

Iskolai étkezéskezelő rendszer

Adatmodell dokumentáció

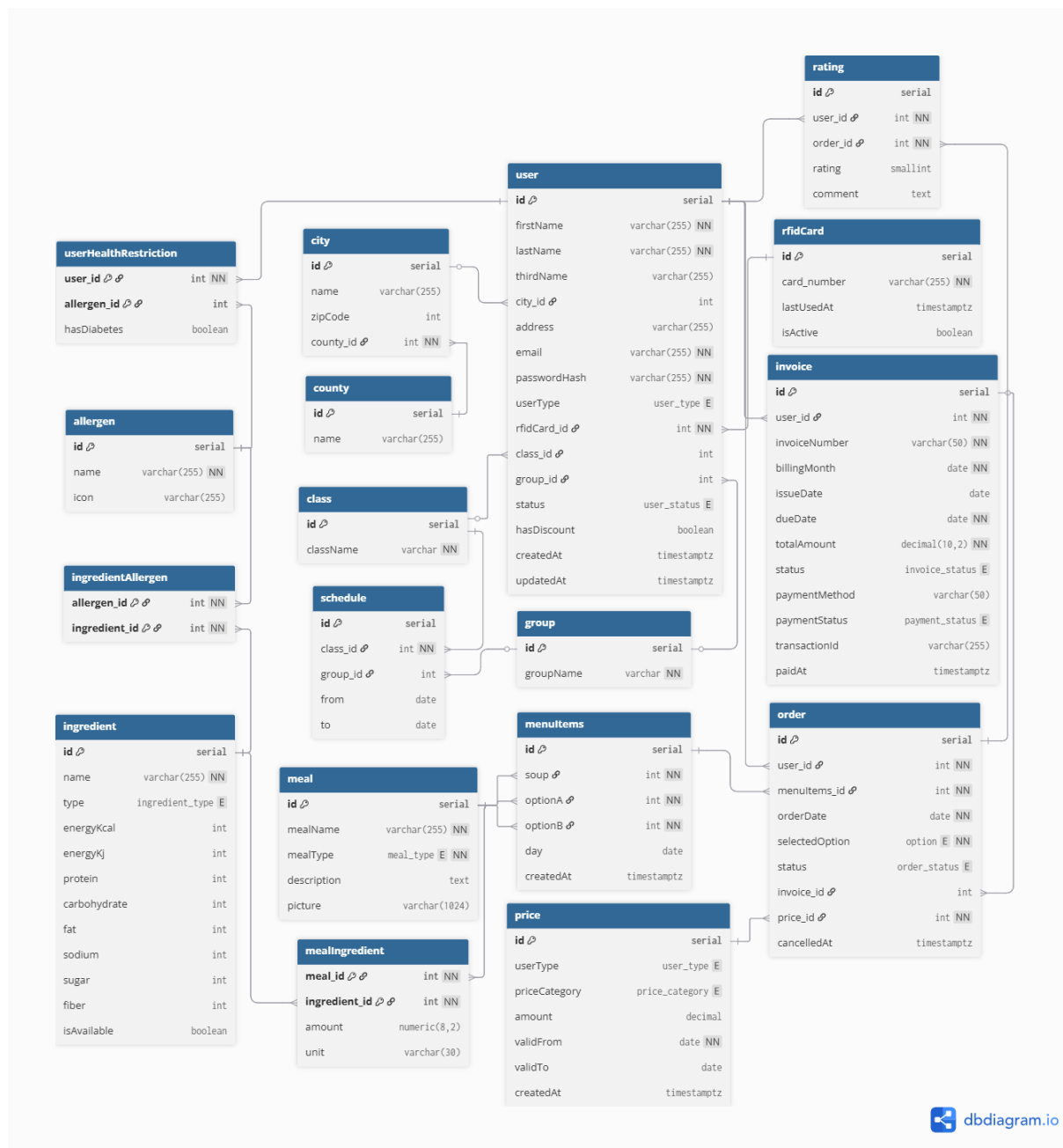


Készítették: Atkári Ariella
Farkas Dominik

Tartalomjegyzék

1.	Adatbázis táblái és mezői	2
2.	Fő táblák részletes ismertetése	3
2.1	User tábla (Felhasználók)	3
2.2	Order tábla (Rendelések)	4
2.3	MenuItem tábla (Napi menük)	4
2.4	Meal tábla (Ételek)	4
3.	Allergén és táplálkozási információk	5
3.1	Ingredient tábla (Hozzávalók)	5
3.2	Allergen tábla (Allergének)	5
4.	Kapcsolótáblák	6
4.1	MealIngredient tábla	6
4.2	IngredientAllergen tábla	6
4.3	UserHealthRestriction tábla (Felhasználók étkezési korlátozásai)	6
5.	Pénzügyi táblák	7
5.1	Invoice tábla (Számlák)	7
5.2	Price tábla (Árak)	7
6.	Egyéb táblák	8
6.1	Class tábla (Osztályok)	8
6.2	Group tábla (Csoportok)	8
6.3	Schedule tábla (Étkezés ütemezése)	8
6.4	County és City táblák (Vármegyék és városok)	8
6.5	RfidCard tábla (RFID Kártyák)	9
6.6	Rating tábla (Értékelések)	9

1. Adatbázis táblái és mezői



2. Fő táblák részletes ismertetése

2.1 User tábla (Felhasználók)

A rendszer összes felhasználójának alapvető adatainak tárolása és kezelése a célja. Ez a tábla szolgál a hitelesítés, autorizáció és a személyes adatok nyilvántartására.

Regisztrálás kizárólag az iskolai email és egy választott jelszó használatával történik meg, így könnyedén áthozhatóak a felhasználó szükséges adatai az iskolai adatbázisból: vezetéknév, keresztnév, harmadik név (**firstName**, **lastName**, **thirdName**), és lakcím adatok: vármegye (**county_id**), város (**city_id**), cím (**address**).

A **userType** megmutatja, hogy mi a szerepköre az adott felhasználónak. Lehet diák, tanár vagy dolgozó, és külsős személy, aki rendelhet étkezést. Ezen kívül még két típus van, az admin aki kezeli az oldalt, és a konyha, aki az étkeztető fél.

