



# OBLIKOVANJE BAZA PODATAKA

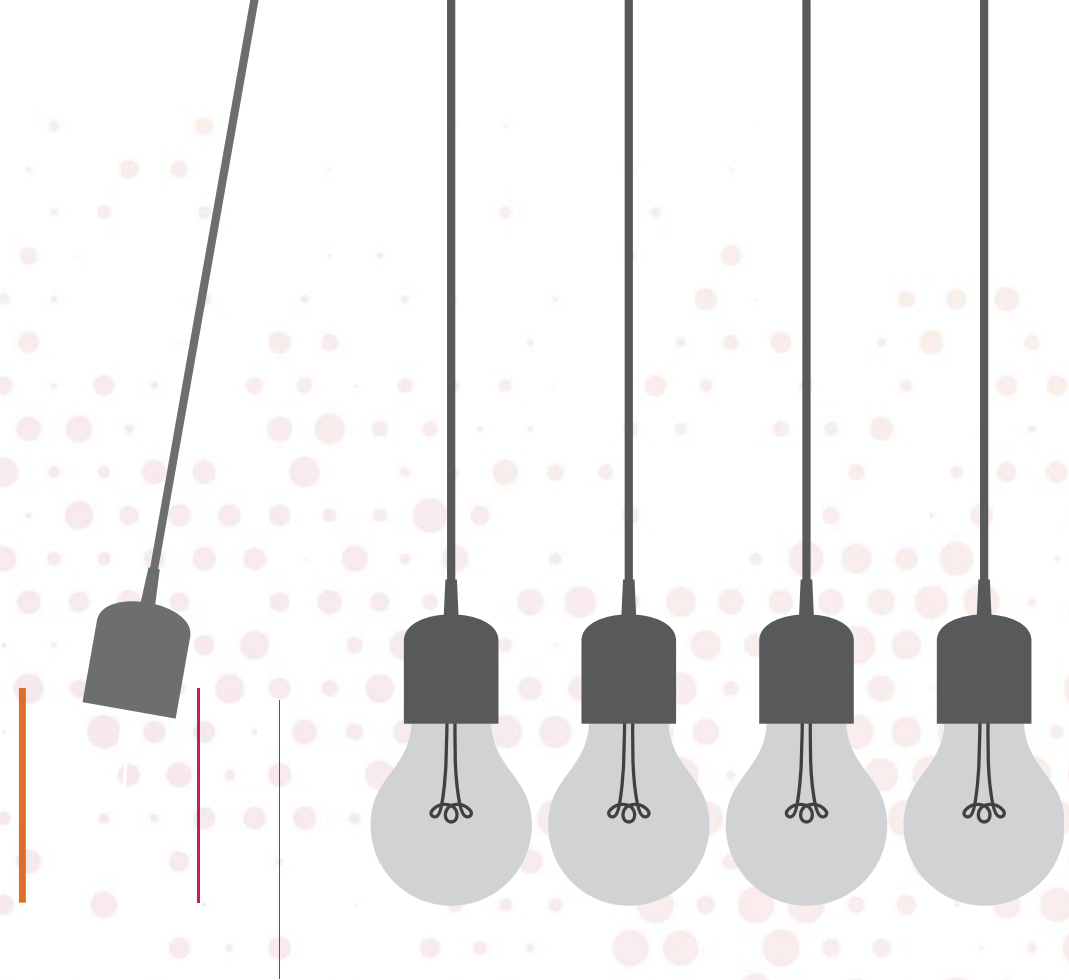
Upute za pohađanje i  
polaganje kolegija  
akademska godina 25/26

