

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Business Intelligence – Pekáreň OPP
Semestrálna práca

Mená študentov: Bc. Patrik Obložinský, Bc. Patrik Provazník, Bc. Radovan Pieter

Odbor a krúžok: Informačný manažment, 3. krúžok

Školský rok: 2021/2022

Meno cvičiaceho: Ing. Veronika Horniaková, PhD., MSc.

Úvod

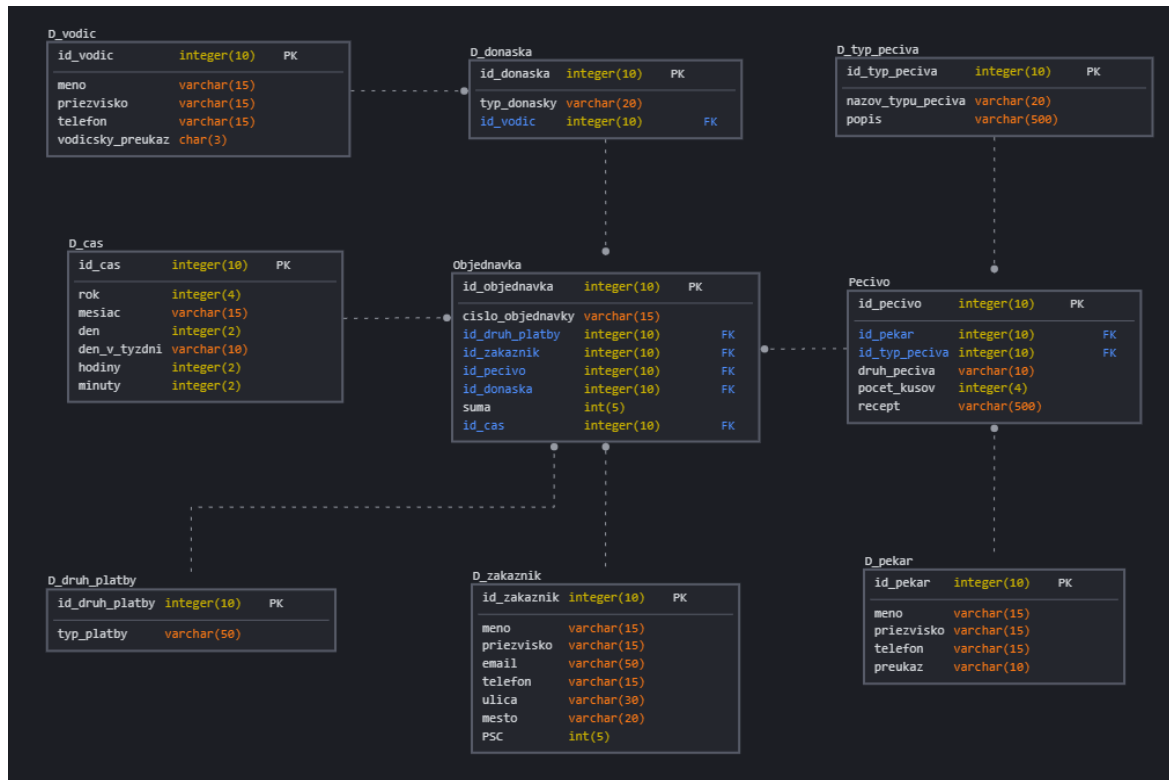
V našej semestrálnej práci sa budeme venovať malej pekárni s názvom OPP. Pekáreň okrem predaja v predajni rozšíril svoje predaje aj do online priestoru, čím vznikla potreba uchovávať dáta o objednávkach a upečenom pečive v digitálnej forme. Keďže prijímanie objednávok funguje online treba preto navrhnuť dátový sklad pre pekáreň, aby boli všetky údaje dostupné, čitateľné a ľahko škálovateľné. Pre logický model nášho databázového skladu sme zvolili štruktúru CONSTELLATION, pretože máme viac ako jednu tabuľku faktov – konkrétne dve. Okrem dvoch tabuliek faktov sa v logickom modeli nachádza 7 dimenzií. Pri modelovaní dátového skladu sme využili USER driven typ – vytvorili sme dátový sklad na základe požiadaviek používateľa, keďže pekáreň doteraz nemala žiadne základy, z ktorých sme mohli vychádzať. V našej semestrálnej práci budeme na vytvorenie databázovej schémy a následnú manipuláciu s dátami používať nástroj SqlDBM (sqldbm.com) a phpmyadmin.

