

## БАЗЕ ПОДАТАКА

### У следећим задацима заокружите број испред траженог одговора

<p>208. Међу понуђеним ентитетима, одредити ентитет са атрибутима који <b>НИСУ</b> одговарајући. Заокружити број испред траженог одговора:</p> <ol style="list-style-type: none"> <li>1. Ентитет: СТУДЕНТ – Атрибути: име, презиме, смер, број бодова, просек</li> <li>2. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина</li> <li>3. Ентитет: КЊИГА – Атрибути: наслов, аутор, година издања, издавач, адреса издавача, телефон издавача</li> <li>4. Ентитет: АВИОН – Атрибути: произвођач, марка, година производње, број седишта</li> </ol>	1
<p>209. Заокружите тачан исказ:</p> <ol style="list-style-type: none"> <li>1. Примарни кључ је атрибут који мора бити целобројног типа.</li> <li>2. Примарни кључ је атрибут који указује на зависност од неке друге табеле.</li> <li>3. Примарни кључ је атрибут који јединствено идентификује врсте у табели.</li> <li>4. Ако табела садржи вишевердносни атрибут, њему се додељује функција примарног кључа.</li> </ol>	1
<p>210. Заокружити број испред траженог одговора.</p> <p>Одредити оператор који би требало користити у WHERE клаузули SELECT наредбе да би били приказани само они ученици чије презиме почиње словом А:</p> <ol style="list-style-type: none"> <li>1. IN</li> <li>2. LIKE</li> <li>3. BETWEEN</li> <li>4. IS LIKE</li> <li>5. BEGINS WITH</li> </ol>	1
<p>211. Заокружити број испред траженог одговора.</p> <p>Табела UCENICI поред осталих података, садржи и вредност стипендије. Одредити оператор који треба употребити у WHERE клаузули SELECT наредбе да би били приказани сви ученици код којих није позната и није унета вредност у колону <b>stipendija</b>:</p> <ol style="list-style-type: none"> <li>1. =NULL</li> <li>2. ISNULL</li> <li>3. ==NULL</li> <li>4. IS NULL</li> <li>5. LIKE NULL</li> </ol>	1

<p>212. Дат је упит:</p> <pre>SELECT * FROM ucenici WHERE odeljenje=4 OR odeljenje=7 OR odeljenje=10</pre> <p>Заокружити оператор који треба користити у датом упиту да би се избегло вишеструко коришћење оператора OR:</p> <ol style="list-style-type: none"> <li>1. LIKE</li> <li>2. BETWEEN</li> <li>3. AND</li> <li>4. IN</li> </ol>	1
<p>213. Заокруживањем редног броја обележити клаузулу коју је потребно користи уколико листа иза резервисане речи SELECT садржи агрегатну функцију и једну или више колона које нису део агрегатне функције:</p> <ol style="list-style-type: none"> <li>1. HAVING клаузулу</li> <li>2. GROUP BY клаузулу</li> <li>3. JOIN клаузулу</li> <li>4. ORDER BY клаузулу</li> </ol>	1
<p>214. Заокружити тачан исказ:</p> <ol style="list-style-type: none"> <li>1. Кардиналност неке везе представља однос броја објеката који се повезују.</li> <li>2. Кардиналност неке везе представља апстракцију у којој се скуп сличних типова објеката представља општим генеричким типом (надтипом).</li> <li>3. Кардиналност неке везе одређује опционалност учешћа у вези.</li> <li>4. Кардиналност показује колико кандидата за примарни кључ има неки тип ентитета.</li> </ol>	2
<p>215. Дата је табела <i>PROJEKAT</i> над којом се извршава упит:</p> <pre>ALTER TABLE PROJEKAT ADD RokKraj date</pre> <p>Одредити шта ће се десити након извршења упита. Заокружити број испред траженог одговора.</p> <ol style="list-style-type: none"> <li>1. у табелу PROJEKAT додаје се ограничење RokKraj</li> <li>2. у табелу PROJEKAT додаје се колона RokKraj</li> <li>3. у табели PROJEKAT биће преименована колона</li> <li>4. у базу података додаје се табела PROJEKAT са само једном колоном</li> <li>5. у табели PROJEKAT промениће се тип података у колони RokKraj</li> </ol>	2

216. Дата је табела **RADNIK** и упит:

IDBR	IME	PREZIME	PLATA	PREMIJA	DATZAP
6234	Marko	Pavlović	1300	3000	1990-12-17
6789	Janko	Nikolić	3900	10	1999-12-23

```
SELECT Ime, Prezime, DATEDIFF(year, DatZap, GETDATE()) AS God FROM Radnik
```