Az **rfidCard_id** használatával tudjuk kideríteni, hogy melyik felhasználó melyik kártyát kapta meg.

Az felhasználók van **class_id**-ja, ami rámutat arra, hogy pontosan melyik osztály tagja. A **group_id** is hasonló, azonban nem minden felhasználó tagja valamely csoportnak, így ez az érték lehet definiálatlan.

Alapból egy regisztrált fiók állapota (**status**) inaktív. Akkor lesz aktív, ha az admin elfogadja a regisztrációs kérelmet. Ezeken kívül az állapot lehet még felfüggesztett (**suspended**), ami akkor használatos, amikor egy felhasználó elhagyja az iskolát (elballag), és már nem innen fog rendelni.

Fontos lekezelni azt, hogy ha a felhasználó kedvezményesen étkezik. Ezt az admin tudja beállítani a **hasDiscount** résznél, ha a felhasználó benyújtja a megfelelő dokumentumokat hozzá személyesen.

A létrehozás időpontja a **createdAt**, a legutolsó módosításé pedig a **updatedAt**.

user	
id 	serial
firstName	varchar(255) NN
lastName	varchar(255) NN
thirdName	varchar(255)
city_id 	int
address	varchar(255)
email	varchar(255) NN
passwordHash	varchar(255) NN
userType	user_type E
rfidCard_id 	int NN
class_id 	int
group_id 	int
status	user_status E
hasDiscount	boolean
createdAt	timestampz
updatedAt	timestampz

2.2 Order tábla (Rendelések)

A felhasználók által leadott étkezési rendelések teljes életciklusának nyomon követésére szolgál.

A rendelésekhez kell a **user_id** és a **menuItems_id** annak érdekében, hogy tudjuk melyik felhasználó melyik napra/napokra rendelt ebédet.

Eltárolja a rendelés dátumát (**orderDate**) és a felhasználó által választott opciót. A **selectedOption** részénél van letárolva, hogy „A” vagy „B” főételből kér a felhasználó.

