

BI-BIG - semestrální práce

# Výsledky voleb 2017 a demografické informace

<b>Úvod</b>	<b>2</b>
<b>Práce</b>	<b>3</b>
Popis databáze	3
Popis a ukázka použitých datasetů	3
Technologie	6
Využití Apache Hive	6
Spustění	6
Import dat	6
Dotazy	8
Export dat	9
Využití Elasticsearch a Kibany	10
<b>Závěr</b>	<b>10</b>

# 1. Úvod

Cílem této práce je předvést ukázkou práce s vybranými databázovými systémy, které jsou navrženy pro distribuované ukládání dat a jejich paralelní zpracování, tak aby bylo možné pracovat s velkými objemy dat. Práce bude předvedena na mnou 7 vybraných datových sadách, které byly staženy z otevřených dat poskytovaných Českým statistickým úřadem a souřadnice obcí z GitHubu z publikované datové sady. Některá data byla předzpracovaná např. upraveno kódování pro univerzální použití mezi systémy. Pro uložení, dotazování, agregaci a shrnutí dat jsem si vybral nástroj Apache Hive. Pro indexaci, analýzu a vizualizaci jsem zvolil Elasticsearch a Kibanu. V těchto nástrojích zpracuji vybrané datové sady o výsledcích voleb a demografických údajích na území České republiky. Mezi datovými sadami jsou zahrnuty také číselníky pro celé názvy místo identifikačních čísel. Veškeré ukázky jsou prováděné na Linuxu Debian verze 18. Projekt je dostupný na adrese [https://gitlab.fit.cvut.cz/martilad/BIG\\_semestral\\_work.git](https://gitlab.fit.cvut.cz/martilad/BIG_semestral_work.git) na školním GitLabu. Podle dokumentace je možné sestavit celý experiment.

## 2.Práce

### Popis databáze

K práci jsem si vybral především datové sady poskytované Českým statistickým úřadem. <https://data.gov.cz/opensource/cz/>. Dále také datovou sadu z <http://33bcdd.github.io/souradnice-mest/>. Konkrétně se jedná o data výsledku voleb do poslanecké sněmovny v roce 2017 pro jednotlivé politické strany. K této datové sadě je aktuální číselník vylosovaných čísel politických stran v těchto volbách a obecné číslo přiřazené každé politické straně. Dalším číselníkem jsou názvy politických stran na obecné číslo přiřazené politické straně při vzniku. Další vybranou datovou sadou jsou data o populaci. Bohužel jsem byl nucen využít mírně zastaralá data ze sčítání lidu v roce 2011. Nad těmito agregovanými datovými sadami jsem vytvářel vizualizace v Kibaně. Další datovou sadou je datová sada s demografickými daty o populaci v roce 2016. V této sadě jsou ukazatele o pohybu, přírůstku a úbytku obyvatelstva v jednotlivých obcích. V této datové sadě lze třeba sledovat procento přistěhovalců a přirozeného přírůstku v jednotlivých obcích. Poslední datovou sadou je datová sada se souřadnicemi jednotlivých obcí v České republice.

### Popis a ukázka použitých datasetů

Datové sady jsou obsaženy v repozitáři který lze stáhnout:

```
git clone git@gitlab.fit.cvut.cz:martilad/BIG_semestral_work.git
```

Datové soubory budou dostupné v BIG\_semestral\_work/data

Všechny datové sady jsou ve formátu CSV a kódování utf-8.

#### 1. ciselnik\_pol\_stran.csv.

Datová sada obsahuje názvy politických stran k jejich kódům.

Sloupce popořadě:

VSTRANA - kód volební strany

NAZEVCELK - název volební strany

NAZEV\_STRV - název strany 50 znaku

ZKRATKAV30 - zkratka 30 znaku

ZKRATKAV8 - zkratka 8 znaku

POCSTR\_SLO - počet politických subjektů ve složení volební strany

SLOZENI - složení strany kód

ZKRATKA\_OF - oficiální zkratka strany

TYPVS - typ volební strany S - politická strana, K - koalice, N - nezávislí, D - združení

**Ukázka:**

```
1,Křesťanská a demokratická unie - Československá strana lidová,Křesťan.a  
demokrat.unie-Českosl.strana lidová,Křesť.demokr.unie-Čs.str.lid.,KDU-ČSL,1,001,KDU-ČSL,S
```

## 2. demograficka\_data.csv

Datová sada obsahuje demografická data k uvedeným územním částem.