Одредити шта је резултат упита. Заокружити број испред траженог одговора:

1. Табела са подацима о именима и презименима радника
2. Табела са подацима о именима, презименима и броју година које су протекле од датума запослења радника до краја века
3. Табела са подацима о именима, презименима и датумима запослења радника
4. Табела са подацима о именима, презименима и броју година које су протекле од датума запослења радника до тренутног датума

2

217. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Sloboda n	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	IMEOD	MESTO
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

```
SELECT Imeod, AVG(Plata) AS ProsekPlata FROM Radnik, Odeljenje  
WHERE Odeljenje.Brod=Radnik.Brod GROUP BY Imeod HAVING  
AVG(Plata)>1000
```

Одредити резултат извршавања датог упита. Заокружити број испред траженог одговора:

1. Упит се не извршава зато што груписање мора да се изврши не само по називу одељења, него и по шифри одељења (BrOd)
2. Групишу по одељењима радници са платом већом од просечне плате
3. Приказују називи одељења и висина просечне плате у њима само за одељења у којима је просечна плата већа од 1000
4. Приказују називи одељења и висина просечне плате у њима, при чему се код одређивања просека узимају у обзир само плате веће од 1000

2

218. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD	BROD	IMEOD	MESTO
5900	Slobodan	Golubović	900	10	50	Skladišta	Zemun
5932	Mitar	Gavrilović	600		30	Marketing	Vračar
6234	Marko	Pavlović	1300	30	10	Plasman	Surčin
6789	Janko	Nikolić	800	10	20	Direkcija	Grocka

```
SELECT Odeljenje.Imeod, Radnik.Ime+ ' ' + Radnik.Prezme as PunoIme
FROM Odeljenje LEFT JOIN Radnik ON Radnik.Brod = Odeljenje.Brod
```

Одредити шта се види као резултат датог упита. Заокружити број испред траженог одговора:

1. Називи свих одељења – и оних у којима има радника и оних где нико није распоређен - са бројем радника у сваком одељењу
2. За сваког распоређеног радника приказује се по један ред са називом одељења и пуним именом радника, док се за раднике који нису распоређени приказује само пуно име радника
3. Приказује се по један ред за сваког распоређеног радника са називом одељења и пуним именом радника. За раднике који нису распоређени, као и за одељења у која нико није распоређен, не формирају се редови у резултујућој табели
4. За свако одељење се приказује онолико редова колико радника ради у том одељењу, док се за одељења у којима нико не ради приказује по један ред са називом одељења

2

219. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD	BROD	IMEOD	MESTO
5900	Slobodan	Golubović	900	10	50	Skladišta	Zemun
5932	Mitar	Gavrilović	600	10	30	Marketing	Vračar
5953	Persida	Kosanović	1100	20	10	Plasman	Surčin
6234	Marko	Pavlović	1300	30	20	Direkcija	Grocka
6789	Janko	Nikolić	800	10	40	Nabavka	Barajevo

```
SELECT Odeljenje.Brod, Odeljenje.Imeod, COUNT(*)
FROM Radnik INNER JOIN Odeljenje ON Radnik.Brod = Odeljenje.Brod
GROUP BY Odeljenje.Brod, Odeljenje.Imeod
```

Одредити приказ који је резултат датог упита. Заокружити број испред траженог одговора:

1. Бројева и назива свих одељења
2. Бројева и назива свих одељења са бројем радника у њима
3. Бројева и назива одељења у којима има радника са бројем радника у њима
4. Бројева и назива одељења у којима нема радника

2

220. Извршава се следећа SELECT наредба:

```
SELECT MIN(Datum_Zaposlenja), Odsek_Id
FROM Zaposleni
GROUP BY Odsek_Id
```

Заокруживањем броја испред одговарајућег исказа, одредити које ће вредности бити приказане:

1. Најранији датум запослења за сваки одсек предузећа.
2. Најранији датум запослења у целом предузећу.
3. Датум запослења последњег запосленог радника у целом предузећу.
4. Датум запослења последњег запосленог радника за сваки одсек.
5. Датум запослења најстаријег запосленог радника у сваком одсеку предузећа.

2

221. Потребно је креирати извештај који приказује имена свих производа чија је цена већа од просечне цене свих производа.

