**ПРОЕКТ ПО БАЗИ ДАННИ**

на тема:

**Електронен дневник**

**Съдържание**

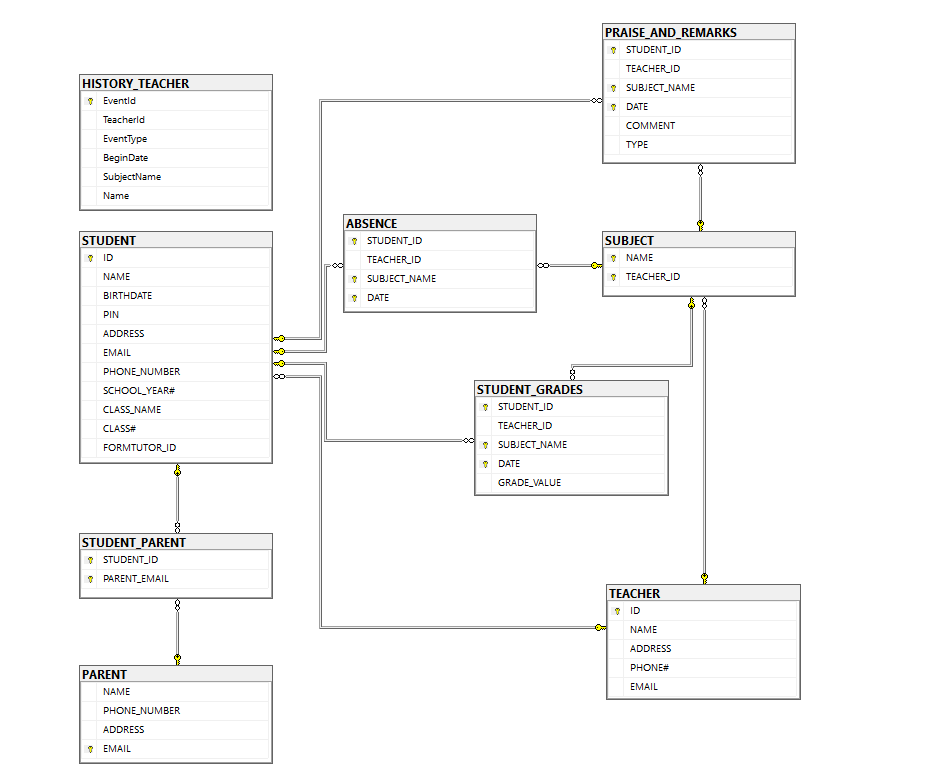
1. Описание на предметната област и на заданието
2. Схема на релациите
3. Примерно съдържание
4. Примерни заявки с резултата им разделени на:
   1. Прости заявки
   2. Заявки върху две и повече релации
   3. Подзаявки
   4. Съединения
   5. Групиране и агрегация
5. Заключения върху:
   1. Подобрения с добавяне на ограничения на ниво таблица
   2. Доразвиване на системата с добавяне на изгледи – индекси и тригери

1. **Описание на предметната област и на заданието**

Базата данни за електронен дневник е предназначена за използване от учебните заведения. Съдържа информация за резултатите от текущия контрол на учениците, получени по време на учебната година.

В дневника е поместена, както информация за оценките, заедно с личните данни на ученик, така и лични данни на преподавателите и родителите на учениците. Съдържа се информация за конкретен учебен предмет и за набора от оценки на съответен ученик. Поместени са, също така, данни за отсъствията по дата на даден ученик и такива за похвалите и забележките, които е получил.

1. **Схема на релациите**

****

Таблицата **STUDENT** съдържа информация за конкретен ученик. Първичният ключ е атрибутът *ID*.

* ID - пореден номер на ученика;
* NAME - две имена;
* BIRTHDATE - рождена дата;
* PIN - ЕГН;
* ADDRESS - адрес на местоживеене;
* EMAIL - личен e-mail на ученика;
* PHONE\_NUMBER - телефонен номер на ученика;
* SCHOOL\_YEAR# - номер на випуска, към който принадлежи ученикът;
* CLASS\_NAME - име на паралелката;
* CLASS# - номер на ученика в класа;
* FORMTUTOR\_ID - идентификационен номер на класния ръководител; външен ключ към TEACHER.ID;

Таблицата **PARENT** съдържа информация за съответен родител на ученик. Първичният ключ е атрибутът *EMAIL*.

* NAME - две имена на родител;
* PHONE\_NUMBER - телефонен номер на родител;
* ADDRESS - адрес на родител;
* EMAIL - e-mail на родител;

Таблицата **STUDENT\_PARENT** е помощна таблица, обуславяща връзката между таблиците **STUDENT** и **PARENT**. Двата й атрибута *STUDENT\_ID* и *PARENT\_EMAIL* заедноформират първичния ключ.

* STUDENT\_ID - идентификационен номер на даден ученик; външен ключ към STUDENT.ID;
* PARENT\_EMAIL - имейл на родител; външен ключ към PARENT.EMAIL;

Таблицата **TEACHER** съдържа информация за конкретен преподавател. Първичният ключ е атрибутът *ID*.

* ID - пореден номер на преподавателя;
* NAME - две имена;
* ADDRESS - адрес на местоживеене;
* PHONE# - личен телефонен номер;
* EMAIL - личен e-mail на преподавателя;

Таблицата **SUBJECT** съдържа информация за учебните предмети, преподавани в конкретното училище. Първичният ключ е формиран заедно от атрибутите *NAME* и *TEACHER\_ID*.

* NAME - име на учебния предмет;
* TEACHER\_ID - идентификационен номер на преподавателя, преподаващ съответния предмет; външен ключ към TEACHER.ID;

Таблицата **ABSENCE** съдържа информация за отсъствията на конкретен ученик. Първичният и ключ се формира заедно от атрибутите *STUDENT\_ID*, *SUBJECT\_NAME* и *DATE*.

* STUDENT\_ID - идентификационен номер на даден ученик; външен ключ към STUDENT.ID;
* TEACHER\_ID - идентификационен номер на даден преподавател;
* SUBJECT\_NAME - наименование на учебен предмет;
* DATE - дата на отсъствие;

Таблицата **STUDENT\_GRADES** съдържа информация за оценките на конкретен ученик. Първичният й ключ се формира заедно от атрибутите *STUDENT\_ID*, *SUBJECT\_NAME* и *DATE*.

* STUDENT\_ID - идентификационен номер на даден ученик; външен ключ към STUDENT.ID;
* TEACHER\_ID - идентификационен номер на даден преподавател;
* SUBJECT\_NAME - наименование на учебен предмет;
* DATE - дата на получаване на оценката;
* GRADE\_VALUE - стойност на оценката;

Таблицата **PRAISE\_AND\_REMARKS** съдържа информация за забележките и похвалите на конкретен ученик. Първичният и ключ се формира заедно от атрибутите *STUDENT\_ID*, *SUBJECT\_NAME* и *DATE*.

