# Zpracování skriptů SQL pro IBM i

Návod k použití

# Obsah

**Obsah** ..... 2

**Úvod** ..... 5

**Použití aplikace** ..... 6
*Adresáře* ..... 6
*Programové soubory* ..... 6
*Umístění a spouštění aplikace* ..... 6

**Spouštění skriptů** ..... 7
*Dotaz bez proměnných parametrů* ..... 7
*Dotaz s proměnnými parametry* ..... 9
*Nedotazový skript* ..... 10

**Obrazovky pro správce aplikace** ..... 12
*Funkce Spuštění* ..... 12
Spustit vybraný ..... 13
Importovat skript ..... 13
*Funkce Parametry* ..... 15
Jazyk aplikace ..... 15
Adresa serveru ..... 16
Jméno uživatele ..... 16
Seznam knihoven s databázovými tabulkami ..... 16
Adresář IFS k centrálnímu uložení skriptů. ..... 16
Velikost okna s výsledky skriptu (dotazu) ..... 17
Značka pro prázdné hodnoty sloupců ..... 17
Počet mezer oddělujících sloupce ve výsledku dotazu ..... 17
Výška písma v počtu tiskových bodů ..... 17
Maska pro výstup čísel ..... 17
*Funkce Editace* ..... 18
Vytvořit nový skript ..... 19
Upravit vybraný ..... 20
Obnovit zobrazení ..... 20
Uložit do serveru ..... 20
Načíst ze serveru ..... 21
Zrušit vybraný ..... 21
*Funkce Export* ..... 21
*Funkce Import* ..... 21

**Obrazovky aplikace pro uživatele** ..... 22
*Funkce Spuštění* ..... 22
*Funkce Parametry* ..... 22

**Pravidla pro tvorbu skriptů** ..... 23
*Definice proměnných hodnot parametrů* ..... 23
Údaje v definičním řádku ..... 24
Zadání hodnot parametrů při spouštění skriptu ..... 25
Příklad zadání parametrů ..... 25
*Definice titulních hlaviček* ..... 27
*Definice sloupcových hlaviček* ..... 28
Vlastní hlavičky ..... 28
Standardní hlavičky ..... 28
*Definice vertikálního a horizontálního členění* ..... 29
Příklad ..... 29
*Úprava čísel pro výstup* ..... 30
Symboly pro tvorbu masek ..... 30
Příklady masek ..... 31
Zadání masek ..... 31
Formát čísel bez masky ..... 32

*Vynechání sloupců z výstupu* ..... *33*
Příklad na vynechání sloupců a tisk titulních hlaviček ..... 34
*Parametry pro tisk* ..... *35*
*Sumarizace výsledků dotazu* ..... *36*
Definice skupinových úrovní ..... 36
Definice sumarizovaných sloupců ..... 36
Úvodní texty pro sumární řádky ..... 37
Příklad 1 ..... 38
Příklad 2 ..... 39
**Práce s členy fyzických a logických souborů ..... 40**
*Vytvoření fyzického a logického souboru se členy* ..... *40*
*Skript k vytvoření alias objektů a zápisu dat do členů* ..... *41*
*Kontrolní výpis dat ze členů logického souboru* ..... *43*

# Úvod

Motivem k vytvoření této aplikace bylo zjištění, že populární program Query/400 (s pozdějším názvem Query for i) není schopen zobrazovat a tisknout všechny znaky kódované v soustavě UCS-2 (CCSID 13488), UTF-16 (CCSID 1200) nebo UTF-8 (CCSID 1208).

Aplikace dovoluje vytvářet, ukládat a spouštět SQL skripty pro databázi IBM DB2 for i. Skript je text příkazů jazyka SQL uložený v textovém souboru s koncovkou *.sql*. Do skriptu lze zapsat jeden nebo více příkazů SQL oddělených středníkem. Příkazy mohou být libovolného druhu a mohou obsahovat parametry označené otazníky. Nejčastěji ovšem půjde o dotazy, to znamená příkazy SELECT. Z uvedeného plyne, že tvůrce skriptů musí znát jazyk SQL alespoň na úrovni příkazu SELECT.

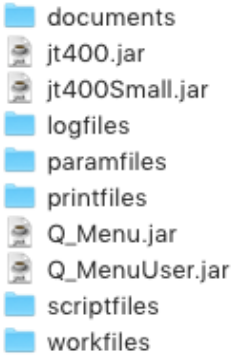
Do skriptů se kromě SQL příkazů obyčejně doplňují ještě další příkazy ve formě speciálních komentářových řádků, které dovolují pružně formátovat výstup výsledků na obrazovce a tisku na papír.

Programy jsou napsány v jazyku Java a vyžadují verzi *Java SE 8* nebo vyšší. Spolupracují s programy soustavy *IBM i Toolbox for Java* (nebo JTOpen). Programy byly vytvořeny a testovány v systémech Mac OS X, macOS a Windows 7, Windows 10 se vzdáleným internetovým připojením k systému IBM i.

Aplikace se neinstaluje, je připravena k okamžitému použití (viz Použití aplikace níže).

## Použití aplikace

Aaplikace se neinstaluje, je celá předávána jako adresář, v němž jsou začleněny další adresáře a soubory.




### *Adresáře*

- *documents* - obsahuje tento dokument v českém a anglickém jazyce,
- *logfiles* - obsahuje textové soubory *err.txt* a *out.txt*, do nichž se zapisuje přesměrovaný výstup ze souborů System.err a System.out (tj. z konzoly).
- *paramfiles* - obsahuje soubor *Parameters.txt* s parametry aplikace,
- *printfiles* - obsahuje textové soubory, které jsou výsledkem běhu skriptů,
- *scriptfiles* - obsahuje soubory skriptů, jejichž jména končí příponou .sql,
- *workfiles* - obsahuje textový soubor *Print.txt*, výsledek posledního běhu skriptu.

Poznámka: Soubory *err.txt* a *out.txt* slouží ke zjištění původu případné chyby v programu.

### *Programové soubory*

- Soubor *jt400Small.jar* - obsahuje podmnožinu tříd ze soustavy IBM i Toolbox for Java.
- Soubor *Q_Menu.jar* - obsahuje třídy Java programů aplikace a spouští aplikaci pro správce aplikace a pro tvůrce skriptů.
- Soubor *Q_MenuUser.jar* - obsahuje třídy Java programů aplikace a spouští aplikaci pro uživatele.

### *Umístění a spouštění aplikace*

Aplikační adresář umístíme na vhodné místo a případně přejmenovat. V něm můžeme vytvořit zástupce (shortcut, alias) souborů *Q_Menu.jar* (programy pro správce) a *Q_MenuUser.jar* (programy pro uživatele). Zástupce pak můžeme umístit třeba na plochu nebo jinam. Programy spustíme poklepáním na zástupce (nebo na originály) souborů *.jar*; ukáže se hlavní nabídka aplikace.

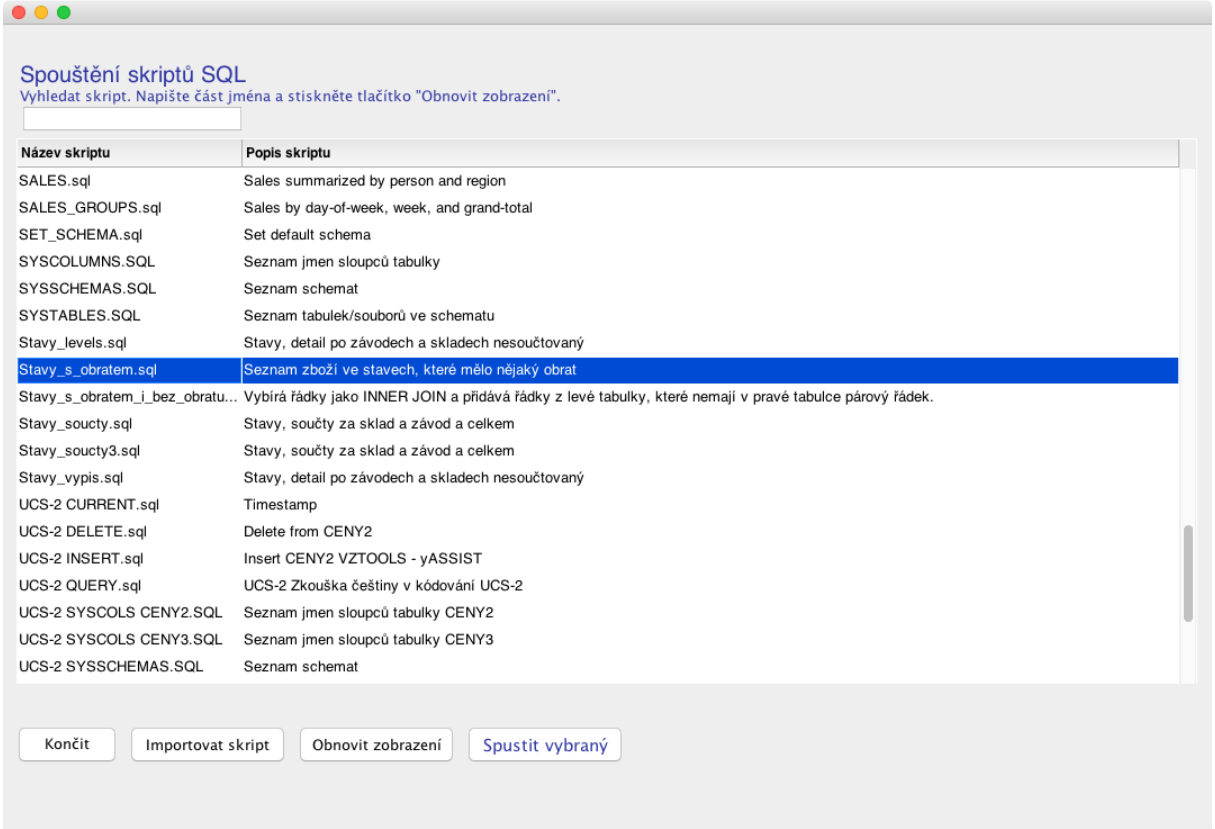
Stejná kopie aplikace funguje v systémech macOS a Windows.

## Spouštění skriptů

Z hlavní nabídky aplikace se po stisku tlačítka *Spuštění* zobrazí okno s nadpisem “Spouštění skriptů SQL” obsahující seznam skriptů. Zpočátku je seznam skriptů prázdný. Aby se do něj dostaly záznamy, musí správce aplikace vytvořit skripty pomocí funkce Editace (viz níže).

V této kapitole je spouštění skriptů ilustrováno na příkladech.

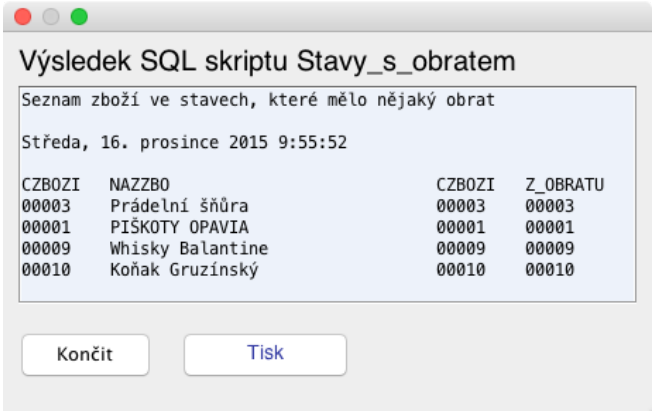
### *Dotaz bez proměnných parametrů*




V seznamu vybereme např. řádek s názvem skriptu “Stavy_s_obratem.sql” a tlačítkem *Spustit vybraný* spustíme dotaz

```
--Seznam zboží ve stavech, které mělo nějaký obrat
SELECT DISTINCT C.CZBOZI, C.NAZZBO, S.CZBOZI, O.CZBOZI Z_OBRATU
  FROM CENY C
INNER JOIN STAVY S ON C. CZBOZI = S. CZBOZI
INNER JOIN OBRATY O ON S. CZBOZI = O. CZBOZI
```

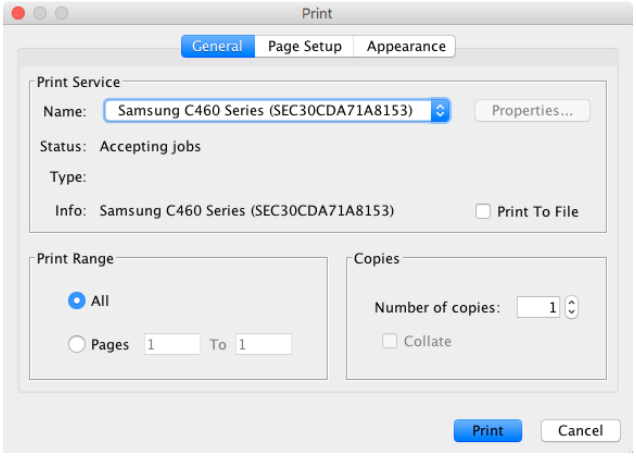
Výsledek dotazu se zobrazí v okně.




Výsledek se zároveň zapíše do souboru “Print.txt” v adresáři “workfiles” a navíc do souboru “Stavy_s_obratem.txt” v adresáři “printfiles”. V hlavičce nad sloupci dat jsou standardní nadpisy shodné se jmény sloupců v příkazu SQL.

Nyní se uživatel může spokojit s výsledkem a ukončit dotaz, nebo může výsledky vytisknout na tiskárně použitím tlačítka *Tisk*.

K provedení tisku se zobrazí dialogové okno podobné tomuto.



