

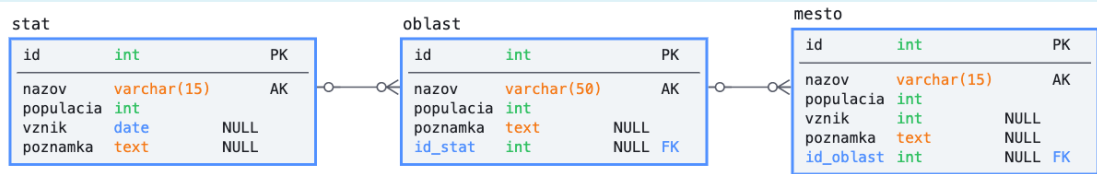
1. Vytvorte dopyt ktorý vráti 2 stĺpce.

V prvom budú názvy takých štátov ktoré majú poznámku a okrem názvov takýchto štátov tu budú aj názvy všetkých miest ktoré nezačínajú písmenom K.

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 zoradte 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



```
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;
```

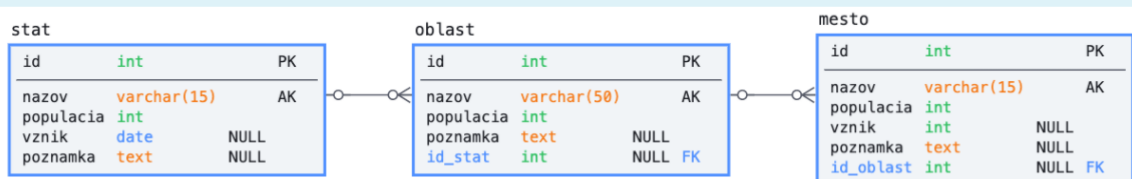
2. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

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

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 oblast 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 || '%';
```

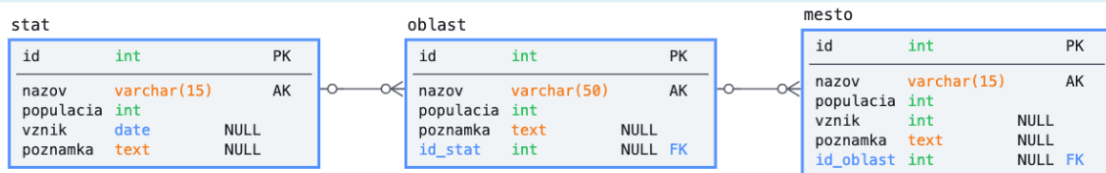
3. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napište 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.

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

- Napište 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 zoradíte 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;
```

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

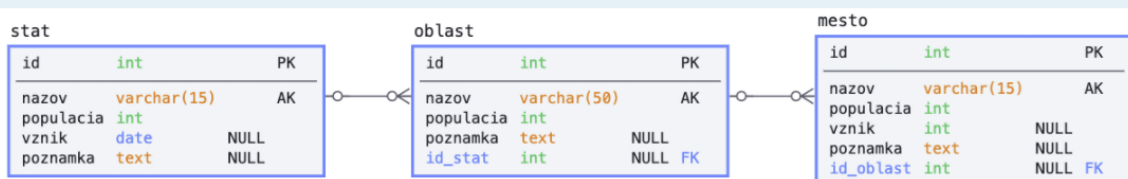
Napište SQL dopyt, ktorý vráti počet miest v jednotných oblastiach.

Výsledný select bude obsahovať 2 stĺpce – počet miest a názov príslušnej oblasti.