* STUDENT\_ID - идентификационен номер на даден ученик; външен ключ към STUDENT.ID;
* TEACHER\_ID - идентификационен номер на даден преподавател;
* SUBJECT\_NAME - наименование на учебен предмет;
* DATE - дата на получаване на забележката или похвалата;
* COMMENT - текст с контекста на забележката или похвалата;
* TYPE - вид, ‘p’ за похвала, ‘r’ за забележка;

Таблицата **HISTORY\_TEACHER** е помощна таблица, влизаща в употреба при добавянето на тригери. Първичният й ключ е атрибутът *EventId*.

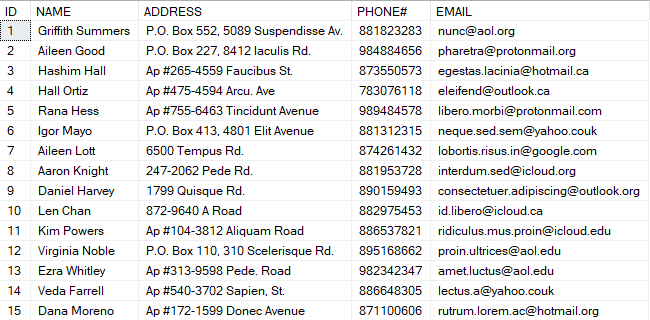
* EventId - пореден номер на събитието;
* TeacherId - идентификационен номер на учителят;
* EventType - вид на събитието, ‘update’, ’insert’, ‘delete’ ;
* BeginDate - начална дата;
* SubjectName - име на учебен предмет;
* Name - име на преподавател;

1. **Примерно съдържание**

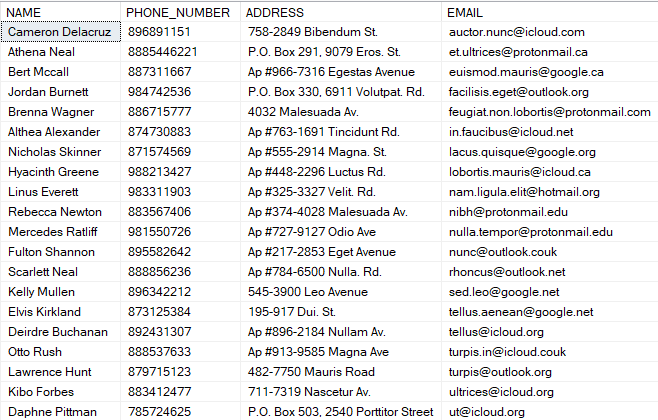
* Съдържание на таблицата **STUDENT**



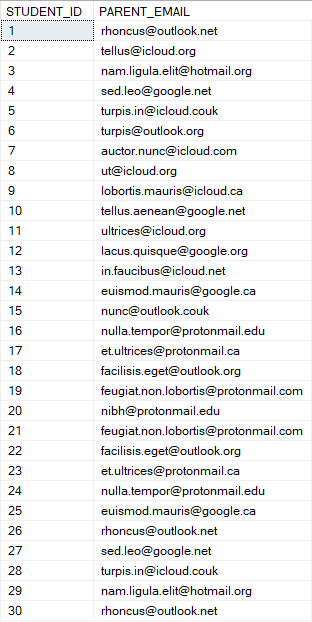
* Съдържание на таблицата **TEACHER**



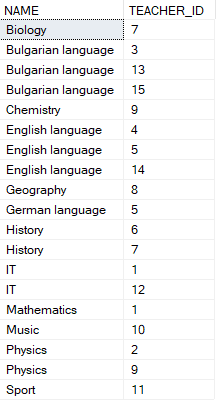
* Съдържание на таблицата **PARENT**



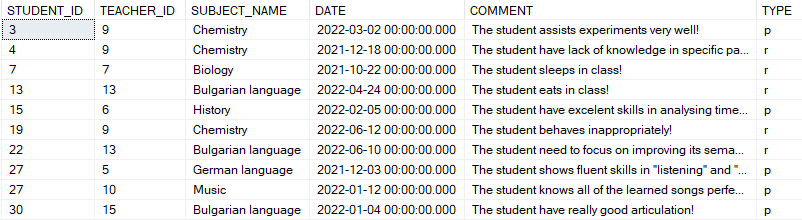
* Съдържание на таблицата **STUDENT\_PARENT**

****

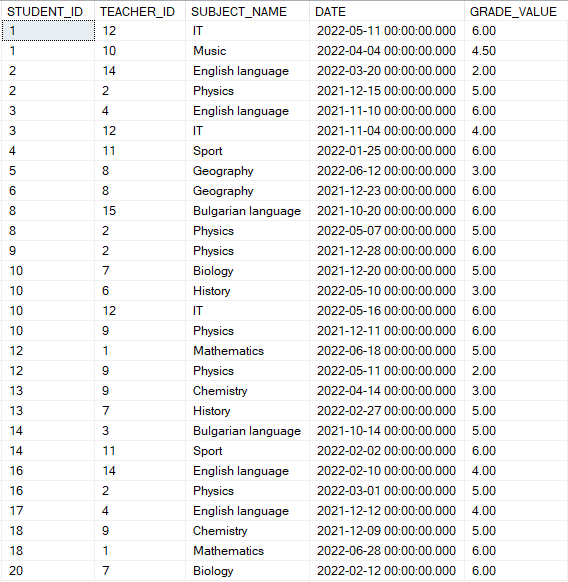
* Съдържание на таблицата **SUBJECT**

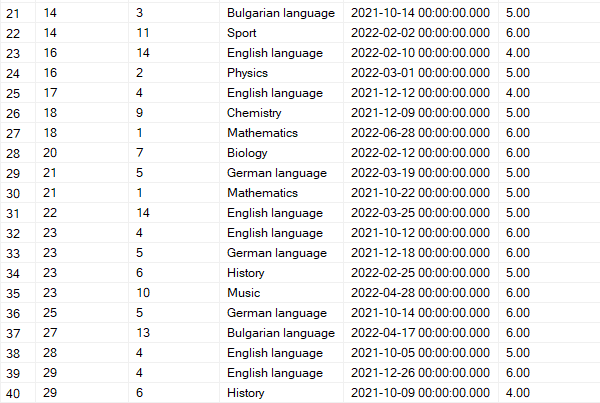
****

* Съдържание на таблицата **PRAISE\_AND\_REMARKS**

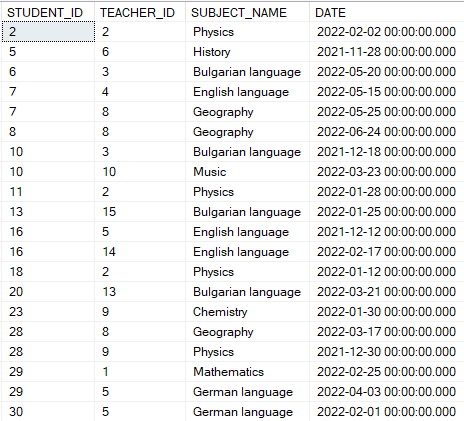
****

* Съдържание на таблицата **STUDENT\_GRADES**

****

****

* Съдържание на таблицата **ABSENCE**

****

1. **Примерни заявки с резултата им**
   1. **Прости заявки**

**-- 1. Заявка, която извежда имената на всички ученици, които са от ‚b‘ клас и ги сортира наобратно.**

SELECT s.NAME

FROM STUDENT s

WHERE CLASS\_NAME = 'b'

