2023. 06. 16.

Projektmunka I.

*Féléves feladat dokumentáció*

Tartalomjegyzék

[Tartalomjegyzék 1](#_Toc137828740)

[Bevezetés 3](#_Toc137828741)

[A projekt témája 3](#_Toc137828742)

[A projekt alapját adó szolgáltatás működése 3](#_Toc137828743)

[Az adatok forrása 3](#_Toc137828744)

[Az adatbázis táblái és tárolt adatai 3](#_Toc137828745)

[VAROSOk 3](#_Toc137828746)

[mOSODAK 4](#_Toc137828747)

[dOLGOZOK 4](#_Toc137828748)

[GEPEK 4](#_Toc137828749)

[uGYFELEK 4](#_Toc137828750)

[RENDELESEK 4](#_Toc137828751)

[RENDELES\_RESZLETEK 5](#_Toc137828752)

[sZOLGALTATASOK 5](#_Toc137828753)

[ERTEKELESEK 5](#_Toc137828754)

[A végleges táblák és kapcsolataik 6](#_Toc137828755)

[Megvalósítás 7](#_Toc137828756)

[A táblák feltöltése adatokkal 7](#_Toc137828757)

[varosok 7](#_Toc137828758)

[mosodak 7](#_Toc137828759)

[dolgozok 7](#_Toc137828760)

[gepek 7](#_Toc137828761)

[ugyfelek 7](#_Toc137828762)

[rendelesek 8](#_Toc137828763)

[rendeles\_reszletek 8](#_Toc137828764)

[szolgaltatasok 8](#_Toc137828765)

[ertekelesek 8](#_Toc137828766)

[Egyéb funkciók létrehozása 8](#_Toc137828767)

[Szekvenciák 8](#_Toc137828768)

[Tárolt eljárások 9](#_Toc137828769)

[Triggerek 9](#_Toc137828770)

[Lekérdezések elemzése, optimalizáció 10](#_Toc137828771)

[1.lekérdezés 10](#_Toc137828772)

[Elemzés 10](#_Toc137828773)

[Optimalizáció 11](#_Toc137828774)

[2. lekérdezés 11](#_Toc137828775)

[Elemzés 11](#_Toc137828776)

[Optimalizáció 12](#_Toc137828777)

[NoSQL adatbázis kezelés 13](#_Toc137828778)

[Az adatok átvitele MongoDB-be 13](#_Toc137828779)

[Lekérdezések 17](#_Toc137828780)

[1. Mely vársoban található mosoda? 17](#_Toc137828781)

[2. Dolgozók átlagfizetése: 17](#_Toc137828782)

[3. dolgozók összfizetése: 17](#_Toc137828783)

[4. Mosodák átlag tisztasága: 18](#_Toc137828784)

[5. Mosodák átlagos biztonsága: 18](#_Toc137828785)

[6. Mosodák átlagos elrendezése: 18](#_Toc137828786)

[7. szolgáltatások átlagos ára: 19](#_Toc137828787)

[Táblákat létrehozó script 19](#_Toc137828788)

[Sequence 20](#_Toc137828789)

# Bevezetés

## A projekt témája

A projekthez egy olyan adatbázist fogok használni, amit én generáltam le egy fiktív szolgáltatás alapjait véve, mivel az interneten nem találtam olyan adathalmazt ami megfelelt volna az igényeimnek. Az adatokat oly módon generáltam le, hogy az megfeleljen a projektmunka követelményeinek.

## A projekt alapját adó szolgáltatás működése

Egy több városra kiterjedő mosoda-lánc adatbázisát terveztem meg. Az adatbázis segítséget nyújt a mosodák forgalmának feltérképezésében, a mosodák kihasználtságában, a gépállomány számának felmérésében és a dolgozók létszámának számontartásában. Minden mosoda három alap szolgáltatással rendelkezik (mosás, szárítás, mosás-szárítás). Ezeknek díjai mosodánként eltérőek (1500ft-2000ft, 2500ft-3500ft). A regisztrált ügyfelek az adatbázis által online is láthatják a szabad gépeket és azok kapacitását. A regisztrált ügyfeleknek lehetőségük van értékelést adni a mosoda szolgáltatásáról (tisztaság, biztonság, elrendezés). Bizonyos szolgáltatásokra rendeléseket is tudnak leadni az online felületen.

