

Aktivní databáze – Oracle triggerry, klasifikace a spouštění triggerů.

Thursday, May 30, 2013 8:16 AM

V řadě IS, které běží nad nějakou databází, potřebujeme v případě vzniku nějaké události (např. modifikujeme řádek v nějaké tabulce) automaticky spustit příkaz, který provede nějaké operace. K tomuto účelu slouží triggerry (spouštěče). Trigger je speciální typ uložené procedury, která se aktivuje při splnění nějaké podmínky na serveru (aktualizace dat, události spojené s DB nebo session).

Triggerry jsou vlastně procedury, které automaticky volá (spouští) SŘBD při definované události. Touto událostí může být buď vložení (INSERT), rušení (DELETE), nebo aktualizace (UPDATE OF) záznamu v tabulce (lze vázat i na update konkrétního sloupce). Pravidla lze kombinovat pomocí OR. Triggerry bývají obvykle volány buď:

- Před specifikovanou událostí – **BEFORE**
- Po specifikované události – **AFTER**
- **INSTEAD OF** - místo specifikované události

Triggerry jsou:

- Spuštěny
- Vyhodnoceny
- Vykonány

Od uložených procedur a funkcí se odlišují tím, že jsou spuštěny *při modifikaci tabulky*, definují se pouze pro db tabulky a nepřijímají argumenty a lze je spustit jen při zmiňovaných DML příkazech.

U triggeru lze rovněž specifikovat podmínku, kdy má být vykonáno jeho tělo (PL/SQL blok) a to použitím klauzule WHEN.

Triggerry jsou plně zkompileované po spuštění příkazem CREATE TRIGGER a po uložení procedurálního kódu v systémovém katalogu.

Trigger lze také

- deaktivovat (zakázat) - ALTER TRIGGER jmeno DISABLE;
- aktivovat – ALTER TRIGGER jmeno ENABLE;
- zrušit – DROP TRIGGER jmeno;

Výhody triggerů jsou:

- nepovolí neplatné datové transakce, zajišťují komplexní bezpečnost, zajišťují referenční integritu přes všechny uzly db, zajišťují audit (sledování), spravují synchronizaci tabulek, zaznamenávají statistiku často modifikovaných tabulek.

Klasifikace triggerů

Příkazové triggerry (statement level)

Triggerry se spustí nad tabulkou bez ohledu na to, kolik tabulka obsahuje řádek. Např. pokud chci logovat změny provedené v DB, která obsahuje tabulky PRACOVISTE, ZAMESTNANCI do tabulky LOGY. Po každé operaci I, U, D nad tabulkami PRAC a ZAM se do tabulky LOGY vloží záznam o modifikaci tabulky a typu modifikace.

```
CREATE TRIGGER tai_pracoviste
AFTER INSERT ON pracoviste
BEGIN
INSERT INTO logy VALUES ( 'PRACOVISTE', 'I' );
END;
/
```

Řádkové trigger – FOR EACH ROW

Trigger se spouští jednou pro každý řádek tabulky. Např. z předchozího příkladu chci mít u každého záznamu informaci, kdy byl zadán a kdo jej zadal.

```
CREATE TRIGGER trbi_pracoviste
BEFORE INSERT ON pracoviste
FOR EACH ROW
BEGIN
:new.zadal := user;
:new.datum := sysdate;
END;
/
```

Vlastní obsluhu události lze nadefinovat v PL/SQL bloku. Uvnitř PL/SQL bloku (a také klauzuli WHEN) řádkového triggeru se lze odkazovat na původní a nový záznam pomocí pseudoproměnných **:new** (obsahuje vkládaný záznam) a **:old** (původní záznam). Je zřejmé, že při vkládání nového záznamu není definována :old a při mazání :new. Jejich jména lze předefinovat v klauzuli **REFERENCINGOLD AS**.

Business rules trigger

Triggery jsou také často používány při realizaci tzv. business rules, tj. integritních omezení specifických pro danou oblast použití. Např. použijeme – studenti si mohou zapsat max 20 kreditů za semestr. Pokud při vkládání dat do tabulky ZAPIS překročíme maximální povolenou hodnotu, dostaneme chybové hlášení – a to zařídí business trigger.