### *Dotaz s proměnnými parametry*

Přikaz SQL může obsahovat parametry označené otazníkem. V seznamu vybereme např. řádek s názvem skriptu *Q0010.sql*. Tento skript má v příkazu SELECT definovaná tři místa proměnných hodnot označená otazníky.

```
--Součty cen větších než zadaná cena v rozmezí čisel zboží

--;? 01; DEC; Cena větší než:; 52.5
--;? 02; CHAR; Číslo zboží od:; 00000
--;? 03; CHAR; Číslo zboží do:; 00999

SELECT S.ZAVOD, S.CZBOZI,
      DECIMAL( SUM(C.CENAJ*S.MNOZSTVI), 9, 2 ) AS CELKEM
FROM STAVY AS S
   INNER JOIN CENY AS C ON S.CZBOZI = C.CZBOZI
   WHERE C.CENAJ*S.MNOZSTVI > ?
   GROUP BY S.ZAVOD, S.CZBOZI
   HAVING S.CZBOZI BETWEEN ? AND ?
   ORDER BY S.ZAVOD, S.CZBOZI
```

Tři komentářové řádky (vyznačené kurzívou) začínají čtyřznakem **`--;?`** v prvním sloupci a definují čtveřici údajů oddělených středníkem. Umožňují sestavit výzvu uživateli k zadání proměnných hodnot na místa otazníků.

*První* údaj je pořadové číslo, které musí přesně odpovídat pořadí daného otazníku v příkazu. Zde je to 01 pro první, 02 pro druhý a 03 pro třetí otazník. Tato čísla mohou, ale nemusí obsahovat vedoucí nuly.

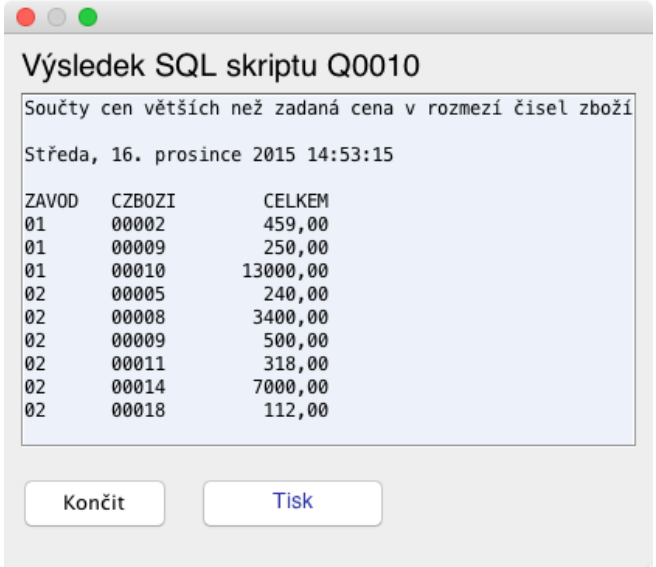
*Druhý* údaj je typ dat pro sloupec v SQL (v tomto příkladu je to DEC a CHAR). Slouží pro vlastní kontrolu správnosti zadané hodnoty a pro dokumentaci. Údaj může být i prázdný, ale nesmí chybět jeho koncový středník.

*Třetí* údaj je doprovodný text k zadávané hodnotě, zde “Cena větší než:”, “Číslo zboží od:” a “Číslo zboží do:”. Také tento údaj může být prázdný, ale s koncovým středníkem.

*Čtvrtý* údaj je předvolená hodnota odpovídající typu dat, zde je to číslo 52.5 a znakově kódovaná čísla zboží “00000” a “00999”. Tento údaj může, ale nemusí končit středníkem.

Stiskem tlačítka *Spustit vybraný* se zobrazí okno s výzvou k zadání hodnot pro tyto parametry.

Ponecháme-li předvolené údaje, zobrazí se po stisku tlačítka *Potvrdit* okno s výsledkem dotazu.




Stejně jako v předchozím dotazu se nyní uživatel může spokojit s výsledkem a ukončit dotaz, nebo může výsledky vytisknout na tiskárně.

### *Nedotazový skript*

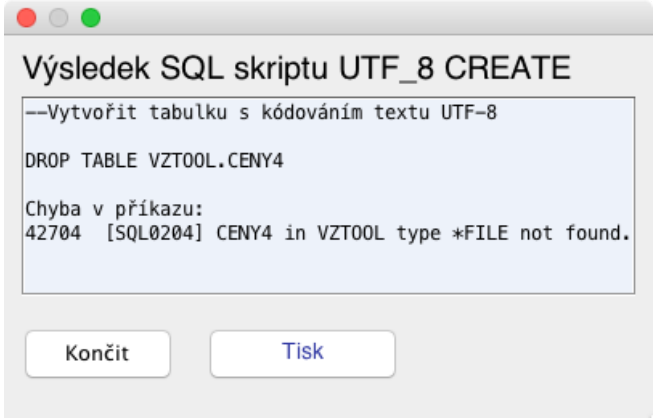
Jako ilustrace nedotazového skriptu je uvedeno několik SQL příkazů s tabulkou CENY4 ve schematu (knihovně) VZTOOL:

- DROP TABLE - zrušení tabulky,
- CREATE TABLE - vytvoření tabulky,
- SET SCHEMA - obnova původního předvoleného schematu.

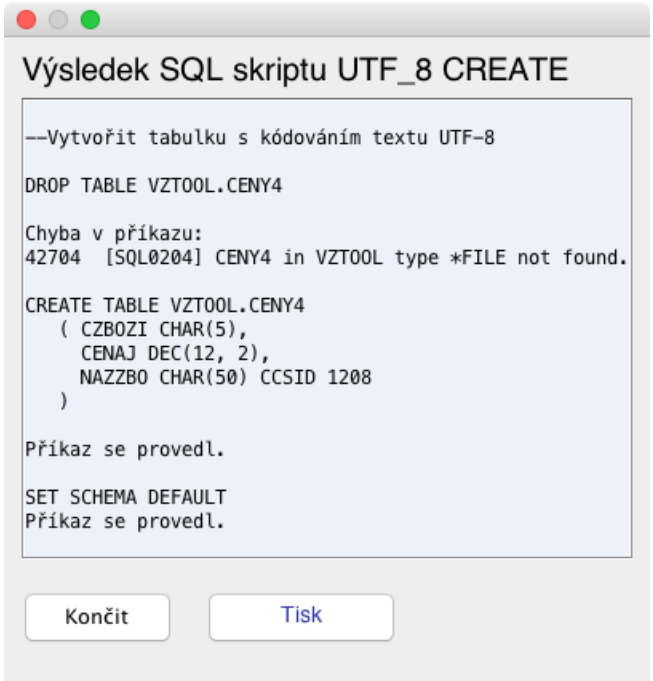
Každý příkaz kromě posledního je ukončen středníkem. U příkazů DROP a CREATE je jméno tabulky kvalifikováno jménem schematu.

```
--Vytvořit tabulku s kódováním textu UTF-8

DROP TABLE VZTOOL.CENY4
;

CREATE TABLE VZTOOL.CENY4
   ( CZBOZI CHAR(5),
     CENAJ DEC(12, 2),
     NAZZBO CHAR(50) CCSID 1208
   )
;

SET SCHEMA DEFAULT
```

Dojde-li při provádění skriptu k chybě (v příkazu DROP tabulka CENY4 neexistuje), zobrazí program chybný příkaz s chybovou zprávou v samostatném okně. Po stisku tlačítka *Končit* pokračuje v dalším výpočtu.




Jestliže potom nedojde k chybě, provedou se další příkazy. Nakonec se zobrazí okno se souhrnem všech příkazů a zpráv.

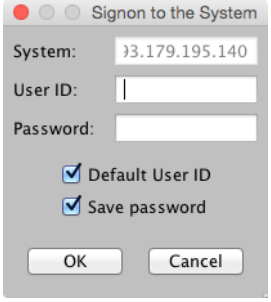


## Obrazovky pro správce aplikace

Při spuštění aplikace se jako první zobrazí nabídka funkcí.

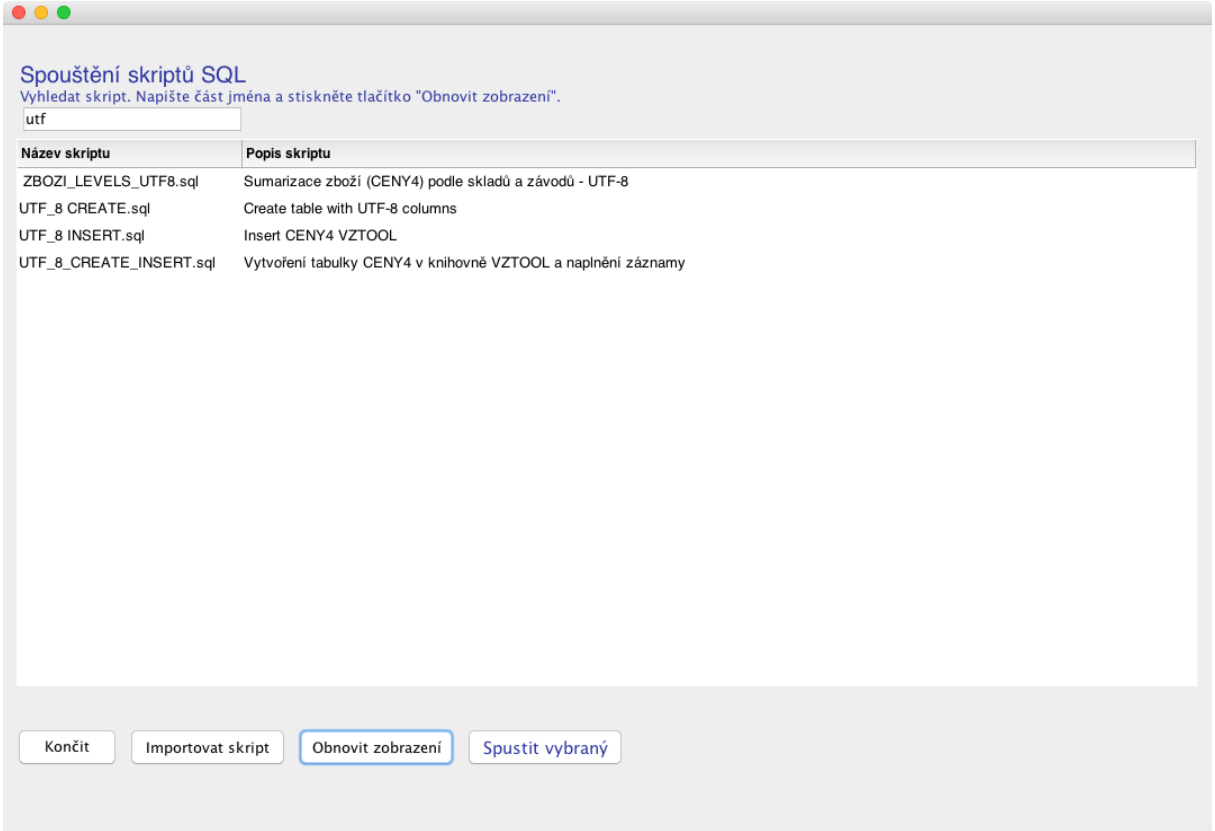
### *Funkce Spuštění*

Program nejprve vyvolá okno pro zadání přihlašovacích údajů k přístupu do systému IBM i.




Po úspěšném přihlášení zobrazí seznam skriptů s tlačítky a se vstupním polem určeným pro *výběru skriptů podle části jména*. Zapíše-li uživatel do pole nějaký text a stiskne tlačítko *Obnovit zobrazení*, zobrazí se v tabulce jen skripty v jejichž jméně se dotyčný text vyskytuje, a to bez ohledu na velikost písmen.

Zapíšeme-li do vstupního pole text *utf*, zobrazí se jen skripty, v jejichž jméně se vyskytuje text UTF s jakoukoliv velikostí písmen.




### Spustit vybraný

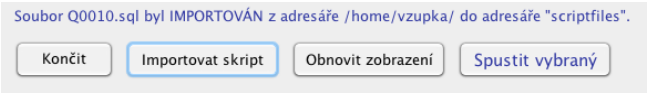
Ze seznamu skriptů uživatel jeden vybere a spustí jej tlačítkem *Spustit vybraný*.

### Importovat skript

V seznamu skriptů může uživatel načíst *opravený* nebo *nový* skript z centrálního úložiště, kam ho zapsal správce skriptů. Úložištěm je adresář v systému IBM i, který určí správce (viz dále Funkce Parametry). Nový skript se uloží do lokálního adresáře “scriptfiles”.

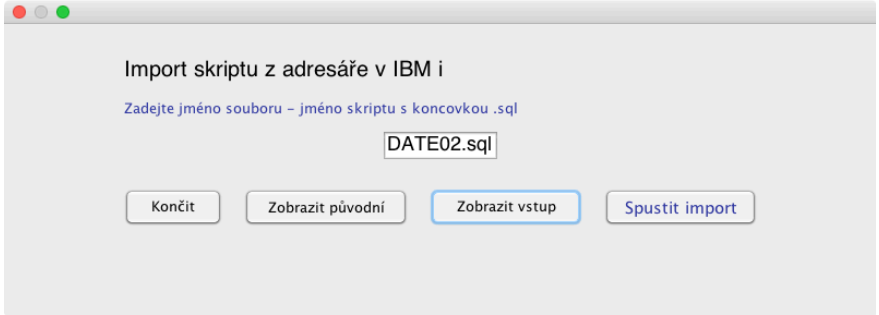
Jestliže je v seznamu např. vybrán řádek se skriptem Q0010.sql, pak tlačítko *Importovat skript* vyvolá okno pro zadání přihlašovacích údajů k přístupu do systému IBM i (není-li uživatel již přihlášen). Po přihlášení se zobrazí výzva s vyplněným jménem souboru, tj. jménem skriptu s příponou .sql.

Po stisku tlačítka Spustit import program importuje skript a zobrazí o tom zprávu.



Není-li v seznamu vybrán žádný řádek, zobrazí se nabídka pro import nového skriptu s předepsanou příponou .sql.

Do vstupního pole je třeba zapsat jméno importovaného skriptu, např. DATE02, a zachovat příponu .sql.