## Az adatok forrása

Az elképzelt adatbázisomhoz nem találtam az interneten megfelelő adathalmazt, ezért magamnak generáltam le a szükséges adatokat hozzá. A „Spawner” nevű alkalmazással generáltam minden táblámhoz adatokat.

## Az adatbázis táblái és tárolt adatai

### VAROSOk

Ebben a táblában tároljuk el azokat a városokat, ahol a mosodák megtaálhatóak.

* VAROS\_ID: Elsődleges kulcs, egyedi azonosító minden városhoz.
* VAROS\_NEV: A város neve.

### mOSODAK

Ebben a táblában tároljuk el meglévő mosodáinkat.

* MOSODA\_ID: Elsődleges kulcs, egyedi azonosító minden mosodához.
* VAROS\_ID: Idegen kulcs, hivatkozik a mosoda városára.
* MOSODA\_NEV: A mosoda neve.
* MOSODA\_CIM: A mosoda címe.
* MOSODA\_TELEFON: A mosoda telefonszáma.

### DOLGOZOK

A mosodákhoz vannak beosztva dolgozók, amit ebben a táblában tárolunk el.

* DOLGOZO\_ID: Elsődleges kulcs, egyedi azonosító minden dolgozóhoz.
* MOSODA\_ID: Idegen kulcs, hivatkozik a dolgozóhoz tartozó mosodára.
* NEV: A dolgozó neve.
* MUNKAKOR: A dolgozó munkaköre.
* FIZETES: A dolgozó fizetése.

### GEPEK

A mosodákban található gépek ebben a táblában tároljuk.

* GEP\_ID: Elsődleges kulcs, egyedi azonosító minden géphez.
* MOSODA\_ID: Idegen kulcs, hivatkozik a géphez tartozó mosodára.
* GEP\_TIPUS: A gép típusa (pl. mosógép, szárítógép).
* KAPACITAS: A gép kapacitása.

### UGYFELEK

Ebben a táblában tároljuk azokat az ügyfeleket akik igénybe vették valamelyik mosoda szolgáltatásait.

* UGYFEL\_ID: Elsődleges kulcs, egyedi azonosító minden ügyfélhez.
* NEV: Az ügyfél neve.
* CIM: Az ügyfél címe.
* TELEFON: Az ügyfél telefonszáma.

### RENDELESEK

A leadott rendeléseket tároljuk ebben a táblában.

* RENDELES\_ID: Elsődleges kulcs, egyedi azonosító minden rendeléshez.
* MOSODA\_ID: Idegen kulcs, hivatkozik a rendeléshez tartozó mosodára.
* UGYFEL\_ID: Idegen kulcs, hivatkozik a rendelést leadó ügyfélre.
* DATUM: A rendelés dátuma.
* OSSZEG: A rendelésért fizetendő teljes összeg.

### RENDELES\_RESZLETEK

Ebben a táblában a leadott rendelések részletetit tároljuk.

* RENDELES\_RESZLET\_ID: Elsődleges kulcs, egyedi azonosító minden rendelés részlethez.
* RENDELES\_ID: Idegen kulcs, hivatkozik a részlethez tartozó rendelésre.
* SZOLGALTATAS\_ID: Idegen kulcs, hivatkozik a rendeléshez kapcsolódó szolgáltatásra.
* MENNYISEG: Hány szolgáltatás volt a rendeléshez.

### SZOLGALTATASOK

Ebben a táblában szolgáltatások adait tároljuk.

* SZOLGALTATAS\_ID: A szolgáltatás azonosítója, elsődleges kulcs.
* MOSODA\_ID: A mosoda azonosítója, amely nyújtja a szolgáltatást, idegen kulcs.
* SZOLGALTATAS\_NEV: A szolgáltatás neve.
* AR: A szolgáltatás ára.

### ERTEKELESEK

Ebbe a táblában az értékeléseket tároljuk el.

* ERTEKELES\_ID: Elsődleges kulcs, egyedi azonosító minden értékeléshez.
* MOSODA\_ID: Idegen kulcs, hivatkozik az értékelést kapott mosodára.
* UGYFEL\_ID: Idegen kulcs, hivatkozik az értékelést leadó ügyfélre.
* DATUM: Az értékelés keletkezésének dátuma.
* TISZTASAG: Az értékelés a tisztaságra (1-5 csillag).
* BIZTONSAG: Az értékelés a tisztaságra (1-5 csillag).
* ELRENDEZES: Az értékelés a tisztaságra (1-5 csillag).

