

Eredu Erlazionala

INTEGRITATE-MURRIZTAPENAK

Indizea

1. Definizioa
2. Betearazteko ardura: aplikazioak vs. DBKSak
3. Atributu murriztapenak
4. Tupla barneko murriztapenak
5. Tupla arteko murriztapenak
6. Integritate-murriztapenen egoerak
7. Egiaztapena transakzioekin

Integritatea (Sendotasuna)

- DBren datuen **zuzentasuna** eta **zehaztasuna** adierazteko hitza
- DBn gordetako datuak okerrak izan daitezke:
 - sarreran errorea egon delako
 - softwareak edo hardwareak akatsen bat egin duelako
 - erabiltzaileak maltzurki jokatu duelako



Integritate-murriztapenak

1. Definizioa: Integritate-murritzapenak

- Definizioa. Integritate-murritzapenak (IM): Domeinuan betetzen diren baldintzak
- Adibideak:
 - Langile baten soldatak ezin du bere buruzagiarenak baino altuagoa izan
 - Lehen mailako ikasle batek programazioan matrikulatuta egon behar du
 - Banketxeko kontu batek ezin du saldo negatiboa izan
- IM-ek domeinuan betetzen diren baldintzak errepresentatzen badituzte, eta datu-baseak domeinuaren isla izan behar badu...
- Orduan: datu-baseak IMak bete egin behar ditu

Integritate-murriztapenak

Datu-basean egindako eragiketak behatzeko erregelak

- DBK-k eta DK-k definituko dituzte
 - DK-k enpresaren politikarekin zerikusia dutenak
- DBren hedapenak bete behar dituzte ➡ DB egoera sendoan
- Integritate-murriztapenak betearazteko ardura

Aplikazioek

DBKSk

Aplikazioek eta DBKSk

2. IM betearazteko ardura aplikazioan

- Sendotasuna ziurtatzeko, datuak atzitzen dituen aplikazio bakoitzak kode bera errepikatu behar du

Kodea errepikatuta

Mantenimendurako ahalegina handiagoa

Sarean datu-trafikoa handiagoa

- Agindua egikaritzeko atzitzen diren datuak
- Sendotasuna egiaztatzeke atzitzen diren datuak

IM betearazteko ardura aplikazioan (2)

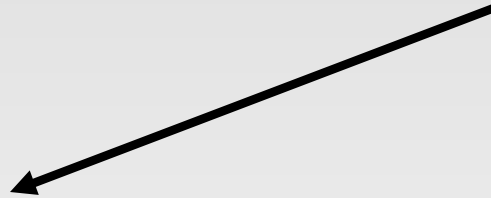
Adibidea

LANGILEA(na, izena, soldata, *szenb*)

SAILA(szenb, sizena, *zuzenna*)

Integritate-murriztapena:

Saileko zuzendariaren soldata gainontzeko langileen soldata baino handiagoa izan behar da



```
read(nax); read(izenax); read(soldatax); read(sailax);
```

```
#sql iter = {select soldata from Langilea where szenb = :sailax}
```

```
while (iter.next())
```

```
    if (iter.soldata() > soldatax) { errorea = true; break; }
```

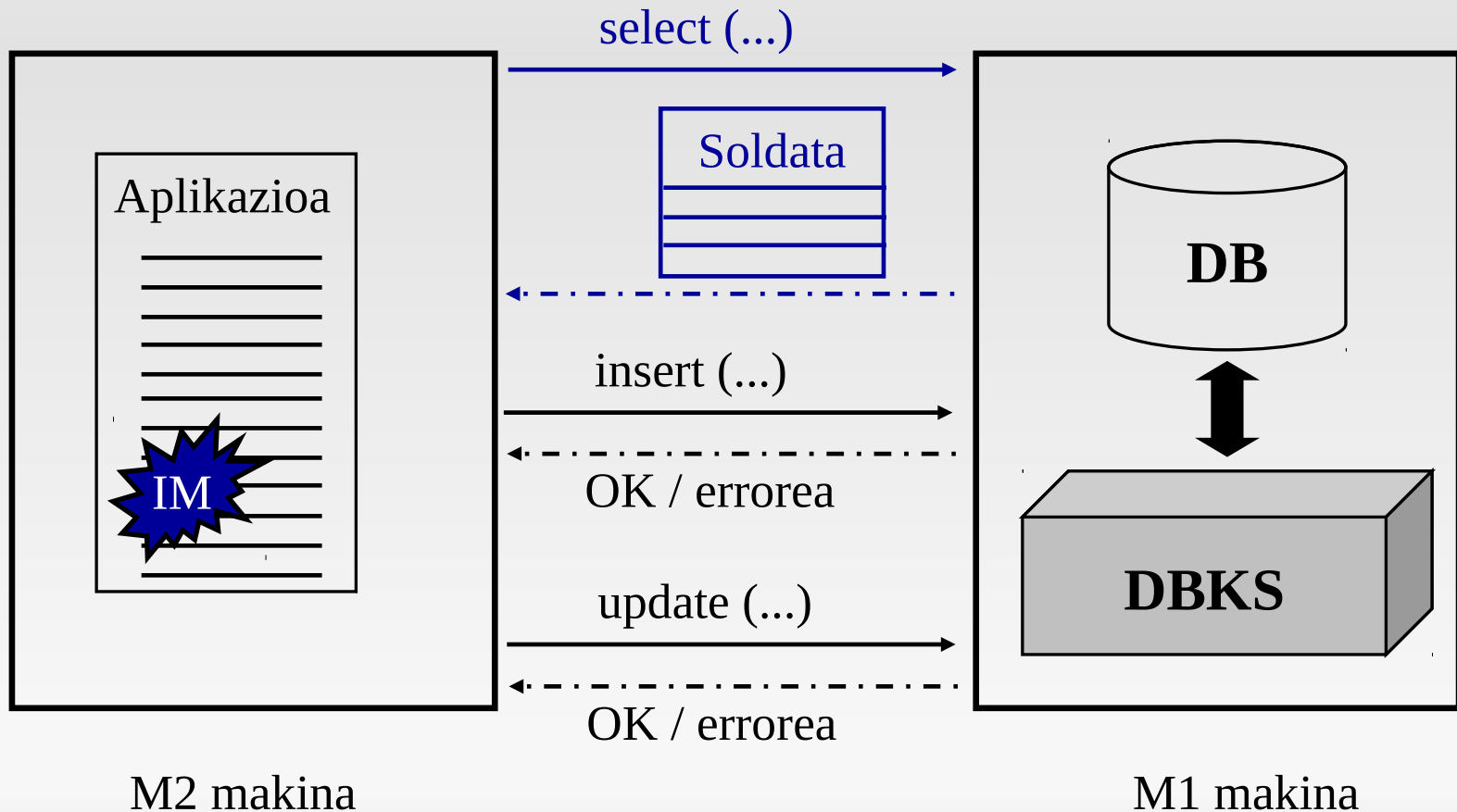
```
if ( !errorea ) {
```

```
    #sql {insert into Langilea values (:nax, :izenax, :soldatax, :sailax);
```

```
    #sql {update Saila set zuzenna = :nax where szenb = :sailax}; }
```

IM betearazteko ardura aplikazioan (3)

Adibidea



IM betearazteko ardura DBKSn

Abantailak

- Kodea ez da errepikatu behar, berrerabil daiteke
 - Baieztapenerako kodea toki bakarrean dago
- Kontsistentzia handitzen du
 - Aplikazio guztiei murriztapen berak “aplikatzen” zaizkie
- Mantentze-lana murrizten dute
 - Integritate-murriztapena aldatzean, DBKSan kodea aldatzen da soilik
- Eraginkortasuna handitzen da sareko trafikoa txikitzen baita
 - IMa baieztatzeako datuak ez doaz sarean zehar

IM betearazteko ardura DBKSn (2)

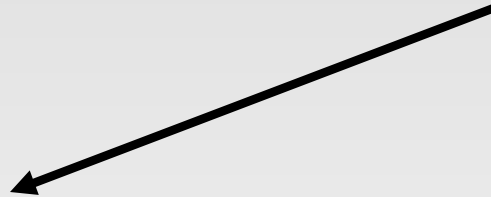
Adibidea

LANGILEA(na, izena, soldata, *szenb*)

SAILA(szenb, sizena, *zuzenna*)

Integritate-murriztapena:

Saileko zuzendariaren soldata gainontzeko langileen soldata baino handiagoa izan behar da



```
create assertion SoldatenKontrola
```

```
check (not exists (select *
```

```
    from Langilea as A
```

```
    where soldata > (select soldata
```

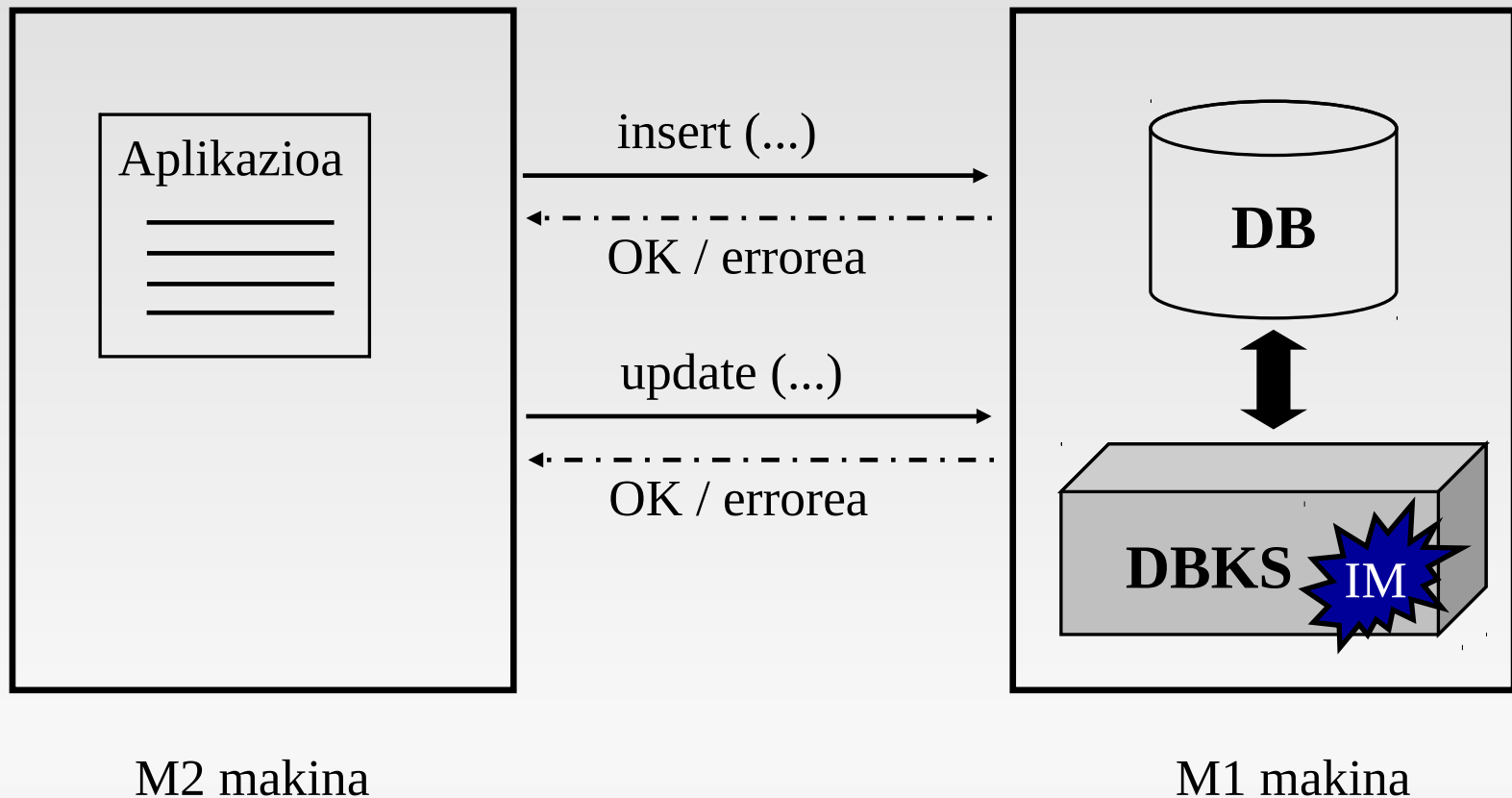
```
        from Langilea inner join saila as B
```

```
            on na=zuzenna
```

```
        where B.szenb=A.szenb))))
```

IM betearazteko ardura DBKSn (3)

Adibidea



Ondorioak

1. Murritzapen guztiak DBKSak egiaztatuko ditu
 - Inkoherentziak ekiditen ditu, ahanzteak ...
2. Horretaz gain, datu gehigarriak behar ez dituzten murritzapenak, aplikazioan egiaztatu ahal izango dira
 - Horrela bezeroan egiaztapena azkartuko da

Integritate-murriztapenak DBKSn

Atributuarenak

Taulako atributu batek har ditzakeen balioen gainean du eragina

Taulabarneko/Tupla barneko IM

Taula baten tupla bakoitzak bete behar dituen baldintzak adierazten ditu

Taularteko IM/Tupla arteko IM

- **Datu-baseak** bete behar dituen baldintzak adierazten ditu. Baldintza **tupla batzuren** gainera da, tuplak taula berekoak edo taula desberdinetakoak izan daitezkeelarik

Datuak Definitzeko Lengoaia (DDL): murriztapenen definizioa

- CONSTRAINT klausula (SQL2)
 - Taula batean definitzen da IMa
 - Baldintzak FALSE ematen badu hausten da
- ASSERTION klausula. Asertzioak (SQL2)
 - IMa unitate independente moduan zehazten da
 - Baldintzak FALSE ematen badu hausten da
- TRIGGER **abiarazleak** (SQL3)
 - IMa zehazten da, baita hainbat gertaera gertatu ondoren integritatea berreskuratzeko aurrera eraman beharreko *ekintzak* ere

3. Atributuaren integritate-murriztapenak

- Atributuarenak
 - Bakartasuna (unique)
 - Oinarrizko gakoarena (primary key)
 - Integritate erreferentziala (foreign key - references)
 - *Null* balioarena
 - Kontrola (check)
 - Domeinuarenak
 - Domeinu hori duen zutabe bakoitzari aplikatzen zaio (edozein tauletakoa), taulako murriztapen moduan lan eginez
 - Domeinuko balioei buruzko bilaketa baldintzaren bat dute

Atributuaren murriztapenak

- Bakartasuna: *unique*
 - Balio errepikatuak hartzea debekatzen du, hau da, taulako bi errenkadek ezin dute balio bera (*Null* ez dena) izan adierazitako zutabeen/zutabeetan
 - Zutabe guztien multzoari aplikatzea ahal da
- Oinarrizko gakoarena: **primary key**
 - Gakoaren parte diren atributuek ezin dute null baliorik hartu (entitate integritatea)
- Gako arrotzarenak: **foreign key**
 - references** <taula> [**on** (**update** | **delete**) (**cascade** | **set null** | **set default** | **no action**)]
 - Modu informalean: gako arrotz batetik beste taula bateko gako bati erreferentzia egiten bazaio, helburu-taulan gako arrotzean agertzen den balioak existitu behar du (integritate erreferentziala)
 - Helburu-taulako atributuek bakartasun murriztapen bat izan behar dute

Atributuaren murriztapenak

- Balioen tartea murrizteko: **CHECK (atributua IN ...)**
 - Bilaketa baldintza bat du eta honek adierazten den taulako elementuei soilik egin diezaioke erreferentzia
- Balioarenak:
 - Domeinuarenak:
 - sistemaren domeinuak: *char(30), int, date, time, ...*
 - erabiltzaileak definitutako domeinuak: *create domain*
 - *null* balioa hartzea debekatzeko: not *null*

Atributuaren murriztapenak. Adibideak (I)

- CREATE DOMAIN toxiko-mota AS NUMBER
CHECK (VALUE BETWEEN 1 AND 20);
- CREATE TABLE hondakina
(EEIFZ NUMBER NOT NULL ,
Hkodea CHAR(10) NOT NULL,
Izena CHAR(20),
Toxikotasuna toxiko-mota NOT NULL,
PRIMARY KEY (EEIFZ, Hkodea))
FOREIGN KEY (EEIFZ) REFERENCES enpresa_ekoizlea (IFZ)
ON DELETE CASCADE);

Atributuaren murriztapenak. Adibideak (II)

- CREATE TABLE osagaia
(Okodea NUMBER UNIQUE NOT NULL,
Izena CHAR(10) UNIQUE,
...);
- CREATE TABLE Garraioa (
EEIFZ NUMBER, ...,
Garraio-mota CHAR(10)
CHECK (Garraio-mota IN
(‘normala’, ‘erradiaktiboa’, ‘errekorra’)),
...)



PRIMARY KEY
jartzea bezala

Atributuaren murriztapenak. Egiaztapena

- Edozein tuplak atributuan balio berria jasotzen duen bakoitzean egiaztatuko da
- Tupla ezabatuz gero ez da egiaztatuko

Atributuaren murriztapenak.

Egiaztapena. *Null* balioa

- Murriztapen hausten da baldintzak **FALSE** ematen badu!! **TRUE** edo **UNKNOWN** bada ez da hausten

```
CREATE TABLE Hondakina (  
    ...  
    CONSTRAINT Bat  
        CHECK ( toxikotasuna IN (1, 2, 3, 4) ))
```

EEIFZ	Hkodea	izena	toxik.
E1	1	a	3
E2	3	b	null
E2	1	d	2



**Egiaztapena
betetzen da !!**

4. Tupla barneko murriztapenak

Atributuarena

Taulako atributu batek har ditzakeen balioen gainean du eragina

Taulabarneko IM/Tupla barneko IM

Taula baten tupla bakoitzak bete behar dituen baldintzak adierazten ditu

Taularteko IM/Tupla arteko IM

- Baldintza **tupla batzuren** gainera da, tuplak taula berekoak edo taula desberdinetakoak izan daitezkeelarik

Tupla barneko murriztapenak

```
CREATE/ALTER TABLE ...
```

```
...
```

```
CONSTRAINT <murriztapen-izena>
```

```
CHECK (<where motako baldintza>)
```

- Taulako **tuplek** bete behar duten baldintza
 - Izena ematea hautazkoa da (CONSTRAINT aginduaren bidez)
 - Behin asertzio anonimo bat definitu ondoren, gerora ezingo zaio izena eman

Tupla barneko murriztapenak

- Tupla bateko balioen gaineko baldintzak
- Adibideak:
 - Ikasle bat Gipuzkoakoa bada, bere telefono zenbakia “943” zenbakiekin hasten da
 - Ikasle batek KBPa badu, orduan 20 irakasgai gainditu ditu
- Baldintzak ondokoen gainean defini daitezke:
 - Tupla bereko beste atributu batzuk
 - Haien izenak erabiliz
 - Taula osoko edo beste tauletako ezaugarriak
 - Azpigoalderak erabiliz. **Ezin da erabili Oracle-n edo SQL Server-en**

**CHECK azpian azpikontsultekin teorian BAI,
praktikan DBKS anitzetan EZ!**

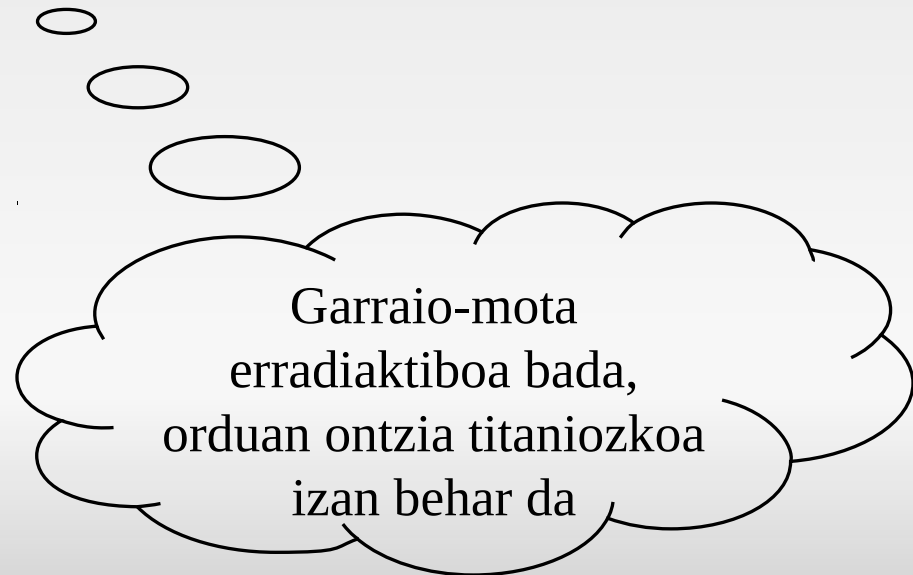
Tupla barneko murriztapenak. Adibidea

- CREATE TABLE Garraioa (
EEIFZ NUMBER, ...,
irteeraData DATE, heltzeData DATE,
garraioMota CHAR(10), ontzia CHAR(20), ...,
CHECK (heltzeData > irteeraData),
CHECK (garraioMota < > 'erradiaktiboa')
OR ontzia = 'Titaniozkoa')
);

garraioMota	ontzia
erradiaktiboa	Titaniozkoa
erradiaktiboa	Altzairuzkoa
normala	Titaniozkoa
normala	Altzairuzkoa



X



Tupla barneko murriztapenak. Egiaztapena

- Baldintza tupla aldatzean edo sartzean aztertzen da (tupla ezabatuz gero, ez da egiaztatuko)
- Check baldintza betetzen ez bada, aldaketa edo txertatzea ezabatzen da.
- Tupletako check baldintza bat ikustezina da beste erlazioetan. Azpigalderetan beste taulak baleude, horietan ez da check baldintza aztertzen.

5. Tupla arteko murriztapenak

CREATE **ASSERTION** <izena> **CHECK** <baldintza>

- Tupla batzuren gainekoa
- Taula osorik kontuan hartuta bete behar den baldintza
 - Funtzio agregaturen bat aplikatuz
- Taularen bat baino gehiago kontuan hartzen duen baldintza. Baldintzako taulak aldatzean, konprobatzen da...
- Egiaztapena: erreferentziatutako taularen batean aldaketa badago
- DBKS gehienek ez dute

Adib.: Lehen mailako ikasleek programazioan matrikulatu behar dute

$\forall x (\text{LehenMailakoIkaslea}(x) \rightarrow \text{ProgramazioanMatrikulatua}(x))$

$\neg \exists x \in \text{Ikaslea non } \neg (x.\text{bereMaila}=1^0 \rightarrow \exists y \in \text{Matrikula non } (x.\text{espZbkia}=y.\text{espZbkia} \wedge y.\text{Irakasgaia}=\text{'Programazioa'}))$

$\neg \exists x \in \text{Ikaslea non } \neg \neg (x.\text{bereMaila}=1^0 \wedge \neg \exists y \in \text{Matrikula non } (x.\text{espZbkia}=y.\text{espZbkia} \wedge y.\text{Irakasgaia}=\text{'Programazioa'}))$

```
CREATE ASSERTION LehenMailakoIkaslea CHECK (  
  NOT EXISTS (  
    SELECT * FROM Ikaslea AS X  
    WHERE X.Maila=1  
    AND NOT EXISTS  
      (SELECT * FROM Matrikula AS Y    WHERE X.espZbkia=Y.espZbkia  
        AND Irakasgaia='Programazioa'))
```

Taularteko murriztapena. Adibideak

- Murriztapena: toxikotasuna 15 baino handiago duten hondakinak titaniozko ontzitan garraiatu behar dira

```
CREATE ASSERTION garraio_arriskutsuak  
CHECK (NOT EXISTS  
      (SELECT *  
        FROM Garraioa NATURAL JOIN Hondakina  
        WHERE toxikotasuna > 15 AND ontzia < > 'Titaniozkoa'))
```

Taula artekoa

Tupla artekoa

$\neg \exists x,y (x.\text{toxikotasuna} > 15 \wedge y.\text{ontzia} \neq \text{'titaniozkoa'})$

$\forall x,y (x.\text{toxikotasuna} \leq 15 \vee y.\text{ontzia} = \text{'titaniozkoa'})$

Tupla barnekoa?

```
CREATE TABLE Hondakina
```

...

```
CONSTRAINT garraio_arriskutsuak CHECK
```

```
(toxikotasuna ≤ 15 OR (EEIFZ, Hkodea) IN
```

```
(SELECT EEIFZ, Hkodea
```

```
FROM Garraioa
```

```
WHERE ontzia = 'Titaniozkoa'))
```



HONDAKINA

EEIFZ	Hkodea	...	toxik
-------	--------	-----	-------

...

E1	3		20
----	---	--	----

...

GARRAIOA

EEIFZ	Hkodea	...	ontzia
-------	--------	-----	--------

...

E1	3		titanio
----	---	--	---------

...

UPDATE Hondakina SET toxikotasuna=10 WHERE EEIFZ=E1 AND Hkodea=3



UPDATE Hondakina SET toxikotasuna=25 WHERE EEIFZ=E1 AND Hkodea=3



UPDATE Garraioa SET ontzia='altzairua' WHERE EEIFZ=E1 AND Hkodea=3



Tupla barneko vs. Tupla arteko murriztapena.

Adibidea

PELIKULA(pkodea, ..., kostua, salneurria)

DISKOA(dkodea, ..., kostua, salneurria)

Murriztapena: Biltegian ditugun pelikulak eta diskoak erosteko ezin da 5.000.000 baino gehiago gastatu

Nola definitu?

```
CREATE TABLE pelikula(  
    ...  
    CONSTRAINT kontrola  
        CHECK ( (SELECT SUM(kostua) FROM pelikula) +  
                (SELECT SUM(kostua) FROM diskoa) <= 5.000.000))
```

Taula barnekoa

```
CREATE ASSERTION kontrola  
    CHECK ( (SELECT SUM(kostua) FROM pelikula) +  
            (SELECT SUM(kostua) FROM diskoa) <= 5.000.000 ))
```

Taula artekoa

Murritzapen-motak: alderaketa

Murritzapen-mota	Non erazagutu?	Noiz egiaztatu?	Murritzapena betetzea bermaturik ?
atributuarena	atributuarekin	Tupla berria sartzean Atributua eguneratzean <i>Tupla ezabatzean EZ</i>	Ez, taulak daturik ez badu
Tupla barnekoa	taularekin	Tupla berria sartzean Tupla eguneratzean <i>Tupla ezabatzean EZ</i>	Ez, definizioan azpi-galdera badago eta bertako taularen bat bada aldatu dena
Tupla artekoa	<i>ASSERTION</i> moduan	Erreferentziatutako taularen bat aldatzean	Bai

Murritzapena noiz egiaztatzen da

Ikasle bat Gipuzkoakoa bada bere telefonoa 943-z hasten da

```
CREATE TABLE Ikaslea (  
    probintzia CHAR(30) CHECK (probintzia in ('Gipuzkoa', ...));,  
    tfnoa CHAR(9)
```

```
    CONSTRAINT GipuzkoakoTfnoa  
        CHECK NOT  
            ((probintzia = "Gipuzkoa") AND NOT (tfnoa LIKE '943%'))))
```

- Noiz egiaztatzen da:
 - UPDATE(probintzia, tfnoa)
 - INSERT(Ikaslea)

Check VS. Asertzioa

- CHECK tuplak sartzean eta eguneratzean SOILIK egiaztatzen da
- Diseinatzailea ziur dagoenean murriztapena tuplak txertatzean eta eguneratzean SOILIK hautsiko dela, orduan CHECK erabili atributu, domeinu eta tupletan
- Bestelako kasuetan asertzioa erabili

Murritzapenak izendatu

- Beharrezkoa:
 - murritzapena ez betez gero, errore-mezu bereziak ateratzeko
 - murritzapena aldatu edo ezabatzeko
- Izenik gabe sortutako murritzapenari ezingo zaio gero eman

CONSTRAINT <identif> CHECK <baldintza>
--

- Okodea NUMBER CONSTRAINT OkodeaGakoaDa PRIMARY KEY
- Garraio-mota CHAR(20) CONSTRAINT motak-aukeran
CHECK (Garraio-mota IN ('normala', 'erradiak', 'errekorra'))
- CREATE DOMAIN toxiko-mota NUMBER
CONSTRAINT balioposible CHECK (VALUE BETWEEN 1 AND 20);

Murritzapenak gehitu eta ezabatu

ALTER (TABLE | DOMAIN) <ident> DROP CONSTRAINT <ident>

ALTER (TABLE | DOMAIN) <ident>

ADD CONSTRAINT <ident> CHECK <definizioa>

DROP ASSERTION <ident>

- ALTER TABLE Garraioa DROP CONSTRAINT motak-aukeran;
- ALTER DOMAIN toxiko-mota DROP CONSTRAINT balioposible;
- ALTER TABLE Garraioa ADD CONSTRAINT beste-mota-aukeran
CHECK (Garraio-mota IN ('normala', 'erradiak', 'errekorra', 'leher'));
- DROP ASSERTION garraio_arriskutsuak;

Taula batzuei eragiten dien IMa

- Asertzio moduan definitzen da
- Baina, ordenagailuan probatu nahi dugunean, **DBKS gehienek ez dituzte onartzen**
- Zer egin?
 - IMak eragiten duen taularen batera lotu eta taula barneko bihurtu
 - **Abiarazleak definitu**

Adib.: ezin du ikaslerik egon 10 irakasgai baino gehiago suspendituak dituenik

1.- **CREATE ASSERTION EZ+10**
CHECK (10 >= ALL (SELECT COUNT(Irakasgaia)
FROM Matrikula
WHERE nota < 5
GROUP BY ikasleHori))

Egiaztatzen da: eragiketak(Matrikula)

2.- **CREATE TABLE IKASLEA**

.....
CONSTRAINT EZ+10
CHECK NOT (izena IN (SELECT ikasleHori
FROM Matrikula
WHERE nota < 5
GROUP BY ikasleHori
HAVING COUNT(Irakasgaia) >10))

Egiaztatzen da: UPDATE(izena) INSERT(Ikaslea)
EZ da egiaztatzen: DELETE(Ikaslea), eragiketak(Matrikula)

3.- **CREATE TABLE MATRIKULA**
...
CONSTRAINT EZ+10
CHECK (10 >= ALL (SELECT COUNT(Irakasgaia)
FROM Matrikula
WHERE nota < 5
GROUP BY ikasleHori))

Egiaztatzen da: INSERT (Matrikula)
EZ da egiaztatzen: DELETE (Matrikula)

6. Integritate-murriztapenen egoerak (I)

```
CONSTRAINT <ident> CHECK <baldintza>  
[ DISABLE | ENABLE [VALIDATE | NOVALIDATE] ]
```

■ DISABLE

- Murriztapena **ez** da jadanik gordetako datuen gainean egiaztatuko. **Definizioa zuzenean katalogoan gordeko da**
- Sartuko diren ondorengo tupla berriekin ere ez da egiaztatuko

■ DISABLE VALIDATE

- Definizioa katalogoan gordetzeko, **murriztapena jadanik gordetako datuen gainean egiaztatu eta bete egin beharko da.**
- Sartuko diren ondorengo tupla berriekin ez da egiaztatuko

Integritate-murriztapenen egoerak (II)

```
CONSTRAINT <ident> CHECK <baldintza>  
[ DISABLE | ENABLE [VALIDATE | NOVALIDATE] ]
```

- **ENABLE NOVALIDATE**

- Murriztapena **ez** da jadanik gordetako datuen gainean egiaztatuko. **Definizioa zuzenean katalogoan gordeko da**
- Sartuko diren ondorengo **tupla berriekin egiaztatuko da**

- **ENABLE VALIDATE**

- Definizioa katalogoan gordetzeko, murriztapena jadanik gordetako **datuen gainean egiaztatu eta bete egin beharko da.**
- Sartuko diren ondorengo **tupla berriekin egiaztatuko da**
- Lehenetsitako aukera

Integritate-murriztapenen egoerak. Adib

HONDAKINA

EEIFZ	Hkodea	...	toxik
E1	1		
E1			7
E2	2		12



ALTER TABLE Hondakina

ADD CONSTRAINT bat **CHECK** (toxik<10) **VALIDATE**

Murriztapena ez da katalogoan gordetzen

ALTER TABLE Hondakina

ADD CONSTRAINT bat **CHECK** (toxik<10) **NOVALIDATE**

Murriztapena katalogoan gordetzen da

Integritate-murriztapenen egoerak. Adib

ALTER TABLE Hondakina

ADD CONSTRAINT bat **CHECK** (toxik<10) **ENABLE**

INSERT INTO Hondakina **VALUES** (E2, 4, ..., **12**)

Tupla berria ez da datu-basean txertatzen



ALTER TABLE Hondakina

ADD CONSTRAINT bat **CHECK** (toxik<10) **DISABLE**

INSERT INTO Hondakina **VALUES** (E2, 4, ..., **12**)

Tupla berria datu-basean txertatzen da

Integritate-murriztapenen egoerak (III)

- Sartu behar den IM bat baino gehiago badago, gomendatzen da:
 - Lehenik, arau guztiak ENABLE NOVALIDATE moduan sartzea
 - DBKSk kontrola berehala itzultzen du, ezer egiaztatu gabe
 - Murriztapen baten huts-egiteak besteen sorrera zapuztea ekiditen dugu
 - Ondoren, VALIDATE egoerara aldatzea
 - Murriztapenak banan-banan egiaztatzen joango dira

```
ALTER TABLE <taulaident> MODIFY CONSTRAINT <ident>  
[ DISABLE | ENABLE [ VALIDATE | NOVALIDATE ] ]
```

7. Egiaztapena transakzioekin

- Transakzio bat zera da
 - DBaren gaineko eragiketa multzo bat da (INSERT, DELETE...)
 - DBKSarekin lan-unitate atomiko batosatzen dute (dena edo ezer ez)
 - Integritate unitate batosatzen dute
 - IMak egiaztatzen diren egoera batetik abiatzen da
 - IMak egiaztatzen diren egoera batera iristen da
 - Exekuzioan, IMak hautsi daitezke
- Transakzioa amaitu ondoren, IM guztiak betetzen dira. Egiaztapena honela egin daiteke:
 - Transakzioan zehar (eragiketa egitean)
 - Berehelako egiaztapena (IMMEDIATE)
 - Transakzioa amaitu ondoren
 - Atzeratutako egiaztapena (DEFERRED)

Integritate-murritzapenen egiaztapena.

Definizioa

CONSTRAINT <ident> **CHECK** <baldintza>

[**NOT DEFERRABLE** |

DEFERRABLE [**INITIALLY** (**DEFERRED** | **IMMEDIATE**)]]

SET CONSTRAINTS

[**ALL** | <ident> {, <ident>}]

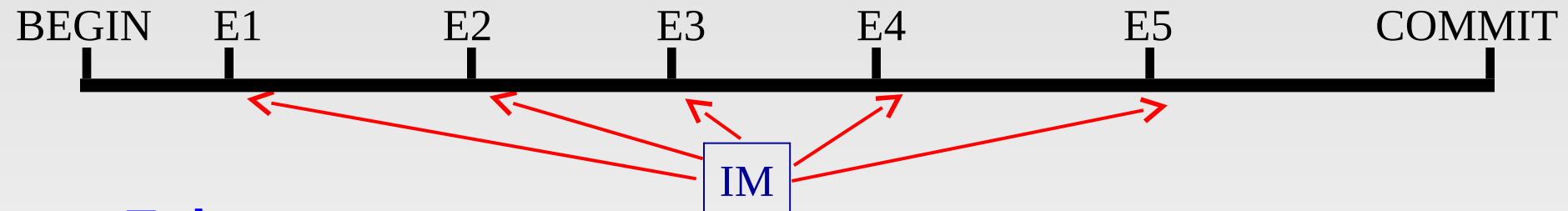
[**DEFERRED** | **IMMEDIATE**]

Integritate-murriztapenen egiaztapena.

Moduak

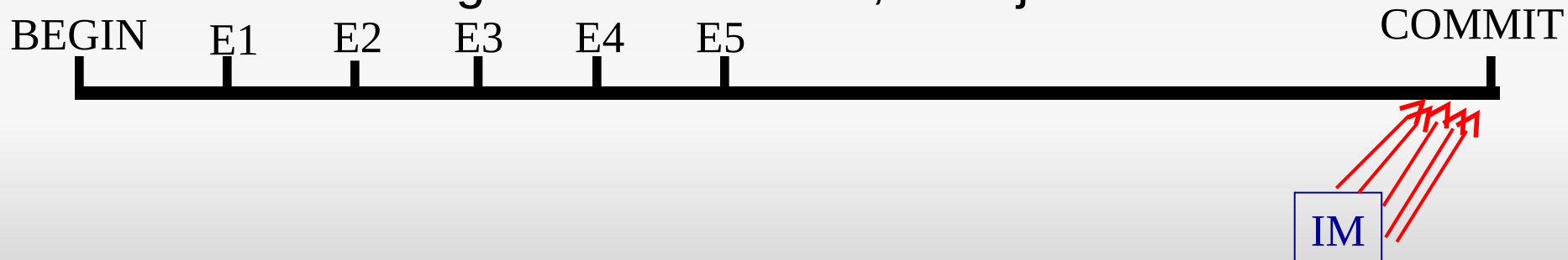
- **Berehalako egiaztapena**

- Murriztapena, eguneraketa gertatu bezain laster egiaztatzen da



- **Egiaztapen atzeratua**

- Murriztapena, transakzioa bukatu baino lehen egiaztatzen da
- COMMIT eragiketaren aurretik, hain juxtu



Integritate-murriztapenen egiaztapena.

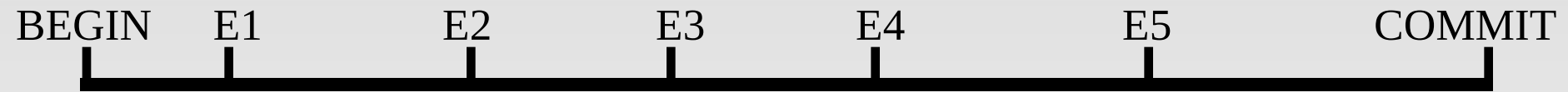
Definizio aukerak

- **NOT DEFERRABLE** (murriztapen atzeraezina)
 - Lehenetsitakoa
 - Egiaztapen-modua berehalakoa da
 - Transakzioaren egikaritzapenan zehar ezin da aldatu
- **DEFERRABLE** (murriztapen atzeragarria)
 - Egiaztapen-modua berehalakoa ala atzeratua, aukeran dago
 - Transakzioak egiaztapenarekin nola hasi behar duen zehaztu behar da
 - » **INITIALLY DEFERRED**
 - » **INITIALLY IMMEDIATE**
 - Transakzioaren egikaritzapenean zehar egiaztapen-modua alda daiteke (**SET CONSTRAINTS** agindua)

Integritate-murritzapenen egiaztapena.

Definizio aukerak. Adibidea (1)

DEFERRABLE INITIALLY IMMEDIATE



IM

SET ... DEFERRED



IM

SET ... DEFERRED

SET ... IMMEDIATE

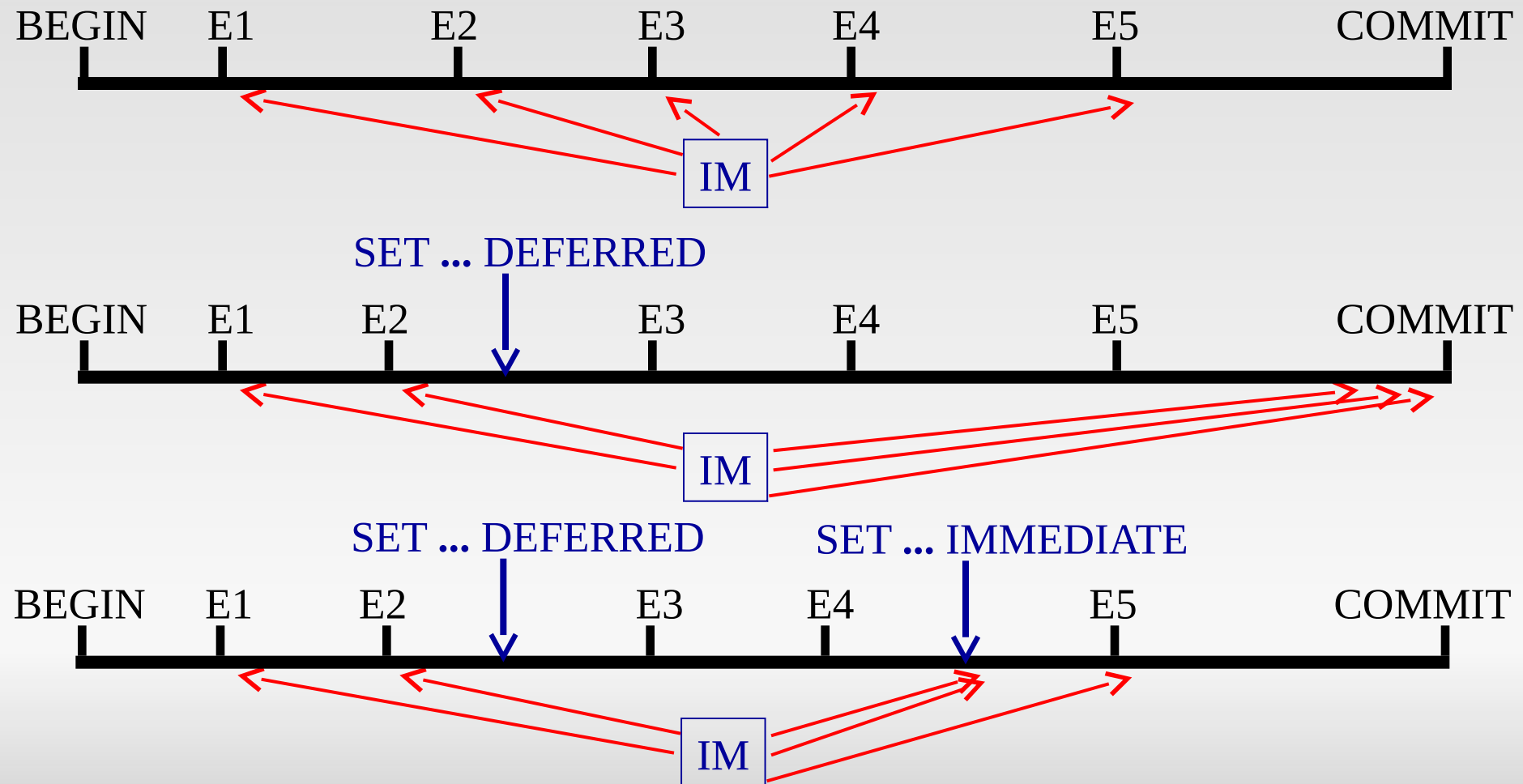


IM

Integritate-murriztapenen egiaztapena.

Definizio aukerak. Adibidea (1)

