

Stored Procedure

(Лаборатори №3)

У. Төрболд

ХШУИС, Мэдээллийн технологийн IV-р түвшний оюутан, turbold1125@gmail.com

1. ОРШИЛ

Энэхүү лабораторид stored procedure буюу тодорхой даалгаврыг багтаасан урьдчилан эмхэтгэсэн SQL хэллэг юм. Эдгээр нь кодыг дахин ашиглах, засвар үйлчилгээ, аюулгүй байдлыг сайжруулдаг. Энэхүү тайлан нь ажилтан болон үйлчлүүлэгчтэй холбоотой янз бүрийн функцуудэд зориулсан stored procedure бичихийг судлах болно.

2. ЗОРИЛГО

Энэхүү лабораторын зорилго Stored procedure – ын талаар практик дээр хэрэгжүүлэх бөгөөд үүнийг хэрэгжүүлснээр кодын зохион байгуулалтыг сайжруулах, дахин ашиглах, гүйцэтгэлийг сайжруулах боломжтой болно. Үүний тулд дараах зорилтуудыг хэрэгжүүлэхийг зорьлоо.

- Ажилтны дэлгэрэнгүй мэдээллийг хэвлэх.
- Тодорхой шалгуурыг үндэслэн шилдэг ажилтныг тодорхойлох.
- Хэрэглэгчийн тодорхойлсон хэмжүүр дээр үндэслэх оны шилдэг захиалагчийг тодорхойлох.
- Үйлчлүүлэгчдийн оруулсан орлогыг шинжилж, төлбөрийн жагсаалт гаргах.

3. ОНОЛЫН СУДАЛГАА

3.1 Stored procedure

Stored procedure гэдэг нь энгийн бөгөөд нийлмэл шинж чанартай, олон дахин ашиглагдах SQL командуудыг нэгтгэсэн өгөгдлийн сангийн объект юм. Хэрэв танд дахин дахин бичдэг query байгаа бол түүнийг stored procedure болгон хадгалаад дуудаж ажиллуулах боломжтой. Stored procedure – д параметруудийг дамжуулж болох бөгөөд ингэснээр stored procedure нь дамжуулсан параметрийн утгууд дээр ажиллах боломжтой болно.

Syntax:

```
CREATE PROCEDURE procedure_name
(
    @parameter1 data_type [DEFAULT value], -- параметрийн утга, төрөл, default утга өгж болно
    @parameter2 data_type,
    ...
)
AS
BEGIN
```

```

-- процедурын их бие буюу хийх функц
SELECT ...
FROM ...
WHERE ...
UPDATE ...
DELETE ...
DECLARE ... - variable зарлах боломжтой
END;

```

4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Ажилтны дэлгэрэнгүй мэдээллийг хэвлэж гаргах stored procedure бич.

```

-- GetEmployeeDetails stored procedure үүсгэх
CREATE PROCEDURE GetEmployeeDetails @EmpID INT
AS
BEGIN
    SELECT * -- stored procedure эхлүүлэх
    FROM Employees -- Employee хүснэгтийн бүх баганыг сонгох
    WHERE EmployeeID = @EmpID; -- EmployeeID нь @EmpID тай тохирч байх нөхцөл
END; -- stored procedure төгсгөх

-- GetEmployeeDetails дуудаж параметрт 9 утгыг оноох
EXEC GetEmployeeDetails @EmpID = 9;

```

Үр дүн:

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	Address	City	Region	PostalCode	Country	HomePhone	Extension	Photo	Notes	ReportsTo	PhotoPath
1	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15 00:00:00.000	7 Houndstooth Rd.	London	NULL	W32 7LT	UK	(71) 555-4444	452	0x151C2...	Anne has a BA ...	5	http://acow

4.2 1998 оны шилдэг ажилтанг захиалга хүргэж өгсөн тоо, хүргэж өгдөг дундаж хугацаагаар нь шалгаруулан олох stored procedure бич.

```

-- FindBestEmployee1998 stored procedure үүсгэх
CREATE PROCEDURE FindBestEmployee1998
AS
BEGIN
    SELECT TOP 1 -- LIMIT 1 тай эквивалент
        e.EmployeeID, e.FirstName, e.LastName,
        COUNT(o.OrderID) AS NumOrdersDelivered, -- Хүргэгдсэн захиалгын тоо
        AVG(DATEDIFF(DAY, o.OrderDate, o.ShippedDate)) AS AvgDeliveryTime -- Хүргэлтийн дундаж хугацаа
    FROM Employees AS e, Orders AS o
    WHERE YEAR(o.ShippedDate) = 1998 and e.EmployeeID = o.EmployeeID -- 1998 онд
    -- хүргэгдсэн байх
    -- бүлэглэх
    GROUP BY e.EmployeeID, e.FirstName, e.LastName
    ORDER BY NumOrdersDelivered DESC, AvgDeliveryTime ASC -- захиалгын тоог бууруулж,
    -- хугацааг өгсөх замаар эрэмбэлэх
END; -- stored procedure төгсгөх

-- FindBestEmployee1998 дуудах
EXEC FindBestEmployee1998

