

SHKOLO

Дигитална трансформация на училищата в България

1. Резултати от създадените функции, приложени върху данните от база данни SHKOLO_DB:

- Функция **TYPE_AVERAGE_GRADE**, която извежда среден успех по вид изпитване в системата;

```
CREATE FUNCTION TYPE_AVERAGE_GRADE(@TYPE_ID INT) RETURNS  
VARCHAR(200)  
AS  
BEGIN  
DECLARE @TYPE_NAME VARCHAR(50), @AVG_GRADE NUMERIC(3,2)  
SELECT @TYPE_NAME = GRADE_TYPE_NAME, @AVG_GRADE =  
AVG(GRADE_VALUE)  
FROM GRADE_TYPE GT JOIN GRADE G  
ON GT.GRADE_TYPE_ID = G.GRADE_TYPE_ID  
WHERE GT.GRADE_TYPE_ID = @TYPE_ID  
GROUP BY GT.GRADE_TYPE_ID, GRADE_TYPE_NAME  
RETURN @TYPE_NAME + ' (ID = ' + CAST(@TYPE_ID AS VARCHAR) + '),  
среден успех: ' + CAST(@AVG_GRADE AS VARCHAR) + '  
END
```

```
SELECT DBO.TYPE_AVERAGE_GRADE(GRADE_TYPE_ID) AS  
TYPE_AVERAGE_GRADE_RESULT  
FROM GRADE_TYPE
```

	TYPE_AVERAGE_GRADE_RESULT
1	Активно участие (ID = 7), среден успех: 6.00.
2	Входно равнище (ID = 1), среден успех: 4.86.
3	Домашна работа (ID = 4), среден успех: 5.80.
4	Класна работа (ID = 5), среден успех: 4.00.
5	Контролна работа (ID = 3), среден успех: 5.13.
6	Проект (ID = 8), среден успех: 5.50.
7	Тест (ID = 2), среден успех: 5.39.
8	Устно изпитване (ID = 6), среден успех: 6.00.

Фигура 1. Резултат от функцията **TYPE_AVERAGE_GRADE**.

- Функция **SUBJECT_AVERAGE_GRADE**, която извежда среден успех по вид предмет в системата;

```
CREATE FUNCTION SUBJECT_AVERAGE_GRADE(@SUBJ_ID INT)  
RETURNS VARCHAR(200)  
AS  
BEGIN
```

```

DECLARE @SUBJ_NAME VARCHAR(50), @AVG_GRADE NUMERIC(3,2)
SELECT @SUBJ_NAME = SUBJECT_NAME, @AVG_GRADE =
AVG(GRADE_VALUE)
FROM GRADE G JOIN [SUBJECT] S
ON G.SUBJECT_ID = S.SUBJECT_ID
WHERE S.SUBJECT_ID = @SUBJ_ID
GROUP BY S.SUBJECT_ID, SUBJECT_NAME
RETURN @SUBJ_NAME + ' (ID = ' + CAST(@SUBJ_ID AS VARCHAR) + '),
среден успех: ' + CAST(@AVG_GRADE AS VARCHAR) + '!'
END

```

```

SELECT DBO.SUBJECT_AVERAGE_GRADE(SUBJECT_ID) AS
SUBJECT_AVERAGE_GRADE_RESULT
FROM [SUBJECT]

```

	SUBJECT_AVERAGE_GRADE_RESULT
1	Английски език (ID = 3), среден успех: 4.80.
2	Биология (ID = 5), среден успех: 5.75.
3	Български език и литература (ID = 2), среден успех: 5.43.
4	География (ID = 9), среден успех: 4.50.
5	Изобразително изкуство (ID = 13), среден успех: 6.00.
6	Информатика (ID = 10), среден успех: 5.50.
7	Информационни технологии (ID = 11), среден успех: 6.00.
8	История (ID = 8), среден успех: 4.67.
9	Математика (ID = 1), среден успех: 5.59.
10	Музика (ID = 12), среден успех: 6.00.
11	Немски език (ID = 4), среден успех: 4.67.
12	ФВС (ID = 14), среден успех: 6.00.
13	Физика (ID = 6), среден успех: 5.00.
14	Химия (ID = 7), среден успех: 5.60.

Фигура 2. Резултат от функцията SUBJECT_AVERAGE_GRADE.

