

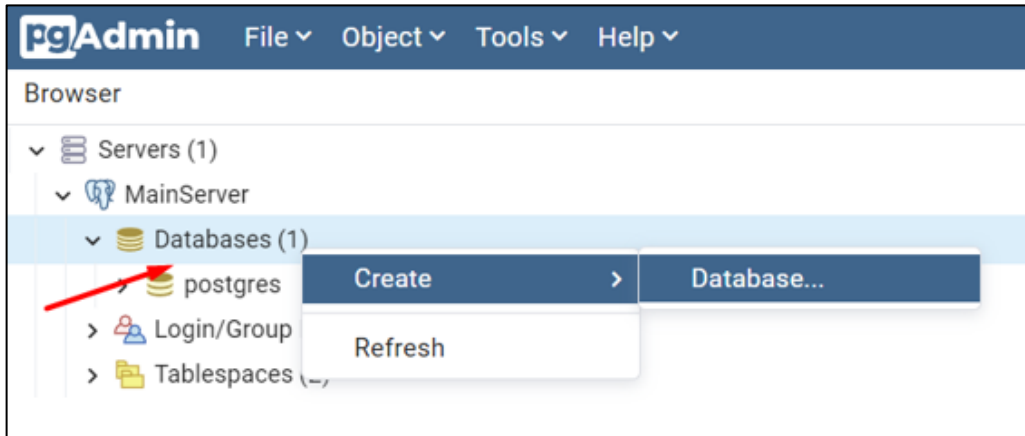
Lab: Data Types and Table Basics

This document defines the **lab exercise assignments** for the [PostgreSQL course @ Software University](#).

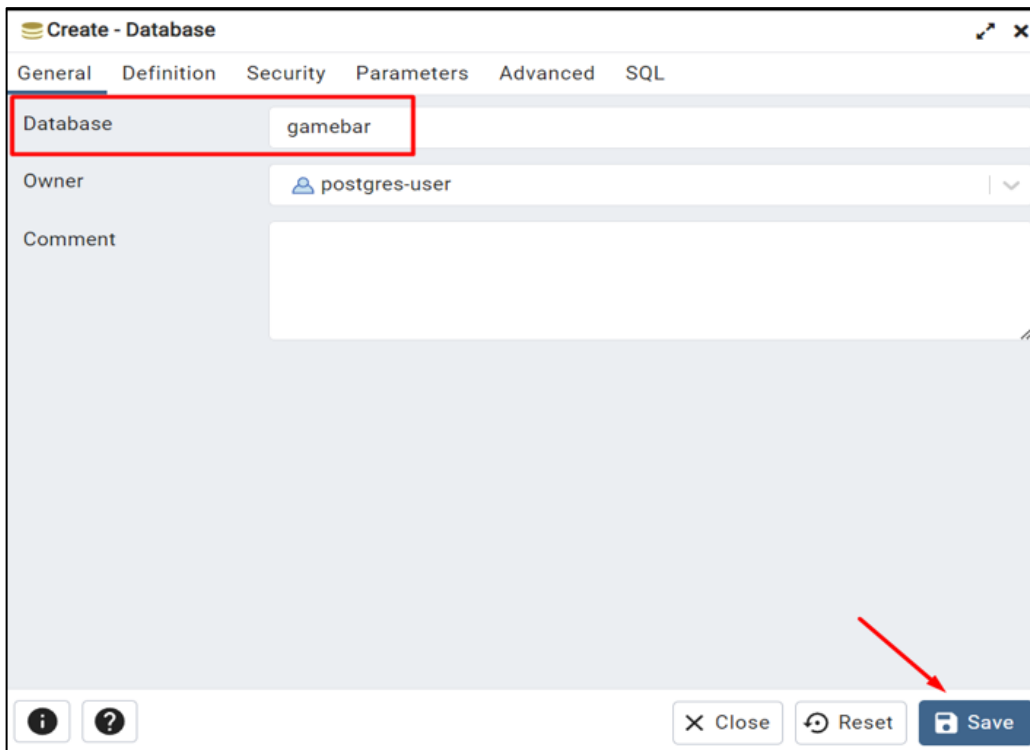
Part I - Simple Database Operations Using PostgreSQL pgAdmin

