Adatbázis alapú rendszerek

2023-2024/2

# IB152L-10

**Quiz**

**Készítette:**

Kávai Illés

Kaszás Zsolt József

Velicán-Benkő Attila

## Munka felosztása

Közösen dolgozunk a feladatok megoldásán.

## Értékelési mód:

A csoport tagjai közösen dolgoznak, minden tag ugyanannyi pontot kap.

## Feladat szöveges leírása

A weboldal egy kvíz játékot kínál, amely lehetőséget biztosít a felhasználóknak saját szobák létrehozására. Ezekben a szobákban a felhasználók saját kérdéseket állíthatnak be, valamint válaszokat adhatnak hozzájuk. Más játékosok is csatlakozhatnak ezekhez a szobákhoz, és megmérkőzhetnek egymással, hogy mennyi kérdésre tudnak válaszolni az adott szobában.

A weboldal PHP-ban készül, az Oracle 12.0.0.0.1 adatbázist és a SQLDevelopert használjuk az adatbázis létrehozásához és kezeléséhez. A dokumentáció elkészítéséhez a Word programot választottuk. Az alkalmazás fejlesztése a Visual Studio Code/PHPStorm fejlesztői környezetben történik. Diagramok készítésére a ClickCharts-t használjuk.

Az adatbázis tartalmazza a következő hét fő adattáblát: felhasznalo, eredmeny, kerdes, tema, valasz, szoba, szoba\_kerdesei. Ezeket az adattáblákat az alkalmazás kezeli és frissíti.

Az alkalmazás belépő képernyővel indul, ahol a felhasználó bejelentkezhet a felhasználói fiókjába. Ha a felhasználó még nem regisztrált, akkor a regisztrációs űrlapot kell kitöltenie. Sikeres bejelentkezés után a felhasználó a főmenübe kerül, ahol lehetőség van szobát létrehozni, szobába csatlakozni, kijelentkezni, valamint profilját megtekinteni.

A szobába való belépést követően lehetőség van kitölteni a kvízt. A játék során mindig csak egy kérdés aktív, és a válaszok mindig csak erre a kérdésre vonatkoznak.

A játékmenet során a játékosok a kvíz kitöltésével versenyeznek, eredményeiket az adatbázisban tárolják. A játék során a játékosok különböző kérdésekre válaszolnak, amelyekhez a program az adattáblákból olvassa be a válaszokat, és a válaszok helyességét is az adatbázisban tárolja.

## Követelménykatalógus

Cél: Az alkalmazás célja, hogy interaktív és szórakoztató módon lehetőséget biztosítson a felhasználóknak szobákban vagy nyilvános helyeken való versengésre, ahol különböző témákhoz kapcsolódó kérdésekre válaszolhatnak.

Felhasználó belépése felhasználónév és jelszó segítségével:

A felhasználók beléphetnek a felhasználói fiókjukba felhasználónév és jelszó megadásával.

* Regisztráció:

A felhasználóknak lehetőségük van regisztrálni az alkalmazásba, ahol meg kell adniuk néhány alapvető információt, mint például felhasználónév, jelszó, email stb.

* Játékszobákhoz való csatlakozás:

A játékosoknak lehetőségük van csatlakozni más játékosok által létrehozott játékszobákhoz, ha ismerik a szoba nevét és jelszavát.

* Új játékszobák létrehozása:

A játékosoknak lehetőségük van új játékszobákat létrehozni, amelyeknek meg kell adniuk a szoba nevét és jelszavát.

* Kérdésekre való válaszadás:

A játékosoknak lehetőségük van válaszolni a véletlenszerűen megjelenített kérdésekre a játékszobákban.

