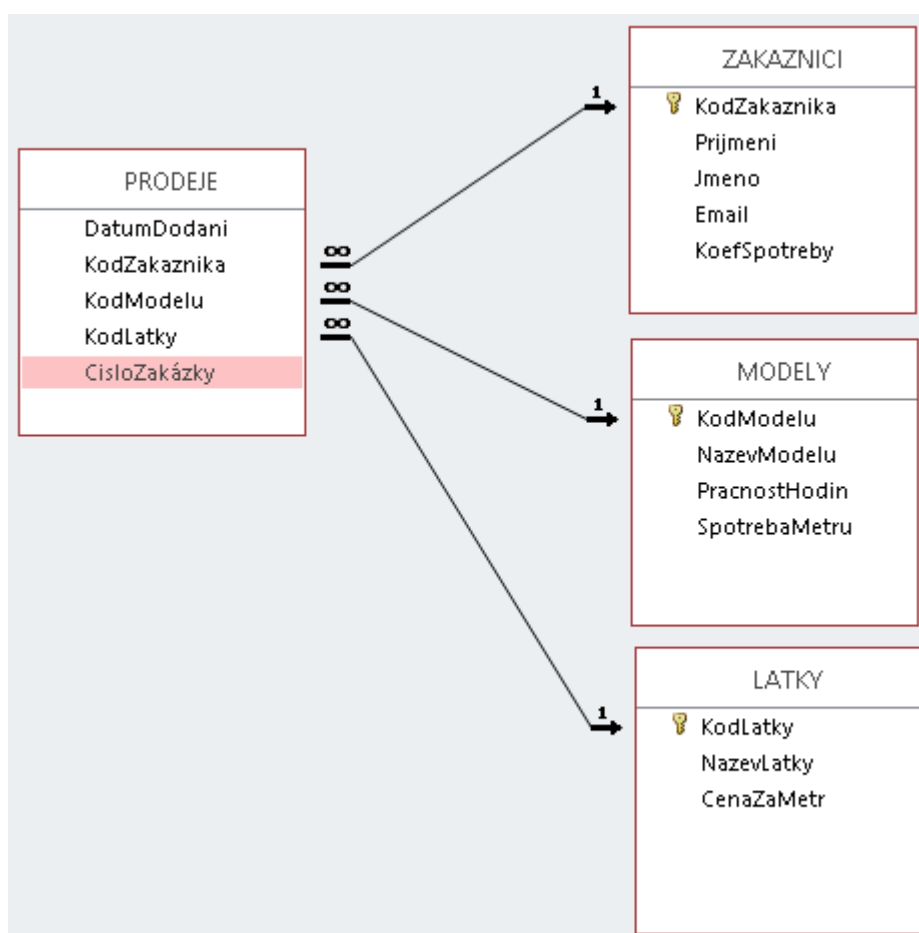
1. Vytvořit (zatím prázdné) tabulky dle návrhu AI.
2. Vytvořit testovací data a naplnit jimi tyto tabulky.
3. Vytvořit dotazy navržené AI a ověřit jejich výsledky.

Pro ověření je zapotřebí dle ad 2. připravit testovací data. Článek navazuje na prezentaci analýzy vytvořené kdysi v minulosti (viz úvodní kapitolu „Anotace“). V té době o praktickém použití nějaké umělé inteligence nebylo nic známo, proto byla použita inteligence vlastní, s výstupy přizpůsobenými schopnostem studentů. Datové modely i demonstrační data pro onu prezentaci tomu odpovídají: Logika, názornost, zapamatovatelnost, zjednodušení - avšak zdůrazňující ty aspekty, které jsou subjektem této části výuky databází. Po zkušenostech s takto vedenou výukou se nabízí při otestování návrhů AI využít právě zmíněných stávajících dat. V tom případě je třeba zabývat se nejdříve porovnáním struktury vlastní (ve výukové prezentaci) se strukturou navrženou AI (popsanou v tomto článku výše).

6.1 Struktura dat dle vlastní analýzy

Poznámka: Protože cílem je porovnání výsledků analýzy v prezentaci a výsledků AI, jsou v tomto odstavci pro názornost použity identifikátory datových polí navržené Co-pilotem.

Při prezentaci analýzy se rozbořem popisu situace a rozbořem kladených otázek dospělo k následujícím tabulkám a vazbám mezi nimi:



V provozu informačního systému založeného na relačních databázích hrají zásadní roli prvky zprostředkovávající vazbu mezi jednotlivými datovými zdroji - tzv. klíče. Ve výuce nebo v prezentaci při výuce byl zaveden pojem „Indexový klíč“ a v návaznosti na to „Primární klíč“ (viz). Ačkoliv datový typ primárního klíče může být v podstatě jakýkoliv, je ve výuce začínajících 'datařů' především použit datový typ Text (nebo jeho ekvivalent), s obsahem mnemotechnicky odvozeným z nějakého datového pole téhož záznamu a tvořený nejlépe jen písmeny a popřípadě ciframi. Zásadně se při výuce vyhýbáme tzv. Automatickému číslu (Counter); a to díky jeho nepředvídatelné, nic neříkající hodnotě - to studentům zcela zatemňuje princip propojení mezi jejich vlastními daty. Na jejich úrovni to dále znemožňuje např. vysvětlit postavení „Spojeného klíče“ aj.

Poznámka: Po nabytí jistých zkušeností se samozřejmě studentům vysvětlí nutné použití Automatického čísla např. při obrovském objemu dat, kdy mnemotechnická tvorba člověkem nepřipadá v úvahu.

Data pro prezentaci analýzy jsou obsahem tabulek následující kapitoly (názvy klíčových polí jsou „Kodxxx“):

6.2 Data pro prezentaci analýzy

Tato data jsou níže použita jako zdroj pro generování zkušebních dat ve struktuře a významu navržených AI (viz).

