

# CLOUD BI

1. forduló



A kategória támogatója: DXC Technology

## Ismertető a feladatlaphoz

Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:

MINDEN kérdésre van helyes válasz.

Olyan kérdés NINCS, amire az összes válasz helyes, ha mégis az összes választ bejelölöd, arra a feladatra automatikusan 0 pont jár.

Több válaszlehetőség esetén a helytelen válasz megjelölése mínusz pontot ér.

A radio button-os kérdésekre egy helyes válasz van.

Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN beküldi azt az addig megjelölt válaszokkal.

Azokat a feladatlapokat, amelyekhez csatolmány tartozik, javasoljuk NEM mobilon elindítani, erre az érintett feladatlapok előtt külön felhívjuk a figyelmet.

Az adatbekérős feladatokra NEM jár részpontoszám, csak a feleletválasztósakra.

Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.

Ha egyszerre több böngészőből, több ablakban vagy több eszközről megnyitod ugyanazt a feladatlapot, nem tudjuk vállalni az adatmentéssel kapcsolatban esetlegesen felmerülő anomáliákért a felelősséget!

A ChatGPT használata nem tiltott, de az arra való hivatkozással észrevételt NEM fogadunk el!

A feltűnően rövid idő alatt megoldott feladatlapok kizárását vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a forduló érvénytelenítésére!

Jó versenyzést kívánunk!



Ebben a fordulóban elsősorban SQL témájú feladatokkal találkozhatasz..

## 1. feladat 1 pont

Mi a célja az SQL adatbázisokban használt indexeknek?

### Válasz

- ☐ Az indexek segítenek az adatbázisok teljes méretének csökkentésében, hogy az adatok könnyebben kezelhetőek legyenek.
- ☐ Az indexek kódolják az adatokat, hogy azok ne legyenek láthatóak a felhasználók számára.
- ☐ Az indexek gyorsítják az adatbázis lekérdezéseit és csökkentik a keresési időt.
- ☐ Az indexek csak dekoratív elemek az adatbázisban, amelyek növelik a lekérdezések bonyolultságát.

## 2. feladat 1 pont

Adott az alábbi **STORE** nevű tábla. Mi lesz az eredménye a tábla alatt található lekérdezésnek?

BUSINESS_ID	ACCOUNTNUMBER	NAME
1,212	ARAS001	Aranyásó Ékszerbolt
1,381	BOKC921	Bodri Kutya Csaolója
1,594	FOFP128	Fogas Fogás Pecabolt
1,811	MOSZ022	Magyarországi szuvenír üzlet
2,029	MAME329	Magyar Mennydörgő Társasbolt
2,340	SZMA122	Százszorszép Magyar Virágbolt
2,767	TATT912	Tarka Tollas Turkáló

```
SELECT count(*) FROM dbo.STORE
WHERE (NAME LIKE 'Magyar' AND BUSINESS_ID BETWEEN 1230 AND 2190)
OR ACCOUNTNUMBER = 'TATT912'
```

Válasz

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4

### 3. feladat 1 pont

Adott az alábbi két tábla. Melyik SQL lekérdezés listázza ki azoknak a diákoknak a nevét és korát, akik 16 évesek, vagy idősebbek, és bármelyik évfolyam "A" betűs osztályába járnak?


TANULOK	
tanulo_id	Int (Primary key)
nev	String
kor	Int
osztaly_id	Int (Foreign key)

OSZTALYOK	
osztaly_id	Int (Primary key)
osztaly_nev	String
tanar_nev	String


Válasz

☐


```
SELECT nev, kor FROM TANULOK
WHERE kor >= 16 AND osztaly_id IN (
  SELECT osztaly_id FROM OSZTALYOK
  WHERE osztaly_nev LIKE '%A');
```

☐   

```
SELECT T.nev, T.kor  
FROM TANULOK T  
LEFT JOIN OSZTALYOK O ON T.osztaly_id = O.osztaly_id  
AND osztaly_nev LIKE '%A'  
WHERE T.kor >= 16;
```

☐   

```
SELECT T.nev, T.kor FROM TANULOK T  
INNER JOIN OSZTALYOK  
WHERE T.osztaly_id = O.osztaly_id  
AND O.osztaly_nev LIKE '%A'  
AND T.kor >= 16;
```

☐   

```
SELECT nev, kor FROM Diakok  
WHERE kor >= 16 AND osztaly_nev LIKE '%A';
```

## 4. feladat 1 pont

Adott egy TANULOK, valamint egy TANTARGYAK nevű tábla. Egészítsd ki az alábbi lekérdezést - a szaggatott vonalas résznél -, hogy minden tanulóhoz hozzá legyen kapcsolva minden tantárgy.

