

CLOUD BI

2. forduló



A kategória támogatója: DXC Technology

Ismertető a feladathoz

Útmutató:

- A **radio button-os kérdésekre** egy helyes válasz van.
- **Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Badge-eket** a 4.forduló után kapsz majd először.
- Az **adatbekérős kérdéseknél** igyekeztünk minden variációt megadni (kisbetű, nagybetű, szóköz), de ha mégis eltérést tapasztalsz a megoldásokban, kérjük, jelezd felénk!

+1: Azért szólunk, hogy senkit ne a végén érjen meglepetés: a játék nem tipp-mix és csapatkategória sincs! Természetesen akinek nem inge...

Jó versenyzést kívánunk!

2.forduló: Ebben a fordulóban SQL alapú kérdésekkel találkozhatsz.

NEM lesz szükség Google Cloud accountra, vagy azon történő munkára a feladatok megoldásához.

Egyes válaszlehetőségeknél "Option1", "Option2" stb. megjelöléssel találkozhatsz, ez szövegileg nem része az adott válasznak, csupán a válaszok későbbi összekapcsolódását biztosítja a magyarázatokkal.

Felhasznált idő: 00:00/27:00

Elért pontszám: 0/7

1. feladat 0/1 pont

Melyik SQL parancs szolgál adatleválogatásra?

Válasz

☐ INSERT

☐ UPDATE

☒ SELECT

Ez a válasz helyes, de nem jelölted meg.

☐ SHOW

Magyarázat

A helyes válasz a SELECT.

Az INSERT és az UPDATE adatbevitelre és módosításra használt.

A SHOW nem a standard SQL része.

2. feladat 0/1 pont

Melyik SQL parancs nem DDL?

Válasz

- ☐ CREATE INDEX
- ☐ DROP TABLE
- ☐ ALTER VIEW

☒ INSERT INTO

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A helyes válasz az INSERT INTO, ugyanis az INSERT INTO egy DML, tehát adat manipulációt végez. A többi utasítás DDL, azaz adatdefinícióra szolgál.

3. feladat 0/1 pont

Adja meg a hiányzó szót az alábbi SQL parancs kipontozott részénél.



```
SELECT SALES_DATE, COUNT(*) CNT  
FROM SALES_TRANSACTIONS  
WHERE REGION='US-NORTH'  
... BY SALES_DATE;
```

Válasz

A helyes válasz:

GROUP

Magyarázat

A helyes válasz "GROUP". Ezzel lesz teljes a GROUP BY klauzula, ami az eredmények csoportosítását szolgálja.

4. feladat 0/1 pont

A következő parancsra vonatkozólag mely állítások nem igazak (3)?

```
UPDATE A
SET B='C'
WHERE D='E'
;
```

Válaszok

- ☒ Option1: A egy mező, B egy tábla.
Ez a válasz helyes, de nem jelölted meg.
- ☐ Option2: D egy oszlop, E egy filterezésre használt érték.
- ☒ Option3: B táblát C-re nevezzük át ha A táblában D mező értéke E.
Ez a válasz helyes, de nem jelölted meg.
- ☐ Option4: B mező értékét C-re állítjuk az A táblában ha D mező értéke E.
- ☒ Option5: Az utasítás mindig csak egy sorban változtatja meg a mező értékét.
Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A jó válaszok, amelyek hamisak: az Option1, Option3 és Option5.

"A" nem mező, ellenben "B" az. Mivel "B" nem tábla, nem az átnevezéséről szól az utasítás, hanem értékadás történik.

Nem ismerjük az "A" tábla szerkezetét, emiatt nem feltételezhetjük, hogy "D" mező szerint egyedi index vagy kulcs van rajta.

5. feladat 0/1 pont

Melyik SQL utasítás válaszol a következő feladatra: "Kik voltak a 2021. év legsikeresebb cipő eladói a tízezer forintnál drágább cipők árát figyelembe véve? Listázzuk ki mindet, elől a legsikeresebbekkel. Egy eladó több boltban is dolgozhatott az év során."

Válasz

☐ Option1:

```
SELECT ERTEKESITO,  
       SUM(ELADAS_DB) AS TOT_ELADAS_DB,  
       AVG(ELADAS_AR) AS AVG_ELADAS_AR  
FROM ERTEKESITESEK  
WHERE ERTEKESITES_EVE = 2021  
      AND CIKK_TIPUS='CIPŐ'  
GROUP BY ERTEKESITO  
HAVING AVG_ELADAS_AR>10000  
ORDER BY TOT_ELADAS_DB DESC  
;
```

☐ Option2:

```
SELECT ERTEKESITO,  
       SUM(ELADAS_DB) AS TOT_ELADAS_DB,  
       AVG(ELADAS_AR) AS AVG_ELADAS_AR  
FROM ERTEKESITESEK  
WHERE ERTEKESITES_EVE = 2021  
      AND CIKK_TIPUS='CIPŐ'  
      AND ELADAS_AR>10000  
GROUP BY ERTEKESITO  
ORDER BY TOT_ELADAS_AR DESC  
;
```

