



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai Intézet

SZAKDOLGOZAT

Geoinformatikai webalkalmazás adatbázis fejlesztés

OE-AMK

2023

Hallgató neve:

Kupeczik Melinda

Hallgató törzskönyvi száma: **T000705/FI12904/A-G**



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Geoinformatikai Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Kupeczik Melinda

Szaktervezési szám: [REDACTED]

Törzskönyvi száma: T000705/FI12904/A-G

Neptun kódja: [REDACTED]

Szak: geoinformatikai szakmérnök

Specializáció:

A dolgozat címe: Geoinformatikai webalkalmazás adatbázis fejlesztés

A dolgozat címe angolul: Development of a Geoinformatics web application database

A feladat részletezése:

Geoinformatikai webalkalmazáshoz PostgreSQL adatbázis tervezése és készítése geoinformatikai SQL adatok és lekérdezések implementációjával.

Intézményi konzulens neve: Nagyné Dr. Hajnal Éva

A kiadott téma elévülési határideje: 2025. 07. 10.

Beadási határidő: 2023. 05. 15.

A szaktervezési feladat: Nem titkos.

Kiadva: Székesfehérvár, 2023. 04. 04.



Intézetigazgató

A dolgozatot beadásra alkalmasnak találom: 2023. 05. 15.

[Signature]
belső konzulens

[Signature]
külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai Intézet

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat/diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozat/diplomamunkában található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: Székely 2023.05.12.


.....
hallgató aláírása

Tartalomjegyzék

1.	Bevezetés	1
2.	Adatbázisok, Geoinformatikai adatbázisok	3
2.1	Adat, adatkezelés, adatbázisok, adatbázisok kezelése	3
2.1.1	Adat(bázis)modellek	5
2.1.2	Adatbáziskezelő rendszerek (DBMS).....	6
2.1.3	PostgreSQL	8
2.1.4	Felhasználói jogosultságkezelés SQL-ben.....	9
2.2	Térinformatikai adatkezelés sajátosságai	10
3.	Az adatbázis-tervezés módszertana	14
3.1	Adatbázis-tervezés alapelvei.....	14
3.2	Redundancia és anomáliák	15
3.3	Adatbázis tervezés lépései.....	15
3.3.1	Specifikáció és adatgyűjtés	16
3.3.2	E/K modell	16
3.3.3	Relációs adatbázis séma, normalizálás	19
4.	Az adatbázis megtervezésének lépései	25
4.1	Informatikai háttér.....	25
4.2	Specifikáció, adatgyűjtés.....	27
4.2.1	Metaadatok — adatforrás, adattípus	27
4.2.2	Jogosultságkezelés — Felhasználói csoportok és jogosultságok	29
4.2.3	Adatlehívás— kulcsszavak	31
4.3	E/K-modell kidolgozása	31
4.4	Relációs adatbázis séma és normalizálása	39
5.	A létrehozott adatbázis bemutatása.....	41
5.1	Minta adattáblák és attribútumaik	41
5.2	Lekérdezés minták	49
6.	Értékelés, továbbfejlesztési lehetőségek.....	51
7.	Összefoglaló.....	52
8.	Summary.....	53
9.	Irodalomjegyzék	54
10.	Ábrajegyzék	56
11.	Táblázatok jegyzéke	57

1. BEVEZETÉS

Szakedolgozatom témája egy egyetemi projektet támogató geoinformatikai webalkalmazáshoz háttérrel nyújtó adatbázis megtervezése geoinformatikai SQL adatok és lekérdezések implementációjával, PostgreSQL környezetben.

A nemzetközi — magyar-kínai kétoldalú tudományos együttműködési — egyetemi projekt célja a vidéki környezet modellezésének kialakítása, és ehhez kapcsolódóan geoinformatikai és távérzékelési technológiák cseréje az ökológiai kiértékelésekhez mindkét, a projektben résztvevő országban.

A vidéki térségek fenntartható fejlődéssel kapcsolatos globális problémái kapcsán az egyes országok közös célja az ökokörnyezeti minőségi paraméterek monitorozása, azonban a különböző országok eltérőképpen közelítik meg a problémát. A projekt célul tűzi ki az egyes modellek összehasonlítását a különböző országok és városaik esetében, vizsgálva a magyar és a kínai modellt, nagyfelbontású távérzékelési adatok segítségével, valamint eljárások kidolgozásával az ökokörnyezet automatikus értékelésére, a távérzékelési adatok és a városi környezeti tényezők közötti kapcsolatok bemutatása mellett. Ezzel segítve a vidéki ökokörnyezet minőségének előrejelzését és monitorozását, valamint az űrfelvételek adatait térbeli adatokkal integrálva javítva az ökokörnyezeti monitoring pontosságát. A kutatás részét képezi a modellezés és az adattárolás új módszereinek keresése is.

A távérzékelésből, attribútum adatokból és meglévő térképekből történő adatgyűjtést követően az adatbázis megtervezése és felépítése, majd az adatok online grafikai megjelenítése a projekt webalkalmazásának segítségével — az információ megosztásával és továbbításával — segítheti az ökokörnyezet értékelését és változásainak jobb megértését.

A projekt részeként webalkalmazást üzemeltet az egyetem, ahol egy projekt-honlapon mutatják be az eredményeket, illetve népszerűsítik a programot. Az e mögött álló adatbázis megtervezését céloztam meg szakdolgozatomban.

A projektben felhasználható adatok köre igen sokrétű:

- nagy méretarányú földhasználati térképek;
- talajszerkezeti adatok;
- WorldView, RADARSAT, Landsat, Lidar 3D pontfelhő adatok;
- vonatkozó kínai műholdfelvételek (Sunflower műholdadatok);
- a mezőgazdasági környezet alapadatai (pl. földhasználat, domborzat és további földdel kapcsolatos hivatalos térképek és adatok), földfelszíni vegetációs adatok (pl. vegetációs index);
- növényfenológiai adatok, a mezőgazdasági terület vegetációs adatai;
- mezőgazdasági területek nedvességi viszonyai;
- kutatási eredmények, módszerek leírásai (pdf dokumentumok).

Feladatom egyik kihívása, hogy a forrásadatok sokrétűségéből adódóan az adatbázisnak képesnek kell lennie a sok különböző formátum kezelésére (kép, adattábla, térképi adatok, 3D pontfelhő adatai). Másik kihívás a jogosultságkezelés megfelelő beépítése az adatbázisba. Természetesen az adatok kulcsszavas lekérdezését, feltöltését, a módosítás lehetőségét, valamint hozzáférést a kutatási eredmények és a kapott módszerek leírásaihoz és a publikációkhoz jogosultsági szinthez kapcsolódóan biztosítani kell a felhasználók számára.

2. ADATBÁZISOK, GEOINFORMATIKAI ADATBÁZISOK

A minket körülvevő, egyre gyorsuló világ adatok formájában számos információt hordoz, melyek mindenki által hozzáférhető és folyamatosan aktualizálható grafikus (térképi) megjelenítésére az elmúlt évtizedek technológiai — és elsősorban informatikai — fejlődése miatt napjainkban egyre nagyobb igény merül fel. Ehhez nyújt számunkra segítséget a térinformatikai (geoinformatikai) megjelenítés, melyhez olyan adatbázis háttérre van szükségünk, ami nagy mennyiségű, eltérő forrásokból érkező és eltérő formátumú adatok kezelésére egyaránt képes, és a folyamatos változásokat is megfelelő rugalmassággal kezeli.

Az adatbázisok tervezésének elméleti és gyakorlati hátterének megismerése előtt szükséges az adat fogalmának definiálása, az adatok gyűjtésének, tárolásának, valamint feldolgozhatóságának vizsgálata és ehhez kapcsolódóan az adatbázisok az alaposabb megismerése, valamint a geoinformatika speciális igényeinek feltérképezése is. Ehhez nyújt segítséget szakdolgozatom 2. fejezete.

2.1 Adat, adatkezelés, adatbázisok, adatbázisok kezelése

Megfelelő mennyiségű és minőségi adat, valamint az adatok hordozta információ tartalom birtokában a mindennapi gazdasági, társadalmi és tudományos élet területein egyaránt segíti életünket a döntéshozatal, az oktatás stb. folyamataiban.

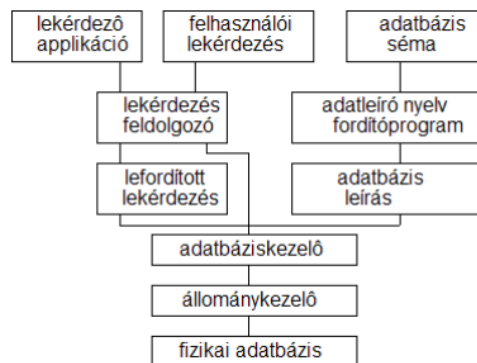
Ebben jutnak nagy szerephez a való világot (illetve egy részét) leíró *adatbázisok*, melyek tulajdonképpen ezen adatok rendezett, tárolt és kezelt gyűjteményei, melyeken további műveletek végezhetőek (ideértve többek között a grafikus megjelenítést is). Ezen adatbázisok szerepe már szinte minden szakterületen elengedhetetlen fontosságú, napi szinten találkozunk magunk is olyan rendszerekkel, melyek mögött, azok üzemeltetését biztosítandó, adatbázis(ok) áll(nak).

Az adatok gyűjtése eleinte célzottan, szervezeten történt, mára azonban ennek módja is megváltozott, bizonyos applikációk vagy eszközök (pl. okosóra, mobiltelefon eszközeinek applikációi) használatával magunk nyújtunk újabbnál újabb információkat az adatbázis-kezelő rendszereket üzemeltető technológiai cégeknek.

Eleinte az adatbázisok papír alapon (vagy még korábban kőbe vésve) születtek, majd a számítástechnika fejlődésével Excel-táblákban, később adatbázis-kezelő rendszerek speciális szoftvereiben valósultak meg. Az első adatbázis-kezelő rendszerek a fájlkezelő rendszerekből alakultak ki az 1960-as évektől, majd a banki és vállalati nyilvántartások egyre nagyobb mennyiségű és sok lekérdezést, valamint módosítást igénylő rendszereiben fejlődtek tovább, ezek voltak a mai adatbázis-kezelő rendszerek elődei. A programozási nyelvek fejlődésével egyre nagyobb igény merült fel a magas szintű lekérdezési nyelveket támogató adatbázis-kezelő rendszerek iránt, így a fejlesztési irányt elsősorban ez az igény határozta meg, s az 1990-es évek elejére így jutottunk el a ma is használatban lévő relációsadatbázis-kezelő rendszerekhez. Az adatbázis-kezelő rendszerek természetesen ma is folyamatosan fejlődnek, ilyen irányvonal a fejlesztésben például az objektum orientált környezet hatására történő fejlesztés. Másik fejlesztési szempont a szoftverrendszerek terjedelmének és bekerülési (fejlesztési és üzemeltetési) költségeinek csökkentése, a felhasználóbarát környezet kialakítása, és ezáltal az egyre szélesebb körben történő hozzáférhetőség biztosítása.

Ebből a tudásbeli és technológiai fejlődésből ered az adatbázisok ereje. A szoftveres megoldás teszi lehetővé a nagy mennyiségű, folyamatosan változó és bővülő adathalmaz hosszú távú tárolását és széleskörű felhasználását. A felhasználhatóság és kezelhetőség körét megállíthatatlanul bővítik a szoftverek fejlesztésével és kezelhetőségük egyszerűsödésével, így elmondható, hogy a jövő fejlődésének zálogát ezek az új szoftverek, és velük együtt a különböző adatbázis-rendszerek hordozzák. [1] [2] [3]

Magát az adatbázis-kezelés hagyományos rendszerét (felhasználói adatgyűjtés visszacsatolása nélkül), annak résztvevőivel és kapcsolataikkal a 2.1 ábra szemlélteti.



2.1. ábra Az adatbázis-kezelés rendszere és résztvevői [3]

2.1.1 Adat(bázis)modellek

A valós objektumoknak (egyedek), azok jellemző tulajdonságainak és kapcsolataiknak absztrakt módon történő leképezésére szolgál az adat(bázis)modell sémája, ami tulajdonképpen az adatbázis szerkezetét leíró logikai séma. Az adat(bázis)modellek sematikus határozzák meg az egyes egyedeket, azok *tulajdonságait* és a köztük lévő *kapcsolatokat* (ezek az adatmodell-elemek), az esetlegesen fennálló korlátozó feltételeket az adatbázis felépítéséhez.

Az adatbázisokat különféle típusú adat(bázis)modell szerint építhetjük fel, az adatbázis tulajdonképpen maga a megvalósított adat(bázis)modell. Az adatbázisok modelljéül szolgáló struktúrák 3 fő iránya terjedt el. A *formalista* modellezésben az adatkapcsolatok leírása diagramokkal, gráfokkal (hierarchikus, illetve hálós adatmodell) történik, míg a *szemantikai* modellezés esetén verbális leírás (szavakkal), a *matematikai* jellegű modellezés pedig matematikai struktúrák segítségével (relációs adatmodell) dolgozik.

Az adatbázis szerkezetének kialakítását meghatározó adat(bázis)modell típusa alapjaiban határozza meg a használható adatbázis kezelő rendszert és nyelvet is.

A *Hálós adat(bázis)modell*t a *Hierarchikus modell* általánosításának tekinthetjük. A *Hierarchikus adat(bázis)modell* rendszerében az adatok hierarchikus elrendezésben kerülnek feltüntetésre a fa struktúrájú gráfban, a fa csomópontjaiban és leveleiben.

A Hierarchikus és Hálós modelleket az 1980-as évek elején felváltó *Relációs adat(bázis)modell* az adatokat táblázatos formában képezi le, a kapcsolatokat — a másik két modellel összehasonlítva — nem előre meghatározottak, maga a kapcsolat nem épül be a modellbe, helyette a kapcsolatok létrehozásának lehetősége adott, az ahhoz szükséges adatokat tároljuk a modellben, így a modell rugalmasabb a másik kettőnél. A modellben a tulajdonságokon van a hangsúly, ezekkel definiáljuk a modellt.

Nagy előnye még továbbá ennek a modellnek, hogy egyszerűen értelmezhető, az adatbáziskezelő rendszerbe a logikai adatmodell relációit módosítás nélkül tudjuk transzferálni, mindemellett ebben a modellben a normál formák segítségével egzakt módon elvégezhető adatbázis tervezése.

Magát az egyedet ebben a modellben egy táblázatos formában definiáljuk, a táblázat adja az egyedhalmazt magát, melyben a tábla (*reláció*) soraiban (*rekord*) a logikailag összetartozó adattulajdonságok (a sorok az egyed értékei), oszlopaiban (attribútum) az azonos minőségű adattulajdonságok jelennek meg, míg mezői (egy sor és oszlop metszete) tartalmazzák magukat az adatokat. Az oszlopok (attribútumok) száma a *fokszám*, míg a sorok (rekordok) száma a *kardinalitás*.

Így a *Relációs adat(bázis)modell*ben a létrehozott adatbázis több táblázatból áll (mind egy-egy egyedhalmaz), de a közöttük fennálló kapcsolatokat a modell felírásakor nem definiáljuk, azok létrehozásához szükséges adatok a modellben megtalálhatóak (például két tábla közös oszloppal rendelkezik, ami magát a kapcsolatot adja). [4] [5]

A *Relációs adat(bázis)modell* bővítésével jelent meg az *Objektum-relációs adat(bázis)modell (ORDBMS)*, mely a relációs adatbázis táblákra alkalmazza az objektum orientált megközelítésben használt objektum, osztály, öröklődés fogalmakat. Leíró/lekérdező nyelve az SQL3 szabvány. [6]

Objektumorientált adat(bázis)modellek az objektumorientált programozás térhódításával terjedtek el. Speciális objektumorientált lekérdezési nyelveket használnak (ODL, OQL).

2.1.2 Adatbáziskezelő rendszerek (DBMS)

Az adat(bázis)modell megvalósításánál az adatbáziskezelő rendszerek két fő feladata az adatdefiníció és a lekérdezés, mindkét feladatra rendelkezésre állnak lekérdező és programozási nyelvek, melyek közül a konkrét feladat ismeretében választhatunk. Maga az adatbáziskezelő rendszer magában foglalja mind az adatszerkezetet leíró nyelvet (DDL), a fizikai szerkezetet megvalósító nyelvet (SDL), valamint az adatok visszakeresését végző lekérdező nyelvet (DML) is.

Emellett a DBMS szoftver biztosítja az adatok módosításának lehetőségét is a felhasználók számára, a jogosultsági szinteknek megfelelően.

Mindhárom, a 2.1.1 fejezetben bemutatott adat(bázis)modellre rendelkezésünkre állnak szabványos adatkezelő rendszerek és nyelvek, a teljesség igénye nélkül az 1. táblázat néhány példát mutat be arra, mely rendszereket használhatjuk az egyes modellekhez.

