Vizsgaremek

# Hallgató neve: Polgár Tibor

# Hallgató e-mail címe: polgartibor76@gmail.com

# A választott téma rövid elnevezése: SMT gyártósor adatbázisa

# A téma szöveges leírása

Egy SMT gyártósor (mely nyomtatott áramköri lapra beülteti és beforrasztja az alkatrészeket és vizuális ellenőrzést végez) adatbázisában nyomon követhető, hogy a gyártósoron elindult egységek mely állomásokon jártak, milyen műveleteket végeztek rajtuk, milyen mérési eredményeik voltak illetve milyen javítást történt az adott egységen.

Az adatbázishoz különböző felhasználók tartoznak, valamint önálló adatmentési stratégiával rendelkezik.

Azért választottam ezt a témát, mert dolgoztam gyártási környezetben és azt tapasztaltam, hogy a mai világban kiemelt fontosságú a vevők, megrendelők számára, az egységek, áruk munkafolyamatának minél teljesebb körű ismerete, általában már elektronikus formában dokumentálva, így biztosítva az elvárt minőség megvalósulását.

Az SMT gyártósor adatbázisa segítséget nyújt a folyamatmérnökök, minőségmérnökök számára is, a gyártás során fellépő hibák kiderítésére, vevői reklamáció esetén a hiba okának és helyének felderítésére, akár több napra, hónapra visszamenőleg. Az adatbázis segítségével kimutatások készíthetők melyek különböző minőségügyi módszerekben felhasználhatók.

Az adatbázis táblákat a gyártógépeken futó programok (erre a célra készített vagy gyártói programok) töltik fel, automatikusan rögzítve az elvégzett munkafolyamatot.  
Emberi beavatkozás és annak rögzítése, úgymint hibák javítása vagy egységek mozgatása, történhet:

- közvetlenül az adatbázis eljárások meghívásával, insertek, updatek használatával pl.: SSMS segítségével.

- külön, erre a célra írt programok segítségével, melyek maguk kommunikálnak az adatbázissal eljárások meghívásával, insertekkel, updatekkel. Ez a megoldás az ajánlott, amely, bár külön programot igényel, a felhasználók számára könnyebben kezelhető, átláthatóbb.

Az adatbázis főbb táblái:

PROCESS – Az egységek aktuális állapota, mely állomáson milyen állapotban vannak  
PROCESSLOG – A PROCESS tábla trigger által töltött táblája

ORDERS – Megrendelések táblája

MEASUREMENT – Mérési adatok táblája

REPAIR – Az elvégzett javítások táblája

A táblatartalmak megfelelőségéről idegen kulcsok gondoskodnak, így csak úgy bármilyen adat nem vehető fel, mert nem fog megfelelni a hivatkozásoknak. A PROCESS táblán egy trigger is van, ami nem engedi az abból való törlést, és logolja a PROCESSLOG táblába a PROCESS táblán végzett műveleteket.

# Adatbázis diagram

# 

# Sémák Nem lettek külön sémák létrehozva, mivel kevés a tábla, így nincs értelme külön sémákba tenni őket. Kb. 20 táblától lenne értelme, hogy külön csoportokba foglaljuk őket, úgymint megrendelések, mérések, felhasználók, állapot.

# **Táblák**

## Tábla neve: dbo.MEASUREMENT

Leírás: Mérési eredmények tárolása

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Measure:id | int IDENTITY(1,1) | - | - | Mérés egyedi azonosítója | - |
| Unit\_id | varchar(30) | NOT NULL | - | Egység azonosító | - |
| Operation\_id | tinyint | NOT NULL | - | Operácó/művelet azonosít | - |
| Measure\_name | varchar(30) | - | - | Mérési lépés neve | - |
| Measure\_result | varchar(15) | - | - | Mérési lépés érték eredménye | - |
| Unitofmeasure | varchar(15) | - | - | Mértékegység | - |
| Evaluation | varchar(8) | - | - | Eredmény kiértékelése | 'Pass','Fail','N/A' érték lehet |
| Measure\_date | datetime2 | - | - | Mérés időpontja |  |