Základe informácie o podniku a potreba dátového skladu

Naším podnikom je pekáreň, ktorá sa zaoberá výrobou a predaj pekárenských produktov. V portfóliu tohto podniku môžeme nájsť rôzny sortiment produktov, kde patria bežné pečivá ako rožky a chlieb, ale taktiež aj množstvo špeciálnych produktov ako napríklad šišky, či koláče. Keďže sme sa rozhodli spustiť online predaj, potrebovali sme si dáta ukladať v nejakej forme, aby zákazníci videli, aké produkty ponúkame. Tiež si musíme ukladať informácie či už o zákazníkoch alebo objednávkach, ale aj o vnútropodnikových (interných) informáciách, ako sú napríklad spôsoby platby, donášky, informácie o pekároch, vodičoch áut a hlavne informácie o našom pečive - typ pečiva, recept, popis a názov pečív, ktoré predávame. Vďaka týmto dátam máme presný prehľad o tom, koľko si aký zákazník objedná pečiva, presne druh pečiva a ako zaplatí zákazník za objednávku. Ďalšie využitie dát uložených v dátovom sklade je možnosť vylepšenia predaja, resp. vylepšenie reklám pre zákazníkov – tým, že máme dáta o objednávkach, vieme určiť, aký typ pečiva alebo aké konkrétne pečivo je obľúbené v akých geografických oblastiach. Pomocou týchto štatistík môžeme presnejšie cieľiť reklamy, do rôznych oblastí a tým pádom zvýšiť predaje v našom online obchode.

Samozrejme pomocou dát vybraných z dátového skladu vieme určiť nielen informácie pre zákazníkov, ale taktiež aj informácie, ktoré pomôžu podniku rozlíšiť potrebu napr. nového zamestnanca alebo ktoré dni je potrebná výpomoc pre pekárov alebo vodičov – teda zistiť, ktoré dni v týždni si ľudia objednávajú viac tovarov, aby sme plne uspokojili dopyt po našich tovaroch.

Prvá časť - Logický model dátového skladu



Dátový model obsahuje 7 dimenzií a 2 tabuľky faktov.

Tabul'ky dimenzií:

- D_cas – dimenzia obsahujúca jednoduché časové údaje aby mohol byť bližšie špecifikovaný čas napr. čas vytvorenia objednávky a čas napečenia pečiva
 - *id_cas* – INTEGER(10) - primárny kľúč
 - *rok* - INTEGER(4)
 - *mesiac* - VARCHAR(15)
 - *den* - INTEGER(2)
 - *den_v_tyzdni* - VARCHAR(10)
 - *hodiny* - INTEGER(2)
 - *minuty* - INTEGER(2)
- D_donaska – dimenzia opisujúca informácie o donáške pečiva pre zákazníka
 - *id_donaska* – INTEGER(10) - primárny kľúč
 - *typ_donasky* – VARCHAR(20)
 - *id_vodic* – INTEGER(10) – cudzí kľúč dimenzie D_vodic

- D_pekar – obsahuje základné atribúty pre opis entity „pekár“
 - *id_pekar* - INTEGER(10) - primárny kľúč
 - *meno* – VARCHAR(15)
 - *priezvisko* - VARCHAR(15)
 - *telefon* – VARCHAR(15)
 - *preukazu* - VARCHAR(10)

- D_zakaznik – obsahuje všetky informácie o zákazníkovi
 - *id_zakaznik* – INTEGER(10) - primárny kľúč
 - *meno* – VARCHAR(15)
 - *priezvisko* - VARCHAR(15)
 - *email* - VARCHAR(50)
 - *telefon* - VARCHAR(15)
 - *ulica* - VARCHAR(30)
 - *mesto* - VARCHAR(20)
 - *PSC* - INTEGER(5)