1. táblázat — Az egyes adat(bázis)modellekre épülő adatkezelő rendszerek [7] [8] [9]

Adat(bázis)modell	Adatbáziskezelő rendszerek (DBMS)	
Hálós	IDMS, IDS, TurboIMAGE, Raima Database Manager, UNIVAC DMS-1100	
Hierarchikus	IBM IMS	
Relációs	Kereskedelmi:	Oracle, MS SQL, IBM DB2, SAP Sybase ASE, Informix
	Nyílt forráskódú:	MySQL, PostgreSQL, SQLite, Sequel Pro, SQL Server Express, Firebird

Az adatbáziskezelő rendszerekkel szemben támasztott főbb elvárások a következők:

- nagymennyiségű adat gyors, biztonságos és hatékony kezelését és hosszú időn keresztüli tárolását segítse;
- az adatokhoz való hatékony hozzáférés egyidejűleg több felhasználó számára biztosított legyen (hibák, ismétlődések elkerülésével);
- adatstruktúrától, hardverkonfigurációtól és adatlekérés módjától való független-séget biztosítsa;
- biztosított legyen a tartósság, vagyis hibák és meghibásodások, valamint szándékos rongálás esetén helyreállítható legyen az adatbázis;
- új adatbázis létrehozása, meglévő szerkezeti módosítása megoldható legyen a megadott adatdefiníciós nyelvvel (DDL — Data Definition Language);
- az adatbázisban lévő adatok lekérdezése mellett azok módosítása adott lekérdező nyelvvel biztosított legyen (DML — Data Manipulation Language). [2] [10]

A mára egyeduralkodóvá vált *Relációs adat(bázis)modell* adatbáziskezelő rendszereinek (RDBMS) a többi modell rendszereihez képest számos előnyös tulajdonsága van. Ezek (a teljesség igénye nélkül) a következők:

- biztonságos, egyszerű használat
- független logikai adatbázis
- tárolt adatok aktualizálhatók/frissíthetőek

- egyidejűleg több felhasználó számára biztosítanak hozzáférést az adatbázishoz
- biztonsági mentést és helyreállítási eljárásokat biztosítanak
- számos interfészt biztosítanak
- korlátozzák a redundanciát és adatismétlődést
- adatintegritást és jobb fizikai adatfüggetlenség kínálnak

Mindezek mellett azonban hátrányos tulajdonságai is vannak az RDBMS rendszereknek:

- az elveszett adatok helyreállítása bonyolult
- egyes alkalmazások feldolgozási sebessége lassú
- működtetése képzett munkaerőt igényel
- drága, összetett szoftverek, melyek az összköltséget növelik [9]

2.1.3 PostgreSQL

A geoinformatikai adatbáziskezelő rendszerek tekintetében mára már széles piaci kínálat áll rendelkezésünkre, többféle rendszer közül választhatunk az adatbázis megvalósításához. Ezek közül a legelterjedtebb **geoinformatikai adatbázisokat támogató adatbáziskezelő rendszerek** közé tartozik többek között az **SQLite**, a **Microsoft SQL Server**, az **Oracle Spatial Database**, az **IBM Db2**, az **SAP HANA** és a **PostgreSQL**.

Jelen szakdolgozat kitűzött feladatában egy, a **PostgreSQL** segítségével létrehozandó adatbázis megtervezésével foglalkozom, ezért a fent felsorolt adatbáziskezelő rendszerek közül ezt emelem ki és mutatom be részletesen.

Az SQL nyelvet használó **PostgreSQL** az egyik legrégebbi és legfejlettebb nyílt forráskódú, ingyenes és széleskörűen bővíthető objektum-relációs adatbáziskezelő rendszer, mely minden nagyobb operációs rendszerrel (Windows, Linux, Macintosh, BSD, Solaris) kompatibilis és a **PostGIS** térinformatikai adatbázis-bővítő segítségével egyszerűen és hatékonyan használható térinformatikai adatbázisok fejlesztésére. Rendkívüli népszerűségét — mind a fejlesztők, mind pedig a rendszergazdák körében — elsősorban letisztultsága, rugalmassága és bővíthetősége adja.

Robosztus, biztonságos és bővíthető — emellett az alábbi tulajdonságok is segítik elterjedését:

- az SQL szabvány nagy részét támogatja,
- online biztonsági mentések,
- rendkívül hibatűrő, az ACID — atomitási, konzisztenciális, elkülönítési, tartósági — tulajdonságokkal garantálja az adatbázis integritását,
- szöveg, kép, hang és video formátumokat is támogatja, ezáltal széleskörű a felhasználhatósága,
- legnépszerűbb és funkciókban gazdag adminisztrációs és fejlesztői platformja, a pgAdmin, geoinformatikai adatbázisok fejlesztéséhez is jól használható,
- számos indexálási technikát kínál,
- kompatibilis a legtöbb programozási nyelvvel és protokollal (pl. C, Go, Perl, Python, Java, .Net, Ruby, C++, ODBC, Tcl, C#),
- elkötelezett közreműködő közössége kiterjedt támogatási rendszert biztosít felhasználóinak,
- a PostGIS-en kívül számos egyéb bővítő modul is a felhasználók rendelkezésére áll (pl. Citus, pg_cron, HyperLogLog, t-digest). [11]

A PostgreSQL-ben használhatóak a leggyakoribb adattípusok, számos file formátumot támogat (2. táblázat).

2. táblázat — Leggyakrabban alkalmazott változótípusok a térinformatikában [12]

Import	TXT, XLS, XLSX, GS, MDB, XML, CSV, JSON, ODBC, DBF
Export	TXT, XLS, XLSX, GS, MDB, XML, CSV, JSON, ODBC, DBF, HTML, SQL, RTF, PDF

2.1.4 Felhasználói jogosultságkezelés SQL-ben

A felhasználói jogosultságok kezelése SQL-ben az adatbáziselemekhez való hozzáférést meghatározó jogosultságok kezelésekor ún. *engedélyazonosítók*at definiálunk, melyek maguk a felhasználói nevek. A különféle jogkörök ezekhez az engedélyazonosítókhoz vannak hozzárendelve.

Adott SQL utasítást tartalmazó modul kizárólag az adott jogosultság mellett hajtható végre. Minden SQL elemnek (modul, séma, munkafázis) van egy tulajdonosa: mind-egyikhez hozzá van rendelve egy-egy engedélyazonosító. Adott elem engedélyazonosítója minden jogosultsággal rendelkezik az adott elem felett. Engedélyazonosítók egyezésekor a jogosultságok rendelkezésre állnak.

Minden SQL rendszernek van egy engedélyezési diagramja az összes jogosultságának nyilvántartására. Az *engedélyezési diagramok* a jogosultságok áttekinthetőségének segítésére szolgáló speciális gráfok, melyek csomópontjai egy felhasználót és egy jogosultság megfeleltetései, az átruházásokat nyilakkal jelöli a gráf. [2]

2.2 Térinformatikai adatkezelés sajátosságai

A geoinformatikai adatokat sokféleségük — változatos adattípus, terjedelem, eltérő gyűjtési időpont, illetve avulási idő stb. — és eltérő forrásuk teszi adatkezelési szempontból érdekessé és értékessé számunkra. A nagy mennyiségű adat tárolása adattárházak használatával hatékonyan és rugalmasan oldhatók meg.

A térinformatikai adatok két fő típusa a **vektoros** (2D/3D alapú geometriai formák, mint például vonal, pont, poligon), valamint a **raszteres adatok** (képpont alapú rácsok — távérzékelési felvételek, magassági adatok).

Az adatokhoz tartozó vetületi információkat, a vetületi rendszerek felépítését és kapcsolatait, a vektoros geometriák kezeléséhez kapcsolódó egységesített rendszert az OGC (Open Geospatial Consortium) szabványai tartalmazzák:

- OGC Spatial Referencing by Coordinates (SRC) — szabvány a vetületi rendszerek felépítéséről és kapcsolataikról
- OGC Simple Feature Access (SFA) — szabvány a vektoros geometriák kezelésére

(<https://www.ogc.org/standards/>) [13]

A geoinformatikai adatbázis megtervezéséhez és üzemeltetéséhez elengedhetetlen, hogy a begyűjtött, nagy számú adat egyértelműen definiált legyen, ezért az adatok egységesítése és annak meghatározása, hogy az adatokat milyen változótípusban tároljuk kiemelt szerepet kap a tervezési folyamatban.

Az adatbázis átláthatóságát és a tároláshoz, illetve a felhasználáshoz szükséges erőforrás-igény optimalizálásához elengedhetetlen a geoinformatikában használt **változótípusok** és adatformátumok helyes definiálása, valamint következetes használata.

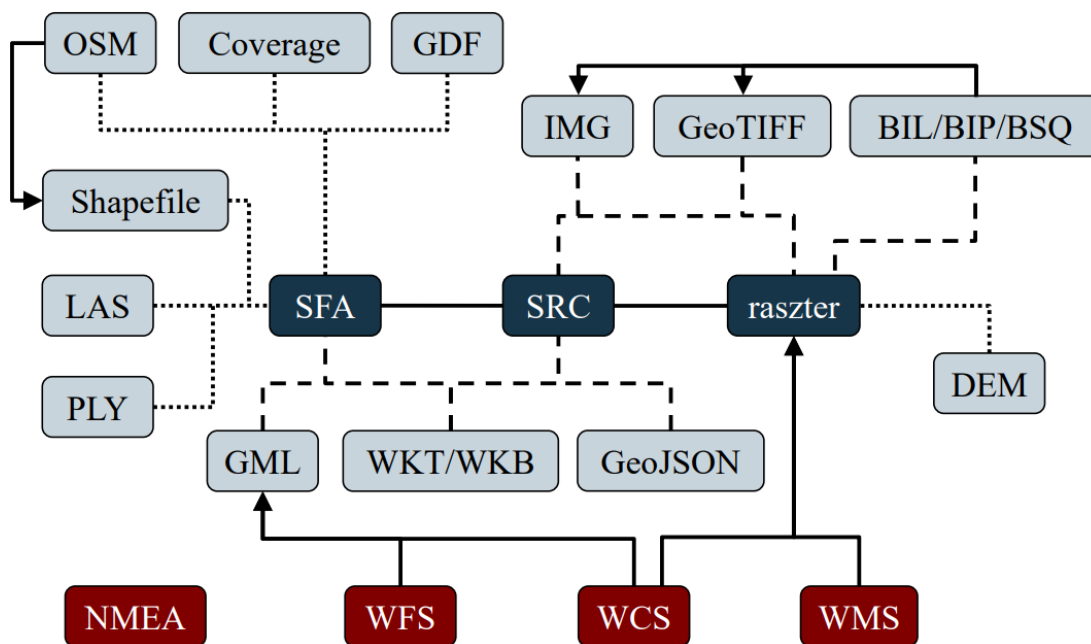
Az adatbázisok lekérdezésének alapját képező relációs algebrai műveleteket — ezek az adattáblákon végzett matematikai műveletek — a lekérdező nyelvek (pl. SQL) implementálják ugyan, és képesek elvégezni az int vagy float típusokra jellemző matematikai műveleteket a string típusú változóban tárolt számokkal is akár, mégis célszerű kerülni a string típusú változó használatát. [4]

A geoinformatikában leggyakrabban alkalmazott változótípusok összefoglaló táblázatát a 3. táblázat tartalmazza.

3. táblázat — Leggyakrabban alkalmazott változótípusok a térinformatikában [4]

Változótípus	Leírás	Specifikáció
string	szöveges változó kvalitatív tulajdonságok leírására	változó neve karakter sorozat maximális hossza
integer (int)	változó egész számok tárolására	változó neve számjegyei maximális hossza
float	lebegőpontos változó nem egész számok tárolására (tizedes törtök); 32 biten ábrázoljuk	változó neve méret pontosság
double	dupla pontosságú lebegőpontos változó; 64 biten ábrázoljuk	változó neve méret pontosság
boolean	logikai értéket hordozó változó	két értéket vehet fel: TRUE vagy FALSE (IGEN/NEM, 1/0, IGAZ/HAMIS, YES/NO)
date	változó dátum tárolására	év, hónap és nap (megadott séma szerint)

A geoinformatikában leggyakrabban használt adatformátumok összefoglaló táblázatát és kapcsolataik sematikus ábrázolását a 2.2 ábra mutatja be.



2.2. ábra Leggyakrabban használt adatformátumok a térinformatikában [13]

A **vektoros** adatformátumok közül a WKT/WKB, a GML és a GeoJSON általánosan használható, szabványos adatformátumok, míg a Shapefile szintén általános célú, azonban szabványnak nem feleltethető meg. A topologikus adatkezelés GDF, OSM és Coverage formátumai, a 3D szkennelt adatkezelés PLY és LAS formátumai, valamint a GPS adatkezelés NMEA formátuma speciális célt szolgálnak.

A vektoros felvétel készítő eszközök (LiDAR, GPS) terjedésével a vektoros adatformátumok szerepe megnőtt. A 3D pontfelhők és a nagy felbontású poligonok esetében nagymennyiségű adat tárolásáról és kezeléséről kell gondoskodnunk, még annak ellenére is, hogy a vektoros tartalmakat helytakarékosnak tekintjük. Ezen adatok hatékony feldolgozását segítik elő az általánosított geometriák használata, egyszerűsítő eljárások alkalmazása (pl. Douglas-Peucker algoritmus), vagy a befoglaló téglalapok/téglatestek használata.

A **WKT/WKB** az SFA szabvány által specifikált, nagyfokú támogatottsággal rendelkező, egymással analóg ábrázolást biztosító adatformátumok, melyek felhasználása széleskörű (PostGIS, MSSQL, DB2 is használja), akár vetületi információk leírására is használhatóak.

Az ESRI által kifejlesztett **Shapefile** adatformátuma file-ok halmaza (.shp, .shx és .dbf a minimálisan szükséges file-ok a felépítéséhez, de ezek mellett további file-okból is állhat), és ma a legnépszerűbb és legszélesebb körben használt adatformátum a geoinformatikai adattárolásban, azonban topológiai információk tárolására nem, csupán geometriai leírásra alkalmas (X és Y koordinátákkal).

Az OGC által támogatott **GML** egy XML alapú adatformátum, elsősorban internetes vektoros adatokhoz, mely vektoros, leíró és topológiai információkat, továbbá raszteres képeket és szenzoradatokat is tárolhatnak.

A JSON formátum kifejezetten térbeli alkalmazásra fejlesztett változata a **GeoJSON** (valamint ennek továbbfejlesztett, az adatokat topológia alapján tároló változata, a **TopoJSON**) adatformátum, mely szintén elsősorban internetes felhasználásra szolgál.

A legnépszerűbb **raszteres** adatformátumok közé tartoznak a képi tartalommal bővített **GeoTIFF**, a **BIP/BIL/BSQ** nyers bináris formátuma, valamint az **IMG** adatformátum.

[13]

3. AZ ADATBÁZIS-TERVEZÉS MÓDSZERTANA

A jó adatbázis tervezéséhez egyaránt szükséges az alapos elméleti és technikai tudás, valamint egy megfelelő matematikai-logikai szemlélet. A tervezés alapelveiről, annak elméleti és technikai háttéréről, valamint a szükséges matematikai-logikai elvekről szól ez, az adatbázis-tervezés módszertanát bemutató fejezet.

3.1 Adatbázis-tervezés alapelvei

Az adatbázis tervezésnél a *valósághű modellezés* (megfelelés a specifikációnak) mellett az *egyszerűsége* (csak a feltétlenül szükséges elemet tartalmazza a modell), a *redundancia és anomáliák elkerülésére*, valamint a *megfelelő kapcsolatok és elemtípusok* megválasztására is figyelmet kell fordítanunk. [2]

Amikor *valósághű modellezésről* beszélünk, azt várjuk el az adatbázistervtől, hogy az alkalmazás specifikációjának teljeskörűen megfeleljen, a létrejött kapcsolatok (bármilyenek is) legyenek értelmesek.

A *redundancia elkerülésének* követelménye annak az elvárásnak a megfogalmazása, hogy az adatbázisban minden kizárólag egyszer szerepeljen. Ezzel összefüggésben fontos, hogy az adatbázisban *megfelelő kapcsolatok* (és csakis ezek) szerepeljenek, elkerülendő a felesleges kapcsolatok szerepeltetése, ami szintén redundanciához vezethet. Ugyan egy attribútum implementálása egyszerűbb, mint egy egyedhalmazé vagy kapcsolaté, azonban nem választhatunk mindent attribútumnak az elemek típusának megválasztásakor, mivel ez további problémák forrása lehet. A *megfelelő elemtípusok* megválasztásához segítségünkre lehet néhány egyszerű szabály arra vonatkozóan, mikor válasszuk az attribútum típust és mikor az egyedtípust.