Sloupce popořadě:

IDHOD - unikátní id databáze csu

HODNOTA - zjištěná hodnota

VUK - kód výstupního ukazatele veřejné databáze

VUK\_TEXT - název výstupního ukazatele

STAPRO\_KOD - kód statistické proměnné ze systému SMS UKAZ

VUZEMI\_CIS - kód číselníku pro referenční území, typologie území

VUZEMI\_KOD - kód položku pro referenční území

ROK - referenční období

CASREF\_OD - začátek intervalu

CASREF\_DO - konec intervalu

VUZEMI\_TXT - text z položky pro referenční území

**Ukázka:**

758382045,13,DEM0007,Živě narození,4355,43,500011,2016,2016-01-01,2016-12-31,Želechovice nad Dřevnicí

## 3. kstrana\_vstrana\_2017.csv

Datová sada obsahuje číselník vylosovaných čísel volebních stran na jejich kódy.

Sloupce popořadě:

KSTRANA - vylosované číslo strany pro volby 2017

VSTRANA - přidělené číslo při vzniku strany

NAZEVCELK - název

**Ukázka:**

1,53,Občanská demokratická strana

## 4. Scitani\_lidu\_2011.csv

Datová sada obsahuje údaje nasbírané během sčítání lidu v roce 2011.

Sloupce popořadě:

TYP\_UZ\_NAZ - typ území, stát, kraj, okres, obec

NAZEV - název území

UZYCIS - kód území

UZYKOD - kód územního číselníku

U01 - obyvatelstvo celkem

U02 - obyvatelstvo muži

U03 - obyvatelstvo ženy

U04 - lidi věk 0-14

U05 - lidi věk 15-64

U06 - lidé 65 a více

U07 - ekonomicky aktivní

U08 - ekonomicky aktivní zaměstnaní

U09 - obydlené domy

U10 - obydlené byty

U11 - hospodařící domácnosti

**Ukázka:**

okres, Zlín, 101, 40851, 190488.0000, 92475.0000, 98013.0000, 26819.0000, 131235.0000, 31945.0000, 94172.0000, 85673.0000, 36519.0000, 72773.0000, 76823.0000

### 5. Souradnice.csv

Datová sada se souřadnice jednotlivých obcí.

Sloupce popořadě:

Obec, Kód obce, Okres, Kód okresu, Kraj, Kód kraje, PSČ, Latitude, Longitude

**Ukázka:**

Abertamy, 554979, Karlovy Vary, CZ0412, Karlovarský kraj, CZ041, 36235, 50.368855, 12.818377

### 6. Volby\_posl\_2017.csv

Datová sada s výsledku pro každou politickou stranu pro každý územní celek v České republice.

Sloupce popořadě:

ID\_OKRSKY - id okrsku

TYP\_FORM - data 1 - t/4, 2 - příloha t/4, 3 - ts

OPRAVA - příznak opravy 1 - ano, 0 - ne

CHYBA - příznak chyby 1 - chyba, 0 - ne

OKRES - kód okresu

OBEC - kód obce

OKRSEK - kód okrsku

KC\_1 - kontrolní číslo 1

KSTRANA - vylosované číslo strany

POC\_HLASU - počet hlasů dané strany

KC\_2 - kontrolní číslo 2

HLASY\_01 - krožkování pro kandidata 1

...

HLASY\_36 - krožkování pro kandidáta 36

KC\_3 - kontrolní číslo 3

KC\_4 - kontrolní číslo 4

POSL\_KAND - poslední číslo kandidáta co měl přednostní hlasy

KC\_SUM - sumární kontrolní číslo t4

**Ukázka:**

1, 2, 0, 0, 2103, 532088, 1, 0, 1, 21, 22, 0, 0, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 3, 0, 1, 0, 0, 0, 0, 238, 260, 31, 534745

### 7. Volby\_scitani.csv

Datová sada vytvořena spojením těchto datasetů pro import, idenxování a vizualizaci pro technologii elasticsearch a kibanu. Datová sada je vytvořena podle příkazu v Kapitole Využití Apache Hive.

# Technologie

## Využití Apache Hive

### Spustění

Pro spuštění je důležité si nainstalovat docker.

Debian: <https://docs.docker.com/engine/installation/linux/docker-ce/debian/>

Dále také docker-compose.

```
curl -L https://github.com/docker/compose/releases/download/1.16.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

Přidělit práva.