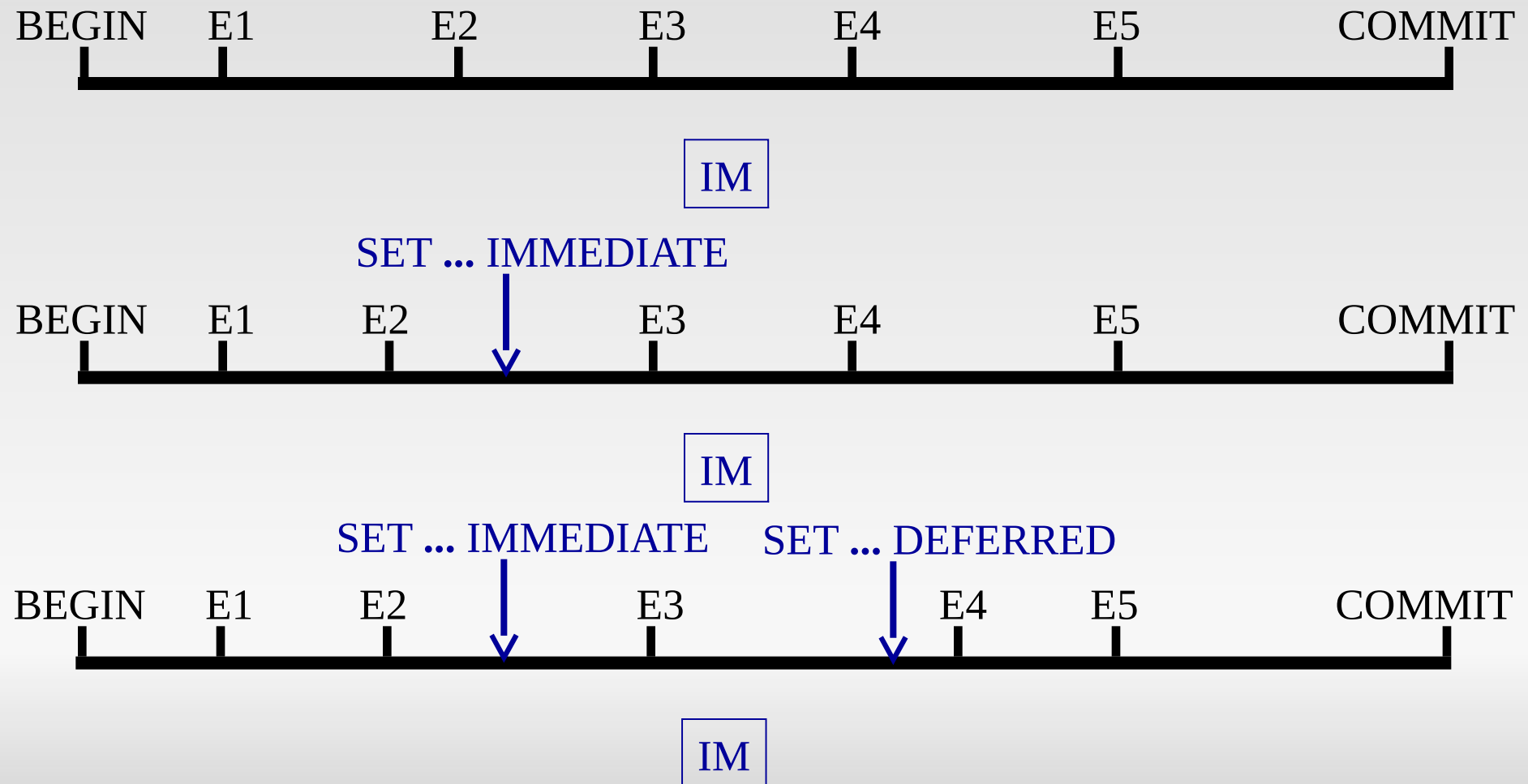
DEFERRABLE INITIALLY IMMEDIATE



Integritate-murriztapenen egiaztapena.

Definizio aukerak. Adibidea (1)

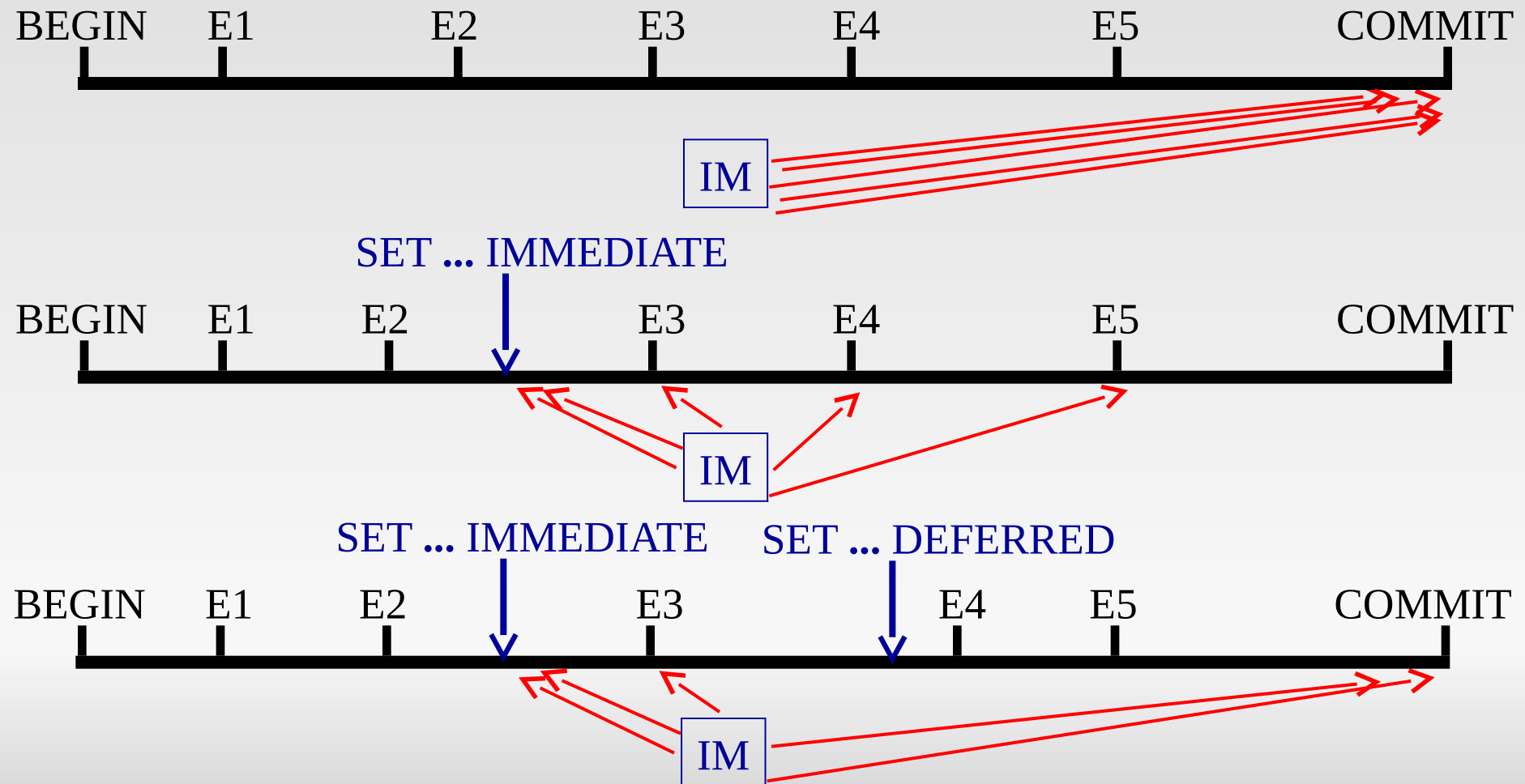
DEFERRABLE INITIALLY DEFERRED



Integritate-murriztapenen egiaztapena.

Definizio aukerak. Adibidea (1)

DEFERRABLE INITIALLY DEFERRED



Integritate-murriztapenen egiaztapena.

Integr.-murriztapen atzeratuak, zertarako?

Integritate-murriztapena: Lehen mailako ikasle berriek lehen mailako irakasgai guztietan matrikulatu behar dute

```
CREATE TABLE IKASLEA ( ...  
    CONSTRAINT matrikulaOsoa CHECK (...) NOT DEFERRABLE)
```

Transakzioa

```
{  
    insert into Ikaslea values (1, Jon, ...);  
    insert into Matrikula values (1, 1010, ...);  
    insert into Matrikula values (1, 1020, ...);  
    ....  
    commit;
```



errorea

Integritate-murritzapenen egiaztapena.

Integr.-murritzapen atzeratuak, zertarako? (2)

Integritate-murritzapena: Lehen mailako ikasle berriek lehen mailako irakasgai guztietan matrikulatu behar dute

```
CREATE TABLE IKASLEA ( ...  
    CONSTRAINT matrikulaOsoa CHECK (...)  
    DEFERRABLE INITIALLY DEFERRED)
```

Transakzioa

insert into Ikaslea values (1, Jon, ...);

insert into Matrikula values (1, 1010, ...);

insert into Matrikula values (1, 1020, ...);

....

set constraints matrikulaOsoa immediate;

commit;

**Egiaztapenik
ez !!!**

**Egiaztapena
orain !!!**