Процедуры

Ситникова Руслана 311 группа

1 Количество книг конкретного издательства

```
---  КОЛИЧЕСТВО КНИГ КОНКРЕТНОГО ИЗДАТЕЛЬСТВА ------------------------------------------------------------------------
GO --------------- Наличие издательства в системе, возвращает id ----------------------
CREATE PROCEDURE Exists_Publishing @Publish_name AS CHAR(50), @id AS INT OUTPUT
AS
IF (SELECT COUNT(*) FROM publishing WHERE publishing_name = @Publish_name) > 0
SET @id = (SELECT publishing_id FROM publishing WHERE publishing_name = @Publish_name)
ELSE
    BEGIN
        SET @id = -1
        PRINT ('Данного издательства в системе нет.')
    END

GO ---------------- Подсчет книг конкретного издательства ----------------------------
CREATE PROCEDURE Count_Books_From_Publishing @Publish_name AS CHAR(50), @amount AS INT OUTPUT
AS
DECLARE @publ_id AS INT
EXEC Exists_Publishing @Publish_name, @publ_id OUTPUT;
IF @publ_id > -1
SET @amount =   (SELECT COUNT(book_edition_id) FROM book_edition
                    INNER JOIN publishing ON book_edition.publishing_id = publishing.publishing_id
                    WHERE publishing_name = @Publish_name)
ELSE SET @amount = -1
-----------------------------------------------------------------------------------------------------------------------
```

Вызов от существующего издательства:

```
DECLARE @out INT
EXEC dbo.Count_Books_From_Publishing 'Эксмо', @out OUTPUT
PRINT @out
```

90 %

Сообщения

```
2
```

Вызов от несуществующего издательства:

```
-----------------------------------------------------------------
DECLARE @out INT
EXEC dbo.Count_Books_From_Publishing 'Азбуккка', @out OUTPUT
PRINT @out
```

100 %

Сообщения

```
Данного издательства в системе нет.
-1
```

2 Количество книг конкретного автора в системе

```
GO ---------------- Подсчет книг конкретного издательства ----------------------------
CREATE PROCEDURE Count_Books_By_Author @Author_name AS CHAR(50), @amount AS INT OUTPUT
AS
DECLARE @auth_id AS INT
BEGIN
    IF (SELECT COUNT(*) FROM author WHERE author_name = @Author_name) > 0
    BEGIN
        SET @amount =   (SELECT COUNT(book_edition_id) FROM book_edition
                            INNER JOIN book ON book_edition.book_id = book.book_id
                            INNER JOIN author ON author.author_id = book.author_id
                            WHERE author_name = @Author_name)
    END
    ELSE
        BEGIN
            PRINT ('Данного автора в системе нет.')
            RETURN -1
        END
END
```

Вызов от существующего автора:

```
DECLARE @out INT
EXEC Count_Books_By_Author 'Булгаков М.А.', @out OUTPUT
PRINT @out
```

100 %

Сообщения

```
2
```

Вызов от несуществующего автора:

```
DECLARE @out INT
EXEC Count_Books_By_Author 'Вася Пупкин', @out OUTPUT
PRINT @out
```

100 %

Сообщения

```
Данного автора в системе нет.


Время выполнения: 2021-11-20T18:46:58.1106747+04:00
```

3 Процедура, получающая подробную таблицу заказов конкретного читателя (используется в другом виде)

```
GO --------------- Вывод заказов конкретного читателя -------------------------
CREATE PROCEDURE Issues_By_Reader @Reader_id AS INT
AS
    SELECT book.title AS 'Название',
           author.author_name AS 'Автор',
           book_edition.book_edition_id AS 'ID издания',
           issued_book.date_of_issue AS 'Дата заказа',
           issued_book.end_of_issue AS 'Дата возврата'
    FROM book INNER JOIN author ON book.author_id = author.author_id
              INNER JOIN book_edition ON book.book_id = book_edition.book_id
              INNER JOIN issued_book ON issued_book.book_edition_id = book_edition.book_edition_id
              INNER JOIN reader ON issued_book.reader_id = reader.reader_id
    WHERE reader.reader_id = @Reader_id
-------------------------------------------------------------------------------
```