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 zoradíte 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;
```

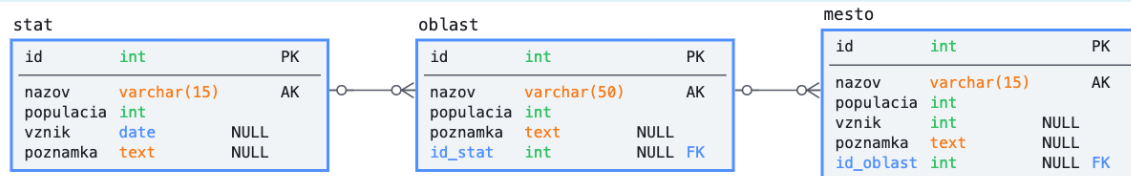
5. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti všetky štáty, ich názov a populáciu v miliónoch (len tieto 2 stĺpce v takomto poradí, pri populácii uveďte aj desatinnú časť v celej dĺžke).
V takomto poradí, pri populácii uveďte aj desatinnú časť v celej dĺžke).

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 desatinnú časť v celej dĺžke). Výsledky zoradíte 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 nazov, cast((populacia) as float)/1000000 as populacia
FROM stat
ORDER BY length(nazov), nazov;
```

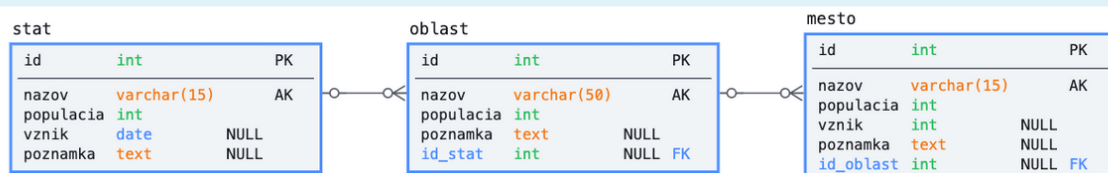
6. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

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.

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 zoradíte 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 taketo 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;
```

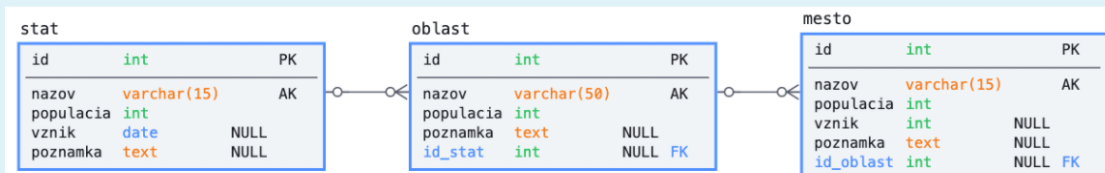
7. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

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 zoradíte ich podľa vzniku od najmladšieho.

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

Napíšte SQL dopyt, ktorý vráti mestá, ich názov, populáciu a kedy vznikli (len tieto 3 stĺpce v takomto poradí), ktoré majú viac ako 100 tisíc obyvateľov a zoradíte 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;
```

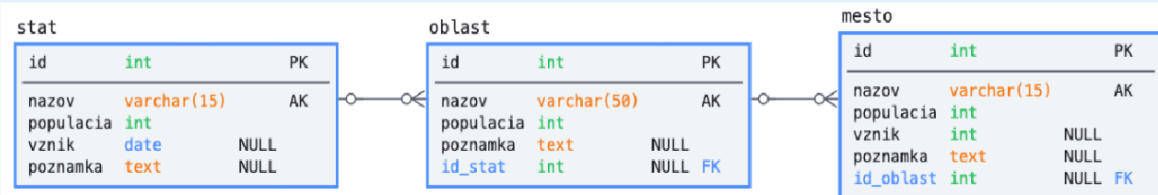
8. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti názvy všetkých miest, oblasti a štátu do ktorého patria pre mestá, ktoré majú uvedenú poznámku.