* Pontszám szerzése helyes válaszokért:

A játékosok pontokat szereznek a helyes válaszokért, amelyeket az alkalmazás rögzít és megjelenít

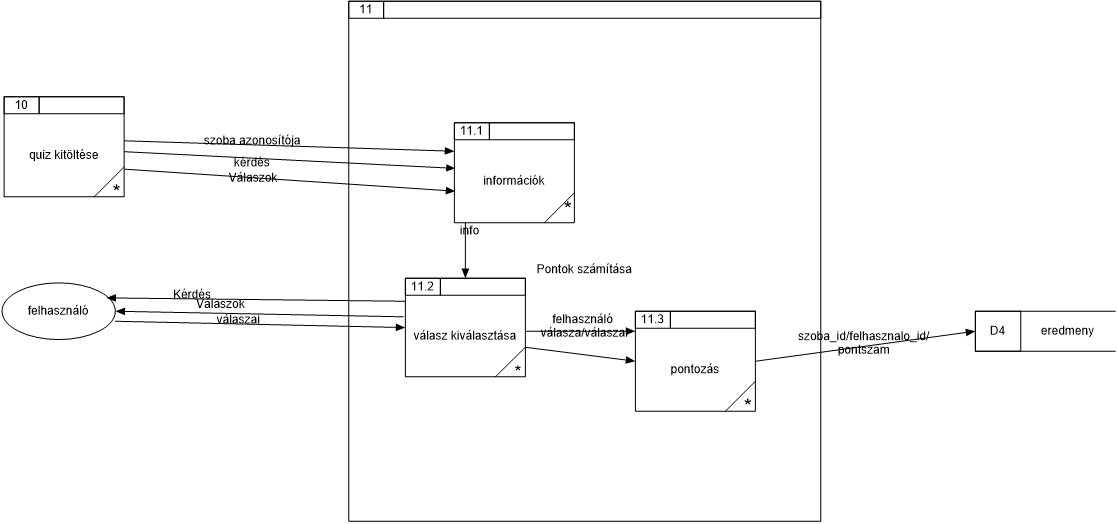
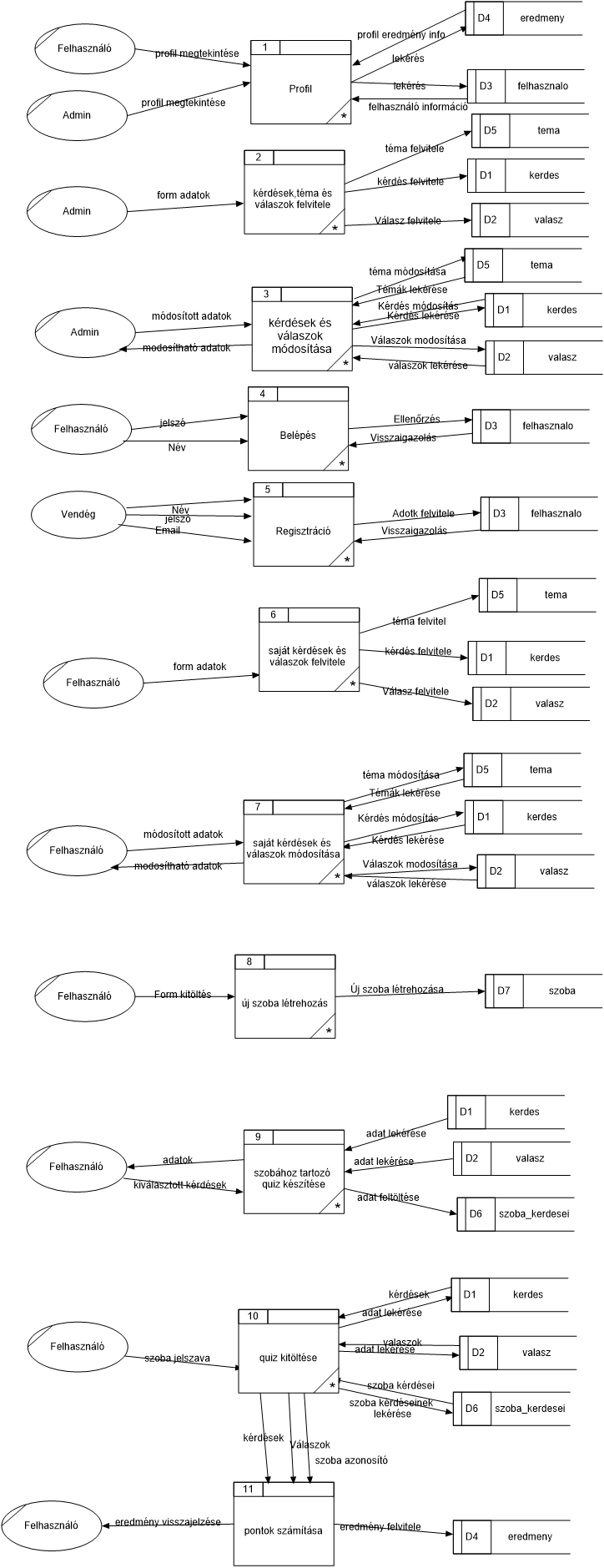
* Ha egy felhasználó törlésre kerül a rendszerből, és ehhez tartozik legalább egy játékszoba, akkor az ehhez a felhasználóhoz kapcsolódó szobákhoz tartozó felhasználói azonosítók automatikusan NULL értékre állnak.
* Az adminisztrátornak lehetősége van egy felhasználói fiókhoz üres felhasználónevet rendelni, amely lehetővé teszi a bejelentkezés letiltását a rendszerbe a megfelelő jogosultságokkal. Ez a funkció segítséget nyújt az adminisztrátornak, hogy a felhasználókat ideiglenesen "letiltsa" vagy "inaktívvá" tegye a rendszerben anélkül, hogy a felhasználói fiókot ténylegesen törölné.

