## **Creating an OLTP database structure**

### **1.1 Creating a database “OLTP\_OnlineCourses”**

1. Open **pgAdmin**.

2. Create a new database named: **OLTP\_OnlineCourses**.

### **1.2 Creating the basic OLTP schema**

1. Open the file create\_oltp\_schema.sql.
2. Run the **entire script** in the database OLTP\_OnlineCourses.

After the execution, you should have the following **9 tables**:

1) Users  
2) Categories  
3) Course  
4) Records  
5) Rating  
6) Certificate  
7) Payment  
8) Instructors  
9) Course\_Instructors

### **1.3 Preparing temporary tables and uploading data**

1. Open the file load\_csv\_to\_oltp.sql.
2. Execute **the part of the script** that contains only **CREATE TABLE** for temporary tables:
3. CREATE TABLE temp\_users (...);
4. CREATE TABLE temp\_categories\_and\_courses (...);
5. CREATE TABLE temp\_records\_and\_ratings (...);
6. CREATE TABLE temp\_cert\_pay (...);
7. CREATE TABLE temp\_instructors (...);

After completion, **5 more temporary tables** will be added, and the total number of tables will be **14**.

## 

## **Step 2: Import data from CSV**

### **2.1 Import CSV files into the corresponding temporary tables** (all \*.csv files are located in the OnlineCourses\_scv\_files folder)

Go to:

**OLTP\_OnlineCourses → Schemas → Tables**

For each table, follow these steps:

#### **temp\_users**

Right click → **Import/Export Data**

In the Filename field, specify the file:  
 Untitled spreadsheet - Users (2).csv

Click **OK**

Then **everything is the same**, but for the rest of the tables :

#### **temp\_categories\_and\_courses**

Select a file:  
 categories\_and\_courses (2).csv

#### 

#### 

#### **temp\_records\_and\_ratings**

Select a file:  
 records\_and\_ratings.csv

#### **temp\_cert\_pay**

Select a file:  
 certificates\_and\_payments.csv

#### **temp\_instructors**

Select a file:  
 Instructors (2).csv

**Step 3: Loading data from temporary tables to main tables**

Now, after the data has been loaded into the temporary tables, we can insert them into the main tables.