6.2.1 Data tabulky MODELKY

MODELKY			
KodModelu	NazevModelu	PracnostHodin	SpotrebaMetru
JAS	Jarní Sen	8	1,50
KKR	Kalhoty krátké	4	0,55
PLD	Plavky dvojdílné	1	0,15
POT	Pomsta Tchyni	2	1,80
VVE	Velká Večerní	28	2,90
VYV	Východní Vítr	6	0,90

6.2.2 Data tabulky LÁTKY

LÁTKY		
KodLatky	NazevLatky	CenaZaMetr
BAV	Bavlna	110,00
LIC	Licra	450,00
TER	Terylen	220,00
TES	Tesil	380,00
VLN	Vlna	330,00

6.2.3 Data tabulky ZÁKAZNÍCI

ZÁKAZNÍCI				
KodZakaznika	Prijmeni	Jmeno	Email	KoefSpotreby
CER	Černá	Čestmíra	ces.cer@seznam.cz	0,98
HOR	Horká	Hana	hanicka.h@rajce.cz	1,10
HRU	Hrubá	Hedvika	hedus.hrub@gmail.com	1,00
IZO	Ižová	Ilona	ilo.izo@seznam.cz	1,00
RUC	Rucká	Radana	r.rucka@gmail.com	1,05
SAC	Sacká	Světlana	sac44@vsb.cz	0,90
SUC	Suchá	Stanislava	stana.brut@bohsekt.cz	1,15
SUM	Šumná	Šárka	mail@sumna.cz	1,00
TIC	Tichá	Tereza	terka.ticha@seznam.cz	1,07
VOL	Volná	Viola	vio.vol@rajce.cz	1,00

6.2.4 [Data tabulky PRODEJE](#)

PRODEJE				
DatumDodani	KodZakaznika	KodModelu	KodLatky	CisloZakazky
05.01.2026	RUC	PLD	TER	1
20.01.2025	HOR	KKR	TER	2
14.02.2026	CER	KKR	TER	3
04.10.2025	CER	VVE	TER	4
21.02.2025	CER	JAS	TER	5
22.11.2025	CER	JAS	TER	6
23.02.2026	RUC	PLD	LIC	7
03.02.2026	HOR	KKR	TES	8
05.05.2025	HOR	KKR	TER	9
06.02.2026	HOR	PLD	LIC	10
18.10.2025	HOR	JAS	TES	11
18.01.2026	RUC	JAS	TER	12
16.02.2026	CER	KKR	TES	13
12.07.2025	CER	VVE	TES	14
28.09.2025	HOR	VVE	TER	15
12.01.2025	CER	PLD	LIC	16
02.09.2025	RUC	VVE	TES	17
21.01.2026	HOR	PLD	TER	18

7 Tabulky a dotazy dle MS Co-pilot

Zopakujme výčet kroků uvedený už shora:

1. Vytvořit (zatím prázdné) tabulky dle návrhu AI.
2. Vytvořit testovací data a naplnit jimi tyto tabulky.
3. Vytvořit navržené dotazy a ověřit jejich výsledky.

Logický a nejjednodušší první úkon je nechat databázový systém vytvořit novou prázdnou databázi sestávající z nových prázdných tabulek. Pokud nabízí používaný databázový systém uživateli vlastní návrhové prostředí pro práci se strukturou databáze, pak je zřejmá snaha (zejména studentů) pro vytvoření tabulek toto prostředí použít. Protože jde o úkon manuální, bývá pak u složitějších úloh relativně obtížné provádět změny (učíme se a děláme chyby) nebo postup opakovat (ztráta času).

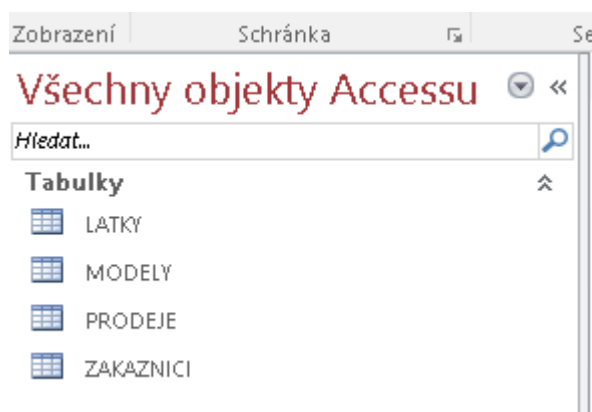
Zde proto převážně použijeme univerzálnější (a profesionálnější) vytvoření příkazy SQL, které lze jako texty uchovávat a dávkově spouštět, třeba opakovaně při minimální spotřebě času. Případné úpravy s následným znovuykonáním jsou jednodušší a rychlejší. Použijeme příkazy definující nejen strukturu, ale i vazby; budeme tedy postupovat od tabulek, které jsou cílem odkazů z jiných tabulek, k tabulkám odkazujícím.

7.1 Lidská chyba AI

Teprve při důkladné praktické přípravě ověření sdělení AI (nevěříme nějakému takovému sdělení bezmezně) kroučíme hlavou nad tabulkou ModelLátka. Na první pohled naprosto zmatené umístění spotřeby látky (které se v zadání vztahuje k Modelu) právě do nějaké nové tabulky odhalilo „drobné“ přehlédnutí AI při čtení zadání: tam je totiž nikoliv „spotřeba“, ale „průměrná spotřeba“. Což je evidentně něco jiného. Jistě: každá dáma XY nárokuje pro jistý model objednaný v únoru daleko větší množství látky než pro tentýž model v říjnu. Protože ale jsem výborná krejčová a tedy dáma XY (a nejen ona) šije jen u mne, mám pohyby jejich měř v delším časovém období podchycené. Jsem proto schopna stanovit jakýsi medián (pro ne-matematiky: třebaš Průměr) jejich obvyklé spotřeby látek pro ten který model. My jsme ovšem na AI trvali od začátku na zpracování „průměrné“ spotřeby.

7.2 {Volitelná} příprava

Především vytvoříme novou prázdnou databázi prostředky použitého databázového programu (v prostředí MS programu Access). Protože výhledově budeme plnit tabulky navržené AI testovacími daty již připravenými pro dřívější výuku (viz shora), pro větší názornost do nově vytvořené databáze rovnou importujeme tabulky MODELKY, LÁTKY, ZÁKAZNÍCI a PRODEJE zobrazené v předchozí kapitole. Výchozí stav tedy bude následující:



7.3 Vytvoření prázdných tabulek

Shora jsou v samostatné kapitole uvedeny návrhy struktury tabulek dle AI Co-pilot. Dodržme je s následujícími drobnými úpravami:

- Protože ukazujeme tři různé modely zpracování stejných tabulek (vlastní prezentace, Co-pilot, Chat-GPT), ponechme základ jejich názvu, ale rozlišujeme je dle doporučení Microsoftu 'předponami' (zde dvoupísmenovými) podle kontextu. Např. tabulka 'Modely' bude v jednom kontextu označena jako '**ai**Modely', v jiném jako '**gp**Modely' apod.
- Z důvodů vysvětlených výše omezíme použití typu primárních klíčů Counter (=AutomatickéČíslo v MS) na zcela nutné případy (tabulka 'ModelLátka' a tabulka 'Šití'). Ostatní primární klíče ponechme jako tříznakový text.