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 zoradíte 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;
```

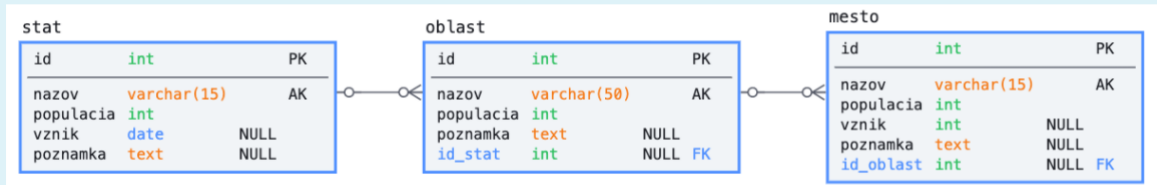
9. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

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

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_oblast
group by s.id
order by priemer desc
limit 1;
```

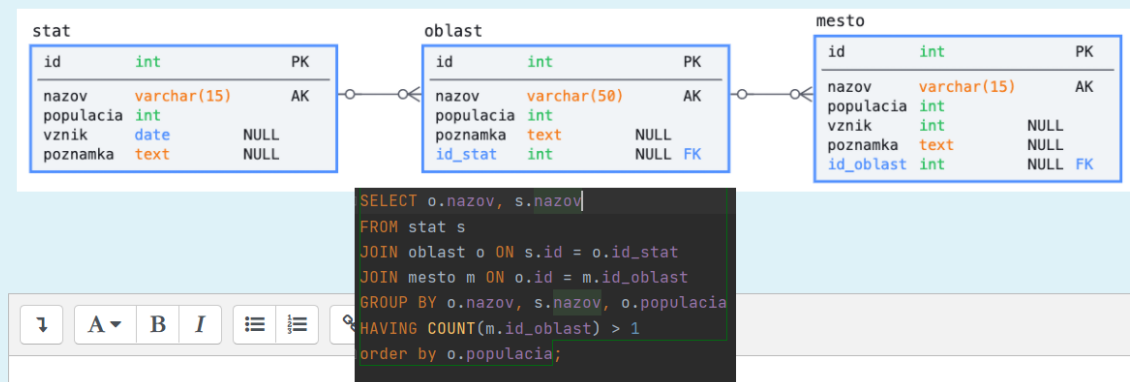
10. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

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 zoradíte ich podľa populácie od najmenšej.

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 zoradíte 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.



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

Napíšte 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).

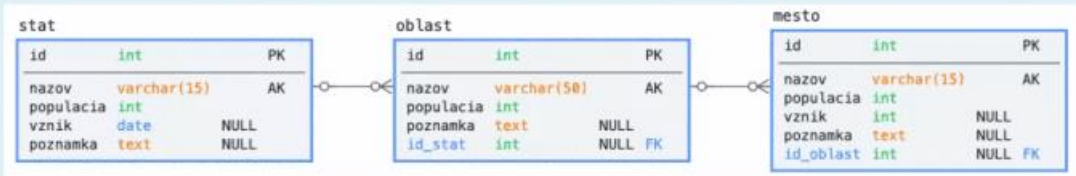


12. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

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

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;
```

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

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.

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 zoradíte 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;

```

14. 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 zoradíte ich podľa populácie od najmenšej.

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 zoradíte ich podľa populácie od najmenšej.

Vo vzorovej DB 9 oblastí.

```

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

```

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

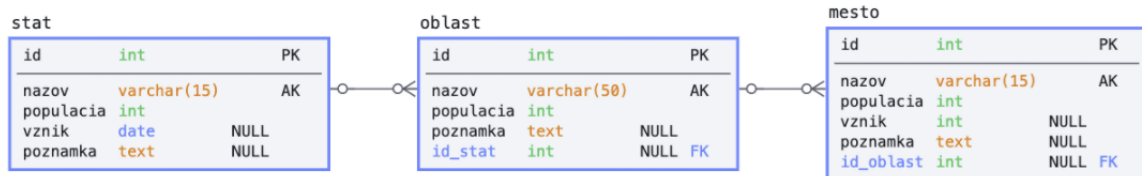
Napíšte SQL dopyt, ktorý vráti počet miest v jednotných oblastiach.

Výsledný select bude obsahovať 2 stĺpce – názov oblasti a počet miest.

