

Погледи

Коришћењем SELECT упита, повезивањем неколико табела, добијају се потребни резултати из стандардне релационе базе података.

SQL сервер омогућава да се SELECT изразу додели име и да се он сачува у бази података.

Тако именован и сачуван SELECT израз се назива View (поглед).

Поглед заузима веома мало меморијског простора пошто он не садржи податке, већ само дефиницију SELECT израза који дефинише како се подаци приказују.

Зато се поглед третира као виртуелна табела и користимо је у FROM клаузули наводећи име погледа.

Поглед може да буде краиран над подацима које враћа један или више других погледа (View објеката).

Други разлог коришћења погледа јесте безбедоносни; ако не смеју сви корисници да виде све податке у табелама, креирају се погледи који реферишу на табелу и свако од њих у SELECT листи приказује колоне које одређена група корисника сме да види.

Администратор базе података забрањује директан приступ табели, а одређеним групама корисника дозвољава да гледају свој поглед у складу са безбедоносним правилима.

Креирање погледа

Прво се креира SELECT упит, тестира се његова ваљаност, па се сними у базу података под јединственим именом.

```
CREATE VIEW sema.ImePogleda
```

```
AS
```

```
SELECT naredba...
```

```
...
```

Назив шеме се може и изоставити и тада се користи дифолтна шема SQL базе података корисника.

Пример:

```
CREATE VIEW dbo.KupciFakture
```

```
AS
```

```
SELECT
```

```
    SC.MusterijaID,
```

```
    SC.MusterijaIme,
```

```
    SO.NarudzbenicaID,
```

```
    SO.DatumNarucivanja,
```

```
    CONVERT(char(10), DatumNarucivanja, 104) AS
```

```
        FormatiranDatumNarucivanja
```

```
FROM
```

```
    dbo.Musterije SC INNER JOIN
```

```
    dbo.Narudzbenice SO ON
```

```
    SC.MusterijaID = SO.MusterijaID;
```

У примеру се везују две табеле dbo.Musterije и dbo.Narudzbenice и њихове фактуре.

Табелама у FROM клаузули су дати алијаси како би упит био прегледнији.

У SELECT листи је додата израчуната колона под називом FormatiranDatumNarucivanja где се помоћу CONVERT функције датум фактурисања (DatumNarucivanja) конвертује у текст по формату за датум број 104 а то је формат dd.mm.yyyy.

Када се покрене пример, не добијају се никакви подаци као резултат извршавања већ само информација да је команда успешно извршена – Commands completed successfully.

Ове информације описују команду CREATE VIEW.

Креирани поглед се може видети у панелу Object Explorer у фолдеру View.

View објекат се третира као виртуелна табела и користи се у FROM клаузули :

```
SELECT *
```

```
FROM dbo.KupciFakture;
```

У резултату се добијају записи са свим купцима са њиховим фактурама; због INNER JOIN везе неће бити приказани сви купци који немају ниједну фактуру.

Такође, подаци су враћени сортирани по случајном редоследу пошто се не користи ORDER BY клаузула, а поглед не може да има ORDER BY клаузулу.

Зато се она пише у упиту који реферише на поглед:

```
SELECT *
```

```
FROM dbo.KupciFakture
```

```
ORDER BY MusterijalID, NarudzbenicalID;
```

Сада ће резултат бити сортиран по првој колони MusterijalID а затим по колони NarudzbenicalID.

Измена постојећег погледа

Измена погледа се врши наредбом:

```
ALTER VIEW sema.ImePogleda
```

```
AS
```

```
SELECT naredba ...
```

```
...
```

Уписује се другачија SELECT наредба без икаквог ограничења у односу на претходну верзију.

Може се потпуно изменити структура података која враћа поглед.

Али, сва безбедоносна подешавања се задржавају, тако да не морају поново да се постављају.

Брисање постојећег погледа:

```
DROP VIEW sema.ImePogleda;
```

Брисањем погледа се брише само поглед не и табеле које референцира али се бришу и сигурносна подешавања.

