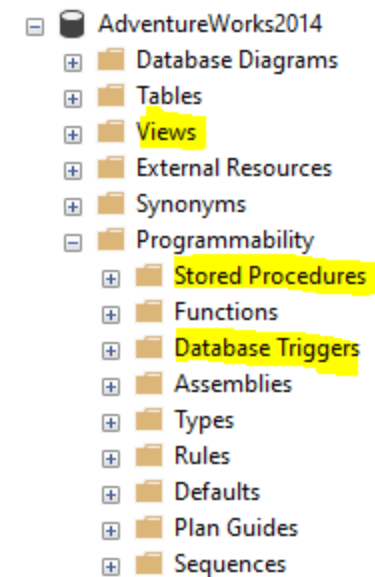


# Baze podataka II

## Modul 9 – Jezik SQL

DDL komande

Pogledi uskladištene procedure  
i okidači



# Summary

- **Pogledi**
  - *Views*
- **Uskladištene procedure**
  - *Stored Procedures*
- **Okidači**
  - *Triggers*



# Lekcija 1: Pogledi

- Uvod u poglede
- Tipovi pogleda
- Prednosti korištenja
- Kreiranje pogleda
- Izmjena pogleda
- Brisanje pogleda
- Skrivanje strukture pogleda

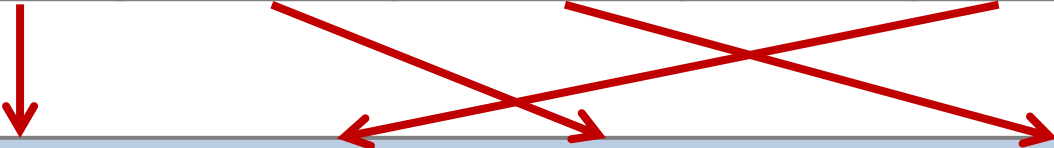


# Uvod u poglede

- Pogled (view) je objekat u bazi podataka koji izgleda kao tabela

Employee (table)					
EmployeeID	LastName	FirstName	BirthDate	Title	...
287	Mensa-Annan	Tete	3/2/1984	Mr.	...
288	Abbas	Syed	4/5/1976	Mr.	...
289	Valdez	Rachel	9/8/1973	NULL	...

- Možemo reći da je to **SELECT** upit sa imenom



EmployeeList (view)			
EmployeeID	Title	LastName	FirstName
287	Mr.	Mensa-Annan	Tete
288	Mr.	Abbas	Syed
289	NULL	Valdez	Rachel

# Tipovi pogleda

- **Standardni**

- Kombinacija jedna ili više korisničkih tabela

- **Sistemske pogledi**

- Uvid u podatke vezane za sam server i korisničke baze (meta podaci)

- **Indeksirani pogledi**

- Pogled koji se pohrani na disk uz pomoć jedinstvenog klastered indeksa

- **Particionirani pogledi**

- Horizontalno particioniranje podataka između jednog ili više servera

# Prednosti korištenja

- **Ciljani podaci za korisnike**
- **Maskiraju kompleksnost baze podataka**
- **Olakšavaju permisije i upravljanje**
- **Organizovanje podataka za eksport**
- **Osigurava kompatibilnost unazad**
- **Idealno za izvještavanje**

# Kreiranje pogleda

- Pogled se kreira preko **CREATE VIEW** komande
- Maksimalno do 32 nivoa dubine
- **ORDER BY** nije moguć bez **TOP**

```
CREATE VIEW HumanResources.EmployeeList
AS
    SELECT E.BusinessEntityID, P.LastName, P.FirstName
    FROM Person.Person AS P
    INNER JOIN HumanResources.Employee AS E
    ON P.BusinessEntityID = E.BusinessEntityID
```

# Izmjena pogleda

- Izmjena se vrši preko **ALTER VIEW** komande

```
ALTER VIEW HumanResources.EmployeeList
AS
SELECT E.BusinessEntityID, P.LastName, P.FirstName, E.Gender, E.HireDate
FROM Person.Person AS P
INNER JOIN HumanResources.Employee AS E
ON P.BusinessEntityID = E.BusinessEntityID
```



