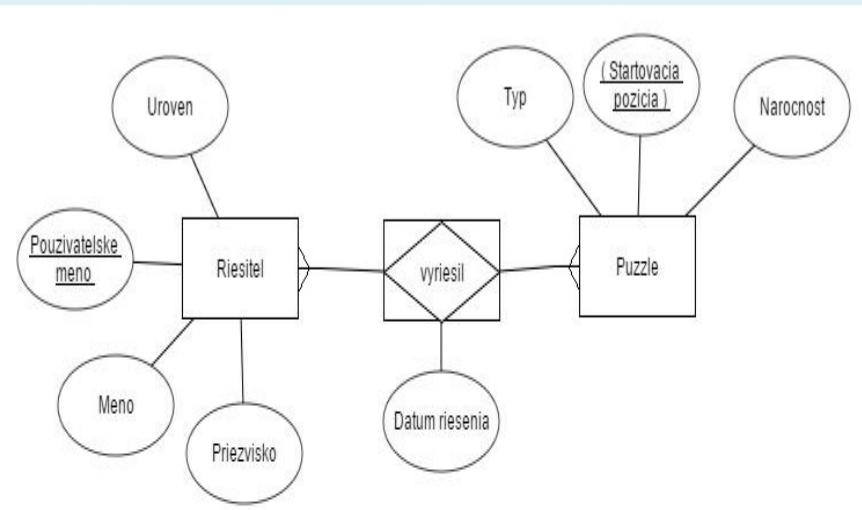


Máme ERD opisujúce databázu šachových hlavolamov ako je ukázané na nasledujúcom obrázku. Koľko stĺpcov bude mať v relačnom modeli tabuľka riešení?

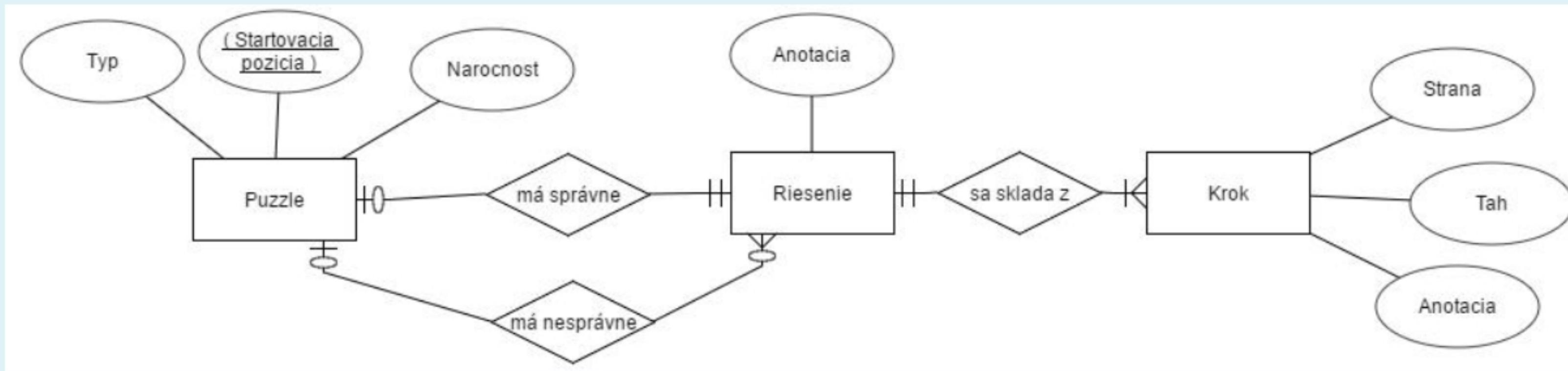


Označte jednu alebo viac odpovedí:

- a. 5
- b. 3 ✓
- c. 1
- d. 2
- e. 4

Skontrolovať

Nasledujúci jednoduchý ERD opisuje databázu šachových hlavolamov (puzzle). Ktorá/é z nasledujúcich možností platí pre prevod vztahu "má správne" do relačného modelu?



Označte jednu alebo viac odpovedí:

- a. vztah "má správne" môže byť v relačnom modele vyjadrený pridaním cudzieho klúča do tabuľky "Puzzle", odkazujúceho sa na tabuľku "Riesenie"
- b. vztah "má správne" bude v relačnom modele vyjadrený prepájacou medzitabuľkou "maRiesenie" s dvoma stĺpcami, ktoré budú cudzími klúčmi prepájajími tabuľky "Riesenie" a "Puzzle", pričom táto medzitabuľka bude zároveň vyjadrovať aj vztah "má nesprávne" z ERD
- c. vztah "má správne" môže byť v relačnom modele vyjadrený pridaním cudzieho klúča do tabuľky "Riesenie", odkazujúceho sa na tabuľku "Puzzle"
- d. vztah "má správne" bude v relačnom modele vyjadrený prepájacou medzitabuľkou "maSpravne" s dvoma stĺpcami, ktoré budú cudzími klúčmi prepájajími tabuľky "Riesenie" a "Puzzle"

Skontrolovať

Vyberte pravdivé tvrdenia o relačnom modeli:

Označte jednu alebo viac odpovedí:

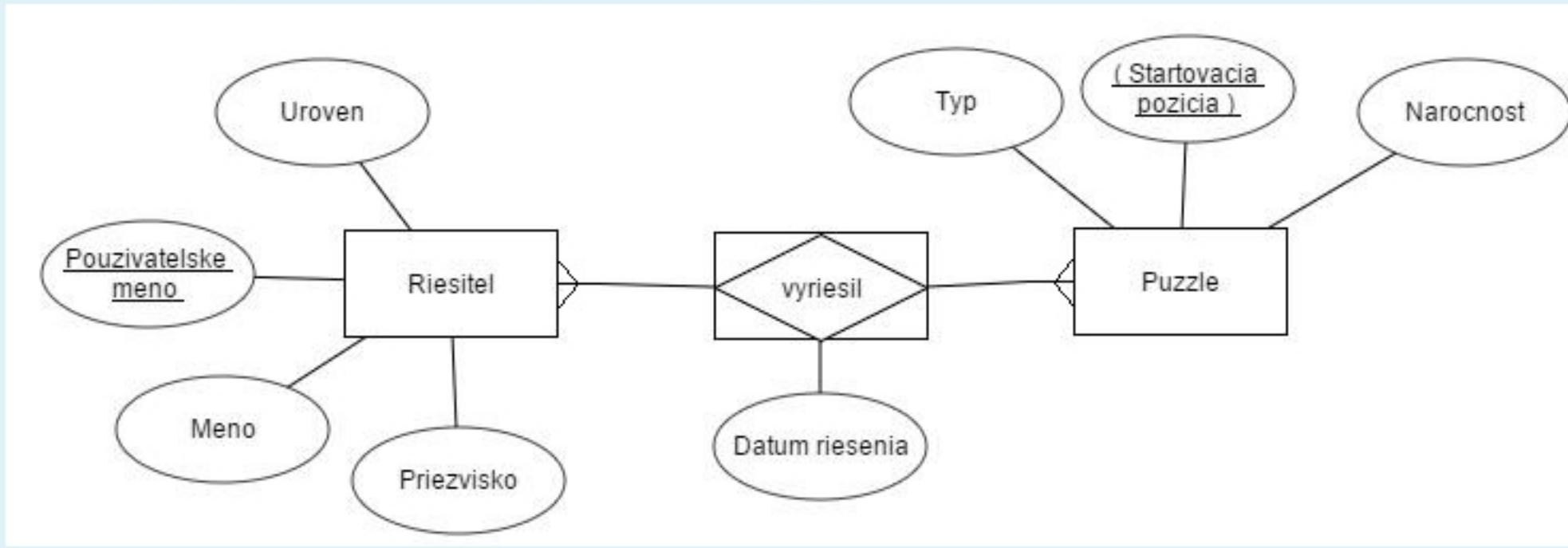
- a. Všetky hodnoty v danom zázname sú jednoznačne a plne závislé na primárnom kľúči záznamu ✓
- b. Na poradí stĺpcov v tabuľke záleží
- c. Pre daný záznam musí byť zadaná hodnota každého stĺpca
- d. Každá tabuľka má jednoznačný názov ✓
- e. Nie všetky stĺpce musia mať svoj názov
- f. Na poradí záznamov nezáleží ✓
- g. Každý stĺpec obsahuje hodnoty toho istého atribútu ✓

Skontrolovať

Správna odpoveď

Hodnotenie pre tento pokus: 2,00/2,00.

Máme ERD opisujúce databázu šachových hlavolamov ako je ukázané na nasledujúcom obrázku. Koľko stĺpcov bude mať v relačnom modele tabuľka pre asociatívnu entity "vyriesil"?



Označte jednu alebo viac odpovedí:

a. 5

b. 3

c. 1

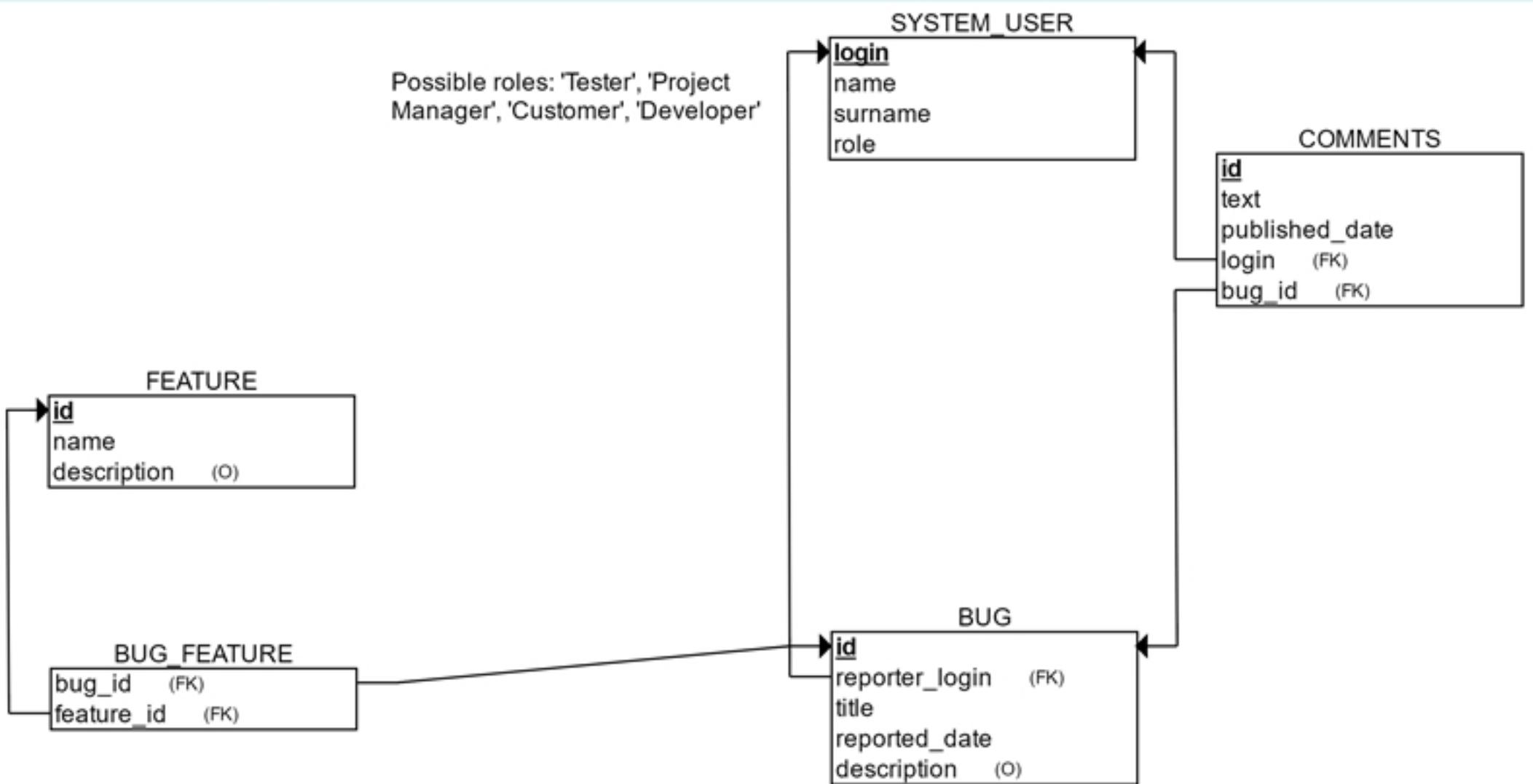
d. 2

e. 4

Skontrolovať

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti name (názov stĺpca "MENO") a surname (názov stĺpca "PRIEZVISKO") používateľov, ktorých login obsahuje c. Zoradťte výsledok abecedne podľa priezviska.

Vo vzorovej DB sú to 4 používatelia.



Napište operátor určený pre zlúčenie (zretazenie, sčítanie) retázcov

Odpoved:

||



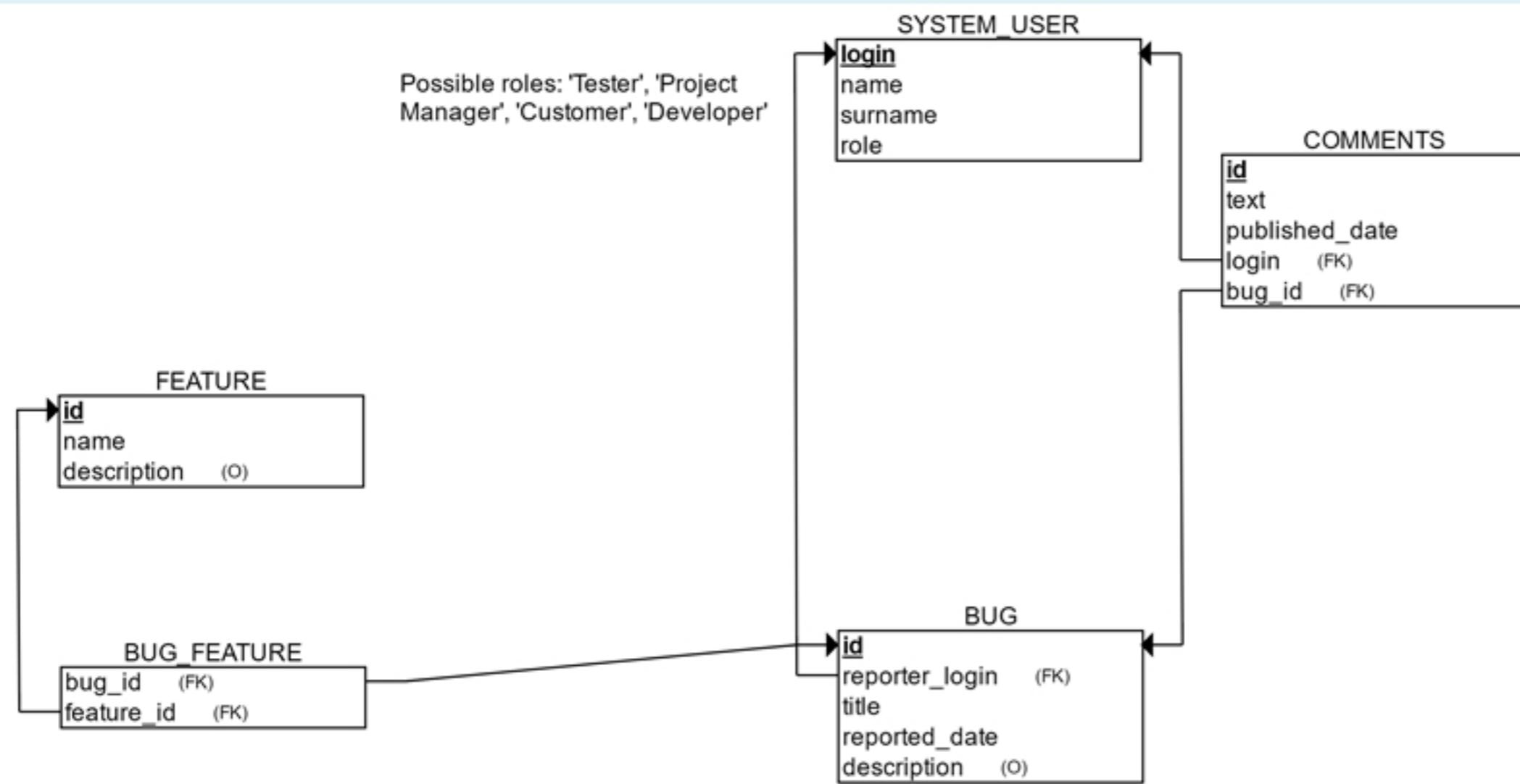
Skontrolovať

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti text komentáru (názov stĺpca "TEXT"), a počet dní (zaokrúhlený dole - floor), ako dlho je komentár dodnes zverejnený (názov stĺpca "VEK") - select má vrátiť všetky komentáre.

Vo vzorovej DB je vo výsledku napr. záznam:

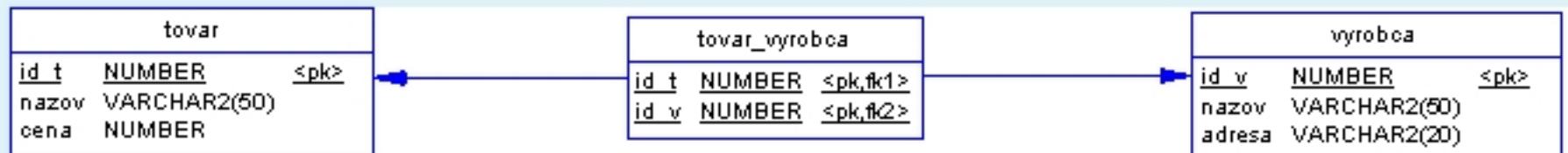
| "Skusal so mto na produkcnom prostredi s verziou 1, nefunguje to." | 769 |

(uvedený vek platí, ak by sme skript vykonali 15. mája 2016).