- Функция SCHOOL_GRADE, която извежда среден успех по всички предмети в дадено училище в съответния град;

```

CREATE FUNCTION SCHOOL_GRADE(@SCH_ID INT) RETURNS
VARCHAR(200)
AS
BEGIN
DECLARE @SCH_NAME VARCHAR(50), @SCH_CITY VARCHAR(50),
@AVG_GRADE NUMERIC(3,2)
SELECT @SCH_NAME = SCHOOL_NAME, @SCH_CITY = CITY,
@AVG_GRADE = AVG(GRADE_VALUE)
FROM SCHOOL S JOIN [ADDRESS] A
ON S.ADDRESS_ID = A.ADDRESS_ID
JOIN CLASS C
ON S.SCHOOL_ID = C.SCHOOL_ID
JOIN STUDENT ST

```

```

ON ST.CLASS_ID = C.CLASS_ID
JOIN GRADE G
ON G.STUDENT_ID = ST.STUDENT_ID
WHERE S.SCHOOL_ID = @SCH_ID
GROUP BY S.SCHOOL_ID, SCHOOL_NAME, CITY
RETURN @SCH_NAME + ' (ID = ' + CAST(@SCH_ID AS VARCHAR) + '), град '
+ CAST(@SCH_CITY AS VARCHAR)
+ ', среден успех: ' + CAST(@AVG_GRADE AS VARCHAR) + '
END

```

```

SELECT DBO.SCHOOL_GRADE(SCHOOL_ID) AS SCHOOL_GRADE_RESULT
FROM SCHOOL

```

	SCHOOL_GRADE_RESULT
1	НУ "Христо Ботев" (ID = 3), град Пловдив, среден успех: 5.50.
2	СУ "Паисий Хилендарски" (ID = 1), град София, среден успех: 5.34.
3	ОУ "Алеко Константинов" (ID = 2), град Варна, среден успех: 5.33.

Фигура 3. Резултат от функцията SCHOOL_GRADE.

- **Функция SCHOOL_ABSENCE**, която извежда сума от отсъствията в дадено училище в съответния град;

```

CREATE FUNCTION SCHOOL_ABSENCE(@SCH_ID INT) RETURNS
VARCHAR(200)
AS
BEGIN
DECLARE @SCH_NAME VARCHAR(50), @SCH_CITY VARCHAR(50),
@ABS_AMOUNT NUMERIC(6,2)
SELECT @SCH_NAME = SCHOOL_NAME, @SCH_CITY = CITY,
@ABS_AMOUNT = SUM(ABSENCE_AMOUNT)
FROM SCHOOL S JOIN [ADDRESS] A
ON S.ADDRESS_ID = A.ADDRESS_ID
JOIN CLASS C
ON S.SCHOOL_ID = C.SCHOOL_ID
JOIN STUDENT ST
ON ST.CLASS_ID = C.CLASS_ID
JOIN ABSENCE AB
ON AB.STUDENT_ID = ST.STUDENT_ID
WHERE S.SCHOOL_ID = @SCH_ID
GROUP BY S.SCHOOL_ID, SCHOOL_NAME, CITY
RETURN @SCH_NAME + ' (ID = ' + CAST(@SCH_ID AS VARCHAR) + '), град '
+ CAST(@SCH_CITY AS VARCHAR)
+ ', брой отсъствия: ' + CAST(@ABS_AMOUNT AS VARCHAR) + '
END

```

```
SELECT DBO.SCHOOL_ABSENCE(SCHOOL_ID) AS
SCHOOL_ABSENCE_RESULT
FROM SCHOOL
```

	SCHOOL_ABSENCE_RESULT
1	НУ "Христо Ботев" (ID = 3), град Пловдив, брой отсъствия: 6.00.
2	СУ "Паисий Хилендарски" (ID = 1), град София, брой отсъствия: 46.50.
3	ОУ "Алеко Константинов" (ID = 2), град Варна, брой отсъствия: 6.00.

Фигура 4. Резултат от функцията SCHOOL_ABSENCE.

- Функция TEACHER_SUBJECT_SCHOOL, която извежда имената на преподавателите, техния предмет и даденото училище в съответния град;

```
CREATE FUNCTION TEACHER_SUBJECT_SCHOOL() RETURNS TABLE
AS
RETURN
SELECT FIRST_NAME, LAST_NAME, SUBJECT_NAME, SCHOOL_NAME,
CITY
FROM TEACHER T JOIN [SUBJECT] S
ON T.SUBJECT_ID = S.SUBJECT_ID
JOIN SCHOOL SC
ON SC.SCHOOL_ID = T.SCHOOL_ID
JOIN [ADDRESS] A
ON A.ADDRESS_ID = SC.ADDRESS_ID

SELECT *
FROM DBO.TEACHER_SUBJECT_SCHOOL() ORDER BY FIRST_NAME,
LAST_NAME
```

