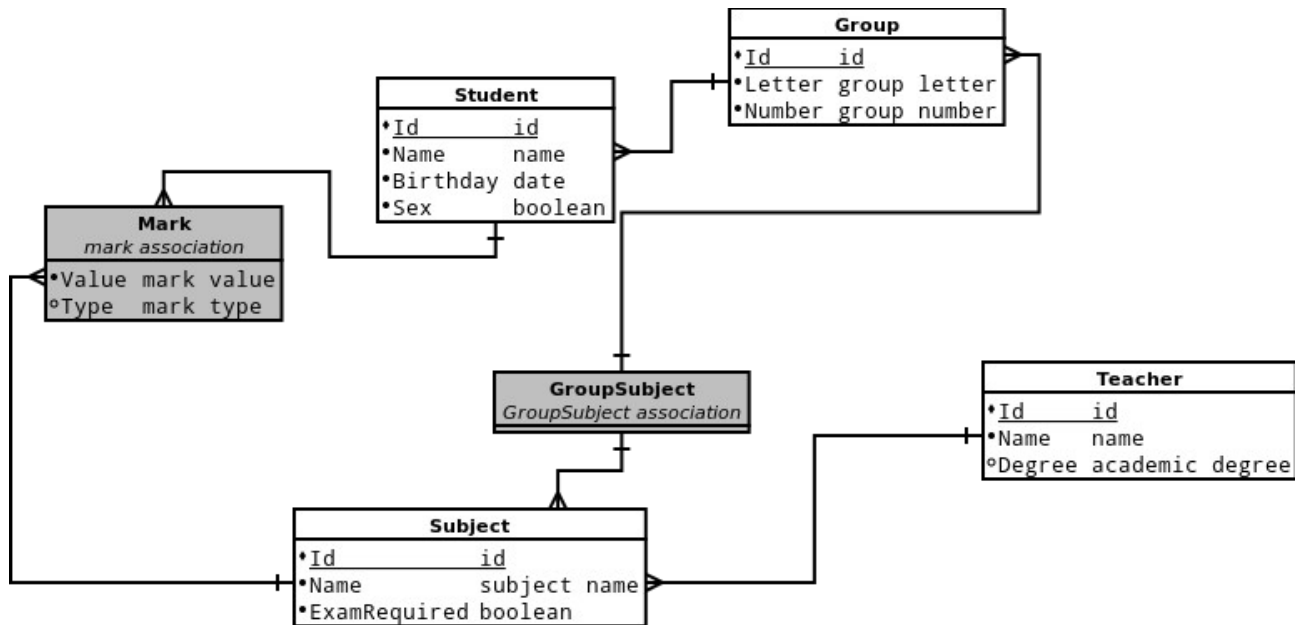


Базы данных. Домашнее задание №2

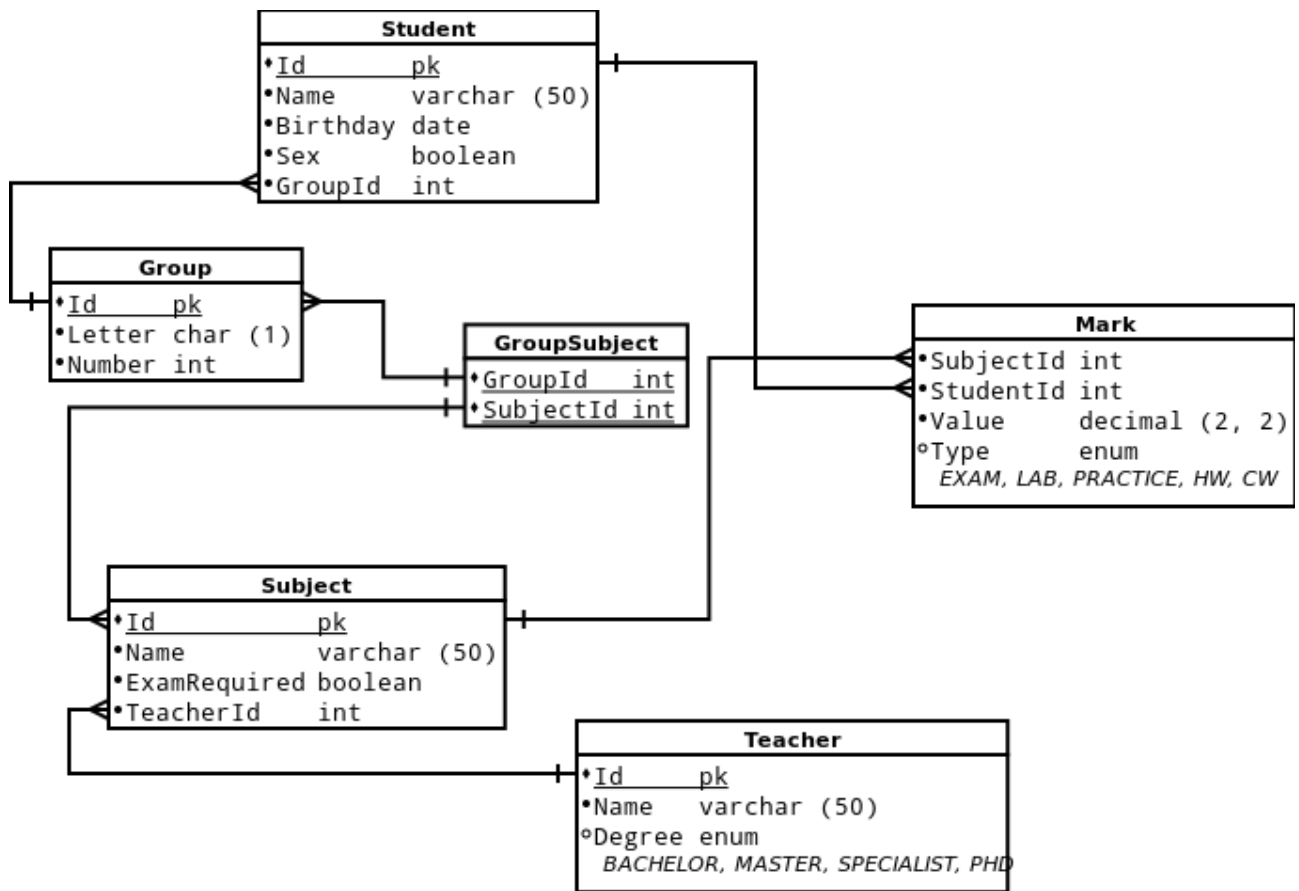
Агапов Георгий, группа М3437

Диаграмма сущность-связь



Легенда: пустой круг — необязательное поле, заполненный — обязательное.

Диаграмма «Физическая модель»



Легенда:

- пустой круг — необязательное поле, заполненный — обязательное.
- стрелочки-связи соединяют колонки, объединенные FOREIGN-ключами (в таблице со стороны Many)

Схема базы

```
CREATE TABLE Student (  
    Id SERIAL NOT NULL PRIMARY KEY,  
    Name VARCHAR (50) NOT NULL,  
    Birthday DATE NOT NULL,  
    Sex BOOLEAN NOT NULL,  
    GroupId INT NOT NULL  
);  
  
CREATE TYPE AcademicDegree AS ENUM ('BACHELOR', 'MASTER', 'SPECIALIST', 'PHD');  
  
CREATE TABLE Teacher (  
    Id SERIAL NOT NULL PRIMARY KEY,  
    Name VARCHAR (50) NOT NULL,  
    Degree AcademicDegree  
);  
  
CREATE TABLE StudentGroup (  
    Id SERIAL NOT NULL PRIMARY KEY,  
    Letter CHAR (1) NOT NULL,  
    Number INT NOT NULL  
);  
  
CREATE TABLE Subject (  
    Id SERIAL NOT NULL PRIMARY KEY,  
    Name VARCHAR (50) NOT NULL,  
    ExamRequired BOOLEAN NOT NULL,  