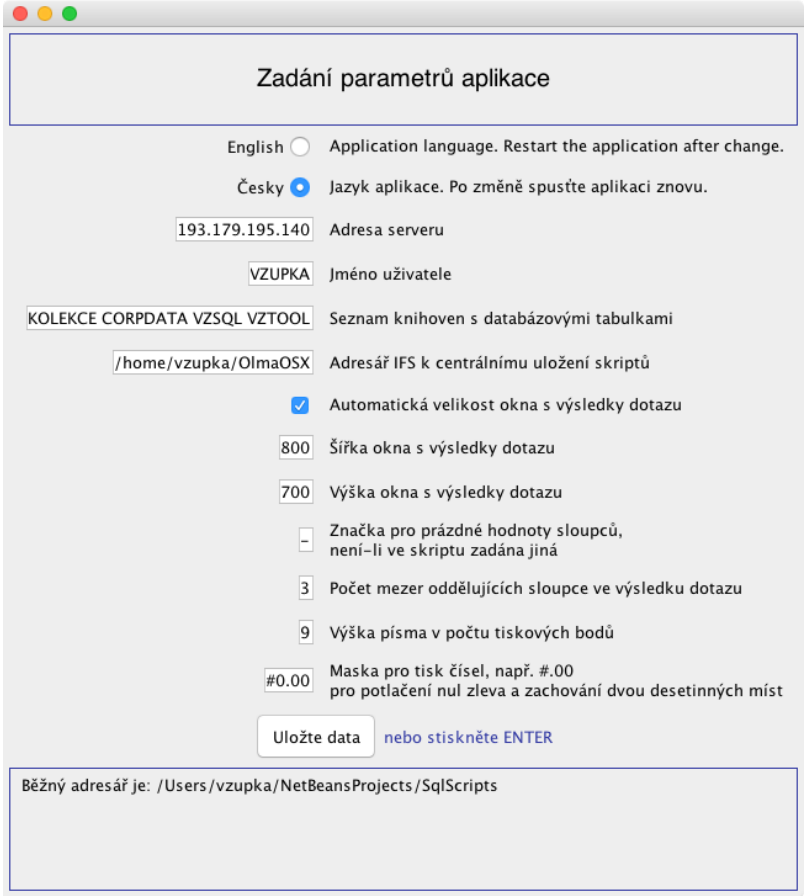



Tlačítko *Spustit import* nebo klávesa *Enter* spustí přenos z IBM i.

Tlačítko *Zobrazit původní* obnoví původní text ve vstupním poli.

Tlačítko *Zobrazit vstup* slouží ke zviditelnění zapsaného údaje v uzpůsobeném vstupním poli.

### *Funkce Parametry*




*Každý uživatel* může zadat jméno svého profilu pro přístup k systému IBM i. Toto jméno pak bude předvoleno v přihlašovacím dialogu *Signon to the System*.

*Správce aplikace* může měnit všechny hodnoty, ale především musí zadat

- adresu serveru,
- seznam knihoven s databázovými tabulkami,
- cestu k adresáři IFS pro centrální uložení skriptů.

*Běžný uživatel* (viz níže Obrazovky aplikace pro uživatele) může volit

- jazyk aplikace (anglický nebo český),
- automatickou velikost výsledkového okna nebo určit jeho pevné rozměry,
- počet mezer oddělujících sloupce ve výsledku dotazu,
- výšku písma pro tiskový výstup.

*Tvůrce skriptů* volí všem skriptům společnou

- předvolenou značku pro prázdné hodnoty sloupců,
- předvolenou masku pro formátování dekadických čísel.

#### Jazyk aplikace

Aplikace může být provozována v anglické (en_US) nebo české (cs_CZ) lokalizaci. Lokalizace se týká nadpisů, zpráv, označení tlačítek, úpravy čísel pro výstup výsledků a úpravy data a času v hlavičce výstupu. Uživatel si může zvolit, ve které lokalizaci bude

aplikaci provozovat. Po změně jazyka se volba plně uplatní (včetně úvodní obrazovky), když uživatel ukončí aplikaci a spustí ji znovu.

### Adresa serveru

Je nutné zadat jedinou IP adresu v podobě tečkové nebo doménové.

### Jméno uživatele

Správce zadá jméno profilu, který má oprávnění zapisovat soubory do IFS adresáře. Bude-li provádět manipulační SQL příkazy (jiné než SELECT), musí mít oprávnění pro zápis a přepis dat v databázových tabulkách (souborech), popřípadě k manipulaci s celými tabulkami (vytváření, rušení).

### Seznam knihoven s databázovými tabulkami

Aplikace používá ke zpracování SQL příkazů jmennou konvenci "*system*". Správce zapíše jedno nebo několik jmen knihoven oddělených čárkou nebo mezerou, anebo žádné jméno. Jméno může být také *LIBL.

V době připojení databáze k programu se vytvoří seznam knihoven, který nahradí seznam z úlohy databázového serveru (server job). Při zpracování SQL příkazu se prohledává seznam knihoven v zapsaném pořadí. Je-li prvním jménem *LIBL, připojí se zadaný seznam k seznamu z úlohy serveru.

Upozornění: Přednost má ovšem jméno knihovny (schematu) zadané přímo v SQL příkazu jako kvalifikáror před jménem tabulky (nebo jiného objektu). Oddělovač ve kvalifikaci může být *tečka* (.) nebo *lomítko* (/). Jakmile je takový příkaz použit, další příkazy běžící v pořadí hledají objekty pouze v této knihovně (schématu). Nenajdou-li je tam, systém to ohlásí chybovou zprávou. Pro nápravu lze použít dvojí opatření:

- Na konci skriptu zapíšeme příkaz SET SCHEMA DEFAULT, který aktivuje původní seznam knihoven.
- Vytvoříme skript s jediným příkazem SET SCHEMA DEFAULT a provedeme jej předtím než spustíme příkaz, který ohlásil nenalezení požadovaného objektu.

### Adresář IFS k centrálnímu uložení skriptů.

Adresář IFS slouží jako centrální úložiště k ukládání (export) a načtení (import) skriptů do/ze serveru IBM i. Není-li zadána cesta k adresáři, nelze tyto funkce používat. Cesta k adresáři musí začínat lomítkem / a může, ale nemusí lomítkem končit. Aplikace si zajistí sama správnost ukončení.

Adresář musí být předem vytvořen pomocí CL příkazu CRTDIR (MKDIR, MD), např.

```
CRTDIR DIR('/home/vzupka/OlmaOSX')
```

V adresáři budou uloženy texty skriptů v kódování UTF-8. Toto kódování lze určit pro adresář i všechny jeho soubory příkazem

```
CHGATR OBJ('/home/vzupka/OlmaOSX') ATR(*CCSID) VALUE(1208) SUBTREE(*ALL)
```

Usnadní se tak pro prohlížení skriptů v emulaci 5250 pomocí CL příkazu WRKLNK.

### Velikost okna s výsledky skriptu (dotazu)

Je-li zaškrtnut čtvereček “Automatická velikost okna s výsledky dotazu”, okno se přizpůsobí rozměrům zobrazených výsledků. V opačném případě bude okno zobrazeno v rozměrech uvedených v polích “Šířka okna s výsledky dotazu” a “Výška okna s výsledky dotazu” s případnými posuvníky.

Poznámka: Není-li vstupní hodnota celé číslo, změní se na 0.

### Značka pro prázdné hodnoty sloupců

Text zapsaný v tomto poli (případně prázdný znak) se bude zobrazovat (a tisknout) všude, kde je výsledná hodnota sloupce prázdná (NULL).

Poznámka: Text pro prázdnou hodnotu sloupce lze také zadávat individuálně pro jednotlivé skripty pomocí kometářového definičního řádku (viz dále).

### Počet mezer oddělujících sloupce ve výsledku dotazu

Tento údaj je číslo udávající počet mezer, které se zařadí na konec každého sloupce v řádcích hlaviček i v řádcích dat ve výsledném textu dotazu. Může být i 0. Není-li vstupní hodnota celé číslo, dosadí se 0.

Poznámka: Počet mezer lze také zadat individuálně pro jednotlivé skripty pomocí kometářového definičního řádku (viz dále).

### Výška písma v počtu tiskových bodů

Tento údaj představuje počet bodů, kde jeden bod měří 1/72 palce (inch). Využije se u tisku výsledků na papír. Není-li vstupní hodnota celé číslo, dosadí se 0.

Poznámka: Výšku písma lze také zadat individuálně pro jednotlivé skripty pomocí kometářového definičního řádku (viz dále).

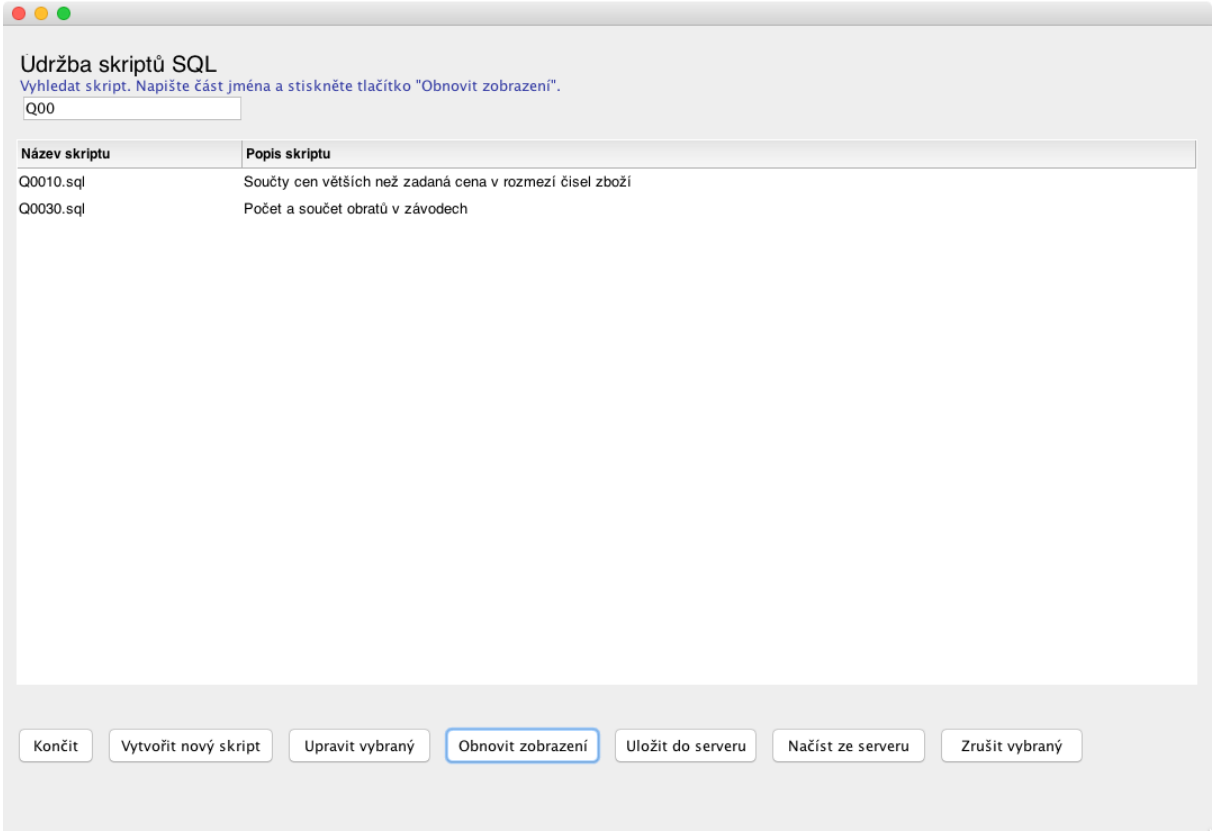
### Maska pro výstup čísel

Předvolená maska pro výstup čísel určuje výstupní obraz dekadických čísel ve všech skriptech. Nejpoužívanější bude asi maska #0.00, která potlačuje nuly zleva, ale nechává nulu před desetinným oddělovačem (čárkou) a zachovává dvě desetinná místa včetně koncových nul. Je-li zadána prázdná hodnota, uplatní se buď masky zadané pro jednotlivé sloupce nebo pravidlo pro standardní výstup čísel (viz dále).

Poznámka: Masky lze zadávat i jednotlivě pro zvolené sloupce výsledné tabulky, a to pomocí komentářových definičních řádků v každém skriptu.

### *Funkce Editace*

Zobrazí tabulku se seznamem skriptů, který je zpočátku prázdný. Je-li neprázdný, může uživatel seznam omezit pomocí textového pole, do nějž napíše text (např. Q00) a pak stiskne tlačítko *Obnovit zobrazení*.




Tato funkce dovoluje vytvořit nový skript (zejména když je seznam na začátku prázdný) nebo skript vybraný ze seznamu upravit, zrušit, uložit do serveru (do adresáře IFS zadaného v parametrech aplikace) nebo znovu načíst ze serveru.