### Táblaszintű megszorítások

|  |  |  |
| --- | --- | --- |
| Megszorítás neve | Megszorítás fajtája  (trigger, constraint) | Értelmezés |
| CK\_\_MEASUREME\_\_Evalu\_\_300424B4k | CONSTRAINT | A mérés kiértékelés csak Pass (jó), Fail (rossz) vagy N/A nem értelmezett lehet |

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_MEASUREMENTId | Measure\_id | Clustered | Elsődleges kulcs |
| IX\_Measurement\_UnitId | Unit\_id | Nonclustered | Az unit\_id nem lehetett Primary key, de legyen rá index, mert fő keresési adat. |

### Triggerek Nincs

### Táblakapcsolatok

|  |  |  |
| --- | --- | --- |
| Idegen kulcs neve | Kapcsolat típusa | Mely táblára / Delete és Update szabály |
| FK\_MEASUREMENTUnitId | 1 : N | PROCESS / Nincs |
| FK\_MEASUREMENTOperation | 1 : N | ORDERROUTING / Nincs |

Tábla neve: dbo.ORDERROUTING

Leírás: A megrendelésekhez (order) tartozható összes munkafolyamat definiálása

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Operation\_id | tinyint | NOT NULL | - | Operáció/művelet azonosítja | - |
| Operation\_name | varchar(30) | - | - | A művelet neve | - |

### Táblaszintű megszorítások

Nincs

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_OrderRoutingId | Operation\_id | Clustered | Elsődleges kulcs |

### Triggerek

### Nincs

### Táblakapcsolatok

Nincs

## Tábla neve: dbo.ORDERS

Leírás: A megrendelések adatai

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Order\_id | int | NOT NULL | - | Megrendelés azonosító | - |
| Material\_number | varchar(25) | NOT NULL | - | Anyagszám | - |
| Order\_initialqty | int | NOT NULL | - | Megrendelés mennyiség kezdőérték | - |
| Order\_actualqty | int |  | 0 | Aktuális érték | - |
| Date\_start | date | NOT NULL | - | Tervezett kezdési dátum | - |
| Date\_planned\_end | date | NOT NULL | - | Tervezett befejezési dátum | - |

### Táblaszintű megszorítások

|  |  |  |
| --- | --- | --- |
| Megszorítás neve | Megszorítás fajtája  (trigger, constraint) | Értelmezés |
| CHK\_Date | CONSTRAINT | [Date\_planned\_end]>=[Date\_start] |

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_OrderOrderId | Order\_id | Clustered | Elsődleges kulcs |

### Triggerek

### Nincs

### Táblakapcsolatok

## Nincs

## Tábla neve: dbo.ORDERSWITCH

Leírás: Kapcsolat az ORDERS és ORDERROUTING tábla közt, itt definiálható mely megrendeléshez mely munkafolyamatok tartoznak, mert előfordulhat, hogy nem kell mind a gyártáshoz. (pl.: Glue nem kell)

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Order\_id | int | NOT NULL | - | Megrendelés azonosító | - |
| Operation\_id | tinyint | NOT NULL | - | Operáció szám az ORDERROUTING táblából | - |

### Táblaszintű megszorítások

Nincs

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_OrderSwitch | Order\_id, Operation\_id | Clustered | Elsődleges kulcs |

### Triggerek

### Nincs

### Táblakapcsolatok

|  |  |  |
| --- | --- | --- |
| Idegen kulcs neve | Kapcsolat típusa | Mely táblára / Delete és Update szabály |
| FK\_OrderSwitch | 1 : N | ORDERS / Nincs |
| FK\_RoutingSwitch | 1 : 1 | ORDERROUTING / Nincs |

## Tábla neve: dbo.PROCESS

Leírás: A egységek aktuális helyzetét, állapotát tárolja. Mely orderben, mely állomáson van, milyen állapotban, ki végezte az adatmódosítást.