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 zoradíte 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;
```

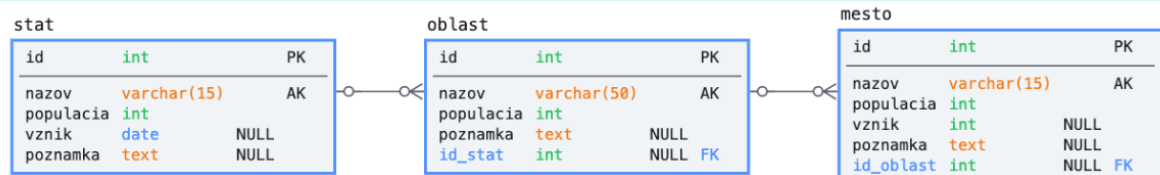
16. Pracujeme s databázou, ktorej relačný model je zobrazený na nasledujúcej schéme.

Napíšte SQL dopyt, ktorý vráti pre každý štát počet miest v jednotlivých štátoch.

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 zoskupíte 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;
```

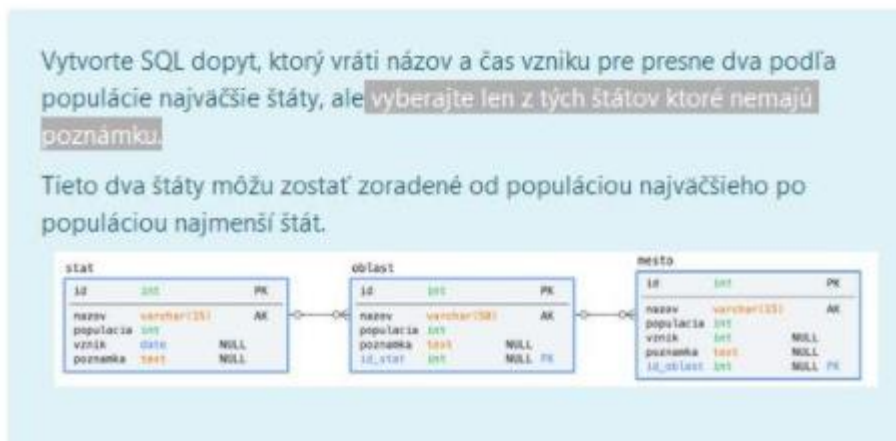
17. Vytvorte SQL dopyt, ktorý vráti názov a čas vzniku pre presne dva podľa populácie najväčšie štáty, ale vyberajte len z tých štátov ktoré nemajú poznámku.

```
Select nazov, vznik
```

```
FROM stat
```

```
where stat.poznamka is null
```

```
order by populacia DESC limit 2;
```



18. Zobrazte z tabuľky OE.orders roky (order_date) v kt. bolo viac ako 10 objednávok.

```
Select order_date, count(*)
```

```
from orders
```

```
group by order_date
```

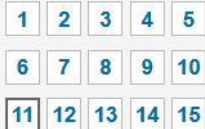
```
having count(*)>10
```

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

Napište SQL dopyt, ktorý vráti počet features, ktoré majú ohlásený nejaký bug.

(Doplniť odpoveď)

NAVIGÁCIA V TESTE



Ukončiť pokus ...

Čas zostávajúci do
ukončenia testu

0:58:03

Otázka 11

Ešte

nezodpovedané

Max. hodnotenie

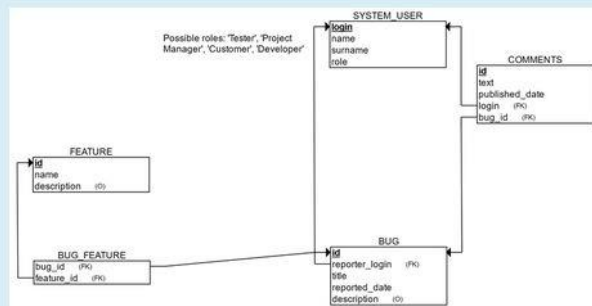
7,00

Označiť otázku

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

Napište SQL dopyt, ktorý vráti počet features, ktoré majú ohlásený nejaký bug. Stĺpec 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

Odpoveď:

20. Pracujeme s databázou pre manažment softvérových bugov, ktorej 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).