Vytvořit nový skript

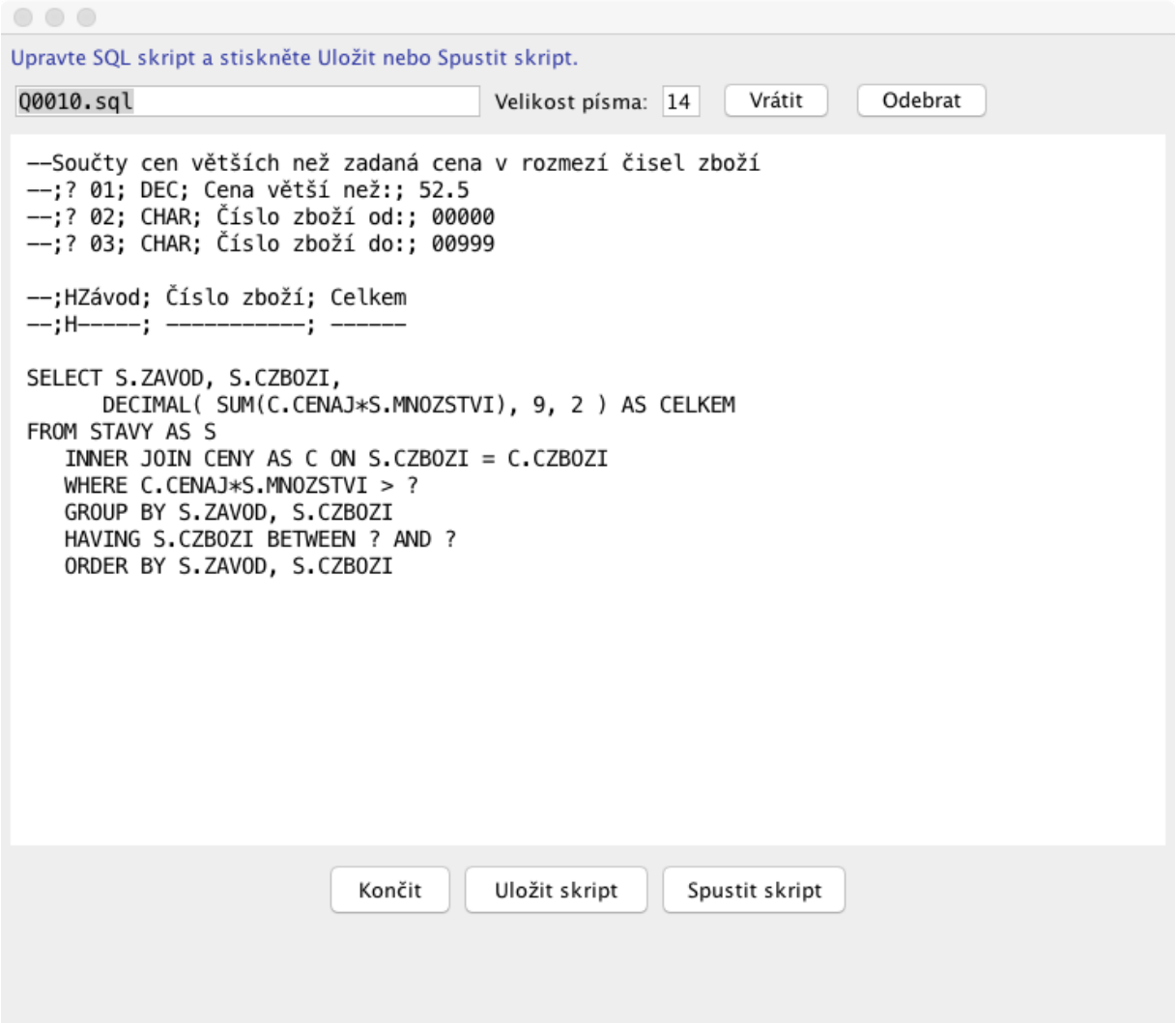
Do prázdného pole napíšeme text skriptu nebo jej přikopírujeme z jiného zdroje a zadáme jeho jméno, přičemž zachováme příponu *.sql* (může být i s jinou velikostí písmen, např. .SQL, Sql apod.).

Text skriptu můžeme upravovat běžným způsobem textových editorů: kopírovat, vyjmout, vložit pomocí klávesových zkratek. V horní části obrazovky můžeme zvolit velikost písma zobrazeného textu a použít dvě tlačítka pro opravy. Tlačítko *Vrátit* (angl. undo) vrátí text po provedené změně do předchozího stavu, zatímco tlačítko *Odebrat* (angl. redo) provede předchozí změnu, čili opět odebere vrácený text.

Tlačítko *Uložit skript* uloží skript do souboru. Tlačítko *Spustit skript*, také uloží skript do souboru a také jej spustí. Tlačítko *Končit* se vrátí k předchozí obrazovce, ale neuloží změněný text.

Poznámka: Jméno skriptu smí obsahovat jen znaky kódu *ASCII* a musí končit příponou *.sql*.

Upravit vybraný




Zobrazený text skriptu, jehož jméno je zobrazeno nahoře ve vstupním poli, můžeme upravit stejným způsobem jako při vytváření nového skriptu (viz výše).

Jestliže před uložením změníme jméno skriptu ve vstupním poli, uloží se skript s novým jménem a dosavadní skript zůstane zachován.

### Obnovit zobrazení

Smaže případnou zprávu a zruší výběr řádku ze seznamu. Je-li v textovém poli zadán text, zobrazí v tabulce jen skripty, v jejichž jméně je text obsažen (bez ohledu na velikost písmen). Je-li text prázdný, zobrazí všechny skripty.

### Uložit do serveru

Vyvolá okno pro zadání přihlašovacích údajů k přístupu do systému IBM i. Po úspěšném přihlášení spustí přenos *vybraného skriptu* z lokálního adresáře “scriptfiles” do adresáře IFS zadaného v parametrech. O přenosu zobrazí zprávu.

Načíst ze serveru

Vyvolá okno pro zadání přihlašovacích údajů k přístupu do systému IBM i. Po úspěšném přihlášení spustí přenos *vybraného skriptu* z adresáře IFS zadaného v parametrech do lokálního adresáře “scriptfiles”. O přenosu zobrazí zprávu.

Zrušit vybraný

Skript z vybraného řádku se rovnou zruší v lokálním adresáři “scriptfiles” (bez kontrolního dotazu). Dále se zobrazí okno pro zadání přihlašovacích údajů k přístupu do systému IBM i. Po úspěšném přihlášení se skript zruší i v adresáři IFS. Jestliže přihlášení nedokončíme (zrušíme), program ohlásí chybovou zprávu a skript v adresáři IFS zůstane zachován (v případě, že tam vůbec je). Není-li vybrán žádný řádek, nic se neruší, jen se o tom zobrazí zpráva.

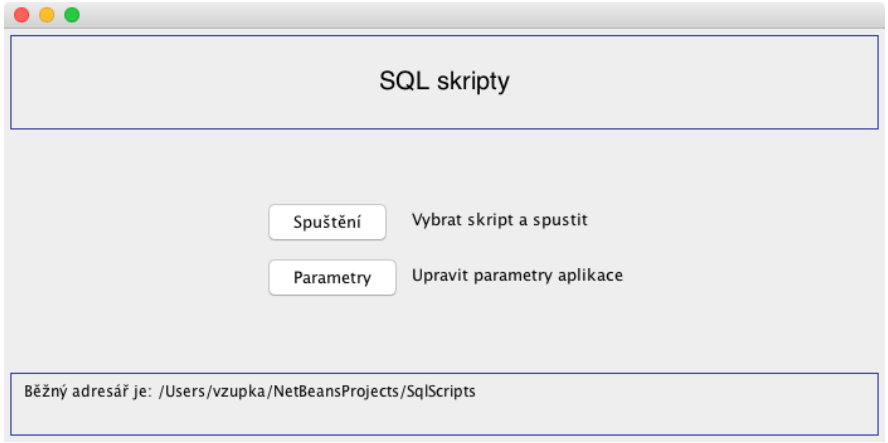
### *Funkce Export*

Tato funkce vyvolá okno pro zadání přihlašovacích údajů k přístupu do systému IBM i. Po úspěšném přihlášení spustí přenos *všech skriptů* z lokálního adresáře “scriptfiles” do adresáře IFS zadaného v parametrech aplikace. O přenosu vypíše protokol. Jestliže přihlášení nedokončíme (zrušíme), program ohlásí chybovou zprávu.

### *Funkce Import*

Tato funkce vyvolá okno pro zadání přihlašovacích údajů k přístupu do systému IBM i. Po úspěšném přihlášení spustí přenos *všech skriptů* z adresáře IFS zadaného v parametrech aplikace do lokálního adresáře “scriptfiles”. O přenosu vypíše protokol. Jestliže přihlášení nedokončíme (zrušíme), program ohlásí chybovou zprávu.

## Obrazovky aplikace pro uživatele

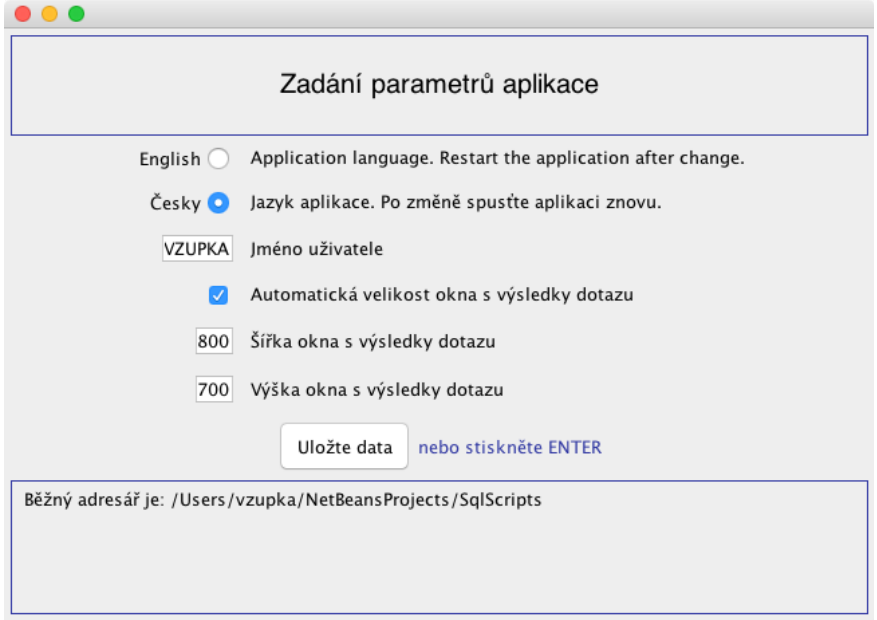



Tato hlavní obrazovka je zkrácená proti té, která je určena pro správce aplikace a tvůrce skriptů.

### *Funkce Spuštění*

Tato funkce je totožná s tou, která je popsána výše.

### *Funkce Parametry*




Tato funkce je zjednodušenou verzí fukce určené pro správce aplikace a tvůrce skriptů (viz výše). Nabízí jen ty parametry, které využije běžný uživatel.

## Pravidla pro tvorbu skriptů

Obsahem skriptu je především text příkazu SQL nebo několika příkazů oddělených středníkem. Poslední příkaz končí posledním řádkem. Neukončuje se středníkem (byl by to prázdný příkaz). Kromě příkazů může skript obsahovat komentáře, a to jednoduché a složené.

*Jednoduchý komentář* začíná dvěma souvislými pomlčkami (znaky --) a končí na konci textového řádku. Jednoduchý komentář je použit k zadávání speciálních parametrických údajů pro program, který je vyhodnocuje před provedením nebo po provedení skriptu.

*Složený komentář* začíná znaky /* a končí znaky */, podobně jako v jiných programovacích jazycích. Může se vyskytovat všude, kde je v příkazu předepsána mezera. Složený komentář se k zadávání parametrů aplikace nepoužívá.

*První řádek skriptu* obsahující komentář začínající znaky -- od 1. sloupce, se považuje za *popis skriptu*. Znaky zapsané za dvojznakem -- se objeví v seznamu skriptů jako popis.

Důležitou složkou skriptů jsou *speciální komentářové řádky*, které začínají v 1. sloupci řádku speciálními čtyřznaky a představují definice následujících parametrů.

--;? proměnná hodnota parametru (v příkazu SQL označená otazníkem),
--;t titulní hlavička,
--;H sloupcový hlavičkový řádek (hlavičky sloupců),
--;T svislé a vodorovné členění řádků, symbol pro prázdné sloupce (null),
--;D masky pro výstupní formát dekadických čísel typu DECIMAL a NUMERIC,
--;O seznam sloupců, které se mají vynechat z výstupu,
--;L skupinová úroveň, její úvodní text a jméno skupinového sloupce,
--;S jméno sumarizovaného sloupce a seznam typů souhrnů (S, A, M, m, C),
--;s seznam úvodních textů pro sumární řádky (místo standardních SUM, AVG, MAX, MIN, COUNT),
--;P parametry pro tisk.

Čtyřznaky byly voleny tak, aby údaje specifikované za nimi mohly obsahovat libovolné znaky kromě středníku, který údaje odděluje.

### *Definice proměnných hodnot parametrů*

Otazníky v SQL příkazu označují *značky parametrů* (parameter markers). Místo nich se před provedením příkazu dosadí určené hodnoty. Dosazování hodnot je umožněno prostřednictvím komentářových definičních řádků. Počet a pořadí definičních řádků určuje počet a pořadí vstupních polí v okně, kde uživatel zadává hodnoty. Definiční řádek začíná čtyřznakovým symbolem **`--;?`** v 1. pozici. Obecný tvar definice je následující. Hranaté závorky označují nepovinný údaj.

```
--;? pořadové-číslo; [typ-dat]; [vysvětlení]; [předvolená-hodnota] [;]
```

Definičních řádků musí být přesně tolik, kolik je v příkazu SQL otazníků. Každý řádek obsahuje za symbolem --;? přesně čtyři údaje oddělené středníkem. Vedoucí a koncové mezery v údajích se ignorují.

Poznámka: Je-li ve skriptu více SQL příkazů než jeden, mohou být proměnné parametry zadány *jen v jednom* SQL příkazu.

### Údaje v definičním řádku

#### Pořadové číslo

První údaj je povinné pořadové číslo, které musí přesně odpovídat pořadí daného otazníku v SQL příkazu. Je to *1* pro první otazník, *2* pro druhý otazník atd. Neurčuje pořadí vstupních polí v okně, to je dáno pořadím definičních řádků ve skriptu.

#### Typ dat

Druhý údaj je nepovinný a označuje některý z následujících datových typů.

DEC
DECIMAL
NUMERIC
INT
INTEGER
BIGINT
DATE
TIME
TIMESTAMP

Vyjmenované typy slouží pro předběžnou kontrolu správnosti hodnoty zadané do vstupního pole, ale také pro dokumentaci. Je-li takový typ zadán, provádí se kontrola hodnoty ještě *před spuštěním SQL příkazu*. Není-li zadán nebo je zadán jiný typ anebo libovolný text, provede se kontrola hodnoty podle typu *až v době provádění SQL příkazu*. Údaj může být prázdný, ale nesmí chybět středník.