Majme databázu s reláciami podľa obrázku

Vyberte z možností dopyt, ktorý realizuje nasledovnú požiadavku: zistite, ktorí výrobcovia nevyrábajú tovar YYY



Označte jednu odpoved' :

- select nazov from vyrobca where id_v = all
(select tv.id_v from tovar_vyrobca tv, tovar t where tv.id_t = t.id_t and t.nazov like 'YYY');
- select nazov from vyrobca where id_v !=
(select tv.id_v from tovar_vyrobca tv, tovar t where tv.id_t = t.id_t and t.nazov like 'YYY');
- select nazov
from vyrobca
where id_v not in
(select tv.id_v from tovar_vyrobca tv, tovar t where tv.id_t = t.id_t and t.nazov = 'YYY');
- select nazov from vyrobca where id_v is not
(select tv.id_v from tovar_vyrobca tv, tovar t where tv.id_t = t.id_t and t.nazov like 'YYY');

Skontrolovať

Majme dopyt:

```
SELECT 'Zamestnanec: ' || e.firstname || e.lastname, 'Zakazník: ' || c.firstname || c.lastname  
FROM customer c JOIN employee e ON c.supportRepId = e.employeeId  
WHERE e.hiredate < '1.1.2010';
```

Ktoré z nasledujúcich indexov databázový systém **vie** využiť pre zefektívnenie jeho vykonávania?

Označte jednu alebo viac odpovedí:

- a. index na báze hešovacej tabuľky nad stĺpcom hiredate tabuľky employee
- b. index na báze vyváženého stromu nad stĺpcom hiredate tabuľky employee
- c. index na báze vyváženého stromu nad stĺpcom firstname tabuľky customer
- d. index na báze hešovacej tabuľky nad stĺpcom supportRepId tabuľky customer

Skontrolovať

Majme dopyt:

```
SELECT 'Zamestnanec: ' || e.firstname || e.lastname, 'Zakazník: ' || c.firstname || c.lastname  
FROM customer c JOIN employee e ON c.supportRepId = e.employeeId  
WHERE e.hiredate < '1.1.2010';
```

Ktoré z nasledujúcich indexov databázový systém **vie** využiť pre zefektívnenie jeho vykonávania?

Označte jednu alebo viac odpovedí:

- a. index na báze hešovacej tabuľky nad stĺpcom hiredate tabuľky employee
- b. index na báze vyváženého stromu nad stĺpcom hiredate tabuľky employee
- c. index na báze vyváženého stromu nad stĺpcom firstname tabuľky customer
- d. index na báze hešovacej tabuľky nad stĺpcom supportRepId tabuľky customer

Skontrolovať

Pracujeme s databázou pre manažment softvérových bugov, ktoréj relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti **text komentáru** (názov stĺpca "TEXT"), a **počet dní (zaokrúhlený dole - floor)**, ako dlho je komentár dodnes zverejnený (názov stĺpca "VEK") - select má vrátiť všetky komentáre.

Vo vzorovej DB je vo výsledku napr. záznam:

| "Skusal so mto na produkčnom prostredi s verziou 1, nefunguje to." | 769 |
(uvedený vek platí, ak by sme skript vykonali 15. mája 2016).

Majme dopyt:

```
SELECT name  
FROM track  
WHERE (unitPrice > 0.5 AND unitPrice <= 1.0)  
      OR composer = 'U2';
```

Ktorý z nasledujúcich indexov databázový systém **nie** využíť pre zefektívnenie jeho vykonávania?

Označte jednu odpoved' :

- a. index na báze hešovacej tabuľky nad stĺpcom unitPrice ✓
- b. index na báze vyváženého stromu nad stĺpcom unitPrice
- c. index na báze vyváženého stromu nad stĺpcom composer
- d. index na báze hešovacej tabuľky nad stĺpcom composer

[Clear my choice](#)

Skontrolovať

Správna odpoveď

Hodnotenie pre tento pokus: 2,00/2,00.

Vyberte pravdivé tvrdenia o NATURAL JOINe:

Označte jednu alebo viac odpovedí:

- a. Výsledkom je kombinácia záznamov z jednej tabuľky so záznamami druhej tabuľky, pre ktoré sú hodnoty v rovnako pomenovaných stĺpcach rovnaké ✓
- b. Je to karteziánsky súčin, kde každý záznam z jednej tabuľky je spojený s každým záznamom z druhej tabuľky
- c. Vyžaduje si, aby spájané tabuľky mali rovnakú relačnú schému, t.j. aby mali rovnaký počet stĺpcov a aby každý pricislúchajúci pári stĺpcov mal rovnaký (resp. kompatibilný) údajový typ
- d. Neodporúča sa používať, vhodnou náhradou je INNER JOIN s USING, kde sa explicitne vymenujú stĺpce s rovnakým názvom, ktoré sa majú použiť pri spájaní ✓

Skontrolovať

Správna odpoveď

Hodnotenie pre tento pokus: 2,00/2,00.

Vyberte pravdivé tvrdenia o príkazových triggeroch:

Označte jednu alebo viac odpovedí:

- a. vykonajú sa raz pre každý riadok ovplyvnený udalosťou
- b. sú to triggery, ktoré nemajú uvedenú klauzulu FOR EACH ROW ✓
- c. vykonajú sa raz pre príkaz, ktorý udalosť spustil ✓
- d. môžu sa použiť len na umožnenie modifikácie pohľadov, ktoré nie sú inherentne modifikovateľné

Skontrolovať

Správna odpoveď

Hodnotenie pre tento pokus: 2,00/2,00.

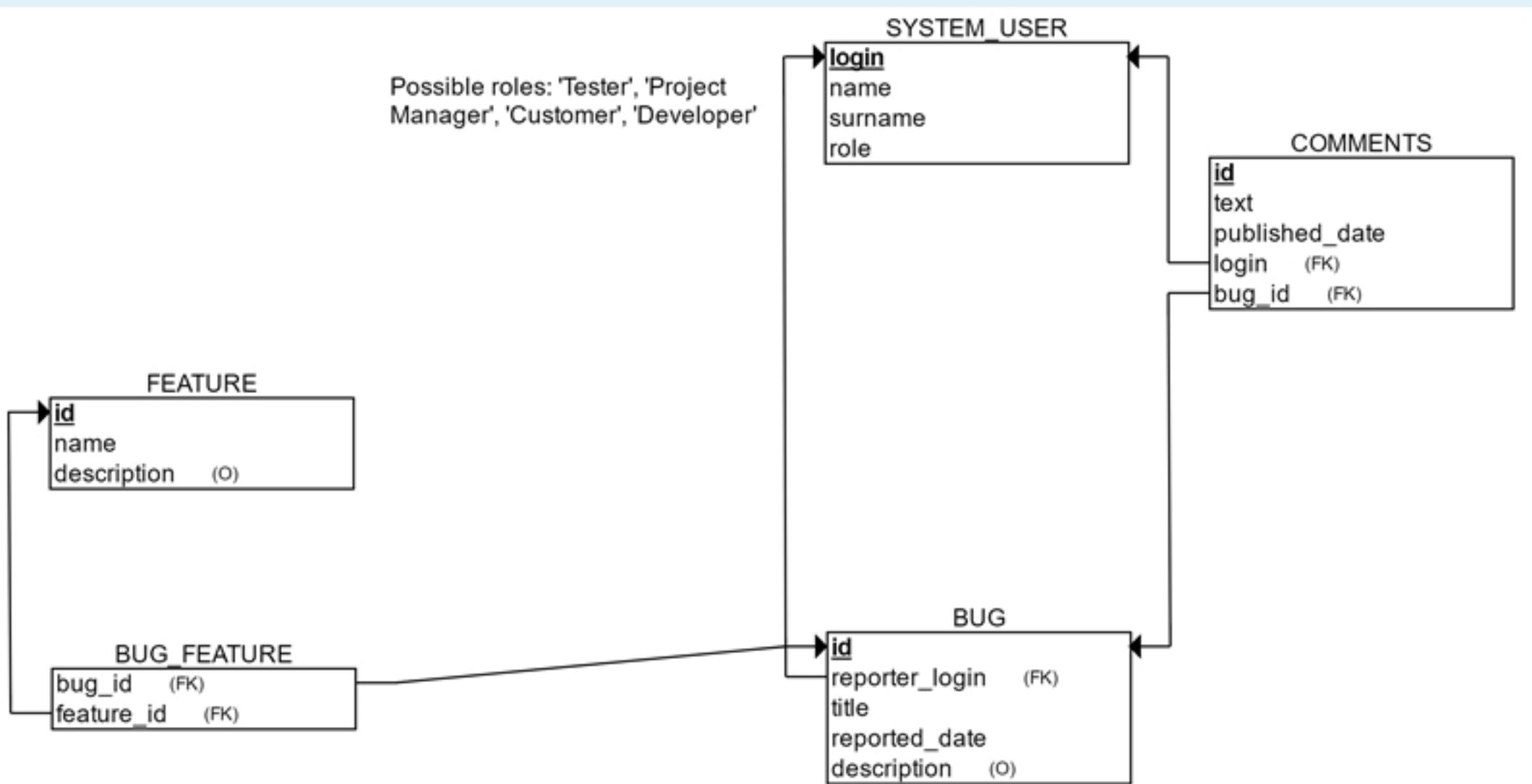
Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti zoznam názvov features (stípec name), ktoré nemajú v roku 2015 ohľásený žiadten bug.

Vo vzorovej DB sú to E-mail notifications, Social networks integration, Performance monitoring, HipChat notifikacie.

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti zoznam dátumov publikovania nejakého komentáru (stĺpec published_date) bez duplikátov, pričom zobrazte iba dátumy z roku 2016.

Vo vzorovej DB je takých dátumov 7.

Pozor, vrátený má byť typ *dátum*, nie dátum formátovaný ako reťazec.



Čo opisuje konceptuálna schéma databázy?

Označte jednu odpoveď:

- a. je to sada dopytov, ktoré vyjadrujú, čo daný program chce z databázy získať, prostredníctvom nej vieme povedať, aké práva pre prístup do databázy daný program potrebuje
- b. Je to konkrétny diagram, ktorý je pri návrhu databázy použitý (napr. entitno-relačný diagram)
- c. opisuje logickú štruktúru údajov a ich vzájomných väzieb a používa sa na komunikáciu medzi databázou a klientským programom (napr. relačná schéma v relačných databázach) ✓
- d. opisuje, ako sú údaje uložené na disku v údajových súboroch, ako sú vyjadrené ich vzájomné väzby a aké údajové typy sú použité

[Clear my choice](#)

Skontrolovať

Vyber pravdivé tvrdenia o entitnej integrite údajov v rámci relačného modelu:

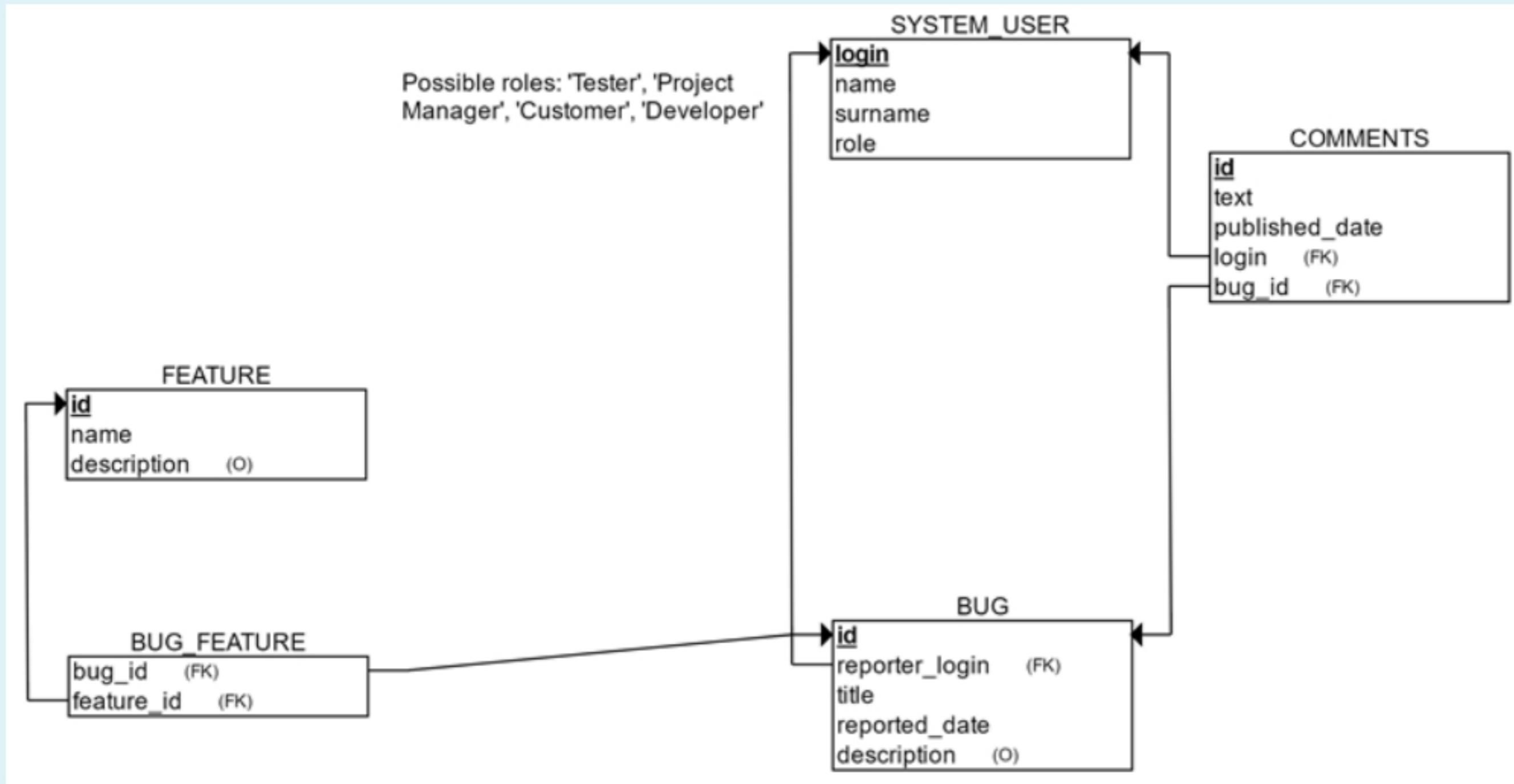
Označte jednu alebo viac odpovedí:

- a. primárny kľúč tabuľky musí byť vždy uvedený (nesmie mať hodnotu NULL)
- b. použitie viachodnotového atribútu (napr. titul s možnosťou uvedenia viacerých titulov v rámci jedného reťazca) porušuje entitnú integritu
- c. rôzne entity v tabuľke sú vždy rozlíšiteľné prinajmenšom primárnym kľúčom
- d. entitná integrita zabezpečuje jednoznačnú identifikateľnosť každej entity v databáze
- e. akékolvek neuvedenie povinného atribútu entity porušuje entitnú integritu databázy

Skontrolovať

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. **Napište SQL dopyt, ktorý vráti zoznam používateľov/používateľa (iba stĺpce name a surname v takom poradí) s najviac komentármí. Pozor, rátajte s tým, že používateľov s rovnakým meno a priezviskom môže byť viac.**

Vo vzorovej DB je to: Juan Robinson, a Julia Jackson.



V prípade potreby vyjadrenia dátumu dajte pozor na použitý formát, Váš lokálny formát nemusí fungovať na vyhodnocovacej databáze! Môžete použiť dátumový literál (napr. `date '2017-05-13'`) alebo funkciu `to_date`.

