

Databáze v geoinformatice IX.

Mgr. Michal Jakl

ZS 2022

Obsah lekce

1. SQL - doplnění
2. PgAdmin – export a import
3. Procvičování SQL - interpretace

Praktické cvičení:

Viz výše

Vyhodnocení úkolu z minula

PostgreSQL Exercises

Home

Getting Started

Exercises ▾

About

Options

PostgreSQL Exercises

Welcome to PostgreSQL Exercises! This site was born when I noticed that there's a load of material out there to help people learn about SQL, but not a great deal to make it easy to learn by doing. PGExercises provides a series of questions and explanations built on a single, simple dataset. It's designed for use as a partner to a good book or Postgres' excellent [documentation](#).

The exercises on this site range from simple select and where clauses, through joins and case statements, and on to aggregations, window functions, and recursive queries. Most people who aren't already pros should find something to test themselves with.

For an introduction to the dataset, go to [Getting Started](#), then select an exercise category from the menu and go!

Site content licensed under [CC BY-SA 3.0](#)

1. SQL – podmínky

CASE - WHEN – THEN – ELSE - END

1. Začínáme příkazem CASE
2. Následuje alespoň jedna dvojice WHEN/THEN příkazů (WHEN *podmínka* THEN *hodnota*);
3. Příkaz ELSE je volitelný a poskytuje způsob, jak podchytit hodnoty, které nejsou uvedeny v části WHEN/THEN
4. Příkaz END musí být na konci a ukončujeme tím výraz;
 1. Pokud není ELSE definováno je pro varianty, které nejsou uvedeny ve WHEN/THEN predikátech, je vrácena hodnota NULL

Syntaxe Simple SQL CASE formy:

```
SELECT
  CASE Sloupec
    WHEN 0 THEN 'No'
    WHEN 1 THEN 'Yes'
    ELSE 'N/A'
  END AS Vysledky
FROM dbo.Tabulka;
```

Syntaxe Searched SQL CASE formy:

```
SELECT
  CASE
    WHEN ZnamkaNaVysvedceni BETWEEN 3 AND 5 OR (Predmet='Chovani' AND ZnamkaNaVysvedceni <>1) THEN 'Špatná známka'
    WHEN ZnamkaNaVysvedceni >= 1 THEN 'Dobrá známka'
    ELSE 'Neplatná hodnota. Znamky na vysvědčení můžou být v rozsahu 1-5'
  END AS Vysledky
FROM dbo.Vysvedceni;
```

Kde se dá použít?

- SELECT – podmíněnou logikou vrátíte nadefinované hodnoty
- WHERE – lze použít i k definování podmínky
- Agregace – výsledek můžete zabalit do agregačních funkcí
- GROUP BY – pokud definujete nějakou agregaci přes pole definované přes CASE, můžete umístit samotný case do GROUP BY
- ORDER – pomocí case lze seřadit výsledky

1. SQL – aktuální datum a čas

```
SELECT
```

```
current_timestamp::date AS my_date,  
current_timestamp::time AS my_time;
```



	my_date date	my_time time without time zone
1	2021-04-14	13:33:04.70808

Current year: `SELECT date_part('year', (SELECT current_timestamp));`

Current month: `SELECT date_part('month', (SELECT current_timestamp));`

Current day: `SELECT date_part('day', (SELECT current_timestamp));`

Current hour: `SELECT date_part('hour', (SELECT current_timestamp));`

Current minute: `SELECT date_part('minute', (SELECT current_timestamp));`

Current second: `SELECT date_part('second', (SELECT current_timestamp));`

1. SQL – časté chyby

1. SQL Skript pouštíme nad špatnou databází – Invalid object name

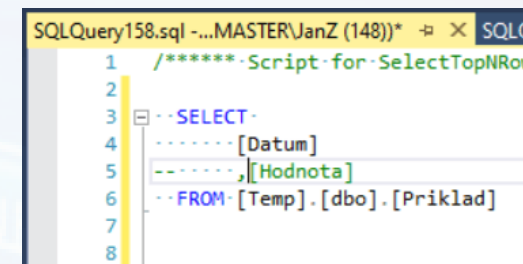
2. Překlepy v SQL klauzulích nebo názvech objektů – Invalid object name/syntax

3. Čárka navíc – Incorrect syntax near the keyword

1. Možná jste si všimli, že jednotlivé sloupce vždy odděluji čárkami na dalším řádku. Má to svůj důvod a jedná se o správný návyk při psaní kódu. SQL skripty se často musí ladit nebo v budoucnu revidovat/opravovat – často některé sloupce chceme zakomentovat. Pokud budeme jednotlivé sloupce oddělovat čárkou tímto způsobem a následně budeme chtít sloupec zakomentovat, tak budeme muset vložit komentovací znak 2x (před čárku a před sloupec na dalším řádku). To je u velkých skriptů otravné a způsobuje to riziko chyb.