ORDER BY s.NAME DESC



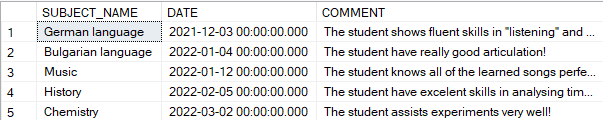
**-- 2. Заявка, която извежда името на предмета, датата и коментара на всички похвали и ги сортира по дата.**

SELECT p.SUBJECT\_NAME, p.DATE, p.COMMENT

FROM PRAISE\_AND\_REMARKS p

WHERE p.TYPE = 'p'

ORDER BY p.DATE



**-- 3. Заявка, която извежда данните на преподавателя с име Hall Ortiz.**

SELECT \*

FROM TEACHER t

WHERE t.NAME = 'Hall Ortiz'



**-- 4. Заявка, която извежда номера и адреса на родител с име Daphne Pittman.**

SELECT p.PHONE\_NUMBER, p.ADDRESS

FROM PARENT p

WHERE p.NAME = 'Daphne Pittman'



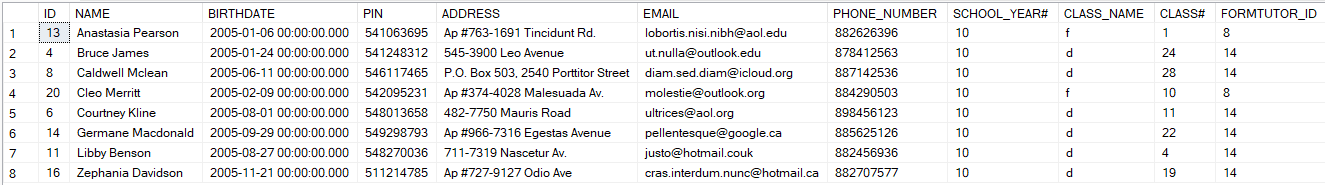
**-- 5. Заявка, която извежда данните на всички ученици от 10 клас и ги сортира по име.**

SELECT \*

FROM STUDENT s

WHERE s.SCHOOL\_YEAR# = 10

ORDER BY s.NAME



* 1. **Заявки върху две и повече релации**

**-- 1. Заявка, която извежда името на класния ръководител на Vivien Crane.**

SELECT t.NAME

FROM STUDENT s, TEACHER t

WHERE s.NAME = 'Vivien Crane' AND s.FORMTUTOR\_ID = t.ID



**-- 2. Заявка, която извежда имената на родителите, клас и имена на учениците от всички класове без 12-ти, чието име на ученик съдържа ‘а’ и ги сортира първо по име на родител и след това по име на ученик.**

SELECT p.NAME Parent\_Name, s.NAME Student\_Name, s.SCHOOL\_YEAR#

FROM STUDENT s, PARENT p, STUDENT\_PARENT sp

WHERE UPPER(s.NAME) LIKE UPPER('%a%') AND s.ID = sp.STUDENT\_ID AND p.EMAIL = sp.PARENT\_EMAIL

EXCEPT

SELECT p.NAME Parent\_Name, s.NAME Student\_Name, s.SCHOOL\_YEAR#

FROM STUDENT s, PARENT p, STUDENT\_PARENT sp

WHERE UPPER(s.NAME) LIKE UPPER('%a%') AND s.SCHOOL\_YEAR# = 12 AND s.ID = sp.STUDENT\_ID AND p.EMAIL = sp.PARENT\_EMAIL

ORDER BY p.NAME, s.NAME



**-- 3. Заявка, която извежда имената на родителите, имената на учениците и клас на всички ученици от 11 и 12 клас и ги сортира по клас, име на ученик, име на родител.**

SELECT p.NAME Parent\_Name, s.NAME Student\_Name, s.SCHOOL\_YEAR#

FROM STUDENT s, PARENT p, STUDENT\_PARENT sp

WHERE s.SCHOOL\_YEAR# = 11 AND s.ID = sp.STUDENT\_ID AND p.EMAIL = sp.PARENT\_EMAIL

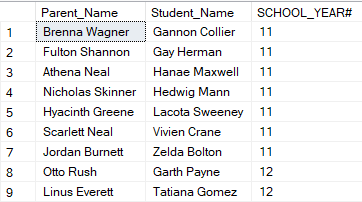
UNION

SELECT p.NAME Parent\_Name, s.NAME Student\_Name , s.SCHOOL\_YEAR#

FROM STUDENT s, PARENT p, STUDENT\_PARENT sp

WHERE s.SCHOOL\_YEAR# = 12 AND s.ID = sp.STUDENT\_ID AND p.EMAIL = sp.PARENT\_EMAIL

ORDER BY s.SCHOOL\_YEAR#, s.NAME, p.NAME



**-- 4. Заявка, която извежда имената на класните ръководители на учениците от 10 клас и предметите, по които те преподават без повторения.**

SELECT DISTINCT t.NAME, sb.NAME

FROM STUDENT s, TEACHER t, SUBJECT sb

WHERE s.SCHOOL\_YEAR# = 10 AND s.FORMTUTOR\_ID = t.ID AND t.ID = sb.TEACHER\_ID

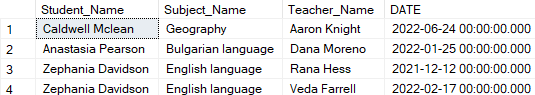


**-- 5. Заявка, която извежда имената на учениците, имената на предметите, имената на преподавателите и датата на отсъствията на учениците от 10 клас, чието име съдържа ‘а’.**

SELECT s.NAME Student\_Name, a.SUBJECT\_NAME Subject\_Name, t.NAME Teacher\_Name, a.DATE

FROM STUDENT s, TEACHER t, SUBJECT sb, ABSENCE a

WHERE s.SCHOOL\_YEAR# = 10 AND UPPER(s.NAME) LIKE UPPER('%a%') AND s.ID = a.STUDENT\_ID AND a.SUBJECT\_NAME = sb.NAME AND a.TEACHER\_ID = sb.TEACHER\_ID AND sb.TEACHER\_ID = t.ID



* 1. **Подзаявки**

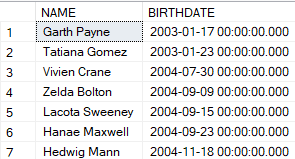
**-- 1. Заявка, която извежда имената и дата на раждане на всички ученици, родени по-рано от Gannon Collier и ги сортира по дата.**

SELECT s.NAME, s.BIRTHDATE

FROM STUDENT s

WHERE s.BIRTHDATE < (SELECT s.BIRTHDATE FROM STUDENT s WHERE s.NAME = 'Gannon Collier')