Az *egyszerűség* kívánalma annak megfogalmazása, hogy az adatbázisba a szükségesnél több elem ne kerüljön bele, elkerülve a bonyolult és a hibák lehetőségét növelő, több tárhelyet igénylő adatbázis létrejöttét. [2]

3.2 Redundancia és anomáliák

A *redundanciák* (valamely tény vagy más adatokból levezethető mennyiség ismételt, többszörös tárolása) és *anomáliák* (pl. tárolási, bővítési, módosítási, törlési — amikor adott függőségek jelenléte okoz problémát az adatbázisban) elkerülése az adatbázis tervezés tipikus problémái, melyek a szükségtelen tároló terület lefoglalása mellett az adatbázis inkonzisztenciáját (ellentmondások előfordulása) is okozhatják. A megfelelő kapcsolatok és elemtípusok megválasztásával, normalizálással, valamint relációk felbontásával kiküszöbölhetjük előfordulásukat. [2]

3.3 Adatbázis tervezés lépései

Az üzemeltethető adatbázisok létrejötte egy többlépcsős folyamat eredménye. Alkalmazáshoz kapcsolódó adatbázis tervezését igen sokféle módon és különböző számosságú lépésből álló tervezési rendszerben végezhetjük, ezek közül a szakdolgozat feladatmegoldását az egyik legrégebbi tervezési módszerrel, az alapinformációk és elgondolások (specifikáció) összegyűjtését követően az E/K diagram megtervezésével (magas szintű terv), majd ezután ennek relációs adatbázissémává történő átalakításával és végül annak normalizálásával oldottam meg, ezért jelen szakdolgozatban ennek a tervezési módszernek az egyes lépéseit mutatom be.

A tervezés egy másik módja lehetett volna például az adatok összeszerkesztése egy táblába, és a tábla megfelelő szabályok szerinti szétbontásával megalkotjuk a relációs adatbázist. Jelen feladat komplexitásából adódóan ez a megoldási mód nem hatékony, nem kivitelezhető.

Az adatbázis modellezés és implementálás eljárásának egyszerűsített sémáját a 3.1 ábra mutatja be. [2]



3.1. ábra Az adatbázis modellezés és implementálás eljárásának lépései [2]

Az adatbázis tervezésénél az adatbázis céljának és felhasználásának meghatározását (specifikáció) a rendelkezésre álló információk összegyűjtésével, a korlátozó feltételek meghatározása mellett és a lehetséges variációk feltérképezésével végezzük el, majd ezt követően rendszerezzük a rendelkezésünkre álló információkat, meghatározzuk az azok közti kapcsolatokat, melyeket E/K (Egyed-Kapcsolat) diagram formájában ábrázolunk és meghatározzuk a kulcsokat, végül azt transzformáljuk egy relációs adatbázis tervbe.

Ezt követően a függőségek meghatározása, majd az adatbázis terv optimalizálása következik, amikor a normalizálás és dekomponálás segítségével tökéletesítjük azt az anomáliák megszüntetése és a redundancia kiküszöbölése érdekében. Az így létrehozott táblák halmaza alkotja az adatbázis tervet, melynek megvalósítása (a file-ok létrehozásával, valamint az indexelés és a nézetek definiálásával, a lekérdezések tervezésével) az adatbáziskezelő rendszer programjának (pl. SQL) segítségével zárja a folyamatot. [1]

3.3.1 Specifikáció és adatgyűjtés

Az adatbázis tervezés folyamata a rendelkezésre álló információ begyűjtésével (pl. milyen információkat kell tárolni?), azok rendszerezésével (pl. mely információelemek kapcsolódnak egymással?), a jövőbeni adatbázis funkciójának meghatározásával, valamint az ehhez kapcsolódó lehetőségek és esetleges fizikai korlátozó tényezők azonosításával (pl. szoftverek, tárhely-korlátok) veszi kezdetét.

Annak meghatározása, hogy milyen adatokkal (forrás, típus ismerete) dolgozunk, illetve milyen szoftverekre építjük azt, valamint, hogy a létrehozandó adatbázis felhasználási köre, helye és megjelenítési formája milyen lesz, alapvetően meghatározza a tervezés menetét és eredményét.

3.3.2 E/K modell

Az adatok szerkezetének grafikus ábrázolására szolgál az E/K (egyed-kapcsolat) modell, mely azonban az adatokon végezhető műveleteket nem tartalmazza, ugyanakkor a további tervező munkához elengedhetetlenül szükséges, hiszen a logikai modell (relációs adat(bázis)modell) megalkotása ebből lehetséges.

Az E/K (egyed-kapcsolat) diagram három elemét az egyedhalmazok, az attribútumok, valamint a köztük lévő kapcsolatok alkotják.

Az egyedhalmazokat a hasonló típusú *egyedek* összessége alkotja, melyben egyednek (entity, entitás) tekintünk egy adott típusú absztrakt objektumot.

Az egyes egyedek tulajdonságait az *attribútumok* (attributes, tulajdonságok) írják le, ezek az egyed jellemző tulajdonságai, míg két (vagy több) egyedhalmazt a *kapcsolatok* (relationship) kötnek össze egymással (ekkor bináris, illetve sokágú kapcsolatról beszélünk). Az *egyszerű attribútumok* értékei kisebb egységekre már nem bonthatóak, míg az *összetett attribútumok* további kisebb összetett attribútumokból áll. *Egyértékű attribútumról* akkor beszélünk, amikor egy konkrét előfordulás során mindösszesen egyetlen értéket vehet fel. A *többértékű attribútum* ugyanakkor több értéket is felvehet.

A *kulcsattribútum* egyértelműen azonosítja az egyedtípus bármely előfordulását, ezt aláhúzással jelöljük az ábrázolásakor.

Magát az E/K diagramban a tárolandó adatok és köztük lévő kapcsolatok grafikus ábrázolására többféle módszer is rendelkezésünkre áll, az egyik ábrázolási módban az egyedeket négyzettel, az attribútumokat ellipszissel, míg kapcsolataikat rombusz jelöléssel (az alakzatokban feliratokkal) történik, míg a másik féle ábrázolás az egyedeket és az attribútumokat táblában (téglalapban) ábrázolja, melynek fejlécében az egyed megnevezése, míg az alatta lévő sorokban az attribútumokat soroljuk fel (többnyire a változótípus megjelölésével). Kapcsolataikat a táblákat összekötő, a kapcsolat típusát jelölő vonalakkal (itt is többféle jelölésmód lehetséges: a vonalon számmal, ill. betűvel jelöljük a kapcsolat típusát, vagy megszakítással és ún. „varjúlábbal”) ábrázoljuk. A kulcsot ebben az ábrázolási módban is aláhúzással (és félkövér betűtípussal) jelöljük.

A különböző alakzatok alkotják a gráf csúcspontjait. A gráf éleit az egyedhalmazok és attribútumaikat, valamint a kapcsolatok és adathalmazait összekötő vonalak adják. [2]

Bármely típusú ábrázolást használjuk is, célszerű a tervezés során egyféle ábrázolási módnál maradni (dolgozatomban a sokszínűség és szemléltetés kedvéért mindkettő formában bemutatom az általam szerkesztett E/K diagramot).

A kapcsolatok ábrázolásának különféle módjai terjedtek el (Chen-féle jelölés, vonalas összekötés, „varjú-lábas” ábrázolás), melyek közül jelen szakdolgozatban a Chen-típusú jelölést alkalmazom, a másik két ábrázolási formára részletesen nem térek ki.

A modellben azokat az attribútumhalmazokat, melyek egyértelműen meghatározzák az adott egyedet *kulcsnak* nevezzük, és aláhúzással jelöljük. Amennyiben egy egyedhalmaznak több kulcsa is van, csak az elsődleges kulcsot (primary key) jelöljük. [2]

Az E/K diagram megtervezéséhez — az MS Word alakzatain túl — számos applikáció/szoftver áll rendelkezésünkre, mind ingyenesen hozzáférhető, mind pedig előfizetési formában. Ezen strukturált diagramok szerkesztésére szolgáló programok összefoglaló táblázatát mutatja be (a teljesség igénye nélkül) a 4. táblázat.

4. táblázat — Online hozzáférhető E/K diagram tervezők

Név	Elérhetőség	Logo	Elérhetőség
Dia Diagram Editor	ingyenesen letölthető		http://dia-installer.de
Smartdraw	online elérhető		https://www.smartdraw.com
Miro	online elérhető		https://miro.com/
Canva	online elérhető		https://www.canva.com
Visual Paradigm	online elérhető		https://online.visual-paradigm.com
ERDPlus	online elérhető		https://erdplus.com
Lucidchart	online elérhető		https://www.lucidchart.com
Figma	online elérhető		https://www.figma.com

3.3.3 Relációs adatbázis séma, normalizálás

Az adatbázis modellezés és implementálás eljárásának következő lépése az E/K modell relációs adatbázissémává történő átalakítása, melyhez a két adatmodell egyes elemeinek megfeleltetését a 3.2 ábra mutatja be.

Maga a relációs modell táblázatokból áll, melyek maguk a *relációk*. Az adatokat táblázatosan tárolja, melyekből a *lekérdezés* (keresett adat lehívása) *relációs műveletekkel* történik. A modell felépítése és főbb jellemzőinek ismertetése a 2.1.1 fejezetben olvasható.

E/K modell	↔	Relációs adatmodell
egyedosztály séma		relációséma
tulajdonságok		attribútumok
(szuper)kulcs		(szuper)kulcs
egyedosztály előfordulása		reláció
e egyed		$(e(A_1), \dots, e(A_n))$ sor
$R(E_1, \dots, E_p, A_1, \dots, A_q)$		$R(K_1, \dots, K_p, A_1, \dots, A_q)$
kapcsolati séma		relációséma
ahol		ahol
E_i — egyedosztály		K_i — az E_i (szuper)kulcsa
A_j — saját tulajdonság		

3.2. ábra Az egyes adatmodellek összevetése [14]

A relációk vizsgálatához és javításához, ezzel együtt a normálformák előállításához először is ismernünk kell a *funkcionális függőség* fogalmát, mely tulajdonképpen a relációkulcsok általánosításának tekinthető. A reláció tartalmaz olyan attribútumot (kulcsot), melytől a többi attribútum függ.

Relációs adat(bázis)modell esetén értelmezhető az *elsődleges kulcs* (primary key), valamint az *idegen kulcs* (foreign key) és a *szuperkulcs* fogalma.

A kulcs a rekord (sor/egyed) egyértelmű azonosítására szolgáló, (általában) a reláció valamelyik attribútuma (oszlopa), vagy oszlopainak kombinációja. Meghatározásánál előfeltétel, hogy minimalitásra törekedjünk, az egyértelmű meghatározáshoz szükséges minimális számú attribútumot (oszlopot) válasszuk.

Idegen kulcsnak a reláció azon attribútumait nevezzük, melyek egy másik relációban kulcsot alkotnak (másik táblázatra hivatkozik). A két tábla egymással logikai kapcsolatban van. Ezeket az ábrázolásnál szaggatott vonallal jelöljük.

Reláció foka a relációsémában az attribútumok (oszlopok) száma, míg a *reláció számszága* a sorok száma.

A helyes értelmezéshez fontos még megismernünk a *gyenge egyed* fogalmát és szerepét. Azokat az egyedeket (egyedhalmazokat) nevezzük gyenge egyednek (egyedhalmaznak), melyeknek saját attribútuma (melyet kulcsként használhatnánk) nincs, és amelyek kulcsában szereplő attribútumok (néhány közülük vagy az összes) más egyedhalmaznak is attribútuma, tehát a gyenge egyedek kizárólag a kapcsolódó egyedhalmazok attribútumaival azonosíthatóak.

Tervezésnél a gyenge egyedhalmazoknak két okból van szerepe: egyrészt a sokágú kapcsolatok binárisra alakításában, másrészt az öröklődési kapcsolat speciális hierarchikus szerkezetének alakítása esetén. Jelölésük általában dupla kerettel (vagy bizonyos speciális esetekben aláhúzással) történik.

A korábban megtervezett E/K modell átalakítása relációs adat(bázis)sémára a normálformák segítségével a kulcsok és a funkcionális függőségek ismeretére épít, és a következő lépésekből áll (minden esetben először az E/K modell gyenge egyedei azonosítsuk, majd a nem gyenge egyedhalmazokkal kezdjük az átalakítást):

1. **Egyedek leképezése** — minden egyedhalmaz átírása relációvá úgy, hogy attribútumaival definiáljuk (relációkat hozunk létre belőlük ugyanazon névvel és attribútumhalmazzal).

2. **Kapcsolatok leképezése** — minden kapcsolatot olyan relációvá írunk át, melyek attribútumai a kapcsolat egyedhalmazainak kulcsai.

Ehhez először a kapcsolatban résztvevő egyedhalmazok minden kulcsattribútumát bele vesszük, majd magának a kapcsolatnak az attribútumait is hozzáadjuk (amennyiben vannak ilyenek).

Az ismétlődések elkerülése miatt azokban az esetekben, amikor adott egyedhalmaz többször is benne van egy kapcsolatban különböző szerepekben, a kulcsattribútumát szerepeinek számának megfelelően kell szerepeltetnünk, azonban át kell neveznünk azt.

3. **Speciális kapcsolatok előfordulásának vizsgálata és azok kezelése/átírása** — gyenge egyedhalmazok, öröklődési kapcsolatok (osztályhierarchia), reláció összevonások, kombinációk.

Gyenge egyedhalmazok előfordulása esetén az átalakítások alkalmával adott egyedhalmazhoz tartozórelációnak az egyedhalmaz attribútumait kell tartalmaznia, hanem más egyedhalmazok kulcsattribútumait is. Ez azt is jelenti, hogy adott egyedhalmaz kulcsattribútuma mellett az ehhez hozzájáruló egyéb egyedhalmazok (kiteljesítő kapcsolatokhoz tartozó) attribútumokat is tartalmaznia kell a relációnak. Fontos azonban megjegyezni, hogy egy kiteljesítő kapcsolat esetén reláció megadása nem szükségszerű.

Kerüljük el a névkonfliktusokat, ne használjuk ugyanazt a nevet többször, ha szükséges, nevezzük át az attribútumokat.

Az **első normálforma (1NF)** létrejöttéhez az alábbi feltételek együttes teljesülése szükséges:

1. oszlopok száma minden sorban azonos, sorrendje lényegtelen;
2. minden sor és oszlop egyedi (különböző, nincs két egyforma);
3. minden sorhoz tartozik egyedi kulcs, melytől funkcionálisan függ a többi attribútum;
4. adott oszlopban minden érték ugyanarra a tárgyra vonatkozik, nincs többértékű attribútum (csak egy-egy attribútum értékét veszik fel az egyes oszlopok). [15]

Röviden és egyszerűen, ha a tábla minden sorának cellájában pontosan egy attribútumérték áll, akkor az 1NF-ban van. Amennyiben az attribútumok nem elemiek, vagy az attribútumok között ismétlődés van, akkor *nulladik normálformáról* beszélhetünk (0NF).

Ha a fenti feltételek teljesülnek, rendelkezésünkre áll maga a reláció, mellyel az adatbázis megalkotásakor dolgozhatunk. Az 1NF teljesülését minden relációs adatbáziskezelő rendszer megköveteli, azonban, amikor a rendelkezésünkre álló adatokból felépítjük adatbázisunkat, akkor a legritkább esetben van rögtön 1NF-ban.

Amennyiben az így létrejött relációk vizsgálatakor ezek a feltételek egy adott tábla esetén nem teljesülnek, javítani szükséges: vagy annyi sorra bontjuk a többes attribútumértéket tartalmazó sort, ahány attribútumérték található benne (plusz sorok beszúrása), vagy pedig szétbontással több relációra bontjuk.

A javítás normalizálással történik — általában három, ritkább esetben négy lépésben, második (2NF), harmadik (3NF), negyedik (4NF – BCNF/Boyce-Codd), illetve ötödik (projection-join/PJNF) normálformára alakítással, melyek normálformák egymásra épülnek. Célszerű tervezésnél is ezt a sorrendet tartani. [15]

A normalizálás folyamata maga egy táblázat szétbontó eljárás (*dekompozíció*), mely a redundancia megszüntetésére irányul, szétbontó relációs műveletekkel. Eredményeként kisebb tárolási igényű táblát kapunk (nincs feleslegesen tárolt adat), az anomáliák (törlési, módosítási, beszúrási) megszüntetésével, és a tábla logikai áttekinthetősége is jobb lesz.

A normalizálás folyamata nem sematikus, általában elmondható, hogy ugyanaz az attribútum halmaz több különböző redundancia-mentes sémával is leképezhető.