#### Vysvětlující text

Třetí údaj je nepovinný vysvětlující text k zadávané hodnotě parametru, např. *Cena od:* . Také tento údaj může být prázdný, ale musí být ukončen středníkem.

#### Předvolená hodnota

Čtvrtý údaj je předvolená hodnota odpovídající typu dat, např. cena *5.50*. Tento údaj může být prázdný a může, ale nemusí končit středníkem.

Údaje typu DEC, DECIMAL, NUMERIC mohou obsahovat jen číslice, desetinnou *tečku* a znaménko před číslem. Desetinná čárka se při této kontrole považuje za chybu.

Údaje typu INT, INTEGER, BIGINT mohou obsahovat jen číslice a znaménko před číslem.

Údaje typu DATE musí být zadány ve formátu ISO, tedy YYYY-MM-DD, např. 2014-02-15.

Údaje typu TIME musí být zadány ve formátu ISO, tedy HH:MM:SS, např. 19:31:05.

Údaje typu TIMESTAMP (časové razítko) mohou být zadány ve formátu ISO, tedy

YYYY-MM-DD HH:MM:SS.MMMMMM, např. 2000-04-05 23:59:59.999999.

Jeho délka je 26 znaků. Mezi datem a časem je mezera. Vedoucí nuly lze vynechat z měsíce, dne, hodiny, minuty a sekundy u tvaru s oddělovači. Koncové nuly mikrosekund mohou být částečně nebo zcela vynechány. Připouští se i tvar s koncovkou 24:00:00.000000 (půlnoc).

Poznámka: Je-li typ sloupce BINARY nebo VARBINARY, je třeba předvolenou hodnotu zapsat ve tvaru s hexadecimálními znaky (0123456789abcdefABCDEF). Jednomu bajtu odpovídají dva hexadecimální znaky.

Řádky definic mohou být zapsány v libovolném pořadí; důležité je jejich pořadové číslo.

Příklad definic:

```
--;?01 ; ; Cena od: ; 5.50;
--;? 02;DEC; Cena do:; 23000
--;? 03; DATE; Od data:;2014-01-29;
--;? 04;  ; Binární data - dva bajty: ; 0F9C;
```

### Zadání hodnot parametrů při spouštění skriptu

V okně výzvy k zadání proměnných parametrů se kontrolují jak data z definic, tak i data, která zadá uživatel ve vstupních polích. Kontrolují se, zda jsou v souladu se *zadaným* typem (v definici --;?), a to ještě předtím, než se spustí provádění skriptu (SQL příkazu).

- U typu DEC, DECIMAL, NUMERIC se kontroluje správnost zadaného čísla. Číslo musí obsahovat jen číslice, znaménko (plus nebo minus) vlevo a desetinnou *tečku*. Desetinná *čárka* se považuje za chybu a program ji ihned ji ohlásí.
  Není-li typ takto přesně zadán, čárka se v *české* lokalizaci nepovažuje za chybu a SQL ji přijme. V *anglické* lokalizaci se desetinná čárka považuje vždy za chybu.
- Typ DATE se kontroluje na správnost zápisu podle ISO.
- Typ TIME se kontroluje na správnost zápisu podle ISO.
- Typ TIMESTAMP se kontroluje na správnost zápisu podle ISO.

Jestliže ve specifikačním řádku typ chybí nebo je zadán jiný text, závada v zadané hodnotě se projeví jako chyba až v době provádění SQL příkazu. Zpráva SQL o chybě bude patrná z výsledku, který se po provedení skriptu zobrazí.

Poznámka 1: Je-li typ sloupce BINARY nebo VARBINARY, je třeba hodnotu zapsat ve tvaru s hexadecimálními znaky (0123456789abcdefABCDEF). Jednomu bajtu odpovídají dva hexadecimální znaky. Hodnota se nekontroluje na správnost. Za chybný znak se dosadí 0.

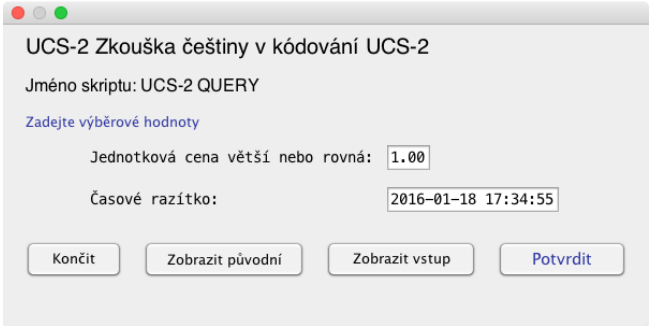
Poznámka 2: Výsledky dotazu ve sloupci typu BINARY nebo VARBINARY jsou také zobrazeny v hexadecimálním vyjádření.

### Příklad zadání parametrů

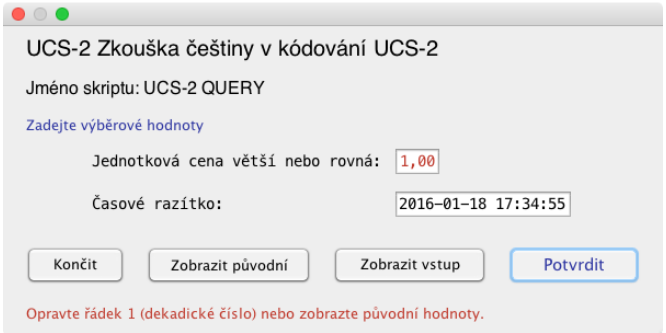
Následující skript se jménem UCS-2 definuje dva parametry.

```
--UCS-2 Zkouška češtiny v kódování UCS-2

--;? 1 ;NUMERIC; Jednotková cena větší nebo rovná: ; 1.00;
--;? 2; TIMESTAMP ; Časové razítko:; 2015-01-18 17:34:55;

select CZBOZI, CENAJ, NAZZBO, DATUM, CAS, RAZITKO from CENY2
  where CENAJ >= ?
  and RAZITKO < ?
order by CZBOZI
```

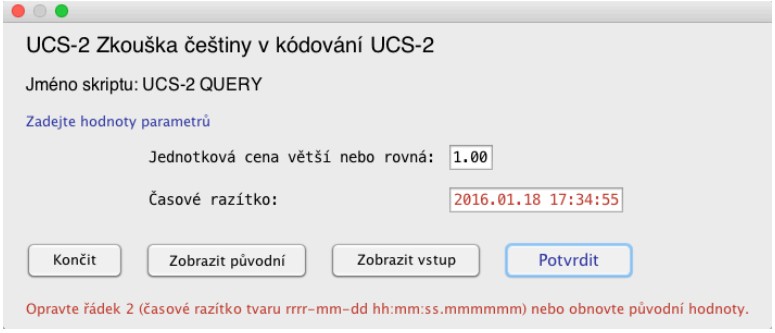
Po spuštění skriptu se zobrazí výzva ke změně hodnot a ke spuštění SQL příkazu (tlačítkem Potvrdit).




Změníme-li desetinnou tečku na čárku, zobrazí se zpráva o první chybě.

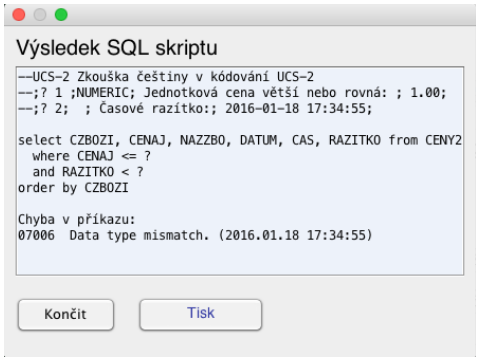



Opravíme-li desetinnou čárku zpátky na tečku, ale v časovém razítku změníme pomlčky na tečky, zobrazí se zpráva o druhé chybě.

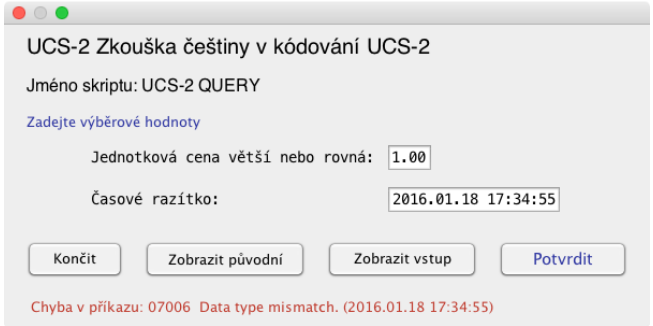



Opravíme-li i druhou chybu, dostaneme výsledek dotazu.

Jestliže v druhé definici vynecháme typ TIMESTAMP a uděláme v něm předchozí chybu (tečky v časovém razítku), spustí se skript a jeho výsledek bude okno s kopií skriptu a s chybovou zprávou z SQL.




Po uzavření výsledkového okna (tlačítkem Končit) se opět zobrazí výzva k zadání parametrů se stejnou chybovou zprávou.




### *Definice titulních hlaviček*

Pro dotazový skript můžeme definovat libovolný počet titulních hlavičkových řádků. Ty se uplatní v pořadí v jakém jsou napsány ve skriptu. Definice titulního hlavičkového řádku začíná v 1. sloupci čtyřznakovým symbolem **`--;t`**. za nímž může následovat libovolný text.

Titulní hlavičkový řádek může obsahovat libovolný text a v něm případně označení proměnných tvaru

*`&sloupec`*

kde *sloupec* je jméno sloupce, za nímž následuje mezera nebo konec řádku. Za tuto proměnnou se dosadí hodnota stejnojmenného sloupce *z prvního řádku* výsledné tabulky, i když je tento sloupec pojmenován v parametrech jako vynechaný (viz dále).

Titulní hlavičkové řádky se vypisují před sloupcovými hlavičkami a tisknou se jen na první stránce.

Například definice s proměnnými &ZAVOD a &CELKEM

```
--;t   Závod z prvního řádku: &ZAVOD , Celková cena z prvního řádku: &CELKEM ,
--;t   ~~~~~~~~~~~~~~~~~~~~~~     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--;t   Celková cena znovu: &Celkem
--;t
```

spolu s parametry pro vynechané sloupce

```
--;O zavod ; celkem
```

kde ZAVOD a CELKEM označují vynechávané sloupce, vytvoří výstup podobný tomuto:

```
Závod z prvního řádku: 02, Celková cena z prvního řádku: 500,00,
~~~~~~~~~~~~~~~~~~~~~~     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Celková cena znovu: 500,00
```

### *Definice sloupcových hlaviček*

#### Vlastní hlavičky

Pro dotazový příkaz SELECT můžeme definovat libovolný počet hlavičkových řádků. Ty se uplatní v pořadí v jakém jsou napsány ve skriptu. Hlavičkový řádek začíná v 1. sloupci čtyřznakovým symbolem **`--;H`** a za ním následuje potřebný počet textů (sloupcových nadpisů) oddělených středníkem.

```
--;H [ [text-1] [; text-2] [; ...] [; text-n] [;] ]
```

Do výstupu dotazu se berou texty včetně vedoucích a koncových mezer. Lze tak například zarovnávat víceřádkové nadpisy sloupců nebo zvětšovat šířku zobrazených sloupců dat.

Například definice

```
--;HZávod;Sklad;Množství ;Pořadové číslo;
--;H     ;     ;na skladě;
--;H=====;=====;=========;==============;
--;H
```

vytvoří ve výsledku dotazu čtyři řádky hlaviček, z nichž čtvrtý je prázdný:

```
Závod Sklad   Množství  Pořadové číslo
              na skladě
===== =====   ========= ==============

01    01         989,22              1
...
```

#### Standardní hlavičky

Není-li zadán žádný řádek definice hlaviček, vypíše se standardní hlavička s *názvy sloupců* definovaných v tabulce, která je výsledkem dotazu. Standardní hlavička může být původní, ale také přejmenovaný název sloupce. Ten může být jednoduchý (ve velkých písmenech) nebo vlastní, zapsaný v uvozovkách.

Následující skript obsahuje frázi GROUP BY ROLLUP (), která určuje seskupování výsledných řádků podle skladu a závodu. V něm jsou všechny hlavičky standardní a značení hodnoty NULL se řídí podle zadání v parametrech aplikace, kde je právě uveden znak pomlčka.

```
SELECT S.ZAVOD, S.SKLAD,
      DECIMAL( SUM(C.CENAJ*S.MNOZSTVI), 9, 2 ) AS "Součet",
      INT( AVG(C.CENAJ*S.MNOZSTVI) ) AS "Průměr",
      INT( MIN( C.CENAJ*S.MNOZSTVI) ) AS MIN,
      INT( MAX( C.CENAJ*S.MNOZSTVI) ) AS MAXIMUM
FROM STAVY AS S
   JOIN CENY AS C ON S.CZBOZI = C.CZBOZI
   GROUP BY ROLLUP( S.ZAVOD, S.SKLAD )
   ORDER BY S.ZAVOD, S.SKLAD
```

| ZAVOD | SKLAD | Součet | Průměr | MIN | MAXIMUM |
|---|---|---|---|---|---|
| 01 | 01 | 6967,99 | 2322 | 8 | 6500 |
| 01 | 02 | 6751,25 | 2250 | 1 | 6500 |
| 01 | 03 | 1,25 | 1 | 1 | 1 |
| 01 | - | 13720,49 | 1960 | 1 | 6500 |
| 02 | 01 | 3661,10 | 1220 | 21 | 3400 |
| 02 | 02 | 7930,00 | 1982 | 112 | 7000 |
| 02 | - | 11591,10 | 1655 | 21 | 7000 |
| - | - | 25311,59 | 1807 | 1 | 7000 |

### *Definice vertikálního a horizontálního členění*

Definiční řádek skupin může být jen jeden. Je-li jich více, uplatní se jen první. Začíná v 1. sloupci řádku čtyřznakovým symbolem **`--;T`** a za ním následují údaje oddělené středníkem definující

- počet prázdných řádků *před* řádkem nebo skupinou řádků,
- počet prázdných řádků *za* řádkem nebo skupinou řádků,
- symbol nahrazující prázdnou (null) hodnotu sloupce,
- počet mezer oddělujících sloupce ve výstupním řádku - od 0 výše,
- jedno nebo více jmen sloupců pro potlačení duplicitních hodnot.