# Organizacija predavanja i vježbi

<b>Nositelji kolegija</b>	Mario Fabijanić Mateo Sokač	<a href="mailto:mario.fabijanic@algebra.hr">mario.fabijanic@algebra.hr</a> <a href="mailto:mateo.sokac@algebra.hr">mateo.sokac@algebra.hr</a>
<b>Asistenti</b>	Matija Kolarić	<a href="mailto:mkolaric@algebra.hr">mkolaric@algebra.hr</a>
	Aron Thompson	<a href="mailto:aron.thompson@algebra.hr">aron.thompson@algebra.hr</a>
<b>Izvođenje nastave</b>	<b>Predavanja</b>	2 sata tjedno - ukupno 30 sati
	<b>Vježbe</b>	2 sata tjedno - ukupno 30 sati

# Informacije o kolegiju

- 5 ECTS bodova = 150 sati rada studenta
  - 30 sati predavanja  
(15 tjedana po 2 sata)
  - 30 sati vježbi  
(15 tjedana po 2 sata)
  - 90 sati rada kod kuće  
(15 tjedana po 6 sati)



# Cilj kolegija

- Proširiti znanje o bazama podataka i SQL jeziku
- Naučiti pripremiti bazu podataka za pristup iz programskog kôda

# Ishodi učenja

Ishod učenja	MINIMALNI ISHODI UČENJA (po uspješnom završetku kolegija, student će moći)	ŽELJENI ISHODI UČENJA (uspješan student bi trebao moći)
I1	Koristiti osnovne naredbe za izradu i uporabu pogleda i okidača u relacijskim bazama podataka.	Koristiti osnovne naredbe za izradu i uporabu pogleda i okidača u relacijskim bazama podataka, uključujući mogućnosti podešavanja i implementacije rekurzivnih funkcija.
I2	Koristiti osnovne naredbe za izradu i uporabu postupaka i funkcija te rješavanje problema sporog zaprimanja upita uporabom indeksa u relacijskoj bazi podataka.	Koristiti osnovne naredbe za izradu i uporabu složenih postupaka i funkcija te rješavanje problema sporog zaprimanja upita uporabom indeksa u relacijskoj bazi podataka.
I3	Koristiti osnovne naredbe za izvođenje operacija CRUD u relacijskoj bazi podataka.	Koristiti osnovne naredbe za izvođenje operacija CRUD u relacijskoj bazi podataka, uključujući glavne i podređene odnose.
I4	Koristiti metode JSON i XML i korisnički definirane tablične tipove podataka u relacijskoj bazi podataka.	Preporučiti implementaciju CRUD operacija pomoću korisničkih procedura i/ili funkcija u relacijskoj bazi podataka te ih implementirati
I5	Koristiti transakcije i razine izolacije u relacijskoj bazi podataka.	Koristiti transakcije i razine izolacije u relacijskoj bazi podataka, uključujući pravilno postupanje s pogreškama.
I6	Koristiti napredne funkcije grupiranja, ugrađene funkcije za rad s prozorima i analitičke funkcije u relacijskoj bazi podataka.	Koristiti napredne funkcije grupiranja, ugrađene funkcije za rad s prozorima i analitičke funkcije u relacijskoj bazi podataka, uključujući dodatne particije RANGE i ROWS.
I7	Usporediti različite nerelacijske modele baza podataka i osnovne naredbe za upravljanje podacima u nerelacijskim bazama podataka.	Usporediti različite nerelacijske modele baza podataka i osnovne naredbe za upravljanje podacima u nerelacijskim bazama podataka te predložiti optimalni model.

# Tematske cjeline

Tjedan nastave	Cjelina	Tjedan nastave	Cjelina
1. Tjedan	Pogledi	9. Tjedan	CRUD operacije
2. Tjedan	Pogledi	10. Tjedan	JSON i XML metode
3. Tjedan	Okidači	11. Tjedan	Transakcije i izolacijski nivoi, upravljanje pogreškama
4. Tjedan	Procedure, funkcije i indeksi	12. Tjedan	Transakcije i izolacijski nivoi, upravljanje pogreškama
5. Tjedan	Procedure, funkcije i indeksi	13. Tjedan	Grouping funkcije, analitičke funkcije, particioniranje
6. Tjedan	Procedure, funkcije i indeksi	14. Tjedan	Usporedba s nerelacijskim bazama
7. Tjedan	Procedure, funkcije i indeksi	15. Tjedan	Ponavljjanje
8. Tjedan	Procedure, funkcije i indeksi		



# Literatura

## OBAVEZNA LITERATURA

- Dušan Petković (2020): Microsoft SQL Server 2019: A Beginner's Guide, Seventh Edition, Oracle Press © 2020, ISBN 9781260458879.

## PREPORUČENA LITERATURA

- Jason Brimhall, Jonathan Gennick, Wayne Sheffield (2015): SQL Server T-SQL Recipes, Fourth Edition, Apress, ISBN 9781484200629.

# Za potpis treba?

Za stjecanje prava na potpis potrebno je prisustvovati nastavi u postotku propisanom Pravilnikom o studijima i studiranju.

## **Dolaznost na predavanja i vježbe**

najmanje 50% fizičke  
prisutnosti na predavanjima

najmanje 60% fizičke  
prisutnosti na vježbama

Tko ne dobije potpis, mora sljedeće godine ponovno upisati kolegij, platiti upis kolegija te nema pravo polaganja ispita.



# Polaganje kolegija

- Kolegij ima definiranih 7 ishoda učenja.
- Da bi student položio kolegij mora po svakom ishodu učenja ostvariti minimalno 50% bodova raspoloživih za taj ishod učenja.
- **Ako student ne ostvari 50% bodova iz nekog ishoda učenja, na slijedećem roku treba opet polagati taj ishod učenja.**
- Metode provjeravanja ishoda učenja:
  - Međuispiti 90 bodova
  - Školske zadaće 10 bodova

# Kako je to raspoređeno po ishodima učenja

ISHOD	Školska zadaca	Međuispit 1	Međuispit 2	Dolaznost	MAX
I1		24			24
I2	2	13			15
I3	2	7			9
I4	2		12		14
I5			12		12
I6	2		12		14
I7	2		10		12
Ukupno	10	44	46	0	100

# Ocjenjivanje

Broj osvojenih bodova	Ocjena
0,00 – 50,00	1 (nedovoljan)
50,01 – 58,00	2 (dovoljan)
58,01 – 75,00	3 (dobar)
75,01 – 92,00	4 (vrlo dobar)
92,01 – 100,00	5 (izvrstan)

# Ispiti

- Na svakom kolegiju vrijedi **pravilo 3 + 1**
  - To znači da student mora položiti ispit iz najviše 4 izlaska
    - 3 redovna izlaska – Uključena u cijenu školarine
    - 1 izvanredni izlazak – Odlukom o naknadi troškova 4. prijava ispita se naplaćuje
  - Vremenski rok za položiti ispit je **12 mjeseci** od dana upisa kolegija
  - Ako student u 12 mjeseci ne položi kolegij, **mora ponovno upisati kolegij te ponovno polagati sve ishoda učenja kako je definirano kolegijem**, bez obzira na broj izlazaka na ispit
- **Vodite računa o rokovima prijave i odjave ispita na IE**
  - Ako niste prijavili ispit na vrijeme, ne možete pristupiti ni pismenom, niti usmenom dijelu, ni obrani projekta.
  - Naknadne prijave nisu moguće, ako ste propustili prijaviti ispit na vrijeme, nema mogućnosti da se naknadno prijavi od strane referade.
  - Ako je student prijavio više ispitnih rokova iz istog kolegija, pri dobivanju ocjene kojom je zadovoljan, dužan je odjaviti svaki sljedeći rok koji je iz tog kolegija prijavio. U suprotnom, studentu se u Infoeduku unosi nedovoljan (1).



# Školske zadaće – kontinuirana provjera znanja

- Kontinuirane provjere znanja nisu obavezne
- Broj bodova koji se odnosi na kontinuirane provjere znanja ne onemogućava studentima da polože ispit pismenim putem, no onemogućavaju najvišu ocjenu
- Kontinuirane provjere znanja pišu se na vježbama
- Nije moguća nadoknada izvan tjedna u kojem se pišu

# Akademski standard ponašanja

- U komunikaciji (pisanoj i usmenoj) pridržavati se pravila poslovne komunikacije primjerene akademskoj razini.
- Potrebno je držati se jasno definiranih rokova za predaju zadataka (zadaca, seminarskih radova, projekata i sl.).
  - Svaki zadatak, domaća zadaća, projekt itd., poslani nakon definiranog roka neće se ocjenjivati.
- Samo oni studenti koji mogu potvrditi svoje pohađanje, smatrat će se prisutnima.
  - Potpisivanje drugih studenata ili registracija njihovom karticom nije dopušteno i može biti predmet stegovnog postupka. Nastavnik će obrisati prisustvo ako utvrdi da je student prijavljen, a da nije prisutan na nastavi.



# Pravila ponašanja na nastavi

- Na nastavu se dolazi na vrijeme.
- Pri ulasku u učionicu student prilazi do stola i prijavljuje se na nastavu karticom te sjeda na dostupno mjesto za rad.
- Ometanje nastave i neaktivno sudjelovanje na nastavi nije dozvoljeno.
  - Repetitivno kršenje ovog pravila sankcionira se prijavom stegovnom povjerenstvu.

**Hvala na pažnji!**





# OBLIKOVANJE BAZA PODATAKA

Predavanje 01 i 02

# Ponavljjanje

- Instalirati MS SQL Server
- Instalirati AdventureWorksOBP

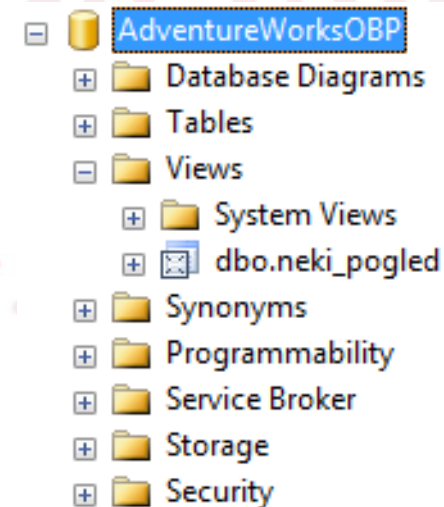


# Pogledi

# Uvod

- **Pogledi** (engl. *views*) su objekti u bazi podataka
  - Često se nazivaju i **virtualne tablice**
  - Ne sadrže podatke, već samo SQL upite
    - Svaki pogled sadrži točno jedan SELECT upit (koji može sadržavati i podupite, spajanja, grupiranja, ...)
  - Dodaju razinu apstrakcije nad tablice
  - Koriste se (skoro) na isti način kao i tablice, primjerice:

```
SELECT * FROM neki_pogled
```
  - Mogu se pronaći u svakoj bazi :





# Prednosti pogleda

- Razlozi za postojanje pogleda su:
  1. Smanjenje kompleksnosti dijelova baze prema korisnicima
    - Primjerice:
      - Upit koji dohvaća ime i prezime kupca te nazive grada i države zahtijeva više spajanja
      - Taj upit je dobar kandidat za staviti u pogled
      - Korisnik koristi pogled ne znajući ništa o spajanjima
  2. Zaštita podataka
    - Primjerice, pogled ne sadržava stupac koji sadrži plaću
  3. Implementacija sučelja prema korisnicima baze
    - Primjerice, promjena naziva stupca u tablici koju koristi pogled ne mijenja način korištenja pogleda