Nem funkcionális követelmények:

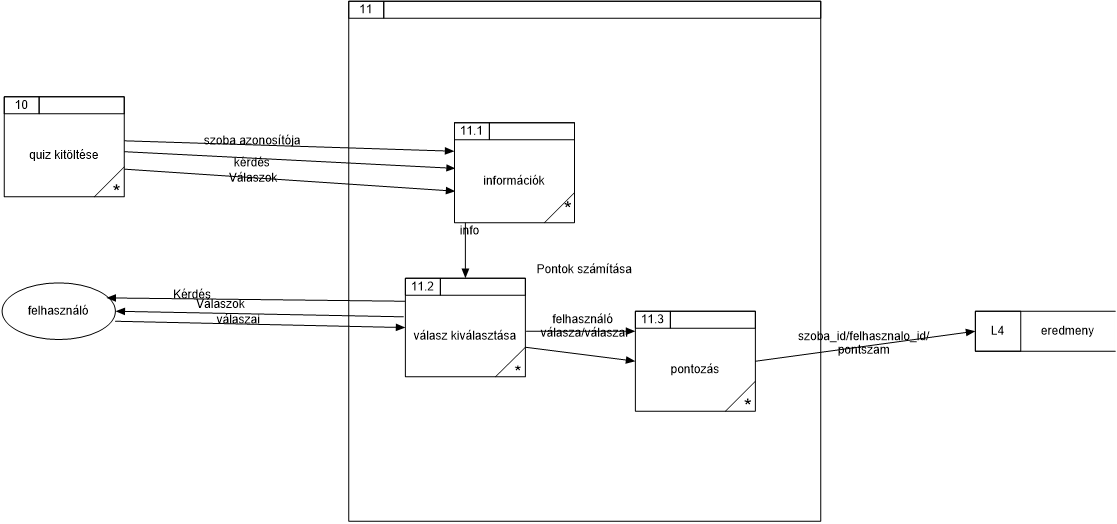
* Megbízhatóság
* Teljesítmény
* Felhasználói felület
* A programnak grafikus felhasználói felületet kell biztosítania a felhasználók számára.
* A felhasználói felületnek ergonomikusnak kell lennie, és az információkat jól átlátható módon kell megjelenítenie.
* A felhasználói felületnek tartalmaznia kell a szükséges gombokat és elemeket, amelyekkel a felhasználók könnyen használhatják a programot.
* Karbantarthatóság
* Biztonság
* A programnak biztosítania kell az adatok biztonságos tárolását és kezelését.
* A programnak lehetővé kell tennie a felhasználók jogosultságainak kezelését, például az adminisztrátori jogosultságokat.
* A programnak megfelelő védelmet kell biztosítania a bizalmas adatokkal szemben, például a jelszavak védelmével.