    TeacherId INT NOT NULL  
);  
  
CREATE TABLE GroupSubject (  
    GroupId INT NOT NULL,  
    SubjectId INT NOT NULL  
);  
  
CREATE TYPE MarkType AS ENUM ('EXAM', 'LAB', 'PRACTICE', 'HW', 'CW');  
  
CREATE TABLE Mark (  
    StudentId INT NOT NULL,  
    SubjectId INT NOT NULL,  
    Value DECIMAL (4, 2) NOT NULL,  
    Type MarkType  
);
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_GroupId FOREIGN KEY (GroupId)
REFERENCES StudentGroup (Id);
```

```
ALTER TABLE Subject ADD CONSTRAINT FK_Subject_TeacherId FOREIGN KEY (TeacherId)
REFERENCES Teacher (Id);
```

```
ALTER TABLE GroupSubject ADD CONSTRAINT FK_GroupSubject_GroupId FOREIGN KEY
(GroupId) REFERENCES StudentGroup (Id);
```

```
ALTER TABLE GroupSubject ADD CONSTRAINT FK_GroupSubject_SubjectId FOREIGN KEY
(SubjectId) REFERENCES Subject (Id);
```

```
ALTER TABLE Mark ADD CONSTRAINT FK_Mark_SubjectId FOREIGN KEY (SubjectId)
REFERENCES Subject (Id);
```

```
ALTER TABLE Mark ADD CONSTRAINT FK_Mark_StudentId FOREIGN KEY (StudentId)
REFERENCES Student (Id);
```

```
ALTER TABLE Mark ADD CONSTRAINT CH_Mark_Value CHECK (Value >= 0 AND Value <=
100);
```