## A végleges táblák és kapcsolataik

A táblák megtervezése[[1]](#footnote-1) után a következő lett az adatbázis végleges modellje:

A screenshot of a computer

Description automatically generated with low confidence

# Megvalósítás

A táblák megtervezése és létrehozása után elkészítettem az ezekhez szükséges adatokat amikkel azokat feltöltöttem, valamint létrehoztam az egyéb szükséges funkciókat. (sequence, trigger)

## A táblák feltöltése adatokkal

Ebben a fejezetben tábla szintű bontásban ismertetem, hogy milyen módon kerültek az adatok létrehozásra, valamint hány rekordot tartalmaz az adott tábla.

### varosok

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **10** darab rekordot szúrtam be.

### mosodak

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **50** darab rekordot szúrtam be.

### dolgozok

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* A tábla **100** darab rekordot tartalmaz.

### gepek

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **100.000** darab rekordot szúrtam be.

### ugyfelek

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **100.000** darab rekordot szúrtam be.

### rendelesek

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **225.000** darab rekordot szúrtam be.

### rendeles\_reszletek

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **225.000** darab rekordot szúrtam be.

### szolgaltatasok

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **150** darab rekordot szúrtam be.

### ertekelesek

* Az adatok létrehozásához a „***Spawner***” nevű alkalmazást használtam, ebben könnyedén be tudtam állítani az egyes mezőket.
* Ebbe a táblába így **5000** darab rekordot szúrtam be.

## Egyéb funkciók létrehozása

Az adatbázishoz létrehoztam szekvenciákat, triggereket és tárolt eljárásokat, amik hasznosak lehetnek, ebben a fejezetben ezekről lesz szó.

### Szekvenciák

Az elsődleges kulcsok későbbi egyszerűbb generálásához szekvenciákat hoztam létre néhány táblához. Minden táblához létrehoztam szekvenciát.

A szekvenciákat létrehozó script megtalálható a mellékletben.

### A picture containing text, font, screenshot, white Description automatically generatedTárolt eljárások

|  |
| --- |
| CREATE OR REPLACE PROCEDURE fizetesEmeles(d\_id number) is  cursor kurzor is select \* from Dolgozok where DOLGOZO\_ID = d\_id for update of FIZETES; rekord kurzor%ROWTYPE;  ujFizetes number; begin  for rekord in kurzor loop  ujFizetes := rekord.fizetes \* 1.1;  UPDATE Dolgozok set FIZETES = ujFizetes where current of kurzor; end loop;  end;  execute fizetesEmeles(1); |

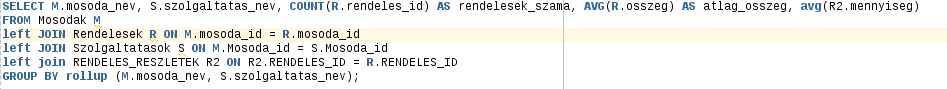
### A picture containing text, font, white, screenshot Description automatically generatedTriggerek

|  |
| --- |
| Ha egy dolgozónak megváltozik a fizetése akkor azt írja ki a nevét, a régi fizetését és az új fizetését.set SERVEROUTPUT ON;  CREATE OR REPLACE TRIGGER fizetesValtozas after update on Dolgozok  for each row when(new.fizetes<>old.fizetes) begin  DBMS\_OUTPUT.PUT\_LINE(:old.nev ||' '|| 'Fizetése megváltozott');  DBMS\_OUTPUT.PUT\_LINE('Régi fizetés: ' || :old.fizetes); DBMS\_OUTPUT.PUT\_LINE('Új fizetés: ' || :new.fizetes); end; |

# Lekérdezések elemzése, optimalizáció

## 1.lekérdezés

A mosodák nevét, a szolgáltatások nevét, a rendelések számát es ezek átlagait kérdezem le. Ez a lekérdezés mosodákra lebontva megmondja, melyik szolgáltatás mennyibe kerül átlagosan és hány vették igénybe a szolgáltatást átlagosan. A lekérdezés a következőképp néz ki:



### Elemzés

Az adatbázis kezelő a lekérdezést a következő módon hajtja végre:

A picture containing text, font, receipt, screenshot

Description automatically generated

* Elsőként HASH JOIN -t hajt végre a **Mosodak**  és a **Rendelesek** táblára. Full table scan segítségével beolvassa a **Rendelesek** tábla adatait. A két tábla feltételnek megfelelő rekordjait access predicates segítségével végzi el, ami két tábla azonosítóit köti össze.
* Második lépésként HASH JOIN -t hajt végre a **Rendeles\_reszletek**. A két tábla feltételnek megfelelő rekordjait access predicates segítségével végzi el, ami két tábla azonosítóit köti össze.
* Harmadik lépésként HASH JOIN -t hajt végre a **Szolgaltatasok** táblára. A két tábla feltételnek megfelelő rekordjait access predicates segítségével végzi el, ami két tábla azonosítóit köti össze.

Végeredmény: **2110 cost**

### Optimalizáció

* A táblák összekapcsolására INNER JOIN -t használtam.

Ezt követően a végrehajtási terv a következőképp néz ki:

A screenshot of a computer

Description automatically generated with medium confidence

* Elsőként HASH JOIN -t hajt végre a **Mosodak**  és a **Rendelesek** táblára. Full table scan segítségével beolvassa a **Rendelesek** tábla adatait. A két tábla feltételnek megfelelő rekordjait access predicates segítségével végzi el, ami két tábla azonosítóit köti össze.
* Ezt követően Nested loops segítségével összekapcsolja a **Rendelesek**  és a **Rendeles\_reszletek** táblát.
* Ennek az eredményét **Index range scan** segítségével kiválogatja az indexértékek közül azokat, amelyek az access predicates feltételnek megfelelnek.
* Harmadik lépésként HASH JOIN -t hajt végre a **Mosodak** táblára. A két tábla feltételnek megfelelő rekordjait access predicates segítségével végzi el, ami két tábla azonosítóit köti össze.
* Negyedik lépésként HASH JOIN -t hajt végre a **Szolgaltatasok** táblára. A két tábla feltételnek megfelelő rekordjait access predicates segítségével végzi el, ami két tábla azonosítóit köti össze.

Végeredmény: **1404 cost**

## 2. lekérdezés

A mosodák, dátum és szolgáltatások szerint csoportositok. A lekérdezést a következő módon írtam meg:

A picture containing text, screenshot, font, number

Description automatically generated

### Elemzés

Az adatbázis kezelő a következő tervet javasolta a lekérdezés végrehajtására:

A picture containing text, screenshot, font, number

Description automatically generated

* Első lépésként HASH JOIN -t hajt végre **Mosodak** táblán. Full table access ssegítségével beolvassa az Rendelesek tábla adatait. **Index range scan** segítségével kiválogatja az indexértékek közül azokat, amelyek az access predicates feltételnek megfelelnek.
* Második lépésként HASH JOIN -t hajt végre a **Rendeles\_reszletek** táblára és összekapcsolja a korábban kiválasztott táblával.
* Harmadik lépésként HASH JOIN -t hajt végre a **Szolgaltatasok** táblára és összekapcsolja a korábban kiválasztott táblával.
* Negyedik lépésként HASH JOIN -t hajt végre a **Dolgozok** táblára és összekapcsolja a korábban kiválasztott táblával.

Végeredmény : **1522 cost**

### Optimalizáció

* A táblák összekapcsolásához INNER JOIN -t használtam.

Az optimalizáció után a következő végrehajtási tervet javasolta az adatbázis kezelő:

A picture containing text, screenshot, number, font

Description automatically generated

* Elsőként Nested Loops segítségével összekpcsolja a **Rendelesek** és a **Rendeles\_reszletek** táblát.
* Ennek az eredményét Index Unique scan segítségével kiválogatja az indexértékek közül azokat, amelyek megfelelnek az access predicates feltételnek.
* Harmadik lépésként HASH JOIN -t hajt végre a **Szolgaltatasok** a **Rendeles\_reszletek** táblára, ami access predicates -t használ a két tábla összekapcsolására.
* Negyedik lépésként HASH JOIN -t hajt végre a **Dolgozok** táblára, amihez filter predicates -t használ a két tábla összekapcsolására.

Végeredmény: **1368 cost**

# NoSQL adatbázis kezelés

A NoSQL adatbázis kiválasztásakor a ***MongoDB***-re esett a választásom. Ennek oka, hogy az adatbázisban tárolt adatok szerkezete inkább dokumentum jellegű, így ennek használatát tartom célszerűnek.

