

Táblák részletes magyarázata

1. lockers

- **locker_id**
- **status**: Jelzi a szekrény aktuális állapotát, ami lehet "nyitott" vagy "zárt".
- **can_be_opened**: Ez a mező jelzi, hogy a szekrény nyitható-e. Alapértelmezésben true, tehát tanári/adminisztrátori jóváhagyás nem szükséges a nyitáshoz.

2. students

- **student_id**
- **full_name**: A diák teljes neve.
- **class**: A diák osztálya.
- **birth_place**: A diák születési helye.
- **birth_date**: A diák születési dátuma.
- **rfid_tag**: A diák RFID azonosítója, amely egyedi és összekapcsolódik a szekrényével.

3. locker_access_history

- **access_id**
- **rfid_tag**: Az RFID tag, amely a diákhoz van rendelve. Kapcsolódik a students táblához.
- **locker_id**: A szekrény azonosítója, amelyet a diák ki akar nyitni. Kapcsolódik a lockers táblához.
- **access_time**: A hozzáférés időpontját rögzíti.
- **approved_by**: Annak az adminisztrátornak, tanárnak az azonosítója, aki jóváhagyta a hozzáférést. NULL lehet, ha nem szükséges jóváhagyás.

4. locker_relationships

- **relationship_id**
- **rfid_tag**: A diák RFID tag-je. Kapcsolódik a "students" táblához.
- **locker_id**: A diákhoz rendelt szekrény azonosítója. Kapcsolódik a "lockers" táblához.

Ez a tábla biztosítja, hogy minden diák csak a saját szekrényét tudja kinyitni.

5. subjects

- **subject_id**
- **subject_name**: A tantárgy neve.
- **teacher_name**: A tantárgyat tanító tanár neve.

6. timetables

- **timetable_id**
- **student_id**: A diák azonosítója, amely a "students" táblához kapcsolódik.
- **subject_id**: A tantárgy azonosítója, amely a "subjects" táblához kapcsolódik.
- **day_of_week**: Az adott nap az órarendben (hétfő-péntek).
- **start_time**: Az óra kezdetének időpontja.
- **end_time**: Az óra végének időpontja.

7. **admins**

- **admin_id**: Egyedi azonosító minden adminisztrátor (tanár vagy dolgozó) számára.
- **full_name**: Az adminisztrátor teljes neve.
- **position**: Az adminisztrátor pozíciója, például "igazgatóhelyettes", "Rendszergazda", "Portás".

Rendszer működése

- A diákok minden reggel beolvastatják az RFID tag-jüket. Ez a rendszer lekérdezi a "locker_relationships" táblából, hogy melyik szekrényhez tartoznak, majd a "lockers" táblában ellenőrzi, hogy a szekrény nyitható-e.
- Ha a szekrény nyitásához tanári vagy adminisztrátori jóváhagyás szükséges, a rendszer ellenőrzi a "timetables" és "subjects" táblát, és rögzíti, hogy ki hagyhatja jóvá a nyitást.
- Az adminisztrátorok (igazgatóság vagy dolgozók) az "admins" táblában, a tanárok pedig a "subjects" táblában vannak nyilvántartva, és ők hagyhatják jóvá a hozzáférést egy adott szekrényhez.
- A rendszer minden szekrényhez és diákhoz külön kapcsolódási naplót vezet, biztosítva a hozzáférések nyomon követhetőségét és ellenőrizhetőségét.

1. Kapcsolat az RFID tag és a szekrény között:

Minden RFID tag-et hozzárendelsz egy adott szekrényhez az adatbázisban. Ez a kapcsolat a "locker_relationships" táblában történik.

Az RFID tag egyedi azonosítót kap, amelyet összekapcsolsz a megfelelő szekrénnel.

2. Adatbázis logika:

A "locker_relationships" táblában minden egyes RFID tag egy adott szekrényhez van hozzárendelve. Amikor egy diák beolvassa az RFID tag-et, az adatbázisból lekérdezed, hogy az adott tag melyik szekrényt nyithatja ki.

3. RFID beolvasás logika:

A beolvasott RFID tag azonosítóját elküldöd a szervernek, amely lekérdezi az adatbázisból a megfelelő szekrény azonosítóját (locker_id).

Ezután ellenőrzöd, hogy az adott szekrény nyitható-e (can_be_opened mező), és ha igen, akkor a szerver jelzést küld a szekrény zárjának, hogy nyissa ki.

Példa az adatbázis-lekérdezésre:

sql

```
SELECT lockers.locker_id, lockers.can_be_opened
FROM locker_relationships
INNER JOIN lockers ON locker_relationships.locker_id = lockers.locker_id
WHERE locker_relationships.rfid_tag = '1234567890';
```

Ez a lekérdezés megadja az RFID tag-hez kapcsolódó szekrény azonosítóját és azt is, hogy nyitható-e a szekrény.

4. Zár vezérlése:

Ha a szekrény nyitható, akkor küld egy nyitási parancsot a szekrény zárjának, amely kinyitja a szekrényt.

Folyamat lépései:

- Diák beolvassa az RFID tag-et.
- A rendszer lekérdezi az adatbázisból a tag-hez tartozó szekrényt.
- Ellenőrzi, hogy a szekrény nyitható-e.
- Ha igen, a zárnyitási parancsot kiadja a megfelelő szekrénynek.

Hardveres megvalósítás:

- Raspberry Pi, ezen futna a szoftver amely kezli a szekrényeket
- RFID tag, a diákok a tag-eket letudják csippantani az RFID leolvasónál
- RFID leolvasó, beolvassa az RFID tag-eket, és a hozzá rendelt szekrényt kinyitja

Backend:

- NodeJs
- Python

Frontend:

- HTML,CSS
- JS
- Bootstrap