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napište SQL dopyt, ktorý vráti v jednom stĺpci mená dokopy s priezviskami používateľov. Výsledný stĺpec pomenujte "MENA" a zabezpečte, aby nemal duplikáty. Vo vzorovej DB bude výsledkom jeden stĺpec s nasledovnými 20 hodnotami (na usporiadanie nezáleží):

- Amy
- Campbell
- Clark
- Jackson
- Jason
- Jose
- Joshua
- Juan
- Julia
- Paul
- Rachel
- Robinson
- Rogers
- Ross
- Sanchez
- Sanders
- Sharon
- Shawn
- Stewart
- Wilson

Vyberte pravdivé tvrdenia o LEFT OUTER JOINe:

Označte jednu alebo viac odpovedí:

- a. Vo výsledku sa objaví každý záznam z ľavej tabuľky aspoň raz
- b. Výsledkom je kombinácia záznamov z jednej tabuľky so záznamami druhej tabuľky, pre ktoré platí podmienka v ON klauzule, a aj záznamy z ľavej tabuľky, ktoré nemajú pár splňajúci podmienku, doplnené o NULL hodnoty ✓
- c. Žiadny OUTER JOIN neumožňuje použiť USING klauzulu, pri OUTER JOINoch je povolená iba ON klauzula ✗
- d. Neodporúča sa používať, vhodnou náhradou je INNER JOIN s USING, kde sa explicitne vymenujú stĺpce s rovnakým názvom, ktoré sa majú použiť pri spájaní

Skontrolovať

NAVIGÁCIA V TESTE

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:55

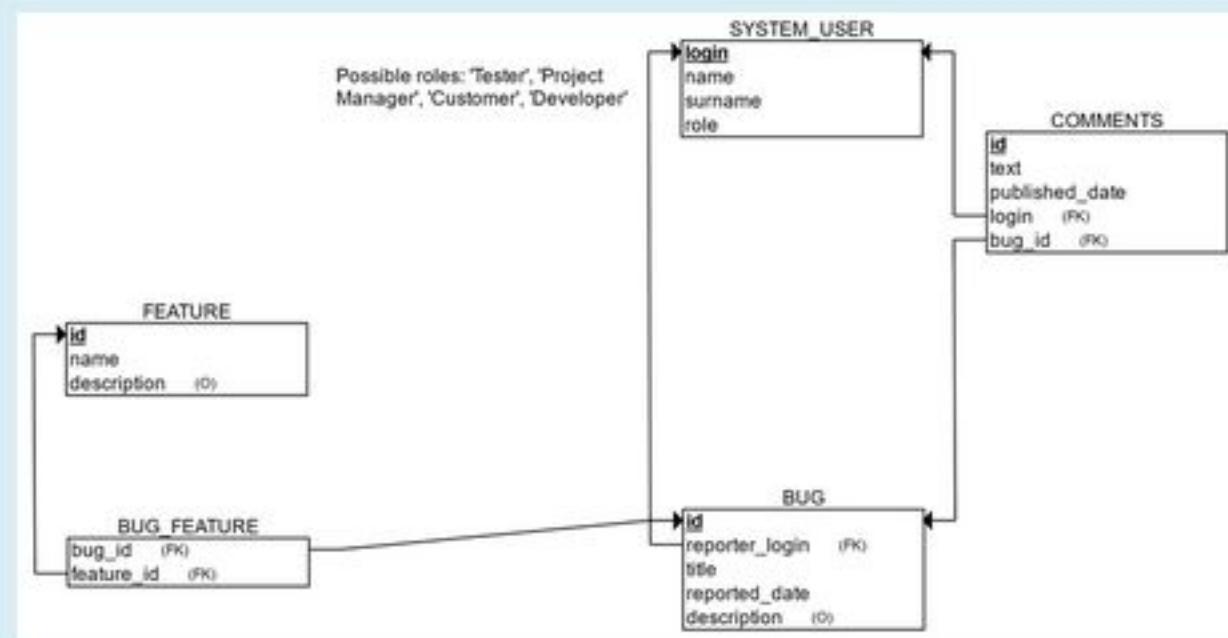
Otázka 12

Ešte
nezodpovedané
Max. hodnotenie
7,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktoréj relačný model je naznačený na nasledujúcej schéme.

Napište SQL dopyt, vráti celé záznamy komentárov, ktoré boli zverejnené v iný mesiac ako apríl (na roku nezáleží) zoradené podľa dátumu (od najstaršieho po najnovší).
Vo vzorovej DB sú to komentáre s id 7 a 10.



Formát dátumu v DB je 'DD.MM.YYYY', napr. '01.04.2016'.

Testovať svoj dopyt môžete na školskej databáze na nasledovnom pripojení (všetci to pripojenie zdieľate, preto vo vlastnom záujme v databáze necommitujte žiadne zmeny):

- hostname: oracle.kpi.fei.tuke.sk
- port: 1521
- SID: db11g
- connection name: exams_bugs
- login: exams_bugs
- password: abc

Odpoved:

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)

- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)

- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:58:09

Otázka 13

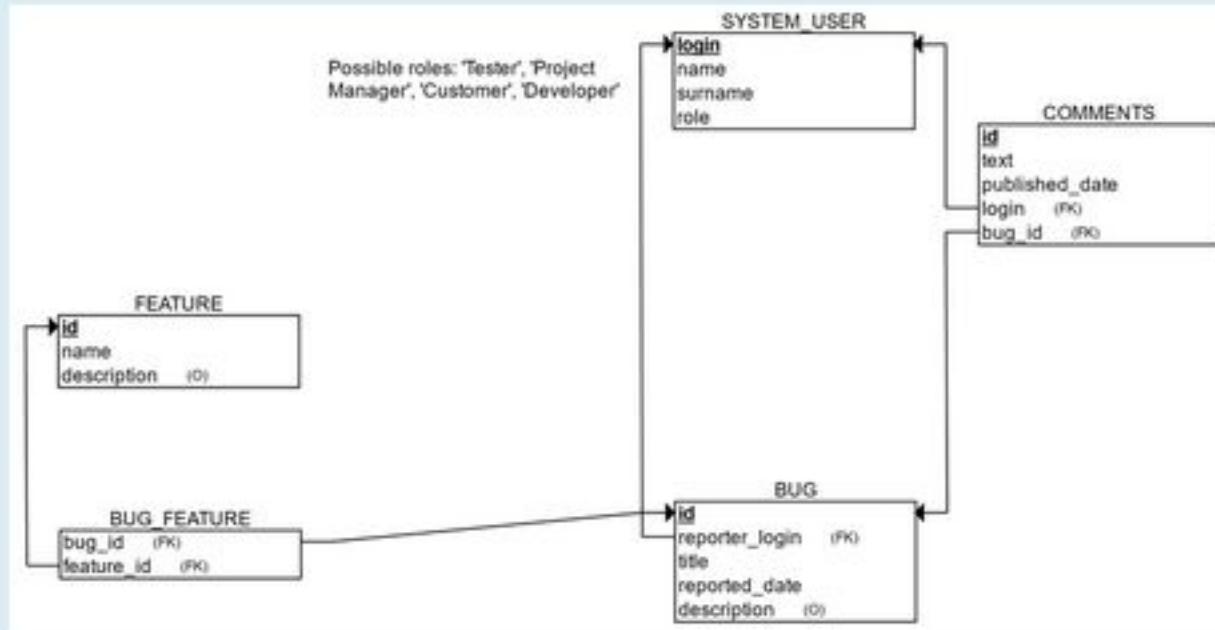
Ešte
nezodpovedané
Max. hodnotenie
7,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktoréj relačný model je naznačený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti v jednom stĺpci všetky dátumy, kedy bol publikovaný nejaký komentár, alebo ohlášený nejaký bug, ale tak, aby sa nedostránili duplikáty.

Vo vzorovej DB bude výsledkom jeden stĺpec s 18 dátumami (niektoré z nich budú duplicity).



Formát dátumu v DB je 'DD.MM.YYYY', napr. '01.04.2016'.

Testovať svoj dopyt môžete na školskej databáze na nasledovnom pripojení (všetci to pripojenie zdieľate, preto vo vlastnom záujme v databáze necommitujte žiadne zmeny):

- hostname: oracle.kpi.fei.tuke.sk
- port: 1521
- SID: db11g
- connection name: exams_bugs
- login: exams_bugs
- password: abc

NAVIGÁCIA V TESTE

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:49

Otázka 14

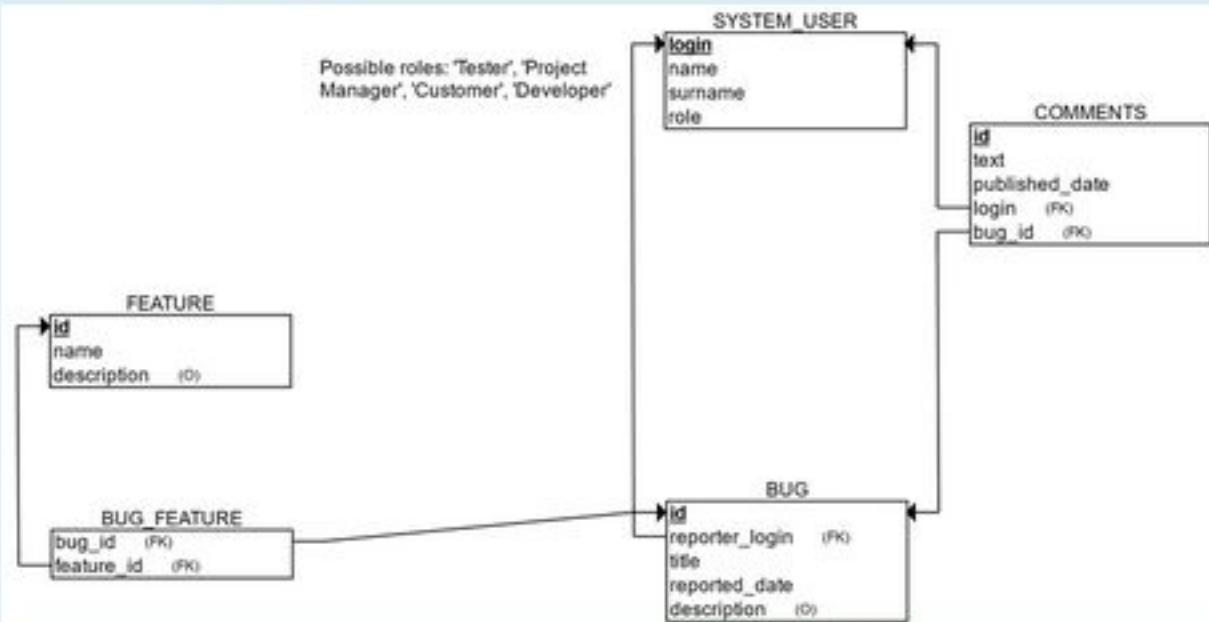
Ešte
nezodpovedané
Max. hodnotenie
9,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktoréj relačný model je naznačený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti meno a priezvisko autora najnovšieho komentáru, vo výsledku nech je jeden stĺpec s názvom "MENO" obsahujúci meno a priezvisko oddelené medzerou.

Vo vzorovej DB je to Rachel Clark.



Formát dátumu v DB je 'DD.MM.YYYY', napr. '01.04.2016'.

Testovať svoj dopyt môžete na školskej databáze na nasledovnom pripojení (všetci to pripojenie zdieľate, preto vo vlastnom záujme v databáze necommitujte žiadne zmeny):

- hostname: oracle.kpi.fei.tuke.sk
- port: 1521
- SID: db11g
- connection name: exams_bugs
- login: exams_bugs
- password: abc

NAVIGÁCIA V TESTE

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:44

Otázka 15

Ešte
nezodpovedané

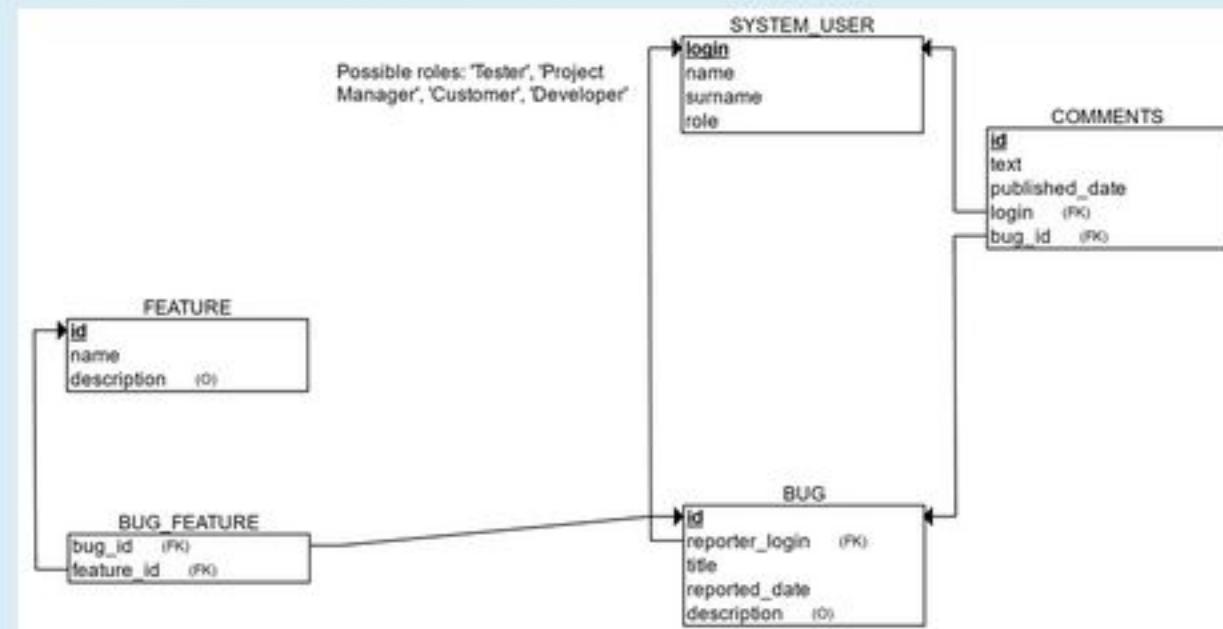
Max. hodnotenie
10,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti zoznam používateľov/používateľa (iba stĺpce name a surname v takom poradí) s najviac ohlásenými bugmi. Pozor, rátajte s tým, že používateľov s rovnakým menom a priezviskom môže byť viac.

Vo vzorovej DB je to iba: Julia Jackson



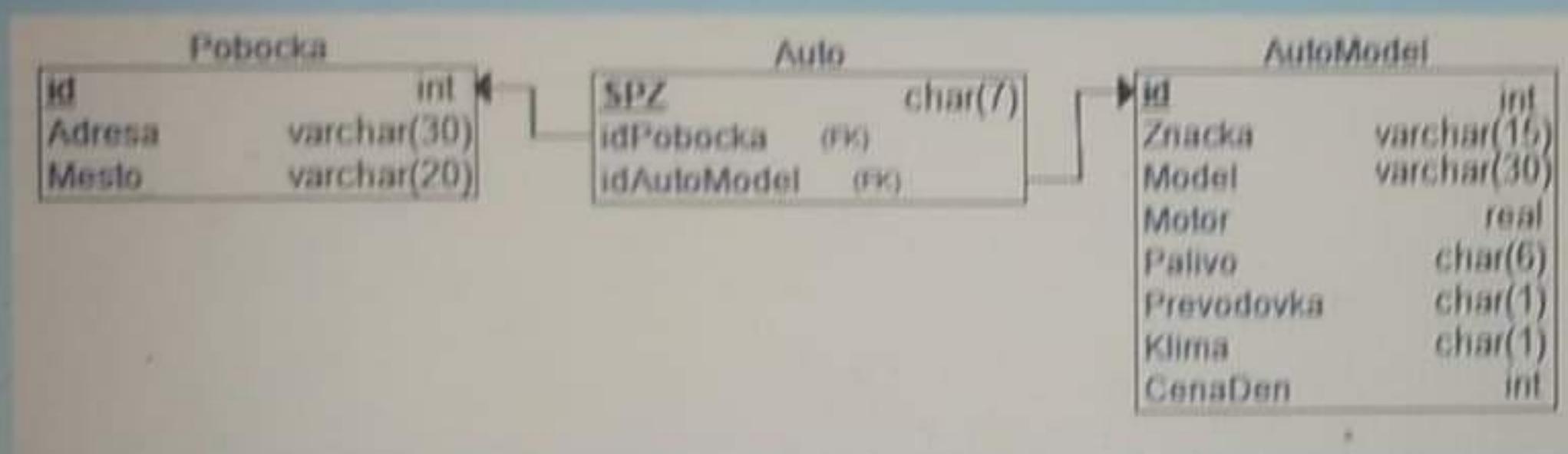
Formát dátumu v DB je 'DD.MM.YYYY', napr. '01.04.2016'.

Testovať svoj dopyt môžete na školskej databáze na nasledovnom pripojení (všetci to pripojenie zdieľate, preto vo vlastnom záujme v databáze necommitujte žiadne zmeny):

- hostname: oracle.kpi.fei.tuke.sk
- port: 1521
- SID: db11g
- connection name: exams_bugs
- login: exams_bugs
- password: abc

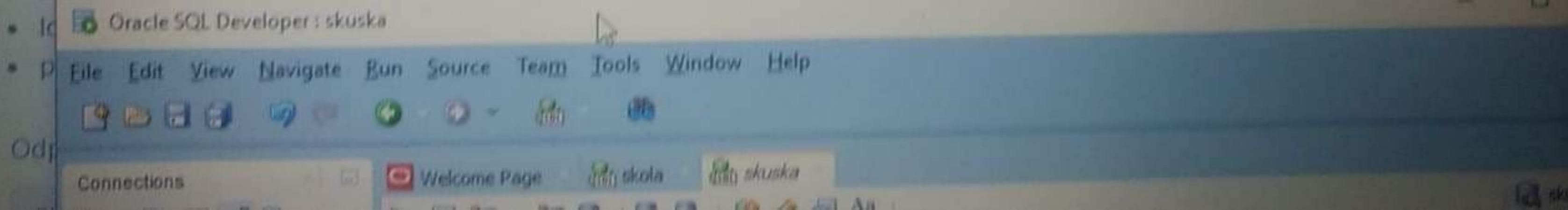
Pracujeme s databázou pre autopožičovňu, ktoréj relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti pobočky, na ktorých nie je žiadne auto jazdiace na benzin (stĺpec palivo). Vráťte celý záznam pobočky. Pozor, ak je na pobočke čo i len jediné auto na benzin, pobočka vo výsledku nesmie byť.

Vo vzorovej DB je to pobočka | 1 | 'Farská 2' | 'Žilina' |.