- D_vodic – obsahuje všetky informácie o vodičovi
 - *id_vodic* – INTEGER(10) - primárny kľúč
 - *meno* - VARCHAR(15)
 - *priezvisko* - VARCHAR(15)
 - *telefon* - VARCHAR(15)
 - *vodicky_preukaz* – CHAR(3)

- D_druh_platby – obsahuje všetky informácie o druhu, resp. spôsobe platby
 - *id_druh_platby* – INTEGER(10) - primárny kľúč
 - *typ_platby* - VARCHAR(50)

- D_typ_peciva – obsahuje všetky informácie o type pečiva
 - *id_typ_peciva* – INTEGER(10) - primárny kľúč
 - *nazov_typu_peciva* - VARCHAR(20)
 - *popis* – VARCHAR(500)

Tabuľky faktov:

- Pecivo – tabuľka faktov obsahujúca atribúty, ktoré opisujú pečivo, ktoré je pečené. Môžu sa následne robiť operácie, kedy zistíme napr. „počet pečiva upečeného daným pekárom“ alebo „počet upečených kusov daného typu pečiva“, atď.
 - *id_pecivo* – INTEGER(10) - primárny kľúč
 - *id_pekar* – INTEGER(10) - cudzí kľúč z dimenzie D_pekar
 - *typ_peciva* – INTEGER(10) - cudzí kľúč z dimenzie D_typ_peciva
 - *druh_pecivo* – VARCHAR(10)
 - *pocet_kusov* – INTEGER(4)
 - *recept* - VARCHAR(500)
- Objednavka – táto tabuľka faktov obsahuje atribúty opisujúce objednávku. Dajú sa následne robiť operácie „Celková suma v danom dni“, „Počet objednávok vytvorených konkrétnym zákazníkom“, atď.
 - *id_objednavka* – INTEGER(10) - primárny kľúč
 - *cislo_objednavky* – VARCHAR(15)
 - *id_cas* - INTEGER(10) - cudzí kľúč z dimenzie D_cas
 - *id_zakaznik* - INTEGER(10) - cudzí kľúč z dimenzie D_zakaznik
 - *id_pecivo* - INTEGER(10) - cudzí kľúč z dimenzie pecivo
 - *id_donaska* - INTEGER(10) - cudzí kľúč z dimenzie D_donaska
 - *id_druh_platby* - INTEGER(10) cudzí kľúč z dimenzie D_druh_platby
 - *suma* – INTEGER(5)

Druhá časť - ETL

Na vytvorenie databázy pekárne sme využili webový portál webzdarma.cz, ktorý nám poskytol. Využívanie služby phpMyAdmin, kde sme si prostredníctvom SQL príkazov vytvorili jednotlivé tabuľky dimenzií a faktov našej pekárne. Príkazy pre vytvorenie našich tabuliek boli nasledovné:

1. Dimenzia D_cas

```
CREATE TABLE D_cas
(
`id_cas` int(10) NOT NULL ,
`rok` int(4) NOT NULL ,
`mesiac` varchar(15) NOT NULL ,
`den` int(2) NOT NULL ,
`den_v_tyzdni` varchar(10) NOT NULL ,
`hodiny` int(2) NOT NULL ,
`minuty` int(2) NOT NULL ,

PRIMARY KEY (`id_cas`)
);
```

2. Dimenzia D_donaska

```
CREATE TABLE D_donaska
(
`typ_donasky` varchar(20) NOT NULL ,
`id_vodic` int(10) NOT NULL ,
`id_donaska` int(10) NOT NULL ,

PRIMARY KEY (`id_donaska`),
FOREIGN KEY (`id_vodic`) REFERENCES D_vodic(`id_vodic`)
);
```

3. Dimenzia D_druh_platby

```
CREATE TABLE D_druh_platby
(
`id_druh_platby` integer(10) NOT NULL ,
`typ_platby` varchar(50) NOT NULL ,

PRIMARY KEY (`id_druh_platby`)
);
```

4. Dimenzia D_pekar

```
CREATE TABLE D_pekar
(
`id_pekar` integer(10) NOT NULL ,
`meno` varchar(15) NOT NULL ,
`priezvisko` varchar(15) NOT NULL ,
`telefon` varchar(15) NOT NULL ,
`preukaz` varchar(10) NOT NULL ,

PRIMARY KEY (`id_pekar`)
);
```