1. Create New Database

First, let us create an **empty database** called "**gamebar**". Right-click on the databases field; choose "**Create**" -> "**Database**":

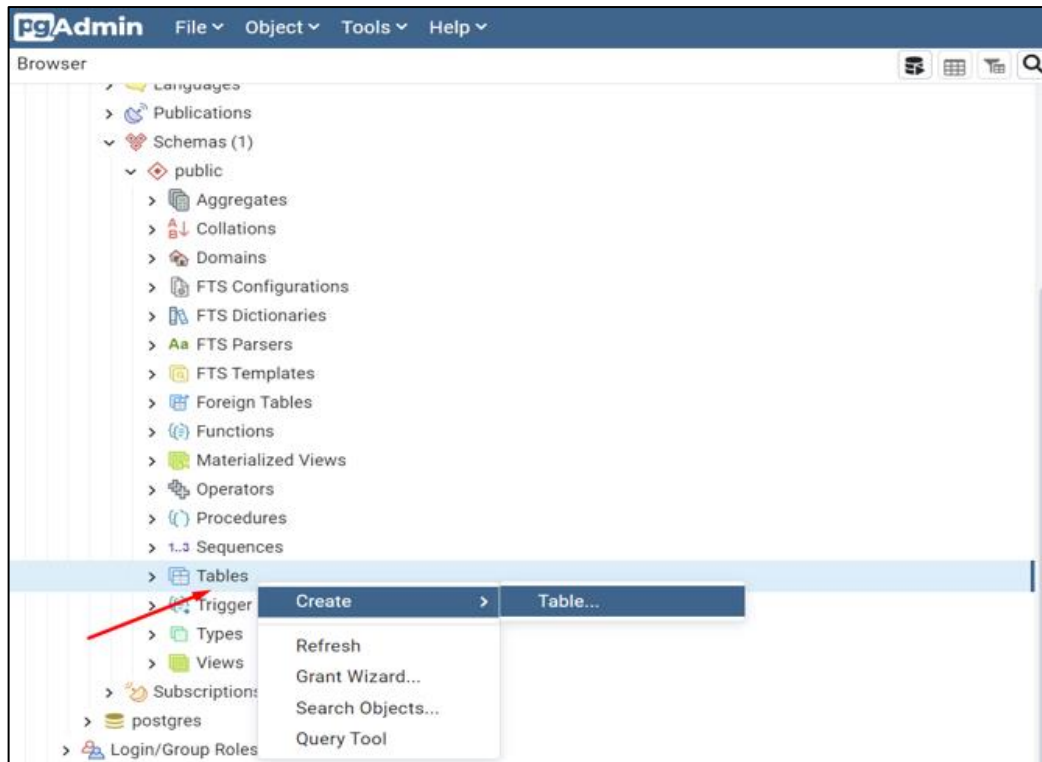


A new "**Create Database**" window will appear. In the "**Database**" field type our new database's name - "**gamebar**".