## Adatfolyam diagram (DFD):

Fizikai

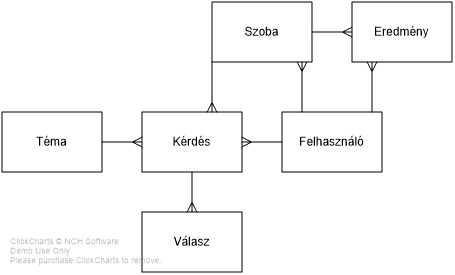


Logikai



## 

## Egyedmodell:



## EKT-diagram:

## 

**Egyed-kapcsolat diagram leképezése relációs adatbázissémákká:**

felhasznalo (id, nev, email, jelszo, admin-e)

eredmeny (id, *szoba\_id*, *felhasznalo\_id*, pontszam)

kerdes (id, kerdes, *tema\_id*)

tema (id, nev)

valasz (id, *kerdes\_id*, valasz ,helyes\_e)

szoba (id, nev, jelszo, *felhasznalo\_id*)

szoba\_kerdesei(id, *szoba\_id*, *kerdes\_id*)

eredmenyei(*felhasznalo.id* , *eredmeny.felhasznalo\_id)*

szobája(*felhasznalo.id* , *szoba.felhasznalo\_id*)

kerdesei(*kerdes.id* , *szoba\_kerdesei. kerdes\_id*)

## 

## Relációs adatelemzés

1. normálforma (1NF):

* felhasznalo: (id, nev, email, jelszo, admin-e)
* eredmeny: (id, szoba\_id, felhasznalo\_id, pontszam)
* kerdes: (id, kerdes, tema\_id)
* tema: (id, nev)
* valasz: (id, kerdes\_id, valasz ,helyes\_e)
* *szoba*: (id, nev, jelszo, felhasznalo\_id)
* szoba\_kerdesei: (id, szoba\_id, kerdes\_id)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 1NF-nek.

2. normálforma (2NF):

* felhasznalo: (id, nev, email, jelszo, admin-e)
* eredmeny: (id, *szoba\_id*, *felhasznalo\_id*, pontszam)
* kerdes: (id, kerdes, *tema\_id*)
* tema: (id, nev)
* valasz: (id, *kerdes\_id*, valasz ,helyes\_e)
* szoba: (id, nev, jelszo, *felhasznalo\_id*)
* szoba\_kerdesei: (id, *szoba\_id*, *kerdes\_id*)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 2NF-nek, hiszen minden táblának van egyetlen elsődleges kulcsa, és nincsenek benne összetett attribútumok.

3. normálforma (3NF):

* felhasznalo: (id, nev, email, jelszo, admin-e)
* eredmeny: (id, *szoba\_id*, *felhasznalo\_id*, pontszam)
* kerdes: (id, kerdes, *tema\_id*)
* tema: (id, nev)
* valasz: (id, *kerdes\_id*, valasz ,helyes\_e)
* szoba: (id, nev, jelszo, *felhasznalo\_id*)
* szoba\_kerdesei: (id, *szoba\_id*, *kerdes\_id*)

Megjegyzés: Az adatbázis már eleve kiindulópontjában megfelel az 3NF-nek, hiszen az egyes táblákban található attribútumok között nincs transzitív függőség.

## Táblák leírása:

felhasznalo: *felhasznalo tartalmának leírása.*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| nev | CHAR | felhasználó belépési neve |
| email | CHAR | Email- cím |
| jelszo | CHAR | jelszó |
| admin-e | INTEGER | adminisztrátor-e vagy sem |
| utolso\_aktivitas\_datum | TIMESTAMP | Az utolsó lepontozott eredmény idejét menti le vagy a regisztráció dátumát |

eredmeny: *Ez az adattábla tárolja a játékos eredményét egy adott szobában*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| szoba\_id | FOREIGN KEY | Szoba azonosítója |
| felhasznalo\_id | FOREIGN KEY | Felhasználó azonosítója |
| pontszam | INTEGER | pontszámuk |

kerdes: *Ez az adattábla tárolja az összes kérdést, amelyek az alkalmazásban megjelennek.*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| kerdes | CHAR | kérdés szövege |
| tema\_id | FOREIGN KEY | Téma azonosítója |
| felhasznalo\_id | FOREIGN KEY | Felhasználó azonosítója |
| globalis\_kerdes | NUMBER | egy kérdést bármelyik játékszobában meg lehet jeleníteni |

tema: *Ez az adattábla tárolja az összes témát, amelyek az alkalmazásban megjelennek.*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| nev | CHAR | Téma megnevezése |

valasz : *Ez az adattábla tartalmazza az összes választ, amelyeket a felhasználók az egyes kérdésekre adhatnak*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| kerdes\_id | FOREIGN KEY | válaszhoz tartozó kérdés azonosítója |
| valasz | CHAR | válasz szövege |
| helyes\_e | INTEGER | A kérdése vonatkozó helyesség |