4. Chyby v logických operátorech AND a OR – (ne)používání závorek

5. Nepoužívání aliasů při joinech vede k chybě Ambiguous column name



```
SQLQuery158.sql -...MASTER\JanZ (148)* - X SQLC
1  /***** Script for SelectTopNRo
2
3  --SELECT
4  .....[Datum]
5  .....,[Hodnota]
6  --FROM [Temp].[dbo].[Příklad]
7
8
```

2. PgAdmin – export a import

The image illustrates the workflow for exporting and importing a PostgreSQL database using PgAdmin. It consists of several interconnected screenshots:

- Initial State:** The PgAdmin interface shows a tree view of Schemas (6) under a database named 'postgis_dtb'. The 'pcr' schema is selected, and a context menu is open with 'Backup...' highlighted.
- Backup Dialog:** The 'Backup (Schema: pcr)' dialog is shown. The 'General' tab is active, with the following settings:
 - Filename: C:\Users\GISLukas\Desktop\pcr.backup
 - Format: Tar
 - Compression ratio: (empty)
 - Encoding: Select an item...
 - Number of jobs: (empty)
 - Role name: Select an item...The 'Backup' button is highlighted.
- File System:** A Windows File Explorer window shows the 'pcr.backup' file saved in the 'DATABAZE' folder on the 'Můj disk'.
- Restore Dialog:** The 'Restore (Database: postgis_dtb)' dialog is shown. The 'General' tab is active, with the following settings:
 - Format: Custom or tar
 - Filename: C:\Users\GISLukas\Desktop\pcr.backup
 - Number of jobs: (empty)
 - Role name: Select an item...The 'Restore' button is highlighted.
- Final State:** The PgAdmin interface shows the 'pcr' schema selected, and a context menu is open with 'Refresh...' highlighted, indicating the database has been restored.

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_AsText(ST_Transform(ST_GeomFromText('POLYGON((743238  
2967416,743238 2967450,743265 2967450,743265.625 2967416,743238  
2967416))',2249),4326)) As wgs_geom;
```


3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_AsText(ST_Transform(ST_GeomFromText('POLYGON((743238  
2967416,743238 2967450,743265 2967450,743265.625 2967416,743238  
2967416))',2249),4326)) As wgs_geom;
```

Vypíše jako text geometrii zadaného polygonu v souřadném systému WGS-84

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_Intersects('POINT(0 0)::geometry, 'LINESTRING ( 0 0, 0 2 )::geometry);
```

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_Intersects('POINT(0 0)::geometry, 'LINESTRING ( 0 0, 0 2 )::geometry);
```

Vrátí „true“, pokud se daný bod a linie prostorově překrývají.

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
CREATE INDEX idx_geo_buildings ON tbl_buildings USING GIST ( geom );
```

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
CREATE INDEX idx_geo_buildings ON tbl_buildings USING GIST ( geom );
```

Vytvoří prostorový index nad sloupcem geom v tabulce tbl_buildings.

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_Distance(  
    ST_Transform('SRID=4326;POINT(-72.1235 42.3521) '::geometry, 3857),  
    ST_Transform('SRID=4326;LINESTRING(-72.1260 42.45, -72.123 42.1546) '::geometry, 3857)  
    ) * cosd(42.3521);  
st_distance
```

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_Distance(  
    ST_Transform('SRID=4326;POINT(-72.1235 42.3521) '::geometry, 3857),  
    ST_Transform('SRID=4326;LINESTRING(-72.1260 42.45, -72.123 42.1546) '::geometry, 3857)  
    ) * cosd(42.3521);  
st_distance
```

Vrátí konstantou vynásobenou vzdálenost v jednotkách souř. systému s EPSG 3857 pro dané geometrie zadané souřadnicemi WGS-84.

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_Buffer(  
  ST_GeomFromText(  
    'LINESTRING(50 50,150 150,150 50)'  
  ), 10, 'endcap=round join=round');
```


3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT ST_Buffer(  
  ST_GeomFromText(  
    'LINESTRING(50 50,150 150,150 50)'  
  ), 10, 'endcap=round join=round');
```

Vrátí obalovou zónu o šířce 10 jednotek se zaoblenými konci i lomovými body pro zadanou linii.

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT nazev, geom as spojnice FROM "OKRESY" WHERE
  ST_Intersects(
    ST_MakeLine(
      (SELECT geom FROM "BODY" WHERE "Name" LIKE 'Brno'), (SELECT geom FROM
        "BODY" WHERE "Name" LIKE 'Děčín'))
    ,ST_Transform(geom,4326));
```

3. Procvičování SQL - interpretace

Přeložte „do lidské řeči“ následující SQL skript = vysvětlete svými slovy, jaký úkon/úkony SŘBD při jeho interpretování provede:

```
SELECT nazev, geom as spojnice FROM "OKRESY" WHERE
  ST_Intersects(
    ST_MakeLine(
      (SELECT geom FROM "BODY" WHERE "Name" LIKE 'Brno'), (SELECT geom FROM
        "BODY" WHERE "Name" LIKE 'Děčín'))
    ,ST_Transform(geom,4326));
```

Vybere všechny okresy, které protíná spojnice měst Brno a Děčín.

Děkuji za pozornost a těším se příště!

Prostor pro zpětnou vazbu

mail@mjakl.cz

Tel. 608 544 839

Web: mjakl.cz

Telegram: t.me/mjakl_cz