Testovať svoj dopyt môžete na školskej databáze na nasledovnom pripojení (všetci to pripojenie zdieľate, preto vo vlastnom záujme z databázy nič nemažte):

- hostname: oracle.kpi.fei.tuke.sk
- port: 1521
- SID: db11a
- User: Oracle SQL Developer : skuska
- Password: skuska



Majme dopyt:

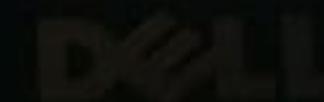
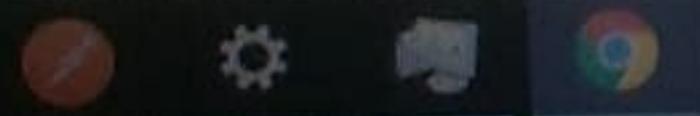
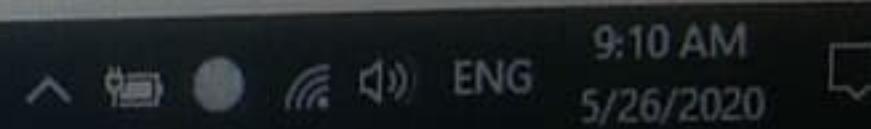
```
SELECT 'Zamestnanec: ' || e.firstname || e.lastname, 'Zakazník: ' || c.firstname || c.lastname  
FROM customer c JOIN employee e ON c.supportRepId = e.employeeId  
WHERE e.hiredate < '1.1.2010';
```

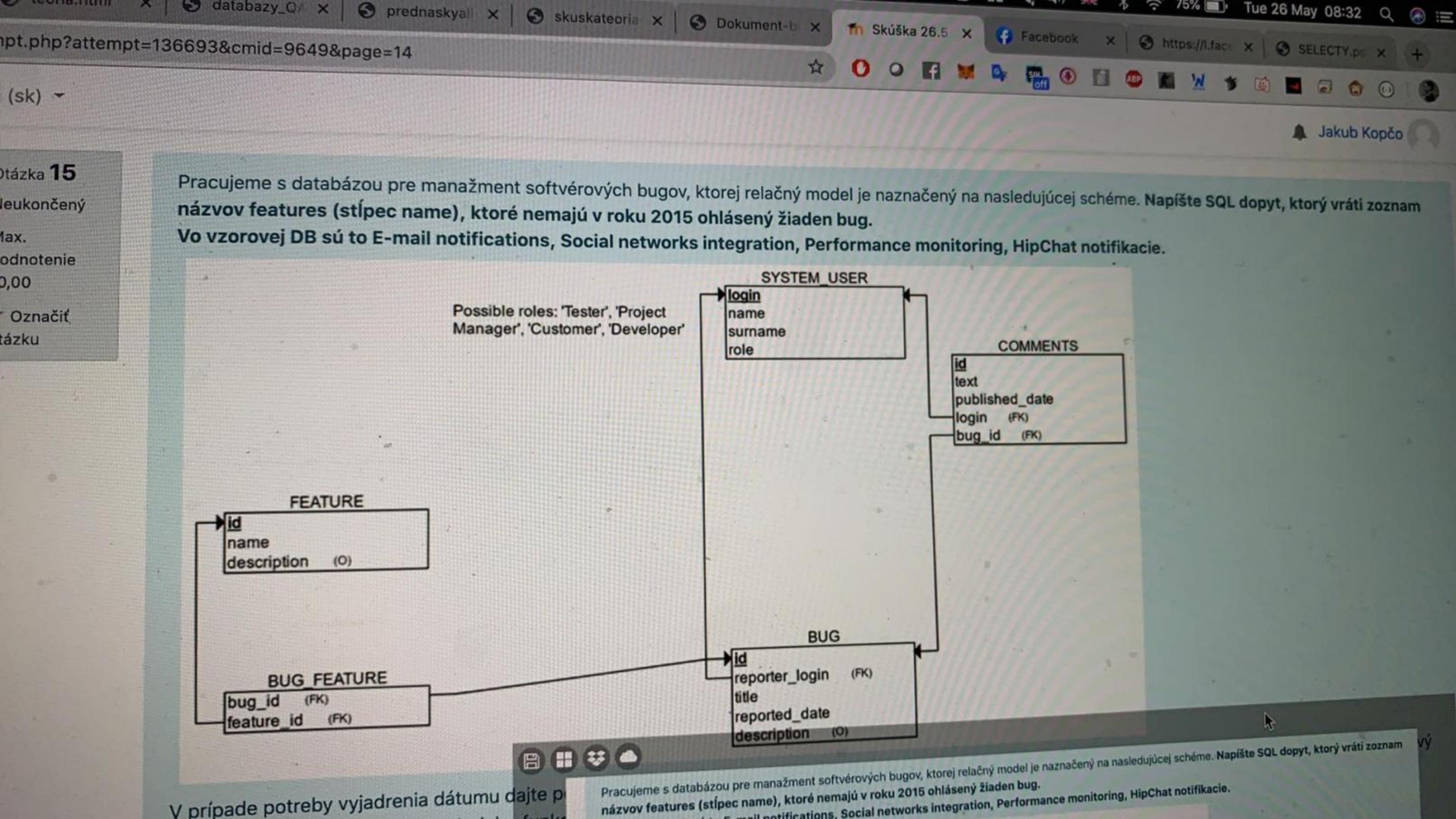
Ktoré z nasledujúcich indexov databázový systém **vie** využiť pre zefektívnenie jeho vykonávania?

Označte jednu alebo viac odpovedí:

- a. zložený index nad stĺpcami firstname a lastname tabuľky customer
- b. index nad stĺpcom birthdate tabuľky employee
- c. index nad stĺpcom employeeId tabuľky employee
- d. index nad stĺpcom supportRepId tabuľky customer

Skontrolovať







26.5.2020

Vyber pravdivé tvrdenia o entitnej integrite údajov v rámci relačného modelu:

Označte jednu alebo viac odpovedí:

- a. primárny kľúč tabuľky musí byť vždy uvedený (nesmie mať hodnotu NULL) ✓
- b. akékolvek neuvedenie povinného atribútu entity porušuje entitnú integritu databázy
- c. použitie viachodnotového atribútu (napr. titul s možnosťou uvedenia viacerých titulov v rámci jednej entity) porušuje entitnú integritu
- d. entitná integrita zabezpečuje jednoznačnú identifikateľnosť každej entity v databáze ✓
- e. rôzne entity v tabuľke sú vždy rozlíšiteľné prinajmenšom primárnym kľúčom ✓

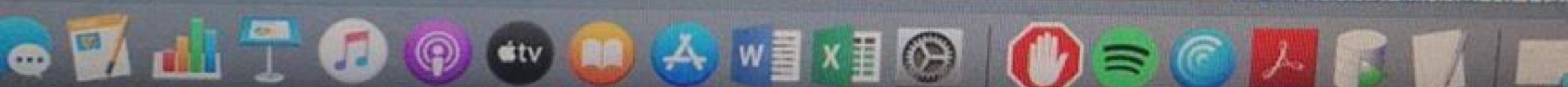
Skontrolovať

Správna odpoveď

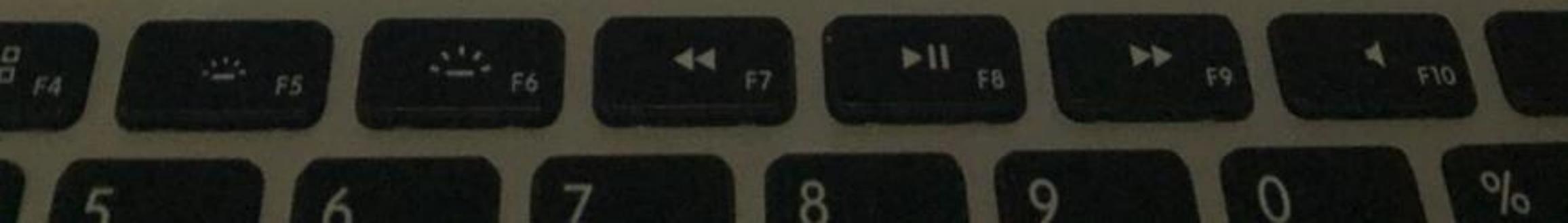
Hodnotenie pre tento pokus: 2,00/2,00.

strana

A Relational Model of Data



MacBook Air



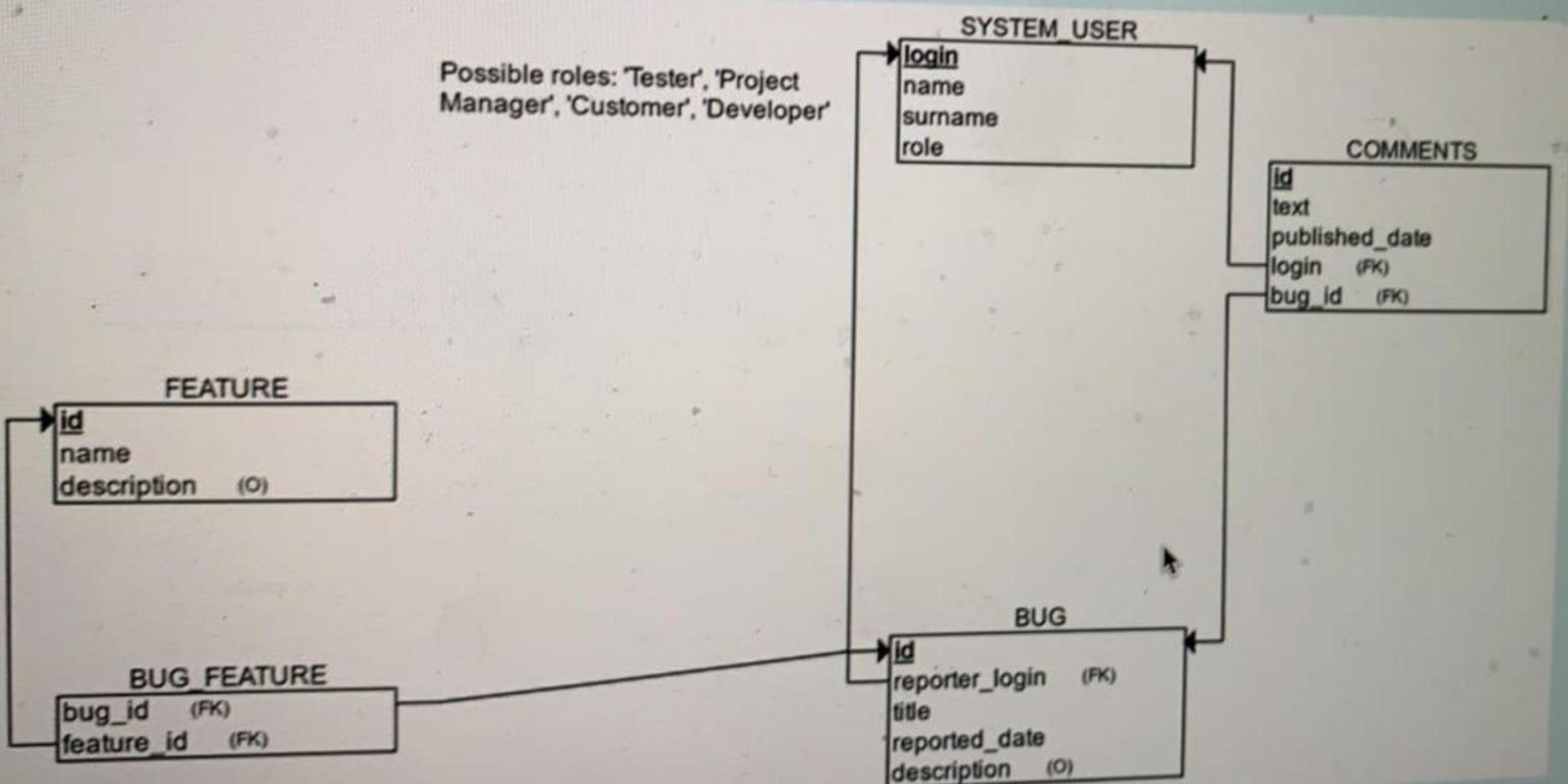
14

nčený

tenie

ačít

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti zoznam názov features (názov stĺpca dajte "FEATURE"), a počet ich bugov (názov stĺpca dajte "POCET"). Zobrazte aj tie features, ktoré nemajú žiadny bug, pričom pri nich má byť počet samozrejme 0.
Vo vzorovej DB má napr. User management 4 bugy, a HipChat notifikacie majú 0 bugov.



V prípade potreby vyjadrenia dátumu dajte p...

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti zoznam názov features (stĺpec name), ktoré nemajú v roku 2015 ohľásený žiadny bug.

vý

nájdete v rámci kurzu. Social networks integration, Performance monitoring, HipChat notifikacie.

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)

- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)

- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:37

Otázka 1

Ešte
nezodpovedané

Max. hodnotenie
2,00

 Označiť otázku

Čo opisuje interná (fyzická) schéma databázy?

Označte jednu odpoveď:

- a. opisuje vlastnosti fyzických zariadení, na ktorých je databáza implementovaná (napr. potrebnú veľkosť RAM pamäte, a pod.)
- b. fyzické uloženie údajov, formát ich uloženia, a pod.
- c. opisuje konceptuálny model používaný databázou, prostredníctvom ktorého komunikuje klientský program s databázou (napr. relačný model v prípade relačných databáz), ktorý je v rámci programu interný (používateľovi programu nie sú údaje prezentované v rovnakom formáte)
- d. sú to konkrétné dátové súbory fyzicky uložené na disku, ktoré obsahujú samotné údaje

[Ďalší](#)

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)

- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)

- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu
0:57:32

Otázka 2

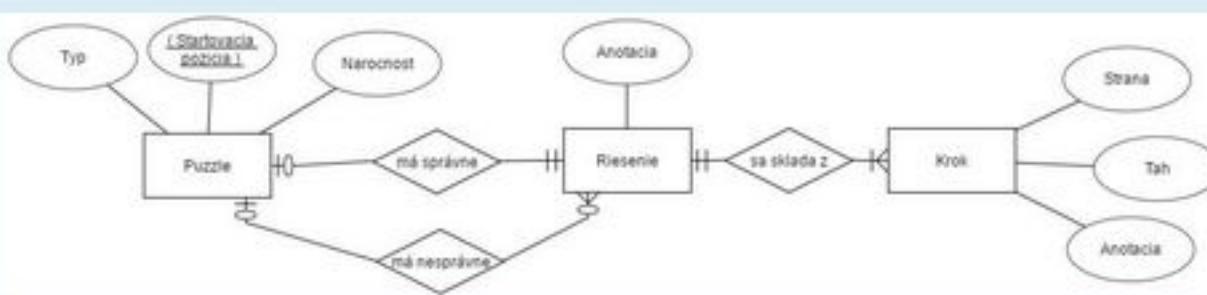
Ešte
nezodpovedané

Max. hodnotenie

2,00

Označiť otázku

Nasledujúci jednoduchý ERD opisuje databázu šachových hlavolamov (puzzle). Ako prevedieme do relačného modelu vzťah "sa sklada z" medzi Riesenim a Krokmi?



Označte jednu alebo viac odpovedí:

- a. do tabuľky "Riesenie" pridáme cudzí kľúč odkazujúci sa na tabuľku "Krok"
- b. vytvoríme prepájaciu tabuľku "skladaSaZ", ktorá bude obsahovať dva cudzie kľúče odkazujúce sa na "Krok" a "Riesenie"
- c. do tabuľky "Krok" pridáme cudzí kľúč odkazujúci sa na tabuľku "Riesenie"
- d. do tabuľky "Krok" pridáme cudzí kľúč odkazujúci sa na tabuľku "Puzzle"
- e. vytvoríme prepájaciu tabuľku "skladaSaZ", ktorá bude obsahovať dva cudzie kľúče odkazujúce sa na "Riesenie" a "Krok"

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)

- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)

- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:29

Otázka 3

Ešte

nezodpovedané

Max. hodnotenie

2,00

 Označiť otázku

Vyberte, ktoré tvrdenia opisujú definíciu 3. normálnej formy:

Označte jednu alebo viac odpovedí:

- a. neklúčové atribúty sú navzájom nezávislé
- b. primárny kľúč nesmie byť zložený z viacerých atribútov
- c. v neklúčových atribútoch sa neopakujú rovnaké hodnoty
- d. primárny kľúč nesmie byť zároveň cudzím kľúčom
- e. neklúčové atribúty sú na primárnom kľúči závislé priamo a nie tranzitívne

[Ďalší](#)

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)
- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)
- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:23

Otázka 4

Ešte

nezodpovedané

Max. hodnotenie
2,00

Označiť otázku

V jazyku SQL je platným zápisom logických podmienok výraz:

Označte jednu alebo viac odpovedí:

- expression OR expression
- expression || expression
- expression AND expression
- expression && expression
- expression ^^ expression

[Ďalší](#)

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)
- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)
- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

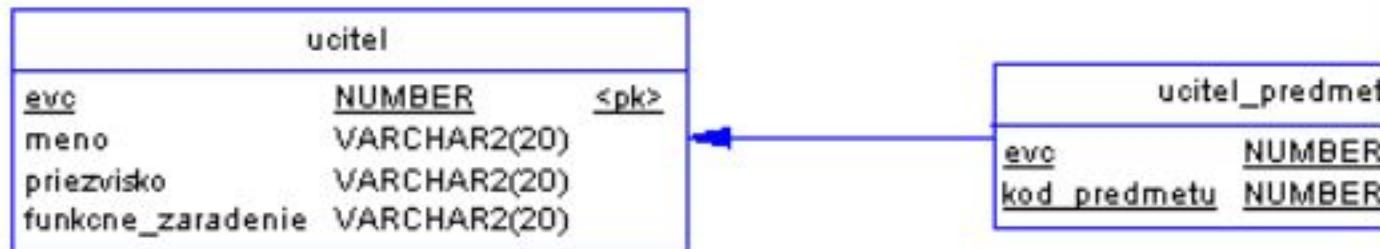
0:57:18

Otázka 5

Ešte
nezodpovedané
Max. hodnotenie
2,00

Označiť otázku

Označte taký SQL skript databázy pre danú relačnú schému
(viď obr.), aby platili všetky integritné pravidlá ?



