PTS Vaja 2

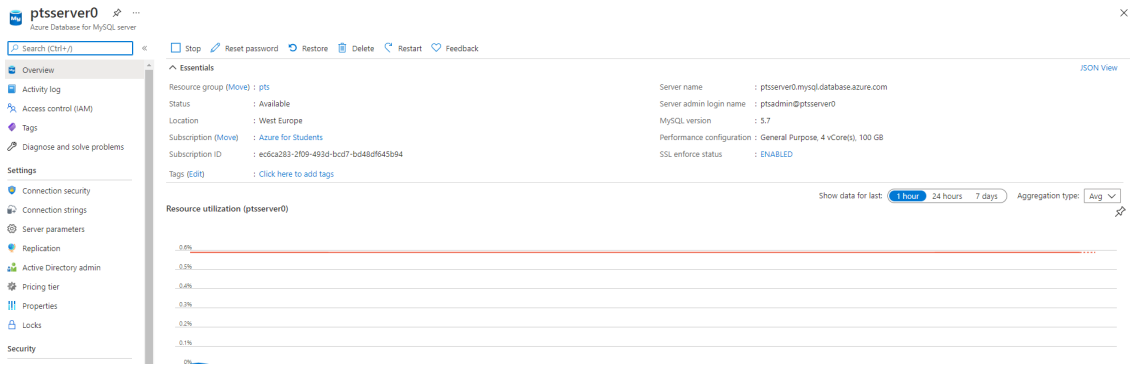
Za vajo 2 bomo uporabili naslednje tehnologije in orodja:

- Microsoft Azure + Azure Database for MySQL
- MySQL Workbench 8.0.20 CE
- Docker orodja (`https://docs.docker.com/get-docker/`)
- Hadoop 3.2.1
- Sqoop 1.4.7

1. Podatkovne baze v oblaku

Podatkovne baze v oblaku prinašajo veliko priložnosti in prednosti za uporabnike, kot npr. ni potrebe po fizičnem upravljanju strežnika, visoka dostopnost, razširljivost itd. MS Azure Cloud je platforma, ki ponuja implementacijo številnih SUPB-jev (npr. Sql Server, MySQL, PostgreSQL) kot storitev v oblaku (t.i. model podatkovne baze kot storitve "database-as-a-service"). Na vajah bomo uporabljali študentsko licenco za delno brezplačni dostop do Azure storitev. Najprej je potrebno aktivirati vašo študentsko licenco na `https://azure.microsoft.com/en-us/free/students/`. Prijavite se z vašim "student.um.si" naslovom.

Po uspešni aktivaciji se je potrebno registrirati/prijaviti v Azure portal, ki je dostopen na `https://portal.azure.com/`, kjer bomo vzpostavili strežnik za podatkovno bazo MySQL. Za prijavo ponovno uporabite vaš študentski email naslov (student.um.si). Ko se prijavite, ustvarite novi vir (angl. resource). V kategoriji "Databases" izberite "Azure Database for MySQL". Vnesite podatke in počakajte par minut da se ustvari vir z MySQL strežnikom.



Slika 1: Ustvarjena instanca strežnika MySQL v Azure-u.

Ko bo vir pripravljen, potrebno je dodatno namestiti oddaljeno povezovanje s strežnikom, ker je strežnik po privzetih namestitvah zavarovan s požarnim zidom (angl. firewall) in ni javno dostopen. V meniju na levi strani izberite opcijo "Connection security" in kot novo pravilo za požarni zid dodajte svoj trenutni IP-naslov ali nastavite dostop na javno. Shranite spremembo.

Zdaj se lahko povežete na MySQL strežnik v oblaku znotraj MySQL Workbench-a. V Azure portalu za vaš vir v meniju na levi strani izberete opcijo "Connection strings". Prikaže se seznam z različnimi znakovnimi nizi za povezovanje s strežnikom, kjer so vidni podatki potrebni za povezavo (*hostname, port, username*. Npr. za JDBC povezavo bo niz za povezavo v obliki:

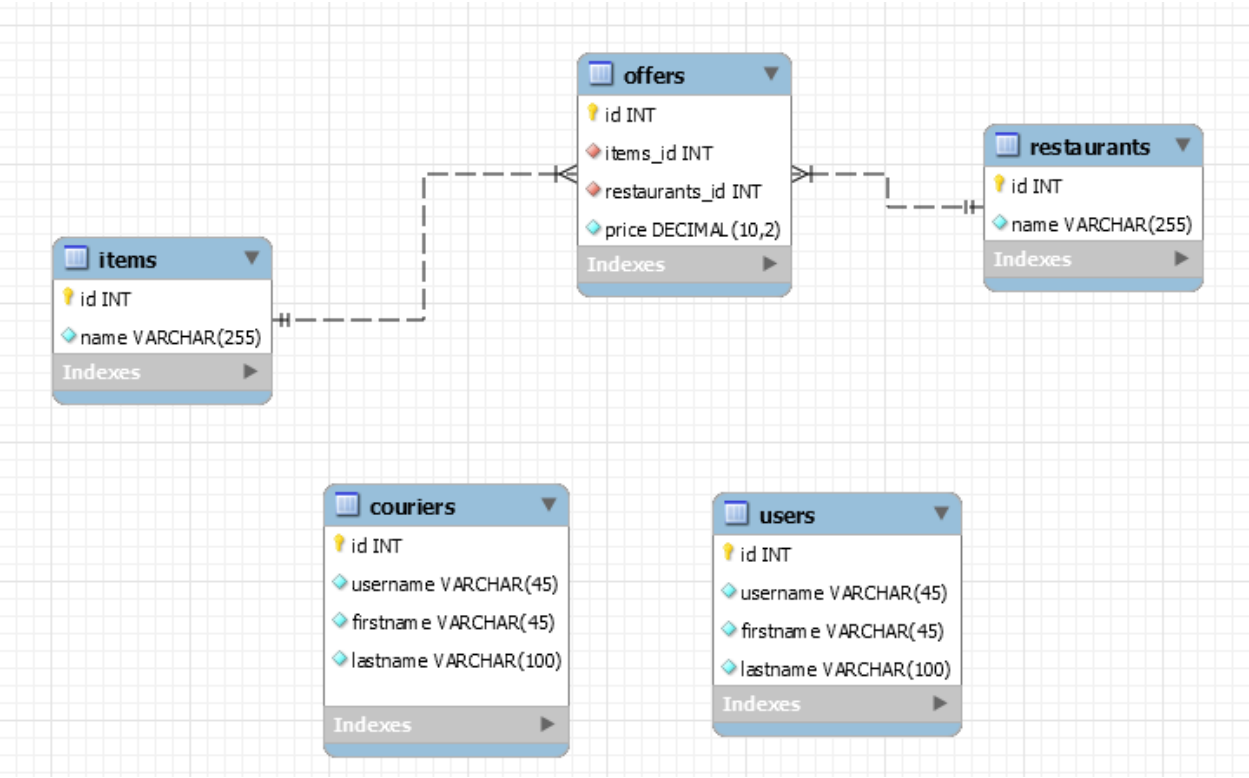
```
String url ="jdbc:mysql://ptsserver0.mysql.database.azure.com:3306/{your_database}?
useSSL=true&requireSSL=false"; myDbConn = DriverManager.getConnection(url,
"ptsadmin@ptsserver0", {your_password});
```

Podatke vpišete v Workbench pri ustvarjanju nove povezave (hostname="ptsserver0.mysql.database.azure.com", username="ptsadmin@ptsserver0", password=your_password). Ko se uspešno povežete, boste v seznamu shem videli le "sys" shemo.

Uvoz podatkov iz CSV dokumenta v MySQL bazo

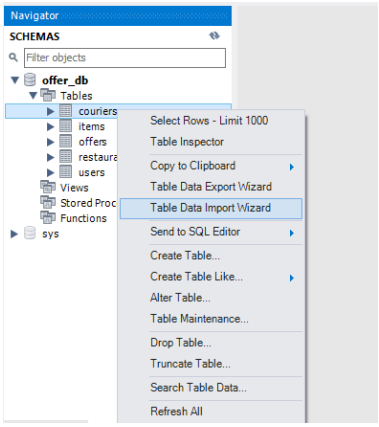
V tem koraku bomo ustvarili novo MySQL bazo *offer_db*, v katero bomo shranili podatke o ponudbi v restavracijah (restavracije, izdelki in njihovi ceni v posameznih restavracijah) ter uporabnikih in dostavljalcih. Za tisti korak lahko uporabite opcijo *Database > Forward Engineer* dostopno v Workbenchu ali pa direktno ustvarite tabele pri uvozu podatkov.

