Bagó Boglárka (SI2JMR), Gábor Krisztián(),Vadi Gergő(), Tamási Miklós(O070E2), Hegedüs Alexandra(XM6YBD)

GDE

aDATBÁZISOK   
PROJEKTMUNKA  
 DOKUMENTÁCIÓ

íVKEMENCE ADATAINAK FELDOLGOZÁSA,

2024. szeptember 28.

Tartalom

[I. Bevezetés 3](#_Toc181098112)

[1.1 Csapat bemutatása 3](#_Toc181098113)

[1.2 Szerepkörök magyarázata 3](#_Toc181098114)

[2. Lépések 5](#_Toc181098115)

[3.Téma ismertetése 7](#_Toc181098116)

[3.1 Témakör 7](#_Toc181098117)

[3.2 Projekt adatai 8](#_Toc181098118)

[3.3 Projekt célja 8](#_Toc181098119)

[4. Funkciók ismertetése 10](#_Toc181098120)

[II.A munkafolyamat részletei 11](#_Toc181098121)

[1.Adatfeldolgozás excellel és szabadszemmel 11](#_Toc181098122)

[Adatok és diagramok kiértékelése 12](#_Toc181098123)

[2.Relációs adatbázis fontossága a projektnél 14](#_Toc181098124)

[3.Adatbázis megtervezése ER diagrammal 15](#_Toc181098125)

[3.1 Definíció 15](#_Toc181098126)

[3.2 Az aktuális entitások és attribútumok meghatározásának lépései 15](#_Toc181098127)

[4.Adatok feltöltése, ellenőrzése 18](#_Toc181098128)

[5.Adatbázis létrehozása 19](#_Toc181098129)

[5.1 DCL és TCL: SQLite-n belül nincs jogosultságkezelés. a. Írják le, hogy melyik résztvevőnek milyen jogosultságra lett volna szüksége, a munkájához. 19](#_Toc181098130)

[5.2 b. Vizsgálják meg, hogy van-e, és ha van, akkor milyen esetben tranzakciókezelésre. 19](#_Toc181098131)

[5.3 1 tranzakciót tegyenek bele akkor is, ha nem feltétlen szükséges csak a példa kedvéért. 19](#_Toc181098132)

[6.Lekérdezések 20](#_Toc181098133)

[6.1 Végezzenek el lekérdezéseket, melyek összecsatolnak táblákat (JOIN). 20](#_Toc181098134)

[6.2. Használjanak aggregációs függvényeket. 20](#_Toc181098135)

[6.3. Írják le, hogy miért végezték el ezeket a műveleteket, és milyen következtetésekre jutottak. 20](#_Toc181098136)

[7. Allekérdezések, UNION 21](#_Toc181098137)

[7.1 Szükség esetén készítsenek egymásba ágyazott lekérdezéseket 21](#_Toc181098138)

[7.2 A részeredményeiket INSERT és SELECT segítségével tegyék ki külön táblákba 21](#_Toc181098139)

[7.3. Vizsgálják meg, hogy mit lehetne kimutatni, vagy ellenőrizni az UNION utasításokkal 21](#_Toc181098140)

[8. Teljesítmény optimalizálás 22](#_Toc181098141)

[8.1 Melyik mezőre milyen kulcsot, indexet alkalmaznának, és miért? 22](#_Toc181098142)

[8.2 . Milyen módszert javasolnának az adatműveletek gyorsítására, ha megkapnának 10 évnyi adatot és azt kellene feldolgozni? 23](#_Toc181098143)

[9. Adattisztítás. Az adatok hibával terheltek 25](#_Toc181098144)

[9.1 Milyen módszerrel, illetve logika alapján kerestek, szűrtek hibás adatokat? 25](#_Toc181098145)

[9.2. Milyen módszerrel kezelték azokat? 25](#_Toc181098146)

[10. Milyen tudásuk alapján tudták azonosítani a hibákat? 26](#_Toc181098147)

[10.1 Mi alapján tekintették a hibás adatokat hibás adatnak? 26](#_Toc181098148)

[10.2 Adatbevitelhez milyen érvényesítési szabályokat, milyen kényszereket javasolnának 26](#_Toc181098149)

[11. Az adatok egy ipari adatgyűjtő program írja az adatbázisba. 27](#_Toc181098150)

[12. Biztonsági mentésre milyen stratégiát választanának? 28](#_Toc181098151)

[III.Összegzés 29](#_Toc181098152)

# I. Bevezetés

## 1.1 Csapat bemutatása

|  |  |  |  |
| --- | --- | --- | --- |
| Szerepkör | Név | NeptunKód | Szak |
| **Lead Developer** | Gábor Krisztián |  | Mérnökinfo |
| **Data Engineer** | Gergő |  | Mérnökinfo |
| **Data and Business Analyst** | Tamási Miklós | O070E2 | Mérnökinfo |
| **Sysadmin** | Bagó Boglárka | SI2JMR | GazdInfo |
| **Project Manager** | Hegedüs Alexandra | XM6YBD | GazdInfo |

## 1.2 Szerepkörök magyarázata

Az első találkozónkon megpróbáltuk a fentiek alapján felosztani a különböző felelősségi köröket, azonban mivel ez egy kisebb volumenű projekt, a csapattagok a saját hatáskörükön túl is foglalkoztak a projekt különböző részeivel.

1. **Projektvezető (Project Manager)**

A projektvezető felelős az egész projekt szervezéséért és koordinálásáért. Feladata, hogy irányítsa a csapat munkáját, figyelemmel kísérje a projekt haladását, és biztosítsa, hogy mindenki időben elvégezze a feladatát. Emellett ő tartja a kapcsolatot az oktatókkal vagy a tanácsadókkal, és gondoskodik arról, hogy a projekt megfeleljen az előírt követelményeknek. Az ő feladata továbbá a határidők betartása, az erőforrások allokálása, és a felmerülő akadályok megoldása. A GitHub felületén létrehoztunk egy projektet, amelyhez egy repositoryt és egy boardot is készítettünk. A repositoryban nyomon követhetjük a kód alakulását a teljes mappastruktúrával együtt, míg a boardon áttekintést kapunk arról, hogy a csapattagok éppen mely feladaton dolgoznak, melyeket fejeztek be, és hol van esetleg szükségük segítségre.

2. **Adat- és üzleti elemző (Data and Business Analyst)**

Az adat- és üzleti elemző feladata az ívkemence működési adatai és üzleti szempontú követelményeinek elemzése. Ő vizsgálja meg, hogyan gyűjthetők össze az adatok, miként dolgozhatók fel, és hogyan használhatók fel ezek az információk a projekt céljainak eléréséhez. Elemzi az adatokat és meghatározza, hogy milyen mutatók fontosak a projekt sikeres teljesítéséhez. Az üzleti elemző felel továbbá az adatok értelmezéséért, valamint a jelentések és vizualizációk elkészítéséért, hogy a csapat többi tagja megfelelően láthassa az eredményeket.