**For this :**

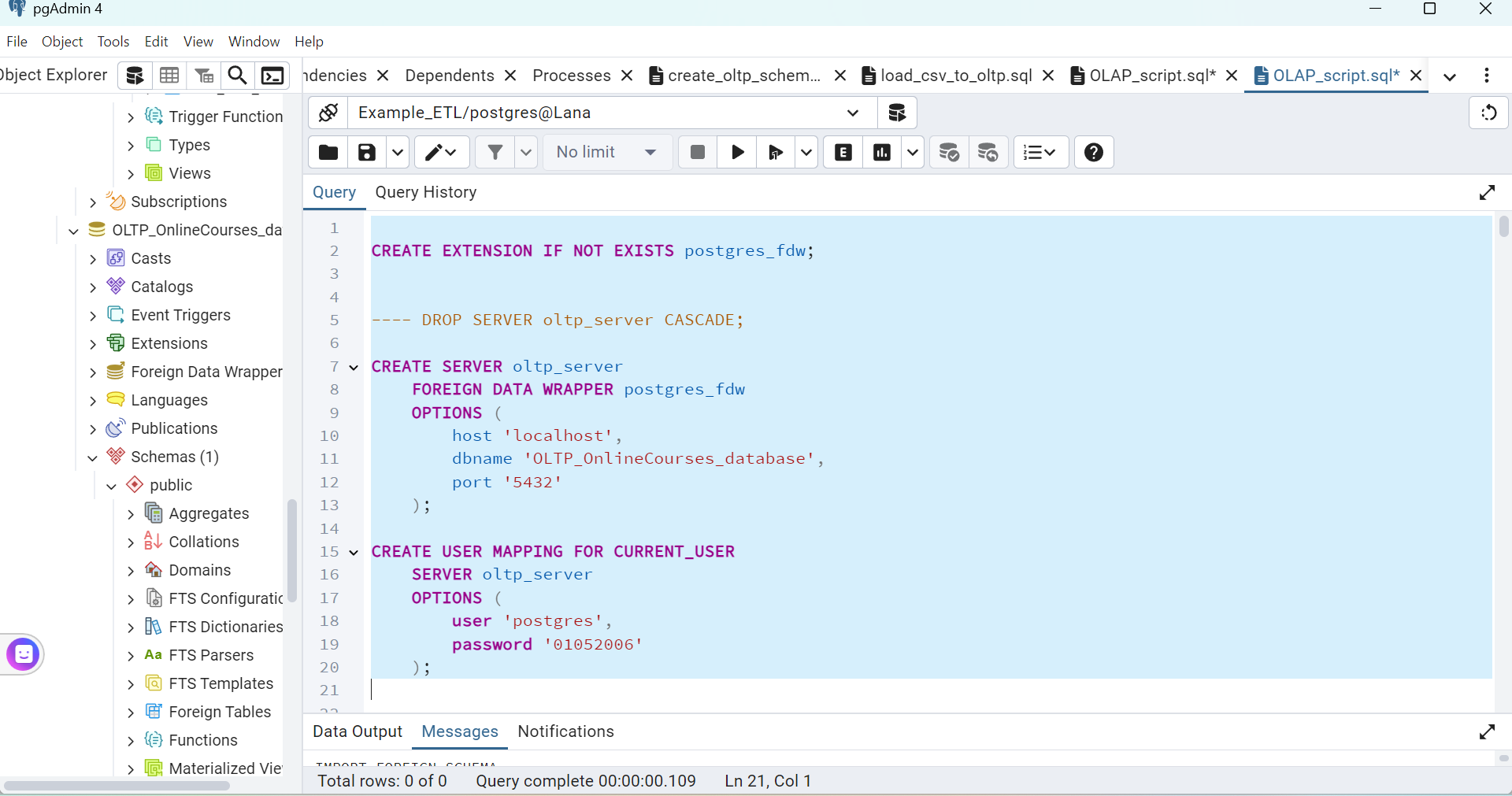
1. Run the rest of the INSERT INTO table\_name script.

The data has now been uploaded to the main tables.

