***Köszönetnyilvánítás***

*Ezúton szeretnénk megköszönni minden tanárunknak és minden olyan személynek, akik közvetlen, vagy közvetett módon is hozzájárultak ahhoz, hogy ez a vizsgaremek szakdolgozat létrejöhessen. Külön köszönet Kerényi Róbert Nándor, Kasza László Róbert, Németh Bence, Farkas Zoltán, Lázár Zsolt, Sándor Péter, Csontos Dénes tanár uraknak, valamint Sándorné Feke Réka és Domján Annamária tanárnőknek, akik egyengették, figyelemmel kísérték utunkat, javaslataikkal és segítő szándékú kritikáikkal előmozdították, hogy teljes értékű szoftverfejlesztő,- és tesztelőkké váljunk.*

***Miért ezt a programot választottuk?***

Webalkalmazásunk megalkotásának elsődleges célja az volt, hogy a különböző szolgáltató,- és vendéglátóipari vállalkozások számára egy igényes felhasználói felülettel, hatékony adatbázissal és stabil backend szerverrel rendelkező olyan programot adjunk, mely lehetővé teszi a felhasználók részére:

- késztermékek (jelen esetben egyelőre csak pizzák) interneten történő megrendelését

- ételek, italok tulajdonságainak adatbázisból való lekérdezését

- a weboldalon való regisztrációt

- új ételek felvételét, módosítását és törlését

- új felhasználó regisztrálását és bejelentkezését.

***A webalkalmazás adatbázisáról***

A mindennapi életben szinte észre se vesszük, de folyamatosan találkozhatunk különböző adatbázisokkal. Már akkor is, ha egy fárasztó nap után kikapcsolódásképpen zenét hallgatunk, filmet nézünk, mivel a különböző alkalmazásokon keresztül érkező médiatartalmak lejátszólistái is adatbázisból származnak. Tovább gondolva, ha a könyvtár számítógépén utánanézünk egy-egy könyvnek, hogy éppen hány példány érhető el abban a pillanatban az adott könyvtárban. Vagy, ha raktárban dolgozunk egy multinacionális, vagy éppen kisebb ipari cégnél, a raktárnyilvántartás nagyon fontos részét az adatbázis képezi, ahol folyamatosan figyelemmel lehet kísérni alapanyagok, félkész, -és késztermékek mozgását, felhasználását. Illetve, hogy a jelen esethez is igazodjunk, vendéglátóipari egységek, éttermek ételrendelő alkalmazásainak adatbázisai.

***Alapfogalmak***

**Adatbázis**: olyan rendszer, amely lehetővé teszi különböző adatok hatékony tárolását, rendezését. Az adatok ebben lehetnek szöveges adatok, kép vagy videófájlok, valamint hangfájlok is, de sok másfajta adat is. Az adatok strukturált formában kerülnek tárolásra benne, adatmezők, oszlopok és rekordok alkotják. Segítségükkel nemcsak tárolhatjuk adatainkat, hanem biztosíthatjuk azok védelmét.

**Adatbázis-kezelő rendszer**: olyan szoftver, amely lehetővé teszi adatbázisok létrehozását, frissítését, az adatok módosítását és törlését. (DBMS - Database Management System). Több fajtája létezik:

- RDBMS (Relational Database Management System), relációs adatbázis kezelő rendszer

- OODBMS (Object Oriented Database Management System) objektum orientált adatbázis kezelő rendszer

**Adatmodell**: olyan absztrakt megjelenítése az adatoknak, amelyek az adatok össze-

függéseit, kapcsolatait, viselkedését és jellemzőit írja le, diagramok, táblázatok vagy más grafikus formában. Ezekkel lehet megtervezni az adatbázisokat így téve hatékonyan használhatóvá őket. Több típusa is létezik ezeknek az adatmodelleknek, a legismertebb és legtöbbet használt az egyed-kapcsolat, vagy entitás-kapcsolat modell (ER modell ð Entity-Relationship modell).

**Kapcsolat típusok:**

- egy-egy kapcsolat ð az egyik entitáshoz pontosan csak egy másik tartozhat és fordítva

- egy-több kapcsolat ð egy entitáshoz több más entitás is tartozhat

- több-több kapcsolat ð több entitás több entitáshoz is kapcsolódhat.