```

Үр дүн:

	EmployeeID	FirstName	LastName	NumOrdersDelivered	AvgDeliveryTime
1	3	Janet	Leverling	42	8

4.3 Өмнөх (1) гэсэн даалгавраар хийсэн stored procedure өмнө нь өгөгдлийн санд үүссэн

бол устгаад шинээр үүсгэдэг байхаар (1) гэсэн stored procedure – ийн кодыг өөрчил.

```
-- GetEmployeeDetails procedure байгаа тохиолдолд устгах
DROP PROCEDURE IF EXISTS GetEmployeeDetails;
-- GetEmployeeDetails stored procedure үүсгэх
CREATE PROCEDURE GetEmployeeDetails @EmpID INT
AS
BEGIN
    SELECT *
    FROM Employees
    WHERE EmployeeID = @EmpID;
END;

-- GetEmployeeDetails дуудаж параметрт 1 утгыг оноох
EXEC GetEmployeeDetails @EmpID = 1;
```

Үр дүн:

EmployeeID	LastName	FirstName	Title	TitleOfCourt...	BirthDate	HireDate	Address	City	Region	PostalCode	Country	HomePhone	Extension	Photo	Notes	ReportsTo	PhotoPath
1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08...	1992-05-01 ...	507 - 20th ...	Seattle	WA	98122	USA	(206) 555-9857	5467	0x15...	Educa...	2	http://accweb

4.4 Оны шилдэг захиалагчийг өөрийн сонгон шалгаруулалтаар олох stored procedure бич.

```
-- FindBestSubscriberByYear stored procedure үүсгэх
CREATE PROCEDURE FindBestSubscriberByYear @Year INT
AS
BEGIN
    SELECT TOP 1
    c.CustomerID, c.CompanyName, SUM(od.Quantity * od.UnitPrice) AS TotalSpent,
    @Year
    FROM Customers as c
    INNER JOIN Orders o ON c.CustomerID = o.CustomerID
    INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
    WHERE YEAR(o.OrderDate) = @Year
    GROUP BY c.CustomerID, c.CompanyName
    ORDER BY TotalSpent DESC
END;

-- FindBestSubscriberByYear дуудах, @Year параметрт 1998 утга оноох
EXEC FindBestSubscriberByYear @Year = 1998;
```

Үр дүн:

	CustomerID	CompanyName	TotalSpent	(No column name)
1	SAVEA	Save-a-lot Markets	42806.25	1998

4.5 (4) даалгавраар хийсэн stored procedure ашиглан бүх оны шилдгийн шилдэг захиалагчийн тухай мэдээллийг хэвлэж гаргах stored procedure бич.

```
-- FindBestSubscriberByYear stored procedure үүсгэх 1996 - 1998 хүртэл
CREATE PROCEDURE GetBestSubscribersAllYears (@Start INT = 1996, @Current INT = 1998)
AS
BEGIN
    WHILE @Start <= @Current
    BEGIN
        EXEC FindBestSubscriberByYear @Year = @Start;
    END
END;
```

```

SET @Start = @Start + 1;      -- year + 1
END;
END;

-- GetBestSubscribersAllYears дуудах
EXEC GetBestSubscribersAllYears;

```

Үр дүн:

	CustomerID	CompanyName	TotalSpent	(No column name)
1	ERNSH	Ernst Handel	17170.40	1996

	CustomerID	CompanyName	TotalSpent	(No column name)
1	QUICK	QUICK-Stop	64238.00	1997

	CustomerID	CompanyName	TotalSpent	(No column name)
1	SAVEA	Save-a-lot Markets	42806.25	1998

4.6 Хамгийн их орлого оруулсан үйлчлүүлэгчдийн нийт орлогын 3 хувьтай тэнцэх мөнгийг олгохоор болжээ. Энэ жагсаалтыг үйлчлүүлэгчийн нэр, орлогын нийт мөнгөн дүн, 3 хувьтай тэнцэх мөнгөн дүн зэргээр хэвлэж гаргах stored procedure бич.

```

-- GeneratePayoutList stored procedure үүсгэх
CREATE PROCEDURE GeneratePayoutList @Customer INT
AS
BEGIN
    SELECT TOP (@Customer)      -- Үр дүнг заасан хэрэглэгчдийн тоогоор хязгаарлах
    c.CompanyName, SUM(od.Quantity * od.UnitPrice) AS TotalSpent, -- Нийт зарцуулсан дүн
    SUM((od.Quantity * od.UnitPrice) * 0.03) AS '3%',           -- Тооцоолсон 3% төлбөрийн дүн
    SUM(od.Quantity * od.UnitPrice) + SUM((od.Quantity * od.UnitPrice) * 0.03) AS Total
                                                                -- зарцуулсан + 3%
    FROM Customers as c
    INNER JOIN Orders o ON c.CustomerID = o.CustomerID
    INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
    GROUP BY c.CustomerID, c.CompanyName                      -- бүлэглэх
    ORDER BY TotalSpent DESC                                   -- буурахаар эрэмбэлэх
END;

-- GetBestSubscribersAllYears дуудаж 5 хүртэлх үйлчлүүлэгч харуулах
EXEC GeneratePayoutList @Customer = 5;