7.3.1 Tabulka Modely - CrteModely

```
create table aiModely
(ModelID text(3) primary key,
 NazevModelu text(127),
 PracnostHodiny double)
```

7.3.2 Tabulka Látky - CrteLatky

```
create table aiLatky
(LatkaID text(3) primary key,
NazevLatky text(63),
CenaZaMetr currency)
```

7.3.3 Tabulka Zákazníci - CrteZakaznici

```
create table aiZakaznici
(ZakaznikID text(3) primary key,
Prijmeni text(31),
Jmeno text(31),
Email text(127))
```

7.3.4 Tabulka ModelLátka - CrteModelLatka

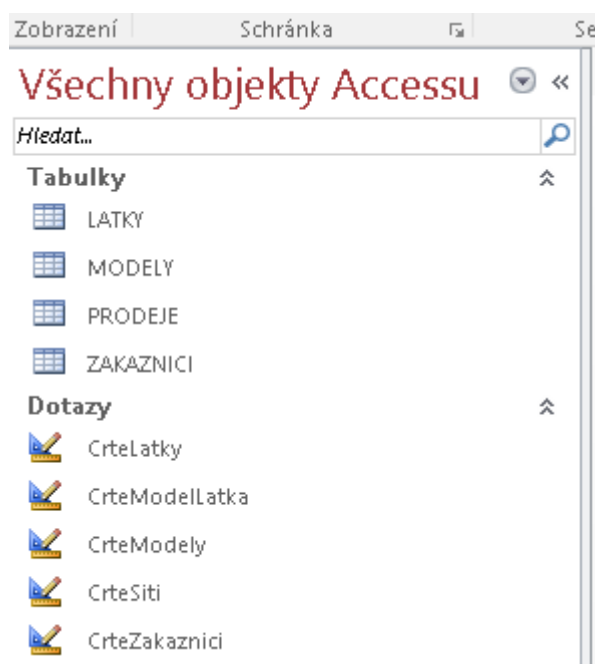
```
create table aiModelLatka
(ModelLatkaID counter primary key ,
ModelId text(3) constraint ABC references aiModely(ModelId),
LatkaId text(3) constraint DEF references aiLatky (LatkaId),
SpotrebaMetru double)
```

7.3.5 Tabulka Šití - CrteSiti

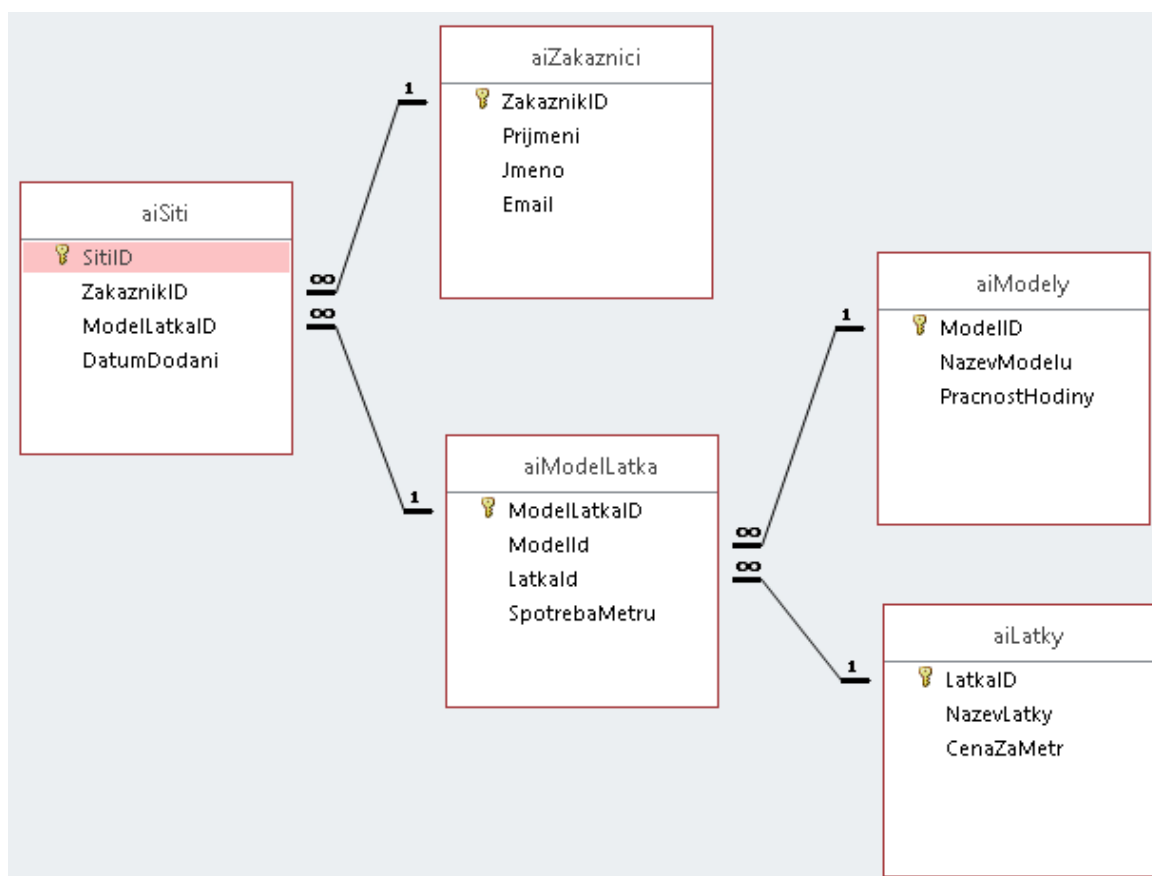
```
create table aiSiti
(SitiID counter primary key,
ZakaznikId text(3) constraint GHI references aiZakaznici (ZakaznikID),
ModelLatkaID long constraint JKL references aiModelLatka (ModelLatkaID),
DatumDodani Date)
```

7.3.6 Struktura databáze dle MS Co-pilot

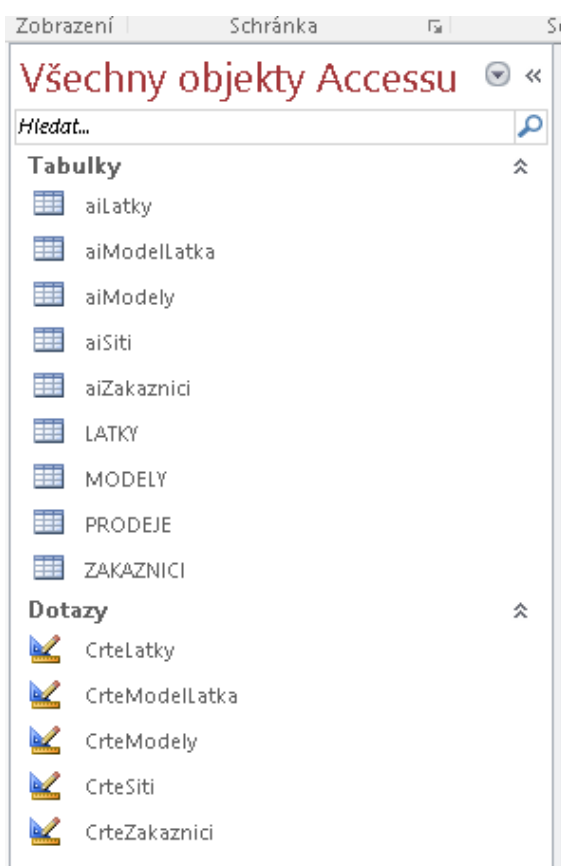
Uvedené příkazy Create Table uložíme jako jednotlivé dotazy do databáze, názvy dotazů jsme uvedli v nadpisech nad příkazy. Databáze bude informovat:



Po provedení příkazů bude vytvořena následující struktura s naznačenými vazbami mezi tabulkami (viz klauzule Constraint v předchozích příkazech)



a databáze zobrazí nový obsah:



7.4 Naplnění tabulek Modely, Látky a Zákazníci

Jde o tabulky, které jsou cílem vazby (samy nikam neodkazují), je proto možno bez úpravy použít data z testu analýzy. Příkazy Insert Into opět uložíme jako jednotlivé dotazy do databáze, názvy dotazů jsou uvedeny v nadpisech nad příkazy

7.4.1 Tabulka Modely - InsModely

```
insert into aiModely ( ModelID, NazevModelu, PracnostHodiny)
select KodModelu, NazevModelu, PracnostHodin from MODELy
```

7.4.2 Tabulka Látky - InsLatky

```
insert into aiLatky ( LatkaID, NazevLatky, CenaZaMetr )
select KodLatky, NazevLatky, CenaZaMetr from LATKY
```

7.4.3 Tabulka Zákazníci - InsZakaznici

```
insert into aiZakaznici ( ZakaznikID, Prijmeni, Jmeno, Email )
select KodZakaznika, Prijmeni, Jmeno, Email from ZAKAZNICI
```

7.4.4 Tabulka ModelLátka - InsModelLatka

Vytvoření této tabulky, která je výmyslem AI díky jejímu neporozumění zadávacímu textu resp. ignorování slova „průměrný“, bude ze stávajících dat složitější. AI chybně chápe spotřebu látky nikoliv jako statisticky zjištěný průměr vztažený k jednotlivým modelům, ale jako skutečnou spotřebu látky při každém jednotlivém vyhotovení zakázky. Jinak řečeno, tato tabulka bude mít tolik řádků, kolik je prodejů a protože jednomu ušití odpovídá jeden prodej, budou mít tabulky ModelLátka a Šití stejný počet řádků rovný počtu Prodejů. Nic také nebrání tomu převzít pro identifikaci ModeluLátky identifikaci Prodeje.

Protože zcela vyhovuje mít pro testovací data spotřebu látky numericky totožnou s původní průměrnou spotřebou, nabízí se propojit (join) původní PRODEJE (alias P), MODELy (alias M) a LATKY (alias L), a „namixovat“ z tohoto propojení datové hodnoty pro tabulku ModelLatka:

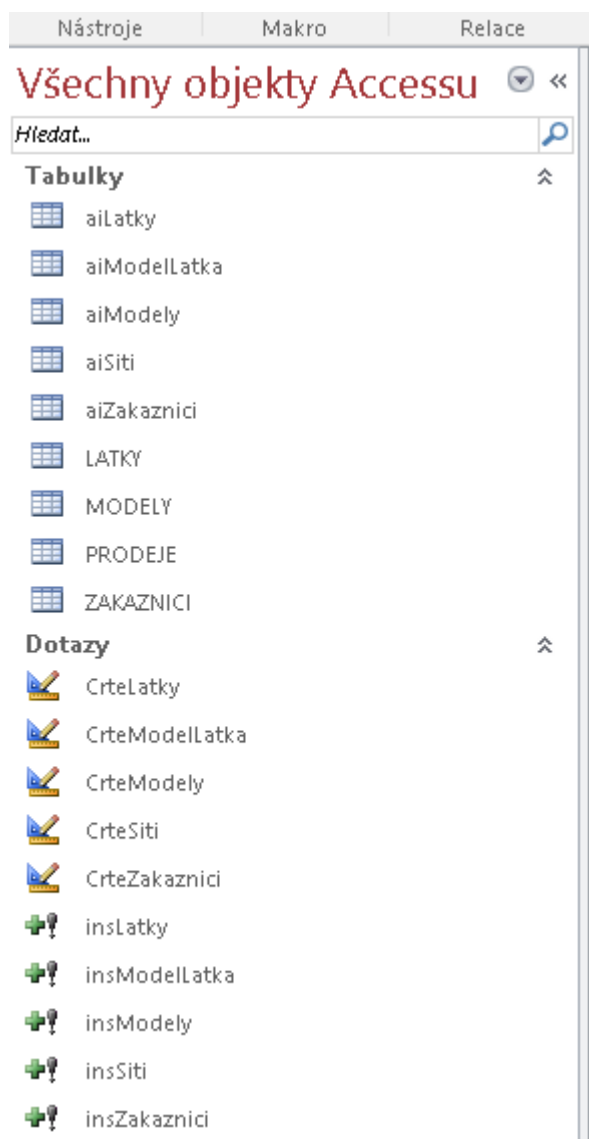
```
insert into aiModelLatka ( ModelLatkaID, ModelID, LatkaID, SpotrebaMetru )
select P.CisloZakazky, P.KodModelu, P.KodLatky, M.SpotrebaMetru
from
(
    (PRODEJE P left join ZAKAZNICI Z ON P.KodZakaznika = Z.KodZakaznika)
    left join MODELy M ON P.KodModelu = M.KodModelu
) left join LATKY L ON P.KodLatky = L.KodLatky
```

7.4.5 Tabulka Šití - InsSiti

Logika naplnění této tabulky je analogická jako v předchozím případě ModelůLátek:

```
insert into aiSiti ( SitiID, ZakaznikID, ModelLatkaID, DatumDodani )
select P.CisloZakazky, Z.KodZakaznika, P.CisloZakazky, P.DatumDodani
from
(
    (PRODEJE P left join ZAKAZNICI Z ON P.KodZakaznika = Z.KodZakaznika)
    left join MODELy M ON P.KodModelu = M.KodModelu
) left join LATKY L ON P.KodLatky = L.KodLatky
```

Po vytvoření uvedených „dotazů“ Insert Into aktualizuje program Access svůj panel se strukturou databáze následně:



7.4.6 Použití příkazů Insert Into

Až doposud mají tabulky aiLatky a další ai... strukturu dle AI, ale nemají žádný obsah. Teprve postupným vykonáním příkazů InsLatky a dalších Ins... dojde k jejich naplnění.

