# SQLAlchemy zadaci

Pre početka izrade zadataka potrebno je inicijalizovati bazu podataka skriptom **init.sql**. Opis baze nad kojom se izvršavaju upiti je dat u nastavku:

Banka putem svojih filijala (prati se naziv i adresa) u raznim mestima (prate se poštanski broj i naziv) opslužuje svoje komitente (prati se naziv i adresa) koji mogu biti bez mesta, a u trenutku prvog pojavljivanja u banci prijavljuju sedište u određenom mestu. Svaki komitent može da ima više računa u svakoj od filijala (prate se status, broj stavki, dozvoljeni minus, i stanje), a mora imati bar jedan račun. Status računa može biti aktivan, blokiran ili ugašen. Račun postaje blokiran kada pređe u nedozvoljeni minus, a aktivira se kada stanje pređe u dozvoljeni minus. Komitenti sa svojih računa vrše transakcije putem stavki prometa (prate se redni broj, datum i vreme) koje mogu biti uplate (prati se osnov i iznos) ili isplate (prati se iznos i provizija), pri čemu je to moguće u bilo kojoj filijali.

ER model baze je dat na slici 1.

Diagram

Description automatically generated

Slika

Zadaci i SQL kod koji predstavlja njihova rešenja su dati u nastavku. **Python** kod klasa koje predstavljaju tabele u bazi podata i **Python** kod koji predstavlja rešenja ovih zadatak se nalazi u **main.py** datoteci.

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Komitent.

**SELECT** **\* FROM** Komitent

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun.

**SELECT** **\*** **FROM** Racun

1. Napisati SQL upit koji ispisuje nazive svih komitenata.

**SELECT** Naziv **FROM** Komitent

1. Napisati SQL upit koji ispisuje naziv i adresu svakog komitenta.

**SELECT** Naziv**,** Adresa **FROM** Komitent

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Komitent sortirane po nazivu u rastućem poretku.

**SELECT** **\***

**FROM** Komitent

**ORDER** **BY** Naziv

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Komitent sortirano po nazivu u opadajućem poretku.

**SELECT** **\***

**FROM** Komitent

**ORDER** **BY** Naziv **DESC**

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Komitent sortirane po nazivu u rastućem poretku, a zatim po adresi isto u rastućem poretku.

**SELECT** **\***

**FROM** Komitent

**ORDER** **BY** Naziv**,** Adresa

1. Napisati SQL upit koji ispisuje naziv i adresu svakog komitenta sortirano po nazivu u rastućem poretku, a zatim po adresi u opadajućem poretku.

**SELECT** Naziv**,** Adresa

**FROM** Komitent

**ORDER** **BY** Naziv**,** Adresa **DESC**

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun, za račune koji na stanju imaju tačno -55000 dinara.

**SELECT** **\***

**FROM** Racun

**WHERE** Stanje **=** -55000

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun, za račune kod kojih je stanje pozitivno.

**SELECT** **\***

**FROM** Racun

**WHERE** Stanje **>** 0

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun samo za blokirane račune.

**SELECT** **\***

**FROM** Racun

**WHERE** Status **=** 'B'

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun, za blokirane račune koji imaju stanje manje od -50000 dinara.

**SELECT** **\***

**FROM** Racun

**WHERE** Status **=** 'B' **AND** Stanje **<** -50000

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun, za blokirane račune ili račune koji imaju stanje manje od -50000 dinara.

**SELECT** **\***

**FROM** Racun

**WHERE** Status **=** 'B' **OR** Stanje **<** -50000

1. Napisati SQL upit koji ispisuje stanje računa, za blokirane račune koji imaju stanje manje od -50000 dinara.

**SELECT** Stanje

**FROM** Racun

**WHERE** Status **=** 'B' **AND** Stanje **<** -50000

1. Napisati SQL upit koji ispisuje sve podatke iz tabele Racun, za račune koji imaju stanje izmedju 12000 i 10000 dinara.

**SELECT** **\***

**FROM** Racun

**WHERE** Stanje **BETWEEN** 12000 **AND** 10000

1. Napisati SQL upit koji ispisuje stanje računa, kamatnu stopu i vrednost kamate za sve račune koji su u minusu Kamatna stopa ima vrednost od 3.