	FIRST_NAME	LAST_NAME	SUBJECT_NAME	SCHOOL_NAME	CITY
1	Валентин	Георгиев	ФВБ	СУ "Паисий Хилендарски"	София
2	Валентина	Иванова	Английски език	СУ "Паисий Хилендарски"	София
3	Георги	Константинов	Математика	СУ "Паисий Хилендарски"	София
4	Гери	Петрова	Български език и литература	НУ "Христо Ботев"	Пловдив
5	Ивана	Иванова	Математика	СУ "Паисий Хилендарски"	София
6	Иванка	Иванова	Български език и литература	ОУ "Алеко Константинов"	Варна
7	Калина	Тодорова	Математика	СУ "Паисий Хилендарски"	София
8	Калоян	Христов	История	СУ "Паисий Хилендарски"	София
9	Кристина	Русева	Физика	СУ "Паисий Хилендарски"	София
10	Маги	Петрова	Математика	ОУ "Алеко Константинов"	Варна
11	Маргарита	Попова	Математика	СУ "Паисий Хилендарски"	София
12	Марина	Русева	География	СУ "Паисий Хилендарски"	София
13	Маринка	Иванова	Математика	НУ "Христо Ботев"	Пловдив
14	Маринка	Тодорова	Английски език	СУ "Паисий Хилендарски"	София
15	Мартин	Костадинов	Английски език	ОУ "Алеко Константинов"	Варна
16	Мартина	Мартинова	Немски език	СУ "Паисий Хилендарски"	София
17	Мими	Тодорова	Български език и литература	СУ "Паисий Хилендарски"	София
18	Петя	Иванова	Информатика	СУ "Паисий Хилендарски"	София
19	Пламена	Попова	Изобразително изкуство	СУ "Паисий Хилендарски"	София
20	Росен	Петков	Български език и литература	СУ "Паисий Хилендарски"	София
21	Сашо	Костадинов	Английски език	НУ "Христо Ботев"	Пловдив
22	Светла	Колева	Български език и литература	СУ "Паисий Хилендарски"	София
23	Симеон	Симеонов	Химия	СУ "Паисий Хилендарски"	София
24	Стоян	Константинов	Музика	СУ "Паисий Хилендарски"	София
25	Таня	Томова	Английски език	СУ "Паисий Хилендарски"	София
26	Теодор	Иванов	Английски език	СУ "Паисий Хилендарски"	София
27	Теодор	Миленов	Информационни технологии	СУ "Паисий Хилендарски"	София
28	Христо	Тодоров	Биология	СУ "Паисий Хилендарски"	София

Фигура 5. Резултат от функцията TEACHER_SUBJECT_SCHOOL.

2. Резултати от създадените съхранени процедури, приложени върху данните от база данни SHKOLO_DB:

- Съхранена процедура CLASS_STUDENTS, която извежда броя на учениците в класовете на дадено училище в съответния град;

```
CREATE PROCEDURE CLASS_STUDENTS
AS
SELECT CLASS_NUMBER, CLASS_LETTER, SCHOOL_NAME, CITY,
COUNT(STUDENT_ID) AS STUDENT_COUNT
FROM STUDENT ST JOIN CLASS C
ON ST.CLASS_ID = C.CLASS_ID
JOIN SCHOOL SC
ON SC.SCHOOL_ID = C.SCHOOL_ID
JOIN ADDRESS A
ON A.ADDRESS_ID = SC.ADDRESS_ID
GROUP BY CLASS_NUMBER, CLASS_LETTER, SCHOOL_NAME, CITY

EXEC CLASS_STUDENTS
```

	CLASS_NUMBER	CLASS_LETTER	SCHOOL_NAME	CITY	STUDENT_COUNT
1	1	A	НУ "Христо Ботев"	Пловдив	5
2	2	A	НУ "Христо Ботев"	Пловдив	5
3	5	A	ОУ "Алеко Константинов"	Варна	5
4	5	A	СУ "Паисий Хилендарски"	София	10
5	5	Б	ОУ "Алеко Константинов"	Варна	5
6	5	В	ОУ "Алеко Константинов"	Варна	5
7	6	A	СУ "Паисий Хилендарски"	София	10
8	7	A	СУ "Паисий Хилендарски"	София	10
9	8	A	СУ "Паисий Хилендарски"	София	8
10	8	Б	СУ "Паисий Хилендарски"	София	8
11	9	A	СУ "Паисий Хилендарски"	София	8
12	9	Б	СУ "Паисий Хилендарски"	София	8
13	10	A	СУ "Паисий Хилендарски"	София	7
14	10	Б	СУ "Паисий Хилендарски"	София	7
15	11	A	СУ "Паисий Хилендарски"	София	7
16	11	Б	СУ "Паисий Хилендарски"	София	7
17	12	A	СУ "Паисий Хилендарски"	София	6
18	12	Б	СУ "Паисий Хилендарски"	София	6

Фигура 6. Резултат от съхранената процедура CLASS_STUDENTS.