3. **Vezető fejlesztő (Lead Developer)**

A vezető fejlesztő felelős az adatok feldolgozását végző rendszerek és szoftverek fejlesztéséért. Feladata az ívkemence adatainak kezeléséhez szükséges algoritmusok és kód megírása, illetve a technikai megvalósítás irányítása. Ő biztosítja, hogy az adatfeldolgozó szoftverek hatékonyan működjenek, és képesek legyenek a nagy mennyiségű adat kezelésére. Szorosan együttműködik az elemzővel és a projekt többi tagjával, hogy a technikai megoldások összhangban legyenek a projekt céljaival.

4. **Adminisztrátor (Sytem Administartor)**

Ez a szerepkör felelős az adatbázisok kezeléséért, karbantartásáért, valamint a hozzáférési jogosultságok beállításáért és felügyeletéért. A rendszergazda gondoskodik arról, hogy az adatokat biztonságosan kezeljék, és csak azok férhessenek hozzá, akik jogosultsággal rendelkeznek. Ide tartozik a felhasználói fiókok és szerepkörök létrehozása, az engedélyek beállítása, valamint az adatbiztonsági irányelvek betartása és frissítése.

5. **Adatmérnök (Data Engineer)**

Az adatmérnök feladata az ívkemence adatainak begyűjtése, tárolása és feldolgozása. Ő tervezi meg az adatok áramlását biztosító technikai infrastruktúrát, és felelős a különböző adatbázisok létrehozásáért, karbantartásáért, valamint azok integrációjáért. Az adatmérnök biztosítja, hogy az ívkemencéből származó nyers adatok hatékonyan feldolgozhatók legyenek, és hozzáférhetők a csapat többi tagja számára elemzés céljából. Szoros együttműködésben dolgozik a vezető fejlesztővel, hogy a fejlesztett rendszerek képesek legyenek kezelni a nagyméretű és komplex adathalmazokat. Ő gondoskodik arról is, hogy az adatok tiszták, konzisztens formátumúak és megfelelően strukturáltak legyenek az elemzési célok érdekében.

## 2. Lépések

A projekt lépései és a hozzájuk tartozó feladatok ütemtervvel való egybehangolása során minden szakaszt alaposan áttekintettünk. Amennyiben eltérések voltak a 1-es verzióhoz képest, azokat szekvenciálisan megjelöltük és korrigáltuk, ahol szükséges volt.

|  |  |  |  |
| --- | --- | --- | --- |
| Verzió | Dátum | Szerző | Tevékenység |
| 1. | 2024.10.08 | Mindenki | Megbeszélés, alapok lefektetése:   * Elemzés Excellel, * Kiindulópontok keresés, * Értelmezés |
| 1. | 2024.10.08 | HA | Dokumentáció írása , főbb pontok hozzáadása, ER Diagram 1.verzió |
| 1. | 2024.10.09 | VG | Adattisztítás, normalizálás |
| 1. | 2024.10.09 | HA | Projekt lépéseinek tervezése, megbeszélés részleteinek hozzáadása a dokumentumhoz |
| 2. | 2024.10.09 | HA | ER Diagram 2.verzió |
| 2. | 2024. 10. 11 | BB | Diagramok és kiértékelésük |
| 1. | 2024.10.14 | HA | Github porject inicializálása  Alap használati útmutató  Python DB táblák készítése |
| 1 | 2024.10.14 | BB,VG, HA | Github szinkronizálás, összekapcsolás PyCharmmal |
| 1. | 2024.10.16 | VG, HA | Py adatisztítás rész bővítése, Diagram elemzés átnézése |
| 1. | 2024.10.27 | Mindenki | Megbeszélés, további stratégia kidolgozása, az újonnan tanultak alapján pl: Index hozzáadása Github project tábla létrehozása, további feladatok elosztása a csapat tagok között |
| 1. | 2024.10.26 | VG, HA | Adattisztítás befejezése, .py fileok elrendezése |
| 1. | 2024.10.26 | GH | A projekt struktúrájának kialakítása, létező fájlok ellenőrzése, mergelése |
| 1. | 2024.10.27 | GH |
| 1. | 2024.10.27 | TM , VG | Kényszerek ellenőrzése, dokumentációja |
| 3. | 2024.10.27 | BB | Engedélyek dokumentációja |
| 1. | 2024.11.01 | TM | Grafikonok és statisztikai elemzések |
| 2. | 2024.11.02 | HA | Grafikonok mentése, statisztikai elemzések mergelése |
| 2. | 2024.11.03 | BB | Adatok törlése funkció beépítése |
| 3. | 2024.11.04 | GH | Funkciók mergelése, korrekciója |
| 4. | 2024.11.05 | HA | Dokumentáció |
| 1. | 2024.10.28 | Mindenki | Megbeszélés, további lépések felmérése |

## 3.Téma ismertetése

### 3.1 Témakör

Az elektromos ívkemence olyan kemence, amely elektromos ív segítségével fűti az anyagot. Az ipari ívkemencék mérete a körülbelül egy tonnás kapacitású, kis egységektől a másodlagos acélgyártáshoz használt 400 tonnás egységig terjednek. Az Electric Arc Kemence (EAF) egy acélkészítő kemence, amelyben acélhulladékot fűtik és olvad fel a kemenceelektródák és a fémfürdő között. A hűtőpanelek (hűtési elemek) az ívkemencékben olyan hűtőrendszerek, amelyek a kemence falait és egyéb forró részeit hűtik, hogy megakadályozzák a túlmelegedést és a szerkezeti károsodásokat. A panelekben csövekben áramoltatják a vizet, célja a kemence élettartamának növelése.

Egy ilyen kemence ára függ a nagyságától, kapacitásától, teljesítményétől…stb , de egészen magas, akár millió euros árat is elérhet. A minőség és a biztonság szempontjából is kulcsfontosságú a megfelelő szabályzás, ezért van szükség az adatok folyamatos értelmezésére , kontextusba helyezésére és vizsgálatára.

