Домашнее задание №7
Тельнов Сергей, М3435

Введем для сокращения общую view

```
create view v_students_cources as
  (select students.id as student_id,
          students.name as student_name,
          academic_plan.course_id as course_id
   from students
   inner join academic_plan
      on students.group_id = academic_plan.group_id
  );
```

1. Напишите запрос, удаляющий всех студентов, не имеющих долгов.

```
delete
from students
where students.id not in
    (select distinct student_courses.student_id
     from v_students_cources as student_courses
     inner join marks
      on student_courses.student_id = marks.student_id
        and student_courses.course_id = marks.course_id
     where marks.value < 60);
```

2. Напишите запрос, удаляющий всех студентов, имеющих 3 и более долгов.

```
with debts as
  (select student_courses.student_id as id,
          count(*) as debt_number
   from v_students_cources as student_courses
   inner join marks
    on student_courses.student_id = marks.student_id
      and student_courses.course_id = marks.course_id
   where marks.value < 60
   group by id)
delete
from students
where students.id in
    (select id
     from debts
     where debts.debt_number >= 3);
```

3. Напишите запрос, удаляющий все группы, в которых нет студентов.

```
delete
from groups
where id in
    (select groups.id
     from groups
     left join students on groups.id = students.group_id
     where students.id is null);
```

4. Создайте view `Losers` в котором для каждого студента, имеющего долги указано их количество.

```
create view Losers as
  (select student_courses.student_id as id,
          student_courses.student_name as name,
          count(*) as debt_number
   from v_students_cources as student_courses
   inner join marks
    on student_courses.student_id = marks.student_id
      and student_courses.course_id = marks.course_id
   where marks.value < 60
   group by id, name);
```

5. Создайте таблицу `LoserT`, в которой содержится та же информация, что во view `Losers`. Эта таблица должна автоматически обновляться при изменении таблицы с баллами.

```
create table LoserT as select * from Losers;

create function update_losers_func()
  returns trigger as
  $$
    begin
      if (TG_OP = 'DELETE') then
        delete
        from LoserT
        where id = OLD.student_id;

        insert into LoserT
        (select *
        from Losers as v
        where v.id = OLD.student_id);
        return OLD;
      elsif (TG_OP = 'INSERT' or TG_OP = 'UPDATE') then
        delete
        from LoserT as t
        where t.id = NEW.student_id;

        insert into LoserT
        (select *
        from Losers as v
        where v.id = NEW.student_id);
        return NEW;
      end if;
    end;
  $$ language plpgsql;

create trigger update_loser_trigger
after insert or update or delete on marks
for each row execute procedure update_losers_func();
```

6. Отключите автоматическое обновление таблицы `LoserT`.
   ```
   drop trigger update_loser_trigger on marks;
   ```

7. Напишите запрос (один), которой обновляет таблицу `LoserT`, используя данные из таблицы `NewPoints`, в которой содержится информация о баллах, проставленных за последний день.
8. Добавьте проверку того, что все студенты одной группы изучают один и тот же набор курсов.
9. Создайте триггер, не позволяющий уменьшить баллы студента по предмету. При попытке такого изменения баллы изменяться не должны.

```
create function decrease_value_check()
  returns trigger as
  $$
    begin
      if NEW.value < OLD.value then
        RAISE EXCEPTION 'cannot decrease mark value';
      end if;
      return NEW;
    end;
  $$ language plpgsql;

create trigger decrease_value_trigger
before update on marks
for each row execute procedure decrease_value_check();
```