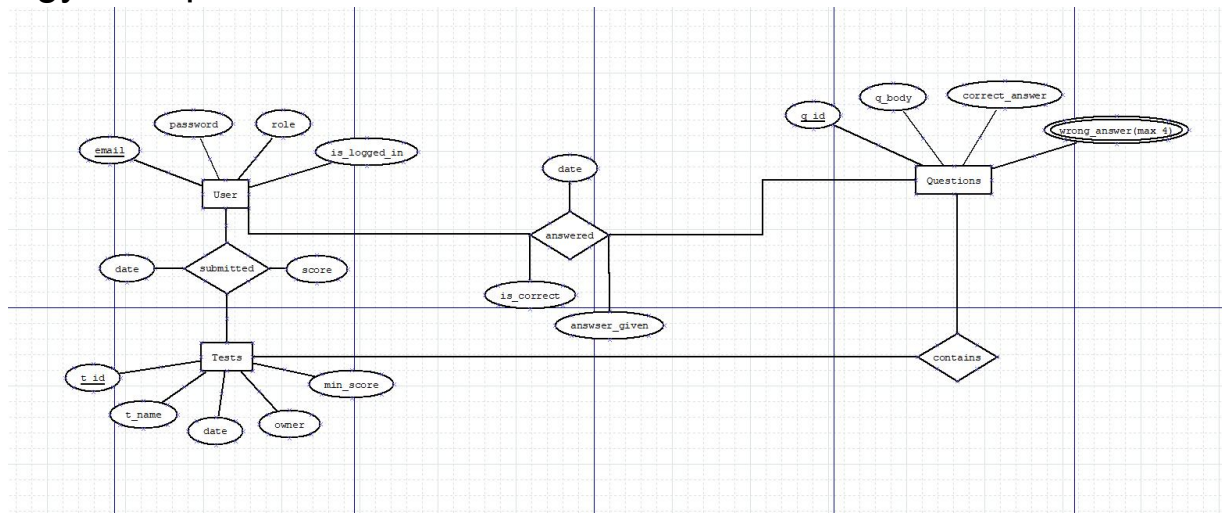


Moodle/CooSpace tesztek

Adatbázisok kötelező feladat

A program célja, hogy a programban bejelentkezett felhasználók (továbbiakban felhasználók) különböző tesztekkel tölthessenek ki. A programot regisztrált és vendég felhasználók használhatják. A vendég felhasználók csak regisztrálhatnak a felületre. Tesztet és hozzá tesztkérdéseket csak tanár vagy adminisztrátor hozhat létre, de bármilyen szerepkörű felhasználó kitöltheti a tesztet. Amikor a felhasználó belép, akkor látja a tesztek, amelyeket kitölthet. Azt is látja, hogy egy tesztet sikeresen töltött-e már ki vagy egyáltalán nem töltött ki, vagy nem sikerült elérni a minimális pontszámot.

Egyed-kapcsolat modell



- Felhasználó és Teszt_beküldés: Egy felhasználó több tesztet is beküldhet.
- Teszt és Kérdések: Minden teszt több kérdést is tartalmazhat.
- Felhasználó és Válaszok: Egy felhasználó több kérdésre is adhat választ.
- Kérdések és Válaszok: Egy kérdésre több felhasználó is adhat választ.

Relációs adatbázisséma

Users(email, name, password, role_id, is_logged_in)

Role(role_id, role_name)

Tests(tid, t_name, date, min_score, owner)

Questions(q_id, q_body, correct_answer)

Wrong_Answers(w_id, wrong_answer)

Question_Wrong_Answer(q_id, w_id)

Answers(a_id, submission_id, answer_given, is_correct, users_email, q_id)

Test_Submission(sub_id, owner, tid, score, date)

Test_Questions(tid, q_id)

Normalizálás

A táblák atomi értékeket tartalmaznak, tehát az 1NF-nek megfelel.

Az összes tábla teljesen függ az elsődleges kulcsoktól, így a 2NF-nek is megfelel.