*szoba: Ez az adattábla tartalmazza a szoba információját*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| nev | CHAR | szoba neve |
| jelszo | CHAR | jelszó |
| felhasznalo\_id | FOREIGN KEY | Azon felhasználó azonosítója, aki a szobát létrehozta |
| utolso\_aktivitas\_datum | TIMESTAMP | Az utolsó lepontozott eredmény idejét menti le vagy a szoba létrehozásának dátumát |

szoba\_kerdesei : *Ez az adattábla tartalmazza az adott szobához tartozó kérdéseket*

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| id | NUMBER | egyedi azonosító |
| szoba\_id | FOREIGN KEY | Szoba azonosítója |
| kerdes\_id | FOREIGN KEY | válaszhoz tartozó kérdés azonosítója |

## Funkció-szerep mátrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Funkciók /Szerepkörök | Felhasználó | Admin | Vendég | | Regisztráció |  |  | x | | Belépés | x | x |  | | Profil | x | x |  | | saját kérdések és válaszok felvitele | x | x |  | | saját kérdések és válaszok módosítása | x | x |  | | új szoba létrehozás | x | x |  | | szobához tartozó quiz készítése | x | x |  | | quiz kitöltése | x | x |  | | kérdések és válaszok módosítása |  | x |  | | kérdések,téma és válaszok felvitele |  | x |  | |

**Hozzáadott triggerek:**

**Trigger: update\_last\_activity\_trigger**

Leírás:

Ez a trigger minden alkalommal frissíti a felhasznalo tábla utolso\_aktivitas\_datum mezőjét, amikor egy új rekordot szúrnak be az eredmeny táblába, és az új rekordnak van egy felhasznalo\_id értéke, amely megegyezik a felhasznalo tábla egyik rekordjának azonosítójával.

Kód:

CREATE OR REPLACE TRIGGER update\_last\_activity\_trigger

BEFORE INSERT ON eredmeny

FOR EACH ROW

BEGIN

UPDATE felhasznalo

SET utolso\_aktivitas\_datum = CURRENT\_TIMESTAMP

WHERE id = :new.felhasznalo\_id;

END;

/

**Trigger: update\_room\_last\_activity\_trigger**

Leírás:

Ez a trigger minden alkalommal frissíti a szoba tábla utolso\_aktivitas\_datum mezőjét, amikor új rekordot szúrnak be, meglévő rekordot frissítenek vagy rekordot törölnek a szoba\_kerdesei táblában, és az adott rekordokhoz tartozó szoba\_id érték alapján azonosítja a megfelelő szobát.

Kód:

CREATE OR REPLACE TRIGGER update\_room\_last\_activity\_trigger

AFTER INSERT OR UPDATE OR DELETE ON szoba\_kerdesei

FOR EACH ROW

DECLARE

v\_room\_id szoba.id%TYPE;

BEGIN

-- Szoba azonosítójának lekérdezése az érintett sor alapján

IF INSERTING THEN

v\_room\_id := :new.szoba\_id;

ELSIF UPDATING THEN

v\_room\_id := :new.szoba\_id;

ELSIF DELETING THEN

v\_room\_id := :old.szoba\_id;

END IF;

-- Szoba utolsó aktivitás dátumának frissítése

UPDATE szoba

SET utolso\_aktivitas\_datum = CURRENT\_TIMESTAMP

WHERE id = v\_room\_id;

END;

/

**Hozzáadott eljárások:**

**Eljárás: get\_average\_score**

Leírás:

Ez az eljárás egy felhasználó által elért átlagos pontszám lekérdezését végzi a megadott felhasználó azonosítója alapján. Az eredményt az avg\_score kimeneti paraméterben adja vissza.

Kód:

CREATE OR REPLACE PROCEDURE get\_average\_score(user\_id IN NUMBER,avg\_score OUT NUMBER)

IS

BEGIN

SELECT AVG(pontszam)

INTO avg\_score

FROM eredmeny

WHERE felhasznalo\_id = user\_id;

END;

/

**Eljárás: get\_all\_room\_results**

Leírás:

Ez az eljárás az összes szobához tartozó eredményeket kérdezi le.