```
--;T [ [počet-před] [;počet-za] [;symbol-pro-null] [;mezery-mezi-sloupci]
       [;jméno-sloupce] ... ] [;] ]
```

Parametr *počet-před* má tvar **Bn** nebo **n**, kde n je celé číslo. Parametr *počet-za* má tvar **An** nebo **n**. Parametr *mezery-mezi-sloupci* má tvar **Sn** nebo **n**. První tři parametery mohou být prázdné. Je-li některý z prvních dvou prázdný nebo chybný, dosadí se číslo *nula*. Je-li parametr *symbol-pro-null* prázdný, dosadí se hodnota z parametrů aplikace. Je-li parametr *počet-mezer* prázdný nebo chybný, dosadí se počet oddělujících mezer z parametrů aplikace.

Pátý a další parametr určuje *jméno sloupce*, u nějž se sleduje opakování stejné hodnoty ve skupně následujících řádků. Hodnota takového sloupce bude zapsána jen v prvním řádku takové skupiny, v ostatních řádcích budou mezery.

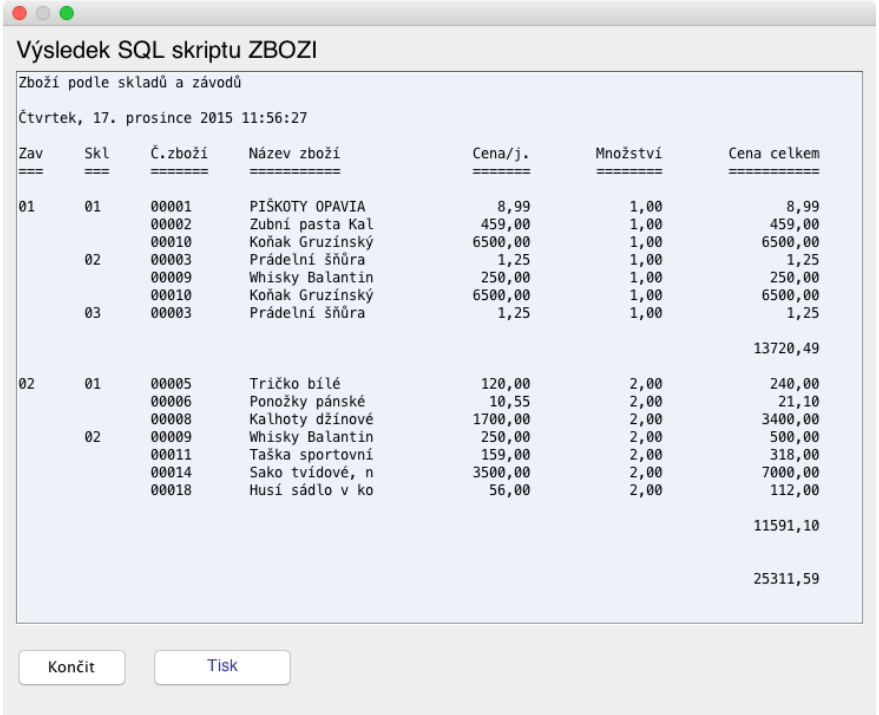
Poznámka: Prázdné řádky se zařazují do výsledného dotazu před nebo za řádky nebo skupiny řádků, které jsou výsledkem sumarizace podle skupinových úrovní (viz dále). Prázdné řádky se zařazují také před nebo za jednotlivé řádky, které obsahují prázdné hodnoty (NULL) v některém sloupci, když není zadána sumarizace podle skupinových úrovní.

#### Příklad

Následující skript ilustruje úpravu hlaviček, součtových skupin a sloupců, u nichž chceme vypisovat jen hodnotu v prvním řádku.

```
--Zboží podle skladů a závodů
--;HZav;Skl;Č.zboží;Název zboží;Cena/j.;Množství;   Cena celkem;
--;H===;===;=======;===========;=======;========;   ===========;
--;H

--;T B1; A1; ; S5; ZAVOD ; SKLAD

SELECT S.ZAVOD, S.SKLAD, S.CZBOZI,
   SUBSTR(C.NAZZBO, 1, 15) AS NAZ,
   DEC(C.CENAJ, 12, 2) AS CJ, S.MNOZSTVI,
   DECIMAL ( SUM(C.CENAJ*S.MNOZSTVI), 12, 5 ) AS CELKEM
FROM STAVY AS S
   JOIN CENY AS C ON S.CZBOZI = C.CZBOZI
   GROUP BY ROLLUP( S.ZAVOD, ( S.SKLAD, S.CZBOZI, C.NAZZBO, C.CENAJ, S.MNOZSTVI ) )
   ORDER BY S.ZAVOD, S.SKLAD, S.CZBOZI
```

Výsledek je vidět na obrázku:



Výsledek SQL skriptu ZBOZI

```
Zboží podle skladů a závodů

Čtvrtek, 17. prosince 2015 11:56:27

Zav     Skl     Č.zboží     Název zboží              Cena/j.        Množství        Cena celkem
===     ===     =======     ===========              =======        ========        ===========

01      01      00001       PIŠKOTY OPAVIA             8,99           1,00               8,99
                00002       Zubní pasta Kal          459,00           1,00             459,00
                00010       Koňak Gruzínský         6500,00           1,00            6500,00
        02      00003       Prádelní šňůra             1,25           1,00               1,25
                00009       Whisky Balantin          250,00           1,00             250,00
                00010       Koňak Gruzínský         6500,00           1,00            6500,00
        03      00003       Prádelní šňůra             1,25           1,00               1,25

                                                                                    13720,49

02      01      00005       Tričko bílé              120,00           2,00             240,00
                00006       Ponožky pánské            10,55           2,00              21,10
                00008       Kalhoty džínové         1700,00           2,00            3400,00
        02      00009       Whisky Balantin          250,00           2,00             500,00
                00011       Taška sportovní          159,00           2,00             318,00
                00014       Sako tvídové, n         3500,00           2,00            7000,00
                00018       Husí sádlo v ko           56,00           2,00             112,00

                                                                                    11591,10


                                                                                    25311,59
```

Kończit  Tisk

Zde jsou skupiny závodů označeny jen v prvním řádku skupiny. Skupiny skladů rovněž (i když se za ně nesčítá). Prázdné hodnoty sloupců jsou nahrazeny mezerou a řádkování součtů je “jeden prázdný před” a “jeden prázdný po” součtu. Sčítá se za závod a celkem.

Poznámka: Čtvrtá hodnota S5 značí počet mezer mezi sloupci. Tento parametr musí být zadán (alespoň koncovým středníkem), jestliže jsou zadána jména sloupců pro potlačení duplicit.

### *Úprava čísel pro výstup*

Dekadická čísla typu DEC, DECIMAL, NUMERIC jsou ve výsledku dotazu upravena formátem. Formát je buď *standardní* pro danou lokalizaci nebo předepsaný *maskou*.

Poznámka: Maska se neuplatňuje pro celočíselné typy INT, INTEGER, BIGINT.

#### Symboly pro tvorbu masek

V následující tabulce jsou uvedeny znaky tvořící masku, jejich umístění a význam.

| Symbol | Umístění v masce | Podléhá lokalizaci | Význam v české lokalizaci |
|---|---|---|---|
| 0 | v čísle | ano | Číslice |
| # | v čísle | ano | Číslice, přičemž nula je potlačena |
| . | v čísle | ano | Oddělovač desetinných míst - čárka |
| - | před číslem | ano | Přidá pohyblivé záporné znaménko. Neobrací číslo! |
| , | v čísle | ano | Oddělovač skupin číslic - mezera |
| % | před číslem, za číslem | ano | Stonásobek označený jako procento |

| Symbol | Umístění v masce | Podléhá lokalizaci | Význam v české lokalizaci |
|---|---|---|---|
| ¤ | před číslem, za číslem | ne | Znak pro měnu nahrazený měnovým symbolem Kč. Je-li zdvojen, je nahrazen mezinárodním měnovým symbolem - CZK. |
| ‰ | před číslem, za číslem | ano | Tisícinásobek označený jako promile |
| libovolný text | před číslem | ne | Pohyblivý text před číslem |
| libovolný text | za číslem nebo uvnitř | ne | Pevný text za číslem |

Upozornění: Zápis znaků 0 mezi posloupnost znaků # a obráceně je chyba.

### Příklady masek

Následují příklady formátování čísel maskami v české lokalizaci.

| Číslo | Maska | Výstup | Poznámka |
|---|---|---|---|
| 123456789.50 | #,###0.00 | 1 2345 6789, 50 | Oddělí čtveřice číslic mezerou |
| 1234567.50 | ,##0.00 ¤ | 1 234 567,50 Kč | Oddělí trojice číslic mezerou a přidá český symbol měny |
| 1234567.50 | #0.00 ¤¤ | 1234567,50 CZK | Přidá mezinárodní symbol lokální (zde české) měny |
| 1234567.50 | #0.00 CZK | 1234567,50 CZK | Přidá konstantu “ CZK” za číslo. Za nebo před číslo lze přidávat libovolné symboly. |
| 1234567.50 | #0**.00 | 1234567,50** | Hvězdičky se přidají těsně za číslo |
| -1234567.55 | #0.0 | -1234567,6 | Zaokrouhlí na jedno desetinné místo (polovinu na sudou číslici) |
| -0.04 | #.00 | -,04 | Potlačí nulu před desetinnou čárkou |
| 5.55 | 00000.000 | 00005,550 | Nepotlačuje vedoucí nuly a přidá nulu na konci |
| 0.56 | #0.00 % | 56 % | Vyjádření zlomku v procentech |
| 0.007 | #0.0 | 7,0 ‰ | Vyjádření zlomku v promilích |
| 12345<br>-12345 | -#0.00<br>-#0.00 | -12345<br>--12345 | Pohyblivé znaménko minus vlevo. Neobrací kladné číslo na záporné. K zápornému přidá ještě jeden minus. |

Jedna z obvyklých masek pro české prostředí je maska #0.00, která potlačuje levostranné nuly v celé části čísla (kromě poslední) a zachovává dvě desetinná místa včetně koncových nul.

### Zadání masek

Masku můžeme zadat jednak v *parametrech aplikace*, jednak v *komentářových řádcích* skriptu. Maska z parametrů aplikace (není-li prázdná) platí pro všechny číselné sloupce (i celočíselné), pro něž není individuálně zadána maska definičním komentářovým řádkem.

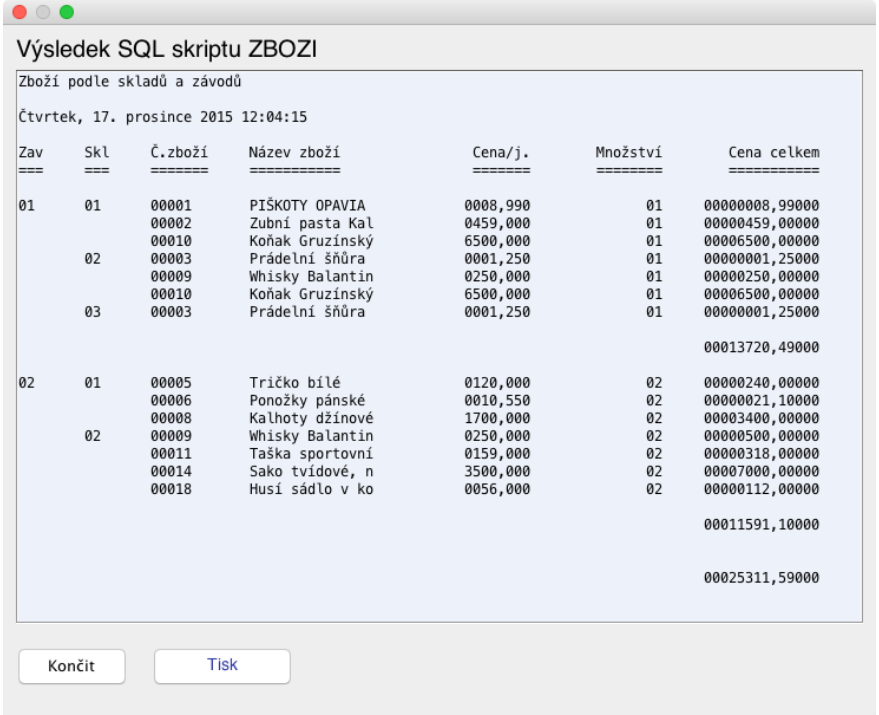
*Definiční řádek masky* začíná v 1. sloupci čtyřznakovým symbolem **`--;D,`** za ním následuje maska a za ní jméno sloupce. Může, ale nemusí být ukončen středníkem. Vedoucí a koncové mezery se odstraní. Definice masky má tvar

```
--;D maska ; jméno-sloupce [;]
```

### Příklad na výstup čísel pomocí masky

Zadáme-li v předchozím skriptu následující definice, upraví se dekadická čísla podle masek.