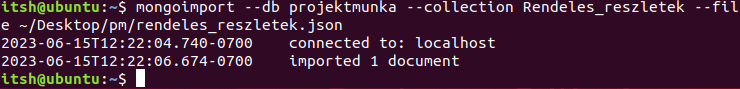
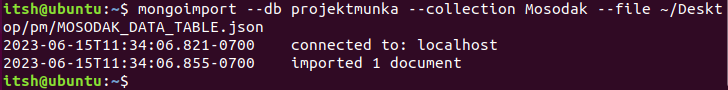
## Az adatok átvitele MongoDB-be

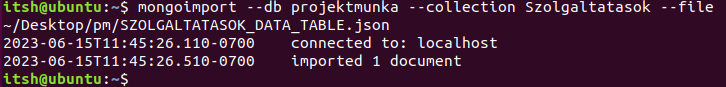
A gyűjtemények szerkezetének megtervezése után importálom az adatokat. Ehhez felhasználtam az ***SQL Developer*** exportálás funkcióját, ami a táblákban található adatokat ***json*** formátumban elmentette, amiket így egyszerűen tudtam importálni ***MongoDB***-be.

A screen shot of a computer

Description automatically generated with low confidenceA screen shot of a computer

Description automatically generated with low confidenceA screenshot of a computer screen

Description automatically generated with low confidenceA screenshot of a computer program

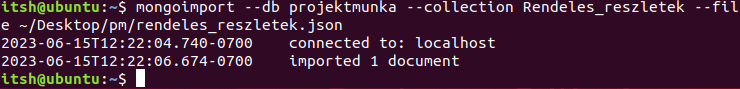
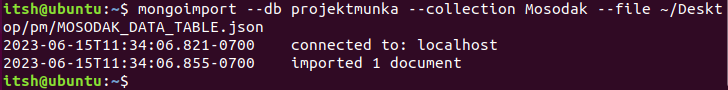
Description automatically generated with medium confidenceA screenshot of a computer screen

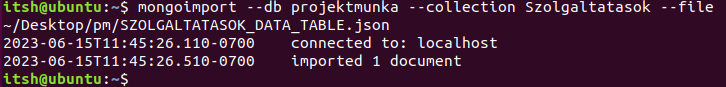
Description automatically generated with low confidenceA picture containing text, screenshot, font

Description automatically generatedA screen shot of a computer

Description automatically generated with low confidenceA screen shot of a computer

Description automatically generated with low confidenceA screenshot of a computer screen

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer screen

Description automatically generated with low confidenceA picture containing text, screenshot, font

Description automatically generated

## Lekérdezések

A lekérdezések szöveges formátumban megtalálhatóak a mellékletben.

### Mely vársoban található mosoda?

A screenshot of a computer

Description automatically generated with medium confidence

### Dolgozók átlagfizetése:

A screenshot of a computer program

Description automatically generated with low confidence

### dolgozók összfizetése: A screenshot of a computer program Description automatically generated with low confidence

### Mosodák átlag tisztasága:

A screenshot of a computer program

Description automatically generated with medium confidence

### Mosodák átlagos biztonsága:A screenshot of a computer program Description automatically generated with medium confidence