- Съхранена процедура GET_SUBJECT_TOTAL, която връща като параметри броя нанесени оценки и среден успех по даден предмет в системата с идентификатор, подаден като входен параметър;

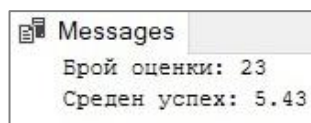
```
CREATE PROCEDURE GET_SUBJECT_TOTAL @SUBJ_ID INT,
@GRADE_COUNT INT OUTPUT, @GRADE_AVERAGE NUMERIC(3,2)
OUTPUT
AS
SELECT @GRADE_COUNT = COUNT(GRADE_ID)
```

```

FROM GRADE G JOIN SUBJECT S
ON G.SUBJECT_ID = S.SUBJECT_ID
WHERE S.SUBJECT_ID = @SUBJ_ID
SELECT @GRADE_AVERAGE = AVG(GRADE_VALUE)
FROM GRADE G JOIN SUBJECT S
ON G.SUBJECT_ID = S.SUBJECT_ID
WHERE S.SUBJECT_ID = @SUBJ_ID

DECLARE @GR_COUNT INT, @GR_AVERAGE NUMERIC(3,2)
EXEC GET_SUBJECT_TOTAL 1, @GR_COUNT OUTPUT, @GR_AVERAGE
OUTPUT
PRINT 'Брой оценки: ' + CAST(@GR_COUNT AS VARCHAR)
PRINT 'Среден успех: ' + CAST(@GR_AVERAGE AS VARCHAR)

```



Фигура 7. Резултат от съхранената процедура GET_SUBJECT_TOTAL, SUBJ_ID = 1 (Математика).

- Съхранена процедура TEACHERS_SCHOOL, която извежда броя на преподавателите в дадено училище в съответния град;

```

CREATE PROCEDURE TEACHERS_SCHOOL
AS
SELECT SCHOOL_NAME, CITY, COUNT(TEACHER_ID) AS
TEACHER_COUNT
FROM TEACHER T JOIN SCHOOL SC
ON T.SCHOOL_ID = SC.SCHOOL_ID
JOIN [ADDRESS] A
ON A.ADDRESS_ID = SC.ADDRESS_ID
GROUP BY SCHOOL_NAME, CITY

EXEC TEACHERS_SCHOOL

```

	SCHOOL_NAME	CITY	TEACHER_COUNT
1	ОУ "Алеко Константинов"	Варна	3
2	НУ "Христо Ботев"	Пловдив	3
3	СУ "Паисий Хилендарски"	София	22

Фигура 8. Резултат от съхранената процедура TEACHERS_SCHOOL.

- Съхранена процедура CLASSES_LEAD_TEACHER, която извежда класовете с техните класни ръководители в дадено училище в съответния град;


```

CREATE PROCEDURE CLASSES_LEAD_TEACHER
AS
SELECT CLASS_NUMBER, CLASS_LETTER, FIRST_NAME, LAST_NAME,
SCHOOL_NAME, CITY
FROM CLASS C JOIN SCHOOL SC
ON C.SCHOOL_ID = SC.SCHOOL_ID
JOIN TEACHER T
ON T.TEACHER_ID = C.TEACHER_ID
JOIN [ADDRESS] A
ON A.ADDRESS_ID = SC.ADDRESS_ID

EXEC CLASSES_LEAD_TEACHER