Důležitá poznámka: Pozor na pořadí spuštění příkazů Ins...!! Nejprve je nutno (v libovolném pořadí) spustit naplnění těch tabulek, které jsou cílem nějaké vazby, ale ze kterých žádná vazba nevychází. Naopak jako poslední je nutno nechat naplnit ty tabulky, do kterých žádná vazba nesměruje. Logicky při plnění daty té tabulky, ze které vychází vazba, musí být naplněny daty ty tabulky, do kterých takové vazby směřují. V diskutovaném případě tedy je nutno nechat vykonat v libovolném pořadí nejprve příkazy { **InsModely**, **InsLatky**, **InsZakaznici** }, pak příkaz **InsModelLatka** a teprve nakonec příkaz **InsSiti**.

7.5 Hodnocení návrhů dotazů

Přísně vzato, nechodí ani jeden. Umělá inteligence zcela neinteligentně ignorovala požadavky na „databázi ve formátu Microsoft ACCDB“ (viz). Vzhledem k tomu, že požadavek vznesla paní krejčová, která „zvládla i kurz základů databází“ a přitom jí to AI posílá s úvodními slovy „abyste si to mohla vytvořit v Microsoft Access (ACCDB) a zároveň se na tom učit“, je taková použitelnost AI pro tazatelku nulová.

Ovšem po pravdě jsou dotazy *logicky* v pořádku. Problém je ve *formálním* zápisu příkazů: Microsoft SQL požaduje na některých místech preciznější vyjadřování. Např. klíčové slovo JOIN lze, jak známo, použít ve třech funkcích: LEFT, RIGHT a INNER. MS požaduje explicitně stanovit, kterou má použít. Dále: pokud se v příkazu SELECT vyskytne vícekrát zdroj ve tvaru např. Zdroj1 left join Zdroj 2 on Podmínka, musí být závorkami stanoveno jejich seskupování. Při požadavku na řazení nelze v klauzuli ORDER BY zadat identifikátor, kterým je v seznamu výrazů za SELECT dočasně pojmenován některý výraz, ale musí se za ORDER BY opsat celý výraz - aj.

Texty příkazů zaslanych AI byly zvýrazněním upraveny na funkční tvar v prostředí databáze Access:

7.5.1 Které modely jsem dodala letos v únoru? (Úloha 1.)

Text dotazu:

```
SELECT M.NazevModelu, S.DatumDodani
FROM ( aiSiti S
      LEFT JOIN aiModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID)
LEFT JOIN aiModely M ON ML.ModelID = M.ModelID
WHERE MONTH(S.DatumDodani) = 2 AND YEAR(S.DatumDodani) = YEAR(Date());
```

Výsledek dotazu:

Úloha 1	
NazevModelu	DatumDodani
Kalhoty krátké	14.02.2026
Plavky dvojdílné	23.02.2026
Kalhoty krátké	03.02.2026
Plavky dvojdílné	06.02.2026
Kalhoty krátké	16.02.2026

7.5.2 Kolik jsem si vydělala letos v únoru? (Úloha 2.)

Text dotazu:

```
SELECT SUM(M.PracnostHodiny * 300) AS Vydeleno
FROM ( aiSiti S
      LEFT JOIN aiModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID )
LEFT JOIN aiModely M ON ML.ModelID = M.ModelID
WHERE MONTH(S.DatumDodani) = 2 AND YEAR(S.DatumDodani)= YEAR(Date());
```

Výsledek dotazu:

Uloha 2
Vydelano
4200

7.5.3 Kolik jsem celkem spotřebovala té které látky? (Úloha 3.)

Text dotazu:

```
SELECT L.NazevLatky, SUM(ML.SpotrebaMetru) AS CelkemSpotreba
FROM ( aiSiti S
      LEFT JOIN aiModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID )
LEFT JOIN aiLatky L ON ML.LatkaID = L.LatkaID
GROUP BY L.NazevLatky
```

Výsledek dotazu:

Uloha 3	
NazevLatky	CelkemSpotreba
Licra	0,45
Terylen	12,25
Tesil	8,4

7.5.4 Který můj model je nejžádanější? (Úloha 4.)

Text dotazu:

```
SELECT TOP 1 M.NazevModelu, COUNT(*) AS PocetDodani
FROM ( aiSiti S
      LEFT JOIN aiModelLatka ML ON S.ModelLatkaID = ML.ModelLatkaID )
LEFT JOIN aiModely M ON ML.ModelID = M.ModelID
GROUP BY M.NazevModelu
ORDER BY COUNT(*) DESC;
```

Výsledek dotazu:

Uloha 4	
NazevModelu	PocetDodani
Plavky dvojdílné	5
Kalhoty krátké	5

7.5.5 Který model jsem ještě vůbec neudala? (Úloha 5.)

Text dotazu:

```
SELECT M.NazevModelu
FROM ( aiModely M
      LEFT JOIN aiModelLatka ML ON M.ModelID = ML.ModelID )
LEFT JOIN aiSiti S ON ML.ModelLatkaID = S.ModelLatkaID
WHERE S.SitiID IS NULL;
```

Výsledek dotazu:

Uloha 5
NazevModelu
Pomsta Tchyni
Východní Vítr

7.6 Databáze ke stažení

Celá databáze popsaná v kapitole AI: Co-pilot je dostupná ke stažení dle Lit. [1] a [2] - viz Literatura.

8 Tabulky a dotazy dle Chat-GPT

Poznámka: Tento výukový text předpokládá, že pilný čtenář - alespoň při prvním čtení - nepřeskakuje zběsile mezi kapitolami dolů a nahoru, ale čte sekvenčně od prvního odstavce do posledního. Pokud ano, má před očima podstatnou logiku realizace dle MS Co-pilot. Pokud ne, necht' se při absenci smyslu informací této kapitoly na zmíněnou realizaci Co-pilota vrátí.

