
7. Integrita a bezpečnost dat v DBS

7.1. Implementace integritních omezení.....	2
7.1.1. Databázové triggery.....	5
7.2. Zajištění bezpečnosti dat.....	12
7.2.1. Bezpečnostní mechanismy poskytované SŘBD.....	13
Literatura.....	16

7.1. Implementace integritních omezení

- Podpora SŘBD

- Podpora pro integritní omezení zahrnutá v SQL/92

- Omezení domén

```
CREATE DOMAIN doména [AS] dat_typ [impl_hodnota]
[seznam_omezení_domény]
```

Omezení domény

```
[CONSTRAINT jm_omezení] CHECK (podmíněný_výraz)
```

Př)

```
CREATE DOMAIN barva CHAR[8]
CONSTRAINT icbarva
CHECK (VALUE IN ('černá', 'bílá', 'modrá',
'červená'))
```

- **Všeobecná omezení**

```
CREATE ASSERTION jm_omezení CHECK (podmíněný_výraz)
```

Př)

```
CREATE ASSERTION icpocet  
CHECK (NOT EXISTS (SELECT r_cislo FROM Ucet  
GROUP BY r_cislo HAVING COUNT(*)>5))
```

- **Omezení bazové tabulky**

- Součást příkazu CREATE TABLE, resp. ALTER TABLE:

```
[CONSTRAINT jm_omezení] definice_omezení
```

- definice omezení pro kandidátní klíče - UNIQUE,
- definice omezení pro primární klíče - PRIMARY KEY,
- definice omezení pro cizí klíče - FOREIGN KEY
- definice omezení typu CHECK:

```
CHECK (podmíněný_výraz)
```

Př)

```
CREATE TABLE Transakce (...  
CHECK (castka ≥ -100000), ...)
```

- Omezení sloupců bázové tabulky
 - NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY REFERENCES ..., omezení typu CHECK:

CHECK (podmíněný_výraz)

- režim kontroly omezení:
 - **IMMEDIATE** - okamžitá kontrola
 - **DEFERRED** - kontrola na závěr transakce. Používáme, pokud připustíme přechodnou nekonzistenci v průběhu transakce.
- specifikuje se v definici omezení nebo příkazem **SET CONSTRAINTS**
- Databázové triggery (viz dále)
- Kontroly naprogramované v aplikaci

Kontroly naprogramované v klientské části aplikace, typicky s podporou vývojových prostředí:

 - kontrola vstupů - omezení zabudovaná v definicích formulářů
 - naprogramované kontroly pro určité události uživatelského rozhraní (triggery rozhraní)

7.1.1. Databázové triggery

- není součástí SQL/92, je součástí SQL/99

Databázový trigger je databázový objekt obsahující kód spouštěný specifikovanou událostí v databázi.