Заокружити број испред упита који одговара постављеном задатку:

1. 

```
SELECT naziv
FROM proizvod
WHERE cena > (SELECT AVG(cena) FROM proizvod)
```
2. 

```
SELECT naziv
FROM proizvod
WHERE cena > AVG(cena)
```
3. 

```
SELECT naziv
FROM proizvod
GROUP BY naziv
HAVING cena > AVG(cena)
```
4. 

```
SELECT naziv
FROM (SELECT AVG(cena) FROM proizvod)
WHERE cena > AVG(cena)
```

2

222. Извршава се упит:

```
SELECT prezime, ime, email
FROM ucenik
ORDER BY prezime
WHERE prosek >= 4.50
```

Наредба се неће извршити. Заокружити број испред разлога услед кога се наредба неће извршити:

1. Наредба се неће извршити једино ако нема ни једног одличног ученика.
2. Услов треба написати у HAVING клаузули
3. Потребно је назначити редослед сортирања (asc, desc).
4. Потребно је променити редослед клаузула.

2

223. Табела **ARTIKLI** садржи следеће колоне: *artikl\_id*, *naziv*, *kategorija*, *cena*, *kolicina*.

Потребно је да се прикаже категорија и минимална цена артикла у свакој категорији. При томе се тражи приказ само оних категорија где је најмања цена производа већа од задате граничне вредности која се преноси упиту кроз параметар *@granica*

Изабрати упит који даје тражени извештај:

1. 

```
SELECT kategorija, MIN(cena)
FROM artikli
WHERE MIN(cena)>@granica
GROUP BY cena
```
2. 

```
SELECT kategorija, MIN(cena)
FROM artikli
GROUP BY kategorija
HAVING MIN(cena)>@granica
```
3. 

```
SELECT kategorija, MIN(cena)
FROM artikli
GROUP BY MIN(cena), kategorija
HAVING MIN(cena)>@granica
```
4. 

```
SELECT kategorija, MIN(cena)
FROM artikli
WHERE MIN(cena)>@granica
GROUP BY kategorija
```

2

224. Табела **RADIONICA** садржи следеће колоне: *radionica\_id*, *naziv*, *zanat*, *lokacija\_id*.

Потребно је да се прикаже колико на свакој локацији има различитих заната. Заокружити број испред упита који даје тражени извештај:

1. 

```
SELECT DISTINCT location_id, COUNT(zanat)
FROM radionica
GROUP BY lokacija_id
```
2. 

```
SELECT location_id, COUNT(zanat)
FROM radionica
GROUP BY lokacija_id
```
3. 

```
SELECT location_id, COUNT(DISTINCT zanat)
FROM radionica
GROUP BY lokacija_id
```
4. 

```
SELECT location_id, COUNT(DISTINCT zanat)
FROM radionica
GROUP BY zanat
```

2

**225.** Заокружити број испред одговора који представља наставак датог исказа:

Уколико поглед (view) треба користити за измену података у табели, поглед **НЕ СМЕ** садржати...

1. WHERE клаузулу
2. Спој више табела <https://www.w3resource.com/sql/update-views/sql-update-views.php>
3. Алијас колоне
4. GROUP BY клаузулу

2

226. Дат је упит за креирање погледа и наведени искази који се односе на дати упит. Заокружити број испред тачног исказа:	2
<pre>CREATE VIEW Pregled_Proseka AS SELECT UcenikID, Ime, Prezime, AVG(Ocena) AS Prosek FROM Testovi WHERE OdeljenjeID IN (1, 2, 3, 4) GROUP BY UcenikID, Ime, Prezime</pre>	
1. Подаци у табели <b>Testovi</b> се могу модификовати коришћењем погледа <b>Pregled_Proseka</b>	
2. Коришћењем датог погледа, подаци се могу само у додавати у табелу <b>Testovi</b> , али не и мењати	
3. Овако дат упит изазива грешку при извршењу	
4. Коришћењем датог погледа, подаци из табеле <b>Testovi</b> се могу само прегледавати, али не и додавати или мењати	
<b>У следећим задацима заокружите бројеве испред тражених одговора</b>	
227. Заокружити бројеве испред тражених одговора. Међу понуђеним алатима, обележити <b>Case</b> алате:	1,5
1. Rational Rose	
2. Oracle Designer	
3. .NET	
4. Microsoft Visio	
5. Java	
6. SQL Express	
228. Обележити команде које се сматрају командама ажурирања података у бази података. Заокружити бројеве испред тражених одговора:	1,5
1. Организовање податка	
2. Додавање нових података	
3. Брисање старих података	
4. Враћање оштећених података у коректно стање	
5. Измена постојећих података	
6. Додела права приступа подацима	
7. Измена структуре постојећих табела у бази	
229. Заокружити бројеве испред тражених одговора. За упите са специфицираним редоследом приказа врста у резултујућој табели користи се клаузула ORDER BY после које се наводи назив колоне:	1,5
1. и службена реч ASC за растући редослед	
2. и службена реч DESC за опадајући редослед	
3. и службена реч ASC за опадајући редослед	
4. и службена реч DESC за растући редослед	
5. службена реч се може изоставити при чему се добија растући поредак	
6. службена реч се може изоставити при чему се добија опадајући поредак	