**Entitás**: az adatbázis kezelő rendszerek az entitások révén kezelik és tárolják az adatokat. Lehet például ügyfél, megrendelés, termék, szállító, dolgozó stb.

**Normálformák:** a relációs adatbázisok tervezésének szabályai, melyek segíte-

nek az adatok redundanciájának csökkentésében. Öt normálforma létezik, melyek:

* első normálforma (1NF) minden adat egyértelműen definiált és oszlo-

pokba van rendezve és ezek nem tartalmaz-

nak ismétlődő adatokat

* második normálforma (2NF) ugyanaz, mint az előbb, és minden nem

elsődleges kulcsú oszlop teljesen függ a

tábla elsődleges kulcsától

* harmadik normálforma (3NF) a második normálforma tulajdonságait

hordozza és minden nem elsődleges kul-

csú oszlop nem függ más nem elsődle-

ges kulcsú oszloptól

* Boyce-Codd normálforma (BCNF) itt a harmadik normálformának fe-

lelnek meg az adatbázis táblái és

minden funkcionális függőség az

oszlopok között a tábla elsődleges

kulcsára vonatkozik

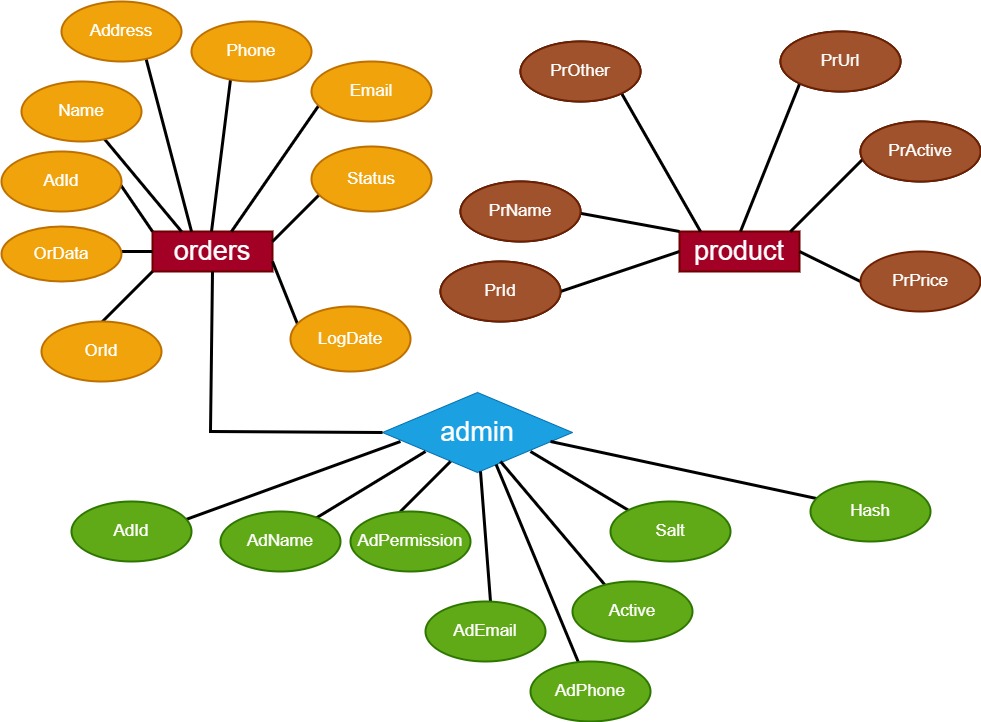
* negyedik normálforma (4NF) itt a Boyce-Codd normálformának felel-