**Adagok**: Újra olvasztott acélhulladék.

**Hűtőpanelek**: A hűtést végző vízzel feltöltött segédrendszer.

### 3.2 Projekt adatai

Projekt neve: Ívkemence adatainak kezelése

Projekt kezdete: 2024.10.08

Projekt elvárt befejezési ideje:2024.11.17

Projekt tényleges befejezési ideje:2024.11.06

Projekt leadási ideje: 2024.11.07

Kommunikációhoz használt felületek, keretrendszerek, programok: GitHub, PyCharm, Discord

Kapott adatok: Adagok.csv, Hűtőpanelek.csv

1. ábra cím: ELEKTROMOS ÍVKEMENCE

  
Forrás megjelölés: Facebook, Fémipari Szakember- Steel Worker csoport

### 3.3 Projekt célja

Az adatok feldolgozása Python és SQL használatával, CSV formátumú fájlokból. Adatok létrehozása, lekérdezése, módosítása és törlése. Ezen folyamatok során kiemelt jelentőséget kap az adattisztítás, az adatok integritásának biztosítása, valamint az anomáliák és hibák detektálása és javítása, továbbá a trendek és mintázatok disszonanciáinak feltárása.(két egyforma folyamat nincsen, de jellegében kevés eltérést mutasson)

A folyamat főbb lépései a következők:

**Adatbevitel és betöltés**: Az adatok importálása CSV fájlokból Python segítségével, mely során a kapott struktúrák SQL adatbázisba kerülnek. Ez a lépés magában foglalhatja az adatok előfeldolgozását, például formátumkonverziókat, hiányzó értékek kezelését, valamint típuskonzisztenciák ellenőrzését.

**Adatlekérdezés és manipuláció**: SQL segítségével specifikus lekérdezések futtatása az adatbázisban, hogy előre meghatározott kritériumok alapján kinyerhessük a releváns információkat. Az adatok módosítása (szerkesztése) során az adatintegritás fenntartása kritikus fontosságú, míg az adatok törlése gondos elemzés után történik, hogy elkerüljük a hasznos információk elvesztését.

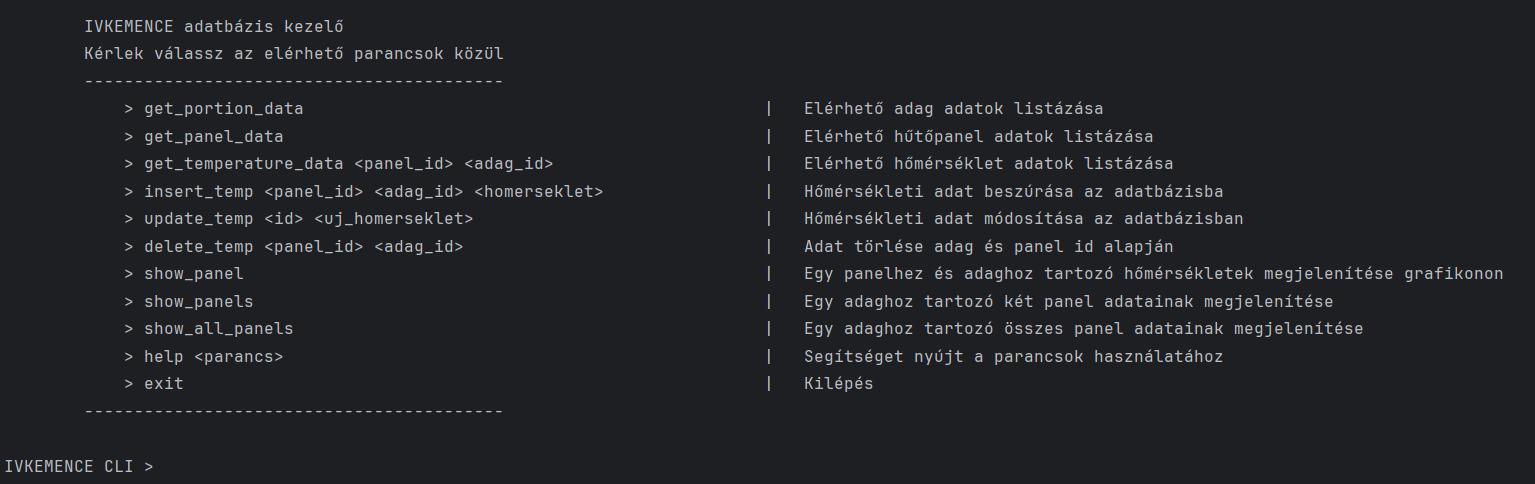
**Anomáliák és hibák kiszűrése**: Az adatok elemzése során matematikai és statisztikai módszerekkel azonosíthatók az anomáliák és hibák. Ez magában foglalja a hibás vagy szokatlan adatok detektálását, például statisztikailag szignifikáns eltérések feltárását, amelyek hibás adatgyűjtésre, mérési hibákra, vagy szabálytalanságokra utalhatnak.

**Trend disszonanciák elemzése**: A kapott adatok trendjeinek vizsgálata, különös tekintettel a hosszú távú mintázatokra és azok időbeli alakulására. A disszonanciák feltárása során figyelembe vesszük a megszokott trendektől való eltéréseket, melyek mögöttes okainak statisztikai módszerekkel történő azonosítása hozzájárul a döntéshozatali folyamatokhoz. A trend disszonanciák előfordulása utalhat váratlan külső tényezőkre, rendszerszintű hibákra vagy nem szándékolt változásokra az adatgyűjtési folyamatban.

Az egész folyamat célja az adatok konzisztenciájának és pontosságának biztosítása, amely nélkülözhetetlen az eredmények tudományos értelmezéséhez, valamint az adatok megbízható felhasználásához döntéstámogató rendszerekben.

## 4. Funkciók ismertetése

Elsődleges célunk az adatok tisztítása és ellenőrzése. Ebben a kezdeti szakaszban az adatokat megvizsgáljuk, javítjuk (.csv fájlok), a javított verziót elmentjük, létrehozzuk az adatbázist (.db fájl) valamint statisztikai módszerekkel elemzéseket és számításokat végzünk és szükség esetén töröljük a hibás adatokat.(.txt fájl) Ez az alapos előkészítés segít biztosítani az adataink megbízhatóságát és minőségét. Az itt generált fájlokat mind az db\_creator/output mappába mentjük ,ezeket nem mentjük el a githubon. A fájlokat a db\_creator/main.py fájl futtatásával hozhatjuk létre.

A továbbiakban az alapvető adatkezelési funkciókat építettük be, beleértve az adatok lekérdezését, bevitelét, módosítását és törlését is. A felhasználó ezeket a műveleteket a terminálon keresztül egyszerűen végrehajthatja, lehetővé téve a folyamatok gyors és rugalmas kezelését.