A normalizálás második lépésében a **második normálforma (2NF)** létrejöttéhez az alábbi feltételek teljesülése szükséges:

1. első normálformában van és
2. a nem kulcs attribútumok (minden másodlagos attribútum) funkcionálisan teljes függésben van az elsődleges kulcstól (részleges függések nincsenek). [15]

Ez fennállhat akkor is, ha a kulcs egyetlen attribútumból áll, vagy ha nincsenek másodlagos attribútumok a relációban (minden attribútum része a kulcsnak).

Ha a reláció nincs 2NF-ben, dekompozícióval hozhatjuk a megfelelő formára. A dekompozíció során egy táblából több kisebb táblát (új relációkat) hozunk létre, melyek között kapcsolat van (amelyek már 2NF-ban vannak). Ebben a formában jelennek meg az idegen kulcsok.

A **harmadik normálforma (3NF)** létrejöttéhez az alábbi feltételek teljesülése szükséges:

1. második normálformában van és
2. közvetlen (tranzitív) függések nincsenek, a nem kulcs attribútumok kizárólag az elsődleges kulcstól függenek funkcionálisan (egymástól nem), ez a függés teljes. [15]

3NF-ba hozni úgy tudjuk a relációt, hogy megszüntetjük a tranzitív függéseket. Ez minden esetben veszteségmentes és függőségörző felbontás.

Amikor még a 3NF esetén is vannak anomáliák, feleslegesen tárolt adatok, ismétlődések előfordulása, szükséges a redundancia további csökkentése, ekkor a **negyedik normálforma (4NF, BCNF/Boyce-Codd)** kialakításával oldhatjuk ezt meg, melynek létrejöttéhez az alábbi feltételek teljesülése szükséges:

1. harmadik normálformában van és
2. legfeljebb egy többértékű függése van (benne nem triviális függőség kizárólag szuperkulcstól van). [15]

BCNF-re hozni a relációt veszteségmentes dekompozícióval hozzuk (de a függőségőrzést nem tudjuk tartani) két részre bontva, majd, ha az új részek nem BCNF-ek, akkor ezt megismételjük.

Az **ötödik normálforma (5NF, projection-join/PJNF)** a legvégső forma, ennél jobbat találni matematikai értelemben nem lehet. Abban az esetben beszélünk 5NF-ről, ha 4NF-ban van a reláció és az egyedben lévő minden kapcsolatfüggés kizárólag az egyed kulcsjelöltjei között létezik. A normalizálási lépése célja az, hogy egy veszteségmentes szerkezetet kapjunk, azonban az 5NF alkalmazása időigényes és bonyolult, általában a 4NF alkalmazása is elegendő. [1]

4. AZ ADATBÁZIS MEGTERVEZÉSÉNEK LÉPÉSEI

Szakdolgozatom témájaként választott adatbázis-tervezés egyik nehézségei a forrásadatok sokrétősége és ehhez kapcsolódóan a különböző fájlformátumok kezelésének képessége (kép, adattábla, térképi adatok, 3D pontfelhő adatai), valamint a jogosultságkezelés megfelelő beépítése az adatbázisba. Mindemellett természetesen az adatbázisokkal szemben támasztott, a 2.1.2 fejezetben ismertetett alapkövetelményeknek biztosítottnak kell lenniük.

4.1 Informatikai háttér

Az adatbázis-tervezés fejlődése szoros kapcsolatban állt az informatikai hardver- és szoftverfejlődéssel. A szoftverrendszerek terjedelmének és bekerülési (fejlesztési és üzemeltetési) költségeinek csökkenésével egyre szélesebb körben váltak elérhetővé olyan felhasználóbarát szoftverek, mint a 3.3.2 fejezetben tárgyalt E/K diagramszerkesztők, illetve a különféle adatbáziskezelő-rendszerek. Ezek közül jelen fejezetben az általam használt szoftvereket, illetve rendszereket ismertetem.

E/K diagram szerkesztők — Dia, Figma

A 3.3.2 fejezetben bemutatott, a strukturált diagramok szerkesztésére szolgáló applikációk/programok közül munkám során a *Dia Diagram Editor*, valamint az online elérhető *Figma* design tervező platform (www.figma.com). Mindkét szerkesztőnek megvannak a maga előnyei és korlátai, melyekkel munkámat segítették.



A *Figma* platformjának egyszerűen, online elérhető (regisztrációt igényel) diagramszerkesztője igen látványos dizájn megalkotását, bemutatását és egyszerű (élő linkkel történő) megosztását teszi lehetővé. Online elérhető-

sége átlépi a fizikai korlátokat, rendszeres automatikus mentés lehetősége, mintatervek nagy tárháza és a közösségi támogatás háttere egyszerűsíti a munkát.

Azonban az ingyenes hozzáférhetőségnek korlátai is vannak: egyetlen design terv megalkotását teszi csak lehetővé térítésmentesen, valamint olyan sok funkció áll a felhasználó rendelkezésére, hogy egy kezdő felhasználó számára használata a bőség zavara miatt igen időigényes.



Egyszerűbb volt számomra a feladat során megjelenő folyamatos (normalizálással együtt járó) változtatások felvezetése miatt a *Dia Diagram Editor* szoftvert használni, melyet ugyan telepíteni szükséges, de ingyenesen letölthető a netről (<http://dia-installer.de/>).

A szoftver egyaránt képes olvasni, kezelni és importálni az UML diagramokat. Ehhez a szoftverhez elérhető a *Parse-Dia-Sql* és *dia2code* (a fent jelölt forrásból szintén ingyenesen letölthetőek) szoftverek, melyek a Dia Diagram Editor-ban megszerkesztett diagramok forráskóddá konvertálásához használható DBMS-ekhez, különböző programnyelveken. Több, különböző programnyelvre is konvertálhatóak a Dia Diagram Editor fájlljai: C, C#, CORBA IDL, ADA, PHP, Java, PHP5, Python, shapefile, Ruby, SQL (a C++-hoz a *dia2code*+ áll rendelkezésünkre).

Mindkét szoftver egyaránt elérhető Windows, Linux és Mac OS operációs rendszerekre. A *Dia Diagram Editor* 0.97.2, míg annak kiegészítéseit, a *dia2code* 0.8.5 és a *Parse-Dia-Sql* 0.23.0 verzióit telepítettem munkámhoz.

pgAdmin 4



Jelen szakdolgozat kitűzött feladatában egy, a **PostgreSQL** segítségével létrehozandó adatbázis megtervezésével foglalkozom, ezért a *pgAdmin 4-et* a PostgreSQL ingyenesen letölthető, az adatbázisok kezelését segítő, számos funkcióval rendelkező nyílt forráskódú kezelőeszközt használom, mely hatékony grafikus felületet biztosítva egyszerűsíti le az adatbázis-objektumok létrehozását, karbantartását és használatát. A *pgAdmin 4* lehetővé teszi számomra, hogy egy interaktív felhasználói felületen keresztül kapcsolódjak a PostgreSQL adatbázis-kiszolgálóval.

A v7.1 verzióját telepítettem a megtervezett adatbázis bemutatásához.

Ahhoz, hogy a korábban megtervezett E/K diagramból közvetlenül generáljunk SQL script-eket, további lehetőségek is rendelkezésre állnak, ilyenek például a SQLDB (www.sqldb.com) vagy a Vertabelo (www.vertabelo.com) online portáljai, azonban ezek az eszközök kizárólag térítés (havi vagy éves előfizetés) ellenében vehetőek igénybe.

4.2 Specifikáció, adatgyűjtés

A feladat specifikációját és az adatok meghatározását a projekt kiírása egyértelműen meghatározta. A feladat egy olyan adatbázis megtervezése, mely egy webalkalmazás (honlap) háttéréül szolgálhat a különböző forrásból származó, eltérő formátumú adatok keresésére és megjelenítésére, valamint a projekt eredményeinek szemléletes bemutatására a felhasználói jogosultságok kezelése mellett.

Az alapkritériumok összefoglalva:

- Jogosultságkezelés (felhasználók definiálása) biztosítása a három alap felhasználói csoporthoz (Admin, Felhasználó, Résztvevő).
- Kulcsszavas adatlehívás.
- Eltérő forrásokból származó, eltérő formátumú adatforrásokat legyen képes kezelni az adatbázis.
- Adathozzáférés és letöltés lehetősége megoldott legyen a felhasználók számára.

Ezen feltételek részletes ismertetését a következő alfejezetek tartalmazzák.

4.2.1 Metaadatok — adatforrás, adattípus

A projektben felhasználható, több forrásból származó metaadatok köre igen sokrétű. Ebből adódóan az adatbázisnak képesnek kell lennie a sok különböző formátum kezelésére: kép, adattábla, térképi adatok, műholdfelvétel, 3D pontfelhő adatai (dataset).

5. táblázat — Adatforrások specifikációja

Adatforrás típusa	Példák
térképi adatok	nagy méretarányú földhasználati térkép
3D pontfelhő	WorldView, RADARSAT, Landsat, Lidar 3D pontfelhő adatok, pontfelhők 3D pdf vizualizációja (előnézethez)
kép, műholdfelvétel	műholdfelvételek (Sunflower műholdadatok) és leíró metaadataik (cím, műhold neve, koordináták, kulcsszavak)
adattábla	mezőgazdasági környezet alapadatai (pl. földhasználat, domborzat és további földdel kapcsolatos hivatalos térképek adatai), földfelszíni vegetációs adatok (pl. vegetációs index) növényfenológiai adatok a mezőgazdasági terület vegetációs és talajnedvességi adatai, talajszerkezeti adatok
szöveges dokumentumok	leírások (pl. műveleti leírások)

Az adatbázisterv az adatok típusát és forrását két külön táblában kezeli. Az adattípus (**Data_type**) meghatározásához az egyedi azonosító mellett a név (**Type_name**) és a leírás (**Type_description**) hozzáadásának lehetősége került még a táblába.

Az adatok forrásával kapcsolatban (**Source**) az egyedi azonosító (**Source_ID**) mellett a név (**Source_name**), e-mail cím (**Source_email**), URL (**Source_url**) és leírás (**Source_description**) tárolása lehetséges.

Az adatokkal (**Metadata**) kapcsolatosan az adatbázis az egyedi azonosító mellett (**Metadata_ID**) tárol dátumot (**Metadata_date**), URL-t (**Metadata_url**), leírást (**Metadata_description**), nevet (**Metadata_name**), földrajzi koordinátát (**Metadata_coordinate_X**, **Metadata_coordinate_Y**, **Metadata_coordinate_Z**) és annak megjelölését, hogy az adott adat forrásadat-e vagy már egy adatfeldolgozás eredménye (**Metadata_SourceData**).

4.2.2 Jogosultságkezelés — Felhasználói csoportok és jogosultságok

A **jogosultságkezelés** megfelelő beépítése az adatbázisba szintén a feladat része volt.

Elvárásként fogalmazódott meg, hogy az *adatokhoz való hozzáférés biztosítása* a lekérdezés, letöltés, feltöltés, módosítás, törlés szintjén jogosultsági szinthez kapcsolódóan történjen.

Az adatokhoz való hozzáférés biztosítása mellett a jogosultságkezeléssel kapcsolatos másik elvárás az volt, hogy az adminisztrációs feladatkör is jól elkülönüljön.

A felhasználók (**User**) esetén adott volt, hogy milyen jellemző attribútumok meghatározására lesz szükség egy-egy felhasználóval kapcsolatosan, így a következőket hoztam létre:

- ID (**User_ID**)
- név — vezetéknév, keresztnév, középső név (**User_firstname, User_middlename, User_surname**)
- belépési felhasználónév (**User_username**)
- belépési jelszó (**User_password**)
- e-mail cím (**User_email**)
- regisztráció dátuma (**User_registration_date**)
- aktív-e vagy sem (**User_activation**)

Szintén követelmény volt, hogy az adatbázis kezelje a felhasználók intézményét (**UserGroup_Country_Institution**) és annak országát (**UserGroup_Country_name** és ezeket külön kezelje (**UserGroup_Country**). Egyedi azonosító (**UserGroup_Country_ID**) és leírás/megjegyzés hozzáadásának lehetősége (**UserGroup_Country_description**) itt is elvárás volt.

Mivel a szakdolgozatom egy olyan projektet támogat, melynek résztvevői célul tűzték ki a meglévő, valamint feldolgozott adatok bemutatását más, a projekten kívüli felhasználók részére is, ezért három felhasználói csoportot (**UserGroup**) definiáltam: Admin, Résztvevő, Felhasználó. Minhárom felhasználói csoport külön azonosítót (**UserGroup_ID**) és

nevet (**UserGroup_name**) kap, rövid leírás (**UserGroup_description**) hozzáadása az egyes csoportokhoz *text* formátumban lehetséges.

Az egyes felhasználói csoportoknak más-más jogosultságok oszthatók ki. A jogosultság-kiosztást a 6. táblázat mutatja be.

A felhasználói jogosultságok (**User_authorization**) kiosztása a három csoportot illetően a táblázatnak megfelelően a **UserGroup_ID** alapján történik, külön tábla kezeli, *boolean* formátumban megadva (Yes/No választási lehetőséggel a hozzárendelés egyértelmű) definiálja az egyes attribútumokat. Elvárás volt itt is, hogy legyen lehetőség külön leírás beillesztésére (**User_authorization_description**).

6. táblázat — Jogosultság-kiosztási tábla

Felhasználói csoport (UserGroup)	Jogosultságok (User_authorization)									
	Adattal kapcsolatos					Felhasználói				
	Lekérdezés (User_authorization_ViewData)	Letöltés (User_authorization_DownloadData)	Feltöltés (User_authorization_UploadData)	Módosítás (User_authorization_ModifyData)	Törlés (User_authorization_DeleteData)	Új felhasználó felvétele (User_authorization_CreateUsers)	Felhasználó aktiválása (User_authorization_ActivateUsers)	Felhasználó deaktiválása (User_authorization_DeactivateUsers)	Felhasználó törlése (User_authorization_DeleteUsers)	Felhasználó módosítása (User_authorization_UpdateUsers)
Admin	X	X	X	X	X	X	X	X	X	X
Felhasználó	X	X								
Résztvevő	X	X	X	X	X					

Ennél részletesebb jogosultság-kiosztás (pl. fájl típus vagy forrás szerint) akkor lehetséges, ha az üzemeltető meghatározza, mely típusú adatokhoz férhetnek hozzá az egyes felhasználók és azokat hogyan szeretné leosztani.

4.2.3 Adatlehívás— kulcsszavak

Feladatom a kulcsszavas adatlehívás lehetőségének megteremtése volt az adatbázis segítségével.

A kulcsszavak (**Keywords**) definiálása az egyedi azonosító mellett (**Keyword_ID**) azok típusával (**Keyword_type**) és leírásának megadásával (**Keyword_description**), valamint azok meghatározásának (**Keyword_date_created**) és esetleges módosításának (**Keyword_date_modified**) megadásával lehetséges, a kapcsolódó táblák idegen kulcsainak (foreign key) segítségével.

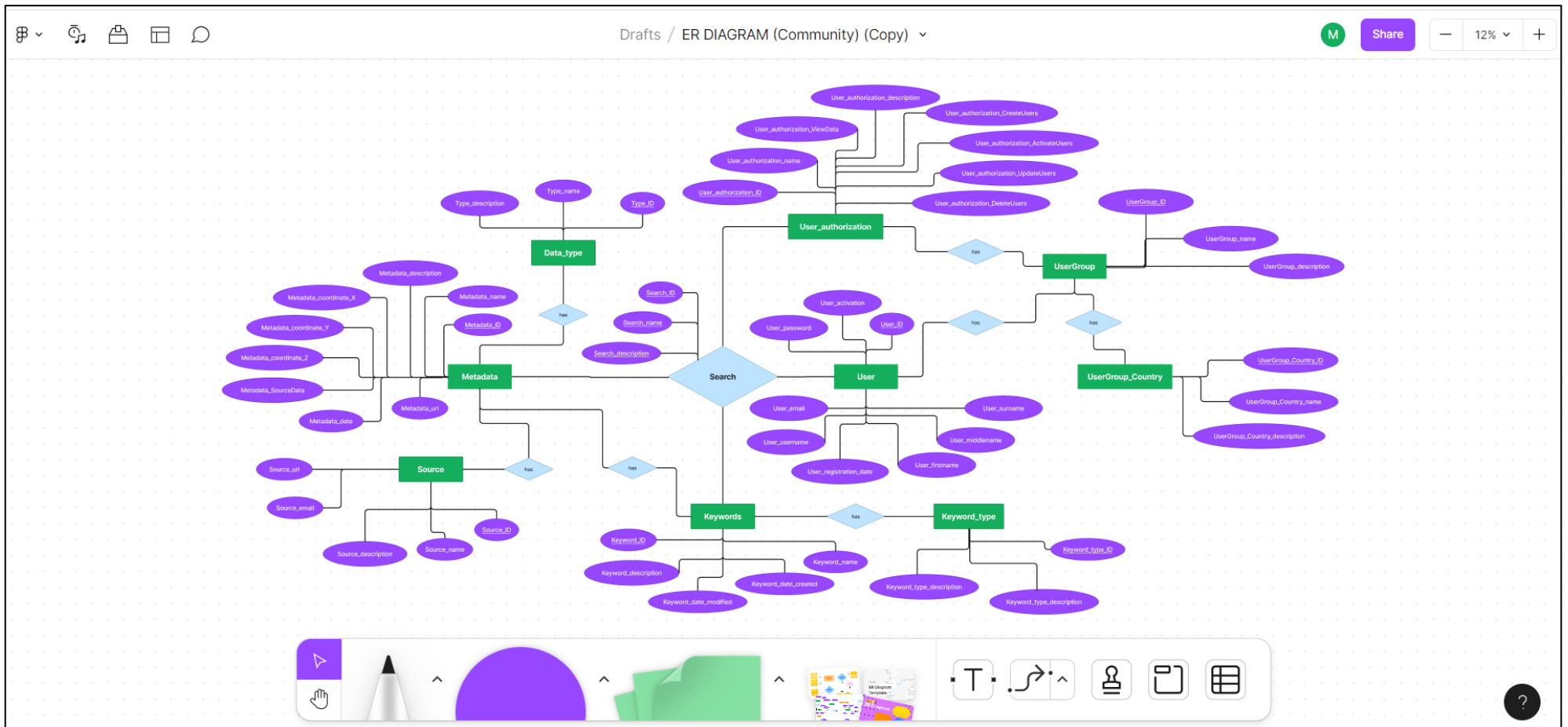
Maga az adatlehívás (**Search**) az adatok, kulcsszavak, felhasználók és jogosultságaik függvényében egyedi azonosítóval (**Search_ID**), névvel (**Search_name**), valamint a keresés dátumának és idejének (**Search_datetime**) és leírásának (**Search_description**) attribútumaival jellemezhető.