Označte jednu odpoveď:

[code sql]

```
CREATE TABLE ucitel
(
    evc NUMBER PRIMARY KEY,
    meno VARCHAR2(20),
    priezvisko VARCHAR2(20),
    funkcie_zaradenie VARCHAR2(20)
);
```

```
CREATE TABLE predmet
(
    kod_predmetu NUMBER PRIMARY KEY,
    nazov VARCHAR2(50),
    kredity NUMBER
);
```

```
CREATE TABLE ucitel_predmet
(
    evc NUMBER,
    kod_predmetu NUMBER
);
```

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)

- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)

- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:57:08

Otázka 6

Ešte
nezodpovedané

Max. hodnotenie
2,00

 Označiť otázku

Vyberte pravdivé tvrdenia o zjednotení ako množinovej operácii
v SQL:

Označte jednu alebo viac odpovedí:

- a. UNION je výkonomovo rýchlejší ako UNION ALL
- b. UNION ALL zjednotí všetky záznamy a pritom odstráni duplikáty
- c. v rámci jedného dopytu sa môže UNION použiť iba raz
- d. ORDER BY sa môže pri použití UNION použiť iba v rámci druhého SELECT-u
- e. UNION ALL zjednotí záznamy tak, že neodstráni duplikáty

[Ďalší](#)

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)

- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)

- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu
0:55:52

Otázka 9

Ešte

nezodpovedané

Max. hodnotenie

2,00

 Označiť otázku

Vyberte pravdivé tvrdenia o riadkových triggeroch:

Označte jednu alebo viac odpovedí:

- a. vykonajú sa raz pre každý riadok ovplyvnený udalosťou
- b. nemôžu reagovať ako náhrada udalosti (teda nemôžu byť v kombinácii s INSTEAD OF)
- c. musia mať definovanú podmienku v klazule WHEN
- d. sú to triggers označené klauzulou FOR EACH ROW

[Ďalší](#)

NAVIGÁCIA V TESTE

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:55:58

Otázka 10

Ešte
nezodpovedané

Max. hodnotenie
2,00

Označiť otázku

Majme nasledovný dopyt v relačnej algebre:

$\rho_{meno}(\pi_{composer}(track))$

Ktorý z nasledujúcich SQL dopytov vráti rovnaký výsledok pri akýchkoľvek údajoch v databáze?

Označte jednu odpoveď:

- a. `SELECT meno
FROM track
WHERE name='composer';`
- b. `SELECT composer AS meno
FROM track;`
- c. `SELECT DISTINCT composer AS meno
FROM track;`
- d. `SELECT DISTINCT composer
FROM track
WHERE composer='meno';`

[Ďalší](#)

NAVIGÁCIA V TESTE

- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)
- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)
- [11](#)
- [12](#)
- [13](#)
- [14](#)
- [15](#)

[Ukončiť pokus ...](#)

Čas zostávajúci do
ukončenia testu

0:58:03

Otázka 11

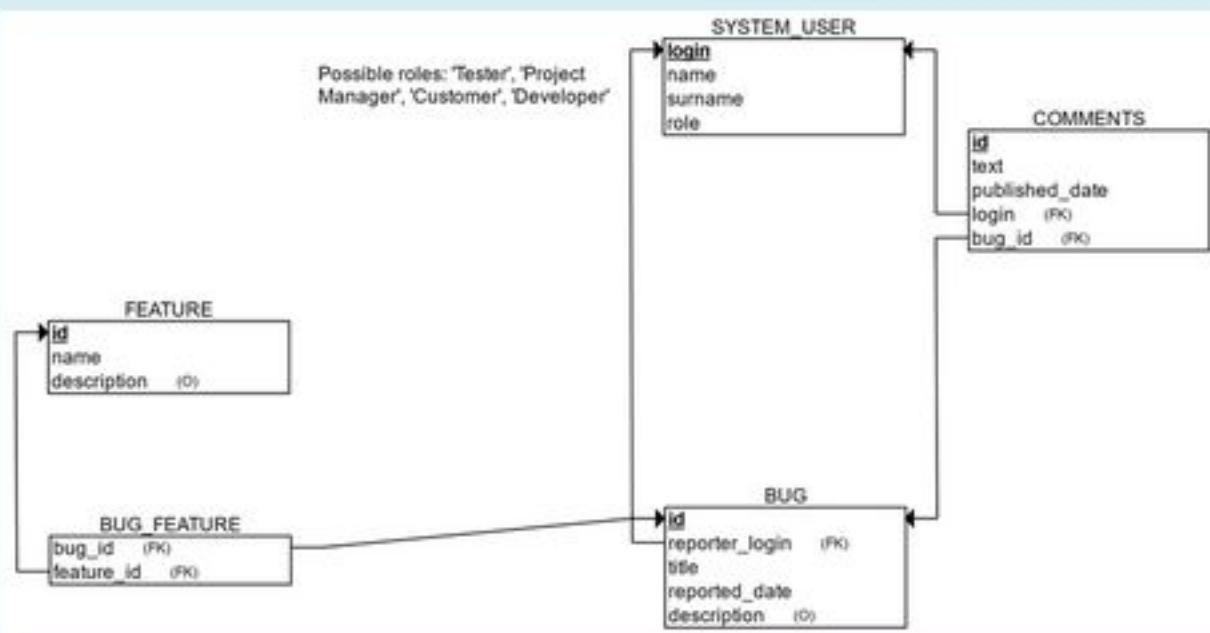
Ešte
nezodpovedané
Max. hodnotenie
7,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktoréj relačný model je naznačený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti počet features, ktoré majú ohlásený nejaký bug. Stípec s výsledkom pomenujte "BUGGED_FEAT".

Vo vzorovej DB je počet takých features 3.



Formát dátumu v DB je 'DD.MM.YYYY', napr. '01.04.2016'.

Testovať svoj dopyt môžete na školskej databáze na nasledovnom pripojení (všetci to pripojenie zdieľate, preto vo vlastnom záujme v databáze necommitujte žiadne zmeny):

- hostname: oracle.kpi.fei.tuke.sk
- port: 1521
- SID: db11g
- connection name: exams_bugs
- login: exams_bugs
- password: abc

Odpoved:

Navigácia v teste

- 1
- 2
- 3
- 4
- 5
- 6
- 7

[Ukončiť pokus ...](#)

Zvyšný čas **0:03:19**

Otázka **6**

Ešte
nezodpovedané
Max.

hodnotenie 2,00

Označiť
otázku

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napište dopyt ktorý vráti dvojice oblast' (prvý stĺpec) - mesto patriace danej oblasti (druhý stĺpec), ale len pre mestá ktoré vznikli medzi rokmi 1200 a 1300 (vrátane) a tiež pre oblasti bez mesta (vo výsledku pre ne bude v pravom stĺpci namiesto názvu mesta NULL).

Výsledky zoradťte abecedne podľa názvu oblasti.

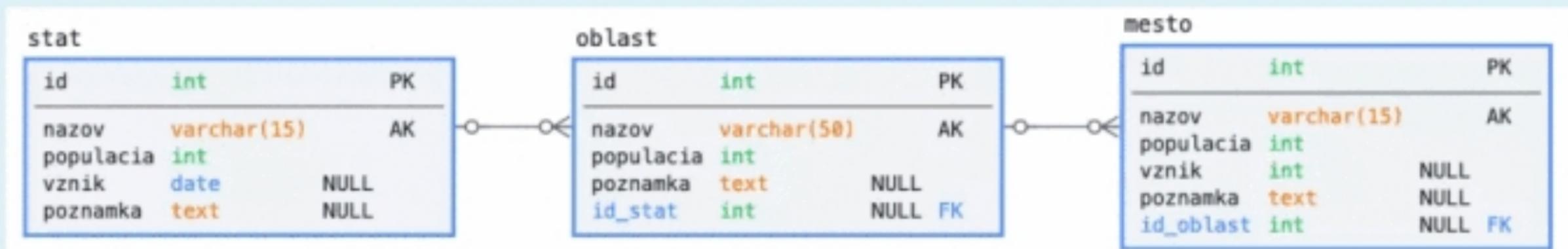
Vo vzorovej DB výsledok dopytu vráti 8 riadkov.



```
SELECT o.nazov, m.nazov
FROM oblast o
LEFT JOIN mesto m ON o.id = m.id_oblast
WHERE m.vznik BETWEEN 1200 AND 1300 OR m.id IS NULL
ORDER BY o.nazov;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti maximum, minimum a priemer populácie všetkých miest.

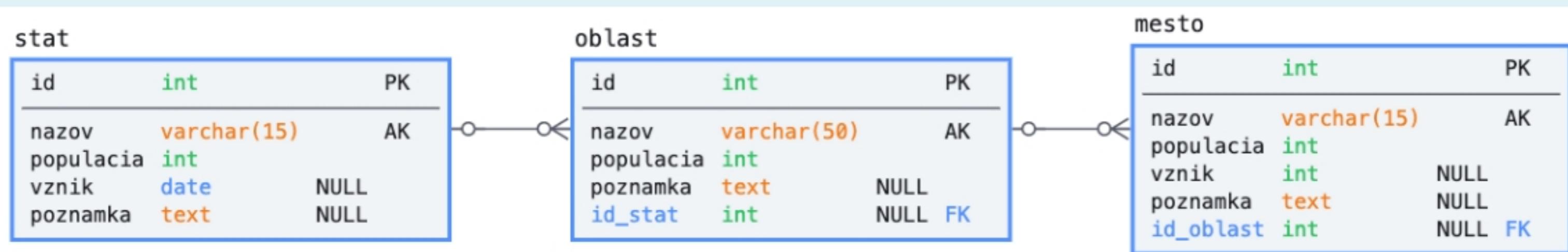


```
SELECT MAX(populacia), MIN(populacia), AVG(populacia)
FROM mesto;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti pre každý štát počet miest v jednotlivých štátoch.
- Výsledný select bude obsahovať 2 stĺpce – názov štátu a miest príslušného štátu.
- Pomenujte ich ako stat a pocet, v uvedenom poradí, bez diakritiky.
- Výsledok zoskupte podľa štátu.

Vo vzorovej DB výsledok dopytu vráti 2 riadky.

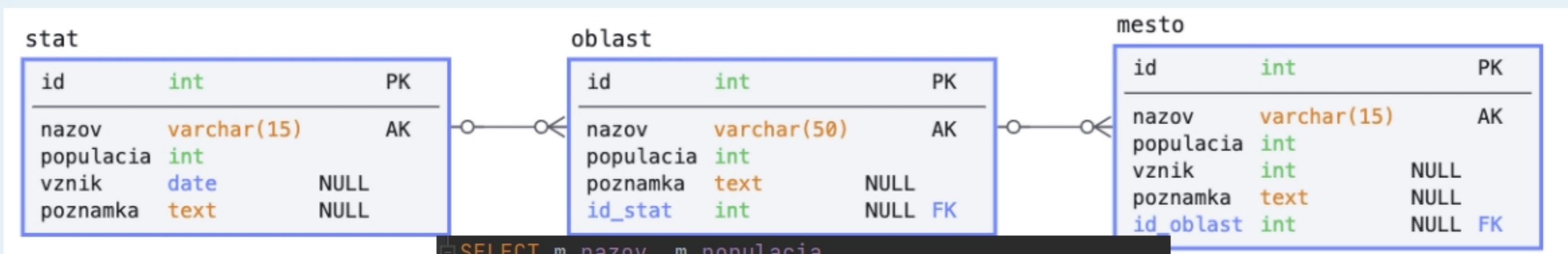


```
-> SELECT s.nazov as stat, COUNT(m.id) as pocet
   FROM stat s
   JOIN oblast o ON s.id = o.id_stat
   JOIN mesto m ON o.id = m.id_oblast
 GROUP BY stat;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti všetky mestá a ich počet obyvateľov (len tieto 2 stĺpce), z oblasti v ktorej súčet obyvateľov jej všetkých miest je najnižší.
- Výsledok zoradťte zostupne.
- Výsledný select bude obsahovať 2 stĺpce – názov mesta a jeho populáciu v uvedenom poradí, bez aliasov.

Vo vzorovej DB výsledok dopytu vráti 3 riadky.

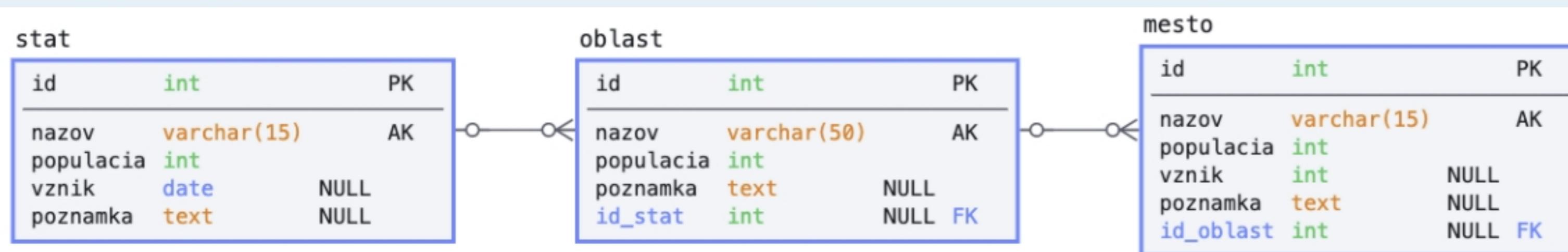


```
SELECT m.nazov, m.populacia
FROM mesto m
JOIN oblast o ON o.id = m.id_oblast
WHERE o.id = (
    SELECT o.id
    FROM oblast o
    JOIN mesto m ON o.id = m.id_oblast
    GROUP BY o.id
    HAVING sum(m.populacia) = (
        SELECT sum(m.populacia)
        FROM oblast
        JOIN mesto m ON oblast.id = m.id_oblast
        GROUP BY oblast.nazov
        ORDER BY 1
        LIMIT 1
    )
)
ORDER BY m.populacia DESC;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti všetky oblasti, ich názov, populáciu a id štátu ktorému patria (len tieto 3 stĺpce v takomto poradí), ktoré majú viac ako 100.000 (stotisíc) obyvateľov a zoradťte ich podľa populácie od najmenšej.

Vo vzorovej DB 9 oblastí.

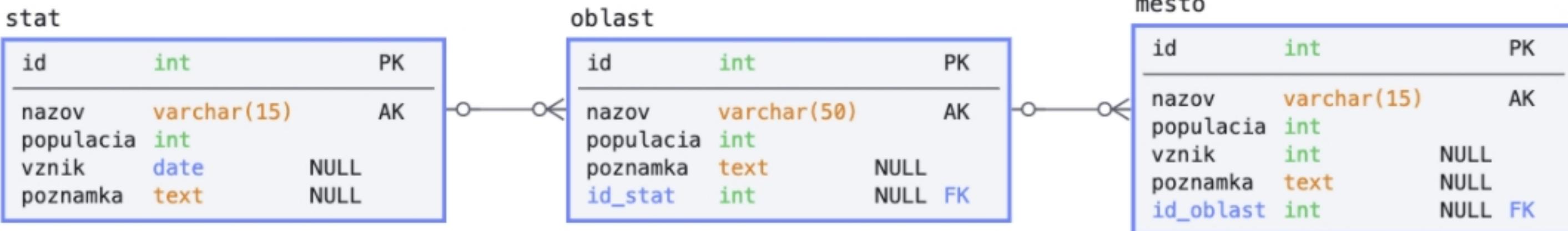


```
SELECT nazov, populacia, id_stat  
From oblast  
WHERE populacia > 100000  
ORDER BY populacia;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti počet miest v jednotných oblastiach.
- Výsledný select bude obsahovať 2 stĺpce – počet miest a názov príslušnej oblasti.
- Pomenujte ich ako pocet a oblast, v uvedenom poradí, bez diakritiky.
- Výsledok zoradťte zostupne.

Vo vzorovej DB výsledok dopytu vráti 5 riadkov.

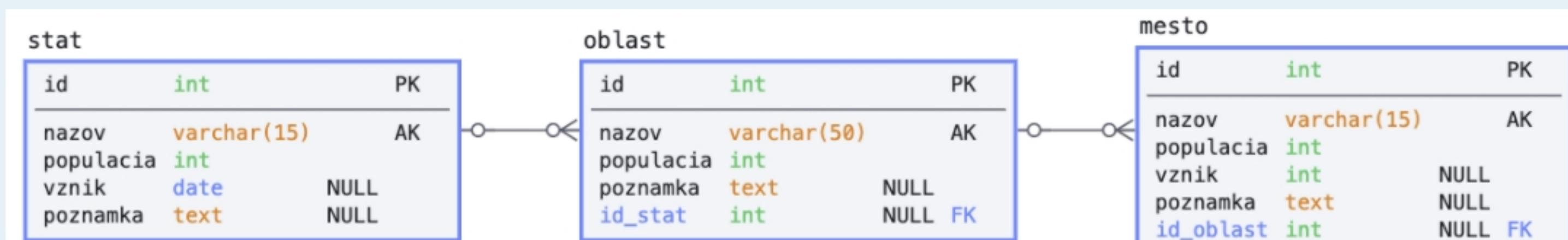


```
SELECT COUNT(m.id) as pocet, o.nazov as oblast  
FROM oblast o  
JOIN mesto m ON o.id = m.id_oblast  
GROUP BY oblast  
ORDER BY pocet DESC;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti počet miest v jednotných oblastiach.
- Výsledný select bude obsahovať 2 stĺpce – názov oblasti a počet miest.
- Pomenujte ich ako oblast a pocet, v uvedenom poradí, bez diakritiky.
- Výsledok zoradťte zostupne.

Vo vzorovej DB výsledok dopytu vráti 9 riadkov.