**get\_portion\_data:** elérhető adagok listázása a kezdeti és vég dátum idővel, azonosítóval

**get\_panel\_data:** elérhető hűtőpanelek listázása névvel,azonosítóval

**get\_temperature\_data:** elérhető hómérsékletek listázása azonosítóval, idővel, adag\_id és hutopanel\_id-val, adag és hűtőpanel azonosító alapján

**insert\_temp:** hőmérséklet létrehozása a mostani időponttal adag és hűtőpanel alapján

**update\_temp:** hőmérséklet módosítása hőmérséklet azonosító alapján

**delete\_temp:** hőmérséklet törlése adag és hűtópanel alapján

**show\_panel:** egy panelhez és adaghoz tartozó adatok kimutatása grafikonon

**show\_panels:** két panelhez és egy adaghoz tartozó adatok kimutatása grafikonon

**show\_all\_panels:** az összes egy adaghoz tartozó panel hőmérsékleteinek egy grafikonra vetített kimutatása.

# II.A munkafolyamat részletei

## 1.Adatfeldolgozás excellel és szabadszemmel



Az adatok első vizsgálatakor a következő megállapításokat tettük:

* Dátumok el vannak csúszva a két file között (nem egyszerre kezdődik az idősor)
* Címek hibás karaktereket tartalmaznak
* Szükség van PK létrehozására, és egyértelmű hozzárendelésre
* Különböző nagyságú időintervallumok vannak megadva
* Nincsenek mértékegységek
* Adagközi idő hiányzik
* Idő alapján párosíthatóak az adatok
* 17- es adagnál eltérés látható

jellegre hasonlóak a görbék? akkor ugyanaz

kétszer ugyanazt az eredményt nem lehet kapni

lehet a kiindulási anyagban szennyeződés

meghibásodás

* a hűtőpanelek csv egy azonos időpontban a hűtőrendszer különböző pontjain mért értékek, ebből kifolyólag ha ugrásszerű változást látunk az hiba
* 07.19.10:45 percnél kiugró hőm. adatsor = berendezés hiba
* 07.18 18:25-től anomália valószínűleg lehűlés és csak rámondták ,hogy adag
* 27-es adag végén van egy kiugró hőmérséklet , csúcs az 1-es panelnél

### Adatok és diagramok kiértékelése

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

A képeken látható vonaldiagramok egy ívkemence hűtőpaneljeinek hőmérsékleti adatait mutatják az idő függvényében. X tengelyen tehát az mérési idő látható másodperc pontossággal, az Y tengelyen pedig a mért hőmérséklet.

**Az adatok és diagram alapján történő következtetések:**

* A hőmérsékletadatok jellemzően a 30°C és 50°C közötti tartományban mozognak, ami normális működésre utal a panelek többségénél.
* Az adatok folyamatosak, nincsenek jelentős mérési szünetek, és a hűtőpanelek működésében általánosan stabilitás mutatkozik.
* Bizonyos esetekben kiszúrtunk ingadozásokat és esetleges szélsőértékeket, ahol a hőmérséklet váratlanul megugrik vagy csökken.

**Panelek hőmérsékletének részletezése:**

* **Panel 5, 9, 13:**
  + Ezek a panelek stabilan a 30-50°C közötti tartományban mozognak, néhány kisebb hőmérséklet-ingadozással.
  + Az ingadozások általában a kemence működésének természetes változásaihoz köthetők, például a be- és kikapcsolási ciklusokhoz, valamint az anyagfeldolgozási szakaszokhoz. A folyamatok jellegükben megegyezőek
  + Nincs jelentős kiugrás vagy 0-ra való visszaesés.
* **Panel 6, 11:**
  + Itt figyelhetők meg jelentősebb hőmérsékleti kiugrások, akár 150-200°C körüli lokális csúcsok is. Ez eltér a többi paneltől, amelyek értékei stabilabbak.
  + Lehetséges okok: Ezeknek a paneleknek a mérésénél időszakos zavarok vannak. Valószínűleg a hibás adatok a szenzorok hibájából adódnak, főleg hogy a forráspont feletti érétkek nem reálisak.
* **Panel 1,3, 10, 14:**
  + Ezeknél a panelek működése során szintén láthatók kisebb ingadozások, azonban ezek mérsékeltek. A hőmérséklet ritkán megy 40-50°C fölé, és viszonylag konstans marad.
  + Az ingadozások oka lehet például a hűtőfolyadék áramlásának változása vagy a kemence hőmérsékleti ciklusai okozta változás, ez lehet reális adat, de meg kell nézni, hogy miből adódik az eltérés.

**Hőmérsékleti eltérések okainak lehetséges magyarázata:**

* **Hűtési rendszer működése:** Bizonyos panelek esetében a hőmérséklet rövid időre jelentősen megnövekszik(60-70 fok körüli értékre), ami utalhat a hűtési rendszerben fellépő átmeneti problémákra. Ezek lehetnek a hűtőfolyadék áramlásának megszakadásai, eltömődések vagy szivárgások.
* **Kemence belső működése:** Azoknál a paneleknél, ahol magasabb hőmérsékleti csúcsokat tapasztalunk (pl. Panel 6 és 11), feltételezhető, hogy közelebbi kapcsolatban állnak a kemence aktívabb zónáival. Ezek a területek hőtermelőbbek, így a panelek is magasabb hőmérsékleteknek vannak kitéve.
* **Hőmérséklet-szenzorok érzékenysége:** A jelentős hőmérsékleti kiugrásokat okozhatja a hőmérséklet-szenzorok átmeneti hibája is, bár a legtöbb adat a megfelelő működésre utal.

## 2.Relációs adatbázis fontossága a projektnél

A relációs adatbázis fontossága a projektünkben abban rejlik, hogy lehetővé teszi az adatok logikailag összekapcsolt, szervezett és biztonságos kezelését. Az adatokat táblázatos formában tároljuk, ahol minden tábla egy-egy jól definiált adatcsoportot képvisel, és az SQL nyelv segítségével könnyen lekérdezhetők, módosíthatók és kezelhetők. Ez a struktúra támogatja a pontos adatnyilvántartást, csökkenti a hibák lehetőségét, és biztosítja az adatok konzisztenciáját.

A fenti ER diagram három táblát tartalmaz: Adag, Homerseklet és Hutopanelek. Az Adag tábla az egyes adagok időbeli adatait tárolja, beleértve a kezdő és befejező időpontokat, a köztes időt és az összes időtartamot. A relációs adatbázis használatával minden egyes adag külön rekordként kezelhető, amelyet egyedi azonosítóval (adag\_id) különítünk el, ezzel biztosítva, hogy az egyes adagokra vonatkozó adatok könnyen visszakereshetők és szétválaszthatók legyenek.