Ustvarite novo bazo (shemo) *offer_db* na podlagi ER modela prikazanega na sliki 2. **Pomembno:** Zaradi lažje integracije podatkov v naslednjih korakih implementacije cevovoda, poimenujte primarne ključe vseh tabel enako (npr. "id")!



Slika 2: ER model podatkovne baze o ponudbi v restavracijah.

Ko ste ustvarili ustrezne tabele, potrebno je uvesti podatke. Za uspešen uvoz podatkov iz lokalnega sistema v bazo je treba ustvariti ustrezne dokumente, ki imajo isto strukturo kot tabele v bazi (npr. dokument z izdelki mora imeti stolpce *id* in *name*). Z eŠtudija prevzemite CSV datoteke: *items.csv, offers.csv, users.csv, couriers.csv* in *restaurants.csv*. V Workbenchu naredite desni klik na posamezno tabelo, in izberite opcijo *Table Data Import Wizard*. Izberite ustrezen CSV dokument in dokončajte uvoz podatkov v vse tabele.

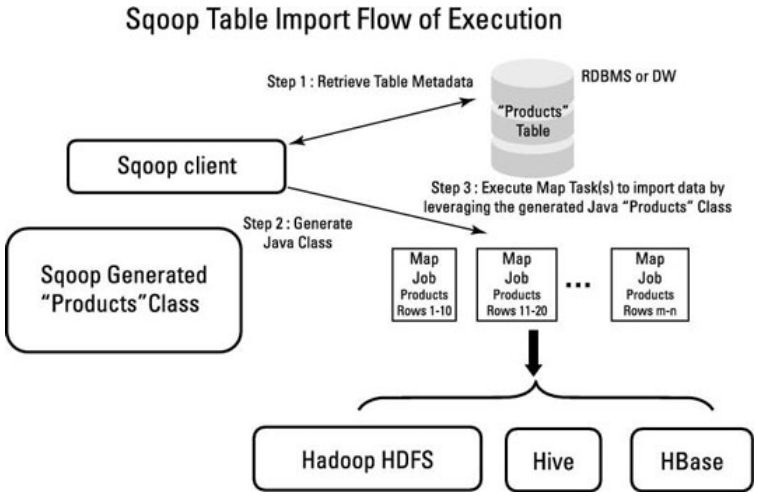


Slika 3: Uvoz podatkov iz CSV dokumenta v MySQL bazo z uporabo *Table Data Import* čarovnika.

Opomba: Ko ne uporabljate strežnik na Azure platformi, potem zaustavite vaš strežnik zaradi manjše uporabe resursov!

2. Apache Sqoop

Apache Sqoop ("SQL-to-Hadoop & Hadoop-to-SQL") je orodje znotraj Hadoop ekosistema namenjeno migraciji podatkov med relacijskimi sistemi in Hadoop rešitvami (HDFS, HBase). Sqoop se lahko uporabi tudi za prenos podatkov v večjih količinah med Hadoopom in zunanjim sistemom za shrambo podatkov. Za uvoz podatkov v Hadoop se uporablja orodje "Sqoop import".



Slika 4: Sqoop izvajanje procesa uvoza podatkov v Hadoop.

Pogoji za namestitev Sqoopa sta nameščena Java 8 in Hadoop. Na vajah bomo uporabljali Sqoop v.1.4.7 kompatibilno z izbrano Hadoop 3.2.1 različico.

Potrebno programsko okolje je pripravljeno v obliki Docker slike, ki je javno dostopna v Docker Hub repozitoriju pod imenom "sestakmartina/hadoop-hbase:latest". Če so Docker orodja nameščena na vašem računalniku, potem lahko prevzamete pripravljeno *docker-compose* datoteko, ki vsebuje definicijo Docker kontejnerja kot servisa, ki vključuje potrebni Hadoop in Sqoop namestitvi.