```
--;D 0000.000; CJ
--;D  00; MNOZSTVI
--;D 00000000.00000; CELKEM
```




### Formát čísel bez masky

Není-li zadána žádná maska, tj. ani v parametrech aplikace, znamená to, že dekadické číslo se zobrazí *podle standardních lokálních pravidel*. Standardní pravidla pro českou lokalizaci jsou:

- Vedoucí nuly se potlačí, z nich se ponechá jen nula před desetinnou čárkou.
- Nulový zlomek se vynechá i s desetinnou čárkou.
- Koncové nuly v desetinných místech se potlačí.
- Trojice číslic v celé části se oddělují mezerou.

| | | |
|---|---|---|
| 12345.00 | 12 345 | Odstraní desetinná místa, doplní mezeru mezi stovky a tisíce |
| 12345.50 | 12 345,5 | Vynechá koncovou nulu z desetinných míst |
| 0.665 | 0,66 | Zaokrouhlí na sudou číslici |

Je zřejmé, že tato možnost se v dotazech použije jen výjimečně.

Poznámka: Výstupní obrazy dekadických čísel v české lokalizaci obsahují desetinnou čárku, přestože v masce je zadána desetinná tečka.

Příklad na standardní lokalizovaný výstup čísel

Když necháme masku v parametrech aplikace *prázdnou*, a neuvedeme ani definici masky pro sloupec SOUCET ve skriptu, budou hodnoty sloupce obsahovat desetinná místa bez koncových nul (nebo žádná desetinná místa bez čárky) a mezeru mezi trojicemi číslic v celé části:

```
ZAVOD   SKLAD        SOUCET        PRUMER          MIN       MAXIMUM
01      01         6 967,99          2322            8          6500
01      02         6 751,25          2250            1          6500
01      03             1,25             1            1             1
01      -         13 720,49          1960            1          6500
02      01          3 661,1          1220           21          3400
02      02            7 930          1982          112          7000
02      -          11 591,1          1655           21          7000
-       -         25 311,59          1807            1          7000
```

### *Vynechání sloupců z výstupu*

Definiční řádek začíná čtyřznakovým symbolem **--;O** v 1. pozici.

Definičních řádků může být libovolný počet. Každý řádek obsahuje za symbolem --;O jména sloupců oddělená středníkem. Údaje mohou obsahovat vedoucí a koncové mezery, které se ignorují. Obecný tvar definice je následující. Hranaté závorky označují nepovinný údaj.

```
--;O [ sloupec; sloupec; ... [;] ]
```

Vyjmenované sloupce se vynechají z výstupu a tisku, a to jak hodnoty, tak sloupcové hlavičky.

### Příklad na vynechání sloupců a tisk titulních hlaviček

V následujícím příkladu jsou zadány titulní hlavičkové řádky s proměnnými a k tomu zadání pro vynechávané sloupce. Jak je vidět, řádků --;O může být i více. V titulních řádcích jsou zadány proměnné jako jména sloupců s předřazeným znakem & (ampersand), které shodou okolností odkazují na vynechané sloupce. Provedením následujícího skriptu se do titulních hlaviček dosadí hodnoty dotyčných sloupců z *prvního* řádku výsledku dotazu.

```
--Zboží podle skladů a závodů - vynechané sloupce a proměnné v titulcích

--;t   Závod z prvního řádku: &ZAVOD , Celková cena z prvního řádku: &CELKEM ,
--;t   ~~~~~~~~~~~~~~~~~~~~~~     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--;t   Celková cena znovu: &Celkem
--;t

--;HZav;Skl;Č.zboží;Název zboží;Cena/j.;Množství;   Cena celkem;
--;H===;===;=======;===========;=======;========;   ===========;
--;H

/* Tři mezery mezi sloupci */
--;T  ;  ;  ; 3

/* Dva vynechané sloupce */
--;O zavod ;
--;O celkem

/* Parametry pro tisk */
--;P a4; fs9; Landscape; lm5; rm30; tm5; bm5;

SELECT S.ZAVOD, S.SKLAD, S.CZBOZI, C.NAZZBO,
   DEC(C.CENAJ, 12, 2) AS CJ,   S.MNOZSTVI,
   DECIMAL (C.CENAJ*S.MNOZSTVI, 12, 2 ) AS CELKEM
FROM STAVY AS S
   JOIN CENY AS C ON C.CZBOZI = S.CZBOZI
   WHERE S.ZAVOD = '02' AND S.CZBOZI <> '00002'
   ORDER BY S.SKLAD desc, S.CZBOZI
```

Vypíše se:

```
Zboží podle skladů a závodů - vynechané sloupce a proměnné v titulcích

Čtvrtek, 3. prosince 2015 17:32:53

   Závod z prvního řádku: 02, Celková cena z prvního řádku: 500,00,
   ~~~~~~~~~~~~~~~~~~~~~~     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   Celková cena znovu: 500,00

Skl   Č.zboží   Název zboží                              Cena/j.      Množství
===   =======   ===========                              =======      ========

02    00009     Whisky Balantine                          250,00          2,00
02    00011     Taška sportovní                           159,00          2,00
02    00014     Sako tvídové, nadměr                     3500,00          2,00
02    00018     Husí sádlo v konzervě                      56,00          2,00
01    00005     Tričko bílé                               120,00          2,00
01    00006     Ponožky pánské bílé, nové                  10,55          2,00
01    00008     Kalhoty džínové                          1700,00          2,00
```

Kdybychom nezadali vynechávané sloupce a titulní hlavičky, dostali bychom tento výsledek:

```
Zboží podle skladů a závodů - vynechané sloupce a proměnné v titulcích

Čtvrtek, 3. prosince 2015 17:30:39

   Závod z prvního řádku: 02, Celková cena z prvního řádku: 500,00,
   ~~~~~~~~~~~~~~~~~~~~~~     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   Celková cena znovu: 500,00

Zav   Skl   Č.zboží   Název zboží                            Cena/j.      Množství      Cena celkem
===   ===   =======   ===========                            =======      ========      ===========

02    02    00009     Whisky Balantine                        250,00          2,00           500,00
02    02    00011     Taška sportovní                         159,00          2,00           318,00
02    02    00014     Sako tvídové, nadměr                   3500,00          2,00          7000,00
02    02    00018     Husí sádlo v konzervě                    56,00          2,00           112,00
02    01    00005     Tričko bílé                             120,00          2,00           240,00
02    01    00006     Ponožky pánské bílé, nové                10,55          2,00            21,10
02    01    00008     Kalhoty džínové                        1700,00          2,00          3400,00
```

Do titulních hlaviček se tedy správně dostala hodnota 01 ze sloupce ZAVOD a hodnota 500,00 ze sloupce CELKEM.

Poznámka: Není nutné vázat titulní hlavičky s vynechávanými sloupci. Obojí lze používat samostatně.

### *Parametry pro tisk*

Definiční řádek parametrů pro tisk může být jen jeden. Je-li jich více, uplatní se jen první. Začíná v 1. sloupci řádku čtyřznakovým symbolem **`--;P`** a za ním následují údaje oddělené středníkem definující

- velikost papíru A4 nebo A3,
- velikost písma tvaru FSn, kde n je počet tiskových bodů, např. fs9,
- orientaci stránky PORTRAIT či P, nebo LANDSCAPE či L,
- levý okraj (left margin) tvaru LMn, kde n je počet milimetrů od okraje tisknutelné oblasti papíru,
- pravý okraj (right margin) tvaru RMn,
- horní okraj (top margin) tvaru TMn,
- spodní okraj (bottom margin) tvaru BMn,

Definiční řádek má tvar

```
--;P  An; FSn; orientace; LMn; RMn ; TMn ;BMn [;]
```

Řádek může, ale nemusí být ukončen středníkem. Jednotlivé údaje mohou být vynechány, ale počet středníků do prvního neprázdného údaje musí být zachován.

- Je-li údaj velikosti papíru prázdný nebo je chybný, dosadí se A4.
- Je-li údaj velikosti písma prázdný nebo je nečíselný, dosadí se jeho hodnota z parametrů aplikace.
- Je-li údaj orientace stránky prázdný nebo je chybný, dosadí se PORTRAIT.
- Je-li údaj okraje prázdný nebo je chybný, dosadí se 10 mm.

### *Sumarizace výsledků dotazu*

Výsledek dotazu (result set) lze sumarizovat podle skupinových úrovní (level summarization). Počet úrovní není výslovně omezen. Skupinovou sumarizaci definuje soustava tří typů parametrických definic:

--;L skupinová úroveň, její úvodní text a jméno skupinového sloupce,
--;S jméno sumarizovaného sloupce a seznam typů souhrnů (S, A, M, m, C),
--;s seznam úvodních textů pro sumární řádky (místo standardních SUM, AVG, MAX, MIN, COUNT),

Řádků --;L je tolik, kolik skupinových úrovní se používá. Řádků --;S je tolik, kolik sloupců se podrobuje sumarizaci. Řádek --;s je nejvýše jeden. Řádky --S; a --s; mají smysl jen tehdy, jsou-li zadány nějaké řádky --;L.

#### Definice skupinových úrovní

Definiční řádek začíná čtyřznakovým symbolem **`--;L`** v 1. pozici. Obecný tvar definice je následující.

```
--;L [level]; [úvodní-text-skupiny]; [sloupec-úrovně]; [NP] [;]
```

První údaj *level* je buď 0 nebo libovolný text, i prázdný. Pro názornost se volí čísla 1, 2, atd.

Druhý údaj *úvodní-text-skupiny* je libovolný text, který navíc může obsahovat *proměnné* tvaru `&sloupec`, za nímž následuje mezera nebo konec řádku. Na místo této proměnné se dosadí hodnota stejnojmenného sloupce *z posledního řádku* skupiny dané úrovně, a to i tehdy, když je tento sloupec pojmenován v parametrech --;O jako vynechaný.

Třetí údaj *jméno-sloupce* představuje sloupec definující skupinu dané úrovně. V úrovni 0 nemá význam.

Čtvrtý údaj NP znamená *novou stránku* (New Page). Uplatní se při tisku na papír. Jestliže se změní skupina označená tímto údajem, první řádek nové skupiny se vytiskne až na další stránce.

Zápis L0 určuje koncovou úroveň (grand total či final total). Řádek s údajem L0 je nepovinný, ale je-li zadán, pak musí být uveden jako první. Jestliže řádek L0 není zadán, koncová úroveň se nezpracuje. Údaj *level* může být u dalších ("nižších") skupinových úrovní libovolný text, který však zpravidla bude pro názornost označovat číslo úrovně. Například následující tři řádky definic

```
--;L0;   .Celkem ****************; za případným středníkem jen komentář
--;L1;   .Závod =====&ZAVOD ==========;      ZAVOD
--;L2;   .Sklad -----&zavod --&sklad ------; SKLAD
```

určují dvě skupinové úrovně L2, L1 a koncovou úroveň L0. Hierarchie úrovní je dána pořadím zápisu definičních řádků, ne číslem u písmena L. Poslední řádek určuje nejnižší úroveň.

#### Definice sumarizovaných sloupců

Aby bylo co sumarizovat (střádat), musí být zadána jména sloupců se svým jedním nebo až pěti sumarizačními typy. K tomu je ovšem také nutné zadat nějaké skupinové úrovně (L0, L1, ...).

Zadání sumarizace spočívá v zápisu jednoho nebo více definičních řádků --;S. Definiční řádky nemusí být nijak zvlášť seřazeny. Každý definiční řádek začíná čtyřznakovým symbolem **`--;S`** v 1. pozici a pokračuje zápisy oddělenými středníkem. Zápisy určují jméno sloupce a způsob, jakým se budou jeho hodnoty střádat. Definiční řádek má tvar

```
--;S [střádaný-sloupec [; ind] [; ind] [; ind] [; ind] [; ind] [;]]
```

kde *ind* může být jeden ze symbolů – indikátorů S, A, M, m, C, které znamenají součet, průměr, maximum, minimum, počet. Indikátory nemusí být seřazeny *v žádném určitém pořadí*, nemusí být zadány všechny, a dokonce nemusí být zadány žádné. Sumární řádky ve výsledku dotazu se ovšem řadí v pořadí: S - součet, A - průměr, M - maximum, m - minimum, C - počet).

Je-li v některé pozici indikátor prázdný nebo chybný, má se za to, že není zadán.

Není-li v definičním řádku --;S zadán žádný indikátor nebo není-li definiční řádek --;S vůbec zadán, zpracování úrovní se neprovádí, i když je třeba zadáno definicemi --;L. Rovněž tak obráceně: Není-li zadán žádný definiční řádek --;L, zpracování úrovní se neprovádí, i když je třeba zadán definiční řádek --;S s nějakým indikátorem.

U sloupců typu DECIMAL, NUMERIC, INTEGER, SMALLINT a BIGINT lze zjišťovat součet, průměr, maximum, minimum a počet. U sloupců typu CHAR, VARCHAR, VARGRAPHIC, DATE, TIME a TIMESTAMP lze zjišťovat pouze maximum, minimum a počet.

*Hodnoty NULL* se do sumarizace nezahrnují.

Poznámka: Je-li chybně zadán součet (S) nebo průměr (A) pro *nenumerický* sloupec, pochopitelně se pro něj žádná hodnota netiskne. Řádek s úvodním textem (viz dále) se však vytiskne (i když třeba zároveň není sumarizován žádný *numerický* sloupec).

Například definice

```
--;S MNOZSTVI; C
--;S CELKEM ;  S ; A ; M ; m
```

určují dva střádané sloupce, přičemž u sloupce MNOZSTVI se vyhodnocuje jen počet řádků (C) a u sloupce CELKEM se pořizuje součet (S), průměr (A), maximum (M) a minimum (m).

### Úvodní texty pro sumární řádky

Standardní označení (zkratky) sumárních hodnot jsou SUM pro součet, AVG pro průměr (average), MAX pro maximum, MIN pro minimum a COUNT pro počet. Lze nahradit jinými pomocí definičního řádku --;s.

Definiční řádek začíná čtyřznakovým symbolem **`--;s`** (malé písmeno s) v 1. pozici. Má tvar

```
--;s [ [sum-text]; [avg-text]; [max-text]; [min-text]; [count-text] [;] ]
```

kde texty zapsané v daných pozicích nahradí standardní označení. Zde je ovšem pořadí zápisů podstatné. Jestliže některý z textů chybí, uplatní se v dané pozici standardní označení.

Například definiční řádek