Nincs tranzitív függőség a nem kulcs attribútumok között, tehát a 3NF-nek is megfelel.

Táblatervek

Users Tábla

Megnevezés	Típus	Megjegyzés
email	VARCHAR(255)	Elsődleges kulcs (PK)
name	VARCHAR(255)	Felhasználó neve
password	VARCHAR(255)	Felhasználó jelszava
role_id	INT	Külső kulcs (FK) Role(role_id)
is_logged_in	TINYINT(1)	Felhasználó bejelentkezési állapota

Role Tábla

Megnevezés	Típus	Megjegyzés
role_id	INT	Elsődleges kulcs (PK)
role_name	VARCHAR(100)	Szerep neve (pl. Tanár, Diák)

Tests Tábla

Megnevezés	Típus	Megjegyzés
tid	INT	Elsődleges kulcs (PK)
t_name	VARCHAR(255)	Teszt neve
date	DATE	Teszt dátuma
min_score	INT	Minimális pontszám
owner	VARCHAR(255)	Külső kulcs (FK) User(email)

Questions Tábla

Megnevezés	Típus	Megjegyzés
q_id	INT	Elsődleges kulcs (PK)
q_body	TEXT	Kérdés szövege
correct_answer	VARCHAR(255)	Helyes válasz

Wrong_Answers Tábla

Megnevezés	Típus	Megjegyzés
w_id	INT	Elsődleges kulcs (PK)
wrong_answer	VARCHAR(255)	Helytelen válaszok

Question_Wrong_Answers Tábla

Megnevezés	Típus	Megjegyzés
q_id	INT	Külső kulcs (FK) Questions(q_id)
w_id	INT	Külső kulcs (FK) Wrong_Answers(w_id)

Answers Tábla

Megnevezés	Típus	Megjegyzés
a_id	INT	Elsődleges kulcs (PK)
submission_id	INT	Teszt kitöltés azonosítója
answer_given	VARCHAR(255)	Adott válasz
is_correct	TINYINT(1)	Helyes válasz-e
Users_email	VARCHAR(255)	Külső kulcs (FK) Users(email)
q_id	INT	Külső kulcs (FK) Questions(q_id)

Test_Questions Tábla

Megnevezés	Típus	Megjegyzés
tid	INT	Külső kulcs (FK) Tests(tid)
q_id	INT	Külső kulcs (FK) Questions(q_id)

Test_Submission Tábla

Megnevezés	Típus	Megjegyzés
sub_id	INT	Elsődleges kulcs (PK)
owner	VARCHAR(255)	Külső kulcs (FK) Users(email)
tid	INT	Külső kulcs (FK) Tests(tid)
score	INT	Kitöltött teszt pontszáma
date	DATETIME	Beküldés dátuma és időpontja

Test_Questions Tábla

Megnevezés	Típus	Megjegyzés
tid	INT	Külső kulcs (FK) Tests(tid)
q_id	INT	Külső kulcs (FK) Questions(q_id)

Összetett lekérdezések

1. Összetett lekérdezés: JOIN és GROUP BY

/classes/class.test_submission.php

```
SELECT tests.tid, tests.t_name, tests.date, tests.min_score,  
tests.owner  
FROM tests JOIN test_submission ON tests.tid = test_submission.tid  
WHERE test_submission.owner = ?  
GROUP BY tests.t_name;
```

A lekérdezés visszaadja azoknak a teszteknek az adatait (teszt azonosító, név, dátum, minimális pontszám, tulajdonos), amelyeket egy bizonyos felhasználó kitöltött. A tesztek csoportosítva vannak a neveik szerint, így minden teszt csak egyszer szerepel a válaszban.

A lekérdezés célja tehát, hogy listázza az adott felhasználó által kitöltött teszteket, azokat az alapadatokat mutatva, amelyek a teszt neve alapján egyediek.

