

Otázka 2
Ešte nezodpovedané
Max. hodnotenie 5,00
Označiť otázku

Uvedte čím sa začína príkaz na zavolanie používateľom vytvorenej funkcie. Napíšte len jedno slovo.

Odpoveď:

Відповідь: SELECT

vedané
nie 5,00
čiť

Ktorá agregáčna funkcia v SQL slúži na spočítanie počtu riadkov v tabuľke? Odpoveď uveďte bez zátvoriek.

Odpoveď:

Відповідь: COUNT

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

Medzi údajové modely patrí:

Označte jednu alebo viac odpovedí:

- ☐ redundantný
- ☐ atomický
- ☐ implementačný
- ☐ konceptuálny

Відповідь:

- implementačný,
- konceptuálny

Otázka 4
Odpoveď bola uložená
Max. hodnotenie 5,00
Označiť otázku

Vyberte vlastnosti primárneho kľúča.

Označte jednu alebo viac odpovedí:

- ☐ jednoznačne sa odkazuje na záznam v inej tabuľke
- ☐ môže byť umelý, vytvorený špecificky pre identifikáciu entít
- ☐ jeho hodnota musí byť jedinečná v rámci všetkých tabuliek databázy
- ☐ jeho hodnota musí byť jedinečná v rámci danej tabuľky

Відповідь:

- môže byť umelý, vytvorený špecificky pre identifikáciu entít
- jeho hodnota musí byť jedinečná v rámci danej tabuľky

5

odpovedané

notenie 5,00

Označiť
zku

Vyberte správne tvrdenia.

Označte jednu alebo viac odpovedí:

- ☐ RAISE NOTICE zastaví vykonávanie tela funkcie
- ☐ Storovaná procedúra vracia výsledok, pričom funkcia nie.
- ☐ PostgreSQL umožňuje viacerým funkciám zdieľať rovnaké meno, pokiaľ majú odlišné argumenty
- ☐ Funkcia môže na výstup vrátiť tabuľku v prípade, že je definovaná s RETURNS TABLE

Відповідь:

- PostgreSQL umožňuje viacerým funkciám zdieľať rovnaké meno, pokiaľ majú odlišné argumenty
- Funkcia môže na výstup vrátiť tabuľku v prípade, že je definovaná s RETURNS TABLE

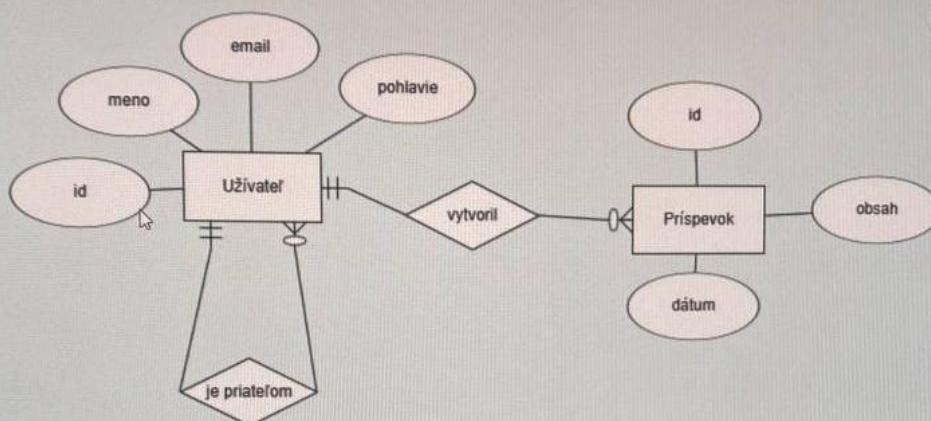
6

ovedané

tenie 5,00

acit'

Majme daný ERD zobrazený na nasledujúcom obrázku.



Vyberte pravdivé tvrdenia:

Označte jednu alebo viac odpovedí:

- ☐ Vzťah "vytvoril" je binárny vzťah.
- ☐ Vzťah "je priateľom" je ternárny vzťah.
- ☐ Vzťah "je priateľom" je rekurzívny vzťah.
- ☐ Vzťah "vytvoril" je rekurzívny vzťah.

Відповідь:

- Vzťah „vytvoril“ je binárny vzťah.
- Vzťah „je priateľom“ je rekurzívny vzťah.

tion 6

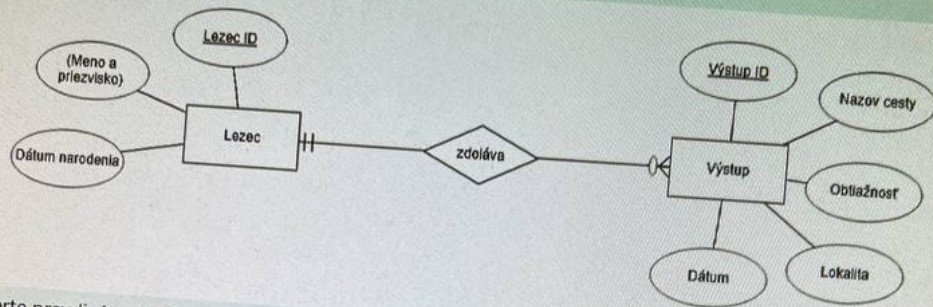
et

red

d out of

question

Majme daný ERD zobrazený na nasledujúcom obrázku.