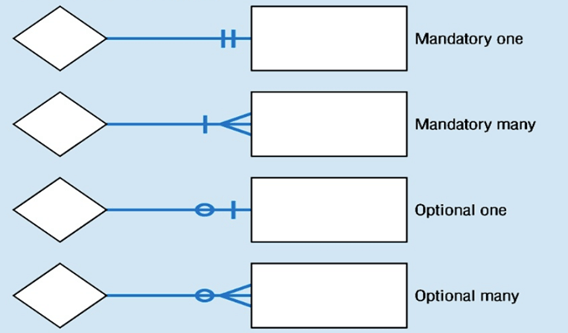
A relációs adatbázis ezen a projekten belül több szempontból is előnyös. Az adatok normalizációja minimalizálja a redundanciát, így hatékonyabb tárolást biztosít, ami különösen fontos a nagy mennyiségű hőmérsékleti adat esetén. Ezenkívül a normalizáció segít elkerülni az anomáliákat, például a beszúrási, törlési és módosítási anomáliákat, amelyek ronthatnák az adatok konzisztenciáját és megbízhatóságát. A normalizált adatstruktúra biztosítja, hogy minden információ pontosan egyszer szerepeljen, így egyetlen frissítéssel, módosítással vagy törléssel az adatbázis minden részében konzisztens marad az információ.

A relációs adatbázis tehát nemcsak az adatkezelés hatékonyságát növeli, hanem lehetőséget ad arra is, hogy az adatokat különféle szempontok szerint rugalmasan lehessen lekérdezni. A táblák közötti kapcsolatok lehetővé teszik, hogy az adagok, hőmérsékleti adatok és hűtőpanelek közötti összefüggéseket részletesen elemezzük, például lekérdezhetjük egy adott hűtőpanel összes hőmérsékleti adatát egy adott időintervallumban, vagy visszakereshetjük, hogy egy adott adag milyen tárolási körülmények között volt. Összességében a relációs adatbázis használata megbízható, átlátható és jól szervezett adatkezelést kínál, amely elengedhetetlen a projekt céljainak eléréséhez.

## 3.Adatbázis megtervezése ER diagrammal

### 3.1 Definíció

ER diagram Az Entity Relationship Diagram rövidítése, más néven ERD egy diagram, amely az adatbázisban tárolt entitáshalmazok kapcsolatát jeleníti meg, segít egy jól megtervezett adatbázis létrehozásában. Az ER diagramok alapfogalom alapján készülnek:

* entitások: hely,személy,objektum,esemény,fogalom valós dolog, kell hogy legyenek attribútumai és elsődleges kulcsa
* gyenge entitás: dupla gyémánt kapcsolja össze az entitással, nincs elsődleges kulcsa, gyenge entitáskészletben ez az erős entitáskészlet elsődleges kulcsának és részkulcsának kombinációja.
* attribútumok: entitás tulajdonsága
* kapcsolatok: ami összekapcsolja az az entitásokat, attribútumokat, gyakran azonosíthatjuk igékkel pl: tamás részt vesz a kémia órán

### 3.2 Az aktuális entitások és attribútumok meghatározásának lépései

#### 3.2.1 Entitás azanosítás:

* adagok tábla: adagok
* panelek tábla: panelek 1- 15-ig
* hőmérsékletek tábla a rekordokat tartalmazza hűtőpanelek csv-ből

#### 3.2.2 Kapcsolat azonozítás:

idő alapján kapcsolhatóak az adagok a panelekkel, a paneleknek függősége van az adagolási ciklusokra (ha egy adagot hozzáadunk megváltozik a hűtőpanel által mért hőmérséklet)

#### 3.2.3 Kardinalitás azonosítása:

* Egy adaghoz több hőm. kapcsolódik 1:N
* Egy hűtőpanelhez több hőmérséklet kapcsolódik 1: N
* Egy hőmérséklethez egy panel tartozik 1:1
* Egy hőmérséklethez egy adag tartozik 1:1

#### 3.2.4 Attribútumok azonosítása:

adagok:

* adag\_id – elsődleges kulcs
* datumido\_kezdet
* datumido\_veg
* kozetes\_ido
* idotartam

homerseklet:

* homerseklet\_id – elsődleges kulcs
* adag\_id – idegen kulcs
* hutopanel\_id – idegen kulcs
* hőmérséklet
* datumido

hutopanelek:

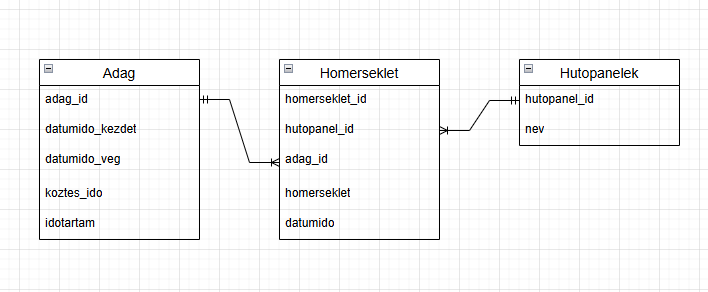
* hutopanel\_id – elsődleges kulcs
* nev

#### 3.2.5 ER Diagram létrhozása

2. ábra ER Diagram 1.verzió



3. ábra ER Diagram 2.verzió



## 4.Adatok feltöltése, ellenőrzése

* dátum és idő egybevonása a könnyebb kezelhetőségért, átláthatóságért kevesebb hibalehetőség(DATETIME mindenhol, figyelmet fordítva arra, hogy ne veszítsünk adatot – mp a hűtőpanelek táblánál)
* adagok számának kinevezése elsődleges kulccsá (id), amit idegen kulcsként a hőmérsékletek tábla is megkap (adag\_id)
* Hűtőpanelek táblához is rendeljünk egy azonosítót (panel\_id)
* Hőmérsékletek táblához is rendeljünk egy azonosítót (homerseklet\_id)
* egy időpont többször egymás után szerepel , alig eltérő adatokkal, érdemes ezt redukálni-? nem, azért mert nem egyformák, és nem tudjuk milyen a vízáramoltatás ( 100 l/sec vagy 10 l/sec így nem tudjuk megítélni mennyit változtat 0,2 fok eltérés, mekkora a súlya)
* sor címek hibás karaktereket tartalmaznak : UTF-8 encodeinggal is hibás karaktereket jelenít meg, valószínűleg más encodeingban lett elmentve (pl.:Windows-1252)
* A rekorodok vegyesen tartalmaznak ’,’-őt és ’.’-ot , ezt fontos átalakítani az adatok elmentésénél mert pl ha REAL type-ba mentjük a rekordokat nem alakítja át a rendszer és későbbi számolásnál probléma lesz vele, ezért feltöltésnél átalakítjuk.
* A kiugró biztosan hibás értékeket tisztítjuk – ehhez bővebb ismeretekre lenne szükség,hogy pontosan megállapítsuk melyek azok az értékek amik már nem az adagok mérésénél tapasztalt normális eltérések ( kiugrás 60 ról 70 fokra ) hanem a szenzor vagy szenzor kezelő program hibájából adódik ( 250 fok ).

