# Povezava za prijavo seminarja na učilnici: <https://ucilnica.fri.uni-lj.si/mod/wiki/view.php?id=50845>

Formirajte skupine po dva (2) študenta in se prijavite na zgornjo povezavo.

# 1. PostgreSQL in PostgresML: podatkovna analitika znotraj SUPB (lahko več skupin, vendar z različnimi podatki in načini uporabe)

PostgresML [1] je orodje, ki naj bi nam omogočalo uporabo priljubljenih orodij za podatkovno rudarjenje (Scikit-Learn, TensorFlow, XGBoost, …) znotraj podatkovne baze PostgreSQL. Raziščite zmožnosti orodja, podporo korakom CRISP-DM, integracijo z izbranim SUPB in različne načine uporabe. Vse skupaj ilustrirajte z uporabo na eni od večjih podatkovnih zbirk, rezultate kritično ovrednotite in predstavite v poročilu (vsaj 10 strani).

Vaša analiza naj vsebuje naslednje točke:

* Vnos podatkov v bazo
* Potrebna predpriprava podatkov za PostgresML (znotraj baze)
* Procesiranje podatkov (vsaj en pristop)
  + Nenadzorovano učenje (gručenje, asociacije, vizualizacija, …)
  + Nadzorovano učenje: izgradnja in uporaba napovednih modelov
  + Priporočilni sistemi (gradnja in uporaba)
* Ovrednotite tudi pomnilniško zahtevnost in možnost uporabe GPU (opcijsko)

Litertura:

1. PostgresML: <https://postgresml.org/>
2. PostgreSQL: <https://www.postgresql.org/>
3. CRISP-DM: <https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining>

# 2. Vgrajena podatkovna analitika z orodjem DuckDB (ena skupina)

DuckDB [1] je vgrajen (embedded) SUPB, soroden SQLite [2], ki nam omogoča uporabo OLAP analitike (večdimenzionalne vrtilne tabele, hierarhična agregacija, …) znotraj podatkovne baze. Raziščite zmožnosti orodja, podporo korakom CRISP-DM, ilustrirajte analitične ukaze in različne načine uporabe. Še posebej se osredotočite na možnost interaktivne analize v uporabniku prijaznem okolju, npr. Jupyter/Python/duckdb in Dbeaver/duckdb. Vse skupaj na koncu preizkusite na eni od večjih podatkovnih zbirk, rezultate kritično ovrednotite in predstavite v poročilu (vsaj 10 strani).

Vaša analiza naj vsebuje naslednje točke:

* Vnos podatkov v bazo
* Potrebna predpriprava podatov za analitiko (znotraj baze)
* Analitično procesiranje podatkov (opis in uporaba razširitev jezika SQL)
* Ovrednotite tudi skalabilnost, hitrost in pomnilniško zahtevnost sistema

Litertura:

1. DuckDB: [https://duckdb.org](https://duckdb.org/)
2. SQLite: <https://www.sqlite.org>
3. Python in DuckDB: <https://duckdb.org/docs/api/python/overview>
4. Dbeaver in DuckDB: <https://duckdb.org/docs/guides/sql_editors/dbeaver>
5. OLAP Hierarchical Aggregation with SQL: <https://medium.com/learning-sql/olap-hierarchical-aggregation-with-sql-6c45ebc206d7>
6. CRISP-DM: <https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining>

# 3. NocoDB: relacijska baza kot pametna preglednica (ena skupina)

NocoDB [1,2] je platforma, ki omogoča gradnjo aplikacij (skoraj) brez programiranja in razširja uporabo popularnih SUPB (PostgreSQL, Microsoft SQL Server, SQLit, MySQL/MariaDB) s spletnim uporabniškim vmesnikom na temelju preglednic (kot Excel). Omogoča preprosto konstrukcijo [3] vseh potrebnih gradnikov aplikacij in uporabniškega vmesnika (datotečne operacije, manipulacije s stolpci, filtriranje vrednosti, nadzor dostopa, …). Obenem omogoča tudi uporabo aplikacijskega programskega vmesnika (API) z različnimi programskimi jeziki, npr. s Pythonom [4].

Lokalno si namestite orodje (npr. Docker), raziščite njegove zmožnosti, podporo korakom CRISP-DM, povezljivost s programskimi jeziki (Python) ter ilustrirajte različne scenarije in načine uporabe. Vse skupaj na koncu preizkusite v obliki aplikacije na eni od večjih podatkovnih zbirk, rezultate kritično ovrednotite in predstavite v poročilu (vsaj 10 strani).

Vaša analiza naj vsebuje naslednje točke:

* Vnos podatkov in porebna predpriprava podatov
* Uporabniške in analitične funkcionalnosti preglednice
* Gradnja aplikacije
* Ocena uporabniške prijaznosti, enostavnosti razvoja, zmogljivosti in pomnilniške zahtevnosti

Litertura:

1. NocoDB: [https://www.nocodb.com](https://www.nocodb.com/)
2. NocoDB GitHub: <https://github.com/nocodb/nocodb>
3. How to get started with NocoDB: <https://www.madeleinesmith.uk/blog/get-started-nocodb>
4. NocoDB Python client: <https://pypi.org/project/nocodb>

# 4. Odprti podatki z ML/DM/AI ali DB/DW/DL analitičnimi orodji

Izberite si kombinacijo SUPB, analitičnega orodja in podatkov. Raziščite zmožnosti izbranega orodja, podporo korakom CRISP-DM, integracijo z izbranim SUPB in različne načine uporabe. Vse skupaj ilustrirajte z uporabo na večji podatkovni zbirki (vsaj 2 milijona zapisov).