Vyberte pravdivé tvrdenia:

Select one or more:

- ☐ všetky atribúty sú atomické
- ☐ v relačnom modeli bude tento vzťah vyjadrený cudzím kľúčom na strane Výstupu
- ☐ každý výstup môže byť realizovaný viacerými lezcami
- ☐ každý lezec môže mať viacero výstupov, ale každý výstup musí byť realizovaný len jedným lezcom

page

Відповідь:

- v relačnom modeli bude tento vzťah vyjadrený cudzím kľúčom na strane Výstupu
- každý lezec môže mať viacero výstupov, ale každý výstup musí byť realizovaný len jedným lezcom

Question 5

Not yet answered

Marked out of 5.00

Flag question

Pomocou ktorej klauzuly je možné filtrovať záznamy:

Select one or more:

- ☐ TRUNCATE
- ☐ GROUP BY
- ☐ WHERE
- ☐ HAVING

Відповідь:

- WHERE
- HAVING

Question 4

Not yet answered

Marked out of 5.00

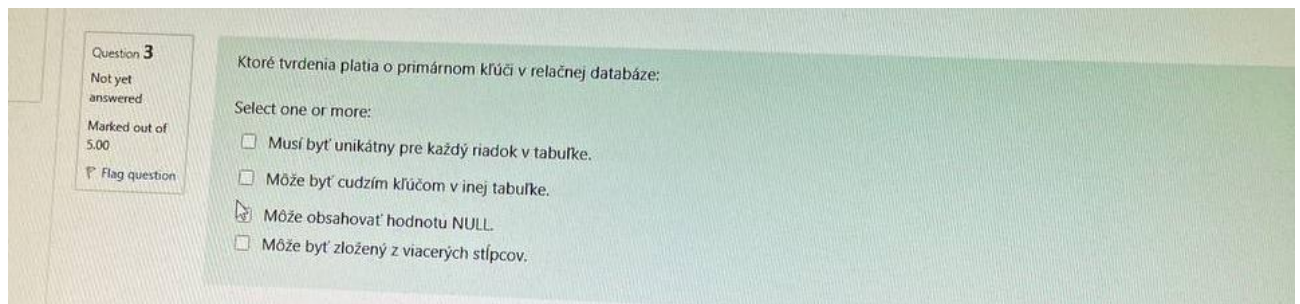
Flag question

Ktoré tvrdenia platia o cudzom kľúči (foreign key) v relačnej databáze:

Select one or more:

- ☐ Môže byť zároveň primárnym kľúčom v tej istej tabuľke.
- ☐ Musí byť typu integer.
- ☐ Nesmie obsahovať žiadne duplicitné hodnoty.

Відповідь: Môže byť zároveň primárnym kľúčom v tej istej tabuľke.



Відповідь:

- Musí byť unikátny pre každý riadok v tabuľke.
- Môže byť cudzím kľúčom v inej tabuľke.
- Môže byť zložený z viacerých stĺpcov.

Ktoré z nasledujúcich tvrdení platia pre modifikovateľné pohľady?

Select one or more:

- ☐ SELECT vo view nesmie obsahovať príkaz DISTINCT.
- ☐ Pohľad môže obsahovať GROUP BY, ak je v ňom len jedna tabuľka.
- ☐ Pohľad možno urobiť modifikovateľným pomocou trigrov.
- ☐ Všetky stĺpce tabuľky s NOT NULL obmedzením musia byť zahrnuté vo view.

Відповідь:

- SELECT vo view nesmie obsahovať príkaz DISTINCT.
- Pohľad možno urobiť modifikovateľným pomocou trigrov.
- Všetky stĺpce tabuľky s NOT NULL obmedzením musia byť zahrnuté vo view.

Vyberte správne tvrdenia.

Select one or more:

- ☐ Premenná NEW v trigger funkcii má dátový typ RECORD
- ☐ Trigger funkcia môže mať ľubovoľnú návratovú hodnotu
- ☐ Trigger funkcia sa vytvára ako regulárna funkcia, avšak nesmie brať žiadne parametre
- ☐ RAISE EXCEPTION zastaví vykonávanie tela funkcie

Відповідь:

- Premenná NEW v trigger funkcii má dátový typ RECORD
- Trigger funkcia sa vytvára ako regulárna funkcia, avšak nesmie brať žiadne parametre
- RAISE EXCEPTION zastaví vykonávanie tela funkcie

Uveďte množinovú operáciu, ktorá v PostgreSQL vytvorí zjednotenie bez duplikátov.

Answer:

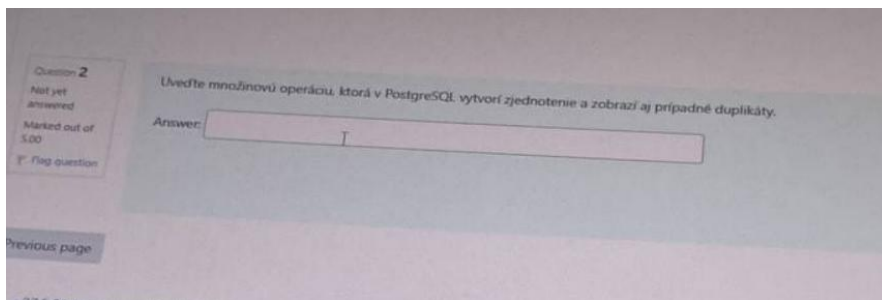
Відповідь: UNION

Použitím akého kľúčového slova sa začína transakcia v PostgreSQL?

