

1. VJEŽBA

1. zadatak: Instalirati sustav za upravljanje bazama podataka i potrebne alate

Procjena potrebnog vremena: 45 minuta

U rješenjima je potrebno dostaviti: ništa

Sljedeći softver treba instalirati na računalo koje će vam u dovoljnoj mjeri biti dostupno tijekom cijelog semestra, a obavezno dostupno tijekom nadziranih provjera/vježbi koje će se održati u dva termina, po jednom na kraju svakog ciklusa predavanja.

- Microsoft SQL Server 2017 (ili noviji) Express Edition. Ova edicija je dovoljna, a najmanje je zahtjevna prema resursima računala
 - umjesto Express, može se koristiti jedna od sljedećih edicija: Enterprise, Free Trial ili Developer. To znači da nije nužno instalirati Express Edition ako već imate instaliranu i funkcionalnu neku od navedenih edicija
 - edicija Azure nije prikladna za vježbe na ovom predmetu
- Alat SSMS - **S**QL **S**erver **M**anagement **S**tudio 17.7. (ili noviji)
- Java SE Development Kit, JDK 8 (ili noviji)

2. zadatak: Kreirati instancu sustava za upravljanje bazama podataka

Procjena potrebnog vremena: 45 minuta

U rješenjima je potrebno dostaviti: ništa

Priprema za kreiranje instance:

- Proučiti što je to instanca SUBP-a
<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/database-engine-instances-sql-server>
- Na lokalnom disku, nevažno gdje točno, kreirajte tri direktorija (kazala, foldera) koja će tijekom vježbi predstavljati zamišljene fizičke medije za pohranu podataka s postojanom (*non-volatile*), stabilnom (*stable*) i tercijarnom (*tertiary*) memorijom. Npr.
 - C:/sbp/nonVolatileStorage
 - C:/sbp/stableStorage
 - C:/sbp/tertiaryStorage

Važnost karakteristika pojedinih tipova memorije bit će detaljnije objašnjena na predavanjima u poglavljima o obnovi.

Instanca koju ćete kreirati treba imati sljedeće značajke:

- Named instance: sbp
- Instance ID: SBP
- Collation: Croatian_CS_AS
- Authentication mode: Mixed Mode
- Baze podataka (root, system, user database) trebaju fizički biti smještene u postojanoj memoriji, tj. na mjestu koje predstavlja zamišljeni fizički medij s postojanom memorijom
- Logički dnevnik treba fizički biti smješten u stabilnoj memoriji, tj. na mjestu koje predstavlja zamišljeni fizički medij sa stabilnom memorijom
- Prostor za sigurnosne kopije (*backup*) treba fizički biti smješten u tercijarnoj memoriji, tj. na mjestu koje predstavlja zamišljeni fizički medij sa stabilnom memorijom

Default instancu sustava za upravljanje bazama podataka, ako je bila kreirana tijekom instalacije, možete (ali ne morate) obrisati, tj. uništiti.

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- kreirati instancu sustava (Database Engine Instance)
- objasniti što je instanca sustava
- uništiti odabranu instancu sustava
- pokrenuti (start) ili privremeno zaustaviti (pause) ili zaustaviti (stop) instancu sustava. Pokazati razumijevanje značenja tih akcija
- u alatu SSMS otvoriti ili zatvoriti konekciju prema instanci sustava
- pokazati gdje se i kako za neku instancu sustava može pogledati gdje su na fizičkom mediju smještene pripadne baze podataka, logički dnevnik i sigurnosne kopije

3. zadatak

Procjena potrebnog vremena: 15 minuta

U rješenjima je potrebno dostaviti: ništa

Kreirati bazu podataka

Za ovu vježbu koristite bazu podataka labprof1 u kojoj se na vrlo pojednostavljeni način evidentiraju podaci o studentima i nastavnim grupama za samo jedan predmet u samo jednoj akademskoj godini.

stud	Studenti koji su upisali predmet.
grupa	Nastavne grupe za predmet. Za svaku grupu se evidentira kapacitet grupe (broj studenata upisanih u grupu ne smije premašiti kapacitet grupe), te redundantno, broj studenata koji su zaista upisani u tu nastavnu grupu. Broj studenata upisanih u nastavnu grupu je (namjerno) redundantni podatak jer je izvediv iz podataka upisanih u relaciju studGrupa.
studGrupa	Sadrži podatak o tome koji student je upisan u koju nastavnu grupu.