- Složky příkazu vytvoření databázového triggeru

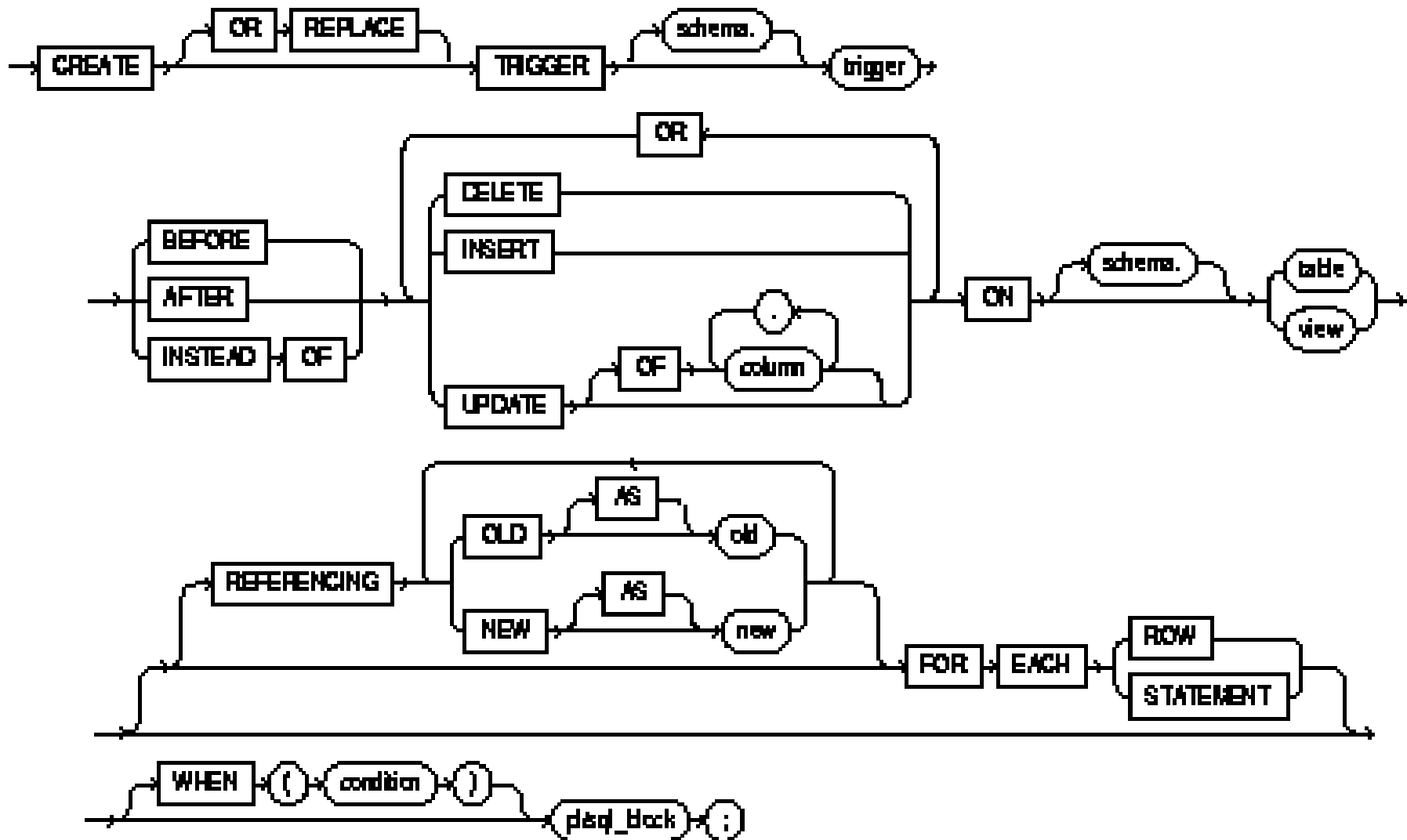
- jméno triggeru,
- jméno tabulky jméno [REFERENCING alias_pro_OLD_a_NEW],
- čas spuštění akce – BEFORE/AFTER,
- událost – INSERT/DELETE/UPDATE [OF sloupec, ...],
- spouštěná akce – FOR EACH {ROW | STATEMENT} [WHEN podmínka] spouštěný_SQL_příkaz

- Typy databázového triggeru

- příkazový,
- řádkový

Poznámka: Databázový trigger je něco jiného, než klientské části aplikace, který se spouští typicky událostí formuláře (např. stisk tlačítka) nebo tiskové sestavy (např. nová stránka).

- Databázové triggery v prostředí ORACLE



Typ „INSTEAD OF“ slouží k aktualizaci neaktualizovatelných pohledů

Př)

Ustav

zkratka	jmeno	areal
---------	-------	-------

Areal

nazev	adresa
-------	--------

**Možná implementace ON UPDATE CASCADE
(Oracle podporuje jen ON DELETE CASCADE)**

```
CREATE OR REPLACE TRIGGER aktualizuj_ustav
  AFTER UPDATE OF nazev ON Areal
  REFERENCING OLD AS puvodni NEW AS novy
  FOR EACH ROW
BEGIN
  UPDATE Ustav SET areal = :novy.nazev
  WHERE areal = :puvodni.nazev;
END;
```

Možná implementace ON DELETE SET NULL (Oracle nepodporuje)

```
CREATE OR REPLACE TRIGGER nuluj_Ustav
  AFTER DELETE ON Areal
  FOR EACH ROW
BEGIN
  UPDATE Ustav SET areal = NULL
  WHERE areal = :old.nazev;
END;
```

Omezení příkazů v těle triggeru

Original
EMP Table

EMP Table	
ENAME	SAL
SMITH	1000
JONES	1000
WARD	1000

SQL Statement That
Fires an AFTER
Row Trigger

```
UPDATE emp  
SET sal=sal *1.1;
```