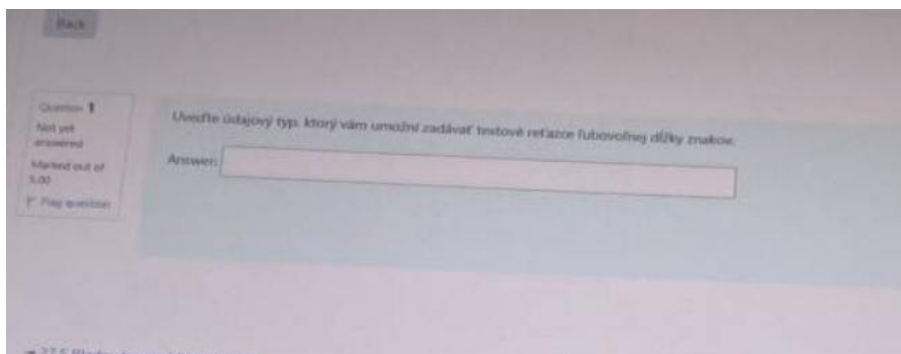
Answer:

BEGIN

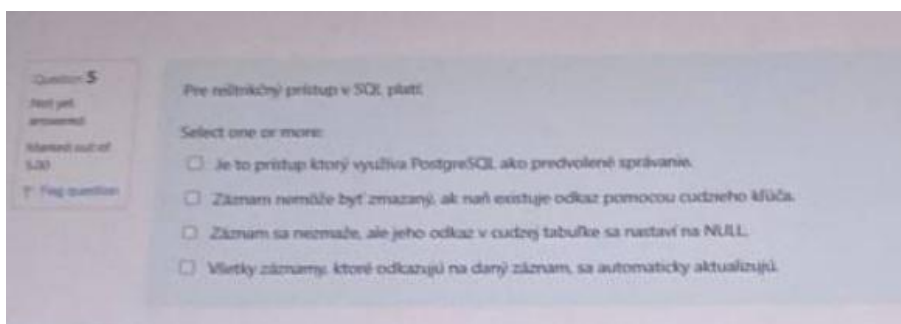
Відповідь: BEGIN



Відповідь: UNION ALL



Відповідь: TEXT



Відповідь:

- Je to prístup ktorý využíva PostgreSQL ako predvolené správanie.
- Záznam nemôže byť zmazaný, ak naň existuje odkaz pomocou cudzieho kľúča.

Question 6
Not yet answered
Marked out of 5.00
Flag question

Majme daný ERD zobrazený na nasledujúcom obrázku.

Vyberte pravdivé tvrdenia:

Select one or more:

- ☐ Entita Futbalista sa prevedie na tabuľku s primárnym kľúčom.
- ☐ Atribúty z entít sa stanú stĺpcami v tabuľkách.
- ☐ Cudzí kľúč sa bude nachádzať v tabuľke Futbalista
- ☐ Vzťah "hra v" sa v relačnom modeli vždy reprezentuje ako samostatná tabuľka

Previous page

Відповідь:

- Entita Futbalista sa prevedie na tabuľku s primárnym kľúčom.
- Atribúty z entít sa stanú stĺpcami v tabuľkách.
- Vzťah „hra v“ sa v relačnom modeli vždy reprezentuje ako samostatná tabuľka.

Time left 0:11:56

Majme daný RM pre jednoduchú evidenciu príspevkov sociálnej siete zobrazený na nasledujúcom obrázku.

Vyberte pravdivé tvrdenia:

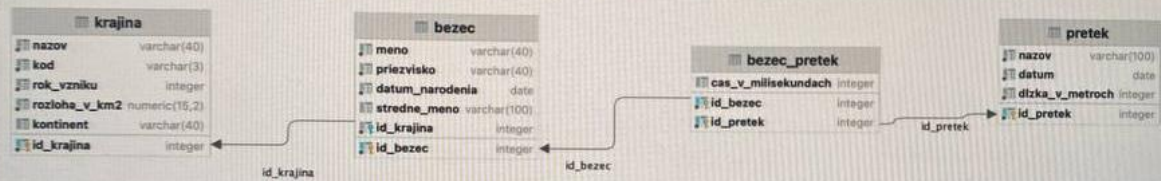
Select one or more:

- ☐ Z databázy je možné zistiť, ktoré tagy používa ktorý používateľ
- ☐ „nazov_tagu“ je voliteľný (optional) atribút
- ☒ Príspevok s „id_rodic“ rovný null nie je komentárom iného príspevku
- ☒ Príspevok k sebe môže mať viac krát priradený ten istý tag

Відповідь:

- Z databázy je možné zistiť, ktoré tagy používa ktorý používateľ
- Príspevok s „id_rodic“ rovný null nie je komentárom iného príspevku

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



Napište SQL dopyt, ktorý vráti meno, priezvisko a názov krajiny bežca z kontinentu Európa, ale vynechajte krajinu Česko. Stĺpce pomenujte meno, priezvisko, stat (bez diakritiky) a nech sú v tomto poradí. Zoradte stĺpce podľa mena vzostupne (A-Z) potom podľa priezviska zostupne (Z-A) a nakoniec podľa názvu krajiny vzostupne (A-Z).

Výsledný dopyt vráti 15 riadkov.

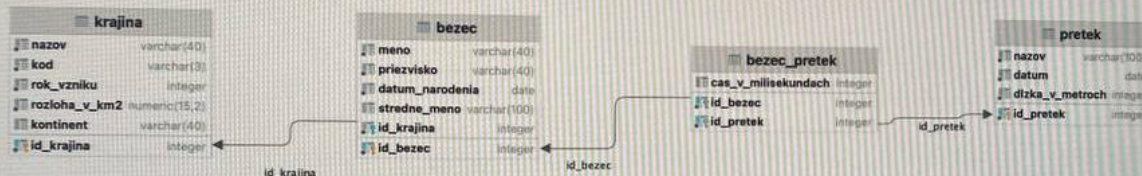
Відповідь:

```
SELECT
    b.meno AS meno,
    b.priezvisko AS priezvisko,
    k.nazov AS stat
FROM bežec b
JOIN krajina k ON b.id_krajina = k.id_krajina
WHERE k.kontinent = 'Európa'
    AND k.nazov <> 'Česko'
ORDER BY
    b.meno ASC, -- A->Z
    b.priezvisko DESC, -- Z->A
    k.nazov ASC; -- A->Z
```