**SELECT \* FROM TANULOK**

**----- TANTARGYAK;**

### Válaszok

## 5. feladat 1 pont

Mi a különbség az **SQL** (relációs) és **NoSQL** (nem relációs) adatbázisok között?

### Válasz

- ☐ Az SQL adatbázisok csak strukturált adatok tárolására alkalmasak, míg a NoSQL adatbázisok alkalmasak mind strukturált, mind nem strukturált adatok tárolására.
- ☐ Az SQL adatbázisok csak kisebb méretű adatbázisokhoz használhatók, míg a NoSQL adatbázisok alkalmazhatók nagy méretű adatbázisokhoz.

- ☐ Az SQL adatbázisok előre definiált sémát követnek, míg a NoSQL adatbázisok dinamikusan változó struktúrájú adatok tárolására alkalmasak.
- ☐ Az SQL adatbázisok csak egyszerű lekérdezéseket és tranzakciókat támogatnak, míg a NoSQL adatbázisok sokféle lekérdezést és komplex tranzakciókat is támogatnak.

## 6. feladat 1 pont

Adott egy vállalati adatbázis a "**Termékek**" és "**Rendelések**" táblákkal. A "Termékek" tábla tartalmazza a különböző termékek adatait, beleértve a termékazonosítót (product\_id), a termék nevét és az árát. A "Rendelések" tábla tartalmazza a vásárlások adatait, beleértve a rendelésazonosítót (order\_id), a rendelő vásárló azonosítóját (customer\_id) és a termékazonosítót (product\_id), amely a vásárolt terméket azonosítja. Válaszd ki azt az SQL lekérdezést, amely meghatározza az öt legdrágább vásárolt terméket, és megjeleníti azokat a rendelések számával együtt, amelyekben ezek a termékek szerepelnek!

### Válasz

☐

```
SELECT P.product_name, P.product_price,
COUNT(*) AS order_count
FROM Products P
JOIN Orders O ON P.product_id = O.product_id
WHERE P.product_id IN (
    SELECT product_id
    FROM Products
    ORDER BY product_price DESC
    LIMIT 5
)
GROUP BY P.product_name, P.product_price
ORDER BY order_count DESC;
```

☐

```
SELECT P.product_name, P.product_price, COUNT(*) AS order_count
FROM Products P
JOIN Orders O ON P.product_id = O.product_id
WHERE P.product_id IN (
    SELECT product_id
    FROM Products
    ORDER BY product_price DESC
    LIMIT 5
)
GROUP BY P.product_name, P.product_price
ORDER BY P.product_price DESC;
```

```

SELECT P.product_name, P.product_price,
COUNT(*) AS order_count
FROM Products P
JOIN (
    SELECT product_id
    FROM Products
    ORDER BY product_price DESC
    LIMIT 5
) AS TopProducts ON P.product_id = TopProducts.product_id
JOIN Orders O ON P.product_id = O.product_id
GROUP BY P.product_name, P.product_price
ORDER BY P.product_price DESC;

```

```

SELECT P.product_name, P.product_price,
COUNT(O.order_id) AS order_count
FROM Products P
JOIN Orders O ON P.product_id = O.product_id
GROUP BY P.product_name, P.product_price
ORDER BY P.product_price DESC
LIMIT 5;

```

```

SELECT P.product_name, P.product_price,
COUNT(*) AS order_count
FROM Products P
JOIN Orders O ON P.product_id = O.product_id
WHERE P.product_id IN (
    SELECT product_id
    FROM Products
    WHERE product_price = (SELECT MAX(product_price)
                           FROM Products)
)
GROUP BY P.product_name, P.product_price
ORDER BY P.product_price DESC;

```

Megoldások beküldése