```
chmod +x /usr/local/bin/docker-compose
```

Dále ve složce naklonované z GitLabu je složka Hive. V této složce rozbalce obsažený adresář a provedeme následující příkazy, kterými nám docker-compose sestaví požadované kontejnery a ty následně spustíme. (Při končení lze kontejnery postupně bezpečně zastavovat v opačném pořadí docker stop <k1> <k2> pro opětovné spouštění)

Příkazy:

```
cd <rozbalený adresář>
docker-compose build
docker-compose up -d namenode hive-metastore-postgresql
docker-compose up -d datanode hive-metastore
docker-compose up -d hive-server
```

Tímto máme spuštěný hive-server lokálně v docker kontejnerech.

Následujícími příkazy spustíme konzoli serveru.

```
docker exec -it hive-server bash
/opt/hive/bin/beeline -u jdbc:hive2://localhost:10000
```

### Import dat

Nyní je třeba do kontejneru naimportovat data.

```
docker cp <soubor v počítači> <kontejner>:<umístění v kontejneru>
docker cp data/ciselnik_pol_stran.csv hive-server:/data/ciselnik_pol_stran.csv
```

Tento příkaz je potřeba provést pro všechny datové sady.

Dále je nutné pro datové sady vytvořit tabuly.

```
CREATE TABLE ciselnik_pol_stran(  
VSTRANA int,  
NAZEVCELK string,  
NAZEV_STRV string,  
ZKRATKAV30 string,  
ZKRATKAV8 string,  
POCSTR_SLO string,  
SLOZENI string,  
ZKRATKA_OF string,  
TYPVS string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
CREATE TABLE kstrana_vstrana_2017(  
KSTRANA int,  
VSTRANA int,  
NAZEVCELK string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
CREATE TABLE souradnice(  
OBEC string,  
KODOBCE int,  
OKRES string,  
KODOKRESU string,  
KRAJ string,  
KODKRAJE string,  
PSC string,  
LATITUDE double,  
LONGITUDE double ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
CREATE TABLE demograficka_data(  
IDHOD int,  
HODNOTA double,  
VUK string,  
VUK_TEXT string,  
STAPRO_KOD int,  
VUZEMI_CIS int,  
VUZEMI_KOD int,  
ROK int,  
CASREF_OD string,  
CASRED_DO string,  
VUZEMI_TXT string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
CREATE TABLE scitani_lidu_2011(  
TYPUZ_NAZ string,  
NAZEV string,  
UZCIS int, UZKOD int,  
U01 double, U02 double, U03 double,  
U04 double, U05 double, U06 double,  
U07 double, U08 double, U09 double,  
U10 double, U11 double) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```



Poslední tabulku nechám kvůli rozměrnosti bez formátování.

```
CREATE TABLE volby_posl_2017(ID_OKRSKY int,TYP_FORM int,OPRAVA int,CHYBA int,OKRES int,OBEC
int,OKRSEK int,KC_1 int,KSTRANA int,POC_HLASU int,KC_2 int,HLASY_01 int,HLASY_02 int,HLASY_03
int,HLASY_04 int,HLASY_05 int,HLASY_06 int,HLASY_07 int,HLASY_08 int,HLASY_09 int,HLASY_10
int,HLASY_11 int,HLASY_12 int,HLASY_13 int,HLASY_14 int,HLASY_15 int,HLASY_16 int,HLASY_17
int,HLASY_18 int,HLASY_19 int,HLASY_20 int,HLASY_21 int,HLASY_22 int,HLASY_23 int,HLASY_24
int,HLASY_25 int,HLASY_26 int,HLASY_27 int,HLASY_28 int,HLASY_29 int,HLASY_30 int,HLASY_31
int,HLASY_32 int,HLASY_33 int,HLASY_34 int,HLASY_35 int,HLASY_36 int,KC_3 int,KC_4
int,POSL_KAND int,KC_SUM int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

A následně naimportovat data:

```
LOAD DATA LOCAL INPATH '/data/ciselnik_pol_stran.csv' overwrite into table ciselnik_pol_stran;
LOAD DATA LOCAL INPATH '/data/kstrana_vstrana_2017.csv' overwrite into table
kstrana_vstrana_2017;
LOAD DATA LOCAL INPATH '/data/souradnice.csv' overwrite into table souradnice;
LOAD DATA LOCAL INPATH '/data/demograficka_data.csv' overwrite into table demographic_data;
LOAD DATA LOCAL INPATH '/data/scitani_lidu_2011.csv' overwrite into table scitani_lidu_2011;
LOAD DATA LOCAL INPATH '/data/volby_posl_2017.csv' overwrite into table volby_posl_2017;
```

Tímto máme vytvořené datové struktury v Apache Hive.

## Dotazy

První dotaz agreguje data z datové sady demografických údajů. Počítá průměrnou hodnotu těchto ukazatelů přes všechny obce v České republice. Data dává do vytvořené tabulky.