230	<p>Заокружити бројеве испред наредби које служе за креирање, брисање и измену структуре релационе базе и објеката који чине релациону базу:</p> <ol style="list-style-type: none"> <li>1. ALTER TABLE</li> <li>2. INSERT</li> <li>3. CREATE TABLE</li> <li>4. DROP TABLE</li> <li>5. UPDATE</li> <li>6. DELETE TABLE</li> <li>7. ADD COLUMN</li> <li>8. ADD CONSTRAINT</li> </ol>	1,5
231	<p>Заокружити бројеве испред тражених одговора.</p> <p>Одредити ентитете који садрже одговарајуће атрибуте:</p> <ol style="list-style-type: none"> <li>1. Ентитет: КЊИГА – Атрибути: наслов, аутор, издавач, година издања</li> <li>2. Ентитет: АУТОМОБИЛ – Атрибути: марка, година производње, боја, власник, година рођења власника, регистарски број</li> <li>3. Ентитет: УЧЕНИК – Атрибути: име, презиме, разред, одељење, број оправданих, број неоправданих, просек</li> <li>4. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина, главни град, број становника главног града, име градоначелника главног града</li> </ol>	2
232	<p>Дати су искази који се односе на спољашњи кључ табеле (<i>foreign key constraint</i>). Заокружити бројеве испред <b>тачних</b> исказа:</p> <ol style="list-style-type: none"> <li>1. Вредност у пољу спољашњег кључа не сме бити NULL</li> <li>2. Вредност спољашњег кључа мора бити јединствена (unique) у колони над којом је постављено ограничење спољашњег кључа</li> <li>3. Вредност у пољу спољашњег кључа мора бити или NULL или једнака некој од вредности из колоне на коју спољашњи кључ референцира</li> <li>4. Више редова у табели може садржати исту вредност у пољу спољашњег кључа и тиме показивати на исти ред у референцираној табели</li> <li>5. Колона спољашњег кључа не мора садржати исти тип података као колона на коју спољашњи кључ референцира</li> </ol>	2
233	<p>Заокружити бројеве испред тражених одговора.</p> <p>Означити операторе који се <b>НЕ МОГУ</b> користити за поређење са подупитом који враћа више вредности:</p> <ol style="list-style-type: none"> <li>1. ALL</li> <li>2. ANY</li> <li>3. BETWEEN</li> <li>4. SOME</li> <li>5. LIKE</li> <li>6. IN</li> </ol>	2



234 Дата је табела **Kupci** са структуром:

```
( Id int primary key, Prezime varchar(50), Adresa varchar(50), Mesto varchar(20), Telefon varchar(5), Status varchar(8) )
```

И табела **NoviKupci** са структуром:

```
( Id int primary key, Prezime varchar(50), Telefon varchar(20), Status varchar(8) )
```

Извршава се упит:

```
INSERT INTO NoviKupci  
SELECT * FROM Kupci WHERE Status <> 'Aktivan'
```

Одредити шта је резултат извршења датог упита. Заокружити број испред траженог одговора:

1. Како табела **NoviKupci** има све колоне које постоје и у табели **Kupci**, упит се извршава без грешке и у табелу **NoviKupci** се уписују записи из табеле **Kupci** са статусом који није **Aktivan**
2. Упит се не извршава, пријављује грешку јер се број колона у табели **Kupci** разликује од броја колона у табели **NoviKupci**
3. Упит би се извршио без грешке да су у SELECT клаузули подупита, уместо \* наведене све колоне табеле **Kupci** које имају своју одговарајућу колону у табели **NoviKupci**
4. Упит јавља грешку због покушаја уписа вредности у поље примарног кључа који је аутоматски, тј креира га сама база

235 Дата је табела **RADNIK**, табела **Odeljenje** и упит:

IDBR	IME	PREZIME	PLATA	SIFRAOD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	
5953	Persida	Kosanović	1100	20