**Step 4: OLAP and ETL**

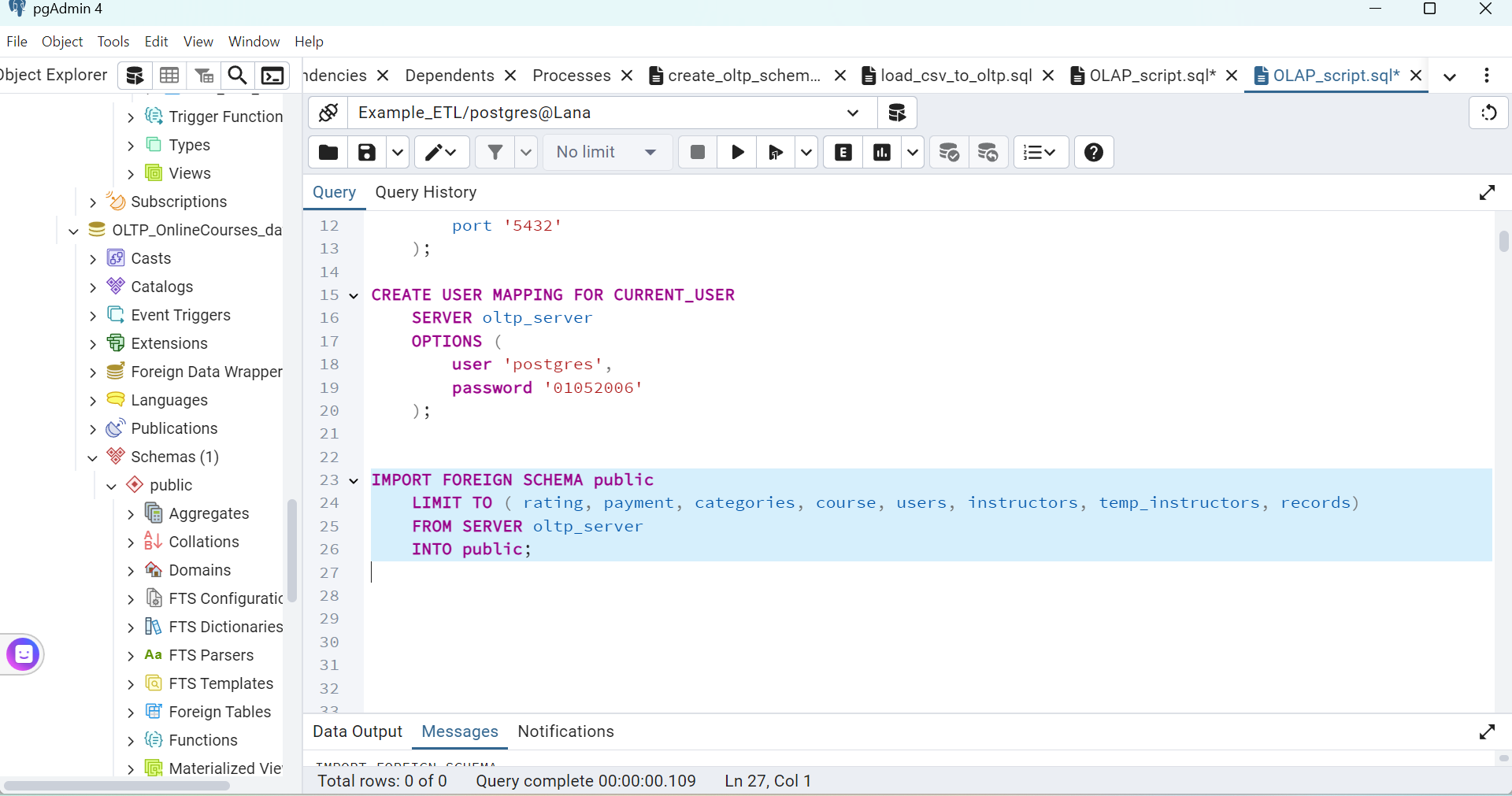
Open the file olap\_setup.sql.

In order to connect to the existing OLTP database “OLTP\_OnlineCourses” and retrieve the data from there, you need this part of the script :



**2. Import tables from the OLTP database**

Next, we import the tables, for this we execute this part of the script :



**3. Creating and uploading dimensions**

Open the file olap\_schema.sql

And we execute this part of the script :

**dwh\_Dim\_Time**

CREATE TABLE dwh\_Dim\_Time (...);

**dwh\_Dim\_Category**

CREATE TABLE dwh\_Dim\_Category (...);

**dwh\_Dim\_Course**

CREATE TABLE dwh\_Dim\_Course (...);

**dwh\_Dim\_User ( SCD Type 2 )**

CREATE TABLE dwh\_Dim\_User (...);