# DEMO – Implementacija sučelja

1. Napraviti tablicu Klijent (IDKlijent, Ime, Prezime, Tel1, Tel2, Tel3)
2. Napuniti tablicu s nekim podacima
3. Neka aplikacije koriste tablicu (npr. `SELECT Ime, Prezime, Tel1, Tel2, Tel3 FROM Klijent`)
  - Ako sad želimo normalizirati tablicu do 3NF, što će se desiti s aplikacijama?
  - Prestat će raditi => moramo i njih mijenjati
  - Rješenje je napraviti pogled koji će se zvati Klijent i koji će osiguravati da sučelje baze prema aplikaciji ostane nepromijenjeno

# Sistemi i korisnički pogledi

- **Sistemi** pogledi omogućuju uvid u razne aspekte RDBMS-a, baze podataka i objekata u bazi podataka
  - Služe za uvid u općenite podatke zajedničke za sve baze podataka
  - Primjerice, pogled koji vraća podatke o tablicama koje postoje u bazi ima smisla u svakoj bazi
- **Korisnički** pogledi služe za uvid u poslovne podatke specifične za korisnikovu bazu podataka
  - Primjerice, pogled koji vraća račun i njegove stavke nema smisla u bazi koja čuva podatke o izbornim jedinicama i glasačima

# Osnovni sistemski pogledi

- Osnovni sistemski pogledi su:
  - `sys.databases`
  - `sys.tables`
  - `sys.columns`
    - Stupac `object_id` sadrži identifikator tablice kojoj stupac pripada
      - Funkcija `object_id('naziv')` vraća identifikator za zadani naziv
      - Funkcija `object_name(id)` vraća naziv za zadani identifikator
  - `sys.key_constraints` – podaci o primarnim ključevim i UNIQUE ograničenjima
    - `parent_object_id` je tablica
  - `sys.foreign_keys`
    - `parent_object_id` je tablica
    - `referenced_object_id` je stupac na koji pokazuje



# Zadaci

- Kad je napravljena baza AdventureWorksOBP?
- Ispišite sve obavezne stupce koji pripadaju tablici Kupac.
- Ispišite naziv stupca koji se najčešće pojavljuje u bazi AdventureWorksOBP te ispišite i koliko je to puta.
- Ispišite sve primarne i UNIQUE ključeve na tablici Proizvod.
- Ispišite nazive svih stranih ključeva u tablici Racun te napišite tablice kamo vode.

# Korisnički pogledi



# Struktura pogleda

- Osnovni dijelovi pogleda:

1. Shema i naziv

- Pravila imenovanja kao i za tablice

2. Opcije

- WITH SCHEMABINDING
- WITH ENCRYPTION
- WITH CHECK OPTION

3. Sadržaj

- Sadržaj čini **jedna SELECT naredba**, uz sljedeća ograničenja:
  - Svi stupci moraju imati jedinstveni naziv
  - Ako sadržava ORDER BY dio, mora koristiti i TOP dio
- Pogled ne može sadržavati ništa osim jedne SELECT naredbe (koja može sadržavati i podupite, spajanja, grupiranja, ...)

# Pravo korištenja pogleda

- Ako korisnik ima pravo SELECT na pogled, to znači da ima pravo pristupiti svim objektima koje taj pogled koristi
  - Ne mora imati nikakva dodatna prava na objektima koje taj pogled koristi
- Prethodno vrijedi ako lanac vlasništva nije prekinut (engl. *unbroken ownership chain*)
  - To znači da je vlasnik pogleda ujedno i vlasnik svih objekata koje pogled koristi
  - U suprotnom, potrebna su dodatna prava
  - <https://msdn.microsoft.com/en-us/library/ms188676.aspx>

# T-SQL operacije s pogledima

- Izrada (mora biti u vlastitoj skupini naredbi):

```
CREATE VIEW shema.naziv_pogleda [ opcije ]  
AS  
SELECT_naredba  
[ opcije ]
```