Перевірка:

	meno ▾	priezvisko ▾	stat ▾
1	Andriy	Shevchenko	Ukrajina
2	Dmytro	Ivanov	Ukrajina
3	Fritz	Weber	Nemecko
4	George	Harris	Veľká Británia
5	Hans	Müller	Nemecko
6	Harry	Walker	Veľká Británia
7	Ján	Novák	Slovensko
8	Jean	Lefevre	Francúzsko
9	Karl	Schmidt	Nemecko
10	Louis	Bernard	Francúzsko
11	Marek	Kováč	Slovensko
12	Oleksiy	Petrenko	Ukrajina
13	Oliver	King	Veľká Británia
14	Peter	Horváth	Slovensko
15	Pierre	Dubois	Francúzsko

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



Napište SQL dopyt, ktorý vráti jeden stĺpec a bude pomenovaný ako pretek. Riadky v danom stĺpci budú predstavovať názvy pretekov, na ktorých sa zúčastnil bežec s menom "Marek" a priezviskom "Kováč". Výsledok zoradíte podľa názvu preteku vzostupne (A-Z).

Výsledný dopyt vráti 3 riadky.

Відповідь:

```
SELECT
    p.nazov AS pretek
FROM pretek p
JOIN bežec_pretek bp ON p.id_pretek = bp.id_pretek
JOIN bežec b ON bp.id_bežec = b.id_bežec
WHERE b.meno = 'Marek'
    AND b.priezvisko = 'Kováč'
ORDER BY p.nazov ASC; -- A->Z
```

Перевірка:

	pretek
1	Beh po Alpách
2	Polmaratón v Prahe
3	Taliansky polmaratón

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

Zvyšný čas 0:55:11

Napište SQL dopyt, ktorý vráti zoznam úplne všetkých krajín spolu s priezviskami bežcov, ktorí pochádzajú z danej krajiny. Priezviská nech sú zobrazené ako jeden reťazec, kde sú jednotlivé priezviská oddelené čiarkou bez medzery (napr. Novak,Kovac,Hruska).

Výstup má obsahovať **4 stĺpce v nasledujúcom poradí**:

1. Názov krajiny, pomenovaný ako krajina
2. Kód krajiny, pomenovaný ako skratka
3. Spojený reťazec priezvisk bežcov, pomenovaný ako bezci
4. Počet bežcov z danej krajiny, pomenovaný ako pocet

Zoradiť výsledok:

- podľa názvu krajiny vzostupne (A->Z),
- potom podľa kódu krajiny vzostupne (A->Z),
- následne podľa počtu bežcov zostupne (9->1),
- a napokon podľa priezvisk (spojeného reťazca) vzostupne (A->Z).

Výsledný dopyt vráti 15 riadkov.

Vідповідь:

SELECT

k.nazov AS krajina,
 k.kod AS skratka,
 STRING_AGG(b.priezvisko, ',' ORDER BY b.priezvisko) AS bezci,
 COUNT(*) AS pocet

FROM krajina k

JOIN bezec b ON b.id_krajina = k.id_krajina

GROUP BY

k.nazov,
 k.kod

ORDER BY

krajina ASC,
 skratka ASC,
 pocet DESC,
 bezci ASC;

Перевірка:

	krajina ▾	skratka ▾	bezci ▾	pocet ▾
1	Argentína	ARG	Diaz,Gomez,Rodriguez	3
2	Austrália	AUS	Anderson,Taylor,Williams	3
3	Brazília	BRA	Costa,Santos,Silva	3
4	Česko	CZE	Dvořák,Novotný,Svoboda	3
5	Egypt	EGY	Ali,Hassan,Ibrahim	3
6	Francúzsko	FRA	Bernard,Dubois,Lefevre	3
7	India	IND	Kumar,Patel,Sharma	3
8	Japonsko	JPN	Nakamura,Saito,Tanaka	3
9	Južná Afrika	ZAF	Dlamini,Mthembu,Nkosi	3
10	Kanada	CAN	Hernandez,Lopez,Martinez	3
11	Nemecko	DEU	Müller,Schmidt,Weber	3
12	Slovensko	SVK	Horváth,Kováč,Novák	3
13	Spojené štáty	USA	Brown,Johnson,Smith	3
14	Ukrajina	UKR	Ivanov,Petrenko,Shevchenko	3
15	Veľká Británia	GBR	Harris,King,Walker	3

Zvyšný čas 0:55:03

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

Napíšte SQL dopyt, ktorý vráti 3 stĺpce v nasledujúcom poradí:

- meno bežca a pomenované ako meno,
- priezvisko bežca a pomenované ako priezvisko,
- názov preteku a pomenované ako pretek.

Výsledok zoradíte vzostupne (1-9, A-Z) podľa mena, priezviska a preteku. Vo výsledku zarátajte iba preteky, ktoré sa budú konať v máji 2025.

Nepoužívajte diakritiku pri pomenovaní stĺpcov.

Výsledný dopyt vráti 33 riadkov.

Відповідь:

```
SELECT
    b.meno AS meno,
    b.priezvisko AS priezvisko,
    p.nazov AS pretek
FROM bezec b
JOIN bezec_pretek bp ON bp.id_bezec = b.id_bezec
JOIN pretek p ON p.id_pretek = bp.id_pretek
WHERE p.datum BETWEEN '2025-05-01' AND '2025-05-31'
ORDER BY
    b.meno ASC,
    b.priezvisko ASC,
    p.nazov ASC;
```

Перевірка:

	meno	priezvisko	pretek
23	Marek	Kováč	Taliansky polmaratón
24	Oleksiy	Petrenko	Polmaratón v Prahe
25	Peter	Horváth	Polmaratón v Prahe
26	Pierre	Dubois	Polmaratón v Prahe
27	Pierre	Dubois	Taliansky polmaratón
28	Rafael	Costa	Taliansky polmaratón
29	Rajesh	Kumar	Polmaratón v Prahe
30	Robert	Johnson	Taliansky polmaratón
31	Sipho	Nkosi	Taliansky polmaratón
32	Takumi	Nakamura	Taliansky polmaratón
33	Vikram	Sharma	Polmaratón v Prahe