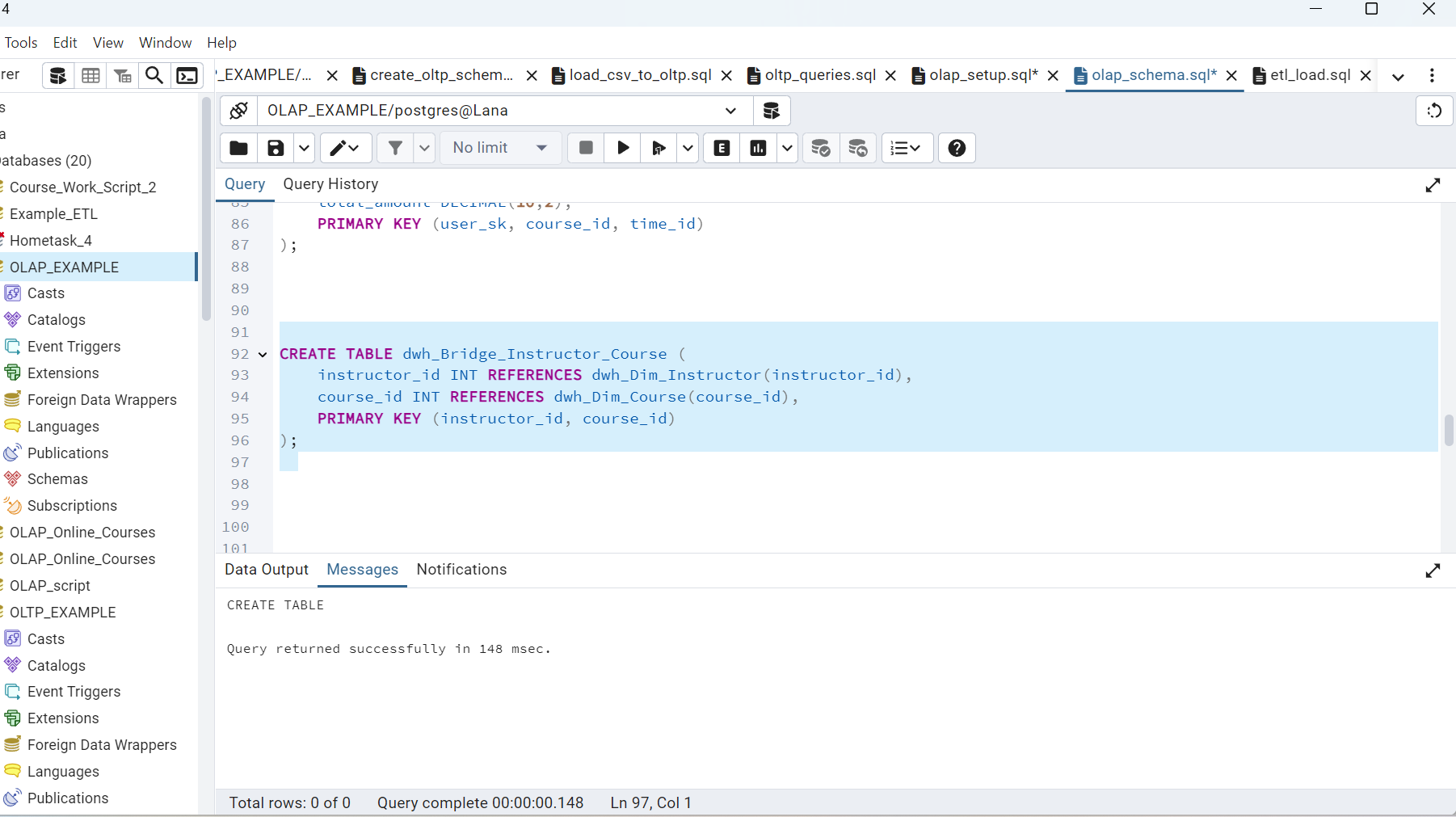
**dwh\_Dim\_Instructor**

CREATE TABLE dwh\_Dim\_Instructor (...);