4.3 E/K-modell kidolgozása

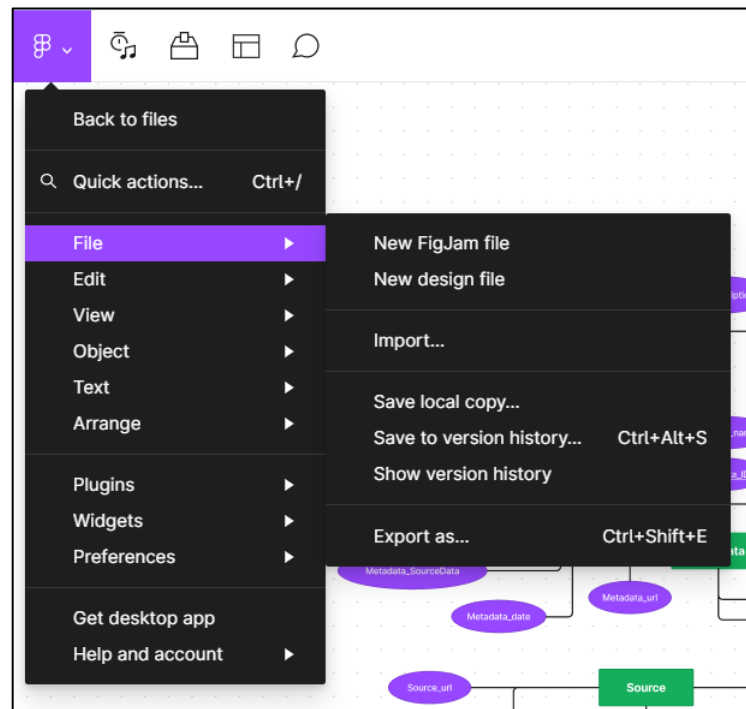
Első körben elkészítettem egy E/K diagramot a rendelkezésemre álló alapinformációk alapján, hogy az adatbázis elemeit és azok kapcsolatait felvázoljam.

Dolgozatomban célul tűztem ki a két különböző ábrázolási mód és több, különböző diagramszerkesztő bemutatását is. A *Dia Diagram Editor* és *pgAdmin 4* szoftverekkel a rendszervázlat, míg a *Figma* online diagramszerkesztő platform segítségével a sematikus ábrázolás módját mutatom be (ld. lent, 4.8. ábra — sematikus rajz— és 4.9. ábra — rendszervázlat — ábrák). Mindkét ábra ugyanazt az E/K diagramot szemlélteti.

A *Figma* online diagramszerkesztő platform segítségével lehetőségem volt egy, már meglévő séma másolására és átszerkesztésére, melynek során a séma tartalmának módosítása mellett annak formai jellemzőit (például szín, vonalvastagság és-típus) is módosíthattam, illetve saját stílusomra szabhattam. A platform kezelőfelületét a 4.1 ábra, míg menürendszerét a 4.2 , illetve alsó szerkesztősávját a 4.5 ábrák mutatják be.



4.1. ábra A Figma platform kezelőfelülete (www.figma.com)



4.2. ábra A Figma platform menürendszere (www.figma.com)



4.3. ábra A Figma platform alsó szerkesztősávja (www.figma.com)

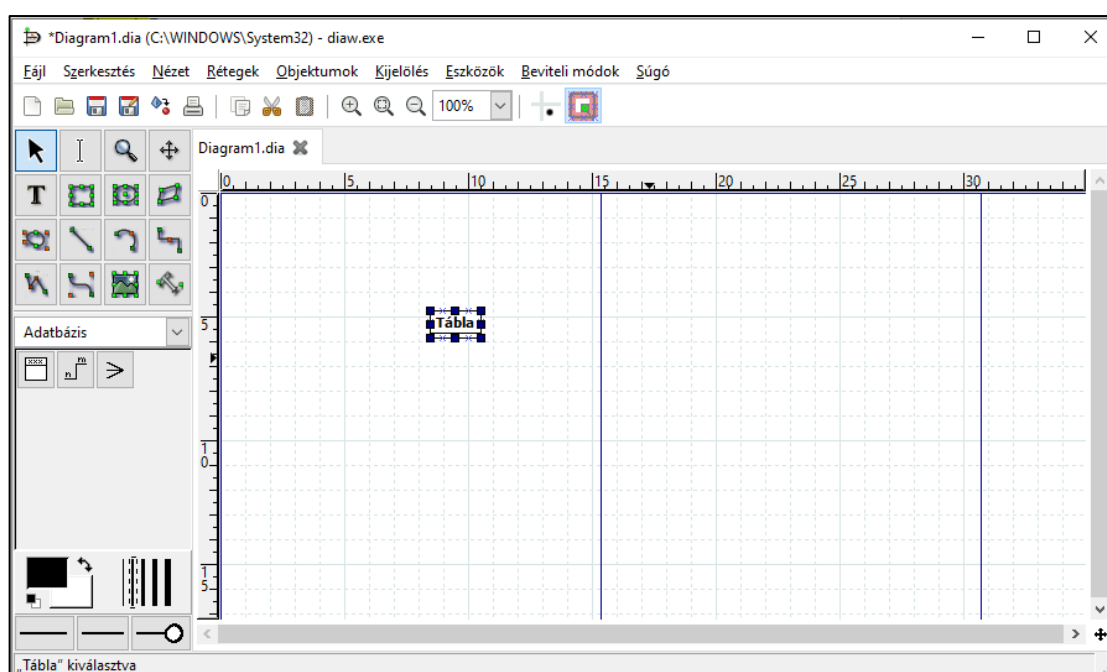
A platformlehetőséget biztosít a felhasználó számára a megalkotott diagramok elmentésére (.jam kiterjesztéssel saját fájlformátumban), valamint exportálására is különböző fájlformátumokat felkínálva (pl. .pdf, .jpg, .png).

A Figma.com platformján megalkotott sematikus E/K modell a 4.8 ábrán látható.

Az egyedek és attribútumaik rendszervázlat-szerű ábrázolására a *Dia Diagram Editor* szoftverben egységes jelölési rendszert alkalmaztam, melyben az adattáblákat (relációk) téglalapokba foglalva tartalmazza a diagram, felső részben jelölve az adattábla nevét, az alsó részben pedig az attribútumait, valamint az egyes attribútumok mellett azok típusait. Az elsődleges kulcsokat aláhúzással, valamint félkövérrel jelöltem.

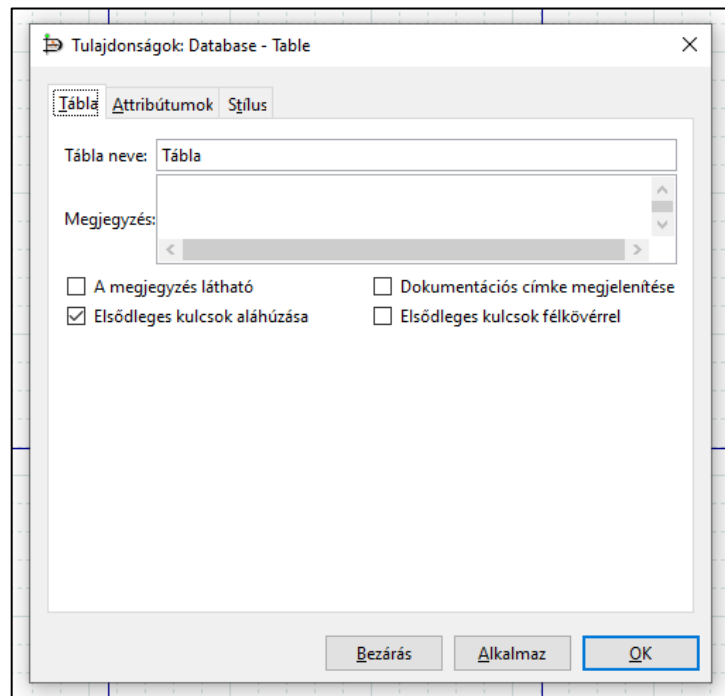
Az attribútum típusok megadása az adatbázis tervező szoftverben segíti az adatbázis létrehozását. A kapcsolatokat a táblák között és a külső kulcsokat a nyilak jelölik, melyeken a kapcsolat típusa is feltüntetésre került. Az attribútumok neve előtti pont, illetve üres kör azt jelölik, hogy az adott attribútum értéke lehet-e NULL vagy sem (+ jelöli azt az esetet, amikor nem vehet az attribútum NULL értéket fel, míg a ° jelölés azt, amikor felvehet)

A *Dia Diagram Editor* szoftver kezelőfelülete, amin a modellt létrehoztam, a 4.4 ábrán látható.



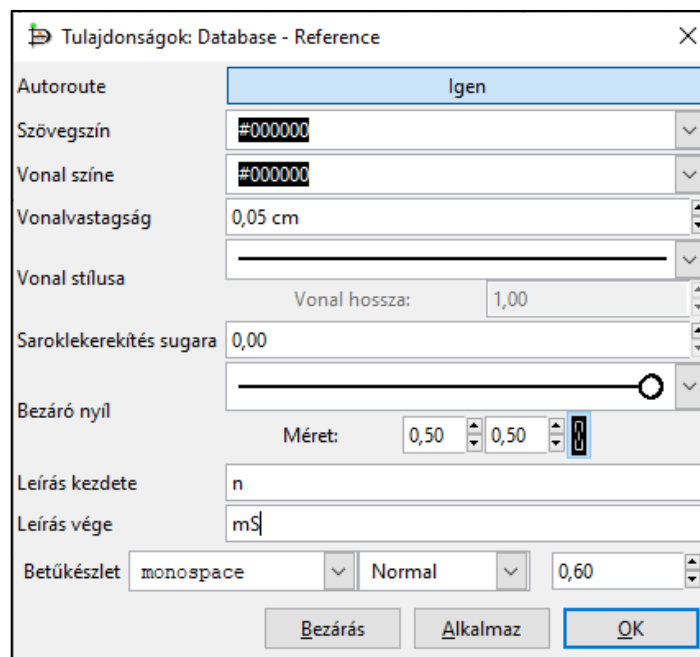
4.4. ábra A *Dia Diagram Editor* szoftver kezelőfelülete

Az adattáblák és attribútumadatok megnevezésénél ékezet nélküli neveket használtam, angol nyelven, a könnyebb programozhatóság érdekében. Az elsődleges kulcsokat jelöltem (aláhúzással és félkövérrrel), az idegen kulcsok külön nincsenek itt jelölve. A *Dia Diagram Editor* szoftver lehetővé teszi annak jelölését is, hogy adott attribútum felvehet-e null értéket vagy sem. Az adattábla létrehozásának lehetőségét és megadható paramétereit a 4.5 ábra mutatja a *Dia Diagram Editor* szoftverben.



4.5. ábra Adattábla létrehozása a Dia Diagram Editor szoftverrel

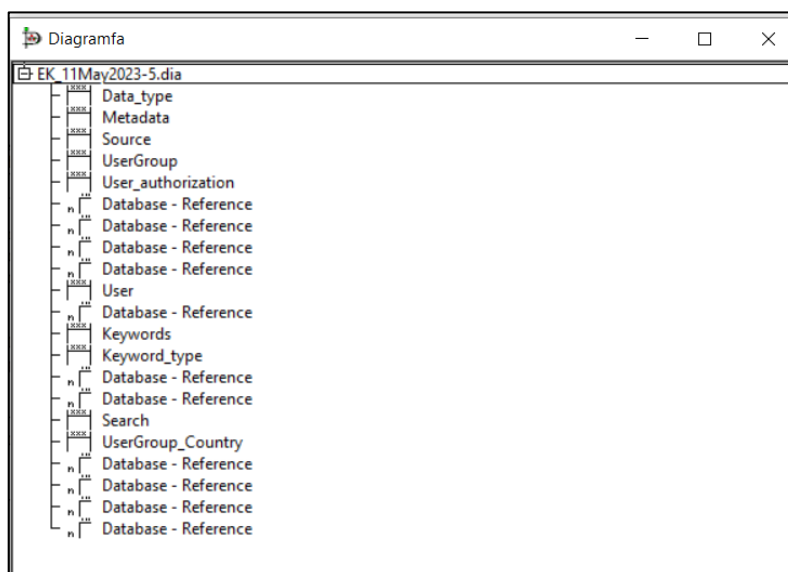
A szoftver a hivatkozások módosítása, specifikációja (tartalmi és formai) külön ablak megnyitásával történik, az 4.6. ábra Hivatkozások szerkesztése a Dia Diagram Editor szoftverrel ábrán látható alapadatok megadásával.



4.6. ábra Hivatkozások szerkesztése a Dia Diagram Editor szoftverrel

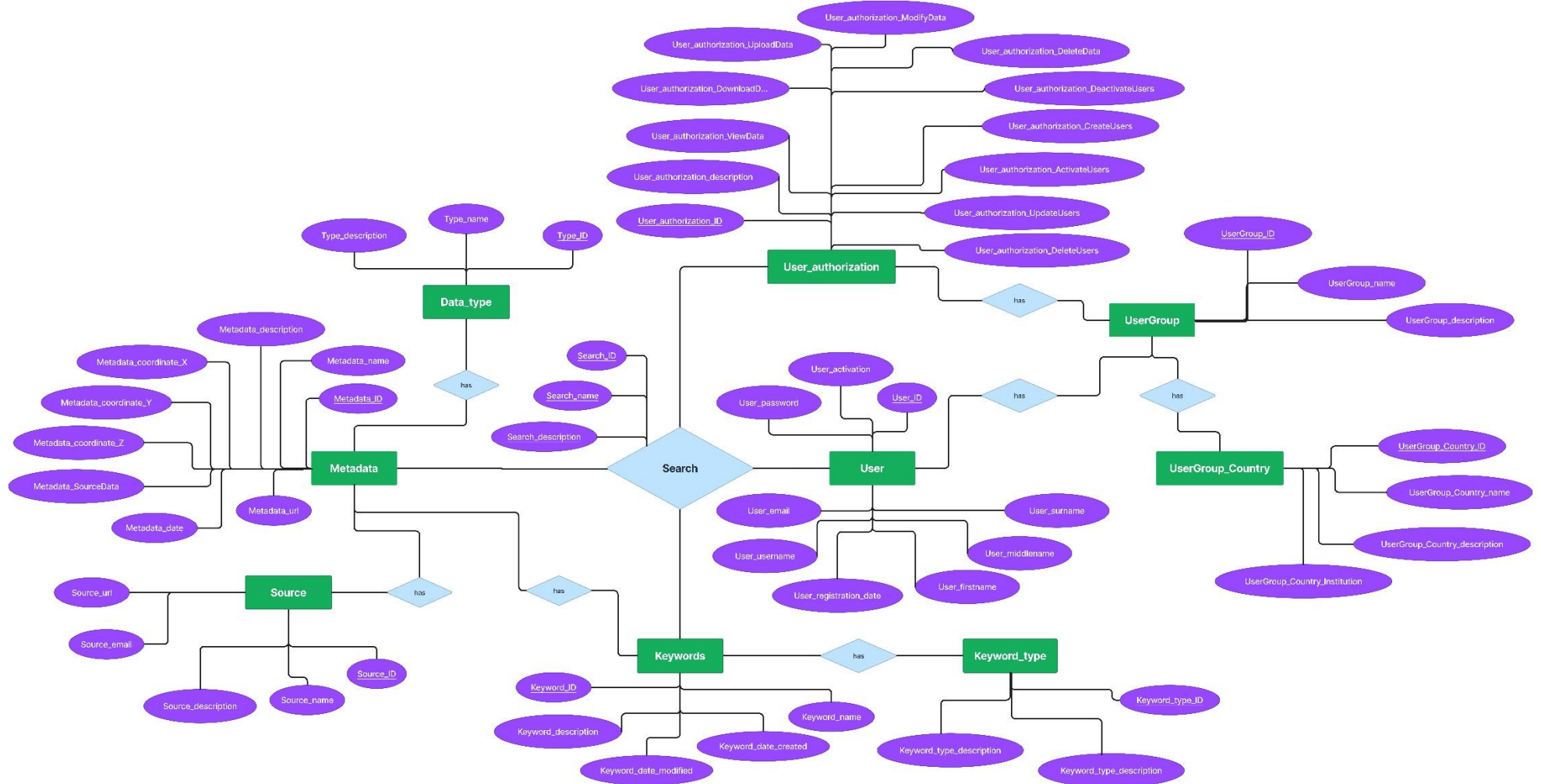
A szoftver lehetőséget biztosít a felhasználó számára a megalkotott diagramok elmentésére (.dia kiterjesztéssel saját fájlformátumban), valamint exportálására is különböző fájlformátumokat felkínálva (a teljesség igénye nélkül: .tif, .tiff, .jpeg, .jpg, .png).