**SELECT** Stanje**,** 3**,** Stanje **\*** -0.03

**FROM** Racun

**WHERE** Stanje **<** 0

1. Napisati SQL upit koja za svaki račun ispisuje sve podatke i dodatno informaciju o tome da li će biti izvan dozvoljenog minusa kada mu se obračuna kamata za račune koji su trenutno u minusu.

**SELECT** **\*,** Stanje **\*** 1 .03 **<** **-**DozvMinus

**FROM** Racun

**WHERE** Stanje **<** 0

1. Napisati SQL upit koji ispisuje stanje računa, kamatnu stopu i vrednost kamate za sve račune koji su u minusu Kamatna stopa ima vrednost od 3. Potrebno je da kolone imaju sledeće nazive Stanje, Kamatna stopa i Kamata.

**SELECT** Stanje **,** 3 **AS** "Kamatna stopa"**,** Stanje **\*** 0.03 **AS** Kamata

**FROM** Racun

**WHERE** Stanje **<** 0

1. Napisati SQL upit koja za ra čune koji nisu u nedozvoljenom minusu ispisuje koliko mogu još maksimalno para da podignu.

**SELECT** IdRac**,** Stanje **+** dozvMinus

**FROM** Racun

**WHERE** Stanje **>** **-**dozvMinus

1. Napisati SQL upit koji ispisuje sve različite nazive komitenata (nazivi bez ponavljanja).

**SELECT** **DISTINCT** Naziv

**FROM** Komitent

1. Napisati SQL upit koji ispisuje račune koji su ugašeni (ugašeni računi imaju Stanje NULL).

**SELECT** **\***

**FROM** Racun

**WHERE** Stanje **IS** **NULL**

1. Napisati SQL upit koji ispisuje račune koji nisu ugašeni (ugašeni računi imaju Stanje NULL).

**SELECT** **\***

**FROM** Racun

**WHERE** Stanje **IS** **NOT** **NULL**

1. Napisati SQL upit koji ispisuje sumu stanja na svim računima.

**SELECT** **SUM(**Stanje**)**

**FROM** Racun

1. Napisati SQL upit koji ispisuje minimalno stanja na računima koji su u plusu.

**SELECT** **MIN(**Stanje**)** **AS** Minimum

**FROM** Racun

**WHERE** Stanje **>** 0

1. Napisati SQL upit koji ispisuje prosečno stanje na računima.

**SELECT** **AVG(**Stanje**),** **SUM(**Stanje**)** **/** **COUNT(**Stanje**),** **SUM(**Stanje**)** **/** **COUNT(\*)**

**FROM** Racun

1. Napisati SQL upit koji za svaki račun ispisuje naziv komitenta čiji je račun i stanje na računu.

**SELECT** Naziv**,** Stanje

**FROM** Komitent**,** Racun

**WHERE** Komitent**.**IdKom **=** Racun**.**IdKom

ili

**SELECT** Naziv**,** Stanje

**FROM** Komitent K**,** Racun R **WHERE** K**.**IdKom **=** R**.**IdKom

1. Napisati SQL upit koji za svaku stavku računa ispisuje u kojoj filijali je bila uplata/isplata, njen iznos i id racuna.

**SELECT** F**.**Naziv **AS** " Naziv filijale"**,** S**.**Iznos**,** S **.**IdRac

**FROM** Filijala F**,** Stavka S

**WHERE** F**.**idFIl **=** S**.**IdFIl

1. Napisati SQL upit koji za svaku stavku računa ispisuje u kojoj Filijali je bila uplata/isplata, njen iznos i id racuna, kao i naziv filijale u kojoj je otvoren račun.

**SELECT** FSt **.**Naziv **AS** "Mesto uplate/isplate "**,**

S**.**Iznos**,**

R**.**IdRac**,**

FRac**.**Naziv **AS** "Matična filijala"