Ако се после брисања погледа, креира нов поглед са истим именом, неопходно је поново подесити безбедоносне параметре.

Опције при креирању погледа

- Опција WITH ENCRYPTION

Ова опција се користи како би се сакрила дефиниција погледа – скрипт који га је креирао; овако се чувају ауторска права али и сакривање информација који би омогућиле анализу кода или упад у систем.

Пример:

```
CREATE VIEW dbo.EvropskeZemlje
```

```
WITH ENCRYPTION
```

```
AS
```

```
SELECT DrzavaID, DrzavaIme, Oblast
```

```
FROM dbo.Drzave
WHERE Oblast = N'Evropa';
```

Када се стартује скрипт са EXEC sp_helptext 'dbo.EvropskeZemlje' добија се порука:
 "The text for object 'dbo.EvropskeZemlje' is encrypted."

- Опција WITH SCHEMABINDING

Креирани поглед се реферише на једну или више табела.

Међутим, то није препрека да се обришу нека од тих табела, али ће поглед престати да функционише.

Опција WITH SCHEMABINDING спречава овакве ситуације: ако се мења структура реферисаних табела па поглед неће више функционисати, сада измене неће бити дозвољене.

- Опција WITH CHECK

Поглед се често користи како би се креирао филтер над подацима, и за то се користи WHERE клаузула.

Ако се измене врше преко погледа могуће је да оне нису у складу са WHERE клаузулом и зато неће бити приказане кроз дати поглед.

Са овом опцијом се онемогућава било каква измена или додавање записа после којих ти записи нису у складу са WHERE клаузулом и неби били приказани кроз поглед.

Ускладиштене процедуре

Ускладиштене процедуре (stored procedures) су SQL скрипт који је сачуван у бази података под одређеним именом.

Скрипт има улазне и излазне параметре и зато ускладиштене процедуре изгледају попут функција у програмским језицима.

Користе се за:

- Енкапсулацију сложених скрипти које врше обраду података или администраторске задатке
- Креирање извештаја
- Као интерфејс за INSERT, UPDATE, DELETE акција које се извршавају ка подацима у табелама базе података
- Приликом покретања скрипти које раде по захтеву или по задатом временском распореду за различите обраде података

Технички гледано, процедуре су функције чија је повратна вредност увек тип int.

Ова повратна вредност се може искористити по жељи, али је неписано правило да се користи као статус извршавања процедуре.

Ако је повратна вредност нула, процедура је успешно извршена.

Ако је повратна вредност различита од нула, процедура није успешно извршена а повратна вредност најчешће представља код грешке.

Синтакса за креирање ускладиштене процедуре:

```
CREATE PROCEDURE sema.ImeProcedure
Lista parametara...
WITH opcije procedure...
AS
```

```
...
SQL skript
...
```

Пример:

```
CREATE PROCEDURE dbo.ListaKupaca
AS
SELECT CI.GradIme, C.MusterijaID, C.MusterijaIme, C.AdresaDostava
FROM dbo.Musterije AS C
INNER JOIN dbo.Gradovi AS CI
ON C.DostavaGradID = CI.GradID
ORDER BY CI.GradIme;
```

У процедури., SELECT упит приказује информације о купцима (табела Musterije) које повезује са листом градова (табела Gradovi) како би се добио назив града из кога је купац.

На крају се резултат упита сортира по називу града.

По покретању примера биће креирана ускладиштена процедура dbo.ListaKupaca која се може видети у панелу Object Explorer.

Са наредбом EXECUTE (EXEC) се покреће ускладиштена процедура:

```
EXECUTE dbo.ListaKupaca;
```

Резултат на екрану:

GradIme	MusterijaId	MusterijaIme	AdresaDostava
Apatin	345	Miki	RF15
Apatin	234	Ana	TK70A
Arandjelovac	098	Dejan	ASE22
Beograd	123	Milena	AHJ1
Beograd	987	Dragan	SD3
Beograd	987	Dragan	SD3
Sabac	056	Jovana	GH09