Mindezek mellett a megszerkesztett diagramból a diagramfát is legenerálja (ld. 4.7 ábra).

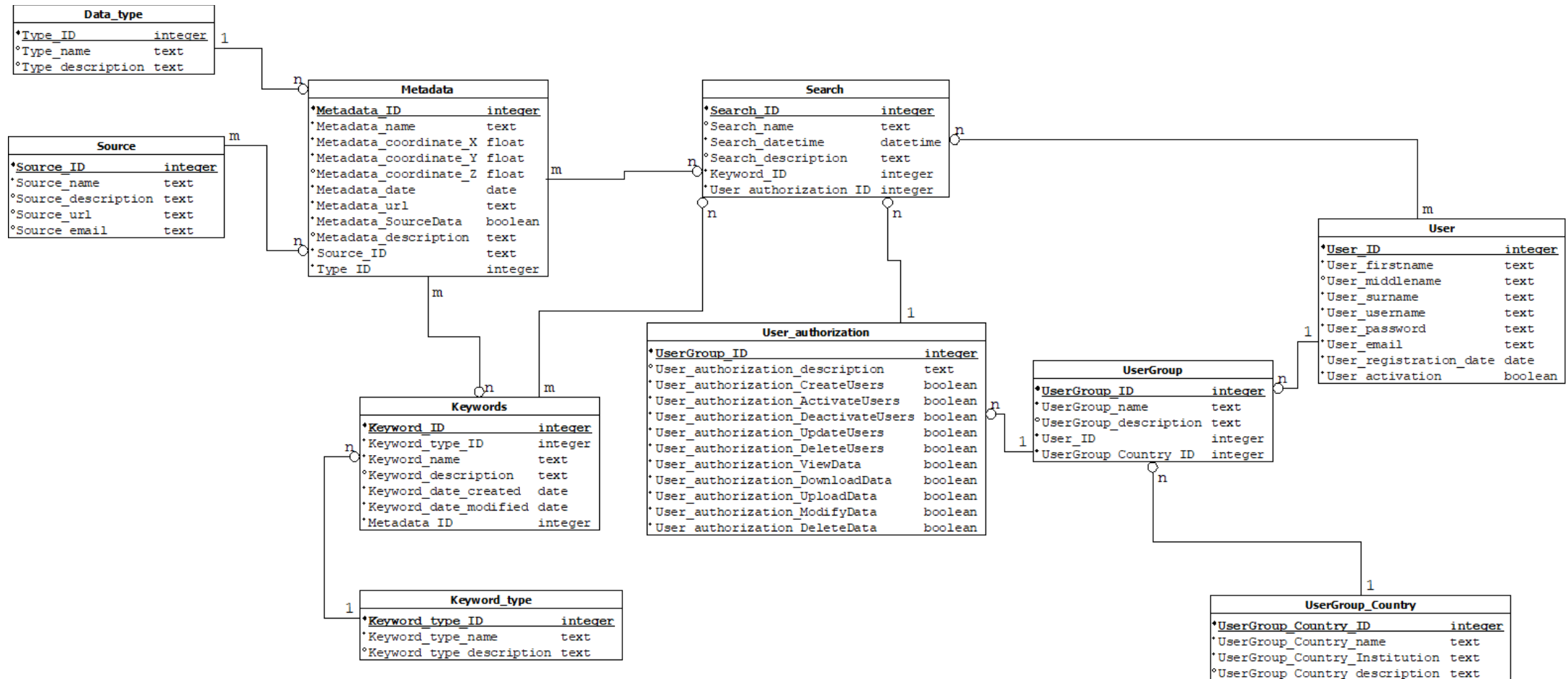


4.7. ábra Diagramfa az elkészült E/K diagramra (Dia Diagram Editor szoftverrel)

Az így megalkotott rendszervázlat-szerű E/K modell a 4.9 ábrán látható, mely tartalmazza a feladat kiírásából a rendelkezésekre álló alapadatok és követelmények kapcsolatrendszerét.



4.8. ábra E/K modell a Figma applikációjával szerkesztve



4.9. ábra E/K modell a Dia Diagram Editor szoftverrel megszerkesztve

A fenti modellben a beállításaim szerint az elsődleges kulcsokat aláhúzással és félkövérrel jelöltem.

4.4 Relációs adatbázis séma és normalizálása

Az adatbázis modellezés következő lépése az E/K modell relációs adatbázissémává történő átalakítása volt, melynek tökéletesítését a normalizálással folytattam.

A relációséma normalizálása során megvizsgáltam a 3.3 fejezetben tárgyalt egyes normálforma-feltételek teljesülését, amennyiben azt tapasztaltam, hogy valamely feltétel nem teljesül, a sémát aszerint módosítottam, fejlesztettem. A tervezés menetének egyes változatait nem, kizárólag a végleges séma értékelését mutatom be itt.

Az így kialakult végleges sémára (ld. 4.9 ábra) elmondható, hogy

1NF teljesül, mert a tábla minden sorának cellájában pontosan egy attribútumérték áll, az attribútumok nem elemiek, és az attribútumok között ismétlődés nincs.

Az elsődleges kulcsok táblában az ID kiterjesztésű attribútumok, melyek külön aláhúzással és félkövér betűtípussal vannak jelölve.

2NF teljesül, mert

1. első normálformában van és
2. a nem kulcs attribútumok (minden másodlagos attribútum) funkcionálisan teljes függésben van az elsődleges kulcstól (részleges függések nincsenek).

Ennek a feltételnek a teljesülése is adott, mivel táblánként 1-1 elsődleges kulcs van (a kulcs egyetlen attribútumból áll), bár vannak másodlagos attribútumok is.

3NF teljesül, mert

1. második normálformában van és
2. közvetlen (tranzitív) függések nincsenek, a nem kulcs attribútumok kizárólag az elsődleges kulcstól függenek funkcionálisan (egymástól nem), ez a függés teljes.

Ez teljesül, ezért további normálformák létrehozására nem volt szükség.

A normalizálási eljárás végére eljutottam a 4.9 ábrán látható, módosított és végleges E/K diagramhoz, mely felhasználható az adatbázis létrehozásához.

5. A LÉTREHOZOTT ADATBÁZIS BEMUTATÁSA

Az elkészült E/R diagram alapján *pgAdmin 4* segítségével generáltam egy minta adatbázist (létrehoztam a táblákat), melybe adatokat ugyan nem töltöttem fel, de az adatbázis tesztelésére, bemutatására és a lekérdezés szemléltetésére így is alkalmas. Ennek menetét és eredményeit mutatom be a következő alfejezetekben.

5.1 Minta adattáblák és attribútumaik

A *pgAdmin 4* segítségével létrehozott adattáblákat, azok attribútumaival, típusuk megjelölésével és rövid leírásukkal a következő oldalakon található táblázatok tartalmazzák.

Metadata			
	Metadata_ID	integer	Adat egyedi azonosítója, elsődleges kulcs
	Metadata_name	text	Adat neve
	Metadata_coordinate_X	float	Adat x koordinátája
	Metadata_coordinate_Y	float	Adat y koordinátája
	Metadata_coordinate_Z	float	Adat z koordinátája
	Metadata_date	date	Adat rögzítési dátuma
	Metadata_url	text	Adat hozzáférési URL
	Metadata_SourceData	boolean	Annak megjelölése, hogy az adat forrásadat-e vagy már egy adatfeldolgozás eredménye
	Metadata_description	text	Adat leírása
	Source_ID	text	Forrás egyedi azonosítója, idegen kulcs
	Type_ID	integer	Adattípus egyedi azonosítója, idegen kulcs
Data_type			
	Type_ID	integer	Adattípus egyedi azonosítója, elsődleges kulcs
	Type_name	text	Adattípus neve
	Type_description	text	Adattípus leírása
Source			
	Source_ID	integer	Forrás egyedi azonosítója, elsődleges kulcs
	Source_name	text	Forrás neve
	Source_description	text	Forrás leírása
	Source_url	text	Forrás URL címe
	Source_email	text	Forráse-mail címe

5.1. ábra Adatokkal kapcsolatos táblák adatai

Keywords

<u>Keyword_ID</u>	integer	Keresőszó egyedi azonosítója, elsődleges kulcs
Keyword_type_ID	integer	Keresőszó típus egyedi azonosítója, idegen kulcs
Keyword_name	text	Keresőszó neve
Keyword_description	text	Keresőszó leírása
Keyword_date_created	date	Keresőszó létrehozásának dátuma
Keyword_date_modified	date	Keresőszó módosításának dátuma
Metadata_ID	integer	Adat egyedi azonosítója, idegen kulcs

Keyword_type

<u>Keyword_type_ID</u>	integer	Keresőszó típus egyedi azonosítója, elsődleges kulcs
Keyword_type_name	text	Keresőszó típusának neve
Keyword_type_description	text	Keresőszó típusának leírása

Search

<u>Search_ID</u>	integer	Lehívás egyedi azonosítója, elsődleges kulcs
Search_name	text	Lehívás neve
Search_datetime	datetime	Lehívás időpontja (dátum és időpont)
Search_description	text	Lehívás leírása
Keyword_ID	integer	Keresőszó egyedi azonosítója, idegen kulcs
User_authorization_ID	integer	Felhasználói jogosultság egyedi azonosítója, idegen kulcs

5.2. ábra Lehívással és keresőszavakkal kapcsolatos táblák adatai

User

<u>User_ID</u>	integer	Felhasználó egyedi azonosítója, elsődleges kulcs
User_firstname	text	Felhasználó keresztnéve
User_middlename	text	Felhasználó középső neve
User_surname	text	Felhasználó vezetéknéve
User_username	text	Felhasználó felhasználónéve
User_password	text	Felhasználó belépési jelszava
User_email	text	Felhasználó e-mail címe
User_registration_date	date	Felhasználó regisztrálásának dátuma
User_activation	boolean	Annak megjelölése, hogy aktív-e a felhasználó (igen/nem)

UserGroup

<u>UserGroup_ID</u>	integer	Felhasználói csoport egyedi azonosítója, elsődleges kulcs
UserGroup_name	text	Felhasználói csoport neve
UserGroup_description	text	Felhasználói csoport leírása
User_ID	integer	Felhasználó egyedi azonosítója, idegen kulcs
UserGroup_Country_ID	integer	Felhasználói csoporthoz tartozó ország egyedi azonosítója, idegen kulcs

5.3. ábra Felhasználóval és jogosultságokkal kapcsolatos táblák adatai (1)

UserGroup_Country

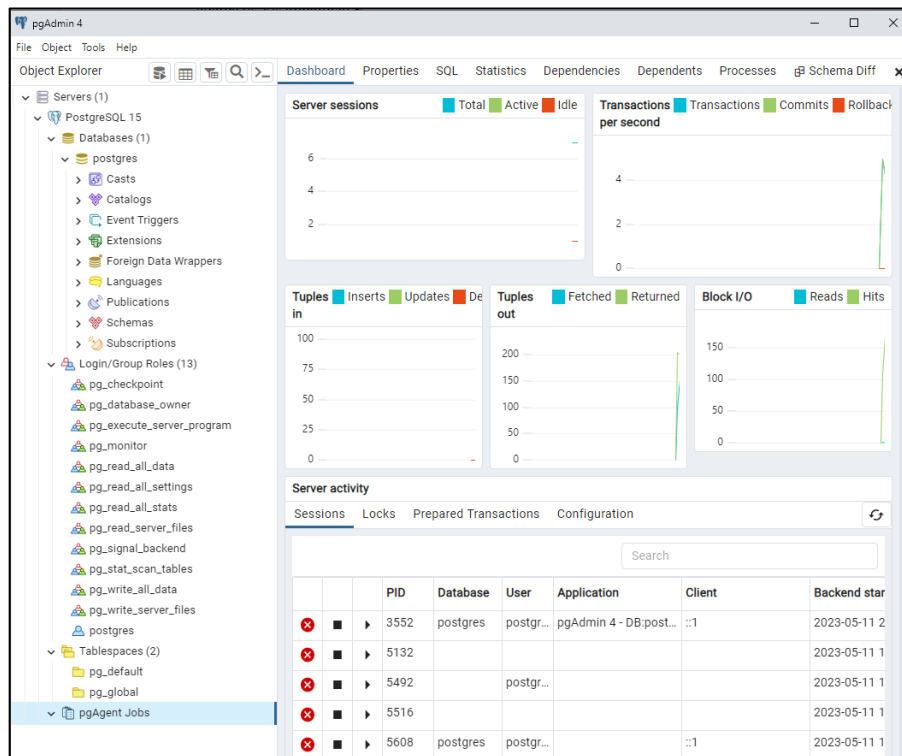
<u>UserGroup_Country_ID</u>	integer	Felhasználói csoporthoz tartozó ország egyedi azonosítója, elsődleges kulcs
UserGroup_Country_name	text	Felhasználói csoporthoz tartozó ország neve
UserGroup_Country_Institution	text	Felhasználói csoporthoz tartozó ország intézménye
UserGroup_Country_description	text	Felhasználói csoporthoz tartozó ország leírása

User_authorization

<u>User_authorization_ID</u>	integer	Felhasználói jogosultság egyedi azonosítója, elsődleges kulcs
User_authorization_description	boolean	Felhasználói jogosultság leírása
User_authorization_CreateUsers	boolean	Felhasználót létrehozhat (igen/nem)
User_authorization_ActivateUsers	boolean	Felhasználót aktiválhat (igen/nem)
User_authorization_DeactivateUsers	boolean	Felhasználót deaktiválhat (igen/nem)
User_authorization_UpdateUsers	boolean	Felhasználói adatokat frissíthet (igen/nem)
User_authorization_DeleteUsers	boolean	Felhasználót törölhet (igen/nem)
User_authorization_ViewData	boolean	Adatot lehívhat (igen/nem)
User_authorization_DownloadData	boolean	Adatot letölthet (igen/nem)
User_authorization_UploadData	boolean	Adatot feltölthet (igen/nem)
User_authorization_ModifyData	boolean	Adatot módosíthat (igen/nem)
User_authorization_DeleteData	boolean	Adatot törölhet (igen/nem)

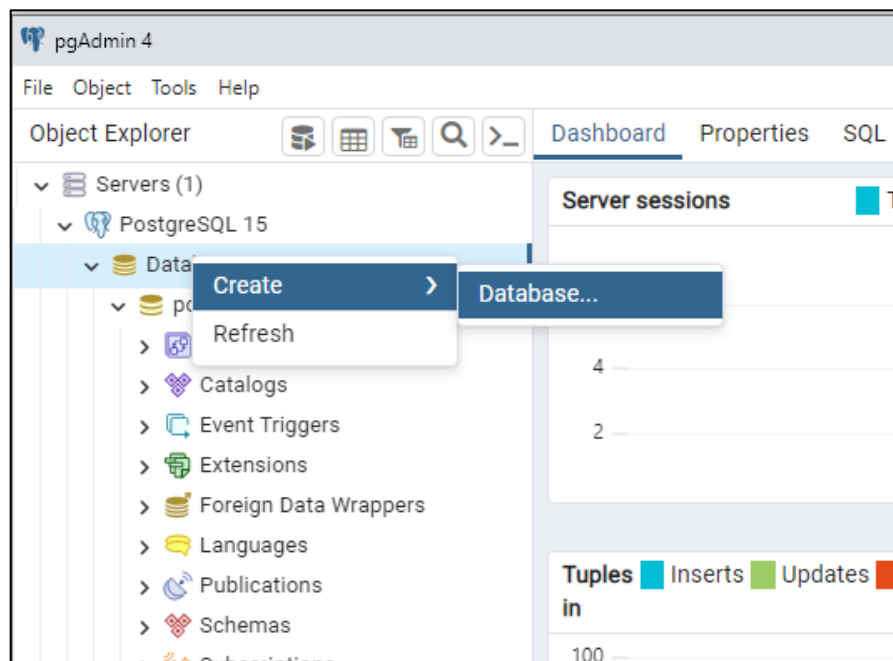
5.4. ábra Felhasználóval és jogosultságokkal kapcsolatos táblák adatai (2)

Az adattáblák létrehozásához a *pgAdmin 4* szoftverében a kezelőfelület főlapján a fejlécen található File, Object, Tools, Help menüpontok közül választhatunk az egyes funkciók eléréséhez. A főlap három részre tagolódik, a fejlécre, a bal oldali menüre és a jobb oldali munkafelületre.