**FROM** Filijala FSt **,** Stavka S**,** Racun R **,** Filijala FRac

**WHERE** FSt**.**idFIl **=** S**.**IdFIl **AND** S**.**IdRac **=** R**.**IdRac **AND** FRac**.**IdFil **=** R**.**IdFil

1. Napisati SQL upit koji za svaku platu ispisuje u kojoj Filijali je bila uplata.

**SELECT** F**.**Naziv **AS** "Naziv filijale "**,** S**.**Iznos

**FROM** Filijala F**,** Stavka S**,** Uplata U

**WHERE** F**.**idFIl **=** S**.**IdFIl **AND** S**.**idSta **=** U**.**idSta **AND** U**.**Osnov **=** 'Plata'

1. Napisati SQL upit koji za svaki id komitenta ispisuje ukupno stanje na njihovim računima.

**SELECT** idKom **,** **SUM(**Stanje**)** **AS** "Suma na racunima"

**FROM** Racun R

**GROUP** **BY** idKom

1. Napisati SQL upit koji za račun ispisuje koliko je bilo uplata na račun i koliko je iznosila njihova suma.

**SELECT** S**.**IdRac **,** **COUNT(\*)** **as** "Broj uplata"**,** **SUM(**S**.**Iznos **)** **AS** "Suma uplata"

**FROM** Stavka S**,** Uplata U

**WHERE** S**.**idSta **=** U**.**idSta

**GROUP** **BY** S**.**IdRac

1. Napisati SQL upit koji za svakog komitenta ispisuje ukupno stanje na njihovim računima.

**SELECT** K**.** Naziv**,** **SUM(**Stanje**)** **AS** "Suma na racunima"

**FROM** Komitent K**,** Racun R

**WHERE** K**.**idKom **=** R**.**idKom

**GROUP** **BY** K**.**idKom **,** K**.**Naziv

1. Napisati SQL upit koji za svakog komitenta koji ima bar jedan aktivan račun ispisuje broj aktivnih računa koje on ima.

**SELECT** K**.**Naziv **,** **COUNT(\*)** **AS** "Broj racuna"

**FROM** Komitent K**,** Racun R

**WHERE** K**.**idKom **=** R**.**idKom **AND** R**.**Status **=** 'A'

**GROUP** **BY** K**.**idKom **,** K**.**Naziv

1. Napisati SQL upit koji ispisuje broj računa na kojima je stanje pozitivno, samo za korisnike koji imaju bar jedan pozitivan račun.

**SELECT** K**.**Naziv **,** **COUNT(\*)** **AS** "Suma na racunima"

**FROM** Komitent K**,** Racun R

**WHERE** K**.**idKom **=** R**.**idKom **AND** Stanje **>=** 0

**GROUP** **BY** K**.**idKom **,** K**.**Naziv

1. Napisati SQL upit koji za komitente koji imaju ukupno pozitivno stanje na računima ispisuje naziv komitenta i sumu na tim računima.

**SELECT** K**.**Naziv**,** **SUM(**Stanje**)** **AS** "Suma na racunima"

**FROM** Komitent K**,** Racun R

**WHERE** K**.**idKom **=** R**.**idKom

**GROUP** **BY** K**.**idKom **,** K**.**Naziv

**HAVING** **SUM(**Stanje**)** **>=** 0

1. Napisati SQL upit koji ispisuje sve komitente koji imaju tačno dva računa.

**SELECT** K**.**Naziv

**FROM** Komitent K**,** Racun R

**WHERE** K**.**idKom **=** R**.**idKom

**GROUP** **BY** K**.**idKom **,** K**.**Naziv

**HAVING** **COUNT(\*)** **=** 2

1. Napisati SQL upit koji ispisuje sve komitente koji imaju tačno dva aktivna računa.

**SELECT** K**.**Naziv

**FROM** Komitent K**,** Racun R

**WHERE** K**.**idKom **=** R**.**idKom **AND** R**.**Status **=** 'A'

**GROUP** **BY** K**.**idKom **,** K**.**Naziv

**HAVING** **COUNT(\*)** **=** 2

1. Napisati SQL upit koji ispisuje sve račune koji su imali transakcije u dve ili više različitih filijala.

**SELECT** IdRac

**FROM** Stavka

**GROUP** **BY** IdRac

**HAVING** **COUNT(DISTINCT(**IdFil**))** **>=** 2

1. Napisati SQL upit koji ispisuje sve račune koji su imali transakcije u dva ili više različitih mesta.

**SELECT** S**.**IdRac

**FROM** Stavka S**,** Filijala F

**WHERE** S**.**IdFil **=** F**.**IdFil

**GROUP** **BY** S**.**IdRac

**HAVING** **COUNT(DISTINCT(**F**.**IdMes**))** **>=** 2

1. Prikazati sve filijale čija adresa ne sadrži reč „ trg“(vršiti case insensitive pretragu).

**SELECT** **\***

**FROM** Filijala F

**WHERE** F**.**Adresa **NOT** **LIKE** '%trg%'