

A) Který z následující regulárních výrazů validuje text obsahující Jméno Příjmení (bez diakritiky, první písmena slova velká, ostatní malá)?

1. `\W\w* \W\w*`
2. `(?i)\w+ \w+`
3. `[A-Z][a-z]+ [A-Z][a-z]+`
4. `[A-Za-z]* [A-Za-z]*`

B) Jakým řetězcům odpovídá tento regulární výraz: `^\d{^d}+\d`

1. Celý řádek, který začíná zpětným lomítkem, následovaný právě jedním číslem, následovaný čímkoliv kromě čísla s libovolným počtem opakování a končí právě jedním číslem.
2. Celý řádek, který začíná zpětným lomítkem a písmenem d. Následuje sekvence znaků zpětné lomítko a písmeno d, která se musí alespoň jednou opakovat. Řádek končí právě jedním číslem.
3. Na začátku řádku musí být zpětné lomítko a písmeno d. Pak se libovolný znak kromě čísla opakuje alespoň jednou a následuje právě jedno číslo.
4. Na začátku řádku musí být zpětné lomítko a alespoň dvě čísla. Pak následuje znak plus a opět jedno číslo.

C) Definice tabulky TB s jediným celočíselným sloupcem X, kde nejnižší hodnoty očekáváme cca minus 1000 a nejvyšší okolo 5000, bude vypadat následovně:

1. `CREATE TABLE TB (X PRIMARY KEY auto_increment);`
2. `CREATE TABLE TB (X TINYINT NOT NULL DEFAULT 0);`
3. `CREATE TABLE TB (X SMALLINT);`
4. `CREATE TABLE TB (X MEDIUMINT UNSIGNED);`

D) Chceme-li zjistit, jakým příkazem bychom vytvořili existující tabulku TB, zapíšeme:

1. `DESCRIBE TB;`
2. `OPTIMIZE TABLE TB;`
3. `CREATE TABLE LIKE TB;`
4. `SHOW CREATE TABLE TB;`

E) Pokud vkládáme nový řádek do tabulky obsahující unikátní klíč a nechceme, aby případný pokus o vložení duplicitní hodnoty skončil chybou, pak použijeme:

1. `CONSTRAINT FOREIGN KEY` s klauzulí `RESTRICT`
2. `TRIGGER` definovaný tak, aby se spustil ještě před zápisem nových dat do tabulky
3. Klauzuli `DISTINCT`
4. Variantu příkazu `INSERT` s klíčovým slovem `IGNORE`

F) Mějme tabulku TB s celočíselným sloupcem X. Co vypíše následující příkaz?

`SELECT A.X AS AX, B.X AS BX FROM TB A, TB B WHERE AX > BX;`

1. Jeden řádek obsahující sloupce s testy „AX“ a „BX“
2. Chybu
3. Všechny možné kombinace dvou čísel ze sloupce X, pro které platí, že první vybrané číslo je větší jak druhé vybrané číslo.
4. Prázdný seznam, protože dvakrát vybereme jedno a to samé číslo, a pro takovou dvojici stejných čísel nikdy nemůže platit nerovnost.

G) Který z následujících dotazů jistojistě vrátí právě jeden řádek? Tabulka TB existuje.

1. SELECT COUNT(*) FROM TB;
2. SELECT * FROM TB WHERE ID=1;
3. SELECT * FROM TB LIMIT 1;
4. SELECT DISTINCT * FROM TB;

H) Pokud chceme očíslovat řádky výstupu příkazu SELECT, použijeme

1. Kombinaci číselného primárního klíče a klauzule ORDER BY
2. Okenní funkci ROW_NUMBER
3. Klauzuli LIMIT s upřesněním offsetu
4. Agregáční funkci COUNT

I) Mějme tabulku TB s celočíselnými sloupci A, B a C. Co vypíše následující příkaz?

SELECT A, B, max(C) FROM TB GROUP BY B, A;

1. Pro každou jedinečnou hodnotu ve druhém sloupci všechny jedinečné hodnoty v prvním sloupci a k nim ve třetím sloupci největší číslo ze sloupce C pro danou unikátní dvojici ze sloupců A a B.
2. Pro každou dvojici čísel z A a B jejich součet.
3. Všechny jedinečné hodnoty ze sloupce A a všechny jedinečné hodnoty ze sloupce B s tím, že stejné hodnoty ze sloupce C budou vypsány za sebou.
4. Seskupené hodnoty, nejdříve ze sloupce B, pak ze sloupce A, a k nim největší hodnotu ze všech čísel ve sloupci C.

J) Pokud chceme spojit tabulky T1 a T2 tak, že ve výpisu budou všechny hodnoty z T1 bez ohledu na to, zda v T2 existuje řádek vyhovující spojovací podmínce, pak použijeme

1. SELECT T1.*, T2.* FROM T1 INNER JOIN T2 ON T1.id=T2.id;
2. SELECT T1.*, T2.* FROM T1 LEFT JOIN T2 ON T1.id=T2.id;
3. SELECT T1.*, Tx.* FROM T1, (SELECT * FROM T2) Tx WHERE T1.id = Tx.id;
4. SELECT T1.*, T2.* FROM T1 JOIN T2;

K) Co vypíše následující příkaz pro tabulku TB s tímto obsahem?

A	B
1	3
2	4

SELECT IF(EXISTS (SELECT A FROM TB WHERE A > B), (SELECT MAX(B) FROM TB), (SELECT A+B FROM TB));

1. Prázdnou množinu
2. Číslo 4
3. Čísla 4 a 6
4. Chybu