nek meg az adatbázis táblái és nem tar-

talmaznak többértékű függőséget

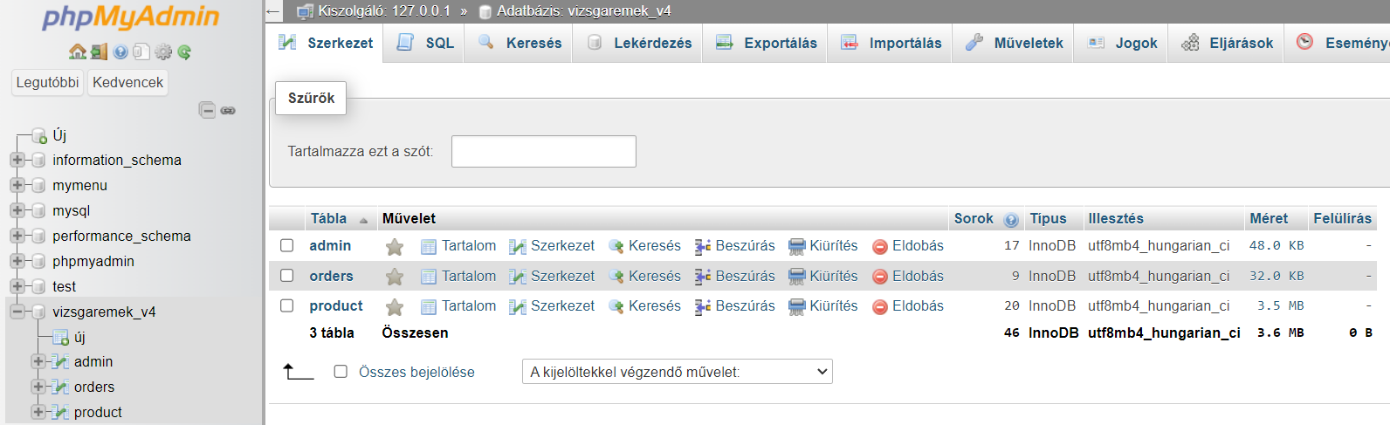
**Funkcionális függőség**: azt jelenti, hogy az egyik adat értéke meghatározható egy másik adat értékéből.

Az adatbázis megtervezésénél fontos lépés egy ER-diagram létrehozása. Az ER-diagram a különböző entitásokat és azoknak a kapcsolatait jeleníti meg. Használatuk számos előnnyel jár az adatbázis tervezési folyamatában. Általuk megérthetjük az üzleti folyamatokat, javítják az adatbázis integritását, lehetővé teszi az adatbázis tervező számára, hogy biztosítsa az adatok helyes tárolását és általuk könnyebben karbantartható és frissíthető az adatbázis.



1. ábra: a vizsgaremek\_v4 pizza adatbázis ER-diagramja

A pizza adatbázisom létrehozását is ez alapján terveztem meg. Az adatbázist és annak tábláit, adatait phpMyadmin felületen MySQL rendszerben valósítottam meg. Az adatbázis-kezelő rendszer SQL (Structured Query Language ðStrukturált Lekérdező Nyelv) nyelven készült.

Az adatbázis három - admin, orders és product - táblából áll. A következő oldalakon szemléltetem és írom le az egyes táblák tulajdonságait, valamint azt, hogy milyen adatokat tartalmaz.

2. ábra: Az ER-diagram alapján a létrehozott vizsgaremek\_v4 pizza adatbázis és táblái képernyőképpel

**1. tábla - admin**



3. ábra: az admin tábla képernyőképe

**Az admin tábla szerkezete:**

- **AdId** ðaz adminisztrátor azonosító száma, amely az elsődleges kulcs (Primary key) a táblában, Auto Increment értékkel, mely az azonosító számokat automatikusan generálja, valamint egy integer egész szám

- **AdName**ðaz adminisztrátor neve, mely egy variálható karakter, azaz tartalmazhat számokat és betűket is azaz varchar, valamint az idegen kulcs (Foreign key) a táblában