ORDER BY BIRTHDATE



**-- 2. Заявка, която извежда данните за най-малкия ученик.**

SELECT \*

FROM STUDENT s

WHERE s.BIRTHDATE <= ALL(SELECT BIRTHDATE FROM STUDENT)



**-- 3. Заявка, която извежда името рождената дата и имейла на учениците, получили забележка след датата на забележката на Anastasia Pearson.**

SELECT s.NAME, s.BIRTHDATE, s.EMAIL

FROM STUDENT s, PRAISE\_AND\_REMARKS pr

WHERE s.ID = pr.STUDENT\_ID AND pr.DATE > (SELECT p.DATE FROM STUDENT s1, PRAISE\_AND\_REMARKS p WHERE s1.ID = p.STUDENT\_ID AND p.TYPE = 'r' AND s1.NAME = 'Anastasia Pearson')

****

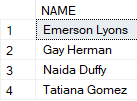
**-- 4. Заявка, която извежда имената на учениците, които имат поне една похвала и ги сортира**

SELECT DISTINCT s.NAME

FROM STUDENT s

WHERE s.ID IN (SELECT p.STUDENT\_ID FROM PRAISE\_AND\_REMARKS p WHERE p.TYPE = 'p')

ORDER BY s.NAME



**-- 5. Заявка, която извежда имената на класните ръководители на ученици, които имат поне една 6-ца без повторения**

SELECT DISTINCT t.NAME

FROM TEACHER t, (select FORMTUTOR\_ID FROM STUDENT s, STUDENT\_GRADES g WHERE s.ID = g.STUDENT\_ID AND g.GRADE\_VALUE = 6) st

WHERE t.ID = st.FORMTUTOR\_ID



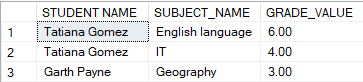
* 1. **Съединения**

**--1. Заявка, която извежда имената и оценките по съответен предмет на всички ученици от 12A клас.**

SELECT STUDENT.NAME AS 'STUDENT NAME', STUDENT\_GRADES.SUBJECT\_NAME,STUDENT\_GRADES.GRADE\_VALUE

FROM STUDENT JOIN STUDENT\_GRADES ON STUDENT.ID = STUDENT\_GRADES.STUDENT\_ID

AND STUDENT.SCHOOL\_YEAR# = 12 AND STUDENT.CLASS\_NAME = 'a';

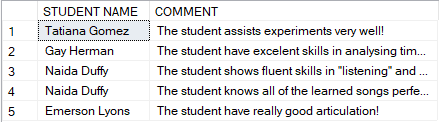


**--2. Заявка, която извежда имената на учениците с похвала**

SELECT STUDENT.NAME AS 'STUDENT NAME', PRAISE\_AND\_REMARKS.COMMENT

FROM STUDENT JOIN PRAISE\_AND\_REMARKS ON STUDENT.ID = PRAISE\_AND\_REMARKS.STUDENT\_ID

AND PRAISE\_AND\_REMARKS.TYPE = 'p';

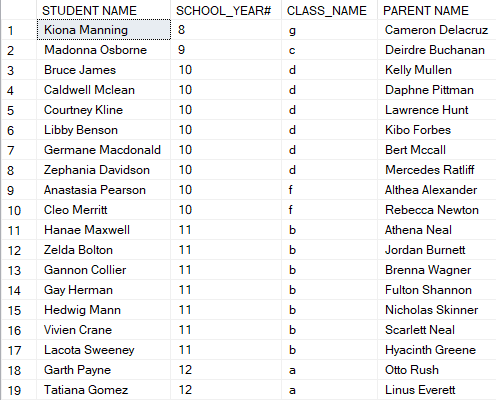


**--3. Заявка, която извежда имената и класа на учениците в гимназиален етап на обучение, както и имената на родителите им.**

SELECT STUDENT.NAME AS 'STUDENT NAME', STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME, PARENT.NAME AS 'PARENT NAME'

FROM STUDENT JOIN STUDENT\_PARENT ON STUDENT.ID = STUDENT\_PARENT.STUDENT\_ID AND STUDENT.SCHOOL\_YEAR# > 7 JOIN PARENT ON STUDENT\_PARENT.PARENT\_EMAIL = PARENT.EMAIL

ORDER BY STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME;



**--4. Заявка, която извежда имената на учениците, изучаващи съответен предмет и неговото наименование.**

SELECT DISTINCT STUDENT.NAME AS 'STUDENT NAME', SUBJECT.NAME AS 'SUBJECT'

FROM STUDENT JOIN TEACHER ON STUDENT.FORMTUTOR\_ID = TEACHER.ID JOIN SUBJECT ON TEACHER.ID = SUBJECT.TEACHER\_ID

UNION

SELECT DISTINCT STUDENT.NAME AS 'STUDENT NAME', STUDENT\_GRADES.SUBJECT\_NAME

FROM STUDENT JOIN STUDENT\_GRADES ON STUDENT.ID = STUDENT\_GRADES.STUDENT\_ID

UNION

SELECT DISTINCT STUDENT.NAME AS 'STUDENT NAME', ABSENCE.SUBJECT\_NAME

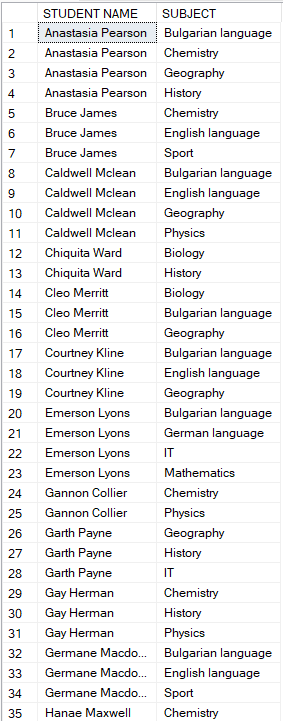
FROM STUDENT JOIN ABSENCE ON STUDENT.ID = ABSENCE.STUDENT\_ID

UNION

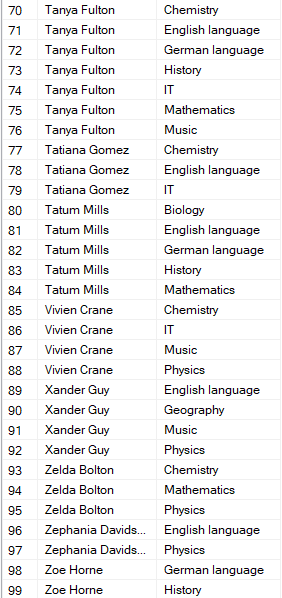
SELECT DISTINCT STUDENT.NAME AS 'STUDENT NAME', PRAISE\_AND\_REMARKS.SUBJECT\_NAME

FROM STUDENT JOIN PRAISE\_AND\_REMARKS ON STUDENT.ID = PRAISE\_AND\_REMARKS.STUDENT\_ID

ORDER BY STUDENT.NAME;