5.5. ábra pgAdmin 4 — Kezelőfelület

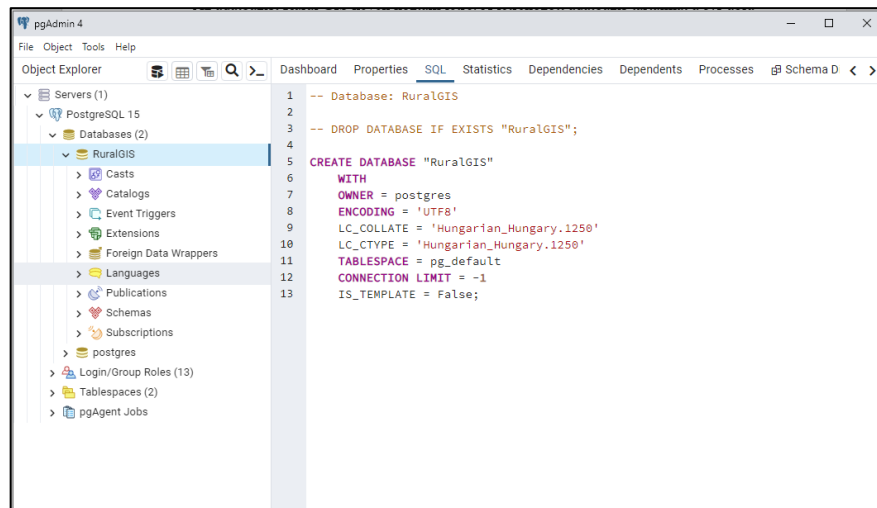
A munka megkezdése előtt fel kell csatlakoznunk a szerverre. Az új adatbázis létrehozása a Databases/Create/Database útvonalon érhető el.



5.6. ábra pgAdmin 4 — Adatbázis létrehozásának elérési útja

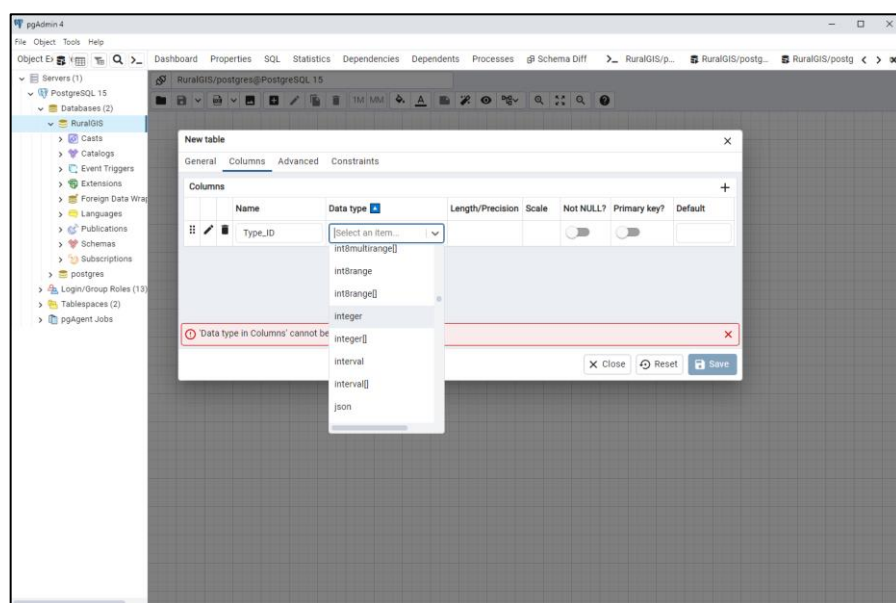
Az adatbázist Rural GIS néven hoztam létre. A létrehozott adatbázis tartalmát (bal oldali menü) és az SQL kódot (jobb oldali munkafelület) az 5.7 ábra mutatja.

Az adatbázis létrehozható importálással is, .bin, .csv és .txt formátumok bármelyikén.

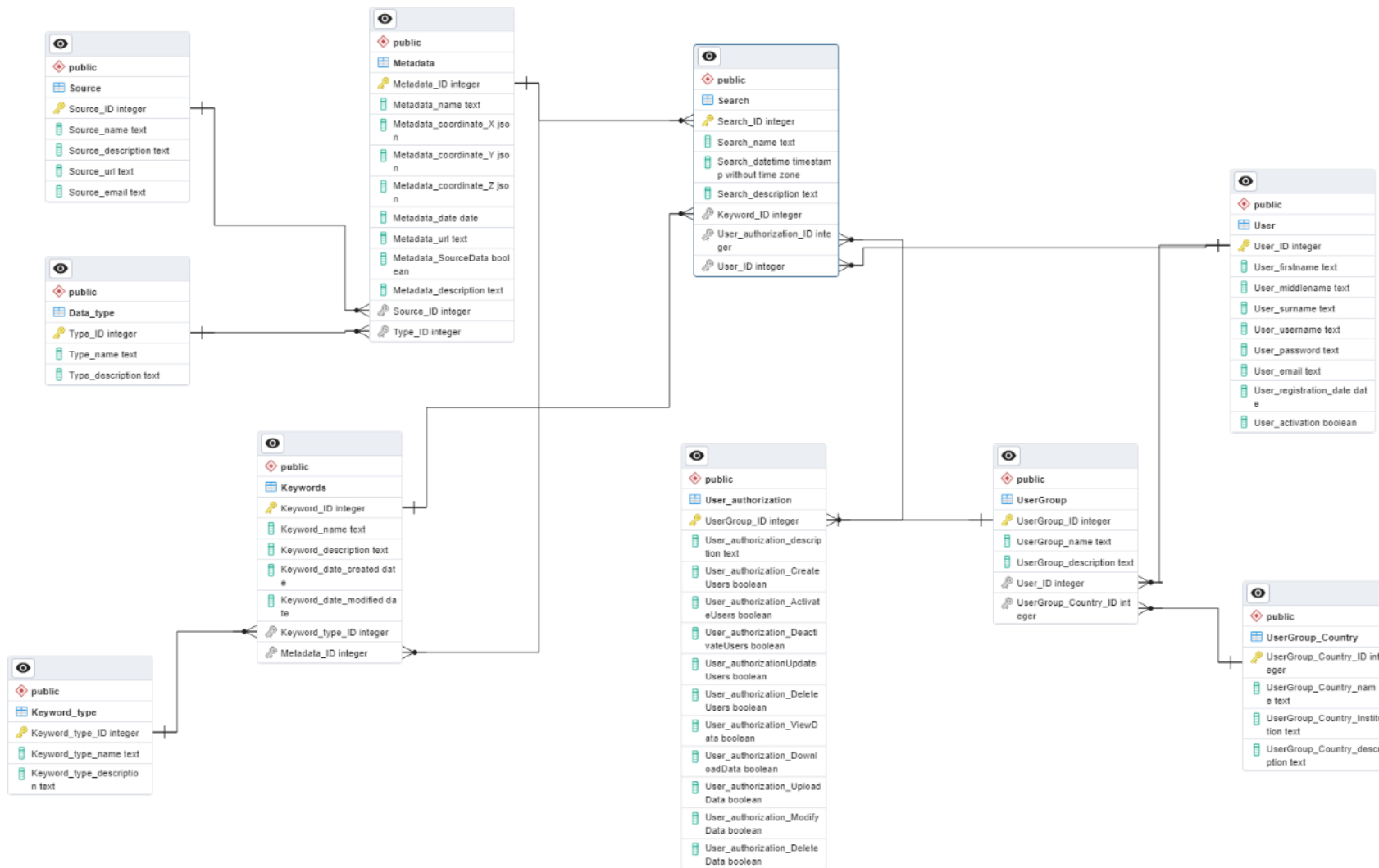


5.7. ábra pgAdmin 4 — RuralGIS adatbázis létrehozása

Ezt követően az adattáblák létrehozása az adattáblák hozzáadásával folytatódott, melyek nevét a 'General' fülön lehet megadni, míg az attribútumokat a 'Columns' fülön lehet definiálni (pl. név, adattípus, kulcs, NULL érték definiálásával). Az elsődleges és idegen kulcsok külön definiálhatóak a 'Constraints' fülön.

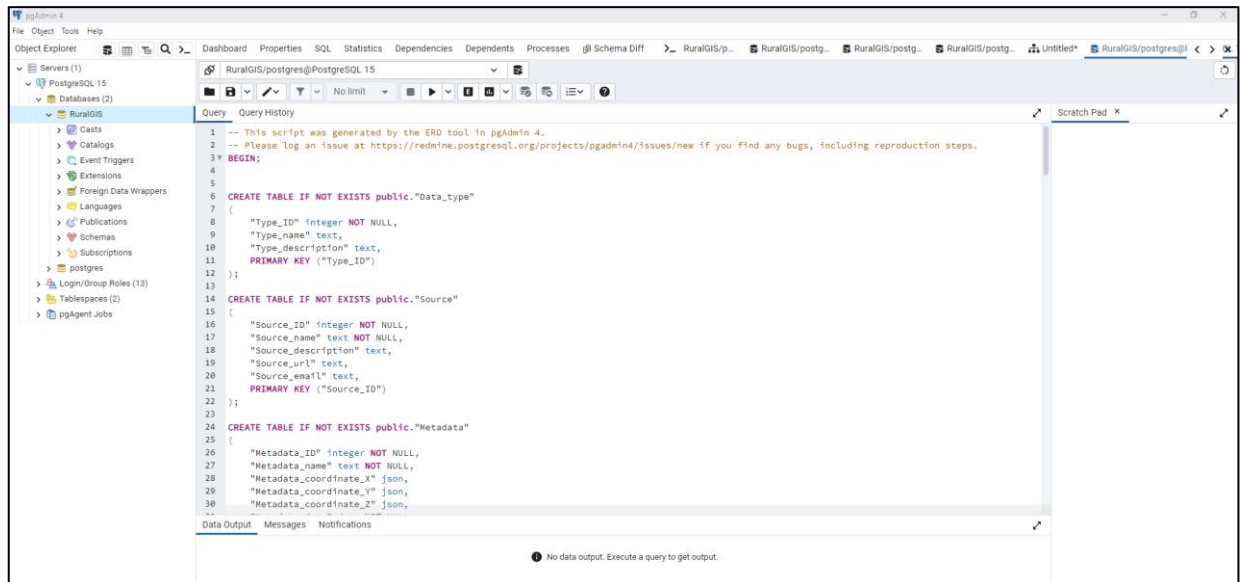


5.8. ábra pgAdmin 4 — Adattábla hozzáadása az E/R diagramhoz



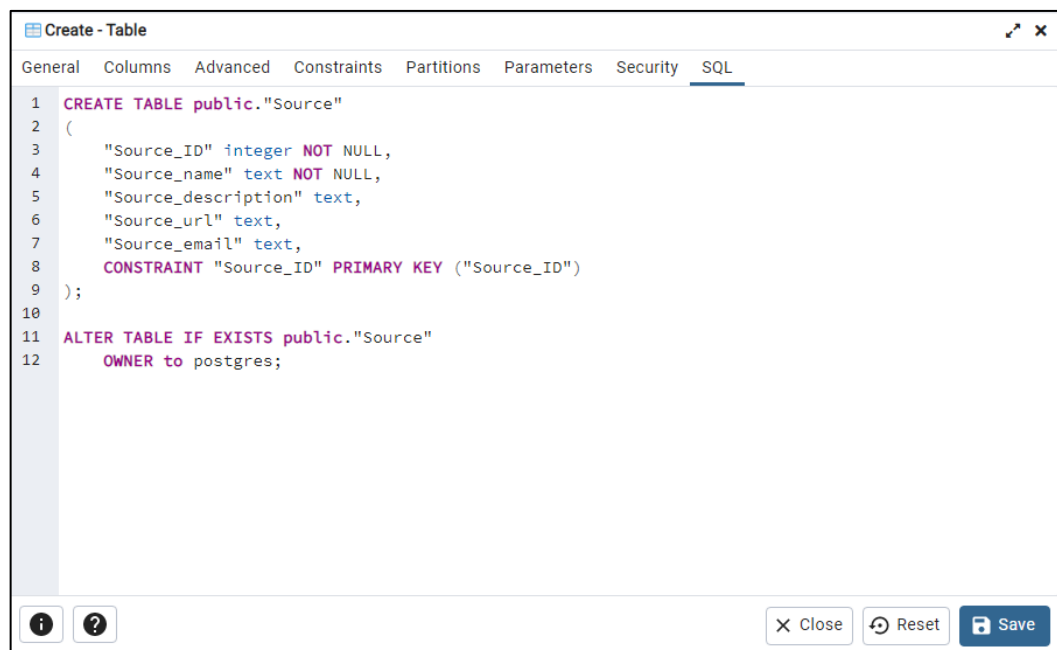
5.9. ábra pgAdmin 4 — E/R diagram

A létrehozott (vagy importált) adatbázisokhoz az E/R diagram is megtekinthető a program segítségével, valamint egy gombnyomásra legenerálható a létrehozott E/K diagramból az SQL script. Ennek eredményét az 5.10 ábra mutatja.



5.10. ábra pgAdmin 4 — SQL script

Táblákat nemcsak az E/R diagramból hozhatunk létre, hanem egyenként is megadhatjuk adataikat. (5.11 ábra)



5.11. ábra pgAdmin 4 — 'Source' tábla létrehozása

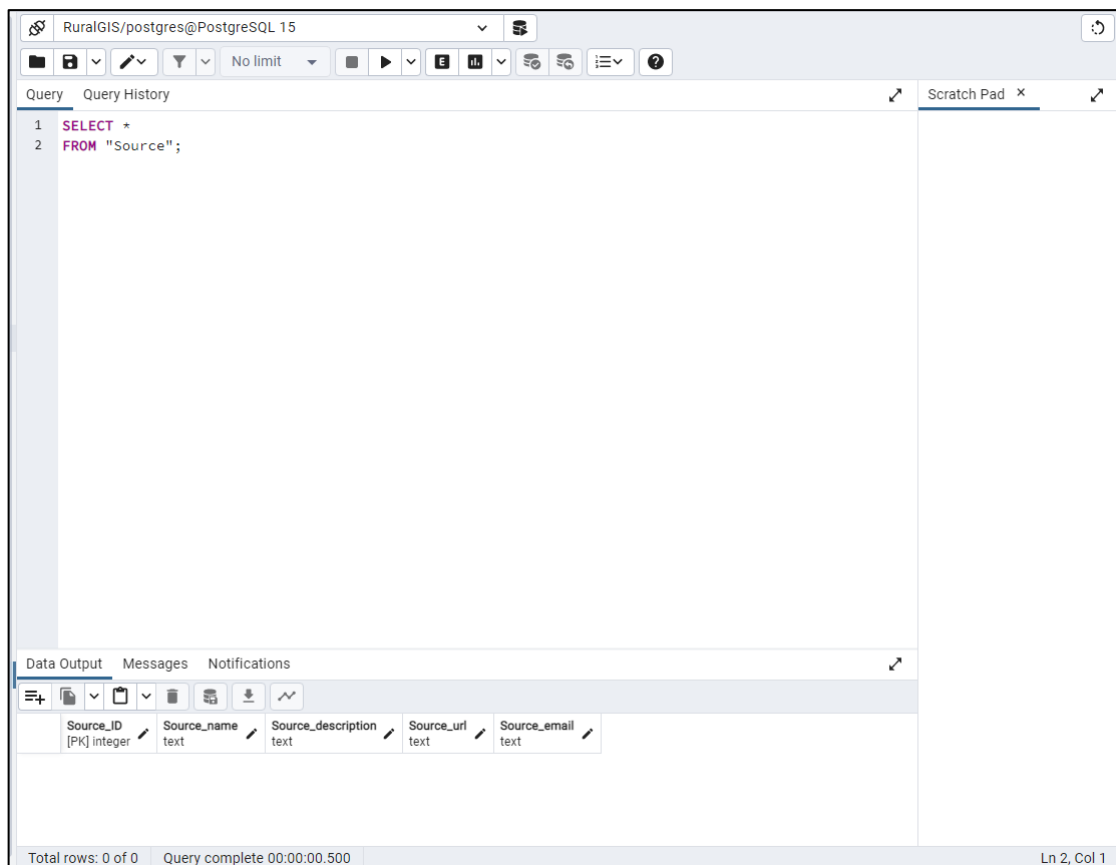
5.2 Lekérdezés minták

Az adatbázisba ugyan adatokat nem töltöttem fel, néhány egyszerű lekérdezéssel szemléltetem, hogyan kereshető az adatbázis.

