МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛОРУССКО-РОССИЙСКОГО ОБРАЗОВАНИЯ»

Лабораторная работа № 16.

**Язык SQL. Создание хранимых процедур**

Выполнил ст. гр.АСОИР-181

Ковалевский Т В

Проверил

Вайнилович Ю.В

Могилев, 2021

**Цель:** научиться создавать хранимые процедуры в СУБД MS SQL Server с использованием команд T-SQL; реализовать хранимые процедуры для вставки, удаления, изменения данных.

USE Observatory

GO

CREATE PROCEDURE EquipmentSmmary

AS

SELECT Оборудование.Описание as descrition, Оборудование.Цена, Оборудование.Оборудование\_id

FROM Оборудование

GO

CREATE PROCEDURE get\_souvenirs @get\_price VARCHAR(50)

AS

SELECT \*

FROM Сувениры ORDER BY

CASE @get\_price

WHEN 'маленькая сумма'

THEN

(SELECT SUM(Сувениры.Цена) as price

FROM Сувениры

GROUP BY Сувениры.Цена

HAVING SUM(Сувениры.Цена) < 14)

WHEN 'средняя сумма'

THEN

(SELECT SUM(Сувениры.Цена) as price

FROM Сувениры

GROUP BY Сувениры.Цена

HAVING SUM(Сувениры.Цена) > 14 and SUM(Сувениры.Цена) < 100)

WHEN 'Большая сумма'

THEN

(SELECT SUM(Сувениры.Цена) as price

FROM Сувениры

GROUP BY Сувениры.Цена

HAVING SUM(Сувениры.Цена) > 100)

END

GO

GO

CREATE PROCEDURE ContractClientByDate @d1 DATETIME, @d2 DATETIME

AS

SELECT \*

FROM Договоры\_с\_клиентами

WHERE Договоры\_с\_клиентами.Дата\_заключения\_договора BETWEEN @d1 and @d2

CREATE PROCEDURE ContractClientByServiseID @serviseID smallint

AS

SELECT \*

FROM Договоры\_с\_клиентами

INNER JOIN Услуга

ON Услуга.Услуга\_id = @serviseID

GROUP BY Договоры\_с\_клиентами.Услуга\_id

GO;

CREATE PROCEDURE PositionSalary @salary smallint

AS

SELECT Должность.Должность\_id, Должность.Зарплата, Должность.Наименование

FROM Должность

WHERE Должность.Зарплата > @salary

GO;

CREATE PROCEDURE FindPositionSalary @title VARCHAR(50)

AS

SELECT Должность.Должность\_id, Должность.Зарплата, Должность.Наименование

FROM Должность

WHERE @title = Должность.Наименование

GO;

CREATE PROCEDURE GetPlanetByDistance @distance int as

CASE @distance

WHEN 10

THEN

SELECT Космические\_объекты.[Расстояние\_от\_Земли св.лет] as length\_,

Космические\_объекты.Описание as title

FROM Космические\_объекты

GROUP BY Космические\_объекты.[Расстояние\_от\_Земли св.лет]

HAVING Космические\_объекты.[Расстояние\_от\_Земли св.лет] < @distance

WHEN 50

THEN

SELECT Космические\_объекты.[Расстояние\_от\_Земли св.лет] as length\_,

Космические\_объекты.Описание as title

FROM Космические\_объекты

GROUP BY Космические\_объекты.[Расстояние\_от\_Земли св.лет]

HAVING Космические\_объекты.[Расстояние\_от\_Земли св.лет] < @distance

WHEN 100

THEN

SELECT Космические\_объекты.[Расстояние\_от\_Земли св.лет] as length\_,

Космические\_объекты.Описание as title

FROM Космические\_объекты

GROUP BY Космические\_объекты.[Расстояние\_от\_Земли св.лет]

HAVING Космические\_объекты.[Расстояние\_от\_Земли св.лет] < @distance

ELSE 'No planets'

END CASE;

CREATE PROCEDURE GetMinOrMaxPlanetSize @size int

AS

SELECT Космические\_объекты.[Размер км], Космические\_объекты.Описание,

CASE

WHEN Космические\_объекты.[Размер км] = min(Космические\_объекты.[Размер км])

THEN 'Найдена минимальный размер планеты'

WHEN Космические\_объекты.[Размер км] = max(Космические\_объекты.[Размер км])

THEN 'Найдена максимальный размер планеты'

END AS texting

FROM Космические\_объекты;

CREATE PROCEDURE EquimentState @stateID smallint

AS

SELECT Оборудование.Описание, Оборудование.Сост\_оборуд\_id, COUNT(Оборудование.Сост\_оборуд\_id)