**4. Creating and loading a bridge table**

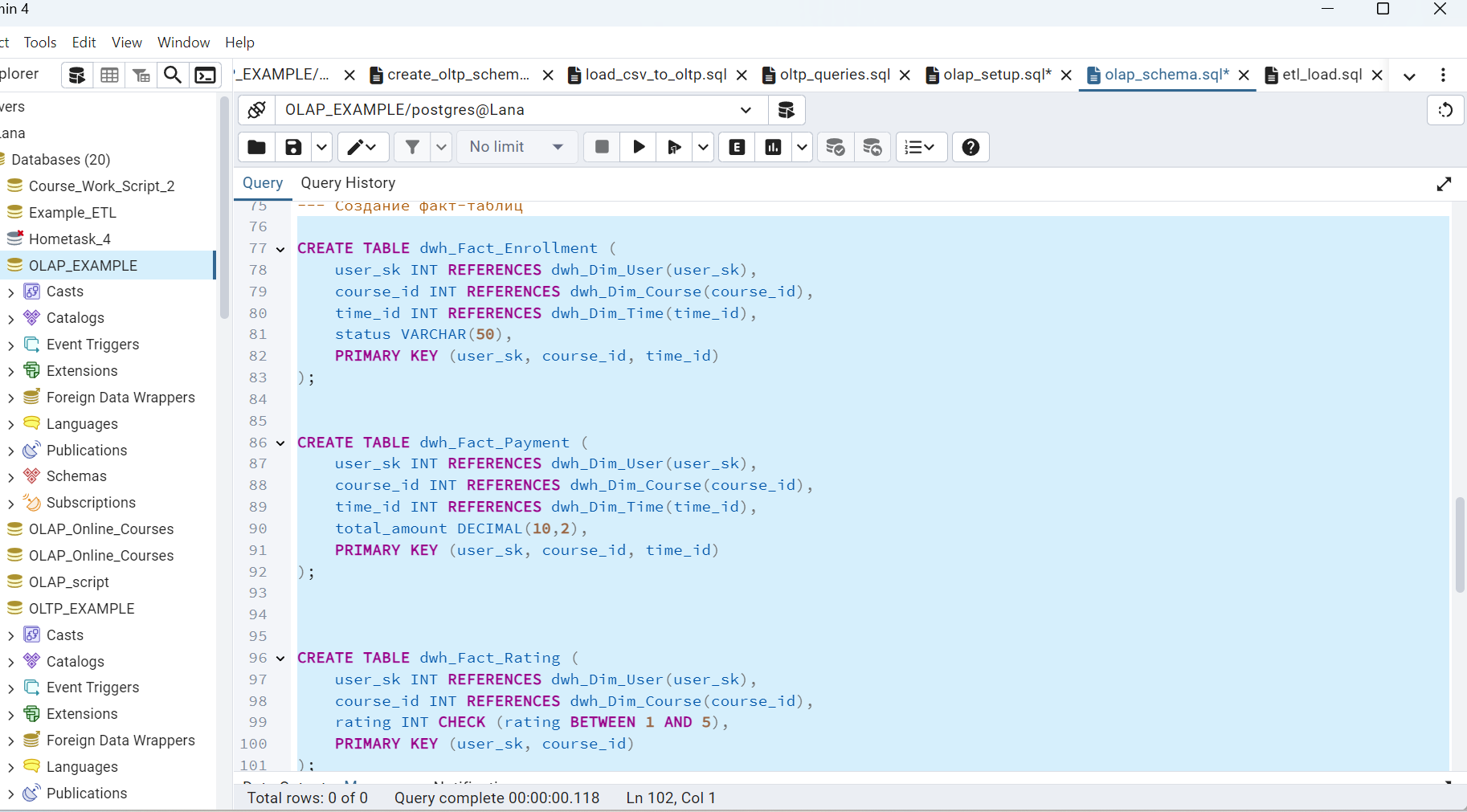
To create a linking table, run this part of the script :

dwh\_Bridge\_Instructor\_Course (Connection of courses and instructors)