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

Zvyšný čas 0:54:51

```

    graph LR
      krajina -- id_krajina --> bezec[id_krajina]
      bezec -- id_bezec --> bezec_pretek[id_bezec]
      bezec_pretek -- id_pretek --> pretek[id_pretek]
  
```

Napište SQL dopyt, ktorý vráti počet pretekov, ktoré jednotliví bežci s vyplneným stredným menom úspešne dokončili (tzn. cas_v_milisekundach **nie je NULL**), a zároveň pochádzajú z krajín, ktorých kód obsahuje písmeno „R“.

Vo výstupe nech sa nachádza aj čiastkový súčet dokončených pretekov týchto bežcov pre jednotlivé krajiny a celkový súčet dokončených pretekov týchto bežcov.

Výstup má obsahovať **3 stĺpce v nasledujúcom poradí**:

1. Kód krajiny, pomenovaný ako krajina
2. Stredné meno bežca, pomenované ako bezec
3. Počet dokončených pretekov, pomenovaný ako dokoncil

Zoradíte výsledok:

- podľa kódu krajiny vzostupne (A->Z),
- potom podľa stredného mena bežca vzostupne (A->Z),
- a nakoniec podľa počtu dokončených pretekov zostupne (9->1).

Výsledný dopyt vráti 9 riadkov.

Vідповідь:

```

SELECT
    k.kod AS krajina,
    b.stredne_meno AS bezec,
    COUNT(*) AS dokoncil
FROM krajina k
JOIN bezec b ON b.id_krajina = k.id_krajina
JOIN bezec_pretek bp ON bp.id_bezec = b.id_bezec
WHERE b.stredne_meno IS NOT NULL
    AND bp.cas_v_milisekundach IS NOT NULL
    AND k.kod LIKE '%R%'
GROUP BY GROUPING SETS ((k.kod, b.stredne_meno), (k.kod), ())
ORDER BY
    krajina ASC NULLS LAST,
    bezec ASC NULLS LAST,
    dokoncil ASC;
  
```

Перевірка:

	krajina ▾	bezec ▾	dokoncil ▾
1	ARG	Andres	3
2	ARG	<null>	3
3	BRA	Henrique	3
4	BRA	<null>	3
5	FRA	Claude	2
6	FRA	<null>	2
7	UKR	Serhiy	3
8	UKR	<null>	3
9	<null>	<null>	11

Otázka 12

Éšte nezodpovedané

Max. hodnotenie 7,00

Označiť otázku

Zvyšný čas 0:54:43

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



Napište SQL dopyt, ktorý vráti 2 stĺpce v nasledujúcom poradí:

- **kontinent** – názov kontinentu, z ktorého pochádzajú bežci pomenovaný ako kontinent,
- **pocet_bežcov** – počet bežcov z daného kontinentu, ktorí zabehli čas kratší ako 140 000 milisekúnd, nepochádzajú z krajiny **India** a zároveň sa narodili po dátume 21. 07. 1985 (vrátane), pomenovaný ako pocet_bežcov.

Do výsledku zahrňte iba tie kontinenty, kde je viac ako 6 takýchto bežcov.

Výsledok zoradte nasledovne:

- najprv podľa pocet_bežcov **zostupne** (9 → 1),
- potom podľa kontinent **vzostupne** (A → Z).

Nepoužívajte diakritiku pri pomenovaní stĺpcov.

Výsledný dopyt vráti 2 riadky.

Відповідь:

```
SELECT
    k.kontinent AS kontinent,
    COUNT(b.id_bežec) AS pocet_bežcov
FROM krajina k
JOIN bežec b ON b.id_krajina = k.id_krajina
JOIN bežec_pretek bp ON bp.id_bežec = b.id_bežec
WHERE bp.cas_v_milisekundach < 140000
    AND k.nazov <> 'India'
    AND b.datum_narodenia >= '1985-07-21'
GROUP BY
    k.kontinent
HAVING
    COUNT(b.id_bežec) > 6
ORDER BY
    pocet_bežcov DESC,
    kontinent ASC;
```

Перевірка:

	kontinent	pocet_bežcov
1	Európa	13
2	Južná Amerika	7

Time left 0:16:45

Question 9
 Not yet answered
 Marked out of 5.00
 Flag question

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

```

    graph LR
      krajina --> bežec
      krajina --> bežec_pretek
      bežec --> bežec_pretek
      bežec_pretek --> pretek
  
```

Napište SQL dopyt, ktorý vráti **3 stĺpce** v nasledujúcom poradí:

- meno a priezvisko bežca zretážené do jedného reťazca s medzerou medzi, pomenované ako meno,
- čas v sekundách (prepočítaný z milisekúnd), pomenovaný ako cas,
- názov preteku, ktorého sa bežec zúčastnil, pomenovaný ako pretek.

Do výsledku zahrňte **len tých bežcov**, ktorí majú **nejaký čas**, **nemajú uvedené stredné meno**, pochádzajú z krajiny, ktorej **rozloha nepresahuje 400 000 km²**.

Výsledok **zoraďte vzostupne (1 → 9, A → Z)** podľa stĺpcov v tomto poradí: pretek, cas, meno.

Nepoužívajte diakritiku pri pomenovaní stĺpcov.

Výsledný dopyt vráti 20 riadkov.