Lekérdezés a 'Source' tábla adataira (összesre):

```
SELECT *
FROM "Source";
```

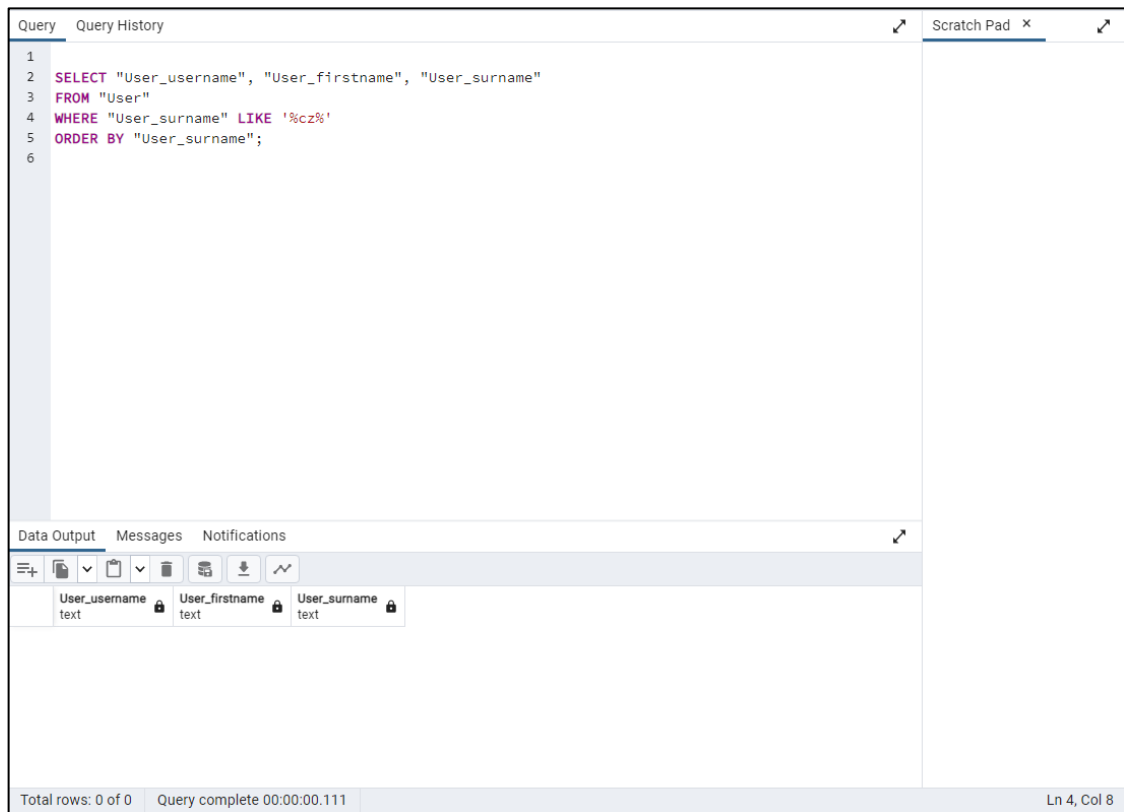
Ennek eredményét a 5.12 ábra mutatja. Mivel az adatbázisom nincs adatokkal feltöltve, a 'Data Output' mezőben csupán az attribútumok oszlopai jelennek meg, alattuk nincs adat listázva.



5.12. ábra pgAdmin 4 — 'Source' tábla adatainak lekérdezése

A User tábla azon felhasználók vezeté- és keresztnévét, valamint felhasználónevét listáztam vezetéknévük alapján sorba rendezve az 5.13 ábra szerint, akiknek a vezetéknévben szerepel a 'cz' betűkombináció.

```
SELECT "User_username", "User_firstname", "User_surname"  
FROM "User"  
WHERE "User_surname" LIKE '%cz%'  
ORDER BY "User_surname";
```



5.13. ábra pgAdmin 4 — Felhasználók listázása a User táblából

6. ÉRTÉKELÉS, TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A feladat ismertetését követően, annak értelmezéséből és saját eddigi ismereteim korlátait figyelembe véve — és az irodalomkutatás során új ismeretekkel gazdagodva — legjobb tudásomnak megfelelően megalkottam egy E/K diagramot, melynek relációs adatbázis-sémába történő konvertálása, illetve többszöri módosítása (normalizálás eredményeképp) alakította ki a terv jelenlegi formáját.

Véleményem szerint a kialakított adatbázis-terv az ismertetett feladat alapkövetelményeinek megfelel, tartalmazza a felhasználói jogosultságkezelést (hozzáférési és letöltési lehetőséget is), az adattípusok és forrásaik definiálását, valamint a kulcsszavas keresés lehetőségét.

Az adatbázis-terv továbbfejlesztési lehetőségeinek csupán a fantáziánk szab határt, a valóságban azonban elsősorban a felhasználók, üzemeltetők igényei határozzák meg a további fejlesztési irányokat.

A lehetséges továbbfejlesztési irányok közül — a teljesség igénye nélkül — kiemelnék néhányat, melyet én fontosnak vagy hasznosnak tartanék.

A modell bővíthető további attribútumokkal — például a beérkező adatok pontosabb megismerésével és specifikálásával lehetőségünk van részletesebb attribútumtáblákat megadni.

A modell egy másik, lehetséges továbbfejlesztési lehetősége a felhasználói jogosultságok önkéntes regisztrációs sémájának kialakítása a jelenlegi adminisztrátori jogosultság-kiosztási lehetőséggel szemben. Ez egyrészt egyszerűsítene új felhasználók becsatlakozását, azonban ellenőrzésüket és az üzemeltetést bonyolítaná.

Természetesen új felhasználói csoportok megadására, a meglévők szétbontására is lehetőség van (a felhasználók jogosultságainak és valós igényeinek pontosítása mellett). Itt megvalósítható lenne például a felhasználói jogosultságok szűkítése adattípusokra vagy adatmezőkre (kizárólagosság).

Jelszó-émlekeztető lekérésének és felhasználó felé történő, e-mail-en való továbbításának lehetősége is beépíthető az adatbázis-tervbe.

7. ÖSSZEFOGLALÓ

Szakedolgozatom témájaként egy egyetemi projektet támogató geoinformatikai webalkalmazás háttéréül szolgáló adatbázis megtervezését választottam, PostgreSQL környezetben.

A nemzetközi egyetemek közötti együttműködés alapjául szolgáló projekt célja a vidéki környezet modellezésének kialakítása, és ehhez kapcsolódóan geoinformatikai és távérzékelési adatok cseréje. A projekt célul tűzi ki az egyes modellek összehasonlítását a különböző országok és városaik esetében.

A projekt feladatai közé tartozik a távérzékelésből, attribútum adatokból és meglévő térképekből történő adatgyűjtést követően az adatbázis megtervezése és felépítése, majd az adatok online grafikai megjelenítése a projekt webalkalmazásának segítségével. Az e mögött álló adatbázis megtervezése volt szakedolgozatom témája.

A tervek alapján elkészített adatbázisnak alkalmasnak kell lennie különböző forrásból származó, eltérő formátumú adatok kulcsszavas lehívására, valamint a projekt eredményeinek szemléletes bemutatására a felhasználói jogosultságok kezelése mellett.

Az adatbázisok, geoinformatikai adatbázisok áttekintése, valamint sajátosságainak megismerését követően az adatbázisstervezés módszertanát és lépéseit tekintettem át, a rendelkezésemre álló információk alapján az adatbázis-terv elkészítéséhez összeállítottam a specifikációt, megterveztem az E/R diagramot (több, rendelkezésre álló diagramszerkesztő használatát bemutatva), melyből a relációséma normalizálásával létrehoztam az adatbázis tervét. A terv alapján a *pgAdmin 4* segítségével PostgreSQL adatbázist hoztam létre és néhány lekérdezést a terv tesztelése céljából.

Megvizsgáltam a lehetséges továbbfejlesztési irányokat — véleményem szerint a modell részletesebb attribútumtáblákkal további attribútumokkal bővíthető, emellett megfontolandó a felhasználói jogosultságok önkéntes regisztrációs sémájának kialakítása (a jelenlegi adminisztrátori jogosultság-kiosztási lehetőséggel szemben), illetve szükség esetén a modell új felhasználói csoportok definiálása vagy jelszó-emlékeztető lekérésének és felhasználó felé történő, e-mail-en való továbbításának lehetősége is beépíthető az adatbázis-tervbe.

8. SUMMARY

As the topic of my thesis, I chose the design of a database in the PostgreSQL environment that serves as the background for a geoinformatics web application supporting a university project.

The goal of the project, which is the basis of cooperation between international universities, is to develop the modeling of the rural environment and, in connection with this, to exchange geoinformatics and remote sensing data. The project aims to compare the individual models for different countries and their cities.

The tasks of the project include the design and construction of the database after collecting data from remote sensing, attribute data and maps, and then the online graphic map of the data with the help of the project's web application. The design of the database behind this was the topic of my thesis.

The database created on the basis of the plans must be suitable for keyword retrieval of data from different sources in different formats, as well as for the visual presentation of the project's results in addition to the management of user rights.

Overview of databases and geoinformatics databases, as well as getting to know their peculiarities, I reviewed the methodology and steps of the following database design, based on the information available to me, I compiled the specification for the preparation of the database design, I designed the E/R diagram (several available diagram editors are presented), from which I created the database plan by normalizing the relational schema. Based on the plan, I created a PostgreSQL database using pgAdmin 4 and some queries to test the plan.

I examined the possible directions for further development — in my opinion, the model can be expanded with additional attributes with more detailed attribute tables, the creation of a voluntary registration scheme for authorizations should be considered (as opposed to the current administrative authorization allocation option), and, if necessary, the model defining new user groups or password- the option of retrieving a reminder and sending it to users by e-mail can also be included in the database design.

9. IRODALOMJEGYZÉK

- [1] B. Halassy, „Az adatbázistervezés alapjai és titkai,” 1994.
- [2] Jeffrey D. Ullmann és Jennifer Widom, Adatbázisrendszerek (Alapvetés), Budapest: Panem, 2009.
- [3] S. Iváncsy és I. Vajk, Informatika 2, Adatbázisok, BME Automatizálási és Alkalmazott Informatikai Tanszék, 2007.
- [4] L. Mucsi, F. Kovács, J. Szatmári és L. Nagyvárad, „Szegedi Tudományegyetem Geoinformatikai, Természet- és Környezetföldrajzi Tanszék,” 02 11 2011. [Online]. Available: <http://www.geo.u-szeged.hu/~laci/ab-Geoinformatanyag/geoinformatika-alapjai.html>. [Hozzáférés dátuma: 24 04 2023].
- [5] J. dr. Szelezsán, Adatbázisok, Gábor Dénes Főiskola, Oktatási segédlet.
- [6] L. Gugolya, Szerző, *Adatbázisok kezelése*. [Performance]. Óbudai Egyetem AMK, 2022.
- [7] Z. Siki, „BME Építőmérnöki Kar, Általános- és Felsőgeodézia Tanszék,” [Online]. Available: <http://www.agt.bme.hu/szakm/adatb/adatb.htm>. [Hozzáférés dátuma: 24 04 2023].
- [8] S. Varma, Szerző, *Network Model in DBMS / DBMS Course*. [Performance]. Scaler Topics, 2023.
- [9] S. Varma, Szerző, *What is a RDBMS (Relational Database Management System)?*. [Performance]. Scaler Topics, 2023.
- [10] B. Tari, „Adatbáziskezelés,” Educatio Társadalmi Szolgáltató Nonprofit Kft, 31 05 2019. [Online]. Available: <http://www.inf.u-szeged.hu/~tarib/adatbaziskezeles>. [Hozzáférés dátuma: 24 04 2023].

- [11] „PostgreSQL,” The PostgreSQL Global Development Group, [Online]. Available: <https://www.postgresql.org/>. [Hozzáférés dátuma: 25 04 2023].
- [12] „dbForge Studio for PostgreSQL,” Devart, [Online]. Available: <https://www.devart.com/>. [Hozzáférés dátuma: 25 04 2023].
- [13] M. Cserép és R. Giachetta, „Eötvös Loránd Tudományegyetem, Informatikai Kar,” 2017. [Online]. Available: https://mcserep.web.elte.hu/data/education/2017-2018-2_TTAF/elte_ttaf_data-formats.pdf. [Hozzáférés dátuma: 25 04 2023].
- [14] Kiss, „ELTE Informatikai Kar (ELTE IK),” [Online]. Available: <https://people.inf.elte.hu/kiss/08abe1/08ab1e02.pdf>. [Hozzáférés dátuma: 24 04 2023].
- [15] Timár & Társai, Építsünk könnyen és lassan adatmodellt!, Veszprém: Veszprémi Egyetem, 1997.
- [16] M. Czenky, Adatmodellezés - SQL és Access alkalmazás, Budapest: ComputerBooks, 2002.
- [17] „Centroszet,” [Online]. Available: <http://centroszet.hu/tananyag/adatbazis>. [Hozzáférés dátuma: 24 04 2023].
- [18] „PostgreSQL Database Development Tools,” Devart, [Online]. Available: <https://www.devart.com/>. [Hozzáférés dátuma: 25 04 2023].

10.ÁBRAJEGYZÉK

2.1. ábra Az adatbázis-kezelés rendszere és résztvevői [3]	4
2.2. ábra Leggyakrabban használt adatformátumok a térinformatikában [8]	12
3.1. ábra Az adatbázis modellezés és implementálás eljárásának lépései [10].....	15
3.2. ábra Az egyes adatmodellek összevetése [4]	19
4.1. ábra A Figma platform kezelőfelülete (www.figma.com).....	32
4.2. ábra A Figma platform menürendszere (www.figma.com).....	33
4.3. ábra A Figma platform alsó szerkesztősávja (www.figma.com).....	33
4.4. ábra A Dia Diagram Editor szoftver kezelőfelülete.....	34
4.5. ábra Adattábla létrehozása a Dia Diagram Editor szoftverrel	35
4.6. ábra Hivatkozások szerkesztése a Dia Diagram Editor szoftverrel	35
4.7. ábra Diagramfa az elkészült E/K diagramra (Dia Diagram Editor szoftverrel)	36
4.8. ábra E/K modell a Figma applikációjával szerkesztve	37
4.9. ábra E/K modell a Dia Diagram Editor szoftverrel megszerkesztve	38
5.1. ábra Adatokkal kapcsolatos táblák adatai	41
5.2. ábra Lehívással és keresőszavakkal kapcsolatos táblák adatai	42
5.3. ábra Felhasználóval és jogosultságokkal kapcsolatos táblák adatai (1)	43
5.4. ábra Felhasználóval és jogosultságokkal kapcsolatos táblák adatai (2)	44
5.5. ábra pgAdmin 4 — Kezelőfelület.....	45
5.6. ábra pgAdmin 4 — Adatbázis létrehozásának elérési útja	45
5.7. ábra pgAdmin 4 — RuralGIS adatbázis létrehozása	46
5.8. ábra pgAdmin 4 — Adattábla hozzáadása az E/R diagramhoz	46
5.9. ábra pgAdmin 4 — E/R diagram	47
5.10. ábra pgAdmin 4 — SQL script.....	48
5.11. ábra pgAdmin 4 — 'Source' tábla létrehozása.....	48
5.12. ábra pgAdmin 4 — 'Source' tábla adatainak lekérdezése	49
5.13. ábra pgAdmin 4 — Felhasználók listázása a User táblából.....	50

11.TÁBLÁZATOK JEGYZÉKE

1. táblázat — Az egyes adat(bázis)modellekre épülő adatkezelő rendszerek [1] [4] [5] .	7
2. táblázat — Leggyakrabban alkalmazott változótípusok a térinformatikában [12].....	9
3. táblázat — Leggyakrabban alkalmazott változótípusok a térinformatikában [3].....	11
4. táblázat — Online hozzáférhető E/K diagram tervezők	18
5. táblázat — Adatforrások specifikációja.....	28
6. táblázat — Jogosultság-kiosztási tábla	30