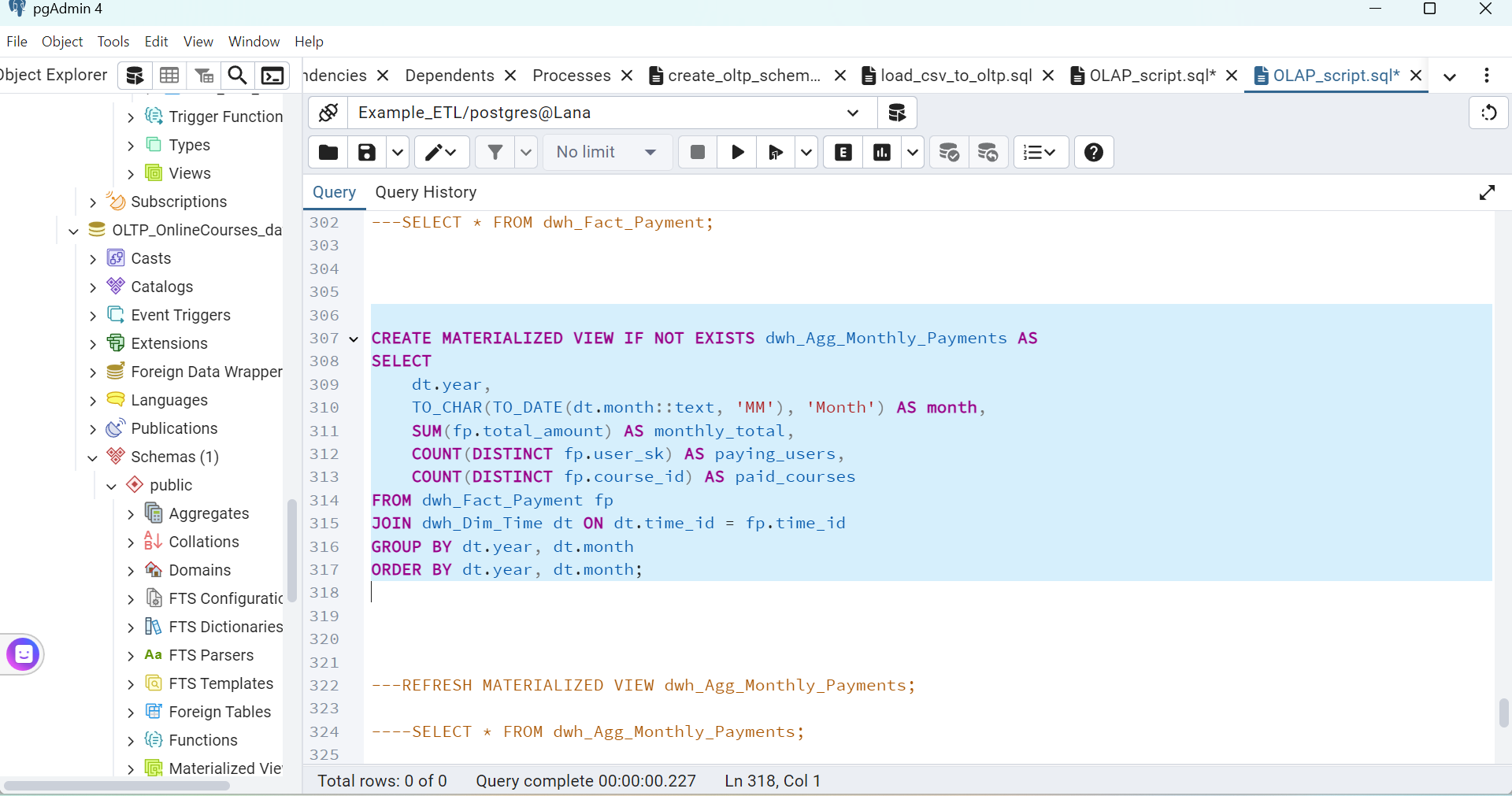
**5. Creating fact tables**

1. **dwh\_Fact\_Rating** (The fact of the ratings), **dwh\_Fact\_Enrollment** (The fact of enrolling in courses) and **dwh\_Fact\_Payment** (The fact of payments)

