CST8215 – Lab 10 Submission

Student Number: \_\_\_\_\_\_\_41019241\_\_\_\_\_

Student Name: \_\_\_\_\_\_\_Adam Di Cioccio\_\_\_\_\_\_\_\_\_\_

Section Number: \_321\_\_ (e.g. 322, 323, 363)

Date: \_\_November 11, 2020\_\_

Objective

Being able to create a new database from a database backup file, using the features in PG Admin

Become comfortable using a data modelling tool such as PG modeler and to creat a Physical Data Model by reverse engineering an existing PG Admin database

Being able to identify the parent / child relationships and the cardinalities in ‘english words’

Identify any ‘many to many’ relationships found in the new physical data model after reverse engineer process and describe the intermediate table and the relationships

Procedure

Where you see *Provide screenshot here* you are being asked to take the screenshot from PG Admin (both the top pane where you have typed in the SQL statement and the bottom pane showing the results of the SQL statement)

Submission

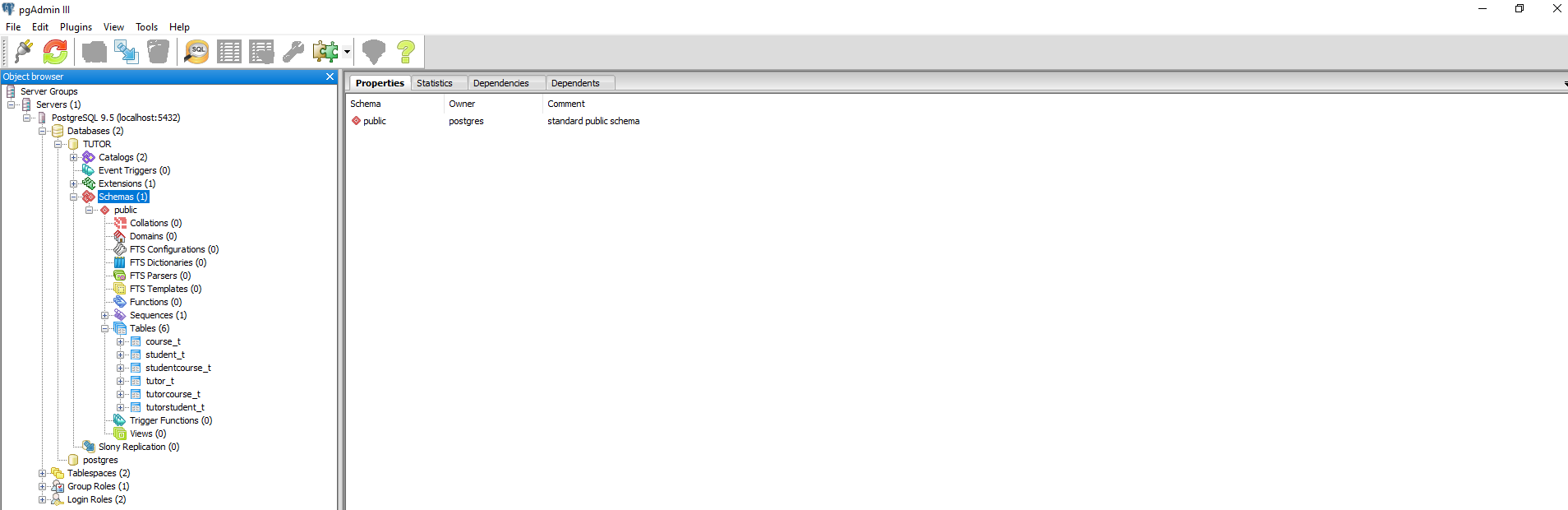
**Step 1 – RESTORE TUTOR database backup**

Create and populate a new PG Admin database named TUTOR using the provided TUTOR.backup file.

Once the database has been restored, verify all the tables have been created and there is data in each of the tables.

1. Expand the explorer window in PG Admin so all the tables are shown and expand each table so that the number of columns is also shown --- this will demonstrate that the tables have been created.

*Provide screenshot here*



1. Using a SQL window, perform ‘select count(\*)’ statement for three of the tables just created and provide three (3) separate screenshots – this will demonstrate that your new tables have been populated correctly.