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Unit\_id | varchar(30) | NOT NULL | - | Egység azonosító | - |
| Order\_id | int | NOT NULL | - | Megrendelés | - |
| Operation\_id | tinyint | NOT NULL | - | Művelet azonosítója | - |
| Status\_unit | varchar(10) | NOT NULL | - | Állapot | 'Checkin','Pass', 'Fail','Scrap'  lehet csak |
| Update\_time | datetime2 | NOT NULL | - | Módosítás dátuma | - |
| Updater\_name | varchar(20) | NOT NULL | - | Módosító User neve | - |

### Táblaszintű megszorítások

|  |  |  |
| --- | --- | --- |
| Megszorítás neve | Megszorítás fajtája  (trigger, constraint) | Értelmezés |
| CK\_\_PROCESS\_\_Status\_\_\_276EDEB3 | Check | A status\_unit mező „értéke” csak 'Checkin','Pass','Fail','Scrap' lehet |

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_PROCESSUnitId | Unit\_id | Clustered | Elsődleges kulcs |

### Triggerek

|  |  |  |
| --- | --- | --- |
| Trigger neve | Trigger típusa | Értelmezés |
| trgProcessLog | DML (INSERT, UPDATE, DELETE) | Tiltja a törlést. A táblába való insert és update műveleteket logolja a PROCESSLOG táblába |

### Táblakapcsolatok

|  |  |  |
| --- | --- | --- |
| Idegen kulcs neve | Kapcsolat típusa | Mely táblára / Delete és Update szabály |
| FK\_PROCESSOrderOrder | 1 : 1 | ORDERS / Nincs |
| FK\_PROCESSOrderOperation | 1 : 1 | ORDERROUTING / Nincs |

## Tábla neve: dbo.PROCESSLOG

Leírás: A PROCESS tábla log táblája, mely tartalmazza milyen művelet történt a PROCESS táblán

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Id | int IDENTITY(1,1) | - | - | Sorszám |  |
| Unit\_id | varchar(30) | NOT NULL | - | Egység azonosító | - |
| Order\_id | int | NOT NULL | - | Megrendelés | - |
| Operation\_id | tinyint | NOT NULL | - | Művelet azonosítója | - |
| Status\_unit | varchar(10) | NOT NULL | - | Állapot | - |
| Update\_time | datetime2 | NOT NULL | - | Módosítás dátuma | - |
| Updater\_name | varchar(20) | NOT NULL | - | Módosító neve | - |
| Action | varchar(10) | NOT NULL | - | Táblaművelet neve | - |

### Táblaszintű megszorítások

*Nincs*

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_PROCESSLOGId | Id | Clustered | Elsődleges kulcs |
| IX\_Processlog\_UnitId | Unit\_id | Non-Clustered | Keresés Unit\_id-ra |

### Triggerek

*Nincs*

### Táblakapcsolatok

|  |  |  |
| --- | --- | --- |
| Idegen kulcs neve | Kapcsolat típusa | Mely táblára / Delete és Update szabály |
| FK\_PROCESSLOGUnit | 1 : N | PROCESS / Nincs |

## Tábla neve: dbo.REPAIR

Leírás: A javítási adatok tárolására (A MEASUREMENT tábla Fail-os egységeinek javítási eredményei)

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Unit\_id | varchar(30) | NOT NULL | - | Egység azonosító | - |
| Fail\_id | int | NOT NULL | - | Javítási azonosító (megegyezik a MEASUREMENT tábla Id-jával) | - |
| Reair\_code | tinyint | NOT NULL | - | Mivel lett javítva kód | - |
| User\_id | int | NOT NULL | - | Javítást végző ID-ja | - |
| Repir\_date | datetime2 | NOT NULL | - | Javítás időpontja | - |

### Táblaszintű megszorítások

*Nincs*

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_REPAIRId | Fail\_id | Clustered | Elsődleges kulcs |
| IX\_Repair\_UnitId | Unit\_id | Nonclustered | Az unit\_id nem lehetett Primary key, de legyen rá index, mert fő keresési adat. |