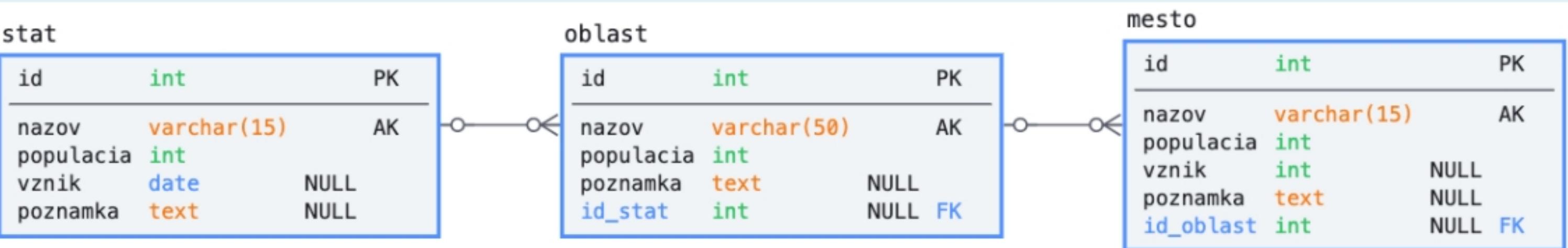


```
SELECT o.nazov as oblast, COUNT(m.id) as pocet
FROM oblast o
LEFT JOIN mesto m ON o.id = m.id_oblast
GROUP BY oblast
ORDER BY pocet DESC;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti názvy všetkých miest, ktorých názov sa nenachádza v plnej dĺžke v názve oblasti, v ktorej sa dané mestá nachádzajú.

Vo vzorovej DB výsledok dopytu vráti 8 riadkov, nájde aj Košice lebo oblasť sa volá Košický kraj (písmeno e v názve kraja chýba)



```
SELECT mesto.nazov FROM mesto
JOIN oblast ON mesto.id_oblast = oblast.id
WHERE oblast.nazov NOT LIKE '%' || mesto.nazov || '%';
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti všetky štáty, ich názov a populáciu v miliónoch (len tieto 2 stĺpce v takomto poradí, pri populácii uveďte aj desatinu časť v celej dĺžke). Výsledky zoradte podľa dĺžky názvu štátu začínajúc najkratším názvom, a v prípade rovnakej dĺžky podľa abecedy vzostupne.

Vo vzorovej DB by bol druhý riadok „Polsko 38.4336“ a Slovensko by bolo posledné v poradí.

stat

id	int	PK
nazov	varchar(15)	AK
populacia	int	
vznik	date	
poznamka	text	

oblast

id	int	PK
nazov	varchar(50)	AK
populacia	int	
poznamka	text	
id_stat	int	NULL FK

mesto

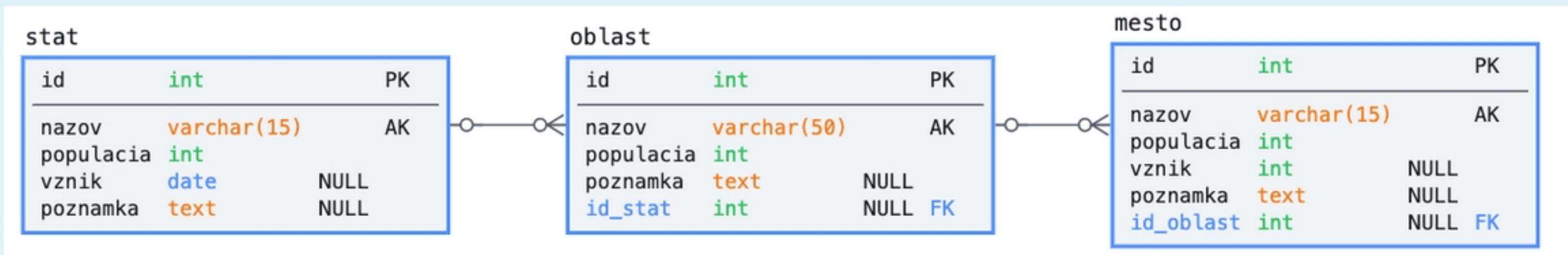
id	int	PK
nazov	varchar(15)	AK
populacia	int	
vznik	int	NULL
poznamka	text	NULL
id Oblast	int	NULL FK

```
SELECT nazov, cast((populacia) as float)/1000000 as populacia
FROM stat
ORDER BY length(nazov), nazov;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti názvy štátov, ich populáciu a dátum vzniku pre štáty, ktoré vznikli v rokoch 1918 alebo 1991 a ich populácia je väčšia ako 10000000.
- Výsledok zoradte od najmladšieho štátu po najstarší.
- Výsledný select nech obsahuje 3 stĺpce: názov štátu, populáciu a dátum vzniku, v uvedenom poradí, bez aliasov.

Vo vzorovej DB sú 2 takéto štáty.

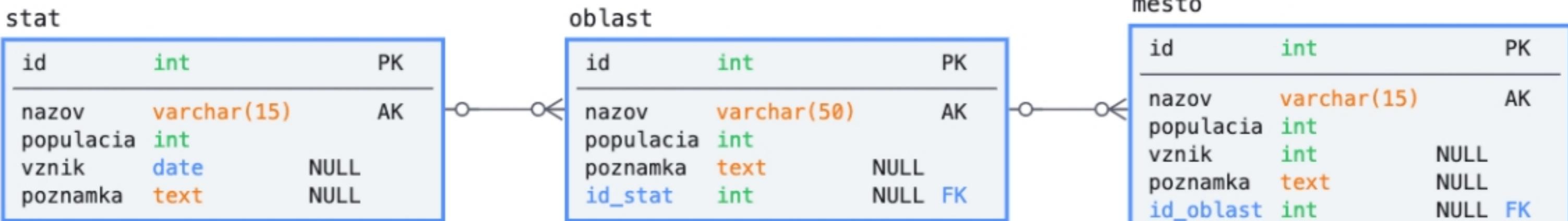


```
SELECT nazov, populacia, vznik
FROM stat
WHERE (date_part('year', vznik) = 1918 OR date_part('year', vznik) = 1991) AND populacia > 10000000
ORDER BY vznik DESC;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti všetky mestá, ich názov, populáciu a kedy vznikli (len tieto 3 stĺpce v takomto poradí), ktoré majú viac ako 100 tisíc obyvateľov a zoradťte ich podľa vzniku od najmladšieho.

Vo vzorovej DB sú 3 takéto mestá.



```
SELECT nazov, populacia, vznik
FROM mesto
WHERE populacia > 100000
ORDER BY vznik DESC;
```

Vytvorte dopyt ktorý vráti 2 stĺpce.

- V prvom budú názvy takých štátov ktoré majú poznámku a okrem názvov takýchto štátov tu budú aj názvy všetkých miest ktoré nezačínajú písmenom K.
- V druhom stĺpci bude pre štáty spĺňajúce spomenuté podmienky určená suma populácie všetkých im priradených oblastí.
- Pre mestá bude v druhom stĺpci populácia najväčšieho mesta v oblasti ku ktorej patria.
- Výsledok zoradťte abecedne podľa názvov v prvom stĺpci.

Vo vzorovej databáze výsledok dopytu vráti 10 riadkov, pričom rozdiel medzi súčtom populácie v oblastiach pre jednotlivé štáty spĺňajúce podmienky by mal byť 4 347 150

stat

id	int	PK
nazov	varchar(15)	AK
populacia	int	
vznik	date	
poznamka	text	

oblast

id	int	PK
nazov	varchar(50)	AK
populacia	int	
poznamka	text	
id_stat	int	FK

mesto

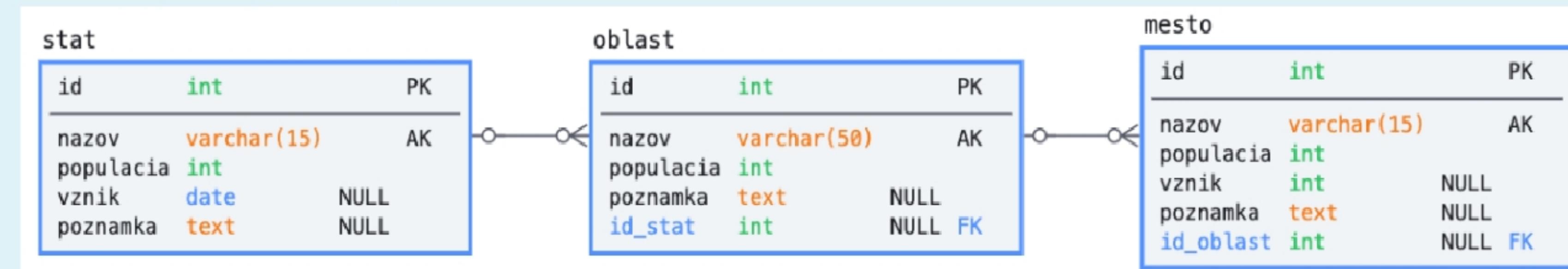
id	int	PK
nazov	varchar(15)	AK
populacia	int	
vznik	int	
poznamka	text	
id_oblast	int	FK

```
SELECT s.nazov, SUM(o.populacia)
FROM stat s
JOIN oblast o ON s.id = o.id_stat
WHERE s.poznamka IS NOT NULL
GROUP BY s.nazov
UNION
SELECT m.nazov, MAX(m2.populacia)
FROM mesto m
JOIN mesto m2 ON m.id_oblast = m2.id_oblast
WHERE m.nazov NOT LIKE 'K%'
GROUP BY m.nazov
ORDER BY nazov;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti názvy všetkých miest, oblasti a štátu do ktorého patria pre mestá, ktoré majú uvedenú poznámku.
- Výsledok zoradťte vzostupne podľa názvu mesta, oblasti a štátu.
- Výsledný select nech obsahuje 3 stĺpce: názov mesta, názov oblasti a názov štátu, v uvedenom poradí, bez aliasov.

Vo vzorovej DB výsledok dopytu vráti 3 riadky.

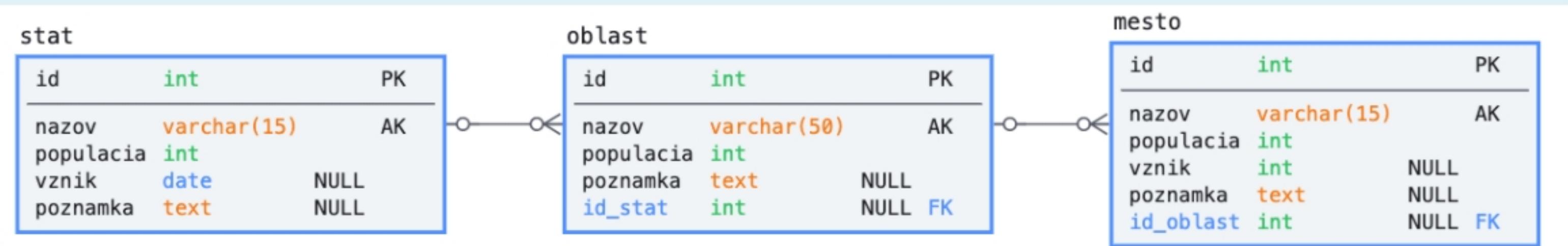


```
SELECT m.nazov, o.nazov, s.nazov
FROM stat s
LEFT JOIN oblast o ON s.id = o.id_stat
LEFT JOIN mesto m ON o.id = m.id_oblast
WHERE m.poznamka IS NOT NULL
ORDER BY m.nazov, o.nazov, s.nazov ASC;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti názov štátu a priemernú populáciu miest štátu, ktorého priemerný počet obyvateľov v mestách je najvyšší.
- Výsledný select bude obsahovať 2 stĺpce – názov štátu a jeho priemernú populáciu miest v uvedenom poradí.
- Pomenujte ich ako stat a priemer, v uvedenom poradí, bez diakritiky.

Vo vzorovej DB výsledok dopytu vráti 1 riadok.

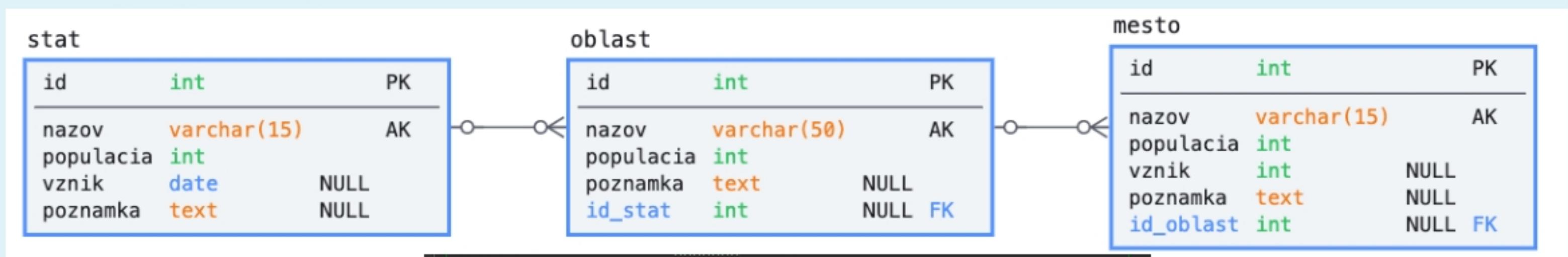


```
select s.nazov as stat, avg(m.populacia) as priemer
from stat s
join oblast o on s.id = o.id_stat
join mesto m on o.id = m.id_oberlast
group by s.id
order by priemer desc
limit 1;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti názvy miest, ich populáciu a názov príslušného štátu, pričom platí podmienka, že štát musí mať viac ako 5 000 000 obyvateľov a mesto musí mať viac ako 50 000 obyvateľov.
- Výsledok zoradte zostupne podľa populácie mesta.
- Výsledný select bude obsahovať 3 stĺpce – názov mesta a jeho populáciu a názov štátu v uvedenom poradí, bez aliasov.

Vo vzorovej DB výsledok dopytu vráti 6 riadkov.



```
SELECT m.nazov, m.populacia, s.nazov
FROM stat s
JOIN oblast o ON s.id = o.id_stat
JOIN mesto m ON o.id = m.id_oblast
WHERE s.populacia > 5000000 AND m.populacia > 50000
ORDER BY m.populacia DESC;
```

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti všetky oblasti, ich názov a názov štátu ktorému patria (len tieto 2 stĺpce v takomto poradí), ktoré majú vo vzorovej DB viac ako 1 mesto a zoradte ich podľa populácie od najmenšej.
- Výsledný SELECT bude obsahovať 2 stĺpce – názov oblasti a názov štátu v uvedenom poradí, bez aliasov.

Vo vzorovej DB výsledok dopytu vráti 4 riadkov.



