Отчёт лабораторной 1

Был установлен клиент PGAdmin. И также установлен, и запущен локальный сервер базы данных.

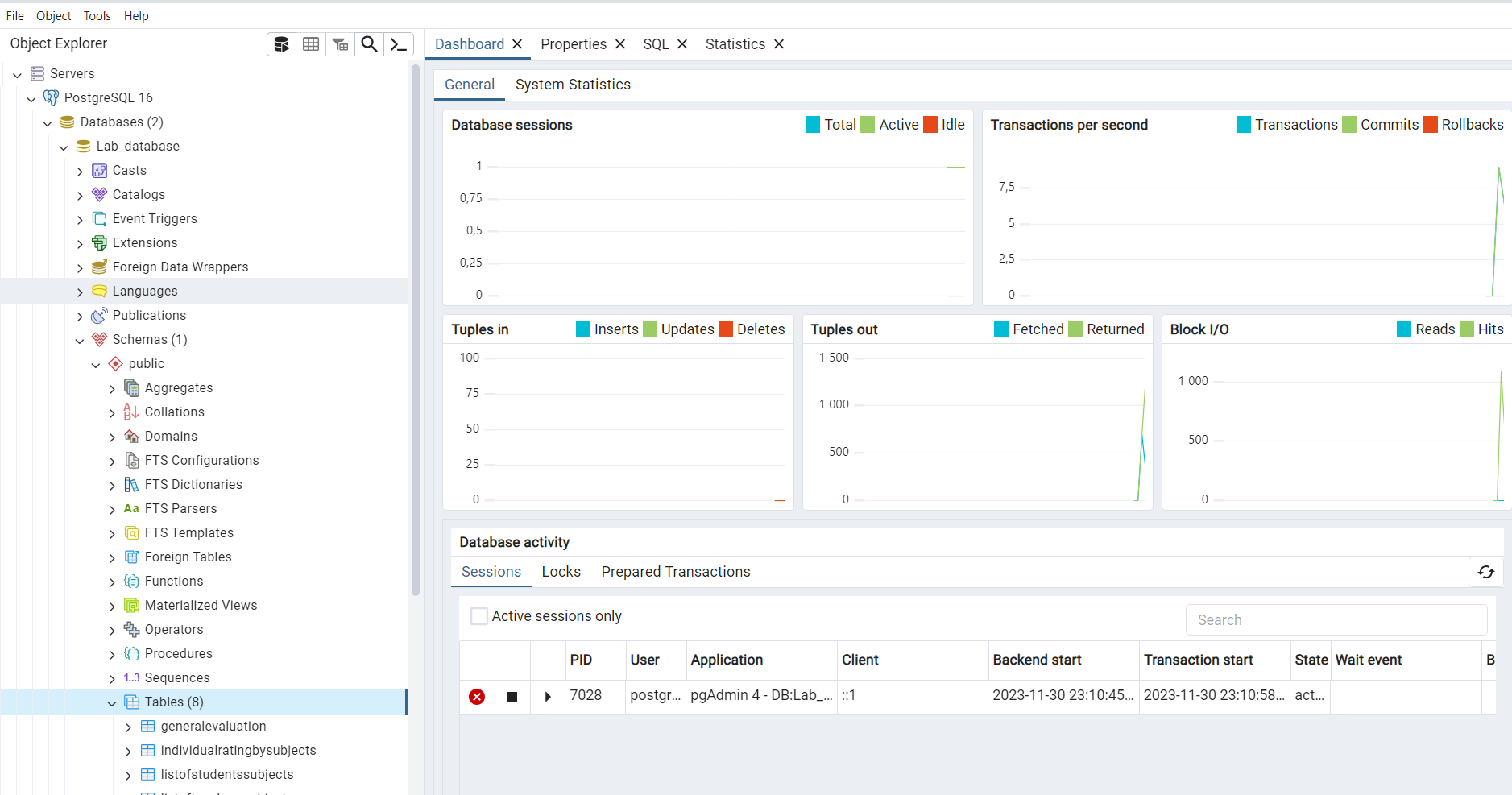


Рисунок 1 – PGAdmin

Далее были спроектированы таблицы и связи в них по предложенной модели из варианта 11.

* Фамилия студента.
* Имя студента.
* Отчество студента.
* Домашний адрес студента.
* Дата рождения.
* Предмет.
* Месяц, за который определяется рейтинг.
* Рейтинговая оценка.
* Максимальная рейтинговая оценка.
* Фамилия, имя, отчество преподавателя.

В данной таблице содержится интерпретация балловой системы. Чтобы учесть максимально возможную рейтинговую оценку было указано ограничение <11.

CREATE TABLE TypeRating (

Id int PRIMARY KEY generated always as identity (increment by 1),

Score int NOT NULL CHECK (Score < 11),

Interpretation varchar(100)

);

Таблица описывающая каждого студента в учебном заведении.

CREATE TABLE Students (

Id int PRIMARY KEY generated always as identity (increment by 1),

Surname varchar(50) NOT NULL,

MiddleName varchar(64),

Name varchar(64) NOT NULL,

Address varchar(100) NOT NULL,

DateOfBirth DATE NOT NULL

);

Описание предметов

CREATE TABLE Subjects (

Id int PRIMARY KEY generated always as identity (increment by 1),

Name varchar(100) NOT NULL

);

Таблица описывающая каждого преподавателя в учебном заведении.

CREATE TABLE Teachers (

Id int PRIMARY KEY generated always as identity (increment by 1),

Surname varchar(50) NOT NULL,

MiddleName varchar(64),

Name varchar(64) NOT NULL

);

Список из пар внешних ключей таблицы студентов и таблицы предметов. Каждому предмету соответствует множество студентов. И каждому студенту – множество предметов.

CREATE TABLE ListOfStudentsSubjects

(

Id serial PRIMARY KEY,

StudentId int NOT NULL REFERENCES Students (Id),

SubjectId int NOT NULL REFERENCES Subjects (Id),

UNIQUE (StudentId, SubjectId)

);

Тот же самый список, но в отношении преподавателей. Несколько преподавателей ведут один предмет и множество предметов ведёт один преподаватель.

CREATE TABLE ListOfTeachersSubjects

(

Id serial PRIMARY KEY,

TeacherId int NOT NULL REFERENCES Teachers (Id),

SubjectId int NOT NULL REFERENCES Subjects (Id),

UNIQUE (TeacherId, SubjectId)

);

Id данной таблицы состоит из Id ListOfStudentsSubjects, так как для каждой пары из ListOfStudentsSubjects найдётся своя рейтинговая оценка.

CREATE TABLE IndividualRatingBySubjects

(

Id int NOT NULL PRIMARY KEY REFERENCES ListOfStudentsSubjects (Id),

TypeRatingId int NOT NULL REFERENCES TypeRating (Id),

UNIQUE (Id, TypeRatingId)

);

Таблица итоговой отчётности, содержит в себе дату обновления рейтинга и ивенты их обновления.

CREATE TABLE GeneralEvaluation

(

Id int NOT NULL PRIMARY KEY REFERENCES ListOfStudentsSubjects (Id),

FirstSemester int REFERENCES TypeRating (Id),

SecondSemester int REFERENCES TypeRating (Id),

UpdateDate Date NOT NULL

);