(Doplniť odpoveď)

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

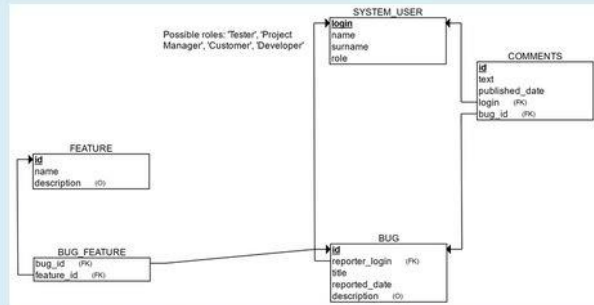
7,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktorej 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

Odpoveď:

21. (Doplniť odpoveď)

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

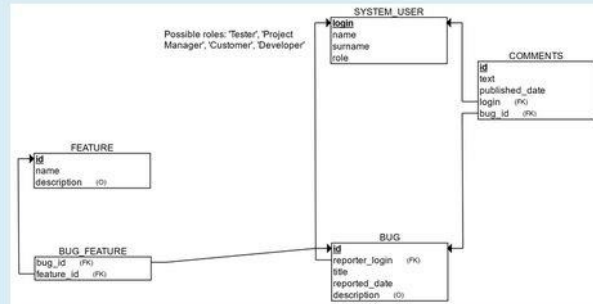
7,00

Označiť otázku

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

Napište SQL dopyt, ktorý vráti v jednom stĺpci všetky dátumy, kedy bol publikovaný nejaký komentár, alebo ohlásený 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

22. (Doplniť odpoveď)

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

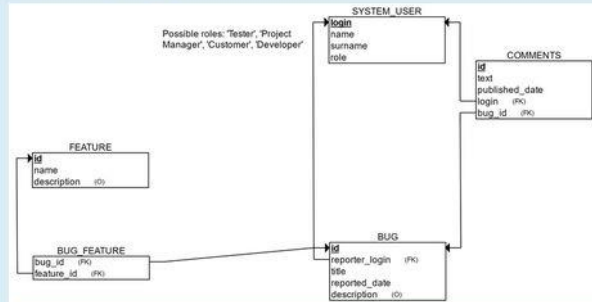
9,00

Označiť otázku

Pracujeme s databázou pre manažment softvérových bugov, ktorej 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

23. (Doplniť odpoveď)

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

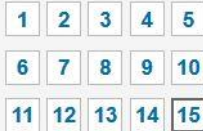
Vo vzorovej DB je vo výsledku napr. záznam:

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

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

24. (Doplniť odpoveď)

NAVIGÁCIA V
TESTE



Ukončiť pokus ...

Čas zostávajúci do
ukončenia testu
0:57:44

Otázka 15

Ešte
nezodpovedané

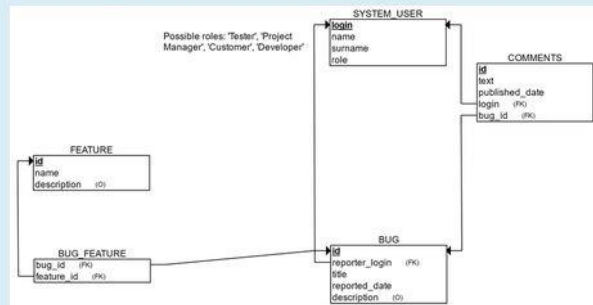
Max. hodnotenie
10,00

Označiť otázku

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

Napište 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

25. (Doplniť odpoveď)

