

PROCEDURAT

- **Nje stored procedure eshte nje grup komandash SQL i parakompiluar , dhe qe ruhet ne database (nen nyjen "Programmability" -> "Stored Procedures").**
- **Te mirat e Stored Procedurave :**
 - Programim Modular - E shkruajme njehere nje stored procedure dhe e perdorim sa here na duhet ne nje aplikacion
 - Performanca - SP ofrojne ekzekutim me te shpejte te kodit dhe reduktojne trafikun e rrjetit
 - Siguria - Perdoruesit aksesojne objektet e DB nepermjet nje menyre te kontrolluar

- **Krijimi:**

```
CREATE PROC Sales.GetCustomerOrders
```

- **Ekzekutimi:**

```
EXEC Sales.GetCustomerOrders
```

- **Modifikimi:**

```
ALTER PROC Sales.GetCustomerOrders
```

- **Fshirja:**

```
DROP PROC Sales.GetCustomerOrders
```

- **Shembull:**

```
CREATE PROC Sales.GetCustomerOrders
@custid AS INT,
@orderdatefrom AS DATETIME = '19000101',
@orderdateto AS DATETIME = '99991231',
@numrows AS INT = 0 OUTPUT
AS
BEGIN
SET NOCOUNT ON;
SELECT orderid, custid, shipperid, orderdate, requireddate,
shippeddate
FROM Sales.Orders
WHERE custid = @custid
AND orderdate >= @orderdatefrom
AND orderdate < @orderdateto;
SET @numrows = @@ROWCOUNT;
RETURN;
END
```

TRIGGERAT

Case Study: Menaxhimi Bibliotekes

- **Supozojme se kemi 3 tabelat**
 - 1.Tabela gjendje: gjendjen e librave ne nje biblioteke
 - 2.Tabela huazime: huazimet per cdo liber te marre
 - 3.Tabela kthime: kthim per cdo liber te kthyer
- **Pemes ngjarjeve(evente) te gjejme nje skeme qe ruan te azhornuar gjendjen e librave ne biblioteke**
 - Huazime (shton rekord ne tabelen Huazime)
 - Kthime (shton rekord ne tabelen Kthime)
 - Gjendje
- **Sa here qe hasim ne nje prej dy veprimeve hua/kthim ekzekutohet nje event qe shton nje record ne tabelat respektive.**

- **Trigerat janë një lloj procedure që thërren në mënyrë automatike sa herë që ndodh një veprim i caktuar mbi një tabelë.**
- **Veprimet janë:**
 - INSERT
 - DELETE
 - UPDATE
- **Llojet e trigerave janë:**
 - AFTER
 - INSTEAD OF

Tabelat llogjike INSERTED & DELETED

- Veprimi **INSERT**: Cdo tabele shoqerohet ne momentin e shtimit te rreshtave ne te nga nje tjeter tabele llogjike me emrin INSERTED e cila ka te njejten strukture me tabelen ku behet INSERT-i.
 - Ne tabelen INSERTED vendosen rreshtat qe po shtohen ne tabelen perkatese.
- Veprimi **DELETE**: krijon nje tabele llogjike me emrin DELETED, ne te cilen vendosen rreshtat e sapo fshire.
- Veprimi **UPDATE**: perdor te dyja tabelat INSERTED dhe DELETED.
 - DELETED ruan versionet e rreshtave para ndryshimit.
 - INSERTED mban rreshtat pas ndryshimit.

- Tabelat **INSERTED** dhe **DELETED** janë tabela llogjike të izoluar për tabelën ku po kryhet veprimi .
- Tabelat llogjike janë disponibel deri në fund të ekzekutimit të veprimit.

Komanda Rollback:

- Trigeri është pjesë e të njëjtin transaksion që shkakton ekzekutimin e tij.
- Kodi i trigerit është pjesë e të njëjtin transaksion.
- Një komandë **ROLLBACK** brenda trupit të trigerit mund të përdoret nëse duam të ç'bejme veprimet e trigerit.

Le te ndertojme nje trigger qe nuk lejon shtimin e nje punonjesi nqs nuk ka lindur akoma kur ai eshte marr ne pune.

```
CREATE TRIGGER TI  
ON EMPLOYEES AFTER  
INSERT  
AS  
BEGIN  
IF EXISTS ( SELECT * FROM INSERTED WHERE BIRTHDATE >  
HITEDATE )  
ROLLBACK;  
END
```

Triggerat INSTEAD OF

- Triggerat **INSTEAD OF** maskojne sjelljen e nje veprimi **INSERT/UPDATE/DELETE** ne tabelen perkatese.
- Ato zevendesojne komanden dhe ekzekutojne kodin qe ndodhet midis **BEGIN END**.

UDF

USER-DEFINED FUNCTIONS

- **Funksionet.**
- **Llojet e funksioneve**
- **Funksionet skalare**
- **Funksionet tabelë e thjeshtë**
- **Funksionet tabelë komplekse**

- Funkzionet e krijuara nga përdoruesit(UDF) nuk kryejnë veprime te cilat ndryshojnë gjendjen e DB.
- Nuk përdoren për të kthyer një bashkësi me rezultate, përdoren procedurat.
- Nuk suportojnë blloqet për kapjen e gabimeve, TRY...CATCH, @ERROR ose RAISERROR.
- Funkzionet nuk mund të thërrasin një store procedure.
- UDF (User-defined functions) nuk mund të përdorin SQL dinamike ose tabela të përkoshme. Lejon krijimin e variablave mbi tabelë.
- Nuk lejon përdorimin e SET
- Nuk lejon përdorimin e FOR XML
- Ekzistojnë tre lloje funksionesh:
 - Funkzionet skalare
 - Funkzionet me vlerë tabelë të thjeshta (Inline Table-Valued functions)
 - Funkzionet me vlerë tabelë me shumë komanda (Multi-statement table-valued functions)

Funksionet tabelë e thjeshtë

- Funksionet tabelle shpesh konsiderohen si VIEW te parametrizuara.
- Ne dallim nga funksionet skalare keto funksione kthejne nje bashkesi rreshtash apo nje "tabelle".
- Trupi i ketyre funksioneve eshte vetem nje komande SELECT e cila nga ana e saj mund te jete komplekse dhe te perfshije JOIN apo nen query te lidhura me te me operatoret IN dhe EXISTS.

Funksionet tabelë komplekse

- Funksionet tabelle komplekse janë të ngjashme me funksionet e thjeshta tabelle.
- Dallimi, funksionet tabelle komplekse nuk janë të perbere vetem nga një komandë SELECT, ato deklarohen në koken e tyre strukturën e tabelës që do të kthejnë dhe i vendosin asaj një emër.
- Në trupin e funksionit shtohen rreshta në këto tabelle të deklaruar në fillim për të përgatitur rezultatin e dëshiruar.