```
SELECT o.nazov, s.nazov  
FROM stat s  
JOIN oblast o ON s.id = o.id_stat  
JOIN mesto m ON o.id = m.id_oberlast  
GROUP BY o.nazov, s.nazov, o.populacia  
HAVING COUNT(m.id_oberlast) > 1  
order by o.populacia;
```

↓ A▼ B I ⏹ Q

Ktorý príkaz SQL sa používa na vrátenie iba rôznych hodnôt?

DISTINCT

Uveďte údajový typ, ktorý vám umožní zadávať finančnú hodnotu v Eurách od -99,99 EUR do 99,99 EUR (po zaokrúhlení na dve desatinné miesta), ale napr. 100 EUR už nie. Bez použitia CHECK.

Uveďte údajový typ, ktorý vám umožní zadávať finančnú hodnotu v Eurách od -99,99 EUR do 99,99 EUR (po zaokrúhlení na dve desatinné miesta), ale napr. 100 EUR už nie. Bez použitia CHECK.

Odpoveď zapíšte bez medzier!

Odpoved:

V dopyte `SELECT * FROM customers WHERE email OPERATOR '%@gmail.com'` chýba operátor (nahradený slovom OPERATOR).

LIKE

Zapíšte údajový typ, ktorý vám umožní zadávať číslo s počtom všetkých číslic 5 a počtom číslic desatinnej časti 3.

numeric(5,3)

Ktorá funkcia vráti tretie až šieste písmeno z textu (String), vo vzorovom príklade atribútu „title“?

Otázka 2
Ešte nezodpovedané
Max. hodnotenie 3,00
Označiť otázku

Ktorá funkcia vráti tretie až šieste písmeno z textu (String), vo vzorovom príklade atribútu „title“?

- a. substr(title, 2,5) ❌
- b. substr(title, 3,3)
- c. substring(title, 3,6) ❌
- d. substr(title, 3,6) ❌

[Zrušiť moju voľbu](#)

Ktoré kľúčové slová sa používajú v súvislosti s transakciami v PostgreSQL?:

Ktoré kľúčové slová sa používajú v súvislosti s transakciami v PostgreSQL?:

Označte jednu alebo viac odpovedí:

- a. ROLLBACK
- b. EXCEPT
- c. GRANT
- d. SAVEPOINT
- e. BEGIN
- f. COMMIT

Pomocou ktorého z uvedených SQL vyberiete záznamy z tabuľky „car“ kde „name“ sa nachádza abecedne medzi (a vrátane) „Audi a „Ferrari“?

Pomocou ktorého z uvedených SQL vyberiete záznamy z tabuľky „car“ kde „name“ sa nachádza abecedne medzi (a vrátane) „Audi“ a „Ferrari“?

- a. `SELECT * FROM car WHERE name IN ('Audi ', 'Ferrari ');`
- b. `SELECT * FROM car WHERE name > 'Audi ' AND name < 'Ferrari ';`
- c. `SELECT * FROM car WHERE name BETWEEN 'Audi ' AND 'Ferrari ';`
- d. `SELECT * FROM car WHERE name <> 'Audi ' AND name <> 'Ferrari ';`

[Zrušiť moju voľbu](#)

Vyberte z nasledujúcich funkcií tie, ktoré sú agregačné:

Otázka **8**
Este
nezodpovedané
Max.
hodnotenie 3,00
 Označiť
otázku

Vyberte z nasledujúcich funkcií tie, ktoré sú agregačné:

Označte jednu alebo viac odpovedí:

- a. ROUND
- b. AVG
- c. COUNT
- d. TO_CHAR

Vyberte vlastnosti primárneho kľúča:

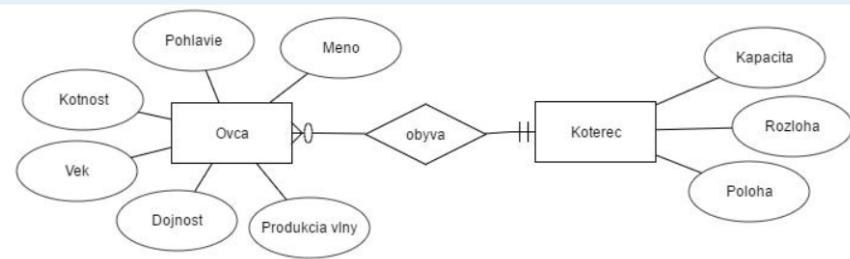
Vyberte vlastnosti primárneho kľúča:

Označte jednu alebo viac odpovedí:

- a. odkazuje sa na záznamy v inej tabuľke
- b. jednoznačne identifikuje záznamy v tabuľke
- c. môže mať hodnotu NULL
- d. má unikátnu hodnotu

Koľko tabuľiek vznikne prepisom nasledovného ER diagramu do relačného modelu?

Koľko tabuľiek vznikne prepisom nasledovného ER diagramu do relačného modelu?

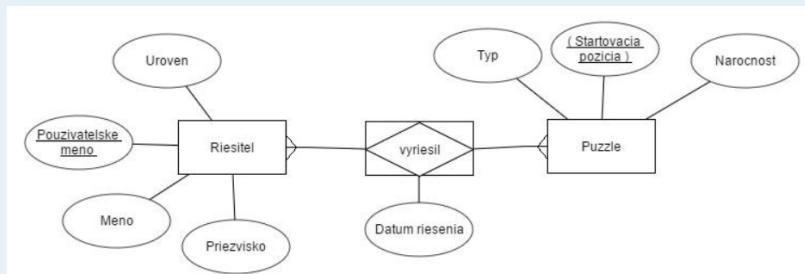


Označte jednu odpoved:

- a. 1
- b. 4
- c. 3
- d. 2

Máme ERD opisujúce databázu šachových hlavolamov ako je ukázané na nasledujúcim obrázku. Koľko stĺpcov bude mať v relačnom modele tabuľka pre asociatívnu entity "vyriesil"?

Máme ERD opisujúce databázu šachových hlavolamov ako je ukázané na nasledujúcim obrázku. Koľko stĺpcov bude mať v relačnom modele tabuľka pre asociatívnu entity "vyriesil"?



Označte jednu alebo viac odpovedí:

- a. 4
- b. 3
- c. 5
- d. 1
- e. 2

Vyberte pravdivé tvrdenia o agregačných funkciách:

Vyberte pravdivé tvrdenia o agregačných funkciách:

Označte jednu alebo viac odpovedí:

- a. Môžu sa použiť iba v selecte, ktorý má klauzulu GROUP BY, bez nej ich nie je možné použiť
- b. Agregujú viaceré hodnoty (napr. hodnoty v stĺpco) do jednej hodnoty, napr. vypočítajú priemernú hodnotu zo zoznamu čísel
- c. Umožňujú vnárať poddopyty aj vo WHERE klauzule
- d. Hodnota NULL je agregačnými funkciemi ignorovaná

Ktoré z nasledujúcich tvrdení o restrikčnom prístupe k zabezpečeniu referenčnej integrity sú pravdivé?

Ktoré z nasledujúcich tvrdení o restrikčnom prístupe k zabezpečeniu referenčnej integrity sú pravdivé?

Označte jednu alebo viac odpovedí:

- a. Ak je aktualizovaná referencia na nejaký záznam, automaticky sa aktualizuje aj primárny klúč, na ktorý sa referencia odkazovala
- b. Ak je zmazaný záznam, na ktorý existuje referencia, referencia je automaticky nastavená na NULL
- c. Ak je aktualizovaný primárny klúč záznamu, na ktorý existuje referencia, databáza vyhodí výnimku
- d. Ak je zmazaný záznam, na ktorý existuje referencia, je automaticky zmazaný aj referujúci záznam
- e. Ak je mazaný záznam, na ktorý existuje referencia, databáza vyhodí výnimku

Vyberte správnu alternatívu k podmienke: cena NOT BETWEEN 2000 AND 300
MUSÍ BYŤ OR

Vyberte správnu alternatívu k podmienke: cena NOT BETWEEN 1000 AND 4000

Vyberte správnu alternatívu k podmienke:

cena NOT BETWEEN 1000 AND 4000

- a. cena ≥ 1000 AND cena ≤ 4000
- b. cena = 1000 AND cena = 4000
- c. cena < 1000 OR cena > 4000
- d. cena ≤ 1000 OR cena ≥ 4000

[Zrušiť moju voľbu](#)

Vyberte, čo bude platiť po vykonaní nasledujúceho príkazu za predpokladu, že tabuľka penazenka(okamih,financie) existuje, obsahuje 100 riadkov a jej riadky nie sú referované žiadnymi cudzími klúčmi: TRUNCATE TABLE penazenka WHERE financie ≥ 100 ;

Vyberte, čo bude platiť po vykonaní nasledujúceho príkazu za predpokladu, že tabuľka penazenka(okamih,financie) existuje, obsahuje 100 riadkov a jej riadky nie sú referované žiadnymi cudzími klúčmi:

TRUNCATE TABLE penazenka WHERE financie ≥ 100 ;

- a. obsah tabuľky sa nezmiení
- b. odstránia sa všetky riadky z tabuľky, pre ktoré platí uvedená podmienka ❌
- c. odstránia sa všetky riadky z tabuľky ❌
- d. odstránia sa všetky riadky z tabuľky, pre ktoré neplatí uvedená podmienka ❌

[Zrušiť moju voľbu](#)

Napíšte príkaz, ktorým obmedzíte počet riadkov, ktorý vráti dotaz na 5. Zariadťte taktiež aby bola prvá hodnota vyniechaná, teda vrátia sa riadky 2-6.

LIMIT 5 OFFSET 1

Aký retazec dostanete aplikáciou funkcie replace('matematika','ka','te')?

matematite

Ktorým príkazom vieme zrušiť transakciu, aby sa neuložili v databáze zmeny?

Ktorým príkazom vieme zrušiť transakciu, aby sa neuložili v databáze zmeny?

- a. CANCEL
- b. UNDO
- c. ROLLBACK
- d. REVERSE
- e. ROLLUP

Zrušiť moju voľbu

Vyberte tvrdenia, ktoré opisujú význam nasledujúceho dopytu: SELECT priezvisko FROM zamestnanec
INTERSECT SELECT priezvisko FROM zakazník;

Vyberte tvrdenia, ktoré opisujú význam nasledujúceho dopytu:

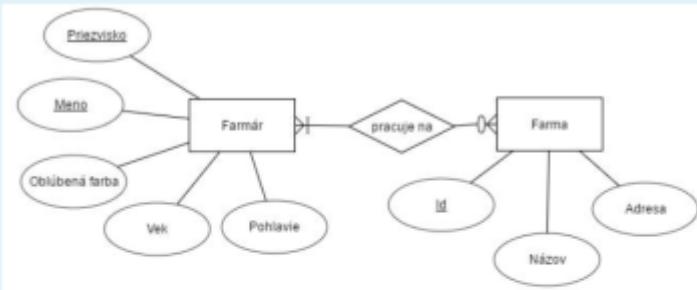
```
SELECT priezvisko FROM zamestnanec
INTERSECT
SELECT priezvisko FROM zákazník;
```

- a. Výstupom selectu sú všetky priezviská, ktoré sa nachádzajú v tabuľke zamestnanec okrem tých, ktoré sú v tabuľke zákazník
- b. Ani jedno z tvrdení nie je správne
- c. Výstupom selectu je zjednotená množina priezvisk z tabuľiek zamestnanec a zákazník
- d. Výstupom selectu je prienik priezvisk, nachádzajúcich sa v tabuľkach zákazník a zamestnanec

Zrušiť moju voľbu

Majme ERD pre farmu ako je ukázaný na nasledujúcom obrázku. Vyberte pravdivé tvrdenia vzhľadom na vzťah "pracuje na".

Majme ERD pre farmu ako je ukázaný na nasledujúcom obrázku. Vyberte pravdivé tvrdenia vzhľadom na vzťah "pracuje na".

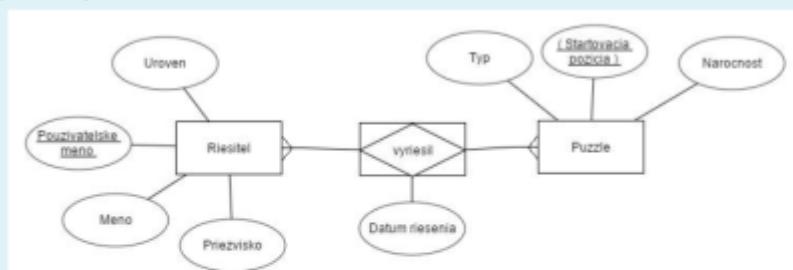


Označte jednu alebo viac odpovedí:

- a. ten istý farmár mohol pracovať na viacerých farmách, ale nemôže naraz pracovať na viacerých farmách v tom istom čase
- b. na farme môže pracovať viaceri farmári, ale každý farmár môže v tom istom časom okamžiku pracovať len na jednej farme
- c. v relačnom modele bude tento vzťah vyjadrený cudzím klúčom na strane farmára
- d. jeden farmár môže v tom istom čase pracovať naraz na viacerých farmách
- e. na jednej farme môže súčasne pracovať viaceri farmári, pričom každý z nich môže naraz pracovať na viacerých farmách

Máme ERD opisujúce databázu šachových hlavolamov ako je ukázané na nasledujúcom obrázku. Prečo je atribút "Pouzívateľské meno" entitnej množiny "Riešiteľ" podčiarknutý?

Máme ERD opisujúce databázu šachových hlavolamov ako je ukázané na nasledujúcom obrázku. Prečo je atribút "Pouzívateľské meno" entitnej množiny "Riešiteľ" podčiarknutý?



Označte jednu alebo viac odpovedí:

- a. lebo používateľské meno je povinný atribút - každý riešiteľ ho musí mať definované
- b. lebo hodnota používateľského mena je odvodnená (derivovaná) z mena a priezviska riešiteľa
- c. lebo používateľské meno riešiteľa musí byť unikátné (jedinečné)
- d. lebo používateľské meno je aktualizovateľné - je ho možné meniť

Vyberte pravdivé tvrdenia o referenčnej integrite:

Vyberte pravdivé tvrdenia o referenčnej integrite:

Označte jednu alebo viac odpovedí:

- a. nulitný prístup riešenia referenčnej integrity namiesto zmazania entity nastaví hodnoty všetkých jej atribútov na null, ale jej primárny klúč nechá, čím zabezpečí, že cudzie klúče sa budú mať na čo odkazovať
- b. kaskádny prístup k riešeniu referenčnej integrity kaskáduje zmazanie (zmenu) na všetky záznamy odkazujúce sa na práve mazaný (menený) záznam
- c. reštrikčný prístup k riešeniu referenčnej integrity zabraňuje zmazať záznam v databáze, ak sa naň odkazuje nejaký cudzí klúč
- d. restrikčný prístup riešenia referenčnej integrity zabráni zmazať záznamy, ktoré sa odkazujú na iné tabuľky

Vyberte pravdivé tvrdenia o zjednotení ako množinovej operácii v SQL:

Vyberte pravdivé tvrdenia o zjednotení ako množinovej operácii v SQL:

Označte jednu alebo viac odpovedí:

- a. UNION ALL zjednotí všetky záznamy a pritom odstráni duplikáty
- b. ORDER BY sa môže pri použíti UNION použiť iba v rámci druhého SELECT-u
- c. UNION je výkonovo rýchlejší ako UNION ALL
- d. v rámci jedného dopytu sa môže UNION použiť iba raz
- e. UNION ALL zjednotí záznamy tak, že neodstráni duplikáty

Vyberte, čo musí platiť, aby bola databáza v 1. normálnej forme:

Vyberte, čo musí platiť, aby bola databáza v 1. normálnej forme:

Označte jednu alebo viac odpovedí:

- a. v žiadnom stĺpci sa nesmú zbytočne opakovať rovnaké hodnoty
- b. databáza nemôže obsahovať zložené stĺpce
- c. všetky primárne klúče musia byť čiselné umelé klúče
- d. hodnoty všetkých stĺpcov v databáze musia byť atomické
- e. databáza nemôže obsahovať viachodnotové stĺpce

Vyberte pravdivé tvrdenia o relačnom modeli:

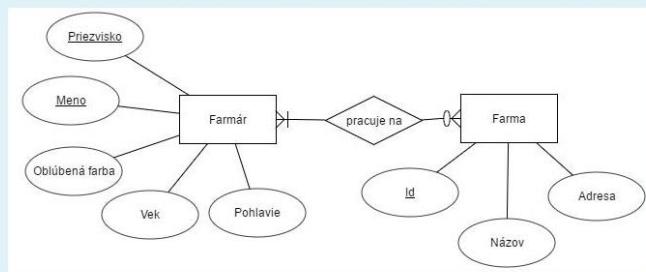
Vyberte pravdivé tvrdenia o relačnom modeli:

Označte jednu alebo viac odpovedí:

- a. Každý stĺpec obsahuje hodnoty toho istého atribútu
- b. Na poradí stĺpcov v tabuľke záleží
- c. Každá tabuľka má jednoznačný názov
- d. Pre daný záznam musí byť zadaná hodnota každého stĺpca
- e. Nie všetky stĺpce musia mať svoj názov
- f. Na poradí záznamov nezáleží
- g. Všetky hodnoty v danom zázname sú jednoznačne a plne závislé na primárnom klúči záznamu

Majme ERD pre farmu ako je ukázaný na nasledujúcom obrázku. Vyberte pravdivé tvrdenia vzhľadom na vzťah "pracuje na".

Majme ERD pre farmu ako je ukázaný na nasledujúcom obrázku. Vyberte pravdivé tvrdenia vzhľadom na vzťah "pracuje na".

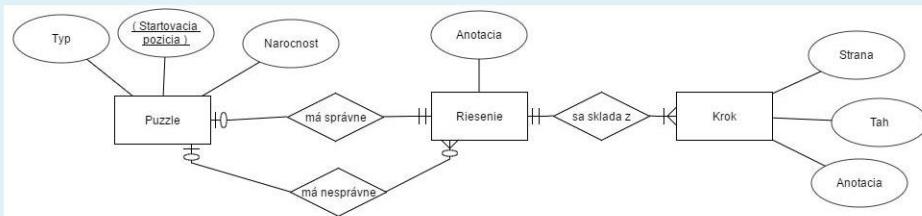


Označte jednu alebo viac odpovedí:

- a. na farme môže pracovať viaceri farmári, ale každý farmár môže v tom istom časovom okamžiku pracovať len na jednej farme
- b. na jednej farme môže súčasne pracovať viaceri farmári, pričom každý z nich môže naraz pracovať na viacerých farmách
- c. ten istý farmár môhol pracovať na viacerých farmách, ale nemôže naraz pracovať na viacerých farmách v tom istom čase
- d. v relačnom modele bude tento vzťah vyjadrený cudzím klúčom na strane farmára
- e. jeden farmár môže v tom istom čase pracovať naraz na viacerých farmách

Nasledujúci jednoduchý ERD opisuje databázu šachových hlavolamov (puzzle). Ktorá/é z nasledujúcich možností platí pre prevod vzťahu "má správne" do relačného modelu?

Nasledujúci jednoduchý ERD opisuje databázu šachových hlavolamov (puzzle). Ktorá/é z nasledujúcich možností platí pre prevod vzťahu "má správne" do relačného modelu?



Označte jednu alebo viac odpovedí:

- a. vzťah "má správne" môže byť v relačnom modele vyjadrený pridaním cudzieho klúča do tabuľky "Puzzle", odkazujúceho sa na tabuľku "Riesenie"
- b. vzťah "má správne" bude v relačnom modele vyjadrený prepájacou medzitabuľkou "maSpravne" s dvoma stĺpcami, ktoré budú cudzími klúčmi prepájajúcimi tabuľky "Riesenie" a "Puzzle"
- c. vzťah "má správne" môže byť v relačnom modele vyjadrený pridaním cudzieho klúča do tabuľky "Riesenie", odkazujúceho sa na tabuľku "Puzzle"
- d. vzťah "má správne" bude v relačnom modele vyjadrený prepájacou medzitabuľkou "maRiesenie" s dvoma stĺpcami, ktoré budú cudzími klúčmi prepájajúcimi tabuľky "Riesenie" a "Puzzle", pričom táto medzitabuľka bude zároveň vyjadrovať aj vzťah "má nesprávne" z ERD

Vyberte správnu alternatívu k podmienke: cena NOT IN (2000,3000)

Vyberte správnu alternatívu k podmienke:

cena NOT IN (2000,3000)

- a. cena = 2000 AND cena = 3000
- b. cena <> 2000 OR cena <> 3000
- c. cena = 2000 OR cena = 3000
- d. cena <> 2000 AND cena <> 3000

Zrušiť moju volbu

Vyberte, čo musí platiť, aby bola databáza v 1. normálnej forme:

Vyberte, čo musí platiť, aby bola databáza v 1. normálnej forme:

Označte jednu alebo viac odpovedí:

- a. všetky primárne klúče musia byť číselné umelé klúče
- b. hodnoty všetkých stĺpcov v databáze musia byť atomické
- c. databáza nemôže obsahovať viachodnotové stĺpce
- d. databáza nemôže obsahovať zložené stĺpce

Uveďte údajový typ, ktorý vám umožní zadávať textové reťazce pevnej (fixnej) dĺžky rozsahu 20 znakov.

Uveďte údajový typ, ktorý vám umožní zadávať textové reťazce pevnej (fixnej) dĺžky rozsahu 20 znakov.

Odpoveď zapíšte bez medzier!

Odpoveď:

Ktoré z nasledujúcich jazykov patria do SQL štandardu?

Ktoré z nasledujúcich jazykov patria do SQL štandardu?

Označte jednu alebo viac odpovedí:

- a. Data Declaration Language - jazyk na deklaráciu štruktúry údajov
- b. Data Manipulation Language - jazyk na manipuláciu s údajmi
- c. Data Updating Language - jazyk na aktualizáciu údajov
- d. Data Definition Language - jazyk na definíciu štruktúry databázy
- e. Data Control Language - jazyk na riadenie prístupu k údajom

Vyberte, ktoré tvrdenia opisujú definíciu 3. normálnej formy:

Vyberte, ktoré tvrdenia opisujú definíciu 3. normálnej formy:

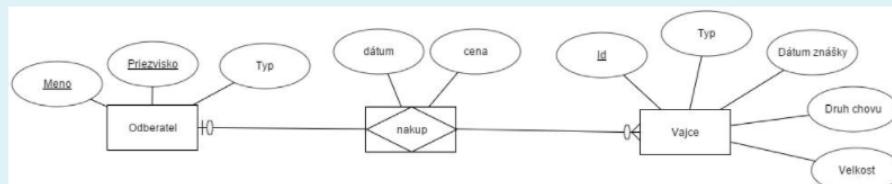
Označte jednu alebo viac odpovedí:

- a. primárny klúč nesmie byť zároveň cudzím klúčom
- b. neklúčové atribúty sú navzájom nezávislé
- c. neklúčové atribúty sú na primárnom klúči závislé priamo a nie tranzitívne
- d. v neklúčových atribútoch sa neopakujú rovnaké hodnoty
- e. primárny klúč nesmie byť zložený z viacerých atribútov

Na nasledujúcom obrázku máme ER model pre predaj vajec. Ktoré z nasledujúcich tvrdení sú pre daný diagram správne?

Otázka 5
Este nezodpovedané
Max. hodnotenie 3,00
Označiť otázku

Na nasledujúcom obrázku máme ER model pre predaj vajec. Ktoré z nasledujúcich tvrdení sú pre daný diagram správne?



Označte jednu alebo viac odpovedí:

- a. kedže stĺpec "typ" je aj v tabuľke "Odberatel" aj v tabuľke "Vajce", jeden z nich musí byť cudzí klúč - pravdepodobne ten v tabuľke odberateľa odkazuje na typ vajec, ktorý obľúbuje
- b. cudzí klúč na tabuľku "Odberatel" bude stĺpec, v ktorom bude kombinácia mena a priezviska obderateľa
- c. cudzí klúč na tabuľku "Vajce" bude stĺpec, ktorý sa bude odkazovať na stĺpec "id" z tabuľky "Vajce"
- d. cudzí klúč na tabuľku "Odberatel" sa bude skladáť z dvojice stĺpcov obsahujúcich hodnoty meno a priezvisko obderateľa
- e. správna kardinalita vzťahu reprezentovaného entitou "nakup" môže byť reprezentovaná aj cudzím klúčom v tabuľke "Vajce", ktorý bude odkazovať na tabuľku "Odberatel", avšak tabuľka "Vajce" potom musí mať aj stĺpce predstavujúce cenu a dátum nákupu vajca

Ktoré z nasledujúcich tvrdení o nulitnom prístupe k zabezpečeniu referenčnej integrity sú pravdivé?

Ktoré z nasledujúcich tvrdení o nulitnom prístupe k zabezpečeniu referenčnej integrity sú pravdivé?

Označte jednu alebo viac odpovedí:

- a. Ak je zmazaný záznam, na ktorý existuje referencia, je automaticky zmazaný aj referujúci záznam
- b. Ak je aktualizovaný záznam, na ktorý existuje referencia, databáza vyhodí výnimku
- c. Ak je zmazaný záznam, na ktorý existuje referencia, referencia sa nastaví na NULL
- d. Ak je aktualizovaný primárny kľúč záznamu, na ktorý existuje referencia, referencia sa nastaví na NULL
- e. Ak je aktualizovaná referencia na nejaký záznam, automaticky sa aktualizuje aj primárny kľúč, na ktorý sa referencia odkazovala

Vyberte pravdivé tvrdenia o INNER JOIN:

Vyberte pravdivé tvrdenia o INNER JOIN:

Označte jednu alebo viac odpovedí:

- a. Je to karteziánsky súčin, kde každý záznam z jednej tabuľky je spojený s každým záznamom z druhej tabuľky
- b. Výsledkom je kombinácia záznamov z jednej tabuľky so záznamami druhej tabuľky, pre ktoré platí podmienka v ON klauzule
- c. Vyžaduje sa, aby spájané tabuľky mali rovnakú relačnú schému, t.j. aby mali rovnaký počet stĺpcov a aby každý pricislúchajúci pári stĺpcov mal rovnaký (resp. kompatibilný) údajový typ
- d. Použitím klauzuly USING namiesto ON vieme určiť stĺpce, ktorých hodnoty musia byť rovnaké v oboch tabuľkách rovnaké, aby sa kombinácia záznamov vyskytla vo výsledku

Vyberte pravdivé tvrdenia o LEFT OUTER JOIN:

Vyberte pravdivé tvrdenia o LEFT OUTER JOIN:

Označte jednu alebo viac odpovedí:

- a. Neodporúča sa používať, vhodnou náhradou je INNER JOIN s USING, kde sa explicitne vymenujú stĺpce s rovnakým názvom, ktoré sa majú použiť pri spájaní
- b. Žiadny OUTER JOIN neumožňuje použiť USING klauzulu, pri OUTER JOINoch je povolená iba ON klauzula
- c. Vo výsledku sa objaví každý záznam z ľavej tabuľky aspoň raz
- d. Výsledkom je kombinácia záznamov z jednej tabuľky so záznamami druhej tabuľky, pre ktoré platí podmienka v ON klauzule, a aj záznamy z ľavej tabuľky, ktoré nemajú pári spĺňajúci podmienku, doplnené o NULL hodnoty

Vyberte pravdivé tvrdenia o RIGHT OUTER JOIN:

Vyberte pravdivé tvrdenia o RIGHT OUTER JOIN:

Označte jednu alebo viac odpovedí:

- a. Žiadny OUTER JOIN neumožňuje použiť USING klauzulu, pri OUTER JOINoch je povolená iba ON klauzula
- b. Neodporúča sa používať, vhodnou náhradou je INNER JOIN s USING, kde sa explicitne vymenujú stĺpce s rovnakým názvom, ktoré sa majú použiť pri spájaní
- c. Výsledkom je kombinácia záznamov z jednej tabuľky so záznamami druhej tabuľky, pre ktoré platí podmienka v ON klauzule, a vo výsledku sú aj záznamy z pravej tabuľky, ktoré nemajú pári spĺňajúci podmienku, doplnené o NULL hodnoty
- d. Je nahraditeľný LEFT OUTER JOINom, ak prehodíme ľavú tabuľku za pravú a naopak

Vyberte pravdivé tvrdenia o CROSS JOIN:

Vyberte pravdivé tvrdenia o CROSS JOIN:

Označte jednu alebo viac odpovedí:

- a. Je to karteziánsky súčin, kde každý záznam z jednej tabuľky je spojený s každým záznamom z druhej tabuľky
- b. Skrátený zápis pre CROSS JOIN je čiarka ,
- c. Vyžaduje sa, aby spájané tabuľky mali rovnakú relačnú schému, t.j. aby mali rovnaký počet stĺpcov a aby každý pricislúchajúci pári stĺpcov mal rovnaký (resp. kompatibilný) údajový typ
- d. Výsledkom je kombinácia záznamov z jednej tabuľky so záznamami druhej tabuľky, pre ktoré platí podmienka v ON klauzule

SQL dopyt, ktorý vráti všetky oblasti, ich názov, populáciu a id štátu ktorému patria(), ktoré majú viac ako 100.000 (stotisíc) obyvateľov

Chážka 11

Odpoveď bola uložená

Max. hodnotenie 3,00

▼ Označit otázku

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti všetky oblasti, ich názov, populáciu a id štátu ktorému patria (len tieto 3 stĺpce v takomto poradí), ktoré majú viac ako 100.000 (stotisíc) obyvateľov a zoradte ich podľa populácie od najmenšej.

Vo vzorovej DB 9 oblastí.