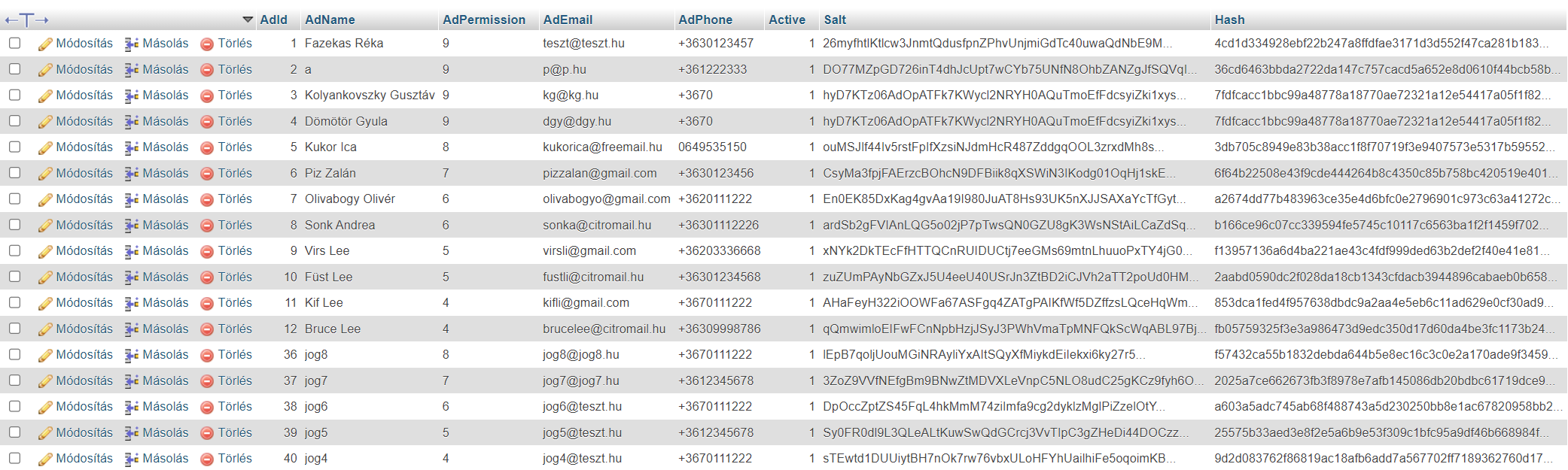
- **AdPermission**ðaz admin felületen a beregisztrált felhasználók jogkör engedélyezési száma. Jelenleg 4-től 9-ig vannak jogosultságok.

- **AdEmai**l ð adminisztrátor, vagy felhasználó e-mail címe

- **AdPhone** ð adminisztrátor, vagy felhasználó telefonszáma

- **Active** ð két aktivitási fokozatot foglal magában: 0 = inaktív, 1 = aktív. Ha az adott dolgozó, vagy felhasználó, illetve pizza valamilyen okból adott pillanatban nem elérhető, akkor 0-ás jelzést kap, azaz inaktív, ha elérhető akkor 1-es státuszú, azaz elérhető, tehát aktív. Csak a megfelelő jogkörrel rendelkező adminisztrátor módosíthatja.

- **Salt** ð ez a salt rendszer egy biztonságos jelszókezelő algoritmusra a bcryptre támaszkodik. Lényege, hogy a felhasználók és adminisztrátorok jelszavaihoz hozzáad egy véletlenszerű karakterláncot, amelyet salt-nak neveznek. A felhasználó által használni kívánt jelszó és ez a salt érték alkotja a hash értéket.

- **Hash** ð a hash-érték különösen fontos a jelszókezelésben. Lényege, hogy nem tárolunk sehol jelszavakat, csak magát ezt a hash-értéket. Minden felhasználói regisztrációkor a jelszó beírásakor a salt-értékkel egy időben generálódik. Ha megtörtént a regisztráció, az adatbázisban csak a salt és a hash-érték kerül tárolásra. Amikor a felhasználó bejelentkezik a saját jelszavával, a backend szerveren és az adatbázisban eltárolásra került hash-értéket keresi ki a rendszer az adott jelszóra és ha minden egyezik, megtörténik a felhasználó beléptetése. Fontos feladata tehát az adatvédelem és az illetéktelen hozzáférés megakadályozása.

4. ábra: az admin tábla adattartalma

**2. tábla - orders (rendelések)**



5. ábra: az orders tábla képernyőképe

**Az orders tábla szerkezete:**

- **OrId** ð a rendelés azonosító száma, amely egy integer egész szám, a tábla elsődleges kulcsa, a rendszer automatikusan generálja, ezért auto increment

- **OrData** ð a megrendelt pizza azonosítószámát és méretét tartalmazza, melyet a product táblából kapunk meg

- **AdId** ð annak az adminisztrátornak, vagy megfelelő jogkörrel felruházott felhasználónak az azonosítószáma, aki a pizzarendelést felvette. Ezt az adatot az admin táblából kapjuk