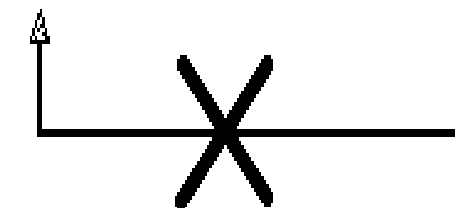


Mutating
EMP Table

EMP Table	
ENAME	SAL
SMITH	1100
JONES	1000
WARD	1000

AFTER Row
Trigger Fired,
Contains:

```
SELECT sal  
FROM emp  
WHERE...
```



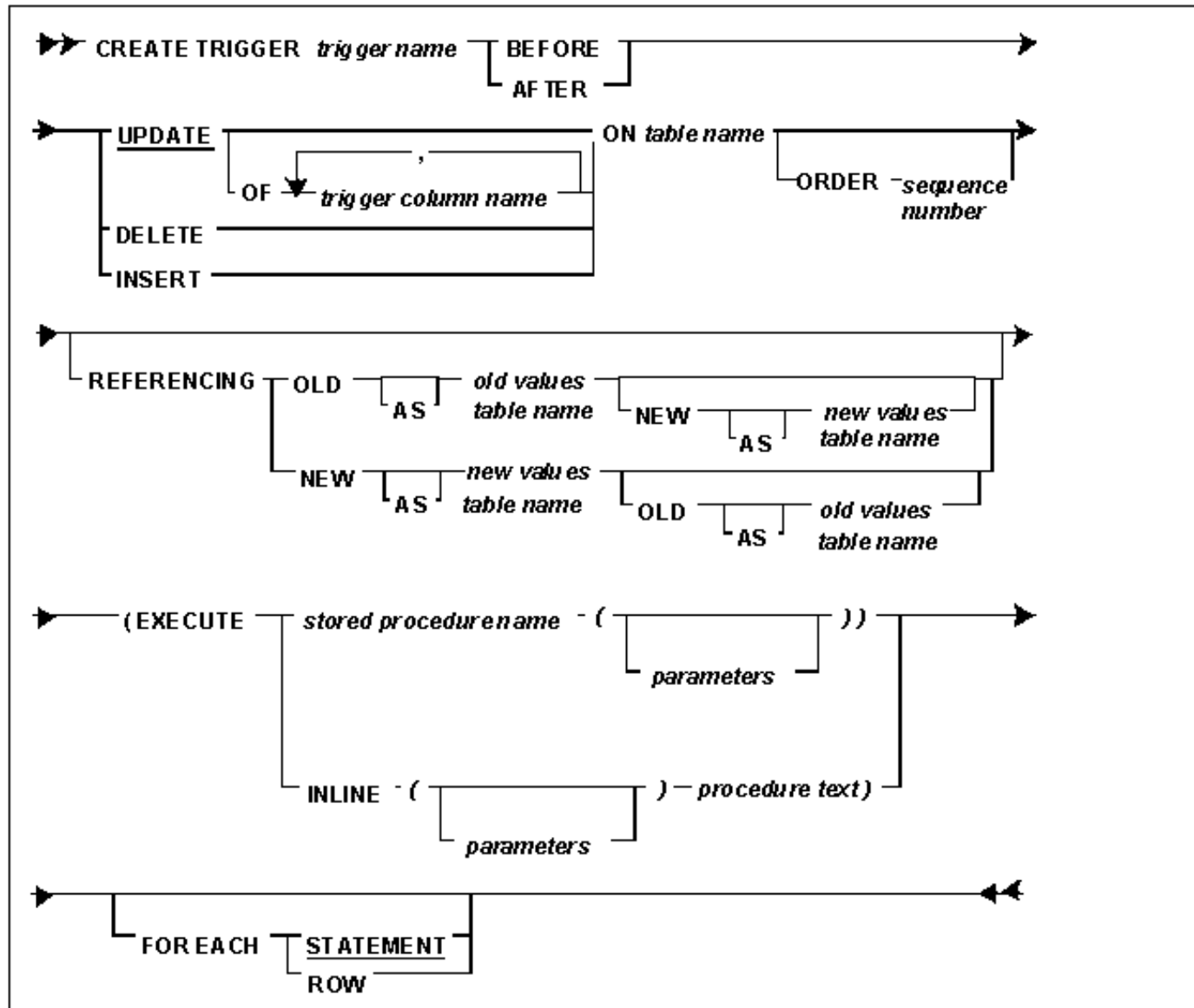
Not allowed because EMP
table is a mutating table