Узнаем id читателя Иванов И.И.

```
SELECT reader_id FROM reader WHERE reader_name = 'Иванов И.И.'
```

100 %

Результаты | Сообщения

	reader_id
1	26

Его заказы:

```
EXEC Issues_By_Reader 26
-------------------------------------------------------------------------------
```

100 %

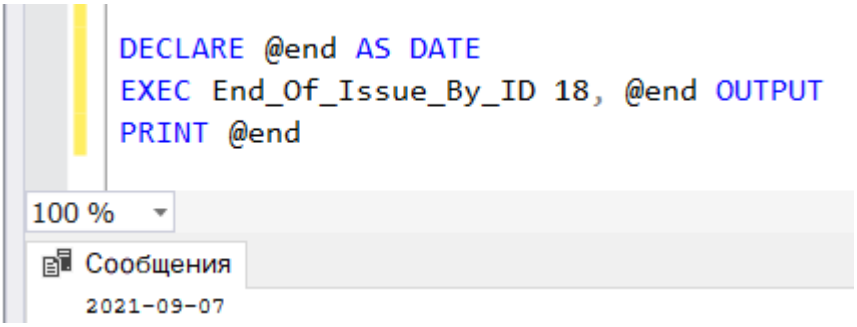
Результаты | Сообщения

	Название	Автор	ID издания	Дата заказа	Дата возврата
1	Мастер и Маргарита	Булгаков М.А.	39	2021-08-23	2021-09-07
2	Стихи детям	Барто А.Л.	41	2021-10-26	2021-11-09

4 Процедура, выводящая дату окончания заказа по ID (не используется)

```
GO --------------- Вывод даты окончания заказа конкретного заказа -------------------------
CREATE PROCEDURE End_Of_Issue_By_ID @Iss_id AS INT, @enddate AS DATE OUTPUT
AS
    SET @enddate = (SELECT issued_book.end_of_issue
                    FROM issued_book
                    WHERE issued_book.issued_book_id = @Iss_id)
```

Вывод:



5 Возвращает 1 если заказ просрочен, иначе 0

```
GO ----------------- Возвращает 1 если заказ просрочен, иначе 0 -----------------------------------
CREATE PROCEDURE Is_Overdue_Issue @Iss_id AS INT, @res AS INT OUTPUT
AS
    IF (SELECT COUNT(expired_book.expired_book_id)
        FROM expired_book WHERE issued_book_id = @Iss_id) > 0
    SET @res = 1
    ELSE SET @res = 0

DROP PROCEDURE Is_Overdue_Issue
```

```
SELECT * FROM issued_book
SELECT * FROM expired_book
```

90 %

Результаты | Сообщения

	issued_book_id	book_id	book_edition_id	amount	reader_id	date_of_issue	end_of_issue
1	18	42	39	15	26	2021-08-23	2021-09-07
2	19	44	40	1	27	2021-09-17	2021-10-01
3	52	45	41	5	26	2021-10-26	2021-11-09

	expired_book_id	issued_book_id	fine_id
1	15	18	1

```
DECLARE @res INT
EXEC Is_Overdue_Issue 18, @res OUTPUT
PRINT @res
```

100 %

Сообщения

```
1

Время выполнения: 2021-11-20T20:39:07.0750785+04:00
```

```
DECLARE @res INT
EXEC Is_Overdue_Issue 52, @res OUTPUT
PRINT @res
```