Відповідь:

```

SELECT
    b.meno || ' ' || b.priezvisko AS meno,
    bp.cas_v_milisekundach / 1000.0 AS cas,
    p.nazov AS pretek
FROM bežec b
JOIN bežec_pretek bp ON bp.id_bežec = b.id_bežec
JOIN pretek p ON bp.id_pretek = p.id_pretek
JOIN krajina k ON b.id_krajina = k.id_krajina
WHERE bp.cas_v_milisekundach IS NOT NULL
      AND b.stredne_meno IS NULL
      AND k.rozloha_v_km2 <= 400000
ORDER BY
    pretek ASC,
    cas ASC,
    meno ASC;
  
```

Перевірка:

	meno	cas	pretek
1	Jakub Dvořák	125	Beh po Alpách
2	Hiroshi Tanaka	145	Beh po Alpách
3	Fritz Weber	150	Beh po Alpách
4	Marek Kováč	152.5	Beh po Alpách
5	Karl Schmidt	140	Maratón v Amsterdame
6	Ján Novák	121	Maratón v Berlíne
7	Jakub Dvořák	135	Maratón v Berlíne
8	Ján Novák	122.5	Nočný beh v Barcelone
9	Takumi Nakamura	142	Nočný beh v Barcelone
10	Hans Müller	150	Nočný beh v Barcelone
11	Fritz Weber	155	Polmaratón v Bruseli
12	Ján Novák	125	Polmaratón v Prahe
13	Hiroshi Tanaka	135	Polmaratón v Prahe
14	Marek Kováč	145	Polmaratón v Prahe
15	Hiroshi Tanaka	140	Taliansky polmaratón
16	Hans Müller	145	Taliansky polmaratón
17	Takumi Nakamura	148	Taliansky polmaratón
18	Marek Kováč	150	Taliansky polmaratón
19	Karl Schmidt	145	Viedenský nočný beh
20	Takumi Nakamura	145.5	Viedenský nočný beh

Time left 0:30:33

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



Napište SQL dopyt, ktorý vráti meno, priezvisko a štát všetkých bežcov, ktorí NIE sú z Európy. Výsledok zoradíte vzostupne podľa štátu, potom priezviska a nakoniec mena. Stĺpce premenujte na meno, priezvisko a stat (bez diakritiky).

Výsledný dopyt vráti 27 riadkov.

Відповідь:

```
SELECT
    b.meno AS meno,
    b.priezvisko AS priezvisko,
    k.nazov AS stat
FROM bezec b
JOIN krajina k ON b.id_krajina = k.id_krajina
WHERE k.kontinent <> 'Európa'
ORDER BY
    stat ASC,
    priezvisko ASC,
    meno ASC;
```

Перевірка:

	meno	priezvisko	stat
7	Rafael	Costa	Brazília
8	Carlos	Santos	Brazília
9	Eduardo	Silva	Brazília
10	Mohamed	Ali	Egypt
11	Ahmed	Hassan	Egypt
12	Omar	Ibrahim	Egypt
13	Rajesh	Kumar	India
14	Amit	Patel	India
15	Vikram	Sharma	India
16	Takumi	Nakamura	Japonsko
17	Kenji	Saito	Japonsko
18	Hiroshi	Tanaka	Japonsko
19	Lerato	Dlamini	Južná Afrika
20	Thabo	Mthembu	Južná Afrika
21	Sipho	Nkosi	Južná Afrika
22	Lucas	Hernandez	Kanada
23	Gabriel	Lopez	Kanada
24	Daniel	Martínez	Kanada
25	William	Brown	Spojené štáty
26	Robert	Johnson	Spojené štáty
27	John	Smith	Spojené štáty

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

Time left 0:44:37

Napište SQL dopyt, ktorý vráti názov, dĺžku v metroch a čas v milisekundách pre všetky preteky, ktoré majú názov Taliansky polmaratón a majú uvedený nejaký čas. Výsledok zoradíte vzostupne (A → Z) podľa názvu, potom dĺžky a nakoniec času. Stĺpce premenujte na nazov, dlzka, cas (bez diakritiky). Výsledný dopyt vráti 14 riadkov.

Відповідь:

```
SELECT
    p.nazov AS nazov,
    p.dlзка_v_metroch AS dlzka,
    bp.cas_v_milisekundach AS cas
FROM pretek p
JOIN bezec_pretek bp ON bp.id_pretek = p.id_pretek
WHERE p.nazov = 'Taliansky polmaratón'
    AND bp.cas_v_milisekundach IS NOT NULL
ORDER BY
    nazov ASC,
    dlzka ASC,
    cas ASC;
```

P.S:

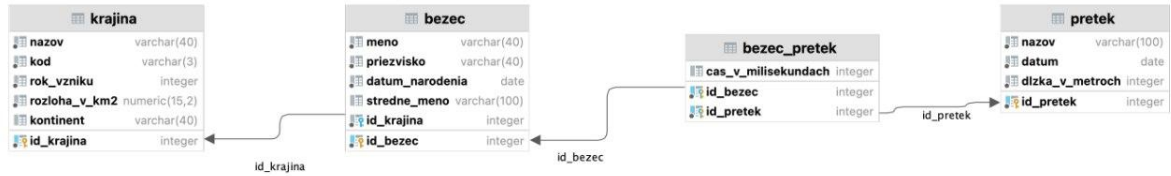
Taliansky polmaratón не забувати за наголос

Перевірка:

	<input type="checkbox"/> nazov ▾	<input type="checkbox"/> dlzka ▾	<input type="checkbox"/> cas ▾
1	Taliansky polmaratón	21097	125000
2	Taliansky polmaratón	21097	128000
3	Taliansky polmaratón	21097	130000
4	Taliansky polmaratón	21097	130000
5	Taliansky polmaratón	21097	130500
6	Taliansky polmaratón	21097	140000
7	Taliansky polmaratón	21097	140000
8	Taliansky polmaratón	21097	140000
9	Taliansky polmaratón	21097	140000
10	Taliansky polmaratón	21097	145000
11	Taliansky polmaratón	21097	145000
12	Taliansky polmaratón	21097	145000
13	Taliansky polmaratón	21097	148000
14	Taliansky polmaratón	21097	150000

14 rows ▾

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



Napište SQL dopyt, ktorý vráti krajinu, zaokrúhlený priemerný čas v sekundách (nahor) a počet bežcov z tejto krajiny, ktorí sa zúčastnili na ľubovoľnom preteku.