# Brisanje pogleda

- **Brisanje se vrši putem DROP VIEW komande**
- **Također se uklanjaju i permisije**
- **Moguće je obrisati više pogleda jednom komandm**
  - Odvojeno zarezom

```
DROP VIEW HumanResources.EmployeeList
```

# Skrivanje strukture pogleda

- **Obfuscating**

- Putem WITH ENCRYPTION klauzule
- Struktura pogleda je pohranjena u kriptovanom obliku
- Generalno nije preporučljivo

- **Koristiti WITH ENCRYPTION i na ALTER VIEW kako bi zadržali enkripciju**

```
CREATE VIEW HumanResources.EmployeeList
WITH ENCRYPTION
AS
SELECT E.BusinessEntityID, P.LastName, P.FirstName
FROM Person.Person AS P
INNER JOIN HumanResources.Employee AS E
ON P.BusinessEntityID = E.BusinessEntityID
```

# Lekcija 2: Uskladištene procedure

- Šta su uskladištene procedure
- Prednosti korištenja
- Kreiranje procedura
- Izmjena procedura
- Brisanje procedura
- Rad sa ulaznim parametrima
- Rad sa izlaznim parametrima
- Skrivanje strukture uskladištene procedure



# Šta su uskladištene procedure

- **Interakcija aplikacije sa serverom baze podataka se obično vrši na dva načina**
  - SQL komande se šalju direktno iz aplikacije
  - Komande i grupe komande se mogu pohraniti na serveru i kao takve pozivati iz aplikacije
- **Može imate ulazne i izlazne parametre**
- **Može vratiti skupove podataka**
- **Izvršavaju se putem EXECUTE komande**
- **Mogu biti napisane i u .NET jeziku**
  - Ako govorimo o SQL Server RDBMS platformi

# Prednosti korištenja

- **Bolja sigurnost u cjelini**
- **Modularno programiranje**
- **Separacija logike baze podataka od aplikacije**
- **Mogućnost referenciranja objekata koji još nisu kreirani**
- **Bolje performanse**

# Kreiranje procedura

- Uskladištena procedura se pravi putem **CREATE PROCEDURE** komande

```
CREATE PROCEDURE Sales.GetSalesPersonNames
AS
BEGIN
    SELECT SP.BusinessEntityID, P.LastName, P.FirstName
    FROM Sales.SalesPerson AS SP
    INNER JOIN Person.Person AS P
    ON SP.BusinessEntityID = P.BusinessEntityID
    WHERE SP.TerritoryID IS NOT NULL
    ORDER BY SP.BusinessEntityID
END;
```

```
EXECUTE Sales.GetSalesPersonNames --izvršenje procedure
GO
```

# Izmjena procedura

- Uskladištena procedura se modifkuje putem ALTER PROCEDURE komande
  - Permisije ostaju iste

```
ALTER PROCEDURE Sales.GetSalesPersonNames
AS
BEGIN
    SELECT SP.BusinessEntityID, P.LastName, P.FirstName
    FROM Sales.SalesPerson AS SP
    INNER JOIN Person.Person AS P
    ON SP.BusinessEntityID = P.BusinessEntityID
    WHERE SP.TerritoryID IS NOT NULL
    AND SP.SalesQuota IS NOT NULL
    ORDER BY SP.BusinessEntityID
END;
```

# Brisanje procedura

- Uskladištena procedura se briše putem DROP PROCEDURE komande
- Ako nismo sigurni za naziv procedure postoji sistemski pogled za provjeru

```
SELECT SCHEMA_NAME(schema_id) AS SchemaName,  
       Name AS ProcedureName  
FROM sys.procedures
```