A rendelésnek is van egy státusza (**status**), mely megmutatja, hogy az állapota rendelt, lemondott vagy fizetett. Ha le lett mondva, akkor a pontos ideje a **cancelledAt** lesz letárolva

Az **invoice_id** rámutat, melyik számlához tartozik a rendelés. A megfelelő végösszeg számításához ki kell választani az árazást a **prices_id**-val.

order	
id 🔗	serial
user_id 🔗	int NN
menuItems_id 🔗	int NN
orderDate	date NN
selectedOption	option E NN
status	order_status E
invoice_id 🔗	int
price_id 🔗	int NN
cancelledAt	timestampz

2.3 MenuItem tábla (Napi menük)

Minden nap más-más ebédet készít a szolgáltató, így le kell tárolni előre, hogy az elkövetkezendő napokban (**day**) mik várhatóak. Mind a leves (**soup**) és a két főétel opció (**optionA**, **optionB**) is rámutat a meal tábla egy rekordjára, így kiválasztva a menü opciókat.

menuItems	
id 🔗	serial
soup 🔗	int NN
optionA 🔗	int NN
optionB 🔗	int NN
day	date
createdAt	timestampz

2.4 Meal tábla (Ételek)

Ez a tábla tartalmazza az összes ismert étel nevét (**mealName**) és kategóriáját (**mealType**), ami leves vagy főétel.

A megfelelő tájékozódás érdekében a **description**-nél lehet részletesebben leírni, hogy pontosan miből is áll az étel. Emellett a **picture**, vagyis egy kép az ételről is hozzájárul az étel egyszerű felismeréséhez.

meal	
id 🔗	serial
mealName	varchar(255) NN
mealType	meal_type E NN
description	text
picture	varchar(1024)


3. Allergén és táplálkozási információk

3.1 Ingredient tábla (Hozzávalók)

A különböző ételek számos hozzávalóból készülnek el, így ezeket hasznos letárolni. Egy hozzávalónak van neve (**name**), és egy típusa (**type**), ami megmutatja hogy hús, tejtermék, gabona, stb.

Ahhoz hogy egy étel tápértékét meg tudjuk határozni, fel kell vinni a hozzávalók értékeit:

- kalória (**energyKcal**),
- energia (**energyKj**),
- fehérje (**protein**),
- szénhidrát (**carbohydrate**),
- zsír (**fat**),
- sótartalom (**sodium**),
- cukor (**sugar**),
- rost (**fiber**)

ingredient	
id 	serial
name	varchar(255) NN
type	ingredient_type E
energyKcal	int
energyKj	int
protein	int
carbohydrate	int
fat	int
sodium	int
sugar	int
fiber	int
isAvailable	boolean

Előfordulhat, hogy egy hozzávaló nem elérhető. Ez az **isAvailable** mezőnél kap helyet.

A mealIngredient kapcsolótábla köti össze, hogy melyik ételhez milyen összetevők kellenek és pontosan mennyi (**amount**, **unit**).

3.2 Allergen tábla (Allergének)

Egy étkezéssel foglalkozó szolgáltatásnak kötelessége az allergénekre felhívni a figyelmet. Ennek következtében el kell tárolnunk az előforduló allergének neveit (**name**) és ikonjait (**icon**). Emellett az allergének egyszerű azonosításához használt **id**-t.

allergen	
id 	serial
name	varchar(255) NN
icon	varchar(255)

4. Kapcsolótáblák

4.1 MealIngredient tábla

Az ételek és hozzávalók több-a-többhöz kapcsolatának kezelése, mennyiségi információkkal.

mealIngredient	
meal_id	int NN
ingredient_id	int NN
amount	numeric(8,2)
unit	varchar(30)

4.2 IngredientAllergen tábla

A hozzávalók és allergének kapcsolatának nyilvántartása, hogy automatikusan lehessen észlelni az allergén veszélyeket.

ingredientAllergen	
allergen_id	int NN
ingredient_id	int NN

4.3 UserHealthRestriction tábla (Felhasználók étkezési korlátozásai)

A felhasználók étkezési korlátozásainak és allergiáinak nyilvántartása.

Az **user_id** mutat rá, hogy melyik felhasználónak milyen étkezési korlátozásai vannak.

userHealthRestriction	
user_id	int NN
allergen_id	int
hasDiabetes	boolean

Az **allergen_id**-val lehet a felhasználóhoz kötni, hogy pontosan mire allergiás.

Fontos figyelembe venni a cukorbetegeket is, így azt le kell tárolni (**hasDiabetes**).



5. Pénzügyi táblák

5.1 Invoice tábla (Számlák)

Célja a havi számlák generálásának, nyomon követésének és fizetési állapotának kezelése.