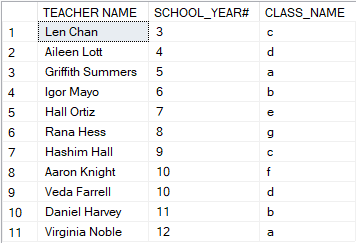


**--5. Заявка, която извежда имената на учителите и съответната паралелка, на която са класни ръководители в случай, че са такива.**

SELECT DISTINCT TEACHER.NAME AS 'TEACHER NAME', STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME

FROM TEACHER JOIN STUDENT ON STUDENT.FORMTUTOR\_ID = TEACHER.ID

ORDER BY SCHOOL\_YEAR#;



* 1. **Групиране и агрегация**

**--1. Заявка, която извежда имената на родителите с повече от едно деца, посещаващи училището и броят на децата им.**

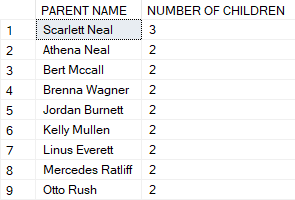
SELECT PARENT.NAME AS 'PARENT NAME', COUNT(STUDENT.ID) AS 'NUMBER OF CHILDREN'

FROM PARENT JOIN STUDENT\_PARENT ON PARENT.EMAIL = STUDENT\_PARENT.PARENT\_EMAIL JOIN STUDENT ON STUDENT\_PARENT.STUDENT\_ID = STUDENT.ID

GROUP BY PARENT.NAME

HAVING COUNT(STUDENT.ID) > 1

ORDER BY COUNT(STUDENT.ID) DESC;

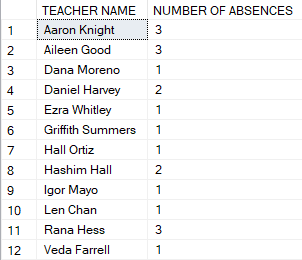


**--2. Заявка, която извежда броя на отсъствията, които даден преподавател е написал в случай, че е написал.**

SELECT TEACHER.NAME AS 'TEACHER NAME', COUNT(\*) AS 'NUMBER OF ABSENCES'

FROM TEACHER JOIN ABSENCE ON TEACHER.ID = ABSENCE.TEACHER\_ID

GROUP BY TEACHER.NAME;



**--3. Заявка, която извежда името на класа с най-много ученици и класния му ръководител, както и броя на учениците в него.**

SELECT STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME, TEACHER.NAME 'TEACHER NAME', COUNT(STUDENT.ID) AS 'NUMBER OF STUDENTS'

FROM STUDENT JOIN TEACHER ON STUDENT.FORMTUTOR\_ID = TEACHER.ID

GROUP BY TEACHER.NAME, STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME

HAVING COUNT(STUDENT.ID) >= ALL(SELECT COUNT(STUDENT.ID) FROM STUDENT,

TEACHER WHERE STUDENT.FORMTUTOR\_ID = TEACHER.ID

GROUP BY TEACHER.NAME, STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME);



**--4. Заявка, която извежда името на най-малкия и най-големия ученик в училището.**

SELECT STUDENT.NAME AS 'STUDENT NAME', STUDENT.BIRTHDATE

FROM STUDENT

WHERE STUDENT.BIRTHDATE = (SELECT MIN(STUDENT.BIRTHDATE) FROM STUDENT)

UNION

SELECT STUDENT.NAME AS 'STUDENT NAME', STUDENT.BIRTHDATE

FROM STUDENT

WHERE STUDENT.BIRTHDATE = (SELECT MAX(STUDENT.BIRTHDATE) FROM STUDENT);



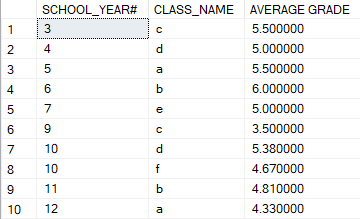
**--5. Заявка, която извежда средния успех на учениците за всеки клас.**

SELECT STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME, ROUND(AVG(STUDENT\_GRADES.GRADE\_VALUE), 2) AS 'AVERAGE GRADE'

FROM STUDENT JOIN STUDENT\_GRADES ON STUDENT.ID = STUDENT\_GRADES.STUDENT\_ID

GROUP BY STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME

ORDER BY STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME;



**--6. Заявка, която извежда името на преподавателя, написал най-много 6-ци, както и на този, написал най-много слаби оценки.**

SELECT TEACHER.NAME AS 'TEACHER\_NAME', STUDENT\_GRADES.GRADE\_VALUE

FROM TEACHER JOIN STUDENT\_GRADES ON TEACHER.ID = STUDENT\_GRADES.TEACHER\_ID AND STUDENT\_GRADES.GRADE\_VALUE > 5.5

GROUP BY TEACHER.NAME, STUDENT\_GRADES.GRADE\_VALUE

HAVING COUNT(STUDENT\_GRADES.GRADE\_VALUE) >= ALL(SELECT COUNT(STUDENT\_GRADES.GRADE\_VALUE)FROM TEACHER

JOIN STUDENT\_GRADES ON TEACHER.ID = STUDENT\_GRADES.TEACHER\_ID AND STUDENT\_GRADES.GRADE\_VALUE > 5.5

GROUP BY TEACHER.NAME, STUDENT\_GRADES.GRADE\_VALUE)

UNION

SELECT TEACHER.NAME AS 'TEACHER\_NAME', STUDENT\_GRADES.GRADE\_VALUE

FROM TEACHER JOIN STUDENT\_GRADES ON TEACHER.ID = STUDENT\_GRADES.TEACHER\_ID AND STUDENT\_GRADES.GRADE\_VALUE < 3

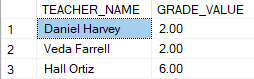
GROUP BY TEACHER.NAME, STUDENT\_GRADES.GRADE\_VALUE

HAVING COUNT(STUDENT\_GRADES.GRADE\_VALUE) >= ALL(SELECT COUNT(STUDENT\_GRADES.GRADE\_VALUE)FROM TEACHER

JOIN STUDENT\_GRADES ON TEACHER.ID = STUDENT\_GRADES.TEACHER\_ID AND STUDENT\_GRADES.GRADE\_VALUE < 3

GROUP BY TEACHER.NAME, STUDENT\_GRADES.GRADE\_VALUE)

ORDER BY STUDENT\_GRADES.GRADE\_VALUE;



**--7. Заявка, която извежда името на най-строгия учител, базирайки се на броя написани забележки.**

SELECT TEACHER.NAME AS 'TEACHER\_NAME', COUNT(PRAISE\_AND\_REMARKS.COMMENT) AS 'NUMBER OF REMARKS'

FROM PRAISE\_AND\_REMARKS, TEACHER

WHERE PRAISE\_AND\_REMARKS.TEACHER\_ID = TEACHER.ID AND PRAISE\_AND\_REMARKS.TYPE = 'r'