Kód:

CREATE OR REPLACE PROCEDURE get\_all\_room\_results (

result OUT SYS\_REFCURSOR

)

IS

BEGIN

OPEN result FOR

SELECT e.szoba\_id, e.felhasznalo\_id, e.pontszam

FROM eredmeny e;

END;

/

**Eljárás: get\_toplist\_for\_room**

Leírás:

Ez az eljárás az adott szobához tartozó legjobb eredményeket kéri le. A felhasználó nevét és a pontszámot adja vissza egy kimeneti kurzoron keresztül, rendezve a pontszámok csökkenő sorrendjében.

Kód:

CREATE OR REPLACE PROCEDURE get\_toplist\_for\_room (

room\_id IN NUMBER,

result OUT SYS\_REFCURSOR

)

IS

BEGIN

OPEN result FOR

SELECT f.nev AS felhasznalonev, e.pontszam

FROM eredmeny e

JOIN felhasznalo f ON e.felhasznalo\_id = f.id

WHERE e.szoba\_id = room\_id

ORDER BY e.pontszam DESC;

END;

/

**Összetett lekérdezések:**

**mod\_account.php 61-69**

SELECT

            felhasznalo.id As felhasznalo\_id,

            felhasznalo.email As felhasznalo\_email,

            felhasznalo.nev AS felhasznalo\_neve,

            TO\_CHAR(felhasznalo.utolso\_aktivitas\_datum, 'YYYY-MM-DD HH24:MI:SS') AS utolso\_aktivitas\_datum,

            MAX(COALESCE(eredmeny.pontszam, 0)) AS utolso\_pontszam

        FROM felhasznalo

        LEFT JOIN eredmeny ON felhasznalo.id = eredmeny.felhasznalo\_id

        GROUP BY felhasznalo.id, felhasznalo.email, felhasznalo.nev, felhasznalo.utolso\_aktivitas\_datum

A lekérdezés két részből áll:

1. Az első rész egy al-lekérdezés, amely összefűzi a felhasználók adatait és az utolsó pontszámokat.
2. A második részben a fő lekérdezés az al-lekérdezés eredményeit dolgozza fel.

Az LEFT JOIN kapcsolatot használva a lekérdezés összekapcsolja a felhasznalo és az eredmeny táblákat a felhasználók azonosítói alapján. A GROUP BY utasítás a felhasználók adatait csoportosítja az azonosítójuk alapján, és kiszámítja az utolsó pontszámot az összes eredmény maximuma alapján.

A COALESCE függvényt használva biztosítják, hogy az üres (NULL) értékeket 0-ra cseréljék a pontszámokban. A TO\_CHAR függvény segítségével formázva van az utolsó aktivitás dátuma a megadott formátumra.

**home.php 28-33**

SELECT sk.szoba\_id, s.nev AS szoba\_neve, COUNT(k.id) AS kerdesek\_szama

            FROM szoba\_kerdesei sk

            JOIN kerdes k ON sk.kerdes\_id = k.id

            JOIN szoba s ON sk.szoba\_id = s.id

            GROUP BY sk.szoba\_id, s.nev

            ORDER BY COUNT(k.id) DESC

A szoba\_kerdesei táblát szoba és kerdes táblákkal összekapcsolva meghatározza az összes szobát, amelyekhez vannak kérdések rendelve. Ezután csoportosítja ezeket a szobákat a szoba azonosítója és neve alapján, és megszámolja a hozzájuk tartozó kérdések számát. Végül a lekérdezés csökkenő sorrendbe rendezve listázza ezeket a szobákat a kérdések száma szerint.

**home.php 73-78**

SELECT t.nev AS tema\_neve, COUNT(\*) AS eloformulasok\_szama

            FROM tema t

            JOIN kerdes k ON t.id = k.tema\_id

            GROUP BY t.nev

            ORDER BY COUNT(\*) DESC

            FETCH FIRST 5 ROWS ONLY

Ez a lekérdezés az adatbázisból visszanyeri az öt legnépszerűbb témát a kérdések alapján. Az eredmények között szerepel a téma neve és az élőformulák száma, ami a téma alatt lévő aktív kérdések számát jelenti.