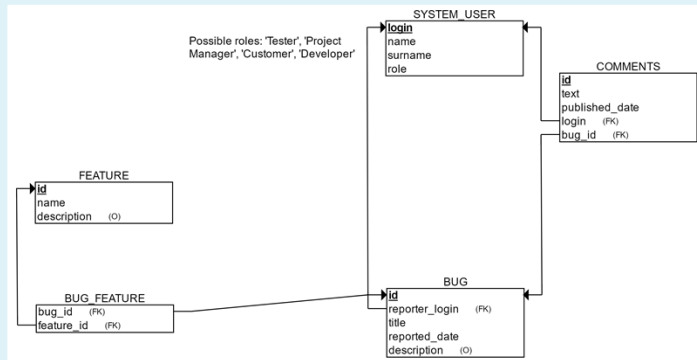
Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. **Napište SQL dopyt, ktorý vráti zoznam názvov features (stĺpec name), ktoré nemajú v roku 2015 ohlásený žiaden bug.**

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

26. (Doplniť odpoveď)

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

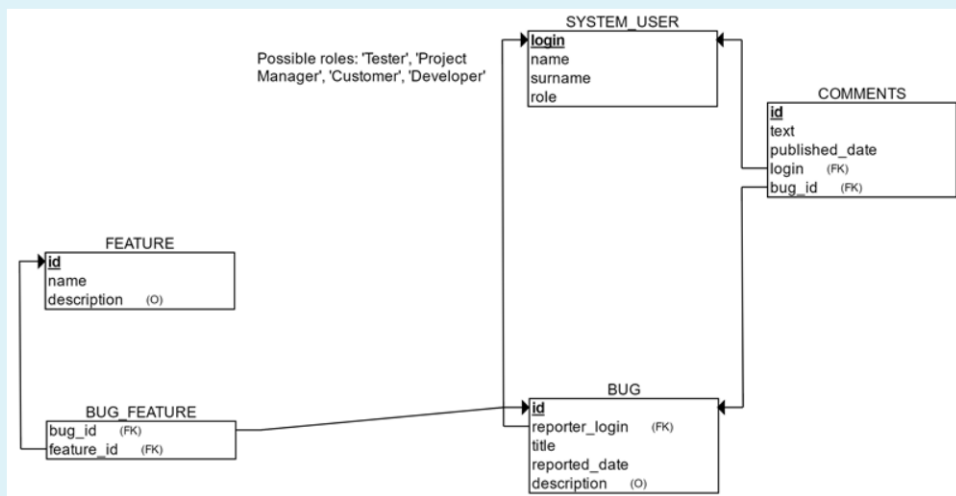
Pozor, vrátený má byť typ *dátum*, nie dátum formátovaný ako reťazec.



27. (Doplniť odpoveď)

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 komentármi. Pozor, rátajte s tým, že používateľov s rovnakým menom 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`.

28. (Doplniť odpoveď)

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti zoznam názvov features (stĺpec name), ktoré nemajú v roku 2015 ohlásený žiaden bug. Vo vzorovej DB sú to E-mail notifications, Social networks integration, Performance monitoring, HipChat notifikácie.

Possible roles: 'Tester', 'Project Manager', 'Customer', 'Developer'

V prípade potreby vyjadrenia dátumu dajte p

29. (Doplniť odpoveď)

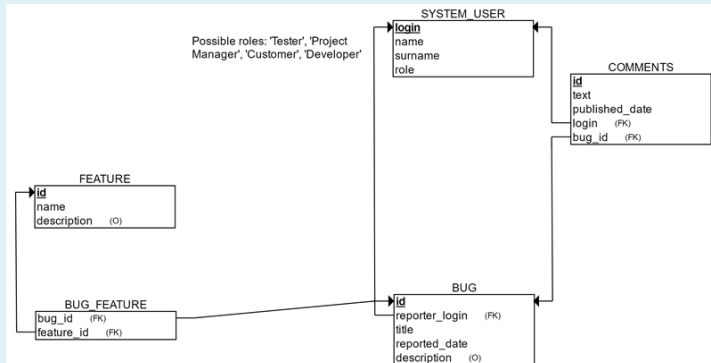
Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. Napíšte SQL dopyt, ktorý vráti zoznam názvov features (názov stĺpca dajte "FEATURE"), a počet ich bugov (názov stĺpca dajte "POCET"). Zobrazte aj tie features, ktoré nemajú žiadny bug, pričom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