Skripta za kreiranje baze podataka i punjenje relacija podacima nalazi se u datoteci *labprof1.sql*. Naredbe koje se tamo nalaze izvršite u instanci sustava sbp pomoću alata SSMS.

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- u alatu SSMS pokazati strukturu baze podataka labprof1, izvršavati SQL naredbe
- uništiti bazu podataka labprof1 i ponovo ju kreirati pomoću skripta *labprof1.sql*

4. zadatak: Najvažniji koncepti programskog jezika Transact-SQL

Procjena potrebnog vremena: 4 sata

U rješenjima je potrebno dostaviti: programski kod za zadatke 4.1 - 4.2.

Proučiti najvažnije o sljedećim elementima jezika Transact-SQL (T-SQL):

- T-SQL funkcije i procedure
 - razlika između funkcije i T-SQL procedure
 - line comment --, block comment /* */

- GO
- USE
- IF ... ELSE
- WHILE
- BEGIN ... END
- CONTINUE, BREAK, RETURN
- upravljanje pogreškama ili iznimkama (exception): TRY ... CATCH, THROW (ne RAISERROR)
- DECLARE @local_variable, SET @local_variable, SELECT @local_variable
- Cursors: DECLARE CURSOR, OPEN, FETCH, CLOSE, DEALLOCATE
- ulazni i izlazni parametri procedura i funkcija

Riješiti zadatke 4.1 - 4.2.

- 4.1. Napisati i testirati T-SQL pohranjenu proceduru ili funkciju (odlučite: proceduru ili funkciju) **upisStudenta** koja studenta s JMBAG-om zadanim kao ulaznim argumentom upisuje u grupu koja u trenutku poziva procedure/funkcije ima najmanju popunjenost kapaciteta (popunjenost kapaciteta je omjer broja studenata upisanih u grupu i kapaciteta grupe). Ako u tom trenutku više grupa ima jednaku (najmanju) popunjenost kapaciteta, studenta treba upisati u grupu s najmanje popunjenim kapacitetom koja istovremeno ima leksikografski najmanju oznaku grupe. Ako je student uspješno upisan u grupu, tada funkcija/procedura treba ažurirati vrijednost redundantnog atributa brojStud u odgovarajućoj n-torci relacije grupa.

Funkcija/procedura mora na sljedeće moguće pogreške reagirati na ovdje opisani način:

- Ako je student sa zadanim JMBAG-om već upisan u bilo koju od grupa, procedura/funkcija treba dojaviti pogrešku error_number=50501, message='Student je već upisan u nastavnu grupu', state=1
- Ako student sa zadanim JMBAG-om ne postoji, procedura/funkcija treba dojaviti pogrešku error_number=50502, message='Student ne postoji', state=1
- Ako niti u jednoj grupi nema slobodnih mjesta, procedura/funkcija treba dojaviti pogrešku error_number=50503, message='Sve grupe su već popunjene', state=1
- Ako se dogodi bilo koja druga pogreška (npr. dijeljenje s nulom), procedura/funkcija treba dojaviti originalnu pogrešku koju je producirao sustav za upravljanje bazama podataka (u ovdje spomenutom primjeru: error_number=8134, message="Divide by zero error encountered", state=1. Uputa: ovu pogrešku možete testirati tako da kapacitet neke grupe postavite na nulu i zatim pokušate upisati nekog postojećeg studenta

- 4.2. Napisati i testirati T-SQL pohranjenu proceduru/funkciju **rasporediPoGrupama**. Procedura/funkcija raspoređuje studente u grupe tako da za svakog studenta koji se nalazi u relaciji stud, prema abecednom redoslijedu, po jednom pozove proceduru/funkciju upisStudenta.