100 %

Сообщения

```
0

Время выполнения: 2021-11-20T20:40:10.7102961+04:00
```

6 Вывести ID просроченных заказов конкретного читателя

```
GO ---------------- Вывести ID просроченных заказов конкретного читателя -------
CREATE PROCEDURE Overdue_Issues_By_Reader @Reader_id AS INT
AS
    DECLARE @countIssues INT
    SET @countIssues = (SELECT COUNT (issued_book_id)
                        FROM book INNER JOIN author ON book.author_id = author.author_id
                                  INNER JOIN book_edition ON book_edition.book_id = book.book_id
                                  INNER JOIN issued_book ON issued_book.book_edition_id = book_edition.book_edition_id
                                  INNER JOIN reader ON issued_book.reader_id = reader.reader_id
                        WHERE reader.reader_id = @Reader_id)
    DECLARE @i INT
    SET @i = (SELECT MIN(issued_book_id) FROM issued_book)

    DECLARE @cred INT
    SET @cred = 1
    WHILE @countIssues >= @cred
    BEGIN
        IF (SELECT reader_id FROM issued_book WHERE issued_book_id = @i)
            = @Reader_id
            BEGIN
                IF (SELECT COUNT(expired_book_id) FROM expired_book WHERE issued_book_id = @i) > 0
                BEGIN
                    PRINT @i
                END
                SET @cred = @cred + 1
                SET @i = @i + 1
            END
        ELSE
            SET @i = @i + 1
    END
```

Вызов процедуры для Иванова И.И.

```
EXEC Overdue_Issues_By_Reader 26
```

100 %

Сообщения

```
18

Время выполнения: 2021-11-22T19:57:29.2041712+04:00
```

7 Получение информации о заказах, сделанных после конкретной даты, штрафы на которые удовлетворяют статусу

```
GO -------------------- Получение информации о заказах, штрафы на которые удовлетворяют статусу ---------------
CREATE PROCEDURE Info_About_Expired_Fines @afterDate AS DATE, @statusName AS CHAR(20)
AS
    DECLARE @i INT
    DECLARE idExpFine CURSOR FOR
        SELECT issued_book.issued_book_id FROM fine
                INNER JOIN fine_status ON fine.status_id = fine_status.status_id
                INNER JOIN expired_book ON expired_book.fine_id = fine.fine_id
                INNER JOIN issued_book ON issued_book.issued_book_id = expired_book.issued_book_id
        WHERE status_name = @statusName AND issued_book.date_of_issue > @afterDate

        OPEN idExpFine
        FETCH NEXT FROM idExpFine INTO @i
        WHILE @@FETCH_STATUS = 0
        BEGIN
            PRINT @i
            FETCH NEXT FROM idExpFine INTO @i
        END
        CLOSE idExpFine
        DEALLOCATE idExpFine
```

Вывод для некоторых параметров:

```
EXEC Info_About_Expired_Fines '2021-08-01', 'Не оплачен'
```

100 %

Сообщения

```
18

Время выполнения: 2021-11-22T20:39:00.4476612+04:00
```

```
EXEC Info_About_Expired_Fines '2020-08-01', 'Не оплачен'
```

100 %

Сообщения

```
18
74
```

```
EXEC Info_About_Expired_Fines '2020-08-01', 'Просрочен'
```

100 %

Сообщения

```
75

Время выполнения: 2021-11-22T20:54:32.2716074+04:00
```

Что в таблицах:

```
SELECT * FROM fine INNER JOIN fine_status ON fine.status_id = fine_status.status_id
                   INNER JOIN expired_book ON expired_book.fine_id = fine.fine_id
                   INNER JOIN issued_book ON expired_book.issued_book_id = issued_book.issued_book_id
```

100 %

Результаты | Сообщения

	fine_id	date_of_fine	end_of_fine	summ	reader_id	status_id	status_id	status_name	expired_book_id	issued_book_id	fine_id	issued_book_id	book_id	book_edition_id	amount	reader_id	date_of_issue	end_of_issue
1	20	2021-10-27	2021-11-10	10	26	44	44	Не оплачен	16	18	20	18	42	39	15	26	2021-08-23	2021-09-07
2	30	2021-10-27	2021-11-10	10	26	44	44	Не оплачен	17	74	30	74	42	39	15	26	2021-08-15	2021-08-29
3	31	2021-10-27	2021-11-10	12	26	45	45	Просрочен	18	75	31	75	42	39	5	26	2021-08-15	2021-08-29