```

Үр дүн:

	CompanyName	TotalSpent	3%	Total
1	QUICK-Stop	117483.39	3524.501700	121007.891700
2	Save-a-lot Markets	115673.39	3470.201700	119143.591700
3	Ernst Handel	113236.68	3397.100400	116633.780400
4	Hungry Owl All-Night Grocers	57317.39	1719.521700	59036.911700
5	Rattlesnake Canyon Grocery	52245.90	1567.377000	53813.277000

5. ДҮГНЭЛТ

Stored procedure нь өгөгдлийн санг удирдахад ихээхэн давуу талтай бөгөөд энэхүү лаборатори нь stored procedure – ын хэрэглээг харуулахад оршино. Stored procedure нь кодыг дахин дахин бичихийг хялбарчилдаг. Энэ нь параметр ашигласнаар өгөгдөлд хяналттай хандах боломжийг

олгож зөвшөөрөлгүй өөрчлөлтийг хязгаарладаг.

6. Хавсралт

6.1.

```
CREATE PROCEDURE GetEmployeeDetails @EmpID INT
AS
BEGIN
    SELECT *
    FROM Employees
    WHERE EmployeeID = @EmpID;
END;
```

```
EXEC GetEmployeeDetails @EmpID = 9;
```

6.2.

```
CREATE PROCEDURE FindBestEmployee1998
AS
BEGIN
    SELECT TOP 1
        e.EmployeeID, e.FirstName, e.LastName,
        COUNT(o.OrderID) AS NumOrdersDelivered,
        AVG(DATEDIFF(DAY, o.OrderDate, o.ShippedDate)) AS AvgDeliveryTime
    FROM Employees AS e, Orders AS o
    WHERE YEAR(o.ShippedDate) = 1998 and e.EmployeeID = o.EmployeeID
    GROUP BY e.EmployeeID, e.FirstName, e.LastName
    ORDER BY NumOrdersDelivered DESC, AvgDeliveryTime ASC
END;
```

```
EXEC FindBestEmployee1998
```

6.3.

```
DROP PROCEDURE IF EXISTS GetEmployeeDetails;
CREATE PROCEDURE GetEmployeeDetails @EmpID INT
AS
BEGIN
    SELECT *
    FROM Employees
    WHERE EmployeeID = @EmpID;
END;
```

```
EXEC GetEmployeeDetails @EmpID = 1;
```

6.4.

```
CREATE PROCEDURE FindBestSubscriberByYear @Year INT
AS
BEGIN
    SELECT TOP 1
        c.CustomerID, c.CompanyName, SUM(od.Quantity * od.UnitPrice) AS TotalSpent, @Year
    FROM Customers as c
    INNER JOIN Orders o ON c.CustomerID = o.CustomerID
    INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
    WHERE YEAR(o.OrderDate) = @Year
    GROUP BY c.CustomerID, c.CustomerID, c.CompanyName
    ORDER BY TotalSpent DESC
END;
```

```
EXEC FindBestSubscriberByYear @Year = 1998;
```

6.5.

```
CREATE PROCEDURE GetBestSubscribersAllYears (@Start INT = 1996, @Current INT = 1998)
AS
BEGIN
    WHILE @Start <= @Current
    BEGIN
        EXEC FindBestSubscriberByYear @Year = @Start;
        SET @Start = @Start + 1;
    END;
END;

EXEC GetBestSubscribersAllYears;
```

6.6.

```
CREATE PROCEDURE GeneratePayoutList @Customer INT
AS
BEGIN
    SELECT TOP (@Customer)
        c.CompanyName, SUM(od.Quantity * od.UnitPrice) AS TotalSpent,
        SUM((od.Quantity * od.UnitPrice) * 0.03) AS '3%',
        SUM(od.Quantity * od.UnitPrice) + SUM((od.Quantity * od.UnitPrice) * 0.03) AS Total
    FROM Customers as c
    INNER JOIN Orders o ON c.CustomerID = o.CustomerID
    INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
    GROUP BY c.CustomerID, c.CompanyName
    ORDER BY TotalSpent DESC
END;

EXEC GeneratePayoutList @Customer = 5;
```