Transaction.

Here can possibly occur R-W problem, e.g. during the function evaluation one thread can try to write already deleted teacher.

```
CREATE FUNCTION lessonSubstitution(studentId VARCHAR(8), prevTeacherId
VARCHAR(8), newTeacherId VARCHAR(8), level CHAR(2), language VARCHAR(10))
RETURNS BOOLEAN
AS $$
    BEGIN
        IF ((newTeacherId IS NONE) OR (newTeacherId NOT IN (SELECT
langSchoolCardNumber FROM Teacher)) OR (newTeacherId = prevTeacherId)
            (language NOT IN ('English', 'German', 'French')) OR (level NOT IN
('A1', 'A2', 'B1', 'B2', 'C1', 'C2')))
            THEN RETURN FALSE;
        ELSE
            UPDATE Lesson
            SET teacherId = newTeacherId
            WHERE (Lesson.studentid = studentId) AND (Lesson.teacherId =
prevTeacherId) AND (Lesson.level = level) AND (Lesson.language = language);
            RETURN TRUE;
        END IF;
    END;
$$
LANGUAGE plpgsql;

BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT lessonSubstitution('34105645', '19221167', '02678669', 'A2', 'French')
COMMIT TRANSACTION;
```

View.

Virtual table with data of students who pay tuition fees in cash.

```
CREATE VIEW cashPayers AS
SELECT *
FROM Person
WHERE
Person.langSchoolCardNumber in  (
    SELECT Payment.studentid
    FROM Payment
    WHERE (Payment.method = 'cash')
);
```

Trigger.

This trigger controls that person cannot be a teacher and a student at the same time.

```
CREATE FUNCTION check_teachers_and_students()
RETURNS TRIGGER
AS $$
    BEGIN
    IF (NEW.langschoolcardnumber IN (SELECT langschoolcardnumber FROM Student))
THEN
        RAISE EXCEPTION 'Student cannot be teacher';
    END IF;
    RETURN NEW;
    END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER teacher_tg_SchoolCardNum BEFORE INSERT OR UPDATE ON teacher
FOR EACH ROW EXECUTE PROCEDURE check_teachers_and_students();
```

Index.

Due to index usage, this query got a 4x speed-up.

```
CREATE INDEX lesson_level_idx ON Lesson(level);
CREATE INDEX lesson_language_idx ON Lesson(language);

SELECT Person.name, Person.street, Person.email
FROM Person
WHERE
 Person.langSchoolCardNumber in (
 SELECT Student.langSchoolCardNumber
 FROM Student
 INNER JOIN Lesson ON (Student.langSchoolCardNumber = Lesson.studentId)
 WHERE (Lesson.level = 'B1') AND (Lesson.language = 'French')
);
```