8.1 {Volitelná} příprava

Především vytvoříme novou prázdnou databázi prostředky použitého databázového programu (v prostředí MS programu Access). Protože výhledově budeme plnit tabulky navržené AI testovacími daty již připravenými pro dřívější výuku (viz shora), pro větší názornost do nově vytvořené databáze rovnou importujeme tabulky MODELKY, LÁTKY, ZÁKAZNÍCI a PRODEJE zobrazené v kapitole shora. Výchozí stav tedy bude následující:



8.2 Vytvoření prázdných tabulek

Shora jsou v samostatné kapitole uvedeny návrhy struktury tabulek dle Chat-GPT. Dodržme je s následujícími drobnými úpravami:

- Protože ukazujeme tři různé modely zpracování stejných tabulek (vlastní prezentace, Co-pilot, Chat-GPT), ponechme základ jejich názvu, ale rozlišujeme je dle doporučení Microsoftu 'předponami' (zde dvoupísmenovými) podle kontextu. Např. tabulka 'Modely' bude v jednom kontextu označena jako 'aiModely', v jiném jako 'gpModely' apod.
- Z důvodů vysvětlených výše omezíme použití typu primárních klíčů Counter (=AutomatickéČíslo v MS) na zcela nutné případy (tabulka 'ModelLatka' a tabulka 'Šití'). Ostatní primární klíče ponechme jako tříznakový text.
- AI iniciativně přidala do svých návrhů datová pole, která by při skutečné praktické realizaci asi fakt nechyběla. Označila je jako „nepovinné“. V tomto článku s nimi není uvažováno z několika důvodů: (*) Článek je určen pro výuku; teprve vyučený student může nepovinná pole lehce použít. (**) Ve všech případech jde o hodnoty s přímou vypovídací schopností, nikoliv o data použitá pro vazby nebo další konstrukce. (***) Při přípravě testovacích dat na takových místech může být v podstatě cokoliv - případné ověřování formální nebo faktické hodnoty není cílem tohoto článku.

8.2.1 Tabulka Modely - CrteModely

```
create table gpModely
  (ModelID text(3) primary key,
   NavezModelu text(127),
   PracnostHodiny double,
   SpotrebaMetru double)
```

8.2.2 Tabulka Látky - CrteLatky

```
create table gpLatky
  (LatkaID text(3) primary key,
   NavezLatky text(63),
   CenaZaMetr currency)
```

8.2.3 Tabulka Zákazníci - CrteZakaznici

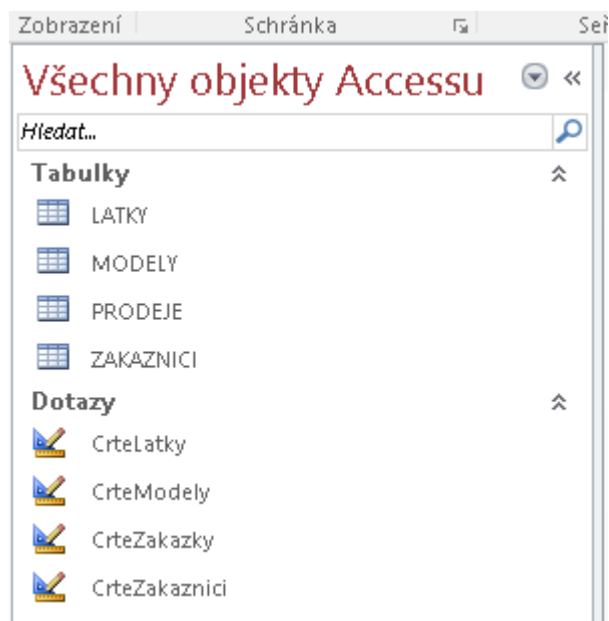
```
create table gpZakaznici
  (ZakaznikID text(3) primary key,
   Jmeno text(31),
   Prijmeni text(31),
   Email text(127))
```

8.2.4 Tabulka Zakazky - CrteZakazky

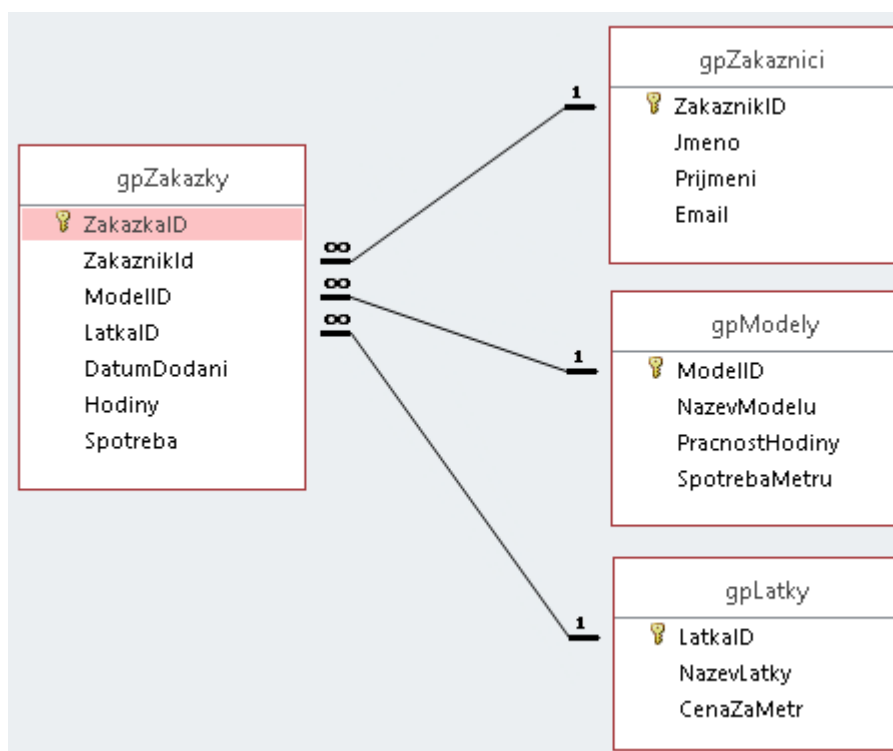
```
create table gpZakazky
  (ZakazkaID counter primary key,
   ZakaznikId text(3) constraint GHI references gpZakaznici (ZakaznikID),
   ModelID text(3) constraint JKL references gpModely (ModelID),
   LatkaID text(3) constraint MNO references gpLatky (LatkaID),
   DatumDodani Date,
   Hodiny double,
   Spotreba double)
```

8.2.5 Struktura databáze dle Chat-GPT

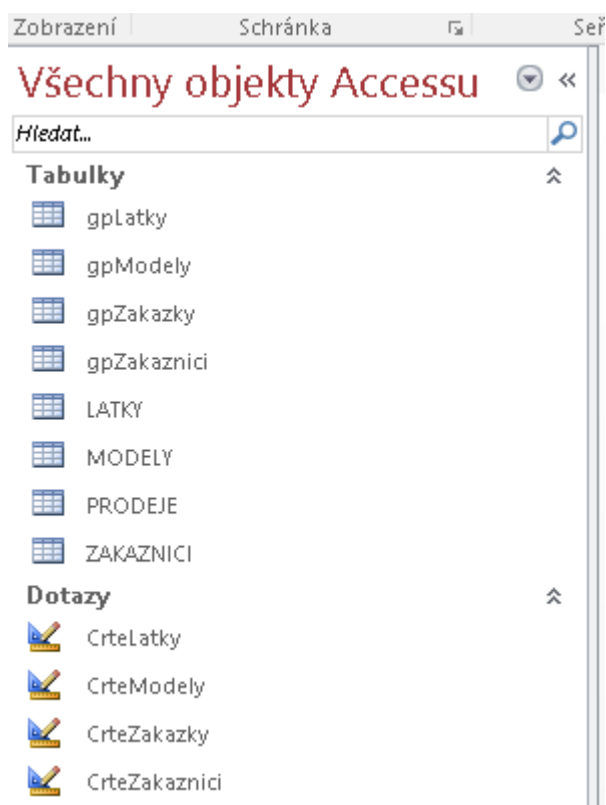
Uvedené příkazy Create Table uložíme jako jednotlivé dotazy do databáze, názvy dotazů jsme uvedli v nadpisech nad příkazy. Databáze bude informovat:



Po provedení příkazů bude vytvořena následující struktura s naznačenými vazbami mezi tabulkami (viz klauzule Constraint v předchozích příkazech)



a databáze zobrazí nový obsah:



8.3 Naplnění tabulek Modely, Látky a Zákazníci

Jde o tabulky, které jsou cílem vazby (samy nikam neodkazují), je proto možno bez úpravy použít data z testu analýzy. Příkazy Insert Into opět uložíme jako jednotlivé dotazy do databáze, názvy dotazů jsou uvedeny v nadpisech nad příkazy

8.3.1 Tabulka Modely - InsModely

```
insert into gpModely ( ModelID, NazevModelu, PracnostHodiny, SpotrebaMetru)
select KodModelu, NazevModelu, PracnostHodin, SpotrebaMetru from MODELKY
```

8.3.2 Tabulka Látky - InsLatky

```
insert into gpLatky ( LatkaID, NazevLatky, CenaZaMetr )
select KodLatky, NazevLatky, CenaZaMetr from LATKY
```

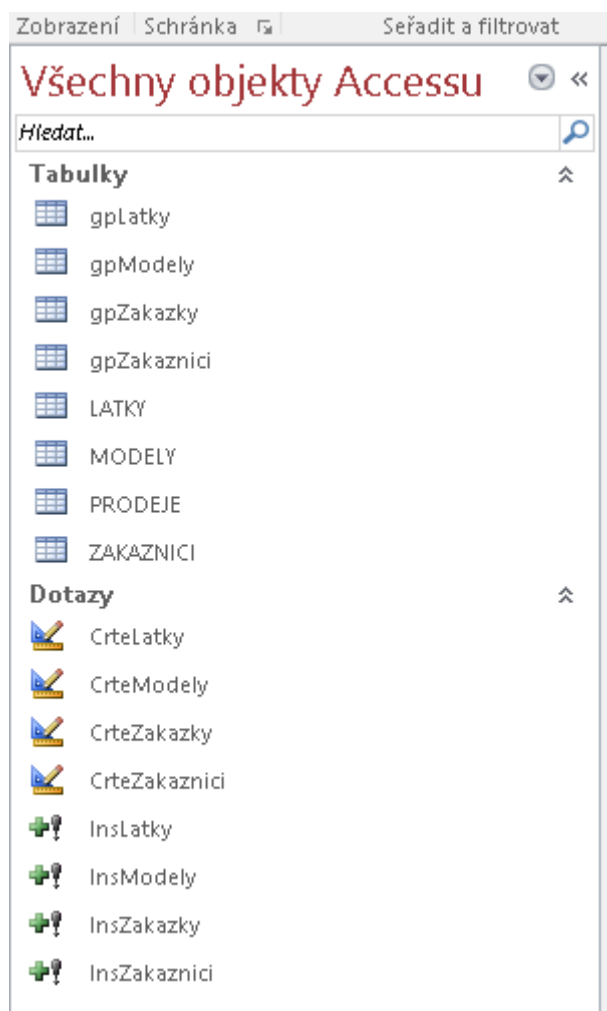
8.3.3 Tabulka Zákazníci - InsZakaznici

```
insert into gpZakaznici ( ZakaznikID, Prijmeni, Jmeno, Email )
select KodZakaznika, Prijmeni, Jmeno, Email from ZAKAZNICI
```

8.3.4 Tabulka Zakazky - InsZakazky

```
insert into gpZakazky
  (ZakazkaID, ZakaznikID, ModelID, LatkaID,
   DatumDodani, Hodiny, Spotreba)
select
  P.CisloZakazky, Z.KodZakaznika, M.KodModelu, L.KodLatky,
  P.DatumDodani, M.PracnostHodin, M.SpotrebaMetru
from
  (
    (PRODEJE P left join ZAKAZNICI Z on P.KodZakaznika=Z.KodZakaznika)
    left join MODELY M on P.KodModelu=M.KodModelu
  )
  left join LATKY L on P.KodLatky=L.KodLatky
```

Po vytvoření uvedených „dotazů“ Insert Into aktualizuje program Access svůj panel se strukturou databáze následovně:



8.3.5 Použití příkazů Insert Into

Až doposud mají tabulky gpLatky a další gp... strukturu dle AI, ale nemají žádný obsah. Teprve postupným vykonáním příkazů InsLatky a dalších Ins... dojde k jejich naplnění ze zdrojů MODELY ... až PRODEJE.

Důležitá poznámka: Pozor na pořadí spouštění příkazů Ins... !! Nejprve je nutno (v libovolném pořadí) spustit naplnění těch tabulek, které jsou cílem nějaké vazby, ale ze kterých žádná vazba nevychází. Naopak jako poslední je nutno nechat naplnit ty tabulky, do kterých žádná vazba nesměřuje. Logicky při plnění daty té tabulky, ze které vychází vazba, musí být naplněna daty ty tabulky, do kterých takové vazby směřují. V diskutovaném případě tedy je nutno nechat vykonat v libovolném pořadí nejprve příkazy { **InsModely**, **InsLatky**, **InsZakaznici** }, a teprve nakonec příkaz **InsZakazky**.

8.4 Hodnocení návrhů dotazů Chat-GPT

Na rozdíl od předchozí AI Co-pilot jsou navržené texty příkazů zaslaných AI Chat-GPT víceméně přímo použitelné. Minimální úpravy vyžadovaly pouze názvy tabulek, které byly z důvodů rozlišení od podobně označených tabulek tohoto článku rozlišeny přidáním prefixu **gp**. Jediný nesoulad se syntaxí pro použitou databázi Microsoftu - použití dočasně pojmenování jednoho výrazu seznamu SELECT v klauzuli ORDER BY - byl opraven přímým uvedením tohoto výrazu. V níže uvedených textech příkazů byly drobné úpravy zvýrazněny barevně a tučným písmem. Co do funkčnosti jsou dotazy po těchto úpravách bez problémů a dávají správné výsledky.