* Orodja:
  + ML/DM/AI:
    - MindsDB: <https://mindsdb.com>
    - Rapid Miner: <https://rapidminer.com/>
    - Knime: <https://www.knime.com/>
    - Orange: <https://orangedatamining.com/>
    - Weka: <https://www.cs.waikato.ac.nz/ml/weka/>
    - R/Rattle: <https://rattle.togaware.com/>
    - Python/Scikit-learn + GUI (če kdo kaj najde)
    - Python/Scikit-learn/Ipython widgets
    - Python/Scikit-learn iz Excela
    - DataMelt: <https://datamelt.org/>
    - H2O (open source): <https://www.h2o.ai/products/h2o/>
    - ELKI: <https://elki-project.github.io/weka>
    - Tableau: <https://www.tableau.com/academic/students>
    - ... lastna izbira (po dogovoru)
  + DB/DW/DL:
    - MinIO: <https://min.io>
    - Microsoft SQL Server Data Tools
    - CRISP-DM: <https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining>
* Podatki:
  + Poiščite primerno velike podatke med viri odprtih podatkov, npr.: <https://www.bernardmarr.com/default.asp?contentID=960>

# 5. Povezovanje COVID-19 podatkovnih zbirk (ena skupina)

Preglejte različne vire podatkovnih zbirk za COVID-19. Preglejte sheme zbirk (strukturo tabel) in identificirajte atribute, preko katerih bi lahko povezovali podatke (npr. časovno in lokacijsko, po ključnih besedah, …). Iz pregledanih zbirk izberite vsaj 10 povezljivih, jih implementirajte v izbrano podatkovno bazo, izvedite polnjenje in korake, potrebne za povezavo (kreiranje novih atributov, definicija tujih ključev, …) in realizirajte vsaj 10 netrivialnih poizvedb nad povezanimi tabelami. Polnjenje in transformacijski koraki naj bodo realizirani tako, da bodo omogočali sprotno polnjenje tabele skozi čas. Na koncu podatke analizirajte in vizualizirajte z orodji iz točke 2.

Viri podatkov:

* Amazon: <https://aws.amazon.com/covid-19-data-lake/>
* EU: <https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data>
* Nature: <https://www.nature.com/collections/ebaiehhfhg>
* Kaggle: <https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge>
* Google: iščite COVID-19 open data

# Normalizacija in uporaba podatkovne tabele »Smetarine« (lahko več skupin uporabi te podatke, vendar z različnimi izbranimi orodji)

Gre za podatke o obračunavanju smetarin komunalnega podjetja za leti 2019 in 2020. Tabela ni normalizirana. Najprej prenesete podatke na svoj računalnik v MySQL bazo. Tabela »smetarine« se nahaja na pb.fri.uni-lj.si v shemi tup. Prenos lahko izvedete na več načinov (izvoz v datoteko, preko ODBC-ja ali PyODBC-ja). Izberite svojega in ga v nalogi opišite.

Podatki za normalizacijo:  
smetarine(StInkasa,DatumInkasa,InkasoOpis,stevilka,LastnikL,PlacnikL,HisnaStevilka,HisnaPoddelilka,DavcniZavezanec,ProracPorab,SkZnesek,SkDavek,SkZaPlacilo,DatumDokumenta,DatumValuta,Izravnava,Storno,ZamObresti,GKSaldo,ArtikelL,DavcnaSkupinaL,DavekOpis,Procent,Kolicina,ProdajnaCena,ProdajnaCenaZDavkom,StClanov,TagX,ObjektL,MerskaEnotaL,Enota,EnotaOpis)

Funkcionalne odvisnosti:

StInkasa->DatumInkasa,InkasoOpis

stevilka-> StInkasa,LastnikL,PlacnikL,SkZnesek,SkDavek,SkZaPlacilo,DatumDokumenta,DatumValuta, Izravnava,Storno,ZamObresti,GKSaldo

ali velja odvisnost stevilka,ArtikelL->Kolicina ?

PlacnikL->HisnaStevilka,HisnaPoddelilka,DavcniZavezanec,ProracPorab

ArtikelL->DavcnaSkupinaL,ProdajnaCena,ProdajnaCenaZDavkom

DavcnaSkupinaL->DavekOpis,Procent

LastnikL,ObjektL->StClanov

MerskaEnotaL->Enota,EnotaOpis

TagX kljukica označuje prvo postavko na računu. Na računu je lahko več objektov in postavke za nov objekt se začnejo s kljukico TagX in novo številko objekta

primer je račun št. 1836508

opisi atributov, ki niso jasni sami po sebi:

inkaso je skupna oznaka za mesečni obracun računov

Izravnava - stotinska izravnava, da ima znesek dve decimalki

Storno - dan ko je bil račun storniran

ZamObresti - koliko je bilo na računu pripisanih zamudnih obresti do dneva inkasa

GKSaldo - kakšno ima plačnik stanje v glavni knjigi / saldakontih na dan inkasa

še način izračuna vrednosti:

SkZnesek = vsota vrednosti postavk po formuli: Kolicina \* ProdajnaCena

SkDavek = vsota vrednosti postavk po formuli: Kolicina \* ProdajnaCena \* Procent / 100

SkZaPlacilo = vsota vrednosti postavk po formuli: SkZnesek + SkDavek + Izravnava

Tako normalizirane podatke uporabite za testiranje različnih orodij, navedenih v tem dokumentu, in realizirajte vsaj 10 netrivialnih poizvedb nad povezanimi tabelami.

Viri podatkov:

* Smetarine: <http://pb.fri.uni-lj.si/phpmyadmin/sql.php?server=1&db=tup&table=smetarine&pos=0&token=54e54d33e247df19eb3d4c9b6f478dee>