### Triggerek Nincs

### Táblakapcsolatok

|  |  |  |
| --- | --- | --- |
| Idegen kulcs neve | Kapcsolat típusa | Mely táblára / Delete és Update szabály |
| FK\_REPAIRPROCESSUnitid | 1 : N | PROCESS / Nincs |
| FK\_EVAUSERId | 1 : N | USERS / Nincs |
| FK\_REPAIRREPAIRDef | 1 : N | REPAIRDEF / Nincs |
| FK\_REPAIRFailid | 1 : 1 | MEASUREMENT / Nincs |

## Tábla neve: dbo.REPAIRDEF

Leírás: A javított hibák / javítási módok definiálása

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Repair\_code | tinyint | NOT NULL UNIQUE | - | A javítási művelet egyedi azonosítója | - |
| Repair\_Definition | varchar(30) | - | - | A javítási művelet leírása, definíciója | - |

### Táblaszintű megszorítások

Nincs

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_REPAIRDEFCODE | Reapir\_code | Clustered | Elsődleges kulcs |

### Triggerek Nincs

### Táblakapcsolatok

Nincs

## Tábla neve: dbo.RIGHTS

Leírás: Felhasználói jogok tábla. Az adatbázis login/user-ektől függetlenül szükséges, más jellegű.

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| Right\_id | tinyint | NOT NULL | - | A jog egyedi azonosítója | - |
| Right\_Definition | varchar(30) | - | - | A jog leírása, definíciója | - |

### Táblaszintű megszorítások

Nincs

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_RIGHTSId | Right\_id | Clustered | Elsődleges kulcs |

### Triggerek Nincs

### Táblakapcsolatok

Nincs

## Tábla neve: dbo.USERS

Leírás: A felhasználók adatainak táblája.

### Mezőleírás

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oszlop neve | Adattípus | Kötelező | Default | Értelmezés | Ellenőrzés |
| User\_id | int | NOT NULL | - | Felhasználó azonosító szám | - |
| User\_name | varchar(20) | NOT NULL, UNIQUE | - | Username (nem a server/db udername) | - |
| Right\_id | tinyint | NOT NULL | - | A hozzátartozó jog ID-ja | - |
| Real\_Name | varchar(50) | NOT NULL | - | A valós teljes név | - |
| Department | varchar(40) | NOT NULL | - | Részleg, ahol dolgozik | - |

### Táblaszintű megszorítások

*Nincs*

### Indexek

|  |  |  |  |
| --- | --- | --- | --- |
| Index neve | Oszlop (ok) | Index típusa | Értelmezés |
| PK\_USERID | User\_Id | Clustered | Elsődleges kulcs |

### Triggerekw

*Nincs*

### Táblakapcsolatok

|  |  |  |
| --- | --- | --- |
| Idegen kulcs neve | Kapcsolat típusa | Mely táblára / Delete és Update szabály |
| FK\_USERRIGHT | 1 : 1 | RIGHTS / Nincs |

# Nézetek

## Nézet neve: dbo.viewMeasurementRepair

### Nézet leírása:

A nézet visszaadja, hogy a fail-os mérések (MESUREMENT tábla) ki által, és milyen hibával lettek javítva.

Létrehozás oka, hogy így lekérdezhető, hogy egy egységen milyen javítások történtek (Unit\_id), vagy egy dolgozó milyen javításokat végzett (User\_name).

## Nézet neve: dbo.viewOneMounthOrderUnit

### Nézet leírása:

Az 1 hónapnál nem régebbi Date\_Start orderekben milyen egységek vannak, melyik állomáson, milyen anyagszámmal.   
Létrehozás oka, láthatóvá tenni, mely egységek gyártása van folyamatban. Operációra szűrve (Operation\_id) megtudható, hol mennyi egység áll. Az utolsó operációra szűrve (jelenleg 190) lekérdezhető, mely egységek gyártása van folyamatban, és nem az utolsó állomáson (legyártva) állnak.

## Nézet neve: dbo.viewNotFinishTooOldUnit

### Nézet leírása:

Melyek azok az egységek, amelyek nem az utolsó állomáson állnak és az „order planned end” dátumán már túl vagyunk. A megrendelés gyártása befejeződött, de lehetnek egységek, amiken nem fejeződött be a gyártás valami okból, ezeket listázza ki.