GROUP BY TEACHER.NAME

HAVING COUNT(PRAISE\_AND\_REMARKS.COMMENT) >= ALL (SELECT COUNT(PRAISE\_AND\_REMARKS.COMMENT)

FROM PRAISE\_AND\_REMARKS, TEACHER

WHERE PRAISE\_AND\_REMARKS.TEACHER\_ID = TEACHER.ID AND PRAISE\_AND\_REMARKS.TYPE = 'r'

GROUP BY TEACHER.NAME);



**--8. Заявка, която извежда броят на учителите, преподаващи едни и същи предмети.**

SELECT COUNT(DISTINCT s1.TEACHER\_ID) AS 'NUMBER OF TEACHERS TEACHING SAME SUBJECTS'

FROM SUBJECT s1, SUBJECT s2, TEACHER t1, TEACHER t2

WHERE s1.TEACHER\_ID = t1.ID AND s2.TEACHER\_ID = t2.ID AND s1.NAME = s2.NAME AND t1.ID != t2.ID;



**--9. Заявка, която извежда имената на учениците с най-висок успех.**

SELECT STUDENT.NAME AS 'STUDENT\_NAME', STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME, AVG(STUDENT\_GRADES.GRADE\_VALUE) AS 'VALUE'

FROM STUDENT JOIN STUDENT\_GRADES ON STUDENT.ID = STUDENT\_GRADES.STUDENT\_ID

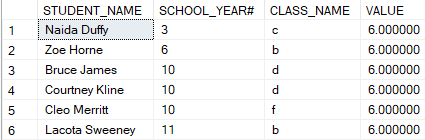
GROUP BY STUDENT.NAME, STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME

HAVING AVG(STUDENT\_GRADES.GRADE\_VALUE) >= ALL(SELECT AVG(STUDENT\_GRADES.GRADE\_VALUE)

FROM STUDENT JOIN STUDENT\_GRADES ON STUDENT.ID = STUDENT\_GRADES.STUDENT\_ID

GROUP BY STUDENT.NAME, STUDENT.SCHOOL\_YEAR#, STUDENT.CLASS\_NAME)

ORDER BY SCHOOL\_YEAR#, CLASS\_NAME, STUDENT\_NAME;



**--10. Заявка, която извежда името на ученика с най-много отсъствия.**

SELECT STUDENT.NAME AS 'STUDENT\_NAME', COUNT(ABSENCE.DATE) AS 'NUMBER OF ABSENCES'

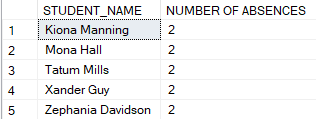
FROM STUDENT JOIN ABSENCE ON STUDENT.ID = ABSENCE.STUDENT\_ID

GROUP BY STUDENT.NAME

HAVING COUNT(ABSENCE.DATE) >= ALL(SELECT COUNT(ABSENCE.DATE)

FROM STUDENT JOIN ABSENCE ON STUDENT.ID = ABSENCE.STUDENT\_ID

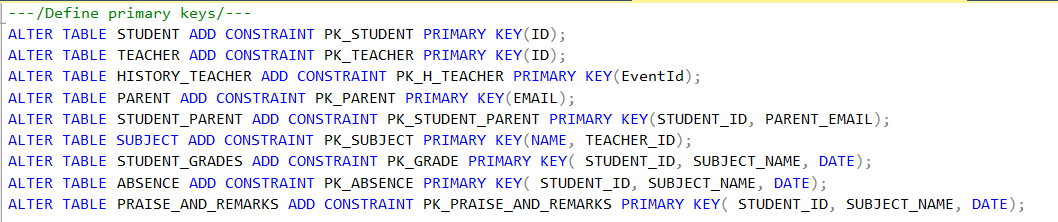
GROUP BY STUDENT.NAME);



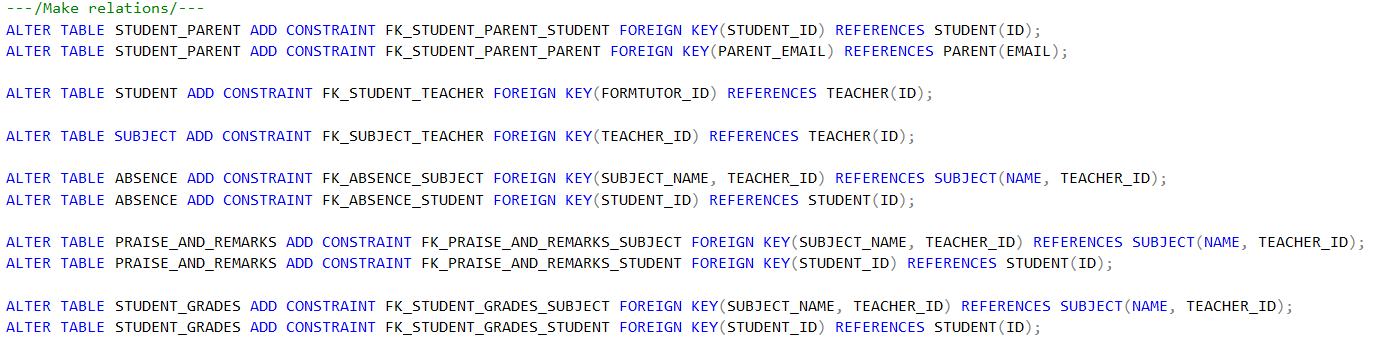
1. **Заключения върху:**
   1. **Подобрения с добавяне на ограничения на ниво таблица**

Добавили сме следните ограничения, съответно:

* за дефиниране на първичните ключове на таблиците



* установяване на връзките между таблиците



* 1. **Доразвиване на системата с добавяне на:**
     1. **Изгледи**

**--Изглед, представящ имената на учениците, заедно с имената на родителите им и адреса, на който живеят**

GO

CREATE VIEW student\_parent\_view

AS

SELECT s.name as student\_name, p.name as parent\_name, p.address

FROM student s inner JOIN STUDENT\_PARENT sp

ON s.id = sp.STUDENT\_ID

inner join parent p on p.email = sp.PARENT\_EMAIL;

GO



**--Изглед, представящ имената на учениците и съответен предмет, който изучават**

GO

CREATE VIEW student\_subject\_view

AS

SELECT DISTINCT STUDENT.NAME AS 'STUDENT\_NAME', SUBJECT.NAME AS 'SUBJECT'

FROM STUDENT JOIN TEACHER ON STUDENT.FORMTUTOR\_ID = TEACHER.ID

JOIN SUBJECT ON TEACHER.ID = SUBJECT.TEACHER\_ID

UNION

SELECT DISTINCT STUDENT.NAME AS 'STUDENT\_NAME', STUDENT\_GRADES.SUBJECT\_NAME AS 'SUBJECT'

FROM STUDENT JOIN STUDENT\_GRADES ON STUDENT.ID = STUDENT\_GRADES.STUDENT\_ID

UNION

SELECT DISTINCT STUDENT.NAME AS 'STUDENT\_NAME', ABSENCE.SUBJECT\_NAME AS 'SUBJECT'

