

Data base PostgreSQL screenshot: -

pgAdmin 4

FileObjectToolsHelp

Object Explorer

Servers (3)

PostgreSQL 12

Databases (2)

my_db

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

Login/Group Roles

Tablespaces

PostgreSQL 14

PostgreSQL 15

DashboardPropertiesSQLStatisticsDependenciesDependentsProcesses

my_db/postgres@PostgreSQL 12*

my_db/postgres@PostgreSQL 12

No limit

Data OutputMessagesNotifications

	truck_id [PK] integer	comment character varying (255)	date character varying (255)	loading_point character varying (255)	no_of_truck integer	product_type character varying (255)	shipper_id integer	truck_type character varying (255)
1	402	this is my first delivery	01-03-2023	Osmanabad	2	chemicals	123	canter

Total rows: 1 of 1Query complete 00:00:00.449Ln 1, Col 27

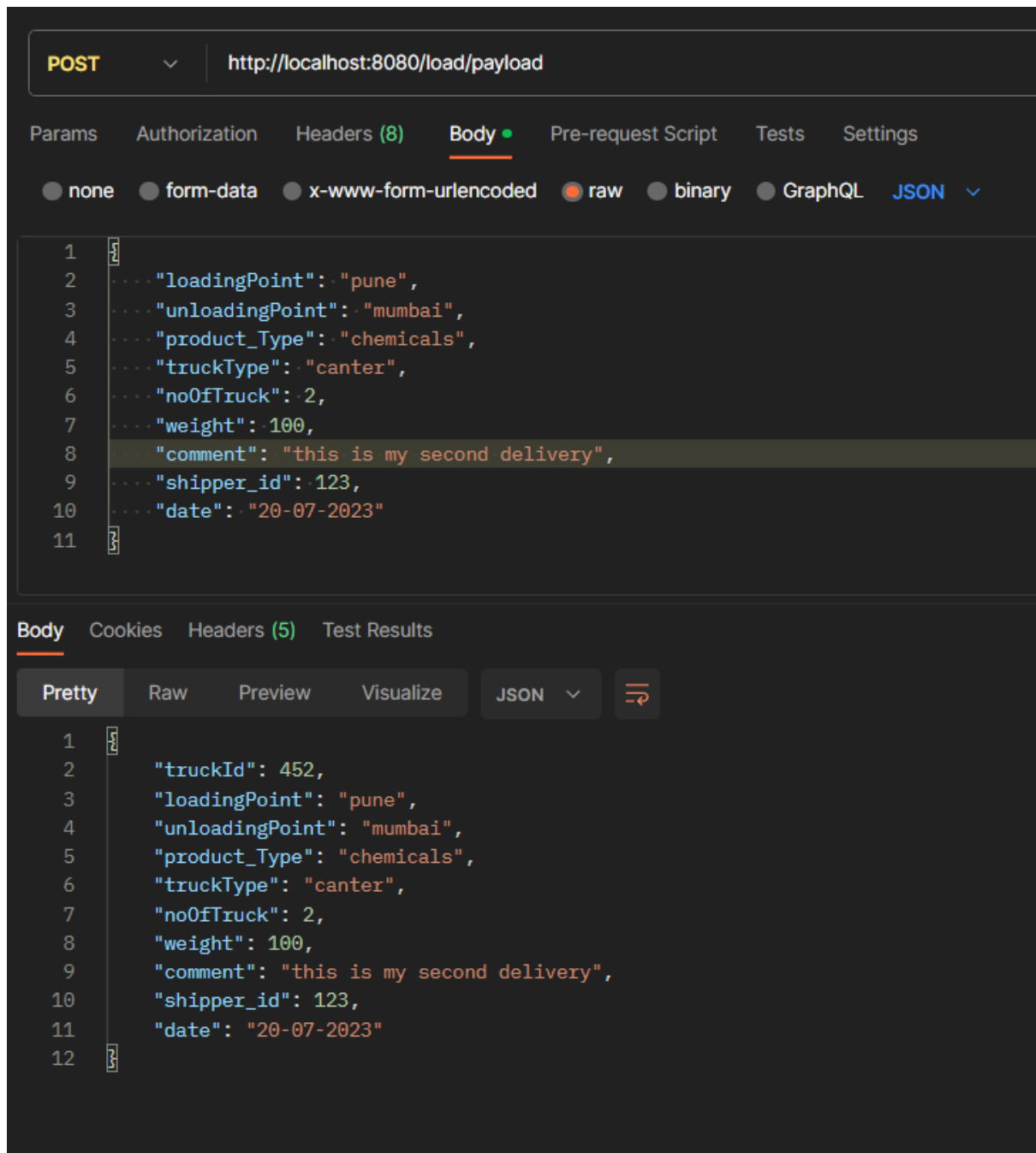
The screenshot shows the pgAdmin 4 interface. The left sidebar displays the 'Object Explorer' with a tree view of the database structure. The main pane shows the 'Data Output' tab for a query executed on the 'my_db' database. The query result is a single row with the following data:

date character varying (255)	loading_point character varying (255)	no_of_truck integer	product_type character varying (255)	shipper_id integer	truck_type character varying (255)	unloading_point character varying (255)	weight integer
1	01-03-2023	Osmanabad	2	chemicals	123	canter	100

Total rows: 1 of 1 Query complete 00:00:00.449 Ln 1, Col 27

POSTMAN SCREENSHOT:-

1.CREATE:-



2.GET ALL:

The screenshot displays a REST client interface with a GET request to `http://localhost:8080/load/all`. The request body is a JSON object. The response is also in JSON format, showing a list of two delivery records.