- **Brisanje**

```
DROP PROCEDURE Sales.GetSalesPersonNames  
GO
```



# Rad sa ulaznim parametrima

- **Parametri**

- Imaju @ prefiks, tip podatka i mogu imati default vrijednost

- **Uvijek radite validaciju ulaznih parametara**

```
CREATE PROCEDURE Sales.OrdersByDueDateAndStatus
@DueDate datetime, @Status tinyint = 5
AS
SELECT SalesOrderID, OrderDate, CustomerID
FROM Sales.SalesOrderHeader
WHERE DueDate = @DueDate
AND [Status] = @Status;
GO
```

```
EXEC Sales.OrdersByDueDateAndStatus '20140712',5;
EXEC Sales.OrdersByDueDateAndStatus '20131115';
EXEC Sales.OrdersByDueDateAndStatus @DueDate = '20110612',@Status = 5;
```

# Rad sa izlaznim parametrima

- **Obavezno je navesti OUTPUT**

- Prilikom deklaracije parametara
- Prilikom izvršenja

```
CREATE PROC Sales.GetOrderCountByDueDate
@DueDate datetime, @OrderCount int OUTPUT
AS
    SELECT @OrderCount = COUNT(1)
    FROM Sales.SalesOrderHeader
    WHERE DueDate = @DueDate;
GO
```

```
DECLARE @DueDate datetime = '20140712';
DECLARE @OrderCount int;
EXEC Sales.GetOrderCountByDueDate @DueDate, @OrderCount OUTPUT;
SELECT @OrderCount;
```

# Skrivanje strukture uskladištene procedure

- **Obfuscating**

- Putem WITH ENCRYPTION klauzule
- Struktura procedure je pohranjena u kriptovanom obliku
- Generalno nije preporučljivo

- **Koristiti WITH ENCRYPTION i na ALTER PROCEDURE kako bi zadržali enkripciju**

```
CREATE PROCEDURE HumanResources.EmployeeList
WITH ENCRYPTION
AS
SELECT E.BusinessEntityID, P.LastName, P.FirstName
FROM Person.Person AS P
INNER JOIN HumanResources.Employee AS E
ON P.BusinessEntityID = E.BusinessEntityID
```

# Lekcija 3: Okidači

- Šta su okidači i njihovi tipovi
- Inserted i Deleted virtuelne tabele
- DDL okidači
- DML okidači



# Šta su okidači i njihovi tipovi

- **Specijalni tipovi uskladištenih procedura, koji se izvršavaju automatski kao rezultat određenog događaja**
  - DML na INSERT, UPDATE i DELETE
  - DDL na CREATE, ALTER i DROP
  - Login okidači na uspostavu sesije
- **AFTER okidači**
  - Izvršavaju se nakon određenog događaja
  - Dio su transakcije koja je njihov uzrok
  - Mogu vratiti transakciju koja je uzrok u početno stanje
- **INSTEAD OF okidači**
  - Izvršavaju alternativni kod

# Inserted i Deleted virtuelne tabele

- Omogućavaju pristup podacima prije i nakon izmjene
  - Samo AFTER i INSTEAD OF

Komanda	inserted	deleted
INSERT	insertovani zapisi	
DELETE		obrisani zapisi
UPDATE	modifikovani zapisi	originalni sadržaj zapisa

# DDL trigeri

- Prate promjene na objektima unutar baze podataka;
  - CREATE, ALTER, DROP i sl.
- Odlični za kontrolu pristupa i praćenje promjena nad objektima baze podataka;

```
CREATE TRIGGER Preventiva
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE, DROP_VIEW
AS
    PRINT 'Brisanje i izmjena nad objektima nisu dozvoljeni!'
    ROLLBACK ;
```

```
DROP VIEW HumanResources.vEmployee --neće raditi
```

# DML okidači

- Primjer pokazuje AFTER tip okidača koji se izvršava nakon UPDATE komande

```
CREATE TRIGGER TR_ProductReview_Update
ON Production.ProductReview
AFTER UPDATE AS
BEGIN
    SET NOCOUNT ON;
    UPDATE PR
    SET PR.ModifiedDate = SYSDATETIME()
    FROM Production.ProductReview AS PR
    INNER JOIN inserted AS I
    ON I.ProductReviewID = PR.ProductReviewID;
END;
```



# Pitanja