SIFRAOD	IMEOD	MESTO
10	Marketing	Vračar
20	Direkcija	Grocka
30	Nabavka	Barajevo

Креирана је ускладиштена процедура са параметром @br int = NULL

Позивом процедуре извршава се упит:

```
UPDATE Radnik SET Radnik.SifraOD = 30  
WHERE Radnik.SifraOD=@br or @br IS NULL
```

Одредити који од понуђених исказа су тачни. Заокружити бројеве испред тражених одговора:

1. Сви радници који раде у одељењу са шифром једнакој вредности која је пренета кроз параметар @br, биће прераспоређени у одељење чија је шифра 30
2. Упит распоређује све нераспоређене раднике у одељење са шифром 30
3. Уколико се параметру @br не пренесе вредност, радници који су до тог момента били нераспоређени, биће распоређени у одељење са шифром 30
4. Уколико се параметру @br не пренесе вредност, СВИ радници ће бити прераспоређени у одељење чији је број 30

236 Табела **Zaposleni** садржи поља: Zaposleni\_Id, Ime, Prezime, Plata, Odsek\_Id.

Дат је упит:

```
SELECT Zaposleni_Id, Ime, Prezime
FROM Zaposleni
WHERE Plata=(SELECT MAX(Plata) FROM Zaposleni GROUP BY Odsek_Id)
```

Дати су искази који описују ефекат извршења упита. Заокружити бројеве испред ТАЧНИХ исказа:

1. Упит се не извршава зато што у подупиту није дозвољено коришћење групних функција
2. Упит се извршава без грешке и из сваког одељења бира и приказује податке о раднику који има највећу плату у том одељењу.
3. Упит се не извршава јер подупит враћа више од једне врсте, а коришћен је оператор за поређење са једном вредношћу.
4. Уколико би се изоставила GROUP BY клаузула, упит би се извршавао без грешке и приказао би једног или више радника са платом једнакој највећој плати (без обзира на одељење).
5. Како подупит садржи груписање, да би се цео упит извршио без грешке, потребно је услов са подупитом написати у HAVING уместо у WHERE клаузули

2

237 Заокружити бројеве под којима су наведене клаузуле SQL наредбе у којима се могу користити аритметичке операције:

1. SELECT
2. FROM
3. WHERE
4. ORDER BY

2

238 Извршава се следећи упит:

```
SELECT cena
FROM proizvod
WHERE cena IN (101,125,150,350)
AND (cena BETWEEN 125 AND 140 OR cena >150)
```

Одредити које две вредности може вратити ова наредба. Заокружити бројеве испред тражених вредности:

1. 101
2. 150
3. 125
4. 110
5. 350

2

239 Заокружити бројеве испред команди које се могу користити за ажурирање постојећих података у бази:

1. DELETE
2. MERGE
3. SELECT
4. UPDATE

zadatak 228

2

- 240 Креиране су табеле **SKOLA** и **OSNOVNASKOLA**, а затим су у табелу **SKOLA** уписани подаци извршавањем следећих наредби:

```
create table Skola(  
    skolaID int primary key, Naziv nvarchar(50), gradID int, tip  
    nvarchar(50) )  
create table OsnovnaSkola(  
    gimID int primary key, Naziv nvarchar(50), gradID int )  
  
insert into Skola values (101, 'Nikola Tesla', 20, 'srednja strucna')  
insert into Skola values (102, 'Dusko Radovic', 20, 'osnovna')  
insert into Skola values (103, 'Sveti Sava', 30, 'osnovna')  
insert into Skola values (104, 'Bora Stankovic', 20, 'gimnazija')
```

У табелу **OSNOVNASKOLA** треба уписати податке о основним школама преписивањем потребних вредности из табеле **SKOLA**. Заокружити бројеве испред упита који ће јавити грешку при извршењу:

1. `select * into OsnovneSkole  
from Skola where tip='osnovna'`
2. `insert into OsnovneSkole  
select * from Skola where tip='osnovna'`
3. `insert into OsnovneSkole(skolaID, Naziv)  
select s.skolaID, s.Naziv from Skola as s where tip='osnovna'`
4. `insert into OsnovneSkole  
select s.skolaID, s.Naziv, s.gradID from Skola as s where  
tip='osnovna'`

- 241 Креиране су и попуњене подацима табеле **Korisnik** и **Prijatelji**. Њихова структура и садржај приказани су на слици:

**Korisnik**

ID	IME	POL
1	Ana	NULL
2	Steva	m
3	Marta	z
4	Petra	z

**Prijatelji**

Korisnik1	Korisnik2
1	2
1	3
2	3

Извршавањем упита добија се табела са подацима.

```
select k.ime, COUNT(*) as [broj prijatelja]  
from Korisnik as k  
left join Prijatelji as p on p.korisnik1=k.id or p.korisnik2=k.id  
where k.pol='z'  
group by k.id, k.ime
```