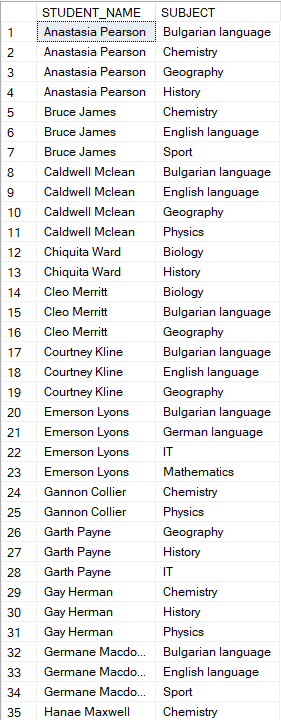
FROM STUDENT JOIN ABSENCE ON STUDENT.ID = ABSENCE.STUDENT\_ID

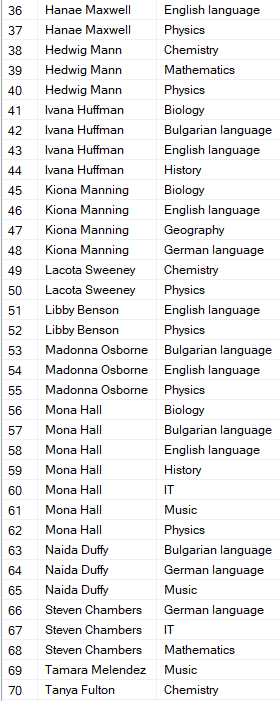
UNION

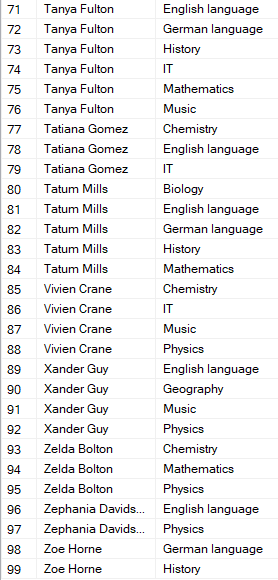
SELECT DISTINCT STUDENT.NAME AS 'STUDENT\_NAME', PRAISE\_AND\_REMARKS.SUBJECT\_NAME AS 'SUBJECT'

FROM STUDENT JOIN PRAISE\_AND\_REMARKS ON STUDENT.ID = PRAISE\_AND\_REMARKS.STUDENT\_ID

GO







**--Изглед, представящ имената на учениците и съответният им класен ръководител**

GO

CREATE VIEW student\_formtutor\_view

AS

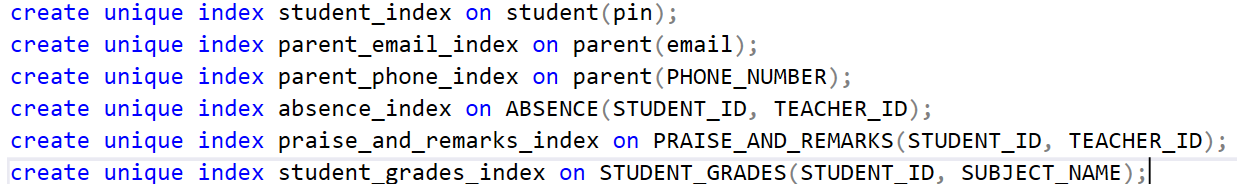
SELECT STUDENT.NAME as student\_name, TEACHER.NAME as teacher\_name