**home.php 112-116**

SELECT s.nev AS szoba\_neve, COUNT(DISTINCT e.felhasznalo\_id) AS aktiv\_felhasznalok\_szama

            FROM eredmeny e

            JOIN szoba s ON e.szoba\_id = s.id

            GROUP BY s.nev

            ORDER BY COUNT(DISTINCT e.felhasznalo\_id) DESC

Ez a lekérdezés a szobákhoz tartozó aktív felhasználók számát határozza meg. Az eredmények között szerepel a szoba neve és az aktív felhasználók száma, amelyek a szobában kitöltött eredmények alapján kerülnek meghatározásra.

**home.php 155-159**

SELECT s.nev AS szoba\_neve, COUNT(e.id) AS kitoltesek\_szama

            FROM szoba s

            JOIN eredmeny e ON s.id = e.szoba\_id

            GROUP BY s.nev

            ORDER BY COUNT(e.id) DESC

Ez a lekérdezés meghatározza a szobákhoz tartozó kitöltések számát az eredmények alapján. Az eredmények között szerepel a szoba neve és a kitöltések száma, ami azt mutatja, hogy egy adott szobában hány eredményt rögzítettek összesen.

home.php 198-204

SELECT \* FROM (

            SELECT f.nev AS felhasznalo\_neve, f.email AS felhasznalo\_email, COALESCE(SUM(e.pontszam), 0) AS osszesitett\_pontszam

            FROM felhasznalo f

            LEFT JOIN eredmeny e ON f.id = e.felhasznalo\_id

            GROUP BY f.nev, f.email

            ORDER BY COALESCE(SUM(e.pontszam), 0) DESC

        ) WHERE ROWNUM <= 5

Ez a lekérdezés meghatározza az öt legaktívabb felhasználót a legtöbb pontszámmal az eredmények alapján. Az eredmények között szerepel a felhasználó neve, e-mail címe és az összesített pontszáma. Csak az első 5 rekordot jeleníti meg.

quiz.php 85.-91

SELECT k.tema\_id, k.id AS kerdes\_id, k.kerdes, v.id AS valasz\_id, v.valasz, v.helyes\_e

        FROM szoba\_kerdesei sk

        JOIN kerdes k ON sk.kerdes\_id = k.id

        JOIN valasz v ON k.id = v.kerdes\_id

        WHERE sk.szoba\_id = :room\_id

        GROUP BY k.tema\_id, k.id, k.kerdes, v.id, v.valasz, v.helyes\_e

        ORDER BY k.tema\_id

Ez a lekérdezés kiválasztja egy adott szoba összes kérdését és az ezekhez tartozó válaszokat. A szoba\_kerdesei, kerdes és valasz táblákat összekapcsolva megkapjuk a szükséges adatokat. Az eredményeket csoportosítja tema\_id alapján, majd rendezve jeleníti meg a tema\_id szerinti sorrendben.

quiz.sql 130-143

CREATE OR REPLACE PROCEDURE get\_toplist\_for\_room (

    room\_id IN NUMBER,

    result OUT SYS\_REFCURSOR

)

IS

BEGIN

    OPEN result FOR

        SELECT f.nev AS felhasznalonev, e.pontszam

        FROM eredmeny e

        JOIN felhasznalo f ON e.felhasznalo\_id = f.id

        WHERE e.szoba\_id = room\_id

        GROUP BY f.nev, e.pontszam

        ORDER BY e.pontszam DESC;

END;

/

Ez a tárolt eljárás visszaadja egy adott szoba legjobbjait a megadott szoba azonosítója alapján. A szoba összes eredményét lekéri, majd ezeket a felhasználó neve és pontszáma szerint csoportosítja. Végül a pontszámok szerint csökkenő sorrendben adja vissza az eredményeket.

new\_question.php 77.sor

SELECT k.id AS question\_id, k.kerdes AS question\_text, v.id AS answer\_id, v.valasz AS answer\_text, v.helyes\_e AS is\_correct, k.globalis\_kerdes AS is\_global FROM kerdes k LEFT JOIN valasz v ON k.id = v.kerdes\_id