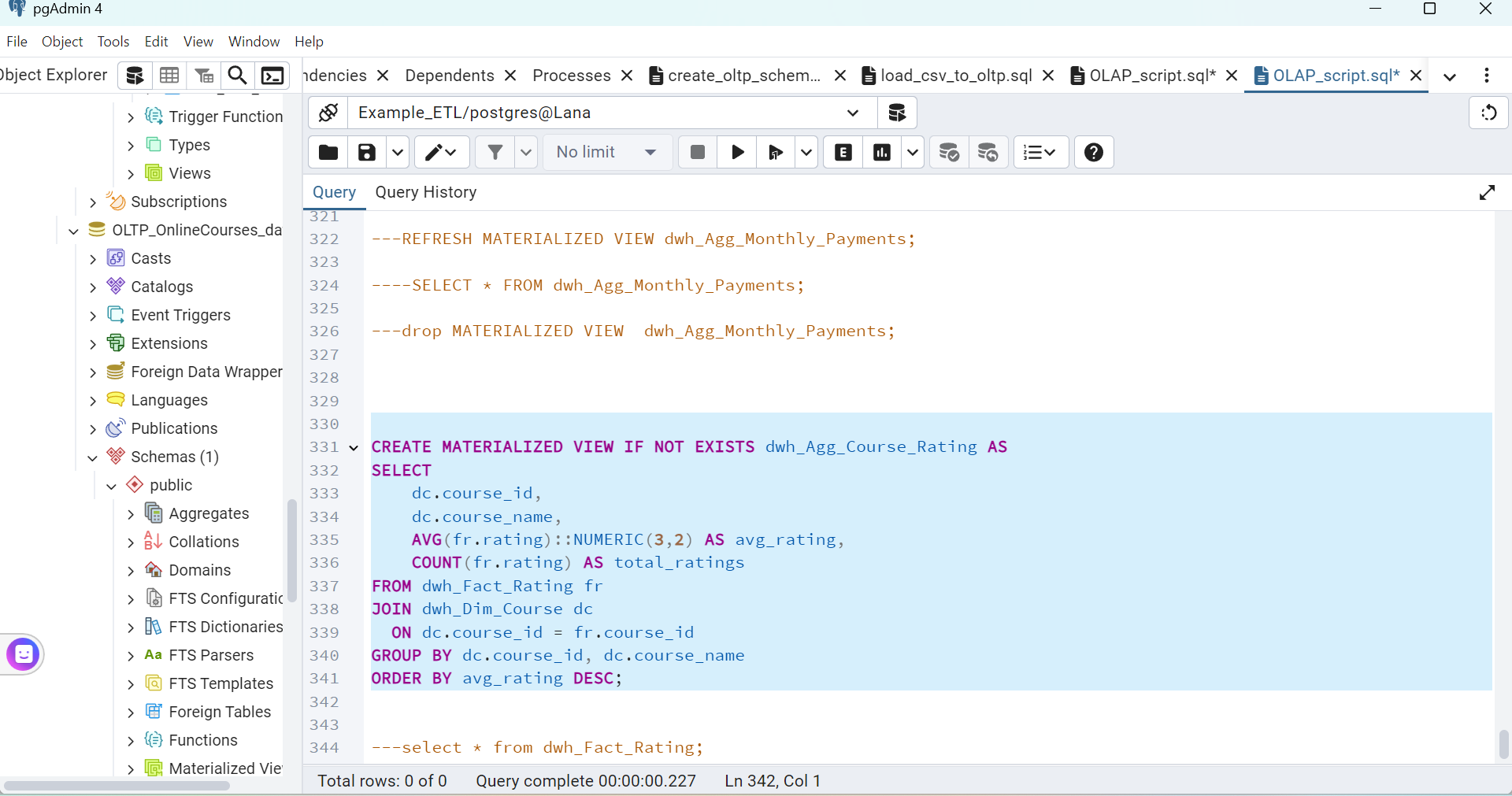


**6. Creating aggregates (materialized views)**

1. Monthly payments:



1. Average course grade :



**Step 5: Download the data**

1. Open the file etl\_load.sql

Uploading data to **Dimension Tables**

To do this, run this part of the script :

**INSERT INTO dwh\_Dim\_Time** — dates from the Payment table, with extraction of the day, month, quarter, and year.

**INSERT INTO dwh\_Dim\_Category** — unique categories from the Categories table.

**INSERT INTO dwh\_Dim\_Course** — courses linked to categories.

**INSERT INTO dwh\_Dim\_User** — users with SCD Type 2 support (tracking changes by name and email fields).

**INSERT INTO dwh\_Dim\_Instructor** — instructors.

**INSERT INTO dwh\_Bridge\_Instructor\_Course** — a bridge table between courses and teachers.

All inserts were made only for new records using **NOT EXISTS** or **NOT IN** to avoid duplication.

Next, upload the data to the **Fact Tables**

**INSERT INTO dwh\_Fact\_Rating**

**INSERT INTO dwh\_Fact\_Enrollment**

**INSERT INTO dwh\_Fact\_Payment**

When inserting into the fact tables, **JOINS** were applied to the dimensions and checking for the absence of duplicate records **(NOT EXISTS)**.

**DISTINCT** was also used in certain places to avoid mistakes when violating unique restrictions.