Funkcija/procedura mora na sljedeće moguće pogreške reagirati na ovdje opisani način:

- Ako poziv procedure/funkcije upisStudenta izazove pogrešku error_number=50501 ili 50502, zanemariti pogrešku i nastaviti raspoređivanje ostalih studenata
- Ako poziv procedure/funkcije upisStudenta izazove pogrešku error_number=50503, procedura/funkcija treba dojaviti pogrešku error_number=50504, message='Nije uspjelo raspoređivanje x studenata', state=1, pri čemu x treba zamijeniti odgovarajućim brojem (koliko je studenata ostalo neraspoređeno).
- Ako se dogodi bilo koja druga pogreška (npr. dijeljenje s nulom), procedura/funkcija treba dojaviti originalnu pogrešku koju je producirao sustav za upravljanje bazama podataka (u ovdje spomenutom primjeru: error_number=8134, message="Divide by zero error encountered", state=1
- testirati:
 - raspoređivanje studenata za početno stanje podataka u relacijama stud, grupa i studGrupa
 - raspoređivanje studenata za početno stanje podataka u relacijama stud, grupa i studGrupa, osim što je kapacitet grupe P1 postavljen na 1
 - raspoređivanje studenata za početno stanje podataka u relacijama stud, grupa i studGrupa, osim što je kapacitet grupe P1 postavljen na 0

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- testiranjem dokazati da procedura/funkcija iz zadataka 4.1. - 4.2 zadovoljavaju zahtjeve postavljene u zadacima
- uz pomoć slobodno odabrane *on-line* dokumentacije i slobodno odabranih vlastitih rješenja i primjera (znači: ne očekuje se da ćete T-SQL sintaksu učiti napamet) napisati i koristiti jednostavnu T-SQL funkciju ili proceduru (pomoću alata SSMS)
- objasniti T-SQL naredbe koje su korištene tijekom rješavanja zadataka 4.1 - 4.2 ili u zadacima koje ćete rješavati u laboratoriju

5. zadatak: Koncepti programskog sučelja JDBC

Procjena potrebnog vremena: 4.5 sati

U rješenjima je potrebno dostaviti: programski kod za zadatke 5.1 - 5.5. i vrijeme (broj milisekundi) iz rješenja zadataka 5.4 i 5.5,

Primjer: U datotekama `RasporediPoGrupama1.java` i `RaporediPoGrupama2.java` nalaze se dva zasebna Java programa pomoću kojih su ilustrirani koncepti programskog sučelja JDBC. Programi obavljaju istu zadaću, ali koriste različite tipove klijent-poslužitelj arhitekture.

Proučiti priložene programe. Zaključiti po čemu se razlikuju tipovi klijent-poslužitelj arhitekture u ovim programima.

Prevesti i testirati programe iz primjera. Slobodno odaberite želite li programe prevoditi i izvršavati na razini ljuške operacijskog sustava ili u odabranom razvojnom okruženju (Eclipse, NetBeans, itd.). U nastavku je prikazano kako se programi mogu prevoditi i izvršavati u ljuhci operacijskog sustava:

1. Priloženi program pohraniti u nekom direktoriju, npr. `c:\temp`
2. U isti direktorij kopirati JDBC driver za SQL Server, npr. `mssql-jdbc-7.2.1.jre8.jar`.
3. Promijeniti program tako da postavke za spajanje na bazu podataka odgovaraju vašoj bazi podataka (u dijelu programa `url == "jdbc:..."`).
4. Prevesti program koristeći prevoditelj `javac`. Iz naredbene linije (*Command prompt*), iz odabranog direktorija, obaviti sljedeće naredbe:
 - Podesiti put do prevodioca (ako već nije postavljen):
`C:\temp>set PATH=%PATH%;"C:\Program Files\Java\jdk1.x.xxx\bin";`
Crveno označeni put ovisi o tome gdje je zaista instaliran Java JDK.
 - Prevesti program:
`C:\temp>javac -cp .;mssql-jdbc-7.2.1.jre8.jar RasporediPoGrupama1.java`
Ako je program uspješno preveden u direktoriju se pojavljuje datoteka `RasporediPoGrupama1.class`.

Program se pokreće na sljedeći način:

```
C:\temp>java -cp .;mssql-jdbc-7.2.1.jre8.jar RasporediPoGrupama1
```

Eventualne nejasnoće u vezi sučelja JDBC mogu se razjasniti pomoću materijala *Uvod u JDBC* i slobodno odabrane dokumentacije dostupne na Internetu.

Riješiti zadatke 5.1 - 5.5.