2. Összetett lekérdezés: JOIN, GROUP BY, belső lekérdezések

/classes/class.test_submission.php

```
SELECT
    u.name AS student_name,
    u.email AS student_email,
    COUNT(ts.tid) AS tests_completed,
    AVG(ts.score) AS avg_score,
    MAX(ts.score) AS best_score,
    (
        SELECT t.t_name
        FROM tests t
        INNER JOIN test_submission ts_inner ON t.tid = ts_inner.tid
        WHERE ts_inner.owner = u.email AND t.owner = ?
        ORDER BY ts_inner.date DESC
        LIMIT 1
    ) AS last_test_name,
    (
        SELECT ts_inner.score
        FROM tests t
        INNER JOIN test_submission ts_inner ON t.tid = ts_inner.tid
        WHERE ts_inner.owner = u.email AND t.owner = ?
        ORDER BY ts_inner.date DESC
        LIMIT 1
    ) AS last_test_score
FROM
    users u
LEFT JOIN
    test_submission ts ON u.email = ts.owner
LEFT JOIN
    tests t ON ts.tid = t.tid
WHERE
    t.owner = ?
GROUP BY
    u.email, u.name
ORDER BY
    avg_score DESC, tests_completed DESC;
```

A lekérdezés az adott tanár tesztjeivel kapcsolatos összes diákot és azok eredményeit listázza ki. Az adatok a következőket tartalmazzák:

- Diák neve és email címe
- A diák által kitöltött tesztek száma
- A diák átlagos pontszáma és legjobb pontszáma
- A diák legutóbbi tesztje és annak pontszáma

A lekérdezés végén az eredményeket az átlagpontszám és a kitöltött tesztek számának csökkenő sorrendjében rendezi, így a legjobban teljesítő diákok kerülnek az elejére.

3. Összetett lekérdezés

/classes/class.test.php

```
SELECT
    t.t_name AS t_name,
    COUNT(ts.owner) AS submission_count,
    ROUND(AVG(ts.score), 2) AS avg_points
FROM
    tests t
JOIN
    test_submission ts ON t.tid = ts.tid
GROUP BY
    t.t_name
ORDER BY
    avg_points DESC
LIMIT $amount;
```

A lekérdezés a tesztek neve, a hozzájuk tartozó beküldések száma és az átlagos pontszám alapján csoportosítja és rendezi a tesztek. Az eredményben szereplő tesztek az átlagos pontszám szerint csökkenő sorrendben adja vissza, és a \$amount változó által meghatározott számú rekordot korlátoz.

Megvalósítás, funkciók

A **Space Coo** egy tesztkitöltő alkalmazás, amely lehetővé teszi a felhasználók számára, hogy különböző tesztekot töltsenek ki. A rendszer kétféle felhasználói kategóriát támogat: regisztrált felhasználókat és vendég felhasználókat. A vendég felhasználók regisztrálhatnak, míg a regisztrált felhasználók hozzáférhetnek a tesztekhez és az azokhoz tartozó kérdésekhez. Csak tanárok és adminisztrátorok hozhatnak létre új tesztekot és kérdéseket, míg bármilyen szerepkörű felhasználó kitöltheti azokat. A felhasználók belépés után megtekinthetik a tesztekot, és információkat kaphatnak arról, hogy sikeresen kitöltötték-e már azokat, vagy ha nem, akkor a minimális pontszámot elérték-e.

Az alkalmazás **reszponzív**, tehát minden eszközön jól használható, és támogatja a **light mode** és **dark mode** megjelenést is. A felhasználók könnyedén navigálhatnak a tesztek között egy **élő keresőmotor** segítségével. Az alkalmazás **PHP** nyelven készült, a **Bootstrap 5.3** segítségével, amely biztosítja a modern és felhasználóbarát megjelenést, míg **JavaScript** script-ek biztosítják az interaktivitást.

A felhasználók nyomon követhetik, hogy mely tesztekot töltötték ki, és ha van rá lehetőség, akkor az online felhasználók is láthatóak az alkalmazásban. Ezen kívül **felhasználói profilkezelés** és **jelszóváltoztatás** is elérhető, így a felhasználók biztonságosan kezelhetik saját adataikat.