Заокруживањем бројева испред понуђених одговора, обележити који од наведених података ће бити приказани у појединим редовима резултујуће табеле:

- |             |             |
|-------------|-------------|
| 1. Ana, 1   | 5. Marta, 1 |
| 2. Ana, 2   | 6. Marta, 2 |
| 3. Steva, 1 | 7. Petra, 0 |
| 4. Steva, 2 | 8. Petra, 1 |

242	<p>Заокружити бројеве испред кључних речи које се НЕ КОРИСТЕ за обележавање ограничења (<i>constraints</i>) у језику SQL:</p> <ol style="list-style-type: none"> <li>1. Foreign key</li> <li>2. Unique</li> <li>3. Distinct</li> <li>4. Check</li> <li>5. Convert</li> <li>6. Union</li> <li>7. Not Null</li> <li>8. Except</li> </ol>	2
243	<p>Одредити ентитете који садрже одговарајуће атрибуте. Заокружити бројеве испред тражених одговора:</p> <ol style="list-style-type: none"> <li>1. Ентитет: СТУДЕНТ – Атрибути: име, презиме, смер, број бодова, просек</li> <li>2. Ентитет: КЊИГА – Атрибути: наслов, аутор, година издања, издавач, адреса издавача, телефон издавача</li> <li>3. Ентитет: АВИОН – Атрибути: произвођач, марка, година производње, број седишта</li> <li>4. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина</li> <li>5. Ентитет: САЈАМ – Атрибути: назив, број излагача, покровитељ, адреса покровитеља, контакт особа покровитеља</li> <li>6. Ентитет: ТУРИСТИЧКА АГЕНЦИЈА – Атрибути: назив, адреса, година оснивања, власник, стручна квалификација власника, запослени, стручна квалификација запослених</li> </ol>	3
244	<p>Обележити ентитете код којих је извршен адекватан избор јединственог идентификатора. Заокружи бројеве испред тражених одговора:</p> <ol style="list-style-type: none"> <li>1. јединствени матични број грађанина (ЈМБГ) за ентитет ОСОБА</li> <li>2. датум рођења за ентитет ОСОБА</li> <li>3. ИСБН број за ентитет КЊИГА</li> <li>4. регистарска ознака за АУТОМОБИЛ</li> <li>5. дестинација за ентитет АРАНЖМАН</li> <li>6. режисер за ентитет ФИЛМ</li> </ol>	3
245	<p>Одредити тачан исказ о оператору ANY који се примењује са подупитом који враћа више вредности:</p> <ol style="list-style-type: none"> <li>1. Оператор ANY може да се користи испред кључне речи DISTINCT.</li> <li>2. Оператор ANY врши поређење са свим вредностима које враћа подупит и враћа TRUE ако све вредности подупита задовољавају услов</li> <li>3. Оператор ANY врши поређење са свим вредностима које враћа подупит и враћа TRUE ако било која од вредности подупита задовољава услов</li> <li>4. Оператору ANY може да претходи оператор LIKE или оператор IN.</li> <li>5. Оператору ANY мора да претходи оператор поређења (=, &lt;&gt;, &gt;, &gt;=, &lt;, &lt;=)</li> <li>6. Услов =ANY(скуп вредности) је еквивалентан услову IN (скуп вредности)</li> </ol>	3

- 246 Креирана је табела **SKOLA**, а затим су у њу уписани подаци извршавањем следећих наредби:

```
create table Skola( skolaID int primary key, Naziv varchar(50) )
insert into Skola values (101, 'Nikola Tesla')
insert into Skola values (102, 'Mihajlo Pupin')
insert into Skola values (103, 'ETS Zemun')
```

За дати упит, треба проценити сценарио који ће се десити и заокруживањем редних бројева испред исказа, означити могуће исходе:

```
select * into StrucneSkole from Skola
```

1. Креира се копија табеле Skola - нова табела под именом StrucneSkole исте структуре као и табела Skola и у њу се преписују сви подаци из табеле Skola
2. Уколико табела са именом StrucneSkole постоји у бази, креира се нова са именом StrucneSkole(1) и у њу се преписују сви редови из табеле Skola
3. Уколико табела са именом StrucneSkole постоји у бази, не креира се нова, само се у постојећу преписују редови из табеле Skola
4. Уколико табела са именом StrucneSkole постоји у бази, упит јавља грешку
5. Уколико се дода услов **where** 1=2 упит се извршава, креира се нова табела исте структуре као и табела Skola, али се у њу не уписује ни један ред
6. Уколико се дода услов **where** 1=2 упит јавља грешку јер је 1=2 увек нетачно тј. **False**
7. Упит јавља грешку јер се кључна реч **into** користи искључиво у комбинацији са **insert**