### Mosodák átlagos elrendezése:

A screenshot of a computer program

Description automatically generated with medium confidence

### szolgáltatások átlagos ára:

A screenshot of a computer program

Description automatically generated with medium confidenceMellékletek

## Táblákat létrehozó script

1. CREATE TABLE varosok (
2. VAROS\_ID INT PRIMARY KEY,
3. VAROS\_NEV VARCHAR(100)
4. );
6. CREATE TABLE mosodak (
7. MOSODA\_ID INT PRIMARY KEY,
8. VAROS\_ID INT,
9. MOSODA\_NEV VARCHAR(100),
10. MOSODA\_CIM VARCHAR(100),
11. MOSODA\_TELEFON VARCHAR(20),
12. FOREIGN KEY (VAROS\_ID) REFERENCES Varosok(VAROS\_ID)
13. );
15. CREATE TABLE dolgozok (
16. DOLGOZO\_ID INT PRIMARY KEY,
17. MOSODA\_ID INT,
18. NEV VARCHAR(100),
19. MUNKAKOR VARCHAR(100),
20. FIZETES INT,
21. FOREIGN KEY (MOSODA\_ID) REFERENCES Mosodak(MOSODA\_ID)
22. );
24. CREATE TABLE gepek (
25. GEP\_ID INT PRIMARY KEY,
26. MOSODA\_ID INT,
27. GEP\_TIPUS VARCHAR(100),
28. KAPACITAS INT,
29. FOREIGN KEY (MOSODA\_ID) REFERENCES Mosodak(MOSODA\_ID)
30. );
32. CREATE TABLE ugyfelek (
33. UGYFEL\_ID INT PRIMARY KEY,
34. NEV VARCHAR(100),
35. CIM VARCHAR(100),
36. TELEFON VARCHAR(20)
37. );
39. CREATE TABLE rendelesek (
40. RENDELES\_ID INT PRIMARY KEY,
41. UGYFEL\_ID INT,
42. MOSODA\_ID INT,
43. DATUM DATE,
44. OSSZEG INT,
45. FOREIGN KEY (UGYFEL\_ID) REFERENCES Ugyfelek(UGYFEL\_ID),
46. FOREIGN KEY (MOSODA\_ID) REFERENCES Mosodak(MOSODA\_ID)
47. );
49. CREATE TABLE rendeles\_reszletek (
50. RENDELES\_RESZLET\_ID INT PRIMARY KEY,
51. RENDELES\_ID INT,
52. SZOLGALTATAS\_ID INT,
53. MENNYISEG INT,
54. FOREIGN KEY (RENDELES\_ID) REFERENCES Rendelesek(RENDELES\_ID),
55. FOREIGN KEY (SZOLGALTATAS\_ID) REFERENCES
56. Szolgaltatasok(SZOLGALTATAS\_ID)
57. );
59. CREATE TABLE ertekelesek (
60. ERTEKELES\_ID INT PRIMARY KEY,
61. UGYFEL\_ID INT,
62. MOSODA\_ID INT,
63. DATUM DATE,
64. TISZTASAG INT,
65. BIZTONSAG INT,
66. ELRENDEZES INT,
67. FOREIGN KEY (UGYFEL\_ID) REFERENCES Ugyfelek(UGYFEL\_ID),
68. FOREIGN KEY (MOSODA\_ID) REFERENCES Mosodak(MOSODA\_ID)
69. );
71. CREATE TABLE szolgaltatsok (
72. SZOLGALTATAS\_ID INT PRIMARY KEY,
73. MOSODA\_ID INT,
74. SZOLGALTATAS\_NEV VARCHAR(100),
75. AR INT,
76. FOREIGN KEY (MOSODA\_ID) REFERENCES Mosodak(MOSODA\_ID)
77. );

## Sequence

1. CREATE SEQUENCE varosok\_seq
2. START WITH 11
3. INCREMENT BY 1
4. NOCACHE
5. NOCYCLE;
7. CREATE SEQUENCE mosodak\_seq
8. START WITH 51
9. INCREMENT BY 1
10. NOCACHE
11. NOCYCLE;
13. CREATE SEQUENCE dolgozok\_seq
14. START WITH 101
15. INCREMENT BY 1
16. NOCACHE
17. NOCYCLE;
19. CREATE SEQUENCE gepek\_seq
20. START WITH 100001
21. INCREMENT BY 1
22. NOCACHE
23. NOCYCLE;
25. CREATE SEQUENCE ertekelesek\_seq
26. START WITH 5001
27. INCREMENT BY 1
28. NOCACHE
29. NOCYCLE;
31. CREATE SEQUENCE ugyfelek\_seq
32. START WITH 100001
33. INCREMENT BY 1
34. NOCACHE
35. NOCYCLE;
37. CREATE SEQUENCE szolgaltatasok\_seq
38. START WITH 151
39. INCREMENT BY 1
40. NOCACHE
41. NOCYCLE;
42. CREATE SEGUQNCE rendelesek\_seq
43. START WITH 225001
44. INCREMENT BY 1
45. NOCACHE
46. NOCYCLE;
47. CREATE SEGUQNCE rendeles\_reszletek\_seq
48. START WITH 225001
49. INCREMENT BY 1
50. NOCACHE
51. NOCYCLE

1. A táblákat létrehozó script megtalálható a mellékletben, a tartalomjegyzék végén. A modell a ***dbdesigner.net*** használatával készült. [↑](#footnote-ref-1)