```
CREATE TABLE prum_dem_ukaz AS
SELECT vuk_text demographic_data,
       AVG(hodnota) AS prum_obec
FROM demographic_data
WHERE vuzemi_cis = 43
GROUP BY vuk_text;
```

Druhý dotaz vytváří tabulku s celkovým přírůstkem obyvatel v České republice za rok 2016 z datové sady demografických údajů.

```
CREATE TABLE celkovy_prirustek_2016 AS
SELECT vuk_text demographic_data,
       SUM(hodnota) AS prum_obec
FROM demographic_data
WHERE vuzemi_cis = 43
      AND vuk = 'DEM0012'
GROUP BY vuk_text;
```

Třetí dotaz vytváří tabulku s obcemi a procentem žen v těchto obcích z datové sady sčítání lidu.

```
CREATE TABLE procento_zen_obce AS
SELECT nazev,
       (u03/u01)*100 AS zeny
FROM scitani_lidu_2011
WHERE typuz_naz = 'obec';
```

Čtvrtý dotaz agreguje data z voleb do poslanecké sněmovny 2017 a sčítání lidu 2011. Počítá průměrné procentuální využití preferenčních hlasů a procento zastoupení daných kategorií lidí v populaci.

```
CREATE TABLE per_vyuz_prefer AS
SELECT nazev,
       u06/u01 AS duchodci,
       u07/u01 AS ve_veku,
       u04/u01 AS deti,
       AVG((HLASY_01 + HLASY_02 + HLASY_03 + HLASY_04 + HLASY_05 + HLASY_06 + HLASY_07 +
HLASY_08 + HLASY_09 + HLASY_10 + HLASY_11 + HLASY_12 + HLASY_13 + HLASY_14 + HLASY_15 +
HLASY_16 + HLASY_17 + HLASY_18 + HLASY_19 + HLASY_20 + HLASY_21 + HLASY_22 + HLASY_23 +
HLASY_24 + HLASY_25 + HLASY_26 + HLASY_27 + HLASY_28 + HLASY_29 + HLASY_30 + HLASY_31 +
HLASY_32 + HLASY_33 + HLASY_34 + HLASY_35 + HLASY_36)/4/poc_hlasu) * 100 AS per_pref
FROM scitani_lidu_2011
INNER JOIN volby_posl_2017 ON uzkod = obec
GROUP BY nazev,
         u01,
         u06,
         u07,
         u04;
```

Pátý dotaz připravuje data pro index a vizualizaci. Spojuje 5 datových sad a vybírá z nich chtěné údaje. Jednotlivé datové sady a atributy jsou vidět v příkazu níže.

```
CREATE TABLE volby_scitani AS
SELECT a.vstrana AS kod_strany,
       a.nazevcelk AS strana,
       a.zkratka_of AS zkrat_strana,
       c.poc_hlasu AS hlasu,
       d.obec AS obec,
       d.kodobce AS kod_obce,
       d.latitude AS latitude,
       d.longitude AS longitude,
       d.kraj AS kraj,
       d.okres AS okres,
       e.u01 AS obyvatelstvo,
       e.u02 AS muzi,
       e.u03 AS zeny,
       e.u04 AS deti,
       e.u05 AS stredni,
       e.u06 AS duchodci,
       e.u07 AS ek_aktiv,
       e.u08 AS zamestnani
FROM ciselnik_pol_stran AS a
INNER JOIN kstrana_vstrana_2017 AS b ON a.vstrana = b.vstrana
INNER JOIN volby_posl_2017 AS c ON c.kstrana = b.kstrana
INNER JOIN souradnice AS d ON d.kodobce = c.obec
INNER JOIN scitani_lidu_2011 AS e ON e.uzkod = d.kodobce;
```

## Export dat

Poslední vytvořenou datovou sadu jsem si vyexportoval do formátu CSV pro export do elasticsearch. Ukázka exportu.

```
insert overwrite local directory '/data' row format delimited fields terminated by ',' select  
* from volby_scitani;  
docker cp hive-server:/data/<name_of_gnerated_file>  
<path_to_BIG_folder/BIG_semestral_work/data>  
mv <name_of_gnerated_file> volby_scitani.csv
```

## Využití Elasticsearch a Kibany

### 3. Závěr