5. Dimenzia D_typ_peciva

```
CREATE TABLE D_typ_peciva
(
`id_typ_peciva` integer(10) NOT NULL ,
`nazov_typu_peciva` varchar(20) NOT NULL ,
`popis` varchar(500) NOT NULL ,

PRIMARY KEY (`id_typ_peciva`)
);
```

6. Dimenzia D_vodic

```
CREATE TABLE D_vodic
(
  `id_vodic` integer(10) NOT NULL ,
  `meno` varchar(15) NOT NULL ,
  `priezvisko` varchar(15) NOT NULL ,
  `telefon` varchar(15) NOT NULL ,
  `vodicky_preukaz` char(3) NOT NULL ,
  PRIMARY KEY (`id_vodic`)
);
```

7. Dimenzia D_zakaznik

```
CREATE TABLE D_zakaznik
(
  `id_zakaznik` integer(10) NOT NULL ,
  `meno` varchar(15) NOT NULL ,
  `priezvisko` varchar(15) NOT NULL ,
  `email` varchar(50) NOT NULL ,
  `telefon` varchar(15) NOT NULL ,
  `ulica` varchar(30) NOT NULL ,
  `mesto` varchar(20) NOT NULL ,
  `PSC` int(5) NOT NULL ,

  PRIMARY KEY (`id_zakaznik`)
);
```

1. Fakt Pecivo

```
CREATE TABLE Pecivo
(
  `id_pecivo` integer(10) NOT NULL ,
  `id_pekar` integer(10) NOT NULL ,
```



```

`id_typ_peciva` integer(10) NOT NULL ,
`druh_peciva` varchar(10) NOT NULL ,
`pocet_kusov` integer(4) NOT NULL ,
`recept` varchar(500) NOT NULL ,

PRIMARY KEY (`id_pecivo`),
FOREIGN KEY (`id_pekar`) REFERENCES D_pekar(`id_pekar`),
FOREIGN KEY (`id_typ_peciva`) REFERENCES D_typ_peciva(`id_typ_peciva`)
);

```

2. Fakt Objednavka

```

CREATE TABLE Objednavka
(
`id_objednavka` integer(10) NOT NULL ,
`cislo_objednavky` varchar(15) NOT NULL ,
`id_druh_platby` integer(10) NOT NULL ,
`id_zakaznik` integer(10) NOT NULL ,
`id_pecivo` integer(10) NOT NULL ,
`id_donaska` integer(10) NOT NULL ,
`id_cas` integer(10) NOT NULL ,
`suma` int(5) NOT NULL ,

PRIMARY KEY (`id_objednavka`),
FOREIGN KEY (`id_druh_platby`) REFERENCES D_druh_platby(`id_druh_platby`),
FOREIGN KEY (`id_zakaznik`) REFERENCES D_zakaznik(`id_zakaznik`),
FOREIGN KEY (`id_pecivo`) REFERENCES Pecivo(`id_pecivo`),
FOREIGN KEY (`id_donaska`) REFERENCES D_donaska(`id_donaska`),
FOREIGN KEY (`id_cas`) REFERENCES D_cas(`id_cas`)
);

```

Po vytvorení tabuliek, bolo nutné dané tabuľky naplniť dátami. Dáta sme do našej databázy vkladali pomocou CSV súborov vytvorených v tabuľkovom editore MS Excel. Postup vkladania dát vyzeral nasledovne:

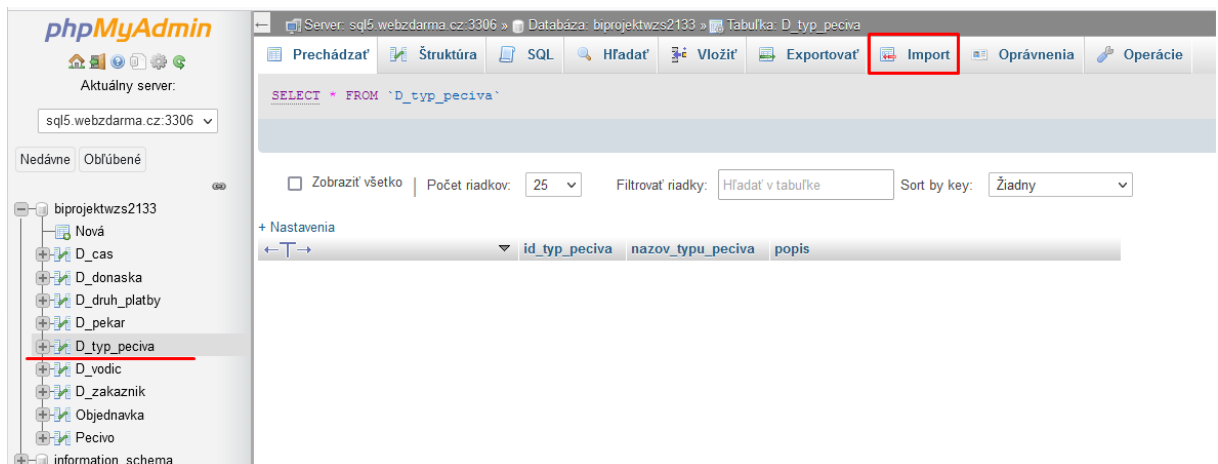
1. Na začiatku sme museli do MS Excel vložiť dáta podľa atribútov jednotlivých tabuliek dimenzií a faktov. Dáta sme pre niektoré atribúty tabuliek vytvorili pomocou generátorov (generátor mien, adries, dátumov,...), pre iné tabuľky sme zase dáta plnili ručne alebo hľadali na ľubovoľných stránkach na internete. Napríklad tabuľka pre dimenziu D_typ_peciva obsahuje atribút id_typ_peciva s dátovým typom integer a dĺžkou 10, atribút nazov_tupu_peciva s dátovým typom varchar a dĺžkou 20 a atribút popis s dátovým typom varchar a dĺžkou 500. Podľa týchto požiadaviek sme niektoré dáta vytvorili ručne, ale taktiež sme dáta naplnili aj pomocou stránok na internete pre excelovský súbor D_typ_peciva. Ukážka MS Excel s vloženými dátami, kde stĺpce sú atribúty danej tabuľky a riadky údaje, ktoré sa do nich vkladajú:

	A	B	C
1	1	rozok	Pecivo bezne
2	2	chlieb	Pecivo bezne
3	3	bageta	Pecivo bezne
4	4	zemla	Pecivo bezne
5	5	celozrnnna zemla	Pecivo specialne
6	6	celozrnnna bageta	Pecivo specialne
7	7	celozrnnny rozok	Pecivo specialne
8	8	pizza rozok	Pecivo specialne
9	9	pletenka	Pecivo bezne
10	10	praclik	Pecivo bezne
11	11	marhulova siska	Klasicky mucnik s jemnym marhulovym dzemom
12	12	nugatova siska	Jemna siska plnena nugatovym kremom
13	13	vanilkova siska	Klasicky nadychany mucnik s jemnym vanilkovym kremom
14	14	jablkova tasticcka	Tasticcka z listkoveho cesta plnena jablkovou naplnou
15	15	jablkovy kolac	Stavnaty jablkovy kolac so stipkou skorice
16	16	makovy zavin	Zavin z jemneho kysnuteho cesta s bohatou makovou naplnou
17	17	kakaovy zavin	Zavin z jemneho kysnuteho cesta s bohatou kakaovou naplnou
18	18	orechovy zavin	Zavin z jemneho kysnuteho cesta s bohatou orechovou naplnou
19	19	cokoladova satocka	Chutny kolac s poriadnou davkou cokolady
20	20	slivkova buchta	Buchta plnena slivkovym lekvaram

2. Pred vložením týchto dát sme museli súbor MS Excelu uložiť ako súbor CSV oddelený čiarkami:

File name:	typ_peciva
Save as type:	CSV (Comma delimited)

3. Následne sme mohli tieto dáta vložiť (importovať) do tabuľky. Na vkladanie dát bolo nutné si vybrať tabuľku, do ktorej ideme dáta vkladat' a následne sme použili funkciu Import v phpMyAdmine:



4. Vo funkcii Import sme si najskôr nahrali CSV súbor s dátami, ktoré ideme vložiť do tabuľky D_typ_peciva, taktiež sme si nastavili znakovú sadu UTF-8, vybrali sme formát pre CSV súbor a posledným krokom bolo nastavenie odelenia našich polí pomocou bodkočiarky. Po týchto nastaveniach sme stlačili tlačidlo Vykonaj a tabuľka D_typ_peciva sa naplnila údajmi.