- **Name, Address, Phone, Email** ð a pizzát megrendelő neve, lakcíme, telefonszáma és email címe, ahová ki kell szállítani a megrendelést

- **Status** ð a megrendelés státuszát tartalmazza. Jelenleg öt ilyen státusz azonosító van, ami egy integer egész szám

 - **LogDate** ð a megrendelés dátuma, DateTime adattípus, év, hónap, nap, óra, perc és másodperc formátumban

6. ábra: az orders (rendelések) tábla adattartalmának képernyőképe

**3. tábla - product (termék)**



7. ábra: a product (termék) tábla képernyőképe

**A product tábla szerkezete:**

- **PrId** ð a termék (pizza) azonosító száma, amely egy integer egész szám, a tábla elsődleges kulcsa, auto increment tulajdonsággal. Új pizza felvételénél automatikusan generálódik

- **PrName** ð a termék (pizza) elnevezése, varchar adattípus

- **PrOther** ð a termék (pizza) alkotóelemei, azaz a feltétek nevei, varchar adattípus

- **PrUrl** ð a termékek (pizzák) képei vannak itt tárolva mediumblob adattípussal. A képek 400 pixel x 400 pixel nagyságúakra vannak átméretezve, hogy ne foglaljanak nagy helyet. Továbbá át vannak kódolva base64-es bináris formátumra és frontend oldalon lettek beillesztve és eltárolva az adatbázisban. A képek konvertálásához egy online felületen elérhető Base64 Image nevű programot használtam, amely megtekinthető a <https://www.base64-image.de/> webcímen

- **PrActive** ð kétfajta aktivitási státusz van. A 0-ás és az 1-es. Ha rendelhető a pizza, akkor 1-es ha nem, akkor 0-ás aktivitási státuszban van.

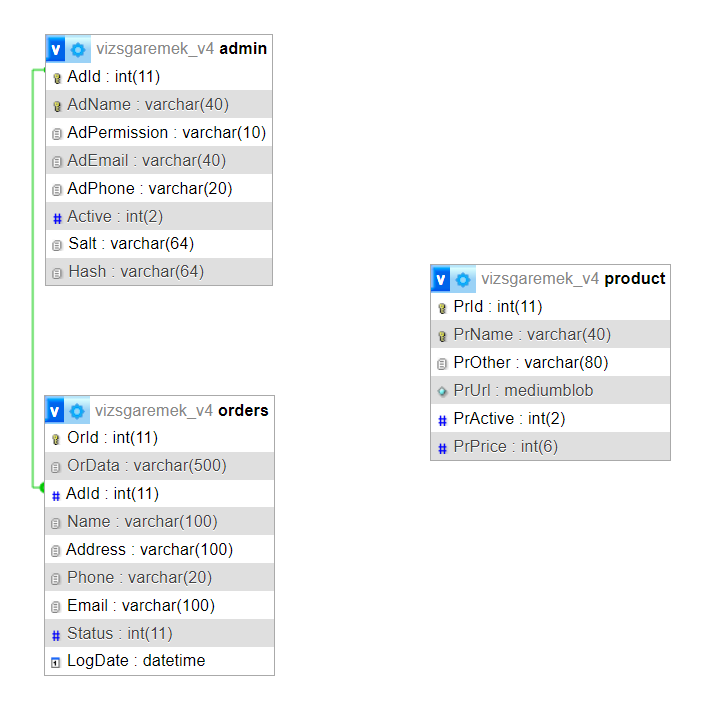
- **PrPrice** ð a pizza ára, integer egész szám adattípussal