```
--;s Součet ; Průměr ; Maximum ; Minimum ;  Počet
```

nahradí všechny standardní zkratky.

Příklad 1

V následujícím příkladu je ilustrována sumarizace *detailního* dotazu, tj. takového, kde není použito fráze GROUP BY. V titulních řádcích jsou zařazeny *proměnné* (dokonce jedna proměnná vícekrát). Dva sloupce budou *vynechány* z výstupu (ZAVOD, NAZ). Sumarizují se dva sloupce (MNOZSTVI, CELKEM) ve třech skupinových úrovních L0 (celkem), L1 (ZAVOD), L2 (SKLAD).

Text skriptu:

```
--Zboží podle skladů a závodů

--;t  >>> Závod 4x:    &ZAVOD , &ZAVOD , &ZAVOD , &ZAVOD .              <<<
--;t  >>> Název zboží: &NAZ ,  Sklad: &SKLAD . <<<
--;t

--;HZav;Skl;Č.zboží;Název zboží;Cena/j.;Množství;   Cena celkem;
--;H===;===;=======;===========;=======;========;   ===========;
--;H

--;T 1; 1;  ; 3

--;O ZAVOD
--;O NAZ;

--;S MNOZSTVI; C
--;S CELKEM ;  S ; A ; M ; m

--;s Součet ; Průměr ; Maximum ; Minimum ;  Počet

--;L0;   .Celkem ****************; za případným středníkem jen komentář
--;L1;   .Závod =====&ZAVOD ==========; ZAVOD
--;L2;   .Sklad -----&zavod --&sklad ------; SKLAD

--;P a4; fs12; P; 10; 10; 10; 10

SELECT
   S.ZAVOD,
   S.SKLAD,
   S.CZBOZI,
   SUBSTR(C.NAZZBO, 1, 15) AS NAZ,
   DEC(C.CENAJ, 12, 2) AS CJ,
   S.MNOZSTVI,
   DECIMAL ( C.CENAJ*S.MNOZSTVI, 12, 5 ) AS CELKEM
FROM STAVY AS S
   JOIN CENY AS C ON S.CZBOZI = C.CZBOZI
   ORDER BY S.ZAVOD, S.SKLAD, S.CZBOZI
```

Výsledek skriptu (zkrácený):

```
Zboží podle skladů a závodů

Pondělí, 23. listopadu 2015 13:43:20

  >>> Závod 4x:    01, 01, 01, 01.              <<<
  >>> Název zboží: PIŠKOTY OPAVIA ,  Sklad: 01. <<<

Skl   Č.zboží          Cena/j.      Množství      Cena celkem
===   =======          =======      ========      ===========

01    00001               8,99          1,00             8,99
01    00002             459,00          1,00           459,00
01    00010            6500,00          1,00          6500,00

   .Sklad -----01--01------
                     Součet                           6967,99
                     Průměr                           2322,66
                    Maximum                           6500,00
                    Minimum                              8,99
                      Počet                3
```

```
02    00003              1,25         1,00            1,25
02    00009            250,00         1,00          250,00
02    00010           6500,00         1,00         6500,00

   .Sklad -----01--02------
                   Součet                          6751,25
                   Průměr                          2250,42
                  Maximum                          6500,00
                  Minimum                             1,25
                    Počet               3

03    00003              1,25         1,00            1,25

   .Sklad -----01--03------
                   Součet                             1,25
                   Průměr                             1,25
                  Maximum                             1,25
                  Minimum                             1,25
                    Počet               1


   .Závod =====01==========
                   Součet                         13720,49
                   Průměr                          1960,07
                  Maximum                          6500,00
                  Minimum                             1,25
                    Počet               7


... atd.



   .Celkem ****************
                   Součet                         25311,59
                   Průměr                          1947,05
                  Maximum                          7000,00
                  Minimum                             1,25
                    Počet              14
```

### Příklad 2

V následujícím skriptu je nejdůležitější kód NP zapsaný u sloupce ZAVOD v definici L1. Způsobuje *skok na novou stránku*, jestliže se změní skupina jednoho závodu na skupinu jiného závodu. Kód NP lze zapsat ke každé úrovni kromě L0.

Text skriptu:

```
--Sumarizace zboží (CENY) podle skladů a závodů, závod na nové stránce

--;HZav;Skl;Č.zboží;Název zboží;Cena/j.;Množství;   Cena celkem;
--;H===;===;=======;===========;=======;========;   ===========;
--;H

--;T 1; 1; - ; 2; ZAVOD; SKLAD

--;S CELKEM  ; S ; C

--;s Součet ; Průměr ; Maximum ; Minimum ;  Počet

--;L0;   .CELKEM ZA PODNIK
--;L1;   .Závod  &ZAVOD ;         ZAVOD; NP
--;L2;   .Sklad  &ZAVOD  &SKLAD ; SKLAD

--;P a4; fs12; L; lm5; rm5; tm5; bm5

SELECT
   S.ZAVOD                                          -- Číslo závodu
   ,S.SKLAD                                         -- Číslo skladu
   ,S.CZBOZI                                        -- Číslo zboží
   ,SUBSTR(C.NAZZBO, 1, 15) AS NAZ                  -- Název zboží
   ,DEC(C.CENAJ, 12, 2) AS CJ                       -- Cena za jednotku
   ,S.MNOZSTVI                                      -- Množství jednotek
```

```
   ,DECIMAL ( C.CENAJ*S.MNOZSTVI, 12, 5 ) AS CELKEM -- Cena množství

FROM STAVY AS S
   JOIN CENY AS C ON S.CZBOZI = C.CZBOZI
ORDER BY S.ZAVOD, S.SKLAD, C.CZBOZI
```

Výsledek skriptu na tiskárně:

```
Sumarizace zboží (CENY) podle skladů a závodů, závod na nové stránce
Úterý, 15. prosince 2015 18:05:02
Zav  Skl  Č.zboží  Název zboží            Cena/j.     Množství     Cena celkem
===  ===  =======  ==========            =======     ========     ===========

01   01   00001    PIŠKOTY OPAVIA            8,99        1,00           8,99
          00002    Zubní pasta Kal         459,00        1,00         459,00
          00010    Koňak Gruzínský        6500,00        1,00        6500,00

     .Sklad  01 01
                                                   Součet            6967,99
                                                    Počet                  3

       02   00003    Prádelní šňůra            1,25        1,00           1,25
            00009    Whisky Balantin         250,00        1,00         250,00
            00010    Koňak Gruzínský        6500,00        1,00        6500,00

     .Sklad  01 02
                                                   Součet            6751,25
                                                    Počet                  3

       03   00003    Prádelní šňůra            1,25        1,00           1,25

     .Sklad  01 03
                                                   Součet               1,25
                                                    Počet                  1

     .Závod  01
                                                   Součet           13720,49
                                                    Počet                  7

Strana 1
```

```
Zav  Skl  Č.zboží  Název zbož
===  ===  =======  ==========

02   01   00005    Tričko bíl
          00006    Ponožky pá
          00008    Kalhoty dž

     .Sklad  02 01

       02   00009    Whisky Bal
            00011    Taška spor
            00014    Sako tvido
            00018    Husí sádlo

     .Sklad  02 02

     .Závod  02

     .CELKEM ZA PODNIK

Strana 2
```

## Práce s členy fyzických a logických souborů

Na příkladu ukážeme, jak lze pracovat s datovými členy (members) fyzických a logických souborů.

Předpokladem je, že soubory jsou vytvořeny pomocí DDS (Data Description Specifications). Vytvoření tabulek pomocí příkazů CREATE TABLE nebo CREATE VIEW totiž neumožňuje dodatečně přidat nebo odstranit datové členy.

### *Vytvoření fyzického a logického souboru se členy*

Vytvořím fyzický soubor CENY.

```
 **********************************************
 *   Soubor CENY - Ceník zboží
 **********************************************
A                                     UNIQUE
A          R CENYR
 *   Číslo zboží
A            CZBOZI         5A
 *   Cena za jednotku (kus)
A            CENAJ          9P 2
 *   Název zboží
A            NAZZBO        50A         CCSID(870)
 *   Definice klíče - Číslo zboží
A          K CZBOZI
```

Vytvořím logický soubor CENYL.

```
 **********************************************
 *   Soubor CENY
 *   Logický soubor
 **********************************************
A          R CENYR                     PFILE(CENY)
A            CZBOZI
A            CENAJ
A          K CZBOZI
```

V souboru CENY definuji maximální počet členů.

```
CHGPF      FILE(CENY) MAXMBRS(12)
```

Odstraním standardní člen stejného jména jako má soubor (CENY).

```
RMVM       FILE(VZTOOL/ ENY) MBR(CENY)
```

Přidám tři členy k souboru CENY.

```
ADDPFM     FILE(VZTOOL/CENY) MBR(CENY_01)
ADDPFM     FILE(VZTOOL/CENY) MBR(CENY_02)
ADDPFM     FILE(VZTOOL/CENY) MBR(CENY_03)
```

V logickém souboru CENYL definuji maximální počet členů.

```
CHGLF  FILE(CENYL) MAXMBRS(12)
```

K logickému souboru CENYL přidám také tři členy.

```
ADDLFM FILE(VZTOOL/CENYL) MBR(CENYL_01) DTAMBRS((VZTOOL/CENY (CENY_01)))
ADDLFM FILE(VZTOOL/CENYL) MBR(CENYL_02) DTAMBRS((VZTOOL/CENY (CENY_02)))
ADDLFM FILE(VZTOOL/CENYL) MBR(CENYL_03) DTAMBRS((VZTOOL/CENY (CENY_03)))
```

### *Skript k vytvoření alias objektů a zápisu dat do členů*

Následující skript CENY_ALIASES smaže alias objekty (jestliže exixtují) a vytvoří nové, pro každý člen fyzického i logického souboru pojmenované jako členy.

```
/* Předem odstraním objekty ALIAS pro členy fyzického a logického souboru */
DROP ALIAS VZTOOL.CENY_01 ;
DROP ALIAS VZTOOL.CENY_02 ;
DROP ALIAS VZTOOL.CENY_03 ;
DROP ALIAS VZTOOL.CENYL_01 ;
DROP ALIAS VZTOOL.CENYL_02 ;
DROP ALIAS VZTOOL.CENYL_03 ;

/* Vytvořím objekty ALIAS pro členy souboru CENY */
CREATE ALIAS VZTOOL.CENY_01 FOR VZTOOL.CENY(CENY_01) ;
CREATE ALIAS VZTOOL.CENY_02 FOR VZTOOL.CENY(CENY_02) ;
CREATE ALIAS VZTOOL.CENY_03 FOR VZTOOL.CENY(CENY_03) ;

/* Vytvořím objekty ALIAS pro členy logického souboru CENYL */
CREATE ALIAS VZTOOL.CENYL_01 FOR VZTOOL.CENY(CENY_01) ;
CREATE ALIAS VZTOOL.CENYL_02 FOR VZTOOL.CENY(CENY_02) ;
CREATE ALIAS VZTOOL.CENYL_03 FOR VZTOOL.CENY(CENY_03) ;

/* Vrátím předvolené schema */
SET SCHEMA DEFAULT;

/* Vymažu záznamy ze členů souboru CENY */
DELETE FROM CENY_01 ;
DELETE FROM CENY_02 ;
DELETE FROM CENY_03 ;
```

```
/* Vložím záznamy do členů souboru CENY */
INSERT INTO CENY_01 values ('00001', 8,99, 'PIŠKOTY OPAVIA') ;
INSERT INTO CENY_01 values ('00002', 459,00, 'Zubní pasta Kalodont') ;
INSERT INTO CENY_01 values ('00003', 1,25, 'Prádelní šňůra') ;

INSERT INTO CENY_02 values ('00004', 10,50, 'Ponožky pánské tmavé') ;
INSERT INTO CENY_02 values ('00005', 120,00, 'Tričko bílé') ;
INSERT INTO CENY_02 values ('00006', 10,55, 'Ponožky pánské bílé, nové')
```

### *Kontrolní výpis dat ze členů logického souboru*

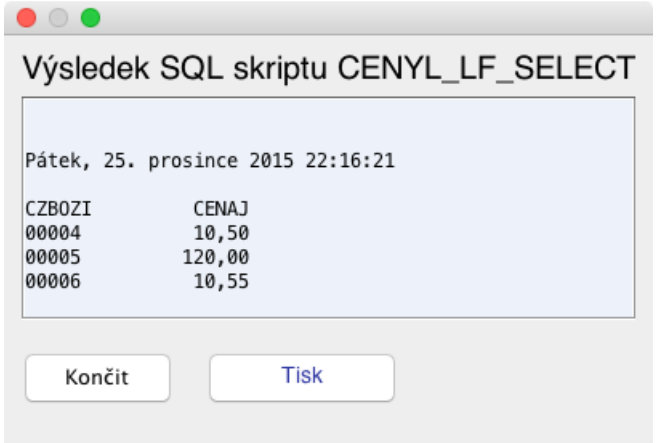
Nyní provedeme dotazy na logické členy.

```
--Výpis z logického souboru CENYL se členy

SELECT * FROM CENYL_01;
SELECT * FROM CENYL_02;
SELECT * FROM CENYL_03;
SELECT * FROM CENYL
```

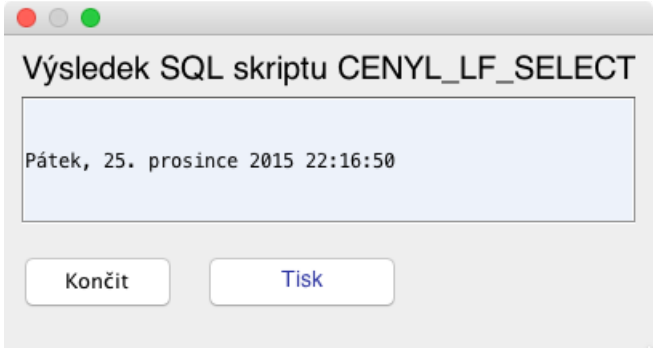
Dostaneme čtyři obrazovky s obsahem logických členů.

Člen CENYL_01:

Člen CENYL_02:




Člen CENYL_03 (prázdný):




Všechny členy dohromady (všimněme si, že standardní člen CENYL nebyl odstraněn):