Model spouštění triggerů a provádění kontrol integritních omezení

- 1. Proved' všechny BEFORE příkazové triggerery platné pro daný příkaz;**
- 2. LOOP FOR EACH řádek ovlivněný SQL příkazem DO**
 - a) Proved' všechny BEFORE řádkové triggerery platné pro daný příkaz;**
 - b) Uzamkni a změň řádek a proved' kontrolu integritních omezení;
(Zámek není uvolněn, dokud není transakce potvrzena.)**
 - c) Proved' všechny AFTER řádkové triggerery platné pro daný příkaz;**
- 3. Dokonči odloženou kontrolu integritních omezení;**
- 4. Proved' všechny AFTER příkazové triggerery platné pro daný příkaz;**

• Triggery SQLBase

CREATE TRIGGER command



7.2. Zajištění bezpečnosti dat

Cíle, které je třeba vzít v úvahu při návrhu databázové aplikace z pohledu bezpečnosti:

Důvěrnost (secrecy) - informace by neměly být přístupné neautorizovaným uživatelům.

Integrita (integrity) - modifikovat data může jen autorizovaný uživatel.

Dostupnost (availability) - autorizovaným uživatelům by nemělo být bráněno v přístupu.

→ ***bezpečnostní politika*** - kdo co může s jakými daty dělat

→ ***bezpečnostní mechanismy*** - zajištění bezpečnostní politiky

7.2.1. Bezpečnostní mechanismy poskytované SŘBD

- Pohledy
- Řízení přístupu

Existují dva hlavní způsoby řízení přístupu:

- **nepovinné (*discretionary access control*)** - založen na přístupových právech. Každému uživateli jsou přiřazena přístupová práva k databázovým objektům.
 - **povinné (*mandatory*)** - založen na bezpečnostních třídách (stupních utajení) objektů, stupních prověření subjektů, a pravidlech provádění operací.
- SQL poskytuje podporu pro nepovinné řízení.
 - pro některé SŘBD existují i verze poskytující podporu povinného řízení

- Typické prostředky pro řízení přístupových práv:
 - definice uživatelů, případně skupin uživatelů (role u Oracle)

Př) Oracle

```
CREATE USER přihl_jméno  
IDENTIFIED {BY heslo | EXTERNALLY} ...
```

- systémová přístupová práva

Př) Oracle, SQLBase

```
GRANT úroveň_oprávnění TO seznam_uživatelů  
REVOKE úroveň_oprávnění FROM seznam_uživatelů
```

- úrovně v SQLBase: CONNECT, RESOURCE, DBA
- Oracle: úroveň např. CREATE TABLE,
 - uživatel PUBLIC
 - možnost vytvářet role:

```
CREATE ROLE jméno_role  
[ {NOT IDENTIFIED | IDENTIFIED {BY heslo |  
EXTERNALLY} } ]
```

- role vytvořené při instalaci: CONNECT, RESOURCE, DBA, ...
- příkazem GRANT lze přiřadit uživateli také roli

- přístupová práva k databázovým objektům (je v SQL/92)

```
GRANT seznam_oprávnění ON db_objekt TO  
  {seznam_uživatelů | PUBLIC} [WITH GRANT OPTION]  
REVOKE seznam_oprávnění ON db_objekt FROM  
  {seznam_uživatelů | PUBLIC} [RESTRICT | CASCADE]
```

- často i pouze pro sloupce tabulky nebo pohledu (ne v SQL/92)

Př) SELECT, UPDATE, INDEX, EXECUTE, ...

- šifrování přihlašovacího jména a hesla při přenosu po síti
- šifrování dat v databázi
- záznamy o manipulacích (audit)

Literatura

- 1. Silberschatz, A., Korth H.F, Sudarshan, S.: Database System Concepts. Fourth Edition. McGRAW-HILL. 2001, str. 225 – 255.**
- 2. Pokorný, J.: Databazová abeceda. Science, Veletiny, 1998, str. 89 – 92, 141 – 144, 197 – 200.**
- 3. Oracle9i Application Developer's Guide - Fundamentals. Oracle Corp. March 2002, str. 15-1 – 15-54.**