8.4.1 Dotaz 1: Které modely jsem dodala letos v únoru?

Text dotazu:

```
SELECT M.NazevModelu, ZK.DatumDodani, ZK.ZakazkaID
FROM gpZakazky AS ZK
INNER JOIN gpModely AS M ON ZK.ModelID = M.ModelID
WHERE Month(ZK.DatumDodani) = 2 AND Year(ZK.DatumDodani) = Year(Date());
```

Výsledek dotazu:

Dotaz 1		
NazevModelu	DatumDodani	ZakazkaID
Kalhoty krátké	14.02.2026	3
Plavky dvojdílné	23.02.2026	7
Kalhoty krátké	03.02.2026	8
Plavky dvojdílné	06.02.2026	10
Kalhoty krátké	16.02.2026	13

8.4.2 Dotaz 2: Kolik jsem si vydělala letos v únoru?

Text dotazu:

```
Předpoklad: účtujete 300 Kč / hod.
SELECT SUM(ZK.Hodiny * 300) AS Vydeleno
FROM gpZakazky AS ZK
WHERE Month(ZK.DatumDodani) = 2 AND Year(ZK.DatumDodani) = Year(Date());
```

Výsledek dotazu:

Dotaz 2
Vydeleno
4200

8.4.3 Dotaz 3: Kolik jsem celkem spotřebovala té které látky?

Text dotazu:

```
SELECT L.NazevLatky, SUM(ZK.Spotreba) AS CelkemSpotrebovano_m
FROM gpZakazky AS ZK
INNER JOIN gpLatky AS L ON ZK.LatkaID = L.LatkaID
GROUP BY L.NazevLatky;
```

Výsledek dotazu:

Dotaz 3	
NazevLatky	CelkemSpotrebovano_m
Licra	0,45
Terylen	12,25
Tesil	8,4

8.4.4 Dotaz 4: Který můj model je nejžádanější?

Text dotazu:

```
SELECT TOP 1 M.NazevModelu, COUNT(ZK.ZakazkaID) AS PocetZakazek
FROM gpZakazky AS ZK
INNER JOIN gpModely AS M ON ZK.ModelID = M.ModelID
GROUP BY M.NazevModelu
ORDER BY COUNT(ZK.ZakazkaID) DESC
```

Výsledek dotazu:

Dotaz 4	
NazevModelu	PocetZakazek
Plavky dvojdílné	5
Kalhoty krátké	5

8.4.5 Dotaz 5: Který model jsem ještě vůbec neudala?

Text dotazu:

```
SELECT M.NazevModelu
FROM gpModely AS M
LEFT JOIN gpZakazky AS ZK ON M.ModelID = ZK.ModelID
WHERE ZK.ZakazkaID IS NULL
```

Výsledek dotazu:

Dotaz 5
NazevModelu
Pomsta Tchyni
Východní Vítr

9 Závěr

Jak je zmíněno už v Anotaci, tento text je určen spíše začátečníkům v relativně složité databázové problematice. V ní tvoří to, co se označovalo jako „báze dat“, jednu samostatnou kapitolu. Pro počáteční seznámení byla zvolena jedna z jednoduchých organizací, známá jako „Relační databáze“. Na ní byla postavena řada databázových 'systémů' (Microsoft Jet, Oracle, FoxPro aj). Navíc jeden z široce používaných dotazovacích jazyků, SQL, jaksi implicitně předpokládá relační organizaci.

Protože dominantním operačním systémem na zatím (počtem) převažující třídě 'osobních' počítačů je systém firmy Microsoft (známý jako MS Windows) a v něm balík 'kancelářských' aplikací obsahující databázovou komponentu (MS Office), je výuka vedena jednak v jejich operačním systému, jednak v jejich databázových komponentách při využití programu Access z MS Office. V praxi se však dnešní studenti = budoucí noví zaměstnanci mohou setkat s nespočtem jiných datových aplikací, podle zvyklostí ve firmě zavedených. Proto je snahou vést výuku podáváním informací, které v takové podobě nebo s minimem odchylek lze najít i v databázových produktech jiných autorů.

10 Literatura a další zdroje ke studiu

[1] HOMOLA, V.: *Realizace úlohy SALON dle návrhu AI Microsoft Co-pilot v databázi ACCDB*. Databáze pouze s testovacími daty a funkčním řešením úloh 1-5. [online]. Ostrava, 2026. [cit. 6. 3. 2026]. Dostupné z: https://hom50.cz/SALON_CP.ACCDB

[2] HOMOLA, V.: *Realizace úlohy SALON dle návrhu AI Microsoft Co-pilot v databázi ACCDB*. Databáze s testovacími daty, funkčním řešením úloh 1-5, včetně v textu popisovaných zdrojových dat a příkazů pro jejich transformaci na testovací data. [online]. Ostrava, 2026. [cit. 6. 3. 2026]. Dostupné z: https://hom50.cz/SALON_C.ACCDB

[3] HOMOLA, V.: *Realizace úlohy SALON dle návrhu AI Chat-GPT v databázi ACCDB*. Databáze pouze s testovacími daty a funkčním řešením úloh 1-5. [online]. Ostrava, 2026. [cit. 6. 3. 2026]. Dostupné z: https://hom50.cz/SALON_GP.ACCDB

[4] HOMOLA, V.: *Realizace úlohy SALON dle návrhu AI Chat-GPT v databázi ACCDB*. Databáze s testovacími daty, funkčním řešením úloh 1-5, včetně v textu popisovaných zdrojových dat a příkazů pro jejich transformaci na testovací data. [online]. Ostrava, 2026. [cit. 6. 3. 2026]. Dostupné z: https://hom50.cz/SALON_G.ACCDB

- [5] DROZDOVÁ, J., HOMOLA, V. (2018): *Informatika pro Geovědní a montánní turismus*. Učební texty pro předmět Výpočetní technika. VŠB-TU Ostrava. ISBN 978-80-248-4145-8.
- [6] DROZDOVÁ, J., HOMOLA, V. (2019): *Databáze pro Geovědní obory*. Učební texty pro předmět Výpočetní technika. VŠB-TU Ostrava. ISBN 978-80-248-4265-3.
- [7] CHLAPEK, D., STANOVSKÁ, I., ŘEPA, V.: *Analýza a návrh informačních systémů*. Praha: Oeconomica, 2011. ISBN: 978-80-245-1782-7.
- [8] DATE, C. J., DARWEN, H.: *Databases, types and the relational model: the third manifesto*. (Version 2 ed.) Reading, MA: Addison-Wesley, 2006. ISBN 0-321-39942-0.
- [9] MICROSOFT: *Description of the database normalization basics*. [online]. Microsoft, 2017. [cit. 20. 2. 2026]. Dostupné z: <https://support.microsoft.com/en-us/help/283878>

Rev. 3 / 2026