AXBOROT TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI MUHAMMAD AL-XORAZMIY NOMLI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

Fan nomi: SQLda dasturlash

3-4-LABORATORIYA ISHI

Bajardi: Sh.No’monjonov

Guruh: SQL001-L2

Tekshirdi: Xojieva Nasiba Jumabaevna

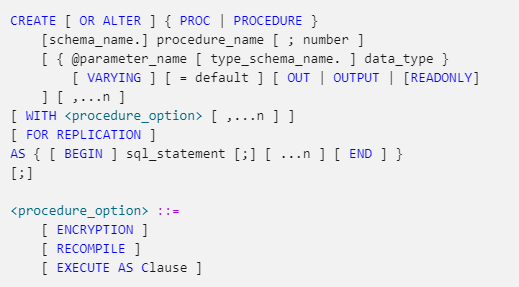
**Toshkent 2022**

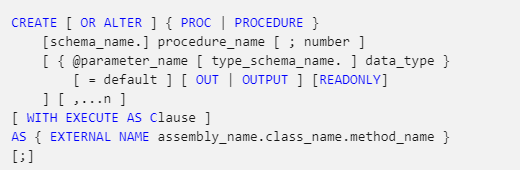
**Mavzu: Funktsiyalar va saqlangan protseduralarni yaratish, o'zgartirish, qo'llash va o'chirish**

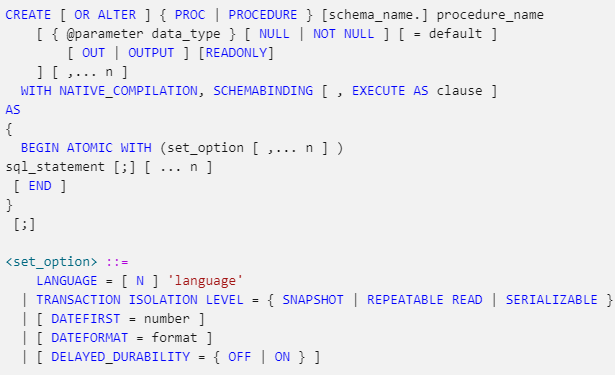
SQL Server, Azure SQL ma'lumotlar bazasi va Analytics Platform System (PDW) da Transact-SQL yoki umumiy til ish vaqti (CLR) saqlangan protsedurasini yaratadi. Saqlangan protseduralar boshqa dasturlash tillaridagi protseduralarga o'xshaydi, chunki ular:

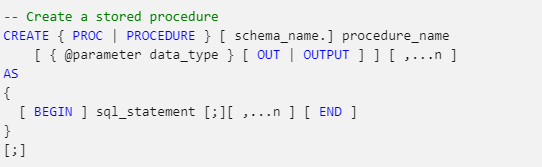
* Kirish parametrlarini qabul qiling va chiqish parametrlari ko'rinishidagi bir nechta qiymatlarni chaqiruvchi protsedura yoki partiyaga qaytaring.
* Ma'lumotlar bazasida operatsiyalarni bajaradigan, shu jumladan boshqa protseduralarni chaqiruvchi dasturlash bayonotlarini o'z ichiga oladi.
* Muvaffaqiyat yoki muvaffaqiyatsizlikni (va muvaffaqiyatsizlik sababini) ko'rsatish uchun qo'ng'iroq qilish tartibi yoki to'plamiga holat qiymatini qaytaring.

**Sintaksis:**

****

****

****

****

## Arguments

#### OR ALTER

**Qo'llaniladi: Azure SQL ma'lumotlar bazasi, SQL Server** (SQL Server 2016 (13.x) SP1 dan boshlab).

Agar u allaqachon mavjud bo'lsa, protsedurani o'zgartiradi.

**schema\_name:** Jarayon tegishli bo'lgan sxemaning nomi. Protseduralar sxemaga bog'langan. Agar protsedura yaratilganda sxema nomi ko'rsatilmagan bo'lsa, protsedura yaratayotgan foydalanuvchining standart sxemasi avtomatik ravishda tayinlanadi.

#### **procedure\_name:** Jarayonning nomi. [Jarayon nomlari identifikatorlar](https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-identifiers?view=sql-server-ver15) qoidalariga mos kelishi va sxemada yagona bo'lishi kerak.