## 5.Adatbázis létrehozása

### 5.1 DCL és TCL: SQLite-n belül nincs jogosultságkezelés. a. Írják le, hogy melyik résztvevőnek milyen jogosultságra lett volna szüksége, a munkájához.

Az adatok rögzítése és feldolgozása során milyen szerepköröket hoznának létre, és azokat milyen jogosultsággal látnák el?

A jogosultságkezelés kidolgozásához a dokumentáció **1.2**-és pontját vettem alapul, ahol megtalálhatóak a szerepkörök és azok magyarázatai.

**Feladat résztvevői:**

1. **Projektvezető (Project Manager)**

2. **Adat- és üzleti elemző (Data and Business Analyst)**

3. **Vezető fejlesztő (Lead Developer)**

4. **Sysadmin (Sysadmin)**

5. **Adatmérnök (Data Engineer)**

**1. Projektvezető (Project Manager)**

* **Jogosultságok:** Olvasási és korlátozott írási jogosultság.
* **Indoklás:** A projektvezetőnek szüksége van arra, hogy megtekinthesse a dokumentációt, de nem szabad, hogy közvetlenül módosítsa az adatbázis tartalmát.
* **SQL parancsok:**

--felhasználó hozzárendelése és törlése a szerepkörhöz

GRANT ‘project\_manager’ TO ‘Gahor\_Krisztian;

REVOKE ‘project\_manager’ FROM ‘Gahor\_Krisztian;

GRANT ‘project\_manager’ TO ‘Hegedus\_Alexandra’;

GRANT SELECT ON adagok.csv, hutopanelek.csv;

GRANT SELECT ON main.py, database\_manager.py TO 'project\_manager';

GRANT SELECT, UPDATE ON dokumentacio TO 'project\_manager';

**2. Adat- és üzleti elemző (Data and Business Analyst)**

* **Jogosultságok:** Csak olvasási jogosultság.
* **Indoklás:** Az elemzőnek szüksége van az adatokra, de nem kell módosítania azokat.
* **SQL parancsok:**

--új szerepkör létrehozása

CREATE ROLE ‘data\_analyst’ WITH LOGIN PASSWORD ‘strong\_password’

GRANT SELECT ON adagok.csv, hutopanelek.csv TO 'data\_analyst';

GRANT SELECT ON filter\_data,py TO 'data\_analyst';

**3. Vezető fejlesztő (Lead Developer)**

* **Jogosultságok:** Teljes írási és módosítási jogosultság.
* **Indoklás:** A vezető fejlesztőnek szüksége van a teljes körű hozzáférésre a fejlesztési környezetben, hogy új táblákat hozhasson létre és módosíthassa az adatbázis sémát.
* **SQL parancsok:**

--új felhasználó létrehozása

CREATE USER ‘Vadi Gergo WITH PASSWORD ‘strong\_password’;

GRANT ‘lead developer’ TO ‘Vadi Gergo’;

GRANT SELECT, UPDATE, INSERT ON adagok.csv, hutopanelek.csv TO ‘lead\_developer’;

GRANT ALL PRIVILEGES ON main.py, database\_manager.py, create and load.py, filter\_data.py, generate\_new\_csv.py TO 'lead\_developer';

GRANT SELECT ON dokumentacio.docx TO ‘lead developer’;

**4. Rendszergazda (Sysadmin)**

* **Jogosultságok:** Olvasási és írási jogosultság .
* **Indoklás:** A sysadmin feladatköre ,hogy eldöntse ki mihez férhet hozzá és az ó feladata a projekt biztonsági rendszerének kezelése.
* **SQL parancsok:**

GRANT SELECT, INSERT, UPDATE, DELETE ON adagok.csv, hutopanelek.csv TO 'lead\_tester';

GRANT SELECT, UPDATE ON filter\_data.py TO ‘lead\_tester’;

**5. Adatmérnök (Data Engineer)**

* **Jogosultságok:** Teljes hozzáférés az adatokhoz és az adatbázis infrastruktúrához.
* **Indoklás:** Az adatmérnöknek kezelnünk kell az adatok áramlását és karbantartását, ami teljes hozzáférést igényel.
* **SQL parancsok:**

GRANT ALL PRIVILEGES ON adagok.csv, hutopanelek.csv TO 'data\_engineer';

GRANT ALL PRIVILEGES ON create\_and\_load.py, generate\_new\_csv.py TO 'data\_engineer';

GRANT SELECT ON main.py, database\_manager.py, filter data.py TO ‘data engineer’;

### 5.2 b. Vizsgálják meg, hogy van-e, és ha van, akkor milyen esetben tranzakciókezelésre.

### 5.3 1 tranzakciót tegyenek bele akkor is, ha nem feltétlen szükséges csak a példa kedvéért.

## 6.Lekérdezések

### 6.1 Végezzenek el lekérdezéseket, melyek összecsatolnak táblákat (JOIN).

A megadott SQL lekérdezések(queries.py) több táblát kapcsolnak össze, lehetővé téve a hőmérséklet- és adagadatok közötti kapcsolat elemzését. Az alábbiakban részletezem az egyes lekérdezéseket, a hozzájuk kapcsolódó aggregációs függvények alkalmazását, és hogy miért fontosak ezek a műveletek a hűtőpanelek és adagok kezelésében.

**Adagok és hőmérséklet lekérdezése:** A homerseklet és adagok táblák összekapcsolása az adagok hőmérsékleteivel. Ez lehetővé teszi a hőmérséklet és az adag időtartamának összehasonlítását, amely segíti az adatok szűrését és a megfelelő adagok kiválasztását.

**Minden panel hőmérsékletének lekérdezése:** Az összes hőmérsékletadat megjelenítése egy adaghoz, lehetővé téve az összes hűtőpanel hőmérsékleti viszonyainak egy grafikonon való megjelenítését.

### 6.2. Használjanak aggregációs függvényeket.

A feladat során a COUNT aggregációs függvényt használtuk a függvény segítségével ellenőriztük, hogy a megadott hűtőpanel és adag létezik-e az adatbázisban.

### 6.3. Írják le, hogy miért végezték el ezeket a műveleteket, és milyen következtetésekre jutottak.

Ezek a műveletek lehetővé teszik az adatok alaposabb elemzését és az adagok és hűtőpanelek közötti kapcsolatok mélyebb megértését. Az összecsatolás és aggregáció során világosabb képet kaphatunk a hőmérsékletek alakulásáról, a különböző adagok hatékonyságáról és a panelek teljesítményéről. Valamint meg tudjuk jeleníteni a kevesebb jogosultsággal rendelkező felhasználók számára az elérhető adatokat.