Possible roles: 'Tester', 'Project Manager', 'Customer', 'Developer'

V prípade potreby vyjadrenia dátumu dajte p

30. (Doplniť odpoveď)

Pracujeme s databázou pre manažment softvérových bugov, ktorej relačný model je naznačený na nasledujúcej schéme. **Napište SQL dopyt, ktorý vráti name (názov stĺpca "MENO") a surname (názov stĺpca "PRIEZVISKO") používateľov, ktorých login obsahuje c. Zoradte výsledok abecedne podľa priezviska.** Vo vzorovej DB sú to 4 používatelia.



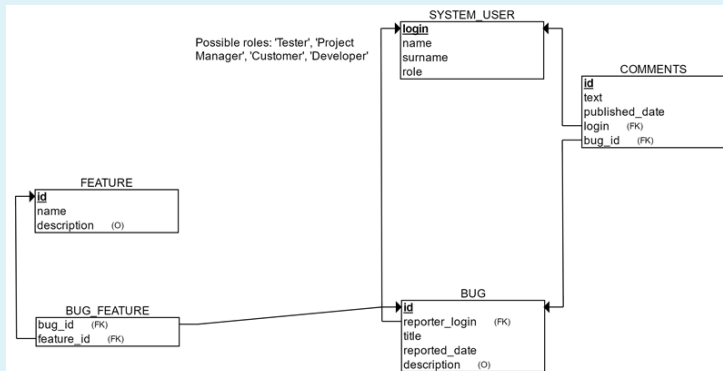
31. (Doplniť odpoveď)

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

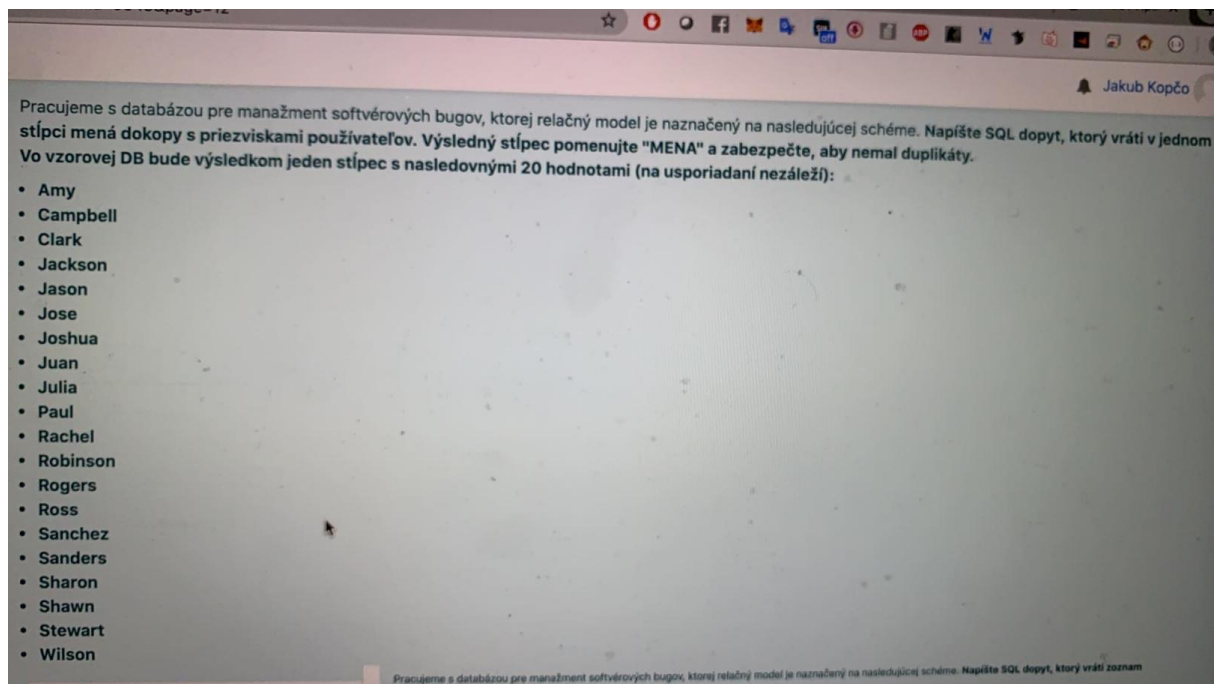
Vo vzorovej DB je vo výsledku napr. záznam:

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

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



32. (Doplniť odpoved')



33. Zobrazte meno (cust_first_name) a priezvisko (cust_last_name) zákazníkov (OE.customers), ktorých účet manažuje (account_mgr_id) zamestnanec (HR.employees) s menom Gerald Cambrault. Zákazníkov zorad'te abecedne najprv podľa priezviska, potom podľa mena.

Odpoveď:

```
select cust_first_name, cust_last_name
from oe.customers
where account_mgr_id = (select employee_id
                        from hr.employees
                        where first_name = 'Gerald' and last_name='Cambrault' )
ORDER BY cust_last_name,cust_first_name;
```

--Selects

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

--1.

--Napište 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 || '%';

--2.

--Napište SQL dopyt, ktorý vráti všetky štáty, ich názov a populáciu v miliónoch (len tieto 2 stĺpce v takomto poradí, pri populácii

-- uveďte aj desatinnú časť v celej dĺžke). Výsledky zoradíte 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 nazov, CAST((populacia) AS FLOAT)/1000000 AS populacia
FROM stat
ORDER BY length(nazov), nazov;
```

```
-----
-----
-----
-----
-----
-----
```

--3.

--Napište 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 10 000 000

--Výsledok zoradíte 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;
```

```
-----
-----
```


--4.

--Napište SQL dopyt, ktorý vráti všetky mestá, ich názov, populáciu a kedy vznikli (len tieto 3 stĺpce v takomto poradí), ktoré majú viac ako 100 000 obyvateľov

--a zoradíte ich od najmladšieho

--Vo vzorovej DB sú 3 takéto mestá.

SELECT nazov, populacia, vznik

FROM mesto

WHERE populacia > 100000

ORDER BY vznik DESC;

--5.

--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 zoradíte abecedne podľa názvov v prvom stĺpci.

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

```
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;
```

```
-----
-----
-----
-----
-----
-----
```

--6.

--Napište SQL dopyt, ktorý vráti názvy všetkých miest, oblastí a štátu do ktorého patria pre mestá, ktoré majú uvedenú poznámku.

--Výsledok zoradíte 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 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;
```

```
-----
-----
-----
-----
-----
-----
```

--7.

--Napište SQL dopyt, ktorý vráti názvov 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_oblast
GROUP BY s.id
ORDER BY priemer DESC
LIMIT 1;
```

```
-----
-----
```


--8.

--Napište 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 zoradíte 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;

--9.

--Napište 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 zoradíte 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 riadky.

```
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_oblast
GROUP BY o.nazov, s.nazov, o.populacia
HAVING COUNT(m.id_oblast) > 1
ORDER BY o.populacia;
```

```
-----
-----
-----
-----
-----
-----
```

--10.

--Napište SQL 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 prvom stĺpci namiesto názvu mesta NULL).

--Výsledok zoradíte 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;

--11.

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

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

--12.

--Napište 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;
```

```
-----
-----
-----
-----
-----
-----
```

--13.

--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 zoradíte 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 populacia DESC;
```

```
-----
-----
-----
-----
-----
-----
```

--14.

--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 obyvateľov a zoradíte ich podľa populácie od najmensej.

--Výsledný select nech obsahuje 3 stĺpce: názov štátu, populáciu a dátum vzniku, v uvedenom poradí, bez aliasov.

--Vo vzorovej DB 9 oblastí.

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

```
-----
-----
```


--15.

--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 zoradíte 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;

--16

--Napište 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 zoradíte 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;
```