Výstupné stĺpce: stat, priemer, pocet (v tomto poradí, bez diakritiky)

Zoradiť výsledok vzostupne podľa priemer (1→9), potom pocet (1→9)

Očakávaný počet riadkov: 14

Відповідь:

SELECT

k.nazov AS stat,
ROUND(AVG(bp.cas_v_milisekundach) / 1000.0, 0) AS priemer,
COUNT(DISTINCT b.id_bezec) AS pocet

FROM Krajina k

JOIN bezec b ON b.id_krajina = k.id_krajina

JOIN bezec_pretek bp ON bp.id_bezec = b.id_bezec

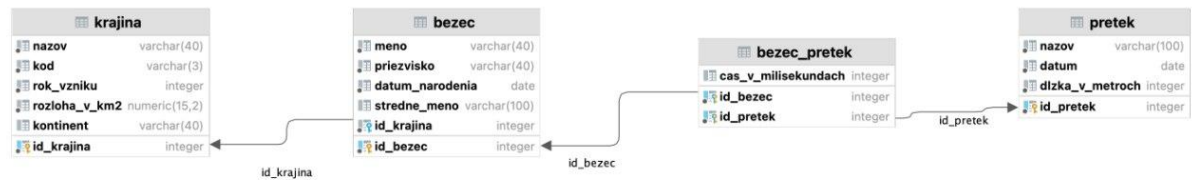
GROUP BY k.nazov

ORDER BY priemer ASC,
pocet ASC;

Перевірка:

	stat	priemer	pocet
1	Česko	130	1
2	India	134	3
3	Slovensko	136	3
4	Spojené štáty	136	3
5	Brazília	137	3
6	Kanada	138	3
7	Francúzsko	138	3
8	Austrália	140	3
9	Ukrajina	141	3
10	Egypt	141	3
11	Južná Afrika	141	3
12	Japonsko	143	3
13	Argentína	145	3
14	Nemecko	148	3

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



Napište SQL dopyt, ktorý vráti zoznam pretekov, ktorých dĺžka je na intervale 40000 až 50000 metrov (vrátane). Dopyt nech vráti 3 stĺpce v nasledujúcom poradí:

- Názov preteku pomenovaný ako „pretek“
- Rok kedy sa bežal pretek pomenovaný ako „rok“
- Dĺžka preteku v metroch pomenovaný ako „dlzka“

Záznamy zoradíte podľa názvu preteku vzostupne (A->Z), roku vzostupne (1->9) a dĺžky zostupne (9->1).

Výsledný dopyt vráti 3 riadky.

Відповідь:

SELECT

p.nazov AS pretek,
EXTRACT(YEAR FROM p.datum) AS rok,
p.dlzka_v_metroch AS dlzka

FROM pretek p

WHERE p.dlzka_v_metroch BETWEEN 40000 AND 50000

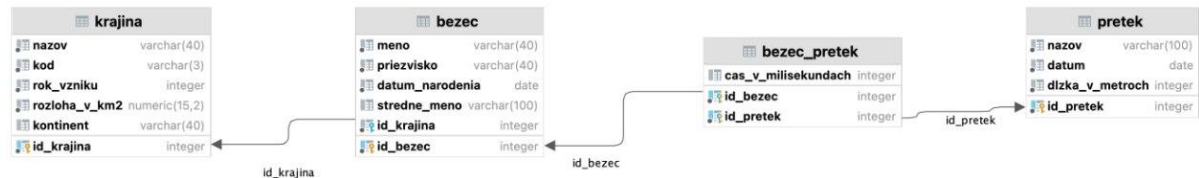
ORDER BY pretek ASC,

rok ASC,
dlzka DESC;

Перевірка:

	pretek	rok	dlzka
1	Londýnsky maratón	2025	42195
2	Maratón v Amsterdame	2025	42195
3	Maratón v Berlíne	2025	42195

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



Napište SQL dopyt, ktorý vráti 2 stĺpce v nasledujúcom poradí:

- meno a priezvisko bežca zretazené a pomenované ako meno,
- čas v sekundach, pomenovaný ako cas, ktorý zabehli bežci na preteku „Maratón v Berlíne“.

Výsledok zoradíte vzostupne (1-9, A-Z) podľa času, priezviska, mena.

Nepoužívajte diakritiku pri pomenovaní stĺpcov.

Výsledný dopyt vráti 18 riadkov.

Відповідь:

SELECT

b.meno || ' ' || b.priezvisko AS meno,
bp.cas_v_milisekundach / 1000.0 AS cas

FROM pretek p

JOIN bezec_pretek bp ON bp.id_pretek = p.id_pretek

JOIN bezec b ON b.id_bezec = bp.id_bezec

WHERE p.nazov = 'Maratón v Berlíne'

ORDER BY cas ASC NULLS LAST,

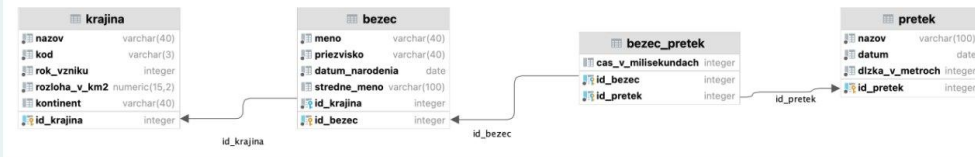
b.priezvisko ASC,

b.meno ASC;

