**Дәріс 8**

**Тақырыбы: Деректерді өңдеуді тиімді жүзеге асыру. Сақталатын процедуралардың орындалуы мен транзакциялар, триггерлер туралы түсінік.**

**Жоспар:**

**1. Деректерді өңдеуді тиімді жүзеге асыру.**

**2. Сақталған процедуралар түсінігі**

**3. Триггер туралы түсініктер**

**4.Транзакция түсінігі**

**Мақсаты: Деректерді өңдеуді тиімді жүзеге асыру жолдарын түсіну. Сақталатын процедуралар, триггерлер мен транзакциялар құру және қолдануды үйрену.**

 **Негізгі түсініктер: Сақталатын процедуралар, триггерлер мен транзакция.**

Негізгі әдебиет: [1-5]

**Сақталатын процедуралар**

SQL Server-мен жұмыс кезінде қолданушылар әр түрлі әрекеттерді орындайтын өз процедураларын құра алады. Сақталатын процедуралар мәліметтер қорының толық бағалы объектілері болып табылады, сондықтан олардың әрқайсысы нақты мәліметтер қорында сақталады. Егер Сақталатын процедуралар процедурасы бар мәліметтер қорының контексінде жүзеге асырылса, Сақталатын процедураларды тікелей шақыруға болады.

SQL Server-де сақтаушы процедураның бірнеше типтері бар:

* жүйелік Сақталатын процедуралары әр түрлі администрациялық әрекеттердің орындалуына арналған;
* қолданушылық Сақталатын процедуралар әр түрлі әрекеттерді орындайды. Сақталатын процедуралар – мәліметтер қорының толық бағалы объектісі. Сондықтан әрбір сақтаушы процедура нақты бір мәліметтер қорында орналасып, орындалады;
* уақытша сұраушы процедуралар аздаған уақытта ғана болады, кейін аутоматты түрде сервермен жойылады. Олар локальды және глобальды болып бөлінеді. Локальды уақытша Сақталатын процедуралар құрылған жердегі байланыстан шақырылады. Глобальды уақытша Сақталатын процедуралар осындай процедуралары бар кез-келеген сервердің байланысы үшін қажет.

Сақталатын процедуралардың құрылуы үшін келесі тапсырмалар шешіледі:

* құрылатын сақтаушы процедураның типін анықтау: уақытша ма немесе қолданушылық па;
* ену құқығын жоспарлау;
* сақтаушы процедураның параметрлерін анықтау;
* сақтаушы процедураның кодын жасау.

Сақтаушы процедураның жаңасын құру және бар сақтаушы процедураға өзгеріс енгізу келесі команданың көмегімен жүзеге асырылады:

<процедураны\_анықтау> : : =

{CREATE | ALTER } PROC [EDURE] процедураның\_аты

[; номер]

[{@ параметрдің\_аты мәліметтер\_типі } [VARING]

 [=default] [OUTPUT ] [,…n]

[WITH {RECOMPILE | ENCRYPTION | RECOMPILE, ENCRYPTION}]

[FOR REPLICATION]

AS

 sql\_операторлар […n]

sp\_, #, ## префикстері арқылы құрылатын процедураның жүйелік немесе уақытша түрін анықтауға болады.

CREATE PROCEDURE – құрылатын Сақталатын процедураларды нақты мәліметтер қорында орналастыру үшін орындау керек. Атаудағы номер – сақтаушы прцедураның процедуралар тобындағы идентификационды номерін анықтайды. Кіріс- шығыс мәліметтерінің берілуі үшін құрылатын Сақталатын процедураларда параметрлер қолданылады. Олардың аттары локальды айнымалылардың аттары сияқты @ символынан басталуы керек. OUTPUT кілттік сөзімен CURSOR мәліметтер типі сақтаушы процедураның кіріс параметрі ретінде қолданылады. VARING кілттік сөзі CURSOR типі бар OUTPUT параметрімен бірге қолданылады. Ол шығу параметрлері нәтижелендіруші жиын болғандығын анықтайды. DEFAULT кілттік сөзі үнсіздік бойынша сәйкес параметрді қабылдайтын мәнді көрсетеді. RECOMPILE – жүйеге сақтаушы процедураның әрбір шақырылуында оның орындалу жоспарын құруға бұйырады. FOR REPLICATION параметрі мәліметтердің репликациясы мен құрылатын сақтаушы процедураның қосылуы кезінде публикациядағы мақала ретінде қажет. ENCRYPTION кілттік сөзі серверге сақтаушы процедура кодының шифрленуін орындауды бұйырады. Ол сақтаушы процедураның жұмысын тарататын авторлық алгоритмдердің қолданылуынан қорғауды қамтамасыз етеді. RETURN командасы сақтаушы процедурадан шығуды жүзеге асырады.

Сақтаушы процедураның өшірілуі DROP PROCEDURE {процедура\_аты} [,…n] командасымен жүзеге асырылады.

Сақтаушы процедураның орындалуы үшін келесі команда қолданылады:

[[EXEC [UTE ] процедура\_аты [; номер]

[[@параметр\_аты=] {мән | @ айнымалы\_аты}

[OUTPUT] | [DEFAULT] ] [,…n]

**Триггерлерді программалау**

Триггерлер сақталатын процедуралардың бір түрі болып саналады. Деректерді манипуляциялау (DML) тілінің операторларын орындайды. Триггерлер деректердің тұтастығын қадағалауға, сонымен бірге транзакцияларды өткізуде қолданылады.

