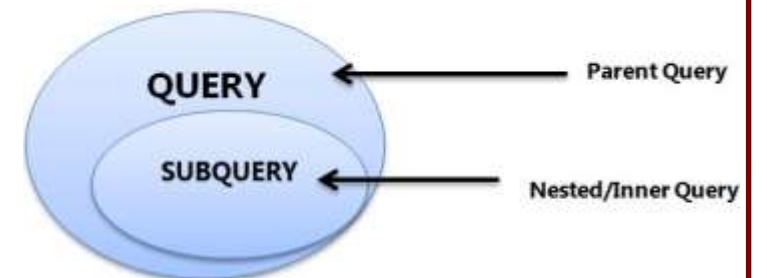


Baze podataka II

Modul 5 – Jezik SQL

Uvod u SQL/DML
Rad sa podupitima



Summary

- Uvod u podupite
- Korelacijski podupiti



Lekcija 1: Uvod u podupite

- Šta je podupit?
- Podupit kao izvorna tabela
- Podupit kao izraz
- Skalarni i tabelarni podupiti



Šta je podupit?

- **SELECT** komanda ugnježdena unutar druge:
 - SELECT, INSERT, UPDATE ili DELETE komande
 - ili drugog podupita
- **Zašto koristiti?**
 - Dijeljenje kompleksnih upita na logičke i jasnije cjeline
 - Kada rezultat upita treba da zavisi od izlaza drugog upita
 - ...

```
SELECT LastName, FirstName --vanjski upit
FROM Person.Person
WHERE Title IN
    (SELECT DISTINCT Title --unutarnji upit
     FROM Person.Person
      WHERE Title IS NOT NULL)
ORDER BY LastName
```

Podupit kao izvorna tabela

- **Podupit se postavlja u FROM klauzulu;**
 - Ponaša se kao zasebna tabela;
 - Upotreba ALIAS-a je mandatorna
- **Ovaj primjer „kreira“ tabelu koju koristi vanjski (OUTER) upit kao izvor podataka**

```
USE Northwind
GO
SELECT T.OrderID, T.CustomerID
FROM ( SELECT OrderID, CustomerID FROM Orders ) AS T
```

Podupit kao izraz

- **Podupit može biti na mnogim pozicijama unutra vanjskog upita:**
 - SELECT lista, FROM klauzula, WHERE ili HAVING uslov
 - kao izraz na navedenim pozicijama

```
SELECT Name, Weight,  
  (SELECT AVG (Weight) FROM Production.Product)  
AS Prosjek',  
  Weight - (SELECT AVG (Weight) FROM Production.Product)  
AS Razlika'  
FROM Production.Product  
WHERE Weight IS NOT NULL AND Weight > 800  
ORDER BY Weight DESC
```

Skalarna i tabelarni podupiti

- **Podupit može da vrati dvije vrste skupova podataka**
 - Skalarni i tabelarni
 - Zavisno od potrebe poslovnog procesa možemo koristiti bilo koji od navedenih
- **Skalarni vraća jedan zapis i isti prosljeđuje vanjskom upitu**
- **Tabelarni vraća podatke u obliku tabele i iste prosljeđuje vanjskom upitu**

Lekcija 2: Korelacijski podupiti

- Šta je korelacijski podupit ?
- Kako se pravi?
- Podupit vs. JOIN
- Oponašanje HAVING klauzule



Šta je korelacijski podupit ?

1. Vanjski upit prosljeđuje vrijednost u unutarnji upit

2. Unutarnji upit koristi vrijednosti kako bi se izvršio

```
SELECT SalesOrderID, CustomerID
FROM Sales.SalesOrderHeader AS SOH
WHERE 10 > ( SELECT OrderQty
              FROM Sales.SalesOrderDetail AS SOD
              WHERE SOH.SalesOrderID = SOD.SalesOrderID
                AND SOD.ProductID = 778 )
```

3. Unutarnji upit vraća vrijednost vanjskom

4. Postupak se nastavlja za naredni red iz vanjskog upita

Kako se pravi

- **Najbolje da se podupit pravi u fazama**

- Napisati njegove komponente odvojeno

```
SELECT Name, ListPrice
FROM Production.Product
      AS P1
```

```
SELECT AVG (ListPrice)
FROM Production.Product
      AS P2
```

- Analizirati logiku i izvršiti spajanje

```
SELECT Name, ListPrice
FROM Production.Product AS P1
WHERE 3500 < (SELECT AVG (ListPrice)
              FROM Production.Product AS P2
              WHERE P1.ProductID = P2.ProductID)
```

Podupit vs. JOIN

```
SELECT Name
FROM Production.ProductSubcategory
WHERE ProductCategoryID IN
    (SELECT ProductCategoryID
     FROM Production.ProductCategory
     WHERE Name = 'Bikes')
```

```
--Join verzija
SELECT PS.Name
FROM Production.ProductSubcategory AS PS
    INNER JOIN Production.ProductCategory AS PC
    ON PC.ProductCategoryID = PS.ProductCategoryID
WHERE PC.Name = 'Bikes'
```

Oponašanje HAVING klauzule

```
USE pubs
SELECT t1.type, t1.title, t1.price
FROM titles AS t1
WHERE t1.price > ( SELECT AVG(t2.price)
                   FROM titles AS t2
                   WHERE t1.type = t2.type )
```

```
USE pubs
SELECT t1.type, t1.title, t1.price
FROM titles AS t1
      INNER JOIN titles AS t2
            ON t1.type = t2.type
GROUP BY t1.type, t1.title, t1.price
HAVING t1.price > AVG(t2.price)
```

Pitanja

