**Лабораторная работа №17**

**ИСПОЛЬЗОВАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ И ХРАНИМЫХ ПРОЦЕДУР.**

1. Создать функцию, получающую на вход даты отправки и доставки и возвращает разницу между ними в днях, а также слова «OK», «NOTICE», «WARNING», соответственно, если разница в днях составляет менее десяти, от десяти до тридцати и более тридцати дней.

Листинг 1.1 – Создание функции

CREATE FUNCTION READ\_DURATTION\_AND\_STATUS

(

@start\_date DATE,

@finish\_date DATE

)

RETURNS NVARCHAR(150)

WITH SCHEMABINDING

AS

BEGIN

DECLARE @days INT;

DECLARE @msg NVARCHAR(150);

SET @days = DATEDIFF(day, @start\_date, @finish\_date);

SET @msg =

(

CASE

WHEN (@days < 10) THEN 'OK'

WHEN (@days >= 10) AND (@days <= 30) THEN 'NOTICE'

WHEN (@days > 30) THEN 'WARNING'

END

);

RETURN CONCAT(@days, @msg);

END

Листинг 1.2 – Вывод информации

SELECT

o.o\_id,

o.o\_date,

o.o\_date\_finish,

dbo.READ\_DURATTION\_AND\_STATUS(o.o\_date, o.o\_date\_finish) AS rdns

FROM orders o

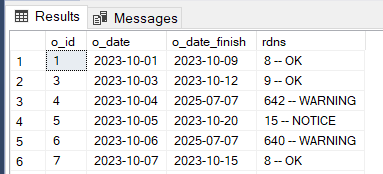


Рис. 1 – Вывод информации

1. Создать функцию, возвращающую список свободных значений автоинкрементируемых первичных ключей в указанной таблице. Свободными считаются значения первичного ключа, которые отсутствуют в таблице, и при этом меньше максимального используемого значения. Например, если в таблице есть первичные ключи 1, 3, 8, то свободными считаются 2, 4, 5, 6, 7.

Листинг 2 – Создание функции и вывод результата

USE SUPIR;

GO

CREATE FUNCTION GET\_FREE\_KEYS\_IN\_CUSTOMERS()

RETURNS VARCHAR(max)

AS

BEGIN

DECLARE @start\_value INT;

DECLARE @stop\_value INT;

DECLARE @free\_keys\_string VARCHAR(max);

--

DECLARE free\_keys\_cursor CURSOR LOCAL FAST\_FORWARD FOR

SELECT start, stop

FROM

(

--

SELECT

min\_t.c\_id + 1 AS start, -- Начало

(

SELECT MIN(c\_id) -1

FROM customers x

WHERE x.c\_id > min\_t.c\_id

) AS stop -- конец

FROM customers min\_t

UNION

--

SELECT

1 AS start, -- начало

(

SELECT MIN(c\_id) - 1

FROM customers x

WHERE c\_id > 0

) AS stop --конец

) AS data

WHERE stop >= start

ORDER BY start, stop;

--

OPEN free\_keys\_cursor;

--

FETCH NEXT FROM free\_keys\_cursor INTO @start\_value, @stop\_value;

--

WHILE @@FETCH\_STATUS = 0

BEGIN

--

WHILE @start\_value <= @stop\_value

BEGIN

--

SET @free\_keys\_string = CONCAT(@free\_keys\_string, @start\_value, ',');

--

SET @start\_value = @start\_value + 1;

END;

--

FETCH NEXT FROM free\_keys\_cursor INTO @start\_value, @stop\_value;

END;

--

CLOSE free\_keys\_cursor;

--

DEALLOCATE free\_keys\_cursor;

--

RETURN LEFT(@free\_keys\_string, LEN(@free\_keys\_string) - 1);

END

GO

SELECT dbo.GET\_FREE\_KEYS\_IN\_CUSTOMERS() AS FREE\_KEYS

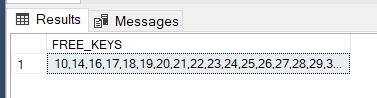


Рис. 2 – Вывод информации

1. Создать функцию, автоматизирующую проверку условий задачи 3 из 16 лабораторной работы. Функция должна возвращать значение 1, если все условия выполнены или -1, -2, -3, если хотя бы одно условие нарушено, при этом модуль числа соответствует номеру условия:
2. Дата начала доставки не может находиться в будущем;
3. Дата конца доставки не может находиться в прошлом;
4. Дата конца доставки не может быть меньше даты начала доставки;

Листинг 3 – Создание функции и вывод результата

USE SUPIR;

GO

CREATE FUNCTION CHECK\_ORDERS\_DATES(

@o\_date DATE,

@o\_date\_finish DATE,

@o\_status CHAR(1))

RETURNS INT

WITH SCHEMABINDING

AS

BEGIN

DECLARE @result INT = 1;

--

IF (@o\_date > CONVERT(date, GETDATE()))

BEGIN

SET @result = -1;

END;

IF ((@o\_date\_finish > CONVERT(date, GETDATE())) AND (@o\_status != 'N'))

BEGIN

SET @result = -2;

END;

IF(@o\_date > @o\_date\_finish)