Najprej prevzamemo Docker sliko s DockerHuba z ukazom:

```
docker pull sestakmartina/hadoop-hbase:latest
```

Ko se v sistemskem terminalu nahajte znotraj mape kjer se nahaja prevzeta datoteka, kontejner v sistemu lahko zaženete z uporabo ukaza:

```
docker-compose up
```

Zgornji ukaz bo v sistemu zagnal kontejner z virtualnim sistemom znotraj katerega bodo nam dostopna orodja znotraj Hadoop ekosistema. V naslednjem koraku potrebujemo ID tistega kontejnerja, ki ga pridobimo z ukazom "docker ps". Potem izvedemo naslednji ukaz s katerim bomo vstopili v terminal znotraj virtualnega sistema:

```
docker exec -it <ID kontejnerja> bash
```

Pomembno: Na začetku je potrebno dodati *hadoop* ukaz v seznam sistemskih spremenljivk, kar lahko naredite z naslednjim ukazom (**potrebno izvesti ob vsakem zagonu Docker kontejnerja!)**:

```
export PATH=$PATH:$HADOOP_HOME/bin
export SQOOP_HOME=$SQOOP_HOME
export PATH=$PATH:$SQOOP_HOME/bin
```

Prvič ko želimo zagnati Hadoop gručo je potrebno reformatirati imensko vozlišče HDFS-a, s čimer bomo pobrisali vse morebitne predhodne metapodatke o podatkovnih vozliščih in pripravili strukturo direktorijev za HDFS. Imensko vozlišče formatiramo z ukazom:

```
$HADOOP_HOME/bin/hdfs namenode -format
```

Zdaj lahko zaženemo tudi osnovne komponente Hadoop gruče oz. HDFS in YARN z uporabo ukaza:

```
$HADOOP_HOME/sbin/start-all.sh
```

Uvoz podatkov iz relacijske podatkovne baze v HDFS z uporabo Sqoop-a

Da bi si olajšali uvoz podatkov v HDFS iz MySQL baze, kjer so podatki razdeljeni v treh tabelah, bomo najprej ustvarili novi pogled v relacijski bazi, v katerem bomo združili stolpce iz vse tri tabele:

```
create view v_offers as
    select offers.id as offer_id, restaurants.id as rest_id, restaurants.name
    as rest_name, items.id as item_id, items.name as item_name, price from offers
    join restaurants on offers.restaurants_id=restaurants.id
    join items on offers.items_id=items.id;
```

Rezultat tistega ukaza si lahko izravno shranimo kot CSV datoteko, ki bo potem predstavljala osnovni nabor podatkov za obdelavo.

Preverimo, ali smo zagnali osnovne procese za Hadoop gručo z ukazom *jps*). V seznamu procesov bi mogli videti 6 procesov če je vse uspešno zagnano v sistemu. Potem lahko izvedemo naslednji ukaz, s katerim bomo uvezli podatke iz MySQL oblačne podatkovne baze v lokalni HDFS:

```
sqoop import --connect "jdbc:mysql://<URL strežnika>:3306/<naziv pod. baze>?
useSSL=true&requireSSL=false&serverTimezone=Europe/Amsterdam" --username <uporabniško ime>
--password "<geslo>" --table v_offers --target-dir="/pts" -m 1
```

V tem ukazu lahko opazite, da smo znakovni niz za povezavo, ki smo ga dobili v Azure portalu, dopolnili z informacijo o časovnem pasu - to je potrebno pri uvozu podatkov zaradi usklajevanja med sistemi. Razen uporabniškega imena in gesla, kot argumente za Sqoop potrebujemo imeni izhodiščne tabele in ciljne poti v HDFS-u. Če je uvoz uspešno izveden, boste v HDFS spletnem vmesniku videli ustvarjeno mapo z dokumentom v obliki *part-m-00000*. V terminalu lahko izpišemo vsebino tega dokumenta z uporabo HDFS ukaza:

```
hadoop fs -cat /pts/part-m-00000
```

Rezultat ukaza *hadoop fs -cat* pa je izpis podatkov iz MySQL baze. Korake ponovite tudi za ostale tabele *users* in *couriers*, da bomo uvezli vse podatke v HDFS.

Opomba: Ko zaključite s delom, potrebno je zaustaviti Hadoop procese z ukazom:

```
$HADOOP_HOME/sbin/stop-all.sh
```