```

	CLASS_NUMBER	CLASS_LETTER	FIRST_NAME	LAST_NAME	SCHOOL_NAME	CITY
1	5	A	Ивана	Иванова	СУ "Паисий Хилендарски"	София
2	6	A	Валентина	Иванова	СУ "Паисий Хилендарски"	София
3	7	A	Калоян	Христов	СУ "Паисий Хилендарски"	София
4	8	A	Мargarита	Попова	СУ "Паисий Хилендарски"	София
5	8	Б	Пламена	Попова	СУ "Паисий Хилендарски"	София
6	9	A	Светла	Колева	СУ "Паисий Хилендарски"	София
7	9	Б	Валентин	Георгиев	СУ "Паисий Хилендарски"	София
8	10	A	Кристина	Русева	СУ "Паисий Хилендарски"	София
9	10	Б	Марина	Русева	СУ "Паисий Хилендарски"	София
10	11	A	Теодор	Иванов	СУ "Паисий Хилендарски"	София
11	11	Б	Христо	Тодоров	СУ "Паисий Хилендарски"	София
12	12	A	Мартина	Мартинова	СУ "Паисий Хилендарски"	София
13	12	Б	Росен	Петков	СУ "Паисий Хилендарски"	София
14	5	A	Маги	Петрова	ОУ "Алеко Константинов"	Варна
15	5	Б	Иванка	Иванова	ОУ "Алеко Константинов"	Варна
16	5	В	Мартин	Костадинов	ОУ "Алеко Константинов"	Варна
17	1	A	Маринка	Иванова	НУ "Христо Ботев"	Пловдив
18	2	A	Гери	Петрова	НУ "Христо Ботев"	Пловдив

Фигура 9. Резултат от съхранената процедура CLASSES_LEAD_TEACHER.

3. Резултати от създадените тригери, приложени върху данните от база данни SHKOLO_DB:

- Тригер TRG_TEACHER_HIST и таблица TEACHER_HISTORY.

```

CREATE TABLE TEACHER_HISTORY
(
    CHANGE_ID INT IDENTITY(1,1) PRIMARY KEY,
    TEACHER_ID INT NOT NULL,
    FIRST_NAME VARCHAR(50) NOT NULL,
    MIDDLE_NAME VARCHAR(50) NOT NULL,
    LAST_NAME VARCHAR(50) NOT NULL,
    GENDER CHAR(1) NOT NULL CHECK (GENDER IN ('F', 'M')),
    PHONE VARCHAR(10),
    EMAIL VARCHAR(50),

```

```
        UPDATED_ON DATETIME NOT NULL,  
        OPERATION CHAR(3) NOT NULL,  
        CHECK(OPERATION = 'INS' OR OPERATION = 'DEL')  
    )
```

```
CREATE TRIGGER TRG_TEACHER_HIST  
ON TEACHER  
AFTER INSERT, DELETE  
AS  
BEGIN  
    SET NOCOUNT ON  
    INSERT INTO TEACHER_HISTORY  
    (  
        TEACHER_ID,  
        FIRST_NAME,  
        MIDDLE_NAME,  
        LAST_NAME,  
        GENDER,  
        PHONE,  
        EMAIL,  
        UPDATED_ON,  
        OPERATION  
    )  
    SELECT  
        I.TEACHER_ID,  
        FIRST_NAME,  
        MIDDLE_NAME,  
        LAST_NAME,  
        GENDER,  
        PHONE,  
        I.EMAIL,  
        GETDATE(),  
        'INS'  
    FROM  
        INSERTED I  
    UNION ALL  
    SELECT  
        D.TEACHER_ID,  
        FIRST_NAME,  
        MIDDLE_NAME,  
        LAST_NAME,  
        GENDER,  
        PHONE,  
        D.EMAIL,  
        GETDATE(),  
        'DEL'
```



```

FROM
  DELETED D
END

```

```

SELECT * FROM TEACHER_HISTORY

```

	CHANGE_ID	TEACHER_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	GENDER	PHONE	EMAIL	UPDATED_ON	OPERATION
1	1	31	Кали	Христова	Тодорова	F	NULL	kali_todorova@gmail.com	2022-10-27 23:26:19.170	INS
2	2	32	Маринка	Петкова	Тодорова	F	0833227488	marinka_todorova@gmail.com	2022-10-27 23:26:19.170	INS
3	3	31	Кали	Христова	Тодорова	F	NULL	kali_todorova@gmail.com	2022-10-27 23:28:15.507	DEL
4	4	33	Таня	Петрова	Томова	F	NULL	NULL	2022-10-27 23:28:36.917	INS
5	5	34	Маринка	Петканова	Иванова	F	0882223144	marinka_ivanova@gmail.com	2022-11-11 17:09:38.970	INS
6	6	35	Гери	Константинова	Петрова	F	0899625678	g_petrova@gmail.com	2022-11-11 17:09:38.970	INS
7	7	36	Сашо	Миленов	Костадинов	M	0811996745	sasho_kostadinov@gmail.com	2022-11-11 17:09:38.970	INS

Фигура 10. Данни в таблицата TEACHER_HISTORY.