```

select nazov, populacia, id_stat from oblast
where populacia > 100000
order by populacia
  
```

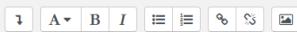
SQL dopyt, ktorý vráti všetky oblasti a počet ich obyvateľov žijúcich mimo miest v danej oblasti.

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti všetky oblasti a počet ich obyvateľov žijúcich mimo miest v danej oblasti (podľa obsahu vzorovej DB).

- Výsledný select bude obsahovať 2 stĺpce – názov oblasti a jeho populáciu žijúcu mimo miest v uvedenom poradí, druhý stĺpec pomenujte na_dedine.
- Výsledok zoradte zopustne podľa vypočítaného počtu obyvateľov, s NULL hodnotami na konci.

Vo vzorovej DB výsledok dopytu vráti 9 riadkov, iba pre oblasť Mesto Kyjev vyjde hodnota v stĺpci na_dedine na 0.



```

SELECT o.nazov, o.populacia - (SELECT SUM(m.populacia) FROM mesto m WHERE m.id = o.id_oblast) AS na_dedine
FROM oblast o
ORDER BY na_dedine ASC
  
```

SQL dopyt, ktorý vráti názov a čas vzniku pre presne dva podľa populácie najväčšie štáty, ale vyberajte len z tých štátov ktoré nemajú poznámku.

Otázka 12

Odpoveď bola uložená.

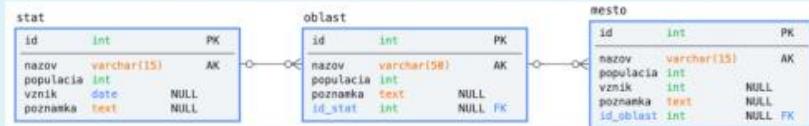
Max.

hodnotenie 3,00

Označiť otázku

Vytvorte SQL dopyt, ktorý vráti názov a čas vzniku pre presne dva podľa populácie najväčšie štátu, ale vyberajte len z tých štátov ktoré nemajú poznámku.

Tieto dva štáty môžu zostať zoradené od populáciou najväčšieho po populáciou najmenší štát.



```

select nazov, vznik from stat
where poznamka is null
order by populacia desc
limit 2
    
```

SQL dopyt, ktorý vráti počet miest v jednotných oblastiach. Výsledný select bude obsahovať 2 stĺpce – počet miest a názov príslušnej oblasti

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napište SQL dopyt, ktorý vráti počet miest v jednotných oblastiach.
- Výsledný select bude obsahovať 2 stĺpce – počet miest a názov príslušnej oblasti.
- Pomenujte ich ako pocet a oblast, v uvedenom poradí, bez diakritiky.
- Výsledok zoradte zopustne.

Vo vzorovej DB výsledok dopytu vráti 5 riadkov.



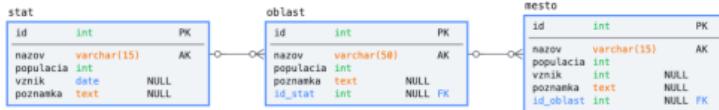
```

select count(m.id) as pocet, oblast.nazov as oblast from oblast
join mesto m on oblast.id = m.id_oblast
group by oblast.nazov
order by pocet desc
    
```

SQL dopyt, ktorý vráti názov štátu a priemernú populáciu miest štátu, ktorého priemerný počet obyvateľov v mestách je najvyšší

- Napište SQL dopyt, ktorý vráti názov štátu a priemernú populáciu miest štátu, ktorého priemerný počet obyvateľov v mestách je najvyšší.
- Výsledný select bude obsahovať 2 stĺpce – názov štátu a jeho priemernú populáciu miest v uvedenom poradí.
- Pomenujte ich ako stat a priemer, v uvedenom poradí, bez diakritiky.

Vo vzorovej DB výsledok dopytu vráti 1 riadok.



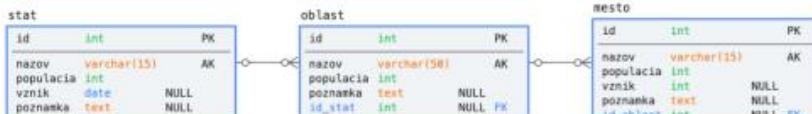
```

select stat.nazov as stat, avg(m.populacia) as priemer from stat
join oblast o on stat.id = o.id_stat
join mesto m on o.id = m.id_ostat
group by stat.nazov
order by priemer desc limit 1;
    
```

SQL dopyt, ktorý vráti všetky mestá a ich počet obyvateľov (len tieto 2 stĺpce), z oblasti v ktorej súčet obyvateľov jej všetkých miest aj najnižší.

- Napište SQL dopyt, ktorý vráti všetky mestá a ich počet obyvateľov (len tieto 2 stĺpce), z oblasti v ktorej súčet obyvateľov jej všetkých miest je najnižší.
- Výsledok zoradťte zostupne.
- Výsledný select bude obsahovať 2 stĺpce – názov mesta a jeho populáciu v uvedenom poradí, bez aliasov.

Vo vzorovej DB výsledok dopytu vráti 3 riadky.



```

select nazov, populacia from mesto
where id_ostat = ( select oblast.id from oblast
join mesto m on oblast.id = m.id_ostat
group by oblast.id
order by sum(m.populacia) limit 1);
    
```

SQL dopyt, ktorý vráti počet oblastí v jednotlivých štátoch. Výsledný select bude obsahovať 2 stĺpce – počet oblastí a názov príslušného štátu.

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

- Napíšte SQL dopyt, ktorý vráti počet oblastí v jednotlivých štátoch.
- Výsledný select bude obsahovať 2 stĺpce – počet oblastí a názov príslušného štátu.
- Pomenujte ich ako pocet a stat, v uvedenom poradí, bez diakritiky.
- Výsledok zoradte zostupne podľa populácie štátu.

Vo vzorovej DB výsledok dopytu vráti 5 riadkov

```

stat
+----+-----+-----+-----+
| id | int  | PK   |
| nazov | varchar(15) | AK   |
| populacia | int |       |
| vznik | date | NULL |
| poznamka | text | NULL |
+----+-----+-----+-----+
oblast
+----+-----+-----+-----+
| id | int  | PK   | |
| nazov | varchar(50) | AK   |
| populacia | int |       |
| poznamka | text | NULL |
| id_stat | int | NULL | FK |
+----+-----+-----+-----+
mesto
+----+-----+-----+-----+
| id | int  | PK   | |
| nazov | varchar(15) | AK   |
| populacia | int |       |
| vznik | int | NULL |
| poznamka | text | NULL |
| id_oblast | int | NULL | FK |
+----+-----+-----+-----+

```

SELECT COUNT(o.id) AS pocet, s.nazov AS stat
FROM stat s
LEFT JOIN oblast o ON s.id = o.id_stat
GROUP BY s.nazov, s.populacia
ORDER BY s.populacia DESC

SQL dopyt, ktorý vráti všetky oblasti, ich názov a populáciu (len tieto 2 stĺpce v takomto poradí), ktoré majú menej ako 1 milión obyvateľov a zoradte ich podľa populácie od najmenšieho.

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti všetky oblasti, ich názov a populáciu (len tieto 2 stĺpce v takomto poradí), ktoré majú menej ako 1 milión obyvateľov a zoradte ich podľa populácie od najmenšieho.

Vo vzorovej DB sú 3 takéto oblasti.

```

stat
+----+-----+-----+-----+
| id | int  | PK   |
| nazov | varchar(15) | AK   |
| populacia | int |       |
| vznik | date | NULL |
| poznamka | text | NULL |
+----+-----+-----+
oblast
+----+-----+-----+-----+
| id | int  | PK   | |
| nazov | varchar(50) | AK   |
| populacia | int |       |
| poznamka | text | NULL |
| id_stat | int | NULL | FK |
+----+-----+-----+
mesto
+----+-----+-----+-----+
| id | int  | PK   | |
| nazov | varchar(15) | AK   |
| populacia | int |       |
| vznik | int | NULL |
| poznamka | text | NULL |
| id_oblast | int | NULL | FK |
+----+-----+-----+

```

SELECT nazov, populacia
FROM oblast
WHERE populacia < 1000000
ORDER BY populacia ASC

SQL dopyt, ktorý vráti všetky štáty, ich názov a populáciu v miliónoch (len tieto 2 stĺpce v takomto poradí, pri populácii uveďte aj desatinu časť v celej dĺžke). **float8 miesto float4**

Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napište SQL dopyt, ktorý vráti všetky štáty, ich názov a populáciu v miliónoch (len tieto 2 stĺpce v takomto poradí, pri populácii uveďte aj desatinu časť v celej dĺžke). Výsledky zoradte podľa dĺžky názvu štátu začínajúc najkratším názvom, a v prípade rovnakej dĺžky podľa abecedy vzostupne.

Vo vzorovej DB by bol druhý riadok „Polsko 38.4336“ a Slovensko by bolo posledné v poradí.



SELECT s.nazov, float8(s.populacia)/1000000 as populacia

FROM stat s

ORDER BY length(s.nazov), s.nazov

