

# PROBLÉMA-ANALÍZIS ENTERPRISE RENDSZEREKBEN

3. forduló



A kategória támogatója: Adnovum Hungary Kft.

RENDELKEZÉSRE ÁLLÓ IDŐ:

10:00

## Ismertető a feladathoz

### Fontos információk

Ezután a forduló után automatikusan jár a [kitartóknak szóló garantált ajándékunk](#), érdemes kitöltened a feladatlapot! :)

Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat.

A kérdésekre **mindig van helyes válasz**, olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a [versenyszabályzatban](#) találsz!

### Harmadik forduló

Csapatunk egy dinamikusan fejlődő startup, termékünk nagyon sikeres így folyamatosan növekszik a felhasználóink száma. Emellett új komponensek és régiók kerülnek bevezetésre. Szükségünk van egy rátermett fejlesztőre, aki képes a felmerülő tervezési, biztonsági és teljesítmény problémákat megoldani. Elvállalod?

## 1. feladat 0/4 pont

Alkalmazásunkban az adatok jól strukturáltak és integritásuk kulcsfontosságú. A rendszert egy webalkalmazással fogjuk megvalósítani, amelyen keresztül az arra jogosult felhasználók léphetnek be és menedzselhetik ezeket az adatokat (tipikus CRUD műveletek, összetett keresés különböző szempontok alapján).

Milyen adatkezelő rendszert lenne a legcélszerűbb alkalmaznunk ehhez a rendszerhez?

### Válasz

- ☐ dokumentum adatbázis
- ☒ relációs adatbázis
- ☐ key-value store
- ☐ filerendszer

### Magyarázat

Strukturált adatoknál, ahol a konzisztencia az első számú szempont érdemes NoSQL megoldás helyett relációs adatbázist választani.

A NoSQL adatbázisok többsége nem valósítja meg az ACID elvárásokat és nem nyújt egyszerű módot kapcsolati-rendszer kialakítására.

## 2. feladat 0/8 pont

A következő DDL által leírt táblát szeretnénk mappelni JPA-val az Book osztályra Hibernate segítségével.

```
CREATE TABLE book (  
  id int,  
  name varchar(255),  
  isbn varchar(255),  
  author_id int,  
  PRIMARY KEY (id)  
  FOREIGN KEY (author_id) REFERENCES author(id)  
);
```

```
public class Book {  
    private Long id;  
    private String name;  
    private String isbn;  
    private Author author;  
  
    // ... getterek és setterek ...  
}
```

A rendszerben már adott egy author adatbázis tábla, amelyet a Author osztályra képeztünk le.

Az adott osztályon kívül mást nem módosíthatunk. Milyen annotációk hozzáadására van szükség ehhez?

### Válasz

- ☐ Semmire
- ☐ Entity, Table, Id, JoinColumn, OneToMany
- ☒ Entity, Table, Id, JoinColumn, ManyToOne
- ☐ Entity, Id, JoinColumn, OneToMany
- ☐ Table, Id, JoinColumn, ManyToOne

### Magyarázat

## 3. feladat 0/4 pont

Az alábbiak közül melyek igazak általában egy RESTful HTTP GET műveletre?

### Válaszok

- ☒ a paraméterek az URL-ben továbbítódnak
- ☒ sikeres lekéréskor HTTP 200-as kódot várunk vissza
- ☐ adatok módosítására szolgál
- ☒ idempotens művelet
- ☒ jól gyorsítótárazható

### Magyarázat

1. válasz : adatok a GET műveleteknél az URL-ben közlekednek. a HTTP törzsben például POST esetén közlekedik adat
2. válasz : adatok a GET műveleteknél az URL-ben közlekednek. a HTTP törzsben például POST esetén közlekedik adat

3. válasz: A GET adatok, erőforrások állapotának lekérésére szolgál

4. válasz: <https://hu.wikipedia.org/wiki/Idempotencia>

5. válasz: a kérés az erőforrás aktuális állapotának megfelelő választ ad vissza, ezért ismételt lekérdezésnél is ugyanazt kapnád feltéve hogy nem változott.

Legfontosabb tudnivalók

Kapcsolat

Versenyszabályzat

Adatvédelem

© 2022 Human Priority Kft.

KÉSZÍTETTE

Megjelenés

☀ Világos ⬆