36. Spouště a uložené procedury

36.1. Spouště (Trigger)

Triggr je procedura, která je automaticky spuštěná DBMS jako reakce na specifickou akci v databáze. Když nastane událost, zkontroluj podmínku, pokud podmínka platí, proveď akci např. když se vloží přihláška studenta, který má průměr 1.0, automaticky ho akceptuj. Syntax silně závislá na DBMS (použité databázi = Database Management System).

Použití:

- komplexnější IO, opravy při narušení IO
- posun programování logiky do DB
- syntax silně závislá na DBMS

```
CREATE TRIGGER jméno_triggra
BEFORE | AFTER | INSTEAD OF událost ON jméno_tabulky [REFERENCING referenční proměnné AS jména]
[FOR EACH ROW]
WHEN (podmínka)
akce
```

Událost – změna v DB, která vyvolá spuštění triggru (nezáleží, kdo jí vyvolal)

- definovaná na relaci - ON jméno_tabulky
- INSERT | DELETE | UPDATE OF [seznam_sloupců]

Podmínka – dotaz nebo test, který je proveden pokud je triggr aktivovaný

- když výsledek dotazu je NOT NULL, podmínka platí

Akce – procedura, která je provedena při spuštění pokud podmínka platí

```
DECLARE
BEGIN
      SQL PŘÍKAZY
END
```

36.1.1. Granularita

Triggr na úrovni příkazu

- FOR EACH STATEMENT
- je aktivován jednou pro celý příkaz

Trigger na úrovni řádku

- FOR EACH ROW
- je aktivován pro každý modifikovaný řádek
- vhodný, když je závislost na hodnotách modifikovaných řádků

pokud bude množina modifikovaných řádků prázdná:

- FOR EACH STATEMENT triggr se provede 1x
- FOR EACH ROW triggr se neprovede ani jednou

36.1.2. Referenční proměnné

- odkazují na modifikovaný řádek
  - old row, new row
- odkazují na modifikované řádky ve formě tabulky
  - old table, new table

	triggr na úrovni příkazu	trigger na úrovni řádku
INSERT	new table	new row, new table
DELETE	old table	old row, old table
UPDATE	new table, old table	new row, old row, new table, old table

36.1.3. BEFORE triggr

granularita FOR EACH ROW, pracuje se stavem databáze před provedením vlastního dotazu, který triggr spustil. Vlastní dotaz bude proveden až po ukončení všech akcí triggru.

Použití:

- validace vstupních dat
- automatické generování a doplňování dat pro nové řádky
- nepoužívají se pro další modifikace, neobsahují INSERT, UPDATE, DELETE

36.1.4. AFTER triggr

granularita FOR EACH ROW a taky FOR EACH STATEMENT, pracuje se stavem databáze po provedení vlastního dotazu, který triggr spustil. Až po kontrole IO spojených s vlastním dotazem. Nejdřív se provede vlastní dotaz, pak akce triggru.

Použití:

- aplikační logika

BEFORE triggry → vlastní dotaz (IO) → AFTER triggry

36.2. Uložené procedury (Procedura)

Uložená procedura (stored procedure) je podprogram uložený a spuštěný v rámci DB serveru.

Podle návratové hodnoty:

- když vrací hodnotu – **funkce**
- když nic nevrací – **procedura**

Použití:

- přepoužití často používaných kombinací SQL příkazů (volání stejné procedury z více triggrů)
- posun programování logiky do DB
- validace dat (datové API)

syntax silně závislá na DBMS

napr. PL/SQL Oracle

```
definice procedury:
CREATE [OR REPLACE] PROCEDURE jméno_procedury
[(parameter1 [typ] datový_typ,
  parameter2 [typ] datový_typ], ...)]
[IS|AS deklarace proměnných]
BEGIN
akce
[EXCEPTION výjimky]
END;

volání procedury:
EXECUTE jméno_procedury(param1, param2, ...)

vymazání procedury:
DROP PROCEDURE jméno_procedury;
```

36.2.1. Typ parametru

Parametre slouží na výměnu dat mezi procedurou a programem, který ji volal.

- IN - pochází z programu, procedura s ním pracuje, ale nemůže jej měnit. (defaultní hodnota)
- OUT - procedura s ním nepracuje, ale může jej měnit. Slouží jako návratová hodnota, která se předá programu.
- IN OUT - procedura s ním pracuje a může jej i měnit.