A rendszer a számláknak a számát (**invoiceNumber**) automatikusan generálja. Emellett fontos tudni, hogy:

- kihez tartozik a számla (**user_id**),
- melyik hónapra szól (**billingMonth**),
- mikor adták ki (**createdAt**),
- kiadás dátuma (**issueDate**)
- befizetési határidő (**dueDate**),
- befizetendő összeg (**totalAmount**)
- fizetés státusza (**status**), ami lehet: kiadott, fizetett
- tranzakció azonosítója (**transactionId**), ami bizonyítja a sikeres fizetést
- mikor lett kifizetve (**paidAt**)

invoice	
id	serial
user_id	int NN
invoiceNumber	varchar(50) NN
billingMonth	date NN
issueDate	date
dueDate	date NN
totalAmount	decimal(10,2) NN
status	invoice_status E
paymentMethod	varchar(50)
paymentStatus	payment_status E
transactionId	varchar(255)
paidAt	timestampz

5.2 Price tábla (Árak)

A különböző felhasználói csoportok és kedvezményes kategóriák árainak időbeli nyomon követésére szolgál.

Figyelembe kell venni, hogy a diákok és a felnőttek két külön árkategóriában rendelhetnek. Ez a **userType** mezőből derül ki. Emellett ezeknek az áraknak megvan a kedvezményes verziójuk is (**priceCategory**). Ennek következtében legalább 4 különböző ár van kiszabva az étkezés összegére (**amount**).

Nem szabad elfelejteni azt sem, hogy ezek az árak mettől (**validFrom**) meddig (**validTo**) érvényesek. Az utóbbi lehet definiálatlan érték, hisz nem minden esetben lehet előre tudni.

price	
id	serial
userType	user_type E
priceCategory	price_category E
amount	decimal
validFrom	date NN
validTo	date
createdAt	timestampz

6. Egyéb táblák

6.1 Class tábla (Osztályok)

Oktatási csoportok, osztályok (9.A, 10.B), letárolására használt tábla.

class	
id 🔗	serial
className	varchar NN

6.2 Group tábla (Csoportok)

Szakkör, speciális csoportok számontartására szolgáló tábla.

group	
id 🔗	serial
groupName	varchar NN

6.3 Schedule tábla (Étkezés ütemezése)

A zökkenőmentes ebédátvételhez el kell tárolni ebben a táblában, hogy melyik osztály és/vagy csoport mettől meddig tud megebédelni.

Minden felhasználó tartozik egy osztályhoz. A tanároknak, dolgozóknak és külsősöknek ez az érték null lesz.

Előfordulhat azonban, hogy egy adott csoport tagjainak kell megszabni, hogy mikor tudnak menni étkezni, ezért a csoportokat és tagjaikat, akik étkeznek külön le kell tárolni.

schedule	
id 🔗	serial
class_id 🔗	int NN
group_id 🔗	int
from	date
to	date

6.4 County és City táblák (Vármegyék és városok)

Címkezelés és regionális statisztikák készítése a feladatuk.

Az lakcím megadásához biztosítunk egy város táblát benne Magyarország összes város nevével (**name**) és irányítószámával (**zipCode**), plusz egy „0000”-ás adattal, ami arra fog utalni, hogy a felhasználó külföldi címet adott meg. Ezek a városok a különböző vármegyékhez lesz hozzákapcsolva, így lecsökkentve az esetleges elírásokat a felhasználó részéről.

county	
id 🔗	serial
name	varchar(255)

city	
id 🔗	serial
name	varchar(255)
zipCode	int
county_id 🔗	int NN

6.5 RfidCard tábla (RFID Kártyák)

A fizikai RFID kártyák nyilvántartása és használati statisztikáinak gyűjtése.

A **cardNumber** mutatja meg a kártyának a pontos azonosító számát.

Ahhoz, hogy követni lehessen, hogy melyik felhasználó étkezett aznap, fontos letárolni mikor használták utoljára. Ezt a **lastUsedAt** rész teszi lehetővé.

A kártya használatához aktívnak kell lennie, amit az admin be tud állítani, vagy inaktívvá kapcsolni (**isActive**).

rfidCard	
id 🔗	serial
cardNumber	int NN
lastUsedAt	timestampz
isActive	boolean

6.6 Rating tábla (Értékelések)

A konyha minőségének folyamatos monitorozása és visszajelzés gyűjtésére alkalmas tábla.

A felhasználóknak van lehetőségük a megrendelt ételekről véleményt írni (**comment**) és pontozni (**rating**), ezáltal jobban a felhasználók igényei szerint tudja a konyha elkészíteni az ételt.

rating	
id 🔗	serial
user_id 🔗	int NN
order_id 🔗	int NN
rating	smallint
comment	text