BEGIN

SET @result = -3;

END;

RETURN @result;

END;

GO

SELECT dbo.CHECK\_ORDERS\_DATES('2025-01-01', '2026-01-01', 'Y');

SELECT dbo.CHECK\_ORDERS\_DATES('2025-01-01', '2026-01-01', 'N');

SELECT dbo.CHECK\_ORDERS\_DATES('2026-01-01', '2027-01-01', 'Y');

SELECT dbo.CHECK\_ORDERS\_DATES('2026-01-01', '2027-01-01', 'N');

SELECT dbo.CHECK\_ORDERS\_DATES('2024-01-01', '2022-01-01', 'Y');

SELECT dbo.CHECK\_ORDERS\_DATES('2024-01-01', '2022-01-01', 'N');

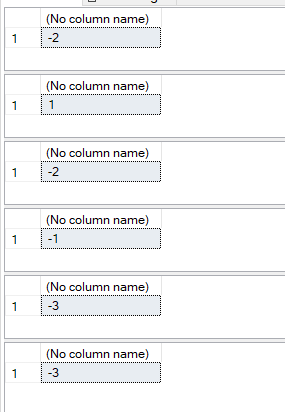


Рис. 3 – Вывод информации

1. Создать хранимую процедуру, формирующую список представлений, триггеров и внешних ключей для указанной таблицы.

Листинг 4 – Создание процедуры и вывод результата

USE SUPIR;

GO

CREATE PROCEDURE SHOW\_TABLE\_OBJECTS

(@table\_name NVARCHAR(150))

WITH EXECUtE AS OWNER

AS

DECLARE @query\_text NVARCHAR(1000) = '';

SET @query\_text =

'

SELECT

''foreign\_key'' AS object\_type,

constraint\_name AS object\_name

FROM information\_schema.table\_constraints

WHERE

table\_catalog = DB\_NAME() AND

table\_name = ''\_FP\_TABLE\_NAME\_PLACEHOLDER\_'' AND

constraint\_type = ''FOREIGN KEY''

UNION

SELECT

''trigger'' AS object\_type,

name AS object\_name

FROM sys.triggers

WHERE OBJECT\_NAME(parent\_id) = ''\_FP\_TABLE\_NAME\_PLACEHOLDER\_''

UNION

SELECT

''view'' AS object\_type,

table\_name AS object\_name

FROM information\_schema.views

WHERE

table\_catalog = DB\_NAME() AND

view\_definition LIKE ''%\_FP\_TABLE\_NAME\_PLACEHOLDER\_%''

';

SET @query\_text = REPLACE(@query\_text, '\_FP\_TABLE\_NAME\_PLACEHOLDER\_', @table\_name);

EXECUTE sp\_executesql @query\_text

GO

EXECUTE SHOW\_TABLE\_OBJECTS 'orders'

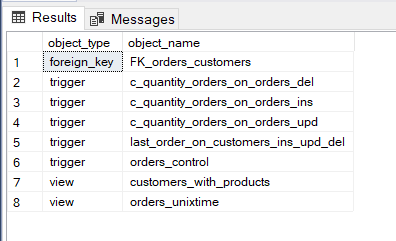


Рис. 4 – Вывод информации

1. Создать хранимую процедуру, автоматически создающую и наполняющую данными агрегирующую таблицу tables\_rc, содержащую информацию о количестве записей во всех таблицах базы данных в формате: имя таблицы, количество записей.

Листинг 5 – Создание процедуры и вывод результата

USE SUPIR;

GO

CREATE PROCEDURE CACHE\_TABELS\_RC

AS

BEGIN

DECLARE @table\_name NVARCHAR(200);

DECLARE @table\_rows INT;

DECLARE @query\_text NVARCHAR(2000);

--

DECLARE tables\_cursor CURSOR LOCAL FAST\_FORWARD FOR

SELECT name

FROM sys.tables;

IF NOT EXISTS

(

SELECT name

FROM sys.tables

WHERE name = 'tables\_rc'

)

BEGIN

CREATE TABLE tables\_rc

(

table\_name VARCHAR(200),

rows\_count INT

);

END;

--

TRUNCATE TABLE tables\_rc;

OPEN tables\_cursor;

FETCH NEXT FROM tables\_cursor INTO @table\_name;

--

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @query\_text = CONCAT('SELECT @cnt = COUNT(1) FROM [', @table\_name, ']');

EXECUTE sp\_executesql @query\_text, N'@cnt INT OUT', @table\_rows OUTPUT;

INSERT INTO tables\_rc (table\_name, rows\_count)

VALUES (@table\_name, @table\_rows);

--

FETCH NEXT FROM tables\_cursor INTO @table\_name;

END;

CLOSE tables\_cursor;

DEALLOCATE Tables\_cursor;

END

GO

EXECUTE CACHE\_TABELS\_RC

SELECT \* FROM tables\_rc

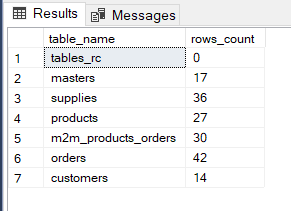


Рис. 5 – Вывод информации