Súbor na importovanie:
Súbor môže byť komprimovaný (gzip, bzip2, zip) alebo nekomprimovaný.
A compressed file's name must end in .[format][compression]. Example: .sql.zip
Prehliadať váš počítač: Prehliadať... typ_peciva.csv (Maximálna veľkosť: 200MB)
You may also drag and drop a file on any page.
Znaková sada súboru: utf8

Častočný import:
☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large files, however it can break transactions.)
Počet dopytov od začiatku (pre SQL), ktoré sa majú preskočiť:

Ďalšie nastavenia:
☒ Povoliť kontrolu outfitu súborov



























































Formát:

Poznámka: Ak súbor obsahuje viac tabuliek, tieto budú spojené do jednej

Formát - špecifické voľby:
☐ Update data when duplicate keys found on import (add ON DUPLICATE KEY UPDATE)
Polia oddelené s:
Polia uzatvorené do:
Polia uvedené pomocou:
Riadky ukončené:
Import these many number of rows (optional):
Column names:
☐ Nepretrúvať pri chybe v príkaze INSERT

Vykonaj

5. Po vykonaní týchto krokov vyzerá naša naplnená tabuľka nasledovne:

← T →							id_typ_peciva	nazov_typu_peciva	popis
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	1	rozok	Pecivo bezne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	2	chlieb	Pecivo bezne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	3	bageta	Pecivo bezne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	4	zemla	Pecivo bezne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	5	celozrnna zemla	Pecivo specialne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	6	celozrnna bageta	Pecivo specialne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	7	celozrnny rozok	Pecivo specialne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	8	pizza rozok	Pecivo specialne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	9	pletenka	Pecivo bezne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	10	praclik	Pecivo bezne
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	11	marhulova siska	Klasicky mucnik s jemnym marhulovym dzemom
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	12	nugatova siska	Jemna siska plnena nugatovym kremom
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	13	vanilkova siska	Klasicky nadychany mucnik s jemnym vanilkovym krem...
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	14	jablkova tasticcka	Tasticcka z listkoveho cesta plnena jablkovou napl...
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	15	jablkovy kolac	Stavnaty jablkovy kolac so stipkou skorice
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	16	makovy zavin	Zavin z jemneho kysnutého cesta s bohatou makovou ...
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	17	kakaovy zavin	Zavin z jemneho kysnutého cesta s bohatou kakaovou...
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	18	orechovy zavin	Zavin z jemneho kysnutého cesta s bohatou orechovo...
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	19	cokoladova satocka	Chutny kolac s poriadnou davkou cokolady
<input type="checkbox"/>		Upraviť		Kopírovať		Zmazať	20	slivkova buchta	Buchta plnena slivkovým lekvárom

Tretia časť SQL - príklady SELECT-ov

Po naplnení tabuliek údajmi sme mohli použiť príkaz SELECT, ktorý bol ďalšou časťou zadania. Naše príklady použitia SELECT-u vyzerajú nasledovne:

Zadanie 1. SELECT-u: Potrebujeme zistiť pomer platieb (druh platby) v jednotlivých mesiacoch za všetky objednávky, ktoré boli vytvorené v roku 2019.

- Týmto spôsobom zistíme, či sa nám oplatí napríklad prijímať platby kartou.