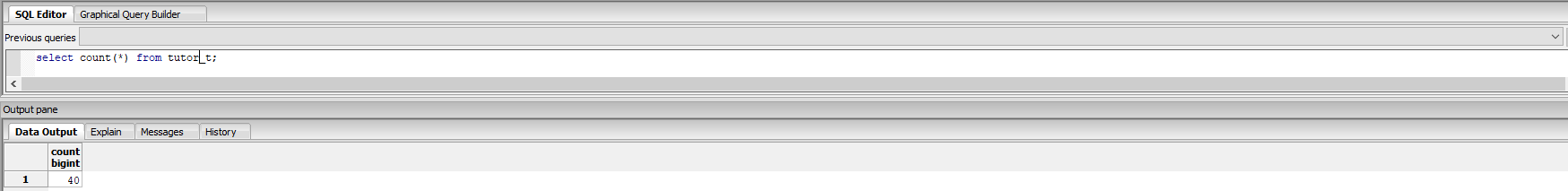
*Provide screenshot here (1)*



*Provide screenshot here (2)*



*Provide screenshot here (3)*

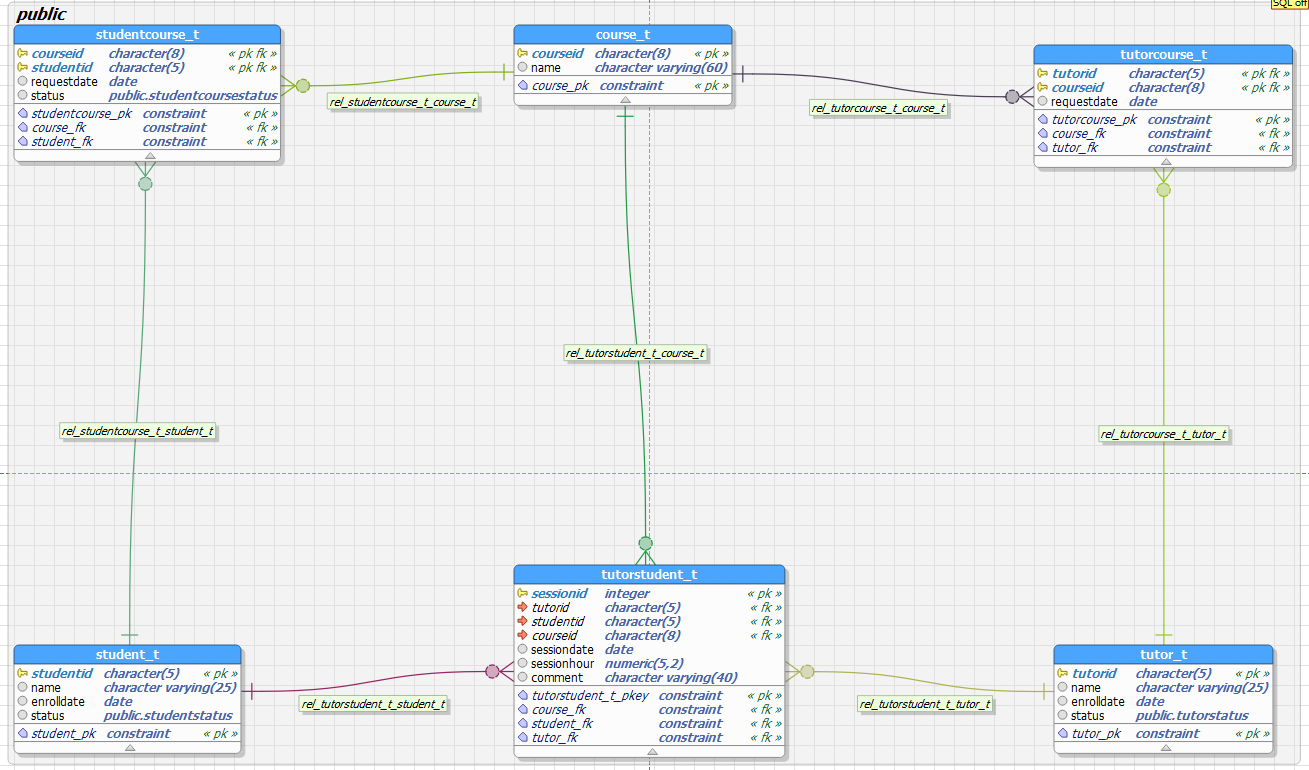


**Step 2 – Generate the physical data model from TUTOR database**

Create the physical data model using PG Modeler for this new TUTOR database.

Ensure the model is clean and readable and produce a screenshot.

*Provide screenshot here (2)*



**Step 3 – Identify the relationships, cardinalities and any associative entities/intermediate tables**

Using the data model screenshot from Step 2, document the model as the following example shows.

Ensure that you identify **every** table shown in your model.

Remember that every relationship is explained both directions (left to right **and** right to left) therefore your explanation needs to speak to both directions.

*Associative entities: studentcourse\_t, tutorstudent\_t, tutorcourse\_t*

*Right to left 🡪 A single COURSE (parent) can be related to zero (0) or many STUDENTCOURSEs (child)*

*Left to Right 🡪 A single STUDENTCOURSE (child) is related to only one (1) COURSE (parent)*

***Right to left 🡪 A single COURSE (parent) can be related to zero (0) or many TUTORCOURSE (child)***

***Left to Right 🡪 A single TUTORCOURSE (child) is related to only one (1) COURSE (parent)***

*Up to down 🡪 A single COURSE (parent) can be related to zero (0) or many TUTORSTUDENTs (child)*

*Down to up 🡪**A single TUTORSTUDENT (child) is related to only one (1) COURSE (parent)*

***Down to up 🡪 A single student (parent) can be related to many student courses (parent) (NOT 0)***

***Up to down 🡪 A single student course (child) can be related to one student (parent)***

*Down to up 🡪* ***A single tutor (parent) can be related to many tutor courses (parent) (NOT 0)***

*Up to down 🡪 A single tutor course (child) can be related to a single tutor (parent).*

***Right to left 🡪 A single tutor student (child) can be related to one student (parent).***

***Left to Right 🡪 A single student (parent) can be related to 0 or many tutor students (child).***

*Right to left 🡪 A single tutor (parent) can be related to many tutor students (child).*

*Left to Right 🡪 A single tutor student (child) can be related to one tutor (parent)*