Zápis

```
CREATE OR REPLACE TRIGGER jméno
BEFORE | AFTER | INSTEAD OF
DELETE | INSERT | UPDATE OF cols
ON tabulka
[ způsob odkazování ]
[ FOR EACH ROW ]
[ WHEN ( podmínka ) ]
AS pl/sql kód
```

Omezení triggerů

- BEFORE a AFTER trigger nelze specifikovat nad pohledy
- V BEFORE triggerech není možné zapisovat do :old záznamů
- V AFTER triggerech nelze zapisovat do :old ani do :new záznamů
- INSTEAD OF trigger pracují jen s pohledy, mohou číst :old i :new, ale nemohou zapisovat ani do jednoho
- Nelze kombinovat INSTEAD OF a UPDATE
- Nelze definovat trigger nad LOB atributem

Dvě zásadní omezení

- Nelze použít transakce, pokud je zpracovávána jiná transakce, tedy prakticky nelze použít transakce vůbec
- Není možné sledovat ani modifikovat data v tabulce, která způsobila vyvolání DML triggeru -> jediné známé řešení je zrcadlení tabulek

Sémantika Oracle Triggerů

- Spouštění probíhá okamžitě při události, nelze je spustit explicitně (např. uživ. Příkazem)
- Vnořené spouštění triggerů - činnost triggeru může vyvolat jiný trigger - kontext aktuálního triggeru se uloží, začne se vykonávat nový trigger, pak se zas obnoví a pokračuje ten původní
- Maximální hloubka zanoření je 32, pak to hodí výjimku

Doplňující pojmy

události (events)

- změna stavu databáze

- časové události
- externí, definované aplikací

podmínky (conditions)

- databázový predikát
- databázový dotaz

akce (actions)

- libovolná manipulace s daty
 - transakční příkazy
 - pravidla zpracování
 - externí procedury

Spouštění triggerů

- Vykonání triggeru
 - Okamžité (immediate)
 - Před událostí
 - Po události
 - Namísto události
 - Odložené (deferred)
 - Na konci transakce
 - Po uživatelském příkazu
 - Následkem uživatelského příkazu
 - Oddělené (detached)
 - V kontextu samostatné transakce vypuštěné z počáteční transakce poté, co nastala událost
 - Možné kauzální závislosti počáteční a oddělené transakce
- Vykonání akce
 - Okamžité (immediate)
 - Následuje ihned po vyhodnocení podmínky
 - Odložené (deferred)
 - Akce je odsunuta na konec transakce
 - Akci vyvolá uživatelský příkaz
 - Oddělené (detached)
 - Probíhá v kontextu samostatné transakce vypuštěné z počáteční transakce ihned po vyhodnocení podmínky
 - Možné kauzální závislosti počáteční a oddělené transakce

Kdy je co spuštěno

BEFORE triggers run the trigger action before the triggering statement is run. This type of trigger is

commonly used in the following situations:

When the trigger action determines whether the triggering statement should be allowed to complete. Using a BEFORE trigger for this purpose, you can eliminate unnecessary processing of the triggering statement and its eventual rollback in cases where an exception is raised in the trigger action.

To derive specific column values before completing a triggering INSERT or UPDATE statement.

[7:01:26 PM] Lukáš Volf: AFTER triggers run the trigger action after the triggering statement is run.

[7:01:50 PM] Lukáš Volf: INSTEAD OF triggers provide a transparent way of modifying views that cannot be modified directly through DML statements (INSERT, UPDATE, and DELETE). These triggers are called INSTEAD OF triggers because, unlike other types of triggers, Oracle fires the trigger instead of executing the triggering statement.

You can write normal INSERT, UPDATE, and DELETE statements against the view and the INSTEAD OF trigger is fired to update the underlying tables appropriately. INSTEAD OF triggers are activated for each row of the view that gets modified.

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>