Перевірка:

	meno	cas
1	Ján Novák	121
2	Rajesh Kumar	125
3	John Smith	125
4	Vikram Sharma	126
5	Carlos Santos	130
6	Andriy Shevchenko	130
7	Pierre Dubois	130.5
8	Jakub Dvořák	135
9	Lucas Hernandez	135
10	Peter Horváth	135
11	James Williams	135
12	Omar Ibrahim	140
13	Robert Johnson	140
14	Alejandro Gomez	150
15	Dmytro Ivanov	150
16	Mohamed Ali	<null>
17	Hans Müller	<null>
18	Kenji Saito	<null>

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



Napište SQL dopyt, ktorý vráti 3 stĺpce v tomto poradí:

1. meno bežca (pomenovaný ako *meno*),
2. priezvisko bežca (pomenované ako *priezvisko*),
3. trvanie behu v sekundách (pomenovaný ako *sekundy*).

Dopyt má vrátiť **bežcov**, ktorí sa umiestnili v prvej polovici výsledkovej listiny pre pretek s názvom „Polmaratón v Bruseli“.

♦ Ak bežec **nemá uvedený čas**, znamená to, že **nedobehol** — aj takito bežci sú vo výsledkovej listine a umiestňujú sa na koniec poradia.

♦ **Zoradenie výsledku:**

- najskôr podľa trvania behu (*sekundy*) vzostupne (najrýchlejší prví),
- potom podľa priezviska (A→Z),
- nakoniec podľa mena (A→Z).

💡 Výsledný dopyt vráti presne **5 riadkov**.

Відповідь:

WITH zoradeni AS (

SELECT

```
    b.meno,
    b.priezvisko,
    bp.cas_v_milisekundach / 1000.0 AS sekundy,
    NTILE(2) OVER (
        ORDER BY
            bp.cas_v_milisekundach IS NULL
            bp.cas_v_milisekundach,
            b.priezvisko,
            b.meno
```

) AS kos

FROM pretek p

JOIN bezec_pretek bp ON bp.id_pretek = p.id_pretek

JOIN bezec b ON b.id_bezec = bp.id_bezec

WHERE p.nazov = 'Polmaratón v Bruseli'

)

SELECT

```
    meno,
    priezvisko,
    sekundy
```

FROM zoradeni

WHERE kos = 1

ORDER BY

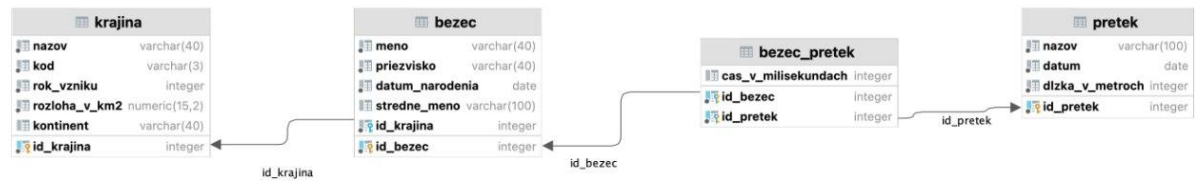
```
    sekundy ASC NULLS LAST,
    priezvisko ASC,
    meno ASC;
```

P.S писав ГПТ так що перевіряйте

Перевірка:

	meno ▾	priezvisko ▾	sekundy ▾
1	Rafael	Costa	132
2	Lerato	Dlamini	140
3	Felipe	Rodriguez	140
4	Louis	Bernard	145
5	Dmytro	Ivanov	148

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



Napište SQL dopyt, ktorý vráti 4 stĺpce v nasledujúcom poradí:

- meno pretekára pomenované ako meno,
- priezvisko pretekára pomenované ako priezvisko,
- názov preteku pomenovaný ako pretek,
- čas v milisekundách, ktorý bežec dosiahol v preteku. Stĺpec nech je pomenovaný ako milisekundy. (Taktiež zobrazte aj pretekárov, ktorí nemajú uvedený čas a tam bude uvedená hodnota <null>.)

Vo výsledku zobrazte len pretek s názvom "Beh po Alpách". Výsledný dopyt zoradte podľa mena, priezviska, názvu preteku vzostupne (A-Z) a podľa času, ktorý bežec dosiahol zostupne (9-1).

Výsledný dopyt vráti 13 riadkov.

Відповідь:

```
SELECT
    b.meno AS meno,
    b.priezvisko AS priezvisko,
    p.nazov AS pretek,
    bp.cas_v_milisekundach AS milisekundy
FROM pretek p
JOIN bezec_pretek bp ON bp.id_pretek = p.id_pretek
JOIN bezec b ON b.id_bezec = bp.id_bezec
WHERE p.nazov = 'Beh po Alpách'
ORDER BY
    meno ASC,
    priezvisko ASC,
    pretek ASC,
    milisekundy DESC NULLS LAST;
```

Перевірка:

	meno	priezvisko	pretek	milisekundy
1	Daniel	Martinez	Beh po Alpách	150000
2	Eduardo	Silva	Beh po Alpách	145500
3	Fritz	Weber	Beh po Alpách	150000
4	Hiroshi	Tanaka	Beh po Alpách	145000
5	Jakub	Dvořák	Beh po Alpách	125000
6	James	Williams	Beh po Alpách	130000
7	John	Smith	Beh po Alpách	132000
8	Juan	Diaz	Beh po Alpách	<null>
9	Lerato	Dlamini	Beh po Alpách	135000
10	Marek	Kováč	Beh po Alpách	152500
11	Noah	Anderson	Beh po Alpách	155000
12	Omar	Ibrahim	Beh po Alpách	145500
13	Rajesh	Kumar	Beh po Alpách	137000