Létrehozás oka, hogy hiába fejeződött be a megrendelés (order) gyártása, ha nincs meg a szükséges darabszám. Így utána lehet nézni, a hiányzó egységek hol is tartanak a gyártási folyamatban. Adott orderre (Order\_id) keresve megkapható az abban még gyártás alatt lévő egységek.

# Függvények

## Függvény neve: dbo.FuncOrderCount

### A függvény funkcionalitásának leírása:

Ez egy skalár függvény, ami azt vizsgálja meg, hogy egy adott orderben van-e még hely (az értéke 1 vagy nagyobb) és létre lehet benne hozni még egységet. A visszaadott érték 1 ha van hely, 0 ha nincs hely, és 2 ha nem létező orderbe akarnak létrehozni.

A PROCESS tábla trigger-je használja egység létrehozáskor, így ellenőrizve le, hogy az orderbe amibe létrehoznának van-e hely. Csak ez a trigger használja ezt a függvényt, de külön használata is lehetséges.

### Függvény paraméterei és visszaadott értéke

|  |  |  |  |
| --- | --- | --- | --- |
| Paraméter neve | Adattípus | Típus | Alapértelmezés |
| @order | int | Input | Nincs |
| Return érték | int | ReturnValue | 0,1,2 lehet az értéke |

# Tárolt eljárások

## Tárolt eljárás neve: dbo.UnitFailHistory

### A tárolt eljárás funkcionalitásának leírása:

Visszaadja egy egységről, hogy mely állomásokon volt 1-nél több hibája (ahol mérést végzünk). Az eljárás egy select eredményét adja vissza.

Egy egység elemzésénél használható, selejtezésre vonatkozó mérnöki döntést segíthet elő.

### Tárolt eljárás paraméterei és visszaadott értéke

|  |  |  |  |
| --- | --- | --- | --- |
| Paraméter neve | Adattípus | Típus | Alapértelmezett érték / és egyéb információk |
| @Unit\_id | varchar(30) | Input | Nincs / A keresett egység azonosítója |
| Return érték |  |  | Nincs, select eredményét adja vissza. |

## Tárolt eljárás neve: dbo.InsertRepair

### A tárolt eljárás funkcionalitásának leírása:

Külső „egységjavító” program ezen eljáráson át tudja a javítás eredményét beszúrni a REPAIR táblába.

2 féle használat lehetséges:

- Megadjuk a javítandó egység Measurement táblában szerepló javítandó mérésének measurement\_id-ját,

- vagy megadjuk, hogy a javítandó egység mely operáció azonosítón végzett, mely mérését akarjuk javítani.

Mindkét esetben a Measurement táblába léteznie kell a Fail-os mérésnek.

Az eljárás hibakezelést tartalmaz, ami hiba esetén megad egy üzenetet és az error\_number, error\_message-t.

### Tárolt eljárás paraméterei és visszaadott értéke

|  |  |  |  |
| --- | --- | --- | --- |
| Paraméter neve | Adattípus | Típus | Alapértelmezett érték / és egyéb információk |
| @Mode | tinyint | Input | NULL / Ha 1 akkor measurement\_id, ha 2 akkor Operation\_id & Measure\_name a használati mód |
| @Unit\_id | varchar(30) | Input | NULL / Egység azonosító |
| @Repair\_code | tinyint | Input | NULL / Javítás kódja. |
| @User\_id | int | Input | NULL / Javítást végző id-ja |
| @Mea\_id | int | Input | NULL / A failos mérés azonosítója, amit javítunk |
| @Op\_id | int | Input | NULL / Az operáció ahol a hiba van |
| @Mea\_name | varchar(30) | Input | NULL / A hiba 'neve' |
| Return érték | int | ReturnValue | 1 = Minden rendben volt  -1 = Sikertelen insertálás |

## Tárolt eljárás neve: dbo.InsertUser

### A tárolt eljárás funkcionalitásának leírása:

Új felhasználó felvitele az USER táblába a jog nevének, a leendő felhasználónév (egyedi kell legyen), valós név és részleg megadásával. Használható egyénileg meghívva, vagy külső program által. Az eljárás, beszúráskor a soron következő User\_id-val hozza létre a bejegyzést, így azt megadni nem kell.  
Az eljárás hibakezelést tartalmaz ami hiba esetén megad egy üzenetet és az error\_number, error\_message-t.

### Tárolt eljárás paraméterei és visszaadott értéke

|  |  |  |  |
| --- | --- | --- | --- |
| Paraméter neve | Adattípus | Típus | Alapértelmezett érték / és egyéb információk |
| @Rightdef | varchar(30) | Input | NULL / A megadandó jog neve |
| @Username | varchar(20) | Input | NULL / Létrehozandó felhasználónév |
| @Realname | varchar(50) | Input | NULL / Valós név |
| @Dep | varchar(20) | Input | NULL / Részleg, ahol dolgozik |
| Return érték | int | ReturnValue | 1 = Minden rendben volt  -1 = Sikertelen beszúrás |

# Jogosultsági rendszer

Az adatbázis „két féle” jogosultsági rendszert használ, „hagyományos” és RIGHTS tábla alapút.

A „hagyományos” adatbázis motor által alapból létrehozható loginok és user-ek.

A RIGHTS táblában definiálva vannak a gyártással kapcsolatos jogok, amik meghatározzák, hogy pontosan kinek, milyen szerepkörhöz van joga. Így magához az adatbázishoz kevesebb felhasználót kell létrehozni, ezáltal jobban módosítható, karbantartható, maga a jogok biztonságosabbak. (Igaz, bonyolultabb is.)

A két rendszer „és” kapcsolata határozza meg a megtehető műveleteket, természetesen abban az esetben, ha ez egy adott eljárásban nincs leprogramozva a ’saját’ jogkör vizsgálat (mert csak olvasást végez), akkor az adatbázis motor jogai a mérvadók.

Az adatbázisba belépés a hagyományos módon loginokkal történik, de az eljárások, függvények megkövetelhetik, hogy bemenő paraméterként meg kell adni az USERS tábla felhasználóját.

Egyszerűbben, egy külső program egy eljárás meghívásához egy loginnal belép az adatbázisba, amihez kapcsolódik egy user, de az eljárás meghívásakor paraméterben meg kell adni a specifikált (USERS tábla) felhasználót és azt kiértékelve fog lefutni vagy nem lefutni az eljárás.

Pl.:  
A dbo.UnitFailHistory eljárást „bárki” használhatja, de a dbo.InsertRepair eljáráshoz Repair jog kell.

## Login objektumok

Külön, a szerverre nincsenek megadva jogok (csak az alap public), minden loginhez user van társítva.

Admin login (jelszó: ADMIN) – SQL szerver Authentication. Default adatbázisa a Factory, ahol az Admin user van társítva hozzá és db\_owner joga van. Az adatbázis adminisztrátora.

Manipulation login (jelszó: MANIPULATION)- SQL szerver Authentication. Default adatbázisa a Factory, ahol a Manipulation user van társítva hozzá és user-ként kapott jogokat a Factory db-ben.

Processreader login (PROCESSREADER) - SQL szerver Authentication. Default adatbázisa a Factory ahol a Processreader user van társítva hozzá és userként a Processread database role-t használja.

Reader login (READER) - SQL szerver Authentication. Default adatbázisa a Factory ahol a Reader user van társítva hozzá

Repair login (REPAIR) - SQL szerver Authentication. Default adatbázisa a Factory ahol a Repair user van társítva hozzá

’SMT Service’ login (SMTSERVICE) – SQL szerver Authentication. Default adatbázisa a Factory ahol a ’SMT Service’ user van társítva hozzá. Ezt a logint használják a soron lévő gyártógépek az adatbázissal való kommunikációhoz.

## User objektumok

Admin – A db\_owner csoport tagja (Factory adatbázisra)

Manipulation – Nem tagja csoportnak, a PROCESS, PROCESSLOG táblákra van select, insert, update, delete joga. Egység állapot módosítására (létrehozás, update) szolgál. Ezt ’használja’ az egység módosító program.

Processreader – A saját készítésű Processread database role-t használja, ami a PROCESS és PROCESSLOG táblákra ad select futtatási jogot.

Reader – A db\_datareader csoport tagja. Selectek írásában jártas mérnökök számára lehet hasznos ez a user.

Repair – A MEASUREMENT,PROCESS,REPAIR,RIGHTS,USERS táblákra csak olvasási joga van, és az InsertRepair tárolt eljárást futtathatja. A Javítós program által használt user.

’SMT Service’ – A db\_datareader és db\_datawriter csoport tagja és futtathatja az InsertRepair és UnitFailHistory tárolt eljárásokat. Jellemzően az SMT sorokon lévő gyártógépek felhasználója.

A RIGHTS tábla jogai

Administrator - Adminisztrátor, mindent „megtehet”

User manupulator - Felhasználók felvétele, módosítása

Repair - Egységek javításához szükséges jog

Packing - Csomagolási jog, jelenleg még nincs szükség rá

RepairPacking - Egységjavításhoz és csomagoláshoz ad egyszerre jogot.

Unit Manipulation - Egységek állapotának módosítására ad jogot

## Database role és Application role objektumok

Database role neve: Processread

Leírása: Lehetővé teszi, hogy csak a Process és Processlog táblákat olvassa, kizárólag ezekre van select joga. A Processreader felhasználó használja.

Application role neve: Ordersadmin Jelszava: ORDERSADMIN

Leírása: Lehetővé teszi, hogy az orderekkel kapcsolatos táblákra (Orders,Orderrouting,Orderswitch) select,update,insert, delete utasítást lehessen végrehajtani.

# Telepítés

Az adatbázis telepítése .sql scriptek futtatásával lehetséges. A futtatandó file-ok és tartalmuk:

1\_Createdatabase.sql

Tartalmazza az adatbázis létrehozást a C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\ elérési útra, tartalmazza a táblák létrehozását, az indexeket, view-eket, függvényt, tárolt eljárásokat, loginokat, user-eket, a mentési stratégia létrehozó job-jait.

2\_Adatfeltoltes.sql

Feltölti kezdeti adatokkal a táblákat.

A fentiek futtatásával az adatbázis már használható.

Opcionális file-ok:

3\_minta\_egyseg.sql

Egy minta egység a valós működés szimulálásához, a script futása 10-15mp.

4\_view\_eljaras\_hasznalat.sql

Az adatbázisban definiált view-k, eljárások, trigger használatához néhány példa, amivel azok működése kipróbálható, tesztelhető.

# Mentési stratégia

A mentési stratégiát meghatározza a sor jövőbeli terheltsége, mikor van rajta gyártás (7/24) és milyen ciklus idejű terméket gyártanak rajta. A jelenlegi mentési stratégia a sor kezdeti, felfutási állapotára készült (nincs 24 órás gyártás, hétvégén éjjel nem gyártanak), a későbbiekben ez módosítható az igényeknek megfelelően. (pl.: sűrűbb log backup készítés).

Az automatizált mentés 3 job végzi, amit a SQL Server Agent-je kezel.

Hetente egyszer, vasárnap éjjel 11-kor készül egy Full Backup.

Naponta hajnali 2 órakor készül egy differenciál Backup.

Két óránként, éjfél után 20 perctől indulva (0:20 , 2:20 stb.) egy log backup készül.

Az adatbázis servert futtató hardvertől függően, az első nagyobb gyártástól érdemes a log backupok készítését óránkénti, vagy még sűrűbbre állítani, mert gyártási adatbázisként elvileg nem veszhet el adat, mivel a vevő felé minőségi követelmény a visszakereshetőség biztosítása.

Visszaállítás a backup könyvtárban található .bak file felhasználásával történik, ami tartalmazza a full, differencial és log backupokat, amiket a jobok már létrehoztak.

{A vizsgamunka repositorija egy kiindulási Full Backupot tartalmaz, melyből az adatbázis felépíthető, de a loginokat és jobokat külön kell létrehozni, hisz az nem az adatbázis része, hanem a szerveré (1\_Createdatabase.sql tartalmazza).}