FROM STUDENT JOIN TEACHER ON STUDENT.FORMTUTOR\_ID = TEACHER.ID

GO

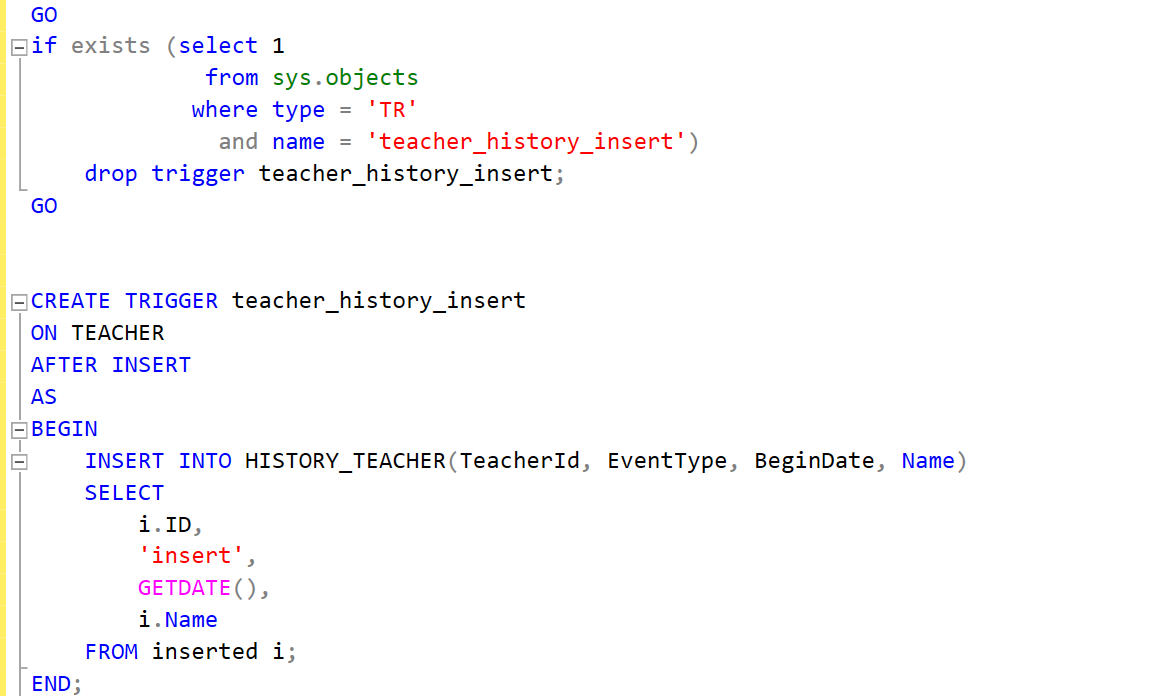


* + 1. **Индекси**

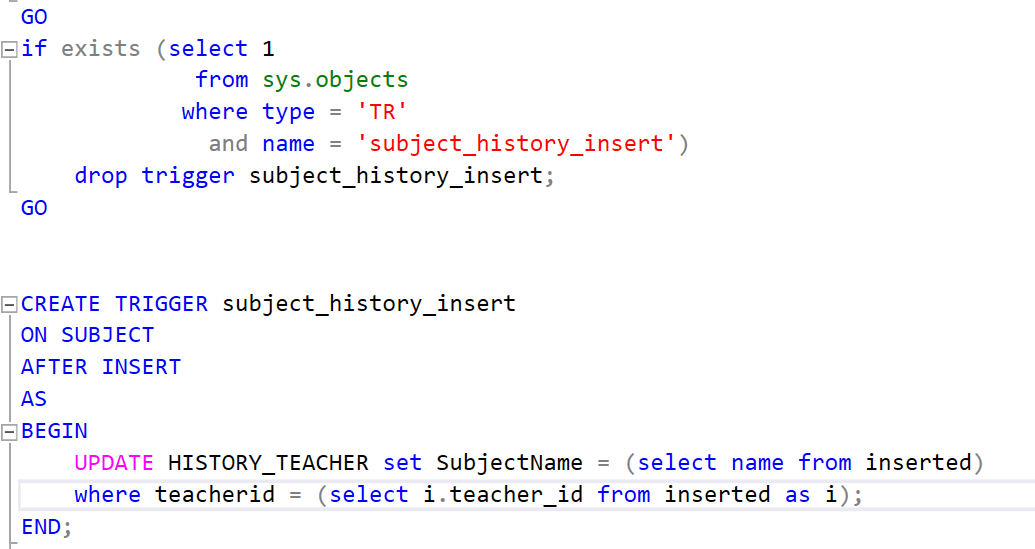


* + 1. **Тригери**

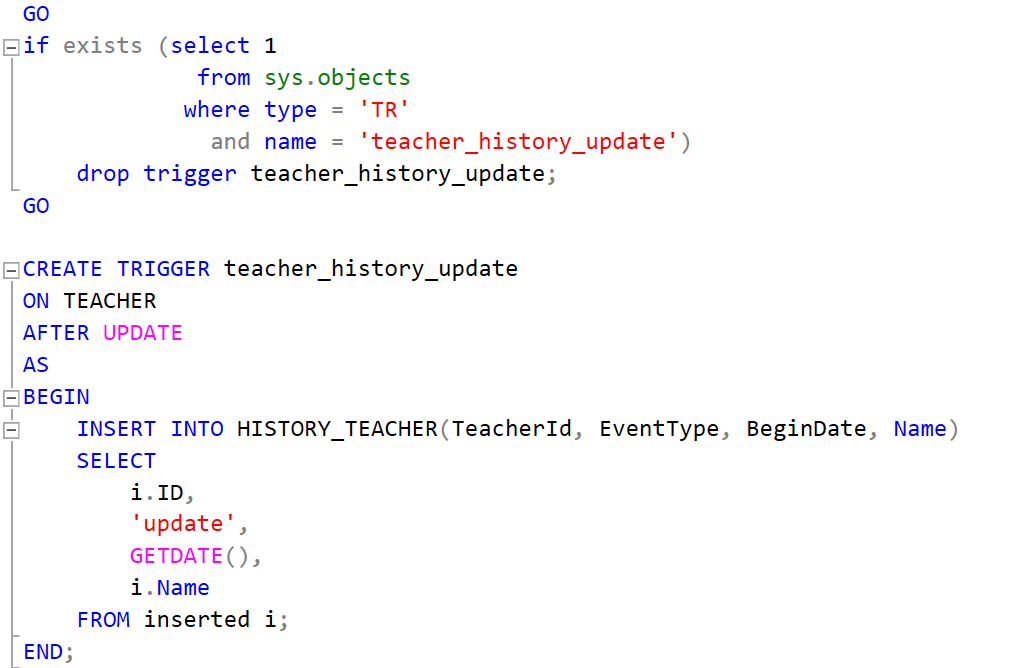
**--Тригeр за добавяне на нов преподавател**



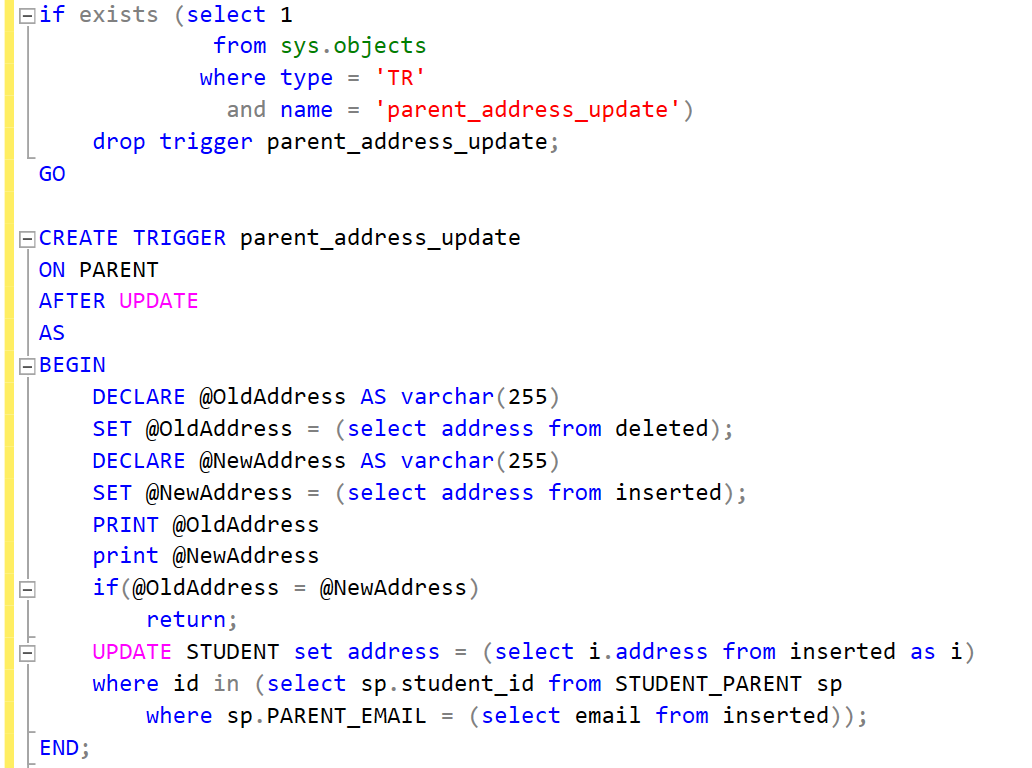
**--Тригер за добавяне на нов учебен предмет**



**–Тригeр за обновяване на данни за конкретен преподавател**



**--Тригeр за промяна на адрес на даден родител**



1. **Оценка на реализацията, препоръки и възможни подобрения**

* Като оценка на реализацията можем да отбележим, че базата ни е ефективна, задоволява нуждите на заведения с насоченост в сферата на образованието, предоставя възможности за бързо и лесно модифициране и е построена спрямо нужните изисквания.
* Възможно подобрение на системата е справянето с проблема премахване на редове от таблиците, което само по себе си води до преправяне на цялата структурна логика на релациите между отделните таблици, тъй като крайната зависимост на всички едни от други на този етап възпрепятства премахването на даден ключов компонент.
* Като допълнителни подобрения можем да разширим идейната цел на базата и да включим нови графи с ‘Извънкласни дейности’, също и да бъдат посочени различни роли на преподавателите, могат да се добавят възможности за съхраняване на предадени от учениците материали, като по този начин се следи тяхната активност и учебно разписание за всеки от отделните класове.