- Izmjena (mora biti u vlastitoj skupini naredbi):

```
ALTER VIEW shema.naziv_pogleda [ opcije ]  
AS  
SELECT_naredba  
[ opcije ]
```

- Uklanjanje (ne mora biti u vlastitoj skupini naredbi):

```
DROP VIEW naziv_pogleda
```

# Primjeri

1. Situacija: neki korisnik nema pravo koristiti tablicu Proizvod, a treba određene podatke. Pomognite mu tako što ćete napraviti pogled koji će dohvaćati sve proizvode.
2. Iskoristite pogled za dohvaćanje svih zapisa
  - a. Iskoristite pogled za dohvaćanje zapisa u potkategoriji 13
  - b. Iskoristite pogled za ispis boja proizvoda i broja proizvoda u svakoj boji, padajuće prema broju proizvoda
  - c. Iskoristite pogled i pokraj naziva proizvoda ispišite i naziv potkategorije
3. Promijenite pogled tako da preimenujete stupac Naziv u NazivProizvoda. Dohvatite podatke kroz pogled.
4. Uklonite pogled



# Primjeri

5. Vašeg šefa (koji voli Management Studio) zanimaju podaci: ime i prezime komercijalista i koliko je ukupno prodao proizvoda. Rezultate želi sortirane opadajuće prema broju prodanih proizvoda.
  - a. Objasnite šefu kako da napiše upit ☹️
  - b. Napravite pogled i objasnite šefu kako da ga koristi 😊
6. Obrišite pogled.
7. Gospođa iz prodaje treba pristup podacima o kreditnim karticama, ali samo za kartice tipa Diners. Napravite pogled. Pomoću pogleda dohvatite sve podatke o karticama tipa Diners, a zatim o karticama tipa Visa.
8. Obrišite pogled.

# Primjeri

9. Marketing treba podatke o svim proizvodima koji su prodani u više od 2000 primjeraka. Napravite pogled koji će to omogućiti.
10. Dodajte u pogled informaciju o broj prodanih proizvoda.
11. Uklonite pogled.



# Blic

- URL: <https://bit.ly/3LFrE1R>



# Izmjena podataka kroz poglede

# Izmjena podataka kroz poglede

- Pogled ne sadrži podatke, nego samo **SELECT** upit
- Podatke je ipak moguće mijenjati kao da se radi o tablici, uz sljedeće uvjete:
  - Sve promjene moraju referencirati stupce iz **jedne tablice**
  - Referencirani stupci ne smiju biti rezultat podupita, agregatnih funkcija niti kombinacija drugih stupaca
  - Referencirani stupci ne smiju biti dijelom GROUP BY, HAVING niti DISTINCT ključnih riječi
  - Pogled sadržava sve neophodne stupce
- Preporuča se podatke mijenjati samo kroz one poglede koji vraćaju podatke iz samo jedne tablice

# Korištenje WHERE pri izmjeni podataka

- SELECT naredba u pogledu može sadržavati i WHERE dio i na taj način vraćati samo podskup redaka
- Ako kroz pogled umetnemo/promijenimo redak koji neće zadovoljiti WHERE uvjet:
  - Predefinirano ponašanje je **dopuštanje umetanja/izmjene** – zbunjujuće jer korisnik više ne vidi redak
  - Moguće definirati da se takvi retci ne smiju umetati/mijenjati korištenjem opcije **WITH CHECK OPTION**:

```
CREATE/ALTER VIEW naziv_pogleda  
AS  
SELECT_naredba  
WITH CHECK OPTION
```

# Primjeri

12. Napravite pogled koji će dohvaćati sve iz tablice Kupac. Možete li napraviti INSERT, UPDATE i DELETE nekog kupca?
13. Promijenite prethodni pogled tako da dohvaća sve stupce osim Prezime. Možete li napraviti INSERT, UPDATE i DELETE nekog kupca? Obrišite pogled.
14. Napravite pogled koji će dohvaćati ime i prezime kupca te sve podatke o gradu.
  - a. Možete li napraviti INSERT grada kroz pogled. Vidi li se kroz pogled?
  - b. Možete li napraviti UPDATE ili DELETE grada kroz pogled?
  - c. Obrišite pogled.



