**Связь 1 к 1: В дочерней таблице просто подтягивается оригинальный столбец, являющейся primary key родительской таблицы**

CREATE TABLE IF NOT EXISTS Student (

**id SERIAL PRIMARY KEY,**

email VARCHAR(80) UNIQUE NOT NULL,

name VARCHAR(40) NOT NULL,

PASSWORD VARCHAR(128) NOT NULL);

CREATE TABLE IF NOT EXISTS StudentInfo (

**id INTEGER PRIMARY KEY REFERENCES Student(id)**,

birthday date,

city VARCHAR(60),

roi TEXT);

Создана связь 1к1 через PRIMERY KEY id

**Связь 1 ко многим – в дочерней таблице создаем свой столбец со своим именем, и заполняем его значениями из столбца primery key родительской таблицы**

CREATE TABLE IF NOT EXISTS Course (

**id SERIAL PRIMARY KEY**,

name VARCHAR(60) NOT NULL,

description TEXT);

CREATE TABLE IF NOT EXISTS HomeworkTask (

id SERIAL PRIMARY KEY,

**course\_id INTEGER NOT NULL REFERENCES Course(id)**,

number INTEGER NOT NULL,

description TEXT NOT NULL);

Создана связь 1 ко многим через PRIMERY KEY id курса

**Связь многие ко многим:**

**1. осуществляется через дополнительную таблицу, содержащую два ключа на две внешних таблицы.**

CREATE TABLE IF NOT EXISTS CourseStudent (

course\_id INTEGER REFERENCES Course(id),

student\_id INTEGER REFERENCES Student(id),

**CONSTRAINT pk PRIMARY KEY (course\_id, student\_id)**);

**Создался составной основной ключ pk состоящий из двух id**

**2. Создаем свой уникальный, ни с чем не связанный основной ключ и подтягиваем из дочерних таблиц нужные id (что бы один и тот же студент мог многократно сдавать одну и ту же домашнюю работу после доработки)**

CREATE TABLE IF NOT EXISTS HomeworkSolution (

id SERIAL PRIMARY KEY,

**task\_id INTEGER NOT NULL REFERENCES HomeworkTask(id),**

**student\_id INTEGER NOT NULL REFERENCES Student(id),**

solution TEXT NOT NULL);