## 7. Allekérdezések, UNION

### 7.1 Szükség esetén készítsenek egymásba ágyazott lekérdezéseket

**Két panel hőmérsékletének lekérdezése:** Két panel hőmérsékletének párhuzamos vizsgálata, ami lehetővé teszi a különböző panelek teljesítményének összehasonlítását egy adag keretein belül.

### 7.2 A részeredményeiket INSERT és SELECT segítségével tegyék ki külön táblákba

### 7.3. Vizsgálják meg, hogy mit lehetne kimutatni, vagy ellenőrizni az UNION utasításokkal

## 8. Teljesítmény optimalizálás

### 8.1 Melyik mezőre milyen kulcsot, indexet alkalmaznának, és miért?

A teljesítmény optimalizálása érdekében fontos, hogy a kulcsokat és indexeket gondosan tervezzük meg, mivel ezek nagymértékben befolyásolják a lekérdezési sebességet és a műveletek hatékonyságát. Az alábbiakban a táblák releváns mezőire javasolt indexeket és kulcsokat mutatom be:

**Elsődleges kulcsok (Primary Key):**

Adag tábla: adag\_id mező. Az adag\_id egyedi azonosítója minden egyes rekordnak, ezért elsődleges kulcsként szolgálhat, ami gyorsítja az egyedi rekordok azonosítását és lekérdezését.

Homerseklet tábla: id mező. Az id mező elsődleges kulcs, amely minden hőmérsékleti adat egyedi azonosítója. Ezen kívül a hutopanel\_id és adag\_id mezőkön idegen kulcsokat (Foreign Key) definiálunk, hogy a kapcsolatok biztosítva legyenek.

Hutopanelek tábla: hutopanel\_id mező. Ez is elsődleges kulcs, amely minden hűtőpanel egyedi azonosítója.

**Idegen kulcsok (Foreign Key):**

Homerseklet tábla: Az adag\_id mező kapcsolódik az Adag táblához, míg a hutopanel\_id mező a Hutopanelek táblához kapcsolódik. Ezek az idegen kulcsok biztosítják az adatok közötti integritást.

**Indexek:**

Adag tábla: start\_datetime, end\_datetime. Mivel a lekérdezések gyakran időalapúak lehetnek (például egy adott időintervallumon belüli adatok lekérése), érdemes indexet létrehozni a start\_datetime és end\_datetime mezőkön. Ezek az indexek gyorsítják az időalapú szűréseket és elemzéseket.

Homerseklet tábla: datumido, hutopanel\_id, és adag\_id. A datumido mezőre érdemes indexet alkalmazni, mert ez gyakran használt mező az idő alapú lekérdezésekhez, és segít a hőmérsékleti adatok gyors elérésében. A hutopanel\_id és adag\_id indexelése szintén gyorsítja a keresést, különösen, ha szűrni szeretnénk a hőmérsékleti adatokat adott hűtőpanel és adag szerint.

Ezekkel a kulcsokkal és indexekkel a lekérdezések gyorsabbá válnak, mivel a rendszer kevesebb rekordot kell átvizsgáljon a releváns adat megtalálásához.

### 8.2 . Milyen módszert javasolnának az adatműveletek gyorsítására, ha megkapnának 10 évnyi adatot és azt kellene feldolgozni?

10 évnyi adat feldolgozása jelentős mennyiségű adatot jelenthet, így szükség lehet további optimalizálási módszerekre az adatműveletek gyorsításához. Az alábbi módszereket javasolnám:

**Particionálás:**

Az adatbázis partícionálása időintervallumok alapján (például évenként vagy havonta) segíthet a nagy mennyiségű adat kezelésében. Így egy adott időszakra vonatkozó lekérdezések csak a releváns partíciókat érintik, nem pedig a teljes adatbázist, ami jelentősen gyorsítja a folyamatot.

**Materializált nézetek**:

Az előre kiszámított és összesített adatokat materializált nézetek formájában tárolhatjuk. Például, ha gyakran van szükség bizonyos aggregált adatokra (mint a havi vagy éves átlaghőmérséklet), ezeket előre kiszámíthatjuk és elmenthetjük egy materializált nézetben. Ezzel a gyakran használt összesített adatokat gyorsabban lehet elérni.

**Index karbantartása és optimalizálása**:

Mivel az indexek hatékonysága az idő múlásával csökkenhet, rendszeresen karban kell tartani és optimalizálni őket, hogy az adatbázis gyorsan válaszoljon a lekérdezésekre. Túl sok index is lassíthatja az adatműveleteket, ezért érdemes csak a legfontosabb mezőkön megtartani az indexeket.

**Adatbázis cache** (gyorsítótár):

Ha gyakran hajtunk végre ugyanazt a lekérdezést, érdemes lehet cache-t használni a lekérdezési eredmények tárolására. Ez azt jelenti, hogy az adatbázis nem minden alkalommal számolja ki az adatokat, hanem a gyorsítótárból olvassa azokat.

**Batch feldolgozás**:

Nagy adathalmazok esetén nem célszerű egyszerre feldolgozni minden adatot. Helyette érdemes batch (tételes) feldolgozást alkalmazni, ahol kisebb adatcsomagokat dolgozunk fel egy időben. Ez csökkenti a memóriahasználatot és a processzor terhelését, valamint csökkenti a teljesítményromlás esélyét.

**Archiválás**:

A nagyon régi adatokat, amelyekre már ritkán van szükség, érdemes lehet archív adatbázisba áthelyezni. Így a fő adatbázis mérete csökken, és csak a releváns, aktuális adatokat tartjuk meg, amelyek gyorsabban lekérhetők.

**Python Pandas optimalizálása nagy adatmennyiségekhez**:

Pythonban a Pandas alapértelmezett módszerei nem mindig optimálisak nagy adatmennyiségekhez. Ilyenkor érdemes a Dask könyvtárat használni, amely lehetővé teszi a Pandas DataFrame-ek kezelését nagyobb adathalmazok esetén is párhuzamosan.

## 9. Adattisztítás. Az adatok hibával terheltek

### 9.1 Milyen módszerrel, illetve logika alapján kerestek, szűrtek hibás adatokat?

Az egyes hőmérsékleti adatok esetében meghatároztunk egy referencia értéket, amely az adott adaghoz és hűtőpanelhez tartozó hőmérsékletek átlagértéke. Ez alapján megvizsgáltuk, hogy az adott hőmérsékleti értékek mennyire térnek el az átlagtól. Az átlag köré egy intervallumot határoztunk meg, amelyet a *kuszob* változó segítségével paraméterezünk. Az intervallum alsó és felső határát az átlag alapján számítottuk ki: az átlaghoz hozzáadtuk és levontuk az átlag egy százalékos értékét, amely a megadott küszöbérték százalékában lett kifejezve. Minden hőmérsékleti értéket összehasonlítottunk az intervallum alsó és felső határaival. Ha egy érték ezen tartományon kívül esett (vagyis túl alacsony vagy túl magas volt az átlaghoz képest), akkor hibás adatnak tekintettük.