3

### Допуните следеће реченице и табеле

247. Допунити реченицу наводећи назив нормалне форме:

Уколико ни један атрибут релације није вишеверносни, нити композитни, тј. не може се раставити, кажемо да је релација у \_\_\_\_\_ **prvoj** \_\_\_\_\_ нормалној форми.

1

248. Допунити реченицу наводећи назив нормалне форме:

Уколико сви атрибути релације који нису део кључа зависе од сваког атрибута који је део кључа кажемо да је релација у \_\_\_\_\_ **drugoј** \_\_\_\_\_ нормалној форми.

1

249. Допунити реченицу наводећи назив нормалне форме:

Уколико сви некључни (споредни) атрибути релације не зависе од неког другог некључног атрибута, тј. ако не постоји транзитивна зависност било ког споредног атрибута од било ког кључа те релације, кажемо да је релација у \_\_\_\_\_ **trećoj** \_\_\_\_\_ нормалној форми.

1

**У следећим задацима уредите и повежите појмове према захтеву**

250. Написати на цртама испред логичких операција редне бројеве њихових приоритета:

- |                      |          |     |
|----------------------|----------|-----|
| 1. највиши приоритет | <u>3</u> | OR  |
| 2. средњи приоритет  | <u>1</u> | NOT |
| 3. најнижи приоритет | <u>2</u> | AND |

1,5

**251.** Дата је табела **GEOGRAFIJA** која поред осталих података садржи називе градова и држава (*Naziv* *nvarchar(50)*). У зависности од услова у **WHERE** клаузули, **SELECT** упитом се приказују географски појмови из табеле. Са леве стране су дати услови нумерисани бројевима од 1 до 5, а са десне групе градова.

Свакој групи градова придружити по један услов уносом редног броја којим је услов нумерисан на линију испред листе градова:

- |   |          |                            |
|---|----------|----------------------------|
| 1. <i>where</i> <i>Naziv</i> <i>like</i> 'L__ %'  | <u>3</u> | SIJERA LEONE, SVETA LUCIJA |
| 2. <i>where</i> <i>Naziv</i> <i>like</i> '__ %N%' | <u>5</u> | LA VALETA, LA KORUNJA      |
| 3. <i>where</i> <i>Naziv</i> <i>like</i> '% L%'   | <u>4</u> | EL RENO, LA KORUNJA        |
| 4. <i>where</i> <i>Naziv</i> <i>like</i> '_L%'    | <u>2</u> | EL SALVADOR, EL RENO       |
| 5. <i>where</i> <i>Naziv</i> <i>like</i> '__ %A'  | <u>1</u> | LAS VEGAS, LOS ANGELES     |

2,5

252. Исписати на цртама испред релација редни број под којим је наведена одговарајућа кардиналност везе:

- |          |          |                          |
|----------|----------|--------------------------|
| 1. 1 : 1 | <u>2</u> | ВЛАСНИК – БРОЈ ТЕЛЕФОНА  |
| 2. 1 : M | <u>3</u> | НАСТАВНИК – ПРЕДМЕТ      |
| 3. M : M | <u>1</u> | ОСОБА – ПАСОШ            |
|          | <u>2</u> | КУПАЦ – МОДЕЛ АУТОМОБИЛА |
|          | <u>2</u> | УТАКМИЦА – ГРАД ДОМАЋИН  |

2,5

253. Уписати редни број почев од 1 на линију испред резервисане речи тако да одговара редоследу навођења при формирању упита.