- 5.1. Napisati i testirati Java program **UpisStudenta1.java** koji poziva proceduru/funkciju iz zadatka 4.1. Program za jedan JMBAG zadan u naredbenoj liniji poziva proceduru **upisStudenta**, te zavisno od rezultata na zaslon (System.out) ispisuje jednu od sljedećih poruka:
- Student je uspješno upisan
 - 50501; Student je već upisan u nastavnu grupu; State=S0001
 - 50502; Student ne postoji; State=S0001
 - 50503; Sve grupe su već popunjene; State=S0001
 - xxxxxx; tekst poruke; State=yyyyy
 - ovo posljednje za slučaj da se dogodi neka druga pogreška, npr. ako u bazi podataka ne postoji relacija studGrupa, ispisuje se 208; Invalid object name 'studGrupa'.; State=S0002
- 5.2. Napisati i testirati Java program **UpisStudenta2.java** koji obavlja istu zadaću kao u zadatku 5.1, ali bez korištenja procedure/funkcije **upisStudenta**, odnosno u skladu s klijent-poslužitelj arhitekturom oslonjenom na klijenta (*client centric*). Zavisno od rezultata na zaslon (System.out) ispisuje jednu od sljedećih poruka:
- Student je uspješno upisan
 - Student je već upisan u nastavnu grupu
 - Student ne postoji
 - Sve grupe su već popunjene
 - xxxxxx; tekst poruke; State=yyyyy
 - ovo posljednje za slučaj da se dogodi neka druga pogreška, npr. ako u bazi podataka ne postoji relacija studGrupa, ispisuje se 208; Invalid object name 'studGrupa'.; State=S0002
- 5.3. Napisati i testirati Java program **ListaPoGrupama.java** koji će ispisati popis studenata po grupama (pretpostavlja se da su studenti već raspoređeni po grupama), poredano po grupama, unutar grupa po prezimenu i imenu:
- ```
P1 0036123456 Ante Antić
P1 0036222256 Jadranka Jadrić
P1 0036223456 Petar Petrić
P1 0036221456 Šime Šimić
P2 0036135456 Ana Aniće
P2 0036243536 Boško Boškić
...
```
- 5.4. Preuzeti datoteku salesOrderItem.csv. Pomoću skripta *labprof1dodatak.sql* kreirati i napuniti podacima relaciju salesOrderItem. Napisati i testirati Java program ServerCentric.java. Program treba biti napisan kao izrazito oslonjen na poslužitelja (server-centric). Program treba za svaku naručenu količinu koja je manje od 10, ispisati koliko u relaciji salesOrderItem ima n-torki u kojima je navedena dotična količina (orderQty). Ispisivati samo one naručene količine za koje postoji barem jedna n-torka. Poredati prema količini. Na kraju ispisati koliko je vremena bilo potrebno za dohvat i ispis podataka.

Ispravan rezultat izvršavanja programa je prikazan u nastavku (jedino će vrijeme izvršavanja gotovo sigurno biti drugačije od ovdje prikazanog). U ispisu se vidi da u relaciji ima 74954 n-torki s naručenom količinom 1, 14200 n-torki s naručenom količinom 2, itd.

```
1 74954
2 14200
3 10058
4 7494
5 4386
7 1735
9 837
Vrijeme: 51 milisek
```

- 5.5. Jednako kao zadatku 5.4, s jedinom razlikom što treba napisati Java program `ClientCentric.java`. Program treba napisati tako da bude izrazito oslonjen na klijenta (client-centric). Rezultat treba biti jednak onom u prethodnom zadatku, osim vremena izvršavanja za koje se očekuje da će biti **značajno** dulje.

```
1 74954
2 14200
3 10058
4 7494
5 4386
7 1735
9 837
Vrijeme: 112 milisec
```

*U rješenju se očekuje namjerno loše dizajniran način obrade podataka ("groteskno" oslonjen na klijenta), onakav kakvog u praksi nikad ne bi trebalo napisati za rješavanje problema ovog tipa.*

Tijekom nadziranih provjera/vježbi očekuju se sljedeća znanja i vještine:

- objasniti što je i čemu služi JDBC
- objasniti programe iz primjera, zadataka 5.1 - 5.5. i zadataka koje ćete rješavati u laboratoriju
- testiranjem dokazati da programi iz zadataka 5.1. - 5.5. zadovoljavaju zahtjeve postavljene u zadacima
- uz pomoć slobodno odabrane *on-line* dokumentacije i slobodno odabranih vlastitih rješenja i primjera (znači: ne očekuje se da ćete Java i JDBC sintaksu učiti napamet) napisati i koristiti jednostavni Java program s osnovnim elementima JDBC-a