Kód pre vytvorenie takto zadaného SELECT-u:

```
SELECT c.mesiac, dp.typ_platby, COUNT(dp.typ_platby) AS pocet_platieb,  
COUNT(dp.typ_platby)/(SELECT COUNT(Objednavka.id_objednavka) FROM  
Objednavka)*100 AS Pomer FROM Objednavka o  
JOIN D_cas c ON o.id_cas = c.id_cas  
JOIN D_druh_platby dp ON o.id_druh_platby = dp.id_druh_platby  
WHERE c.rok = 2019  
GROUP BY c.mesiac, dp.typ_platby
```

mesiac	typ_platby	pocet_platieb	Pomer
January	hotovost	16	2.0513
January	karta	10	1.2821
January	gastrolistok	13	1.6667
January	paypal	15	1.9231
February	paypal	13	1.6667
February	karta	16	2.0513
February	gastrolistok	12	1.5385
February	hotovost	12	1.5385
March	gastrolistok	8	1.0256
March	hotovost	11	1.4103
March	karta	12	1.5385
March	paypal	16	2.0513
April	gastrolistok	14	1.7949
April	paypal	9	1.1538
April	hotovost	12	1.5385
April	karta	12	1.5385
May	karta	23	2.9487
May	hotovost	22	2.8205
May	gastrolistok	20	2.5641
May	paypal	14	1.7949
June	gastrolistok	22	2.8205
June	hotovost	20	2.5641
June	paypal	20	2.5641
June	karta	21	2.6923

July	gastrolistok	13	1.6667
July	hotovost	12	1.5385
July	karta	9	1.1538
July	paypal	14	1.7949
August	hotovost	14	1.7949
August	gastrolistok	19	2.4359
August	paypal	19	2.4359
August	karta	15	1.9231
September	gastrolistok	13	1.6667
September	paypal	12	1.5385
September	karta	10	1.2821
September	hotovost	18	2.3077
October	gastrolistok	24	3.0769
October	hotovost	21	2.6923
October	paypal	21	2.6923
October	karta	25	3.2051
November	gastrolistok	10	1.2821
November	karta	9	1.1538
November	hotovost	13	1.6667
November	paypal	19	2.4359
December	paypal	27	3.4615
December	hotovost	25	3.2051
December	gastrolistok	26	3.3333
December	karta	29	3.7179

Z tejto štatistiky nám vyplýva, že pomery všetkých platieb sú približne rovnaké – zákazníci platia dostupnými metódami takmer rovnako, takže sa nám oplatí udržiavať všetky druhy platieb. V prípade, že by časom klesol určitý typ platby, môžeme začať uvažovať, že zrušíme jeho používanie v našej spoločnosti.

Zadanie 2. SELECT-u: Chceme zistiť, či zákazníci, ktorí bývajú v Bratislave dlhodobo objednávajú viac ROŽKOV pred víkendom (v piatok) alebo po víkende (pondelok)

- Týmto spôsobom sa snažíme zistiť, či máme posilniť pekárenskú výrobu v našom podniku pred víkendom alebo po víkende, resp. či vôbec zvýšiť/posilniť výrobu v tieto dni.

Kód pre vytvorenie takto zadaného SELECT-u:

```
SELECT  c.den_v_tyzdni,  z.mesto,  tp.nazov_typu_peciva,  SUM(p.pocet_kusov)  as
pocet_rozkov
FROM Objednavka o
JOIN D_cas c ON o.id_cas = c.id_cas
JOIN Pecivo p ON o.id_pecivo = p.id_pecivo
JOIN D_typ_peciva tp ON p.id_typ_peciva = tp.id_typ_peciva
JOIN D_zakaznik z ON o.id_zakaznik = z.id_zakaznik
WHERE c.den_v_tyzdni IN ("Monday", "Friday")
AND tp.nazov_typu_peciva = "rozok"
AND z.mesto = "Bratislava"
GROUP BY c.den_v_tyzdni
```

den_v_tyzdni	mesto	nazov_typu_peciva	pocet_rozkov
Monday	Bratislava	rozok	415
Friday	Bratislava	rozok	328

Z výsledku nám vyšlo, že zákazníci v Bratislave objednávajú o 87 rožkov menej pred víkendom ako po víkende. Z tejto štatistiky nám vyplýva, že by sme mali v pondelok posilniť výrobu – najatť pomocníka na pečenie, tým pádom zvýšiť výrobu v tento deň, aby sme uspokojili väčší dopyt po rožkoch.