Request:

```
1 {
2   ... "loadingPoint": "pune",
3   ... "unloadingPoint": "mumbai",
4   ... "product_Type": "chemicals",
5   ... "truckType": "canter",
6   ... "noOfTruck": 2,
7   ... "weight": 100,
8   ... "comment": "this is my second delivery",
9   ... "shipper_id": 123,
10  ... "date": "20-07-2023"
11 }
```

Response:

```
1 {
2   {
3     "truckId": 402,
4     "loadingPoint": "Osmanabad ",
5     "unloadingPoint": "jaipur",
6     "product_Type": "chemicals",
7     "truckType": "canter",
8     "noOfTruck": 2,
9     "weight": 100,
10    "comment": "this is my first delivery",
11    "shipper_id": 123,
12    "date": "01-03-2023"
13  },
14  {
15    "truckId": 452,
16    "loadingPoint": "pune",
17    "unloadingPoint": "mumbai",
18    "product_Type": "chemicals",
19    "truckType": "canter",
20    "noOfTruck": 2,
21    "weight": 100,
22    "comment": "this is my second delivery",
23    "shipper id": 123,
```

3.GET BY ID :-

The screenshot displays a REST client interface with a GET request to `http://localhost:8080/load/402`. The request body is a JSON object with the following fields: `loadingPoint` (pune), `unloadingPoint` (mumbai), `product_Type` (chemicals), `truckType` (canter), `noOfTruck` (2), `weight` (100), `comment` (this is my second delivery), `shipper_id` (123), and `date` (20-07-2023). The response body is also a JSON object with fields: `truckId` (402), `loadingPoint` (Osmanabad), `unloadingPoint` (jaipur), `product_Type` (chemicals), `truckType` (canter), `noOfTruck` (2), `weight` (100), `comment` (this is my first delivery), `shipper_id` (123), and `date` (01-03-2023).

Request:

```
GET http://localhost:8080/load/402
```

Body:

```
{
  "loadingPoint": "pune",
  "unloadingPoint": "mumbai",
  "product_Type": "chemicals",
  "truckType": "canter",
  "noOfTruck": 2,
  "weight": 100,
  "comment": "this is my second delivery",
  "shipper_id": 123,
  "date": "20-07-2023"
}
```

Response:

```
{
  "truckId": 402,
  "loadingPoint": "Osmanabad ",
  "unloadingPoint": "jaipur",
  "product_Type": "chemicals",
  "truckType": "canter",
  "noOfTruck": 2,
  "weight": 100,
  "comment": "this is my first delivery",
  "shipper_id": 123,
  "date": "01-03-2023"
}
```

4.DELETE:-

The screenshot displays a REST client interface with a DELETE request to `http://localhost:8080/load/452`. The 'Body' tab is selected, showing a JSON payload. Below the request, the 'Body' tab of the response section is also selected, showing a single line of text.

Request Details:

- Method: DELETE
- URL: `http://localhost:8080/load/452`
- Body Type: raw
- Body Content:

```
1  {  
2    "loadingPoint": "pune",  
3    "unloadingPoint": "mumbai",  
4    "product_Type": "chemicals",  
5    "truckType": "canter",  
6    "noOfTruck": 2,  
7    "weight": 100,  
8    "comment": "this is my second delivery",  
9    "shipper_id": 123,  
10   "date": "20-07-2023"  
11 }
```

Response Details:

- Tab: Body
- Content:

```
1
```

5.UPDATE:-

The screenshot displays a REST client interface with a PUT request to `http://localhost:8080/load/402/surat`. The request body is a JSON object with the following fields: `loadingPoint` (pune), `unloadingPoint` (mumbai), `product_Type` (chemicals), `truckType` (canter), `noOfTruck` (2), `weight` (100), `comment` (this is my second delivery), `shipper_id` (123), and `date` (20-07-2023). The response is also a JSON object, formatted as `JSON` in the Pretty view, with fields: `truckId` (402), `loadingPoint` (surat), `unloadingPoint` (jaipur), `product_Type` (chemicals), `truckType` (canter), `noOfTruck` (2), `weight` (100), `comment` (this is my first delivery), `shipper_id` (123), and `date` (01-03-2023).

```
PUT http://localhost:8080/load/402/surat

{
  "loadingPoint": "pune",
  "unloadingPoint": "mumbai",
  "product_Type": "chemicals",
  "truckType": "canter",
  "noOfTruck": 2,
  "weight": 100,
  "comment": "this is my second delivery",
  "shipper_id": 123,
  "date": "20-07-2023"
}
```

```
{
  "truckId": 402,
  "loadingPoint": "surat",
  "unloadingPoint": "jaipur",
  "product_Type": "chemicals",
  "truckType": "canter",
  "noOfTruck": 2,
  "weight": 100,
  "comment": "this is my first delivery",
  "shipper_id": 123,
  "date": "01-03-2023"
}
```