# Primjeri

15. Napravite pogled koji će dohvaćati sve kupce iz Sarajeva. Pomoću tog pogleda umetnite kupca iz Zagreba. Vidi li se kupac kroz pogled?
16. Promijenite pogled tako da ne dopušta umetanje/izmjenu redaka koji neće biti vidljivi kroz njega. Probajte umetnuti novog kupca.
17. Obrišite pogled.

# Dodatne opcije pogleda

# Ovisnosti pogleda

- Izradom pogleda se **ne uspostavljaju čvrste veze** na tablice koje pogled koristi
  - Promjena strukture tablica može pogled učiniti neispravnim
- Moguće zahtijevati od RDBMS-a da napravi čvrste veze
  - Korištenjem opcije **WITH SCHEMABINDING**  
`CREATE VIEW naziv_pogleda`  
`WITH SCHEMABINDING`  
`AS`  
`SELECT_naredba`
  - Dodatna ograničenja:
    - Prilikom navođenja stupaca ne možemo koristiti \*
    - Uz nazive tablica moramo navesti i shemu

# Zaštita pogleda

- Sadržaj pogleda je moguće vidjeti kroz GUI ili pomoću sistemske procedure **sp\_helptext**

```
EXECUTE sp_helptext naziv_pogleda
```

- Moguće je sadržaj pogleda zaštititi:

- Korištenjem opcije **WITH ENCRYPTION**

```
CREATE VIEW naziv_pogleda
```

```
WITH ENCRYPTION
```

```
AS
```

```
SELECT_naredba
```

- U slučaju korištenja zaštite potrebno čuvati originalnu definiciju pogleda (najbolje u SQL skripti)
- Nije prava enkripcija – kao ključ se koristi GUID baze i ID objekta

# Postavljanje više opcija

- Moguće opcije:
  - WITH SCHEMABINDING i WITH ENCRYPTION dolaze prije AS
  - WITH CHECK OPTION dolazi na kraju
- Moguća je bila koja kombinacija navedenih opcija
- Ako navodimo obje opcije prije AS, pišemo jedan WITH i opcije navodimo odvojene zarezima, primjerice:

```
CREATE VIEW vw1  
WITH SCHEMABINDING, ENCRYPTION  
...
```
- Redoslijed navođenja opcija nije bitan



# Primjeri

18. Napravite tablicu Osoba sa stupcima IDOsoba, Ime, Prezime, OdjelID i Placa i umetnite nekoliko redaka.
- a. Napravite pogled koji dohvaća sve iz tablice
  - b. Iskoristite pogled za dohvaćanje podataka
  - c. Uklonite stupac OdjelID iz tablice
  - d. Možete li iskoristiti pogled za dohvaćanje podataka?
  - e. Promijenite definiciju pogleda tako da bude čvrsto vezan uz objekte koje koristi
  - f. Uklonite stupac Placa iz tablice
  - g. Uklonite pogled

# Primjeri

19. Napravite pogled koji dohvaća ime i prezime kupca, te naziv države.

1. Pogledajte SELECT upit pogleda kroz sučelje i pomoću sistemske procedure sp\_helptext
2. Zaštitite pogled
3. Pogledajte SELECT upit pogleda kroz sučelje i pomoću sistemske procedure sp\_helptext
4. Promijenite pogled tako da bude zaštićen i čvrsto vezan uz tablice
5. Promijenite pogled tako da bude zaštićen, čvrsto vezan uz tablice i da ne dopušta izmjene koje neće biti vidljive kroz pogled
6. Uklonite pogled