### 9.2. Milyen módszerrel kezelték azokat?

**Hibás rekordok naplózása**: Mielőtt töröltük volna az intervallumon kívül eső rekordokat, minden ilyen rekordot rögzítettünk egy naplófájlba (torolt\_rekordok.txt). Ez a naplófájl tartalmazta az adag azonosítóját, a hűtőpanel azonosítóját, az intervallum alsó és felső határát, valamint a hibás hőmérsékleti értékeket. Ezzel biztosítottuk, hogy a hibás adatok később visszakereshetők legyenek, ami segíti az esetleges elemzési visszatekintéseket vagy a hibák további elemzését.

**Hibás adatok törlése az adatbázisból**: Miután a hibás rekordokat naplóztuk, a megfelelő SQL DELETE művelet segítségével eltávolítottuk azokat az *Homerseklet* táblából. Ezáltal az adatbázisban csak a hitelesített és érvényes hőmérsékleti adatok maradtak, amelyek megbízható alapot biztosítanak az elemzésekhez.

**Az adatok konzisztenciájának biztosítása**: A törlés után elvégeztük az adatbázis mentését (commit), ezzel véglegesítve a módosításokat. Az adattisztítási folyamat befejeztével a hibás adatok eltávolításra kerültek, így az adatbázis már csak a konzisztens, hiteles adatokat tartalmazza.

## 10. Milyen tudásuk alapján tudták azonosítani a hibákat?

### 10.1 Mi alapján tekintették a hibás adatokat hibás adatnak?

Ahhoz, hogy alaposan megértsük a problémát, két fő területre bontottuk a feladatot: egyrészt a folyamathoz szükséges szakmai ismeretekre, másrészt a programozói szaktudásra, amely a kódoláshoz elengedhetetlen.

Először elmélyültünk a folyamatok alapjaiban, hogy átlássuk, milyen mechanizmusok mentén működhet egy ilyen rendszer. Itt különösen nagy hangsúlyt fektettünk a domain tudás megszerzésére, mivel csak így tudtuk igazán kontextusba helyezni a problémát, és megérteni a projekt által támasztott specifikus igényeket. Ezen a ponton azt vettük észre, hogy önmagában a szakmai ismeretek nem elegendőek – a megoldás másik fontos fele a programozásban rejlik.

Ennek érdekében konzultáltunk egy, jelenleg is az anyagtudomány területén oktató szakemberrel, aki segített a folyamatok és technológiák részletes megismerésében. A programozási oldal elemzéséhez pedig az órákon tanultakat vettük alapul, hogy a kódolás során a szükséges technikai eszközökkel hatékonyan oldjuk meg a feladatot.

Ez a kétféle szaktudás – a domain és a programozási ismeretek – kombinációja tette lehetővé, hogy átfogó megértést és hatékony megoldást alakítsunk ki.

### Adatbevitelhez milyen érvényesítési szabályokat, milyen kényszereket javasolnának

Az adatbevitel során alkalmazandó érvényesítési szabályok és kényszerek az adatok pontosságának, teljességének és megbízhatóságának biztosítása érdekében szükségesek. Az alábbi főbb szabályokat és kényszereket javasolnánk:

1. **Kötelező érték (NOT NULL)**: Bizonyos mezők nem lehetnek üresen hagyhatók, így elkerülhető a hiányos adatok tárolása.
2. **Elsődleges kulcs (PRIMARY KEY)**: Minden rekord egyedi azonosításához szükséges, amely biztosítja, hogy egy táblán belül minden sor egyedi legyen.
3. **Idegen kulcs (FOREIGN KEY)**: Az adatbázis összefüggéseinek biztosítása érdekében a külső kulcsokkal garantálhatjuk, hogy csak a panelekhez kapcsolódjanak, amelyek szerepelnek a hutopanelek táblában.
4. **Értékkorlátok (CHECK)**: Az érvényesítési szabályok biztosítják, hogy egy mező értéke -100 és 100 fok közötti tartományra korlátozzuk, hogy ésszerű adatok kerüljenek az adatbázisba.
5. **Index**:Egy adatbázisban létrehozott speciális struktúra, amellyel gyorsítjuk az adatok keresését és lekérdezését egy adott oszlopban vagy oszlopcsoportban. Az indexek létrehozásával javíthatjuk a lekérdezések teljesítményét, de ugyanakkor több helyet is igényelnek, és lassíthatják az adatbevitelt, mivel a változásoknál az indexeket is frissíteni kell.
6. **Egyéb**: Nem nevezhető kényszernek inkább csak plusz beállítási lehetőség az **AUTOINCREMENT**. Ez a beállítás automatikusan növekszik minden új rekord hozzáadásakor. Ezzel biztosítható, hogy az adott oszlop egyedi értékeket kapjon, és nincs szükség manuális adatbevitelre, Amivel gyorsíthatjuk és hibalehetőségtől tudjuk megvédeni az adatbázist

Ezekkel az érvényesítési szabályokkal biztosítottuk, hogy az adatbázis megfelelően kezelje és megőrizze az adatokminőségét, csökkentve a hibalehetőségeket, valamint az indexek a keresési műveleteket gyorsítják.

## 11. Az adatok egy ipari adatgyűjtő program írja az adatbázisba.

Az adatok rögzítése és feldolgozása során milyen szerepköröket hoznának létre, és azokat milyen jogosultsággal látnák el?

## 12. Biztonsági mentésre milyen stratégiát választanának?

Biztonsági mentésre a github verziókezelő rendszert választottuk. A GitHub felhőalapú platform, amely biztonságos tárolási megoldást nyújt, így a mentéseket nem egyetlen fizikai helyen tároljátok. Ha a helyi gépen bármilyen probléma merülne fel, a távoli GitHub-tárolóból mindig visszaállíthatók az adatok. A ’main’, azaz fő ’branch’-en a végső verzió található meg. Az aktuális feladatokat külön-külön branchen készítettük el, és amikor elérte a végleges verziót a project manager vagy vezető fejlesztő nézte át és fogadta el, vagy javította, ahol szükség volt rá. Így tudtuk elkerülni a potenciális problémákat:

* hibás kód
* adatvesztés
* eltérő verziók a tagok között

Így a lokális verziók mindig a legutolsó helyes verziót tartalmazták és ha esetlegesen valamilyen probléma adódott a main branchről egyszerűen tudtuk korrigálni.

# III.Összegzés