☐ Option3:

```
SELECT ERTEKESITO, BOLT,  
       AVG(ELADAS_DB) AS AVG_ELADAS_DB,  
       SUM(ELADAS_AR) AS TOT_ELADAS_AR  
FROM ERTEKESITESEK  
WHERE ERTEKESITES_EVE = 2021  
      AND CIKK_TIPUS='CIPŐ'  
      AND ELADAS_AR>10000  
GROUP BY ERTEKESITO, BOLT  
ORDER BY TOT_ELADAS_AR DESC  
;
```

Option4:

```
SELECT ERTEKESITO,  
       AVG(ELADAS_DB) AS AVG_ELADAS_DB,  
       SUM(ELADAS_AR) AS TOT_ELADAS_AR  
FROM ERTEKESITESEK  
WHERE ERTEKESITES_EVE = 2021  
      AND CIKK_TIPUS='CIPŐ'  
      AND ELADAS_AR>10000  
GROUP BY ERTEKESITO  
ORDER BY TOT_ELADAS_AR DESC  
;
```

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A helyes válasz az Option4: az értékesítés tábla szűrve van a megadott feltételek szerint, az értékesítőkre csoportosított teljes eladási ár szerint csökkenő sorrendben kapjuk meg az eredményeket.

```
SELECT ERTEKESITO,  
        AVG(ELADAS_DB) AS AVG_ELADAS_DB,  
        SUM(ELADAS_AR) AS TOT_ELADAS_AR  
FROM ERTEKESITESEK  
WHERE ERTEKESITES_EVE = 2021  
      AND CIKK_TIPUS='CIPŐ'  
      AND ELADAS_AR>10000  
GROUP BY ERTEKESITO  
ORDER BY TOT_ELADAS_AR DESC  
;
```

Minden más esetben van logikai vagy szintaktikai hiba: az Option1 lekérdezés eladás darabszám szerint rendez és átlagárakkal számol.

Az Option2 lekérdezés felösszegzett eladási ár szerint rendez, de nincs ilyen mérőszám, csak az átlag.


Az Option3 lekérdezés szintaktikailag helyes, az árakat összegzi, azok szerint rendez visszafelé, viszont boltra is csoportosít, tehát nem a feltett kérdést válaszolja meg, hanem az adott boltokban végzett eladói értékesítéseket rangsorolja.

6. feladat 0/1 pont

Melyik lekérdezés válaszolja meg a következő kérdést: "Kik azok az ügyfelek 2021-ben, akiknek nemadtunk el cipőt, de eladtunk táskát?"

Válasz

☐ Option1:




```
SELECT
    UGYFEL_AZON
FROM
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='TASKA'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS T
INNER JOIN
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='CIPO'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS C
ON T.UGYFEL_AZON = C.UGYFEL_AZON
WHERE C.ELADAS_AR = 0
;
```

☐ Option2:

```
SELECT
    UGYFEL_AZON
FROM
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='TASKA'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS T
LEFT OUTER JOIN
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='CIPO'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS C
ON T.UGYFEL_AZON=C.UGYFEL_AZON
WHERE C.ELADAS_AR = 0
;
```

☐ Option3:



```
SELECT
    UGYFEL_AZON
FROM
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='TASKA'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS T
LEFT OUTER JOIN
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='CIPO'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS C
ON T.UGYFEL_AZON = C.UGYFEL_AZON
WHERE C.ELADAS_AR IS NULL
;
```

Ez a válasz helyes, de nem jelölted meg.

☐ Option4:

```
SELECT
    UGYFEL_AZON
FROM
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='TASKA'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS T
FULL OUTER JOIN
    (SELECT UGYFEL_AZON,
        SUM(ELADAS_AR) AS ELADAS_AR
    FROM ERTEKESITESEK
    WHERE CIKK_TIPUS='CIPO'
        AND ERTEKESITES_EVE = 2021
    GROUP BY UGYFEL_AZON) AS C
ON T.UGYFEL_AZON = C.UGYFEL_AZON
WHERE COALESCE(T.ELADAS_AR,0)>=0
;
```

Magyarázat

A helyes válasz az Option3: a táska eladások ügyfélre csoportosított halmazához kívülről kapcsolva a cipő eladások ügyfélre csoportosított halmazát, azon rekordok kellenek, ahol nincs cipő-halmazbeli rekord. Ezt az IS NULL fejezi ki.

```

SELECT
    UGYFEL_AZON
FROM
    ( SELECT UGYFEL_AZON,
            SUM(ELADAS_AR) AS ELADAS_AR
      FROM ERTEKESITESEK
      WHERE CIKK_TIPUS='TASKA'
            AND ERTEKESITES_EVE = 2021
      GROUP BY UGYFEL_AZON) AS T
LEFT OUTER JOIN
    ( SELECT UGYFEL_AZON,
            SUM(ELADAS_AR) AS ELADAS_AR
      FROM ERTEKESITESEK
      WHERE CIKK_TIPUS='CIPO'
            AND ERTEKESITES_EVE = 2021
      GROUP BY UGYFEL_AZON) AS C
ON T.UGYFEL_AZON = C.UGYFEL_AZON
WHERE C.ELADAS_AR IS NULL
;

```

Az Option1 lekérdezés INNER JOIN-t használ, tehát csak a táska és cipő értékesítései ügyfelek lesznek benne.