8. ábra: a product (termékek) tábla adattartalma képernyőképpel

**A vizsgaremek\_v4 nevű pizza adatbázis táblái közötti kapcsolat**

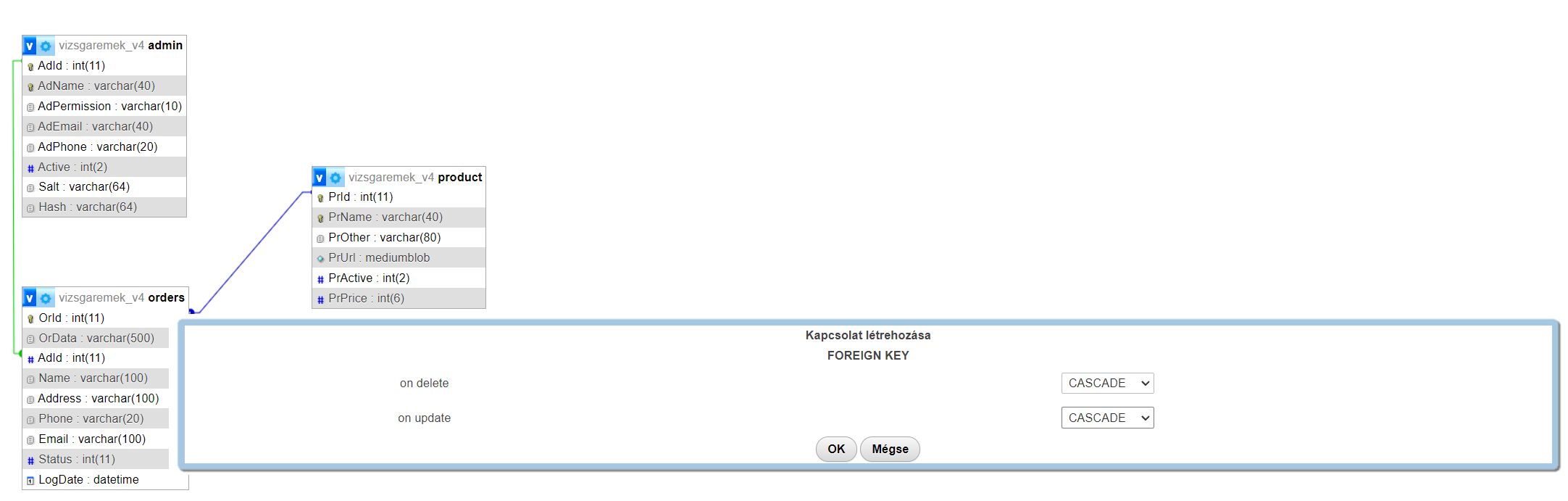
A phpMyadmin Tervező menüpontjában megnézhetjük az adatbázis táblái közötti kapcsolatot. Az ábra jól szemlélteti, hogy az admin tábla az AdId elnevezésű, elsődleges kulccsal ellátott mezője az orders tábla ugyanolyan nevű mezőjéhez kapcsolódik. Erre így azért volt szükség, mert egy megrendeléshez több termék is kapcsolódhat, ezért ez egy egy a többhöz típusú kapcsolat. Ugyanúgy mint az admin és az orders táblában is, egy adminhoz több rendelés is kapcsolódhat.



9. ábra: a vizsgaremek\_v4 elnevezésű adatbázis táblái közötti kapcsolat

**A CASCADE művelet**

A CASCDE művelet azt jelenti, hogy az eszközölt változtatások átterjednek minden olyan táblára, amelyekben az adott rekordot hivatkozó kulcsok vannak. Ha két tábla egymáshoz idegen kulcsok segítségével kapcsolódik és egy rekordot törlünk, beszúrunk, vagy frissítünk, akkor ez a művelet az összes kapcsolódó táblában is végbemegy. Általában ezt a műveletet akkor alkalmazzuk, amikor ezen kapcsolatok fontosak az adatintegritás és a hivatkozások egységessége szempontjából. Viszont, ha ez a CASCADE művelet rosszul van használva, az adatvesztéshez és nem kívánt összeomláshoz vezethet. Így ezért az javasolt, hogy használatát alaposan fontoljuk meg!



10. ábra: a CASCADE művelet képernyőképe

***Felhasznált irodalom***

1. *Szelezsán János: Adatbázisok LSI 1992*
2. *Kupcsikné Fitus Ilona: Adatbázisok példatár LSI 1992*
3. *Luke Welling - Laura Thomson: PHP és MySQL webfejlesztőknek - Hogyan építsünk webáruházat? Perfact kiadó 2010*
4. *Steven Suehring - Janet Valade: PHP, MySQL, JavaScript & HTML5, Panem kiadó Tantusz könyvek 2014*

*5) Richard Blum: PHP, MySQL & JavaScript, Panem kiadó, Tantusz*

*könyvek 2020*