**Transact-SQL protseduralari bo'yicha ko'rsatmalar** :

* Parametr sifatida barcha Transact-SQL ma'lumotlar turlaridan foydalanish mumkin.
* Jadval qiymatli parametrlarni yaratish uchun foydalanuvchi tomonidan belgilangan jadval turidan foydalanishingiz mumkin. Jadvalda ko'rsatilgan parametrlar faqat INPUT parametrlari bo'lishi mumkin va READONLY kalit so'zi bilan birga bo'lishi kerak. Qo'shimcha ma'lumot olish uchun [Jadval qiymatidagi parametrlardan foydalanish (ma'lumotlar bazasi mexanizmi) bo'limiga qarang.](https://docs.microsoft.com/en-us/sql/relational-databases/tables/use-table-valued-parameters-database-engine?view=sql-server-ver15)
* **kursor** ma'lumotlar turlari faqat OUTPUT parametrlari bo'lishi mumkin va VARYING kalit so'zi bilan birga bo'lishi kerak.

**CLR protseduralari bo'yicha ko'rsatmalar**:

* Boshqariladigan kodda ekvivalentga ega bo'lgan barcha mahalliy SQL Server ma'lumotlar turlari parametr sifatida ishlatilishi mumkin. CLR turlari va SQL Server tizimi ma'lumotlar turlari o'rtasidagi yozishmalar haqida qo'shimcha ma'lumot olish uchun [CLR parametr ma'lumotlarini xaritalash](https://docs.microsoft.com/en-us/sql/relational-databases/clr-integration-database-objects-types-net-framework/mapping-clr-parameter-data?view=sql-server-ver15) -ga qarang . SQL Server tizimining ma'lumotlar turlari va ularning sintaksisi haqida ko'proq ma'lumot olish uchun [Ma'lumotlar turlari (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15) ga qarang.
* Jadval qiymati yoki **kursor** ma'lumotlar turlaridan parametr sifatida foydalanish mumkin emas.
* Agar parametrning ma'lumotlar turi foydalanuvchi tomonidan belgilangan CLR turi bo'lsa, sizda bu turdagi EXECUTE ruxsati bo'lishi kerak.

**Endi esa SQL Serverda jadval yaratib insert, update, delete metodlarni amaliyotda sinab ko’ramiz:**

**Dastur kodi:**

**1-qadam. Jadval yaratish. Foydalanuvchi ma’lumotlarini kiritib chiqamiz:**

**create table Employee (Employee\_ID int primary key identity(1,1), Employee\_Name varchar(50), Employee\_Designation varchar(50))**

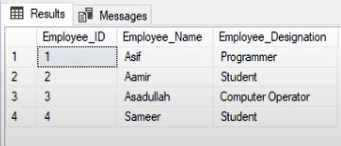
**insert into Employee values("Sameer', 'Student')**

**insert into Employee values("Asif, 'Programmer')**

**insert into Employee values("Asadullah, 'Computer Operator')**

**insert into Employee values("Aamir, 'Student')**

**select \* from Employee**

****

**2-qadam. Insert method**

**create procedure InsertRecord**

**@Emp\_Name varchar (50).**

**@Emp\_Design varchar (50)**

**as**

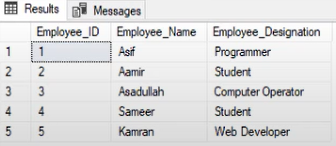
**begin**

**insert into Employee values(@Emp\_Name, @Emp\_Design)**

**end**

**execute InsertRecord "Kamran", "Web Developer**

**select \* from Employee**

****

**3-qadam. Update method**

**create procedure UpdateRecord**

**GID int,**

**GEmp\_Name varchar (50),**

**QEmp\_Designation varchar(50)**

**as**

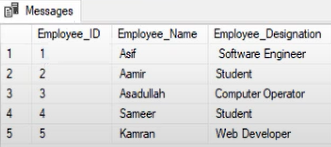
**begin**

**update Employee set Employee\_Name-eEmp\_Name, Employee\_Designation @lmp\_Designation where Employee\_ID-EID**

**end**

**exec UpdateRecord 1, "Asif',' Software Engineer'**

**select \* from Employee**

****

**4-qadam. Delete method**

**create procedure DeleteRecord**

**(@Employee\_ID int)**

**as**

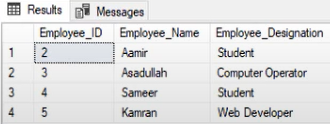
**begin**

**delete from Employee where Employee ID-@Employee ID**

**end**

**execute DeleteRecord 1**

**select \* from Employee**

****