За формирање упита за издвајање дела података из табеле која се налази у оквиру базе података користе се клаузуле у следећем редоследу:

- |          |          |
|----------|----------|
| <u>4</u> | GROUP BY |
| <u>3</u> | WHERE    |
| <u>1</u> | SELECT   |
| <u>5</u> | ORDER BY |
| <u>2</u> | FROM     |

3

254. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD	BROD	IMEOD	MESTO
5900	Slobodan	Golubović	900	10	50	Skladišta	Zemun
5932	Mitar	Gavrilović	600		30	Marketing	Vračar
5953	Persida	Kosanović	1100	20	10	Plasman	Surčin
6234	Marko	Pavlović	1300	30	20	Direkcija	Grocka
6789	Janko	Nikolić	800	10	40	Nabavka	Barajevo

Повезати упите и њихова значења уписом броја упита на одговарајућу линију:

- 1 `SELECT odeljenje.imeod,  
radnik.prezime  
FROM odeljenje INNER JOIN  
radnik  
ON radnik.brod = odeljenje.brod` 3 Приказује све раднике (и који јесу и који нису распоређени у одељења) и само она одељења у којима има радника
- 2 `SELECT odeljenje.imeod,  
radnik.prezime  
FROM odeljenje LEFT JOIN radnik  
ON radnik.brod = odeljenje.brod` 1 Приказује само одељења у којима има радника и само раднике распоређене у одељењима
- 3 `SELECT odeljenje.imeod,  
radnik.prezime  
FROM odeljenje RIGHT JOIN  
radnik  
ON radnik.brod = odeljenje.brod` 2 Приказује сва одељења (и она у којима има и она у којима нема радника) и само оне раднике који су распоређени у одељења

3

255. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD	BROD	IMEOD	MESTO
5900	Slobodan	Golubović	900	10	50	Skladišta	Zemun
5932	Mitar	Gavrilović	600		30	Marketing	Vračar
5953	Persida	Kosanović	1100	20	10	Plasman	Surčin
6234	Marko	Pavlović	1300	30	20	Direkcija	Grocka
6789	Janko	Nikolić	800	10	40	Nabavka	Barajevo

Повезати упите и њихова значења уписом броја датог испред описа значења упита на одговарајућу линију:

- 4 `SELECT odeljenje.imeod,  
radnik.prezime  
FROM odeljenje LEFT JOIN radnik  
ON radnik.brod = odeljenje.brod  
WHERE radnik.brod IS NULL` 1. Приказује само раднике који нису распоређени у одељења
- 3 `SELECT odeljenje.imeod,  
radnik.prezime  
FROM odeljenje FULL JOIN radnik  
ON radnik.brod = odeljenje.brod` 2. Приказује све раднике (и који јесу и који нису распоређени у одељења) и само она одељења у којима има радника
- 1 `SELECT odeljenje.imeod,  
radnik.prezime  
FROM odeljenje RIGHT JOIN  
radnik  
ON radnik.brod = odeljenje.brod  
WHERE odeljenje.brod IS NULL` 3. Приказује сва одељења - и она у којима има и оне у којима нема радника и све раднике – и оне који су распоређени у одељења, као и оне који нису распоређени
4. Приказује само одељења у којима нема радника

3

256. На левој страни су наведене категорије SQL команди, а са десне су набројане команде. На линију испред команде уписати број под којим је наведена категорија којој команда припада:

1. DDL – Data Definition Language	<u>3</u>	GRANT
2. DML – Data Manipulation Language	<u>2</u>	UPDATE
3. DCL – Data Control Language	<u>4</u>	COMMIT
4. TCL – Transaction Control Language	<u>1</u>	DROP
	<u>2</u>	DELETE
	<u>1</u>	ALTER

3

257. Табела **ZAPOSLENI** је креирана и попуњена извршавањем следећих наредби:

```
create TABLE zaposleni(  
  id INTEGER NOT NULL PRIMARY KEY, rukovodilacId INTEGER, ime VARCHAR(30)  
  NOT NULL,  
  FOREIGN KEY (rukovodilacId) REFERENCES zaposleni(id))  
  
INSERT INTO zaposleni VALUES(1, NULL, 'Petar');  
INSERT INTO zaposleni VALUES(2, 1, 'Mihajlo');  
INSERT INTO zaposleni VALUES(3, 2, 'Milica');  
INSERT INTO zaposleni VALUES(4, 3, 'Lazar');  
INSERT INTO zaposleni VALUES(5, NULL, 'Sofija');
```

Очекивани ефекти извршења упита нумерисани су бројевима од 1 до 6. Уносом редног броја одговарајућег описа на предвиђену линију испред упита, повезати упит и опис резултата његовог извршења:

1. Сви радници који су руководиоци неком другом раднику
2. Сви радници који нису руководиоци ником
3. Сви радници који немају надређене руководиоце
4. Сви радници који имају надређеног руководиоца
5. Празна табела

5     `select * from zaposleni  
where id not in (select distinct rukovodilacId from zaposleni)`

2     `select * from zaposleni  
where id not in  
(select rukovodilacId from zaposleni where rukovodilacId is not null)`

3     `select * from zaposleni where rukovodilacId is null`

3