Az Option2 lekérdezés hiába használ külső kapcsolást (OUTER JOIN), mert utána a WHERE feltételben explicit hivatkozik a nulla értékre, ezzel INNER JOIN-ná konvertál.

Az Option4 lekérdezés a cipő és táska eladásokban érintett minden ügyfelet ki fog listázni, mert FULL OUTER JOIN-t használunk, a WHERE feltételnek minden táska eladás megfelel, a cipőkre nincs feltétel, tehát ott is minden rekordot megkapunk.

7. feladat 0/1 pont

A Google Cloud BigQuery-n lévő értékesítési táblánkban (SALES) kétmilliárd sor szerepel az elmúlt tíz évre és 159 oszlopa van.

A tábla értékesítési nap szerint partícionált (SALES_DATE) és ügyfélszám (CUSTOMER_ID) szerint clusterezett.

Az értékesítés időpontja külön oszlopban szerepel. Az ügyfélazonosítók érkezési sorrendben kerülnek kiosztásra.

A legfrissebb ügyfeleknek van a legnagyobb azonosítójuk. Körülbelül százezer ügyfél szerepel a táblában.

Melyik lekérdezés fut le a legkevesebb költséggel a táblán?

Válasz

☐ Option1:



```
SELECT * FROM SALES --LAST 200 SALES
WHERE CUSTOMER_ID BETWEEN (SELECT MAX(CUSTOMER_ID)-199 FROM SALES)
                        AND (SELECT MAX(CUSTOMER_ID) FROM SALES)
ORDER BY SALES_DATE;
```

☐ Option2:



```
SELECT SALES_DATE,
       SUM(SALES_PRICE) AS TOTAL_PRICE,
       AVG(SALES_PRICE) AS AVG_PRICE
FROM SALES
WHERE SALES_DATE BETWEEN DATE '2021-10-01'
                        AND DATE '2021-12-31'

GROUP BY SALES_DATE
ORDER BY SALES_DATE;
```

☒ Option3:



```
SELECT CUSTOMER_ID,
       MIN(SALES_DATE) AS FIRST_SALE_DATE,
       MAX(SALES_DATE) AS LAST_SALE_DATE,
       SUM(SALES_PRICE) AS TOTAL_PRICE,
       AVG(SALES_PRICE) AS AVG_PRICE
FROM SALES
WHERE SALES_DATE BETWEEN DATE '2021-01-01'
                        AND DATE '2021-12-31'
AND CUSTOMER_ID = 111111111
GROUP BY CUSTOMER_ID;
```

Ez a válasz helyes, de nem jelölted meg.

☐ Option4:

```
SELECT CUSTOMER_ID,  
       MIN(SALES_DATE) AS FIRST_SALE_DATE,  
       SUM(SALES_PRICE) AS TOTAL_PRICE,  
       AVG(SALES_PRICE) AS AVG_PRICE  
FROM SALES  
GROUP BY CUSTOMER_ID  
HAVING FIRST_SALE_DATE > DATE '2021-01-01';
```

Magyarázat

A helyes válasz az Option3. Ebben az esetben szűrünk a partíciós dátumra és egy konkrét customerre is, csak kiválasztott mezőket használunk, kihasználva a BigQuery columnar adattárolás előnyeit.

```
SELECT CUSTOMER_ID,  
       MIN(SALES_DATE) AS FIRST_SALE_DATE,  
       MAX(SALES_DATE) AS LAST_SALE_DATE,  
       SUM(SALES_PRICE) AS TOTAL_PRICE,  
       AVG(SALES_PRICE) AS AVG_PRICE  
FROM SALES  
WHERE SALES_DATE BETWEEN DATE '2021-01-01'  
      AND DATE '2021-12-31'  
AND CUSTOMER_ID = 111111111  
GROUP BY CUSTOMER_ID;
```

Az Option1 lekérdezés sem mezőre sem dátumra nem szűkít, több teljes tábla scant tartalmaz.

Az Option2 lekérdezés bár szűr dátumokra, de nem használ customer adatot.

Az Option4 lekérdezés hasonló, bár szerepel benne a customer, mint mező, de csak dátumra filterez.



[Legfontosabb tudnivalók](#) [Kapcsolat](#) [Versenyszabályzat](#) [Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos