**NORMALIZATION**

**NORMALIZATION :**

Normalization is the process to eliminate data redundancy and enhance data integrity in the table. Normalization also helps to organize the data in the database. It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables.

Normalization organizes the columns and tables of a database to ensure that database integrity constraints properly execute their dependencies. It is a systematic technique of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update, and Deletion anomalies.

There are three main types of normal forms (NF) : First Normal Form(1NF), Second Normal From(2NF), Third Normal Form(3NF) .

**First Normal Form(1NF) :** A relation will be 1NF if it contains an atomic value and distinct (unique) values.

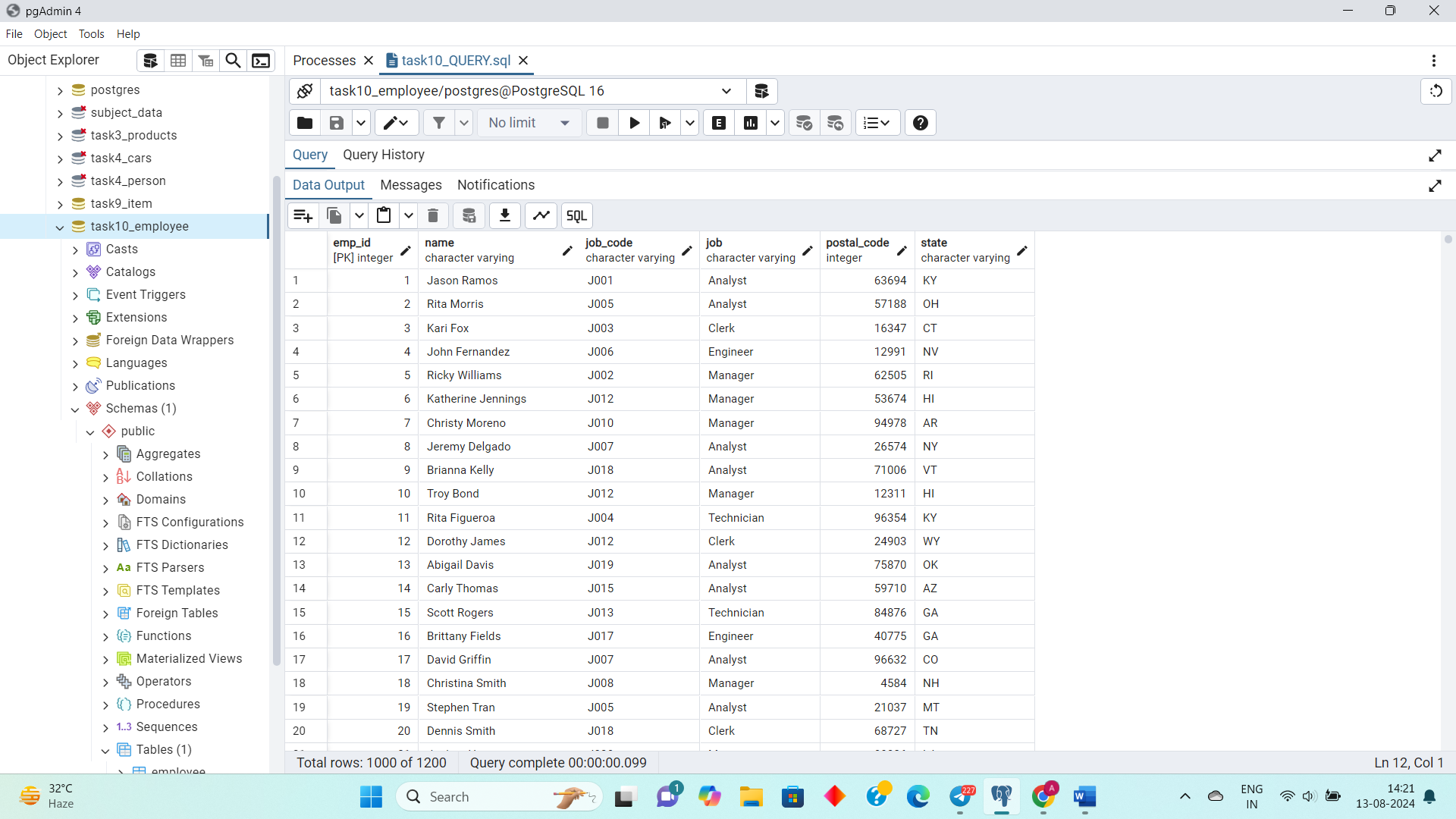
**Second Normal Form(2NF) :**In the second normal form, all non-key attributes are fully functional dependent on the primary key.

**Third Normal Form(3NF) :** In 3NF there is no transitive dependency for non-prime attributes.

**EXAMPLE:**

We have an ‘employee’ table with the following columns:

* **emp\_id** : Unique identifier for each employee.
* **name** : Name of the employee.
* **job\_code** : Code associated with the job.
* **job** : Description of the job.
* **postal\_code** : Postal code where the employee lives.
* **state** : State where the employee lives.

****

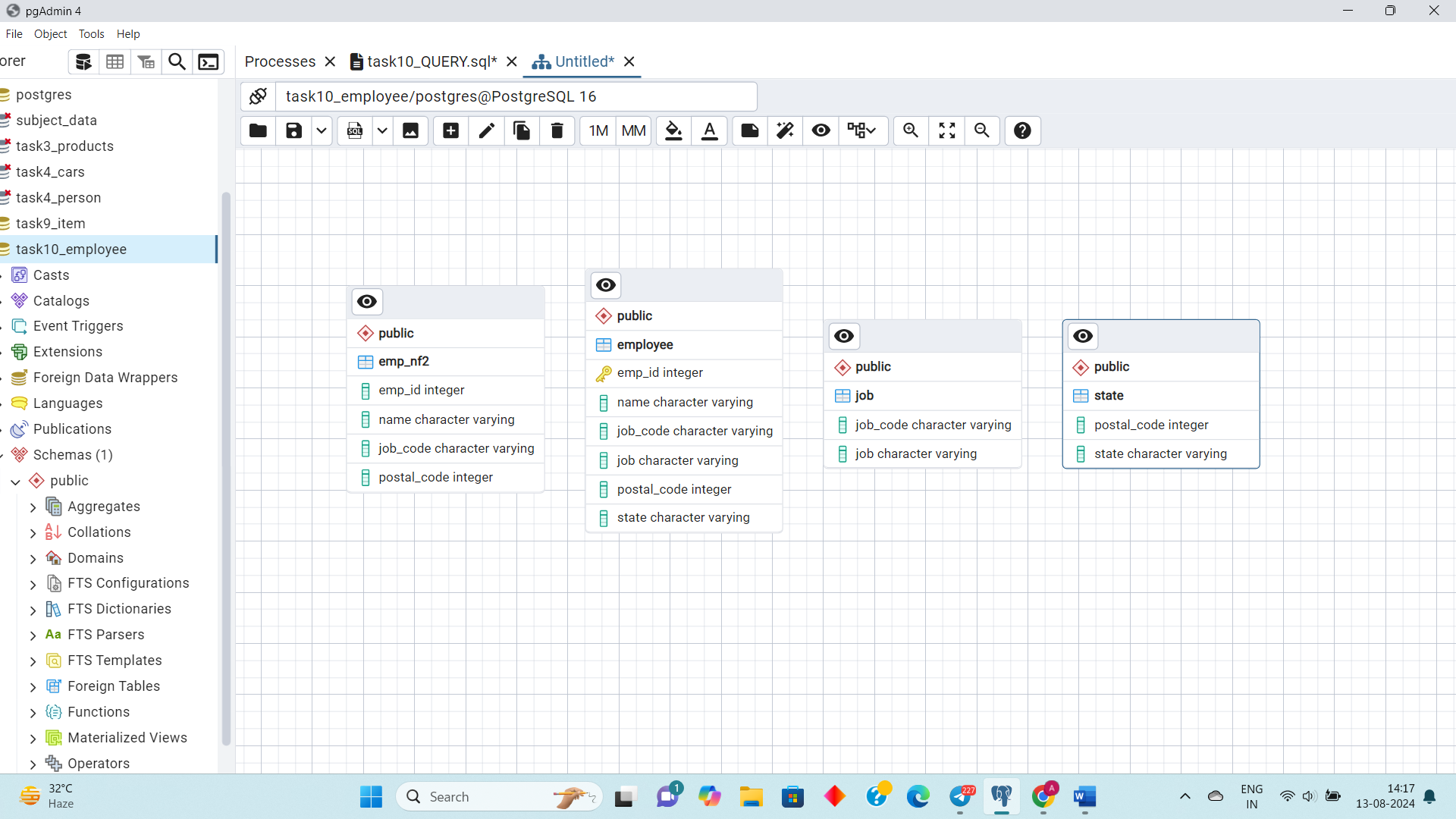
**Step 1: First Normal Form (1NF)**

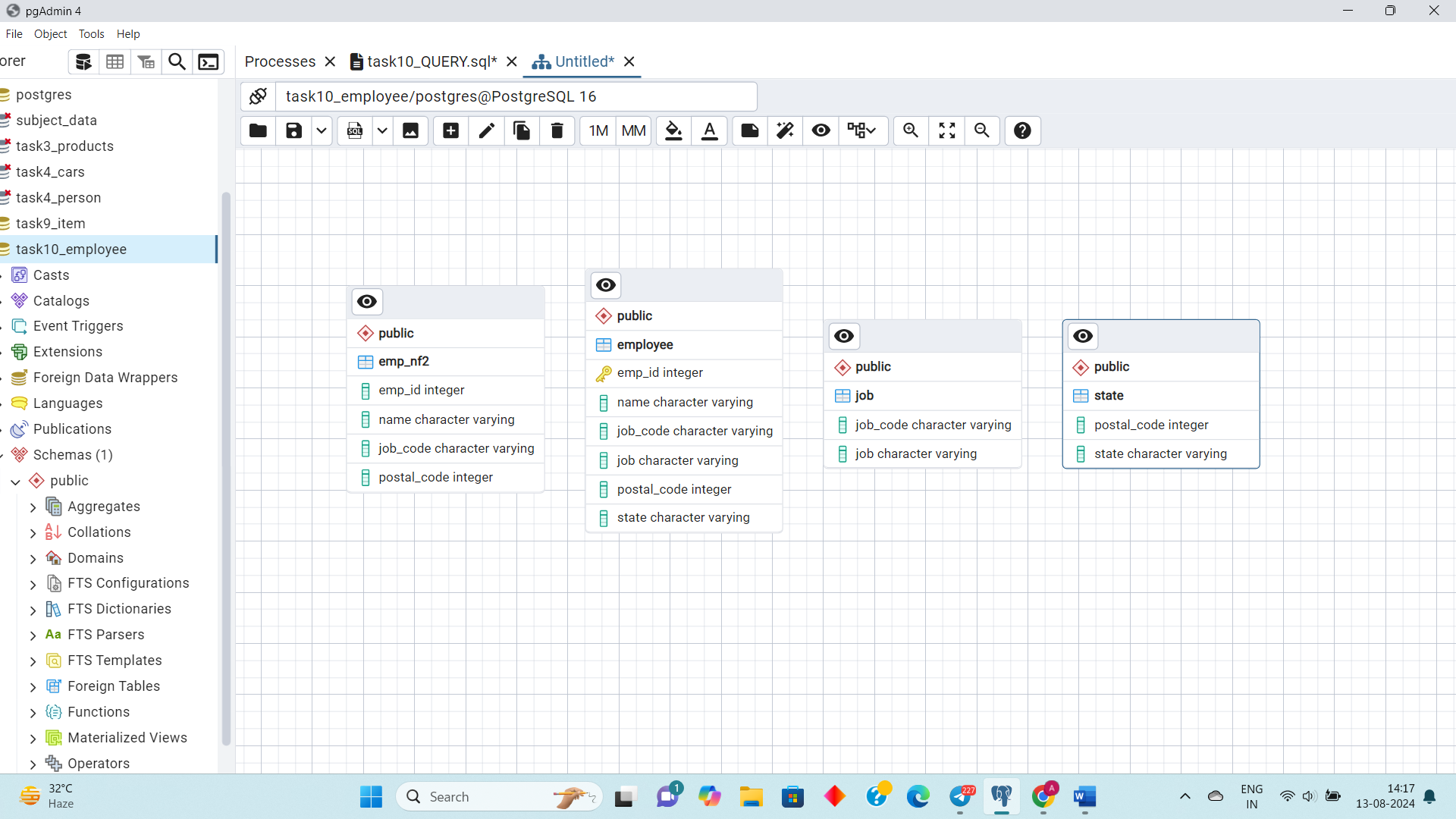
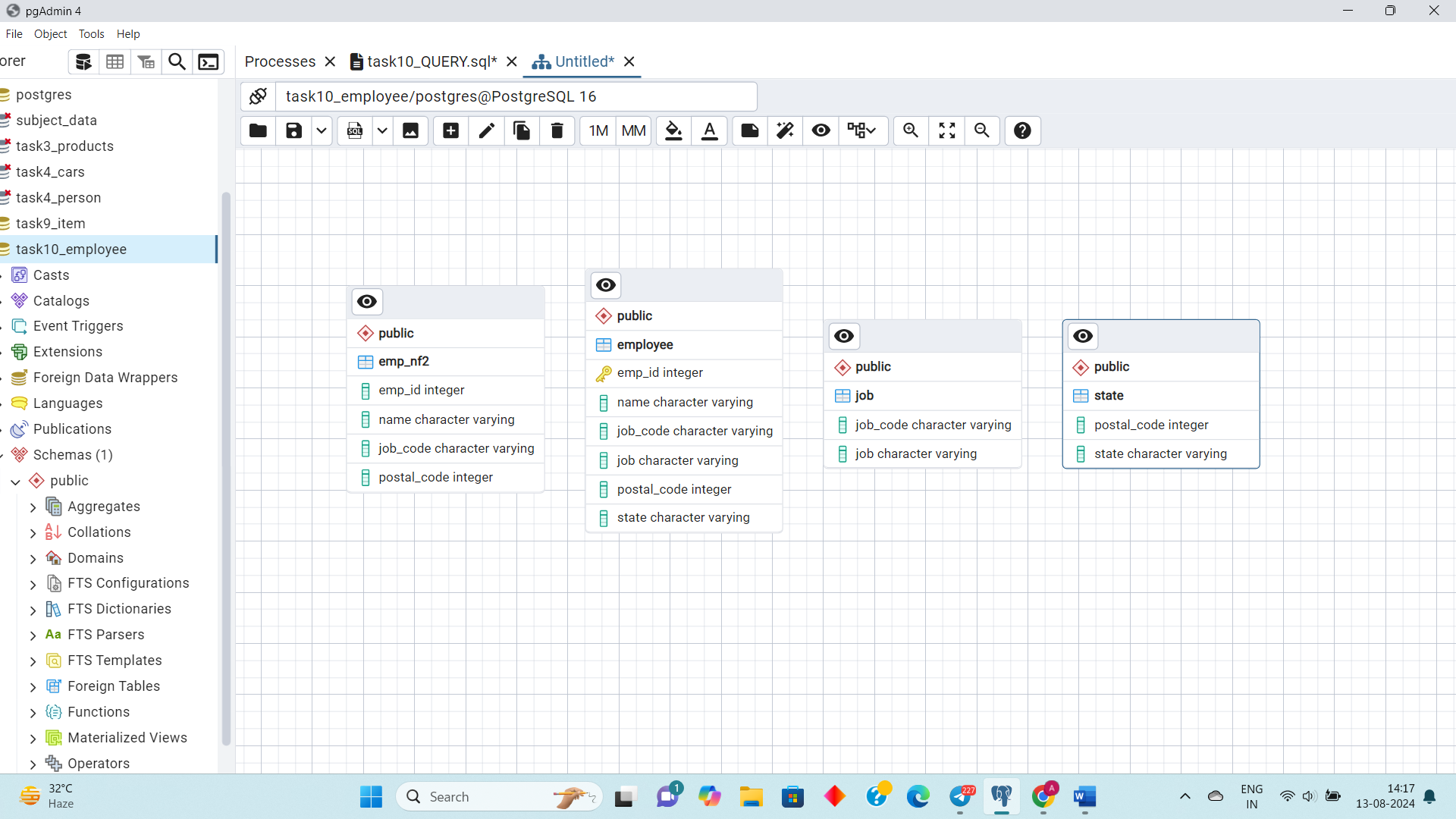
**1NF Requirement:** Eliminate repeating groups and ensure that each field contains only atomic (indivisible) values.

In this case, the original table seems to be in 1NF because all columns contain atomic values.

### Step 2: Second Normal Form (2NF)

**2NF Requirement:** Ensure the table is in 1NF, and all non-key attributes are fully dependent on the primary key. Remove partial dependencies.

Here, **‘job\_code’** and **‘job’** are related, and **‘postal\_code’** and **‘state’** are related. To move to 2NF, we'll separate these into different tables.



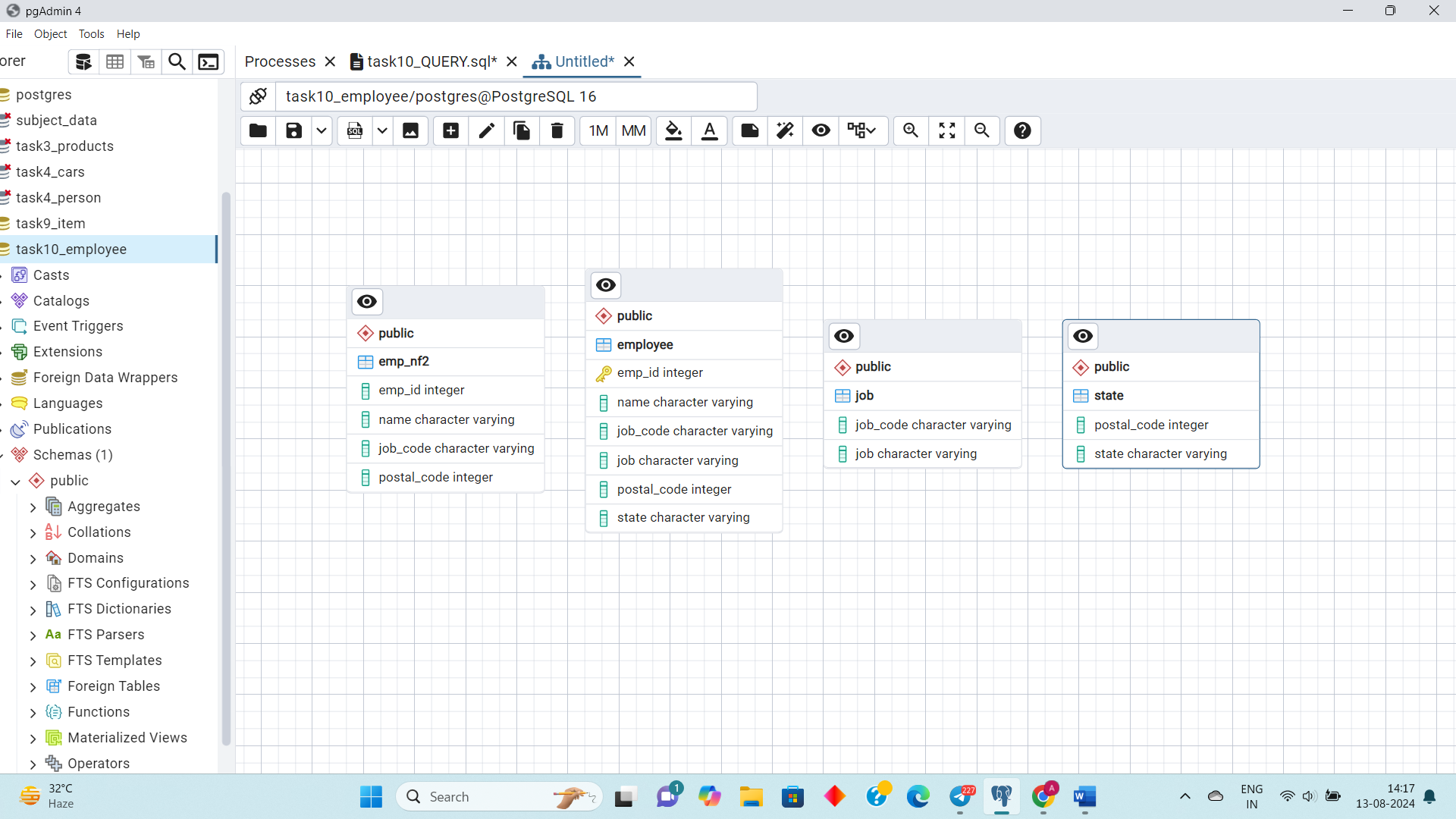
### Step 3: Third Normal Form (3NF)

**3NF Requirement:** Ensure the table is in 2NF, and all the attributes are only dependent on the primary key (remove transitive dependencies).

In this case, the **‘job’** and **‘state’** tables are already normalized. We need to ensure there are no transitive dependencies in the **‘emp\_nf2’** table, which is the case here as well.

Therefore, the structure after 2NF is already in 3NF, as there are no transitive dependencies.

**ERD DATABASE:**

****