-- Ограничение на величину суммарной оценки можно реализовать только на триггерах, не стал заморачиваться

-- Реализовать проверку на привязанность группы студента к предмету при добавлении оценки также представляется проблематичным

Тестовые данные

```
INSERT INTO StudentGroup (Id, Letter, Number) VALUES (1, 'M', 3437), (2, 'M', 3338);
INSERT INTO Student (Id, Name, Birthday, Sex, GroupId)
VALUES (1, 'George Agapov', '1994-04-30', True, 1), (2, 'George Konoplich', '1993-11-18', True,
1),
      (3, 'Mikhail Ivanov', '1995-09-12', True, 2), (4, 'Alina Ruslanova', '1995-02-04', False, 2);
INSERT INTO Teacher (Id, Name, Degree) VALUES (1, 'George Korneev', 'PHD'), (2, 'Andrew
Stankevich', 'PHD'), (3, 'Ivan Sorokin', 'MASTER');
INSERT INTO Subject (Id, Name, ExamRequired, TeacherId)
VALUES (1, 'Java: basic', True, 1), (2, 'Java: advanced', True, 1),
      (3, 'Translation methods', False, 2), (4, 'C++', False, 3), (5, 'Discrete math: semester 3', True, 2);

INSERT INTO GroupSubject (GroupId, SubjectId) VALUES (1, 1), (2, 1), (1, 4);

INSERT INTO Mark (StudentId, SubjectId, Value, Type)
VALUES -- Java : basic
      (1, 1, 50, 'HW'), (1, 1, 15, 'EXAM'),
      (2, 1, 10, 'PRACTICE'), (2, 1, 20, 'CW'), (2, 1, 20, 'HW'), (2, 1, 19, 'EXAM'),
      (3, 1, 34, 'HW'),
      (4, 1, 24, 'CW'), (4, 1, 20, 'EXAM'),
      -- C++
      (1, 4, 20, 'HW'), (1, 4, 25, 'HW'), (1, 4, 45, 'HW'),
      (3, 4, 45, 'HW')
```

Тестовые запросы

```
-- Оценки всех, сдавших какой-л. предмет
SELECT st.name, su.name, SUM(m.Value) as sum
FROM Subject su
JOIN GroupSubject gs ON gs.SubjectId = su.Id
JOIN StudentGroup g ON g.Id = gs.GroupId
JOIN Student st ON st.GroupId = g.Id
LEFT JOIN Mark m ON m.SubjectId = su.Id AND m.StudentId = st.Id
GROUP BY st.id, su.id
HAVING ((SUM(m.Value) IS NOT NULL) AND SUM(m.Value) >= 60) AND (NOT
su.examRequired OR BOOL_OR(m.Type = 'EXAM'));
```

-- Оценки всех, не сдавших какой-л. предмет

SELECT st.name, su.name, SUM(m.Value) as sum

FROM Subject su

JOIN GroupSubject gs ON gs.SubjectId = su.Id

JOIN StudentGroup g ON g.Id = gs.GroupId

JOIN Student st ON st.GroupId = g.Id

LEFT JOIN Mark m ON m.SubjectId = su.Id AND m.StudentId = st.Id

GROUP BY st.id, su.id

HAVING ((SUM(m.Value) IS NULL) OR SUM(m.Value) < 60) OR (su.examRequired AND NOT
BOOL_OR(m.Type = 'EXAM'));

-- Все предметы преподавателя Andrew Stankevich

SELECT su.name as subjectName, t.name as teacherName FROM Subject su

JOIN Teacher t ON t.Id = su.TeacherId

WHERE t.Id = 2;

-- Все девочки университета

SELECT st.name, CONCAT(g.Letter, g.Number) as groupName

FROM Student st

JOIN StudentGroup g ON g.Id = st.GroupId

WHERE st.sex = False;

-- Все студенты, слушающие курсы преподавателя Ivan Sorokin

SELECT st.name, su.name FROM Subject su

JOIN GroupSubject gs ON gs.SubjectId = su.Id

JOIN StudentGroup g ON g.Id = gs.GroupId

JOIN Student st ON st.GroupId = g.Id

WHERE su.TeacherId = 3;