FROM Оборудование INNER JOIN Сост\_оборуд ON Оборудование.Сост\_оборуд\_id = Сост\_оборуд.Сост\_оборуд\_id

GROUP BY Оборудование.Сост\_оборуд\_id

HAVING COUNT(Оборудование.Сост\_оборуд\_id)>1

CREATE PROCEDURE SortDescriptionEquimentByAlph @type\_sort VARCHAR(15)

AS

SELECT Оборудование.Описание

CASE

WHEN @type\_sort = 'asc'

THEN SELECT Оборудование.Описание FROM Оборудование ORDER BY Оборудование.Описание ASC

WHEN @type\_sort = 'desc'

THEN SELECT Оборудование.Описание FROM Оборудование ORDER BY Оборудование.Описание DESC

END AS texting

FROM Оборудование

CREATE PROCEDURE get\_data\_reporting (d1 DATETIME, d2 DATETIME)

AS

SELECT \*

FROM Отчёты

WHERE Отчёты.Дата\_создания BETWEEN d1 and d2

GROUP BY Отчёты.Дата\_создания

CREATE PROCEDURE CreateReport @id smallint, @dt\_tm datetime

AS

IF EXISTS (SELECT \* FROM Отчёты

WHERE Отчёты.Дата\_создания = @dt\_tm OR Отчёты.Отчёты\_id = @id

)

RETURN 0

INSERT dbo.Отчёты (Отчёты\_id, Дата\_создания)

VALUES(@id, @dt\_tm)

CREATE PROCEDURE UpdateStateEquiment @id smallint, @decsrp varchar(200), @new\_descrp VARCHAR(200)

AS

if EXISTS (SELECT \* FROM Сост\_оборуд

WHERE Сост\_оборуд.Сост\_оборуд\_id != @id

)

RETURN 0

UPDATE dbo.Сост\_оборуд

SET Описание\_состояния = @new\_decsrp

WHERE Описание\_состояния = @decsrp

CREATE PROCEDURE DeleteStateEquiment @id smallint

AS

if EXISTS (SELECT \* FROM Сост\_оборуд

WHERE Сост\_оборуд.Сост\_оборуд\_id != @id

)

RETURN 0

DELETE FROM Сост\_оборуд

WHERE Сост\_оборуд.Сост\_оборуд\_id = @id

CREATE PROCEDURE EmployeeExperience @years int

AS

IF EXISTS (SELECT Сотрудники.id, Сотрудники.ФИО, Сотрудники.Стаж

FROM Сотрудники

WHERE Сотрудники.Стаж != @years

)

RETURN 0

SELECT Сотрудники.Договор\_id, Сотрудники.ФИО,

Сотрудники.Контактный\_номер, Сотрудники.Должность\_id,

Сотрудники.Оборудование\_id

FROM Сотрудники

INNER JOIN Должность ON Должность.Должность\_id = Сотрудники.Должность\_id

INNER JOIN Оборудование ON Оборудование.Оборудование\_id = Сотрудники.Оборудование\_id

GROUP BY Сотрудники.Договор\_id

CREATE PROCEDURE UpdateEmployeePhone @id smallint, @newphone VARCHAR(50)

AS

IF EXISTS(SELECT \* FROM Сотрудники WHERE Сотрудники.id = @id)

RETURN 0

UPDATE Сотрудники

SET Контактный\_номер = @newphone

WHERE id = @id

CREATE PROCEDURE GetUniqueBreakEquiment

AS

SELECT DISTINCT Списанное\_оборуд.Причина\_списания, Списанное\_оборуд.Списанное\_оборуд\_id

FROM Списанное\_оборуд

CREATE PROCEDURE InsertBreakEquiment @id smallint, @reason VARCHAR(50)

AS

IF EXISTS (SELECT \* FROM Списанное\_оборуд WHERE Списанное\_оборуд.Списанное\_оборуд\_id = @id)

RETURN 0

INSERT INTO Списанное\_оборуд(Списанное\_оборуд\_id, Причина\_списания)

VALUES (@id, @reason)

CREATE PROCEDURE UpdateBreakEquiment

@id smallint, @reason VARCHAR(100)

AS

IF not EXISTS (SELECT \* FROM Списанное\_оборуд WHERE Списанное\_оборуд.Списанное\_оборуд\_id = @id)

RETURN 0

UPDATE Списанное\_оборуд

SET Причина\_списания = @reason

WHERE Списанное\_оборуд\_id = @id

CREATE PROCEDURE UpdateSouvenir

@id smallint, @price VARCHAR(100)

AS

IF not EXISTS (SELECT \* FROM Сувениры WHERE Сувениры\_id=@id)

RETURN 0

UPDATE Сувениры

SET Цена = @price

WHERE Сувениры\_id = @id