Триггер – реляциялық ДҚ ішінде белгілі бір жағдайларға (ситуацияларға) байланысты орындалатын компиляцияланған SQL –процедура. Триггерлер қолданушылар үшін ыңғайлы, сонда да оларды қолдануда енгізу-шығару операциялары үшін қосымша ресурстарды талап етеді. Егер осы ресурстарды тиімді пайдалануда сақталатын процедуралар мен қолданбалы бағдарламалар қолданылса, триггерлерді қолданбауға болады.

Триггерлер SQL Server-дің ДҚ-ының деректерінің тұтастығын қолдайтын ерекше құрал.

Әр триггер нақты бір кестеге арналады. Ондағы деректерді модификациялау бір транзакция сияқты болады. Қате кетсе немесе деректердің тұтастығы бұзылса, осы транзакция “жүріп” өтеді. Триггер жасаған өзгертулер орындалмайды. Триггер тиімді, сонымен бірге қауіпті құрал. Логикасы дұрыс жұмыс істемеген триггер тұтас ДҚ-ын жойып жібере алады, сондықтанда оларды тексеру керек.

Кәдімгі ішкі бағдарламалардан айырмашылығы – триггерлік жағдайда оның жұмысы ашық түрде көрінбейді.

Триггерлер арқылы келесідей мақсатқа жетуге болады:

- енгізілетін деректердің дұрыстығын тексереді және деректердің тұтастығын күрделі шектеуде қолданады;

- кестені толықтыру кезінде кейбір іс-әрекеттер туралы хабар береді, т.б.

Триггерлік жағдайлар INSERT, DELETE және UPDATE -ден тұрады.

Триггерлерді дұрыс қолдана білгенде қуатты механизм бола алады, себебі, стандартты функциялар ДҚ ішінде сақталады және “келісілген түрде” әрбір жаңарған сайын белсенді болып тұрады.

MS SQL Server ортасында триггер былайша құрылады және өзгертіледі:

{ CRERATE | ALTER } TRIGGER триггер\_аты

ON {кесте\_аты |көрсету\_аты}

[ WITH ENCRYPTION ]

{

{{ FOR | AFTER | INSTEAD OF}

{[ INSERT ] [,] [ DELETE ] [,] [ UPDATE ]}

[WITH APPEND]

[ NOT FOR REPLICATION ]

AS

Sql\_оператор […n]

} |

{{ FOR | AFTER | INSTEAD OF }{[INSERT] [,]

[UPDATE] }

[WITH APPEND]

[ NOT FOR REPLICATION ]

AS

{ IF UPDATE (баған\_аты)

[{AND | OR} UPDATE (баған\_аты)] […n]

|

IF (COLUMNS\_UPDATES(оператор\_ сұрыптау\_биті)}

өзгерту\_маскасының \_биті)

{оператор\_ салыстыру\_биті} маска\_биті [...n]}

sql\_оператор […n] } }

Мысал,

1) CREATE TRIGGER Триггер

ON Сатушы1

FOR INSERT

AS

BEGIN

 PRINT 'енгізуге болмайды'

END

 2) CREATE TRIGGER Триггер14

ON Сатушы1

FOR DELETE, INSERT, UPDATE

WITH APPEND

AS

BEGIN

 PRINT 'енгізуге болмайды'

END

Триггер ағымдағы бір ДҚ –на құрылуы мүмкін, басқа да ДҚ-на көңіл аударуға болады, сонымен бірге алыс орналасқан серверлерге де көңіл аударуға болады.

**WITH ENCRYPTION** аргументі көрсетілгенде, администратормен қоса ешкім қолдана алмау үшін сервер триггердің кодын шифрлайды.

Шифрлау авторлық алгоритмдерді жасыру үшін қолданылады.

**Транзакцияларды орындау**

**Транзакция дегеніміз** – бір бүтін ретінде қарастырылатын SQL-инструкциялардың тізімі. Транзакцияға кіретін инструкциялар бір-бірімен байланысқан іс-әрекеттер орындайды. Әр инструкция жалпы міндеттің бір бөлігін орындайды, бірақ ол бөліктің де шешімін табу керек, барлық инструкцияларды орындау керек.

ДҚ мазмұнын өзгертетін транзакция ұғымының бағдарлама үшін маңызы үлкен, себебі транзакция оның тұтастығын қамтамасыз етеді. Реляциялық ДҚ транзакциялар келесі ережеге бағынады:

Транзакцияға енетін инструкциялар бөлінбейтін бүтін болып саналады. Барлық инструкциялар сәтті аяқталған болады немесе біреу сияқты ешқайсысы орындалмауы керек (сурет19).



11-сурет.SQL-дегі транзакция түсінігі

Транзакцияның концепциясы – клиент-серверлік деректер қорының бір бөлігі. Транзакция механизмін қолдану – ДҚБЖ-нің жетілгендігінің көрсеткіші. Көпқолданушылар режимінде жеке жұмыс істеудің негізін құрайды, яғни бір ДҚ-мен паралелль бірнеше қолданушы немесе қолданбалы бағдарламалар жұмыс істей алады. ДҚБЖ-нің негізгі міндетінің бірі – жеке жұмыс істеуді қамтамасыз ету, яғни әр қолданушы ДҚ-мен өзі ғана жұмыс істеп отырғандай болуы, ДҚБЖ-нің мұндай міндетін транзакцияның параллельдігі деп атайды.