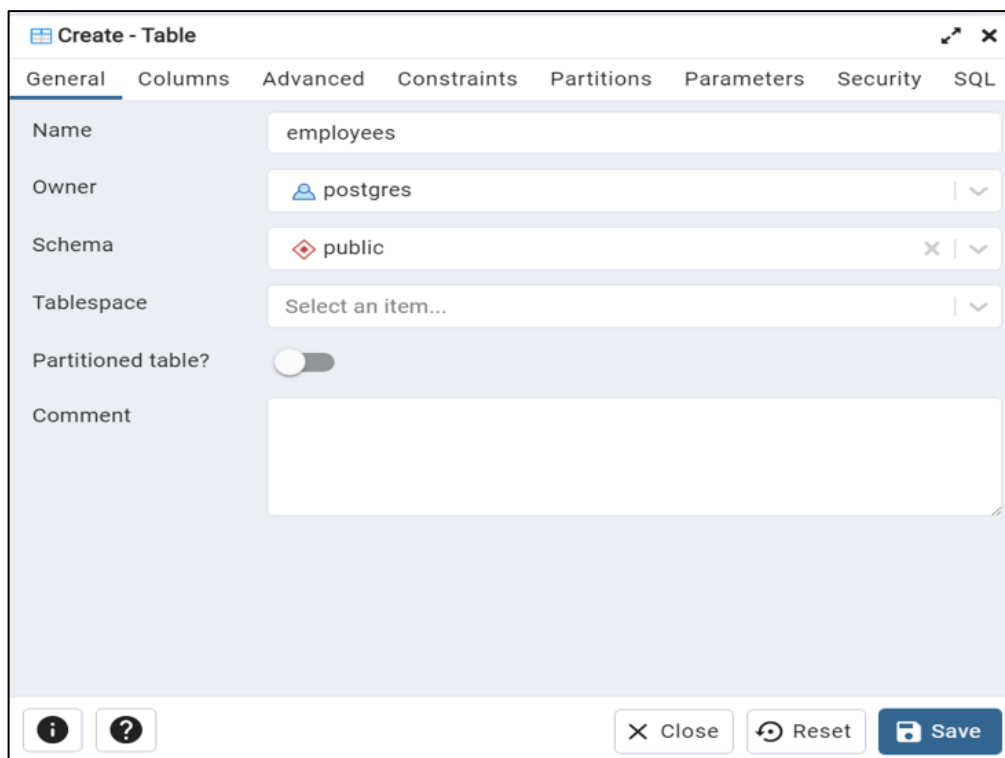
2. Create New Table

Open the newly created "**gamebar**" database, then open "Schemas" -> "public". Right-click the "**Tables**" and select "**Create**" -> "**Table**".



The table creation tab will appear. In the "**Table Name**" field type the name of your new table – "**employees**". From the "**Columns**" tab you can start creating your table fields.

First, create an "**id**" field. Set the data type to **serial**, **Primary key(PK)**, and **Not NULL**. Create one more field – "**name**" with data type **character varying**.



Click **Save** button in order to finish the table creation.

Create - Table

General **Columns** Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id	serial			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	name	character varying			<input type="checkbox"/>	<input type="checkbox"/>	

Close Reset Save

3. View and Edit Tables

Extensions Foreign Data Wrappers Languages Publications Schemas (1) public Aggregates Collations Domains FTS Configuration FTS Dictionary FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (1) **employees**

Count Rows Create Delete/Drop Refresh... Restore... Backup... Drop Cascade Import/Export Data... Reset Statistics Maintenance... Scripts Truncate View/Edit Data All Rows First 100 Rows Last 100 Rows Filtered Rows... Search Objects... PSQL Tool Query Tool Properties...

Dashboard Properties SQL Statistics Dependencies public.employees/gamebar/postgres

public.employees/gamebar/postgres@PostgreSQL 14

Query Query History

```
1 SELECT * FROM public.employees
2 ORDER BY id ASC
```

Data output Message Peter

id [PK] integer

1+ [default]

X Cancel OK

Now you can modify your table and add **2** more columns to it (adding columns to tables was already described above):

- **salary** - numeric, specified to the second decimal place, and has 10 digits in total
- **devices_number** - integer

	salary	numeric	10	2	<input type="checkbox"/>	<input type="checkbox"/>	
	devices_number	integer			<input type="checkbox"/>	<input type="checkbox"/>	

4. Create Tables "departments" and "issues"

Similar to "employees" create **2** more tables.

Table "departments":

- **id** - serial, primary key, Not NULL;
- **name** - character varying, max length 50;
- **code** - character, fixed length 3;
- **description** - text;

























Columns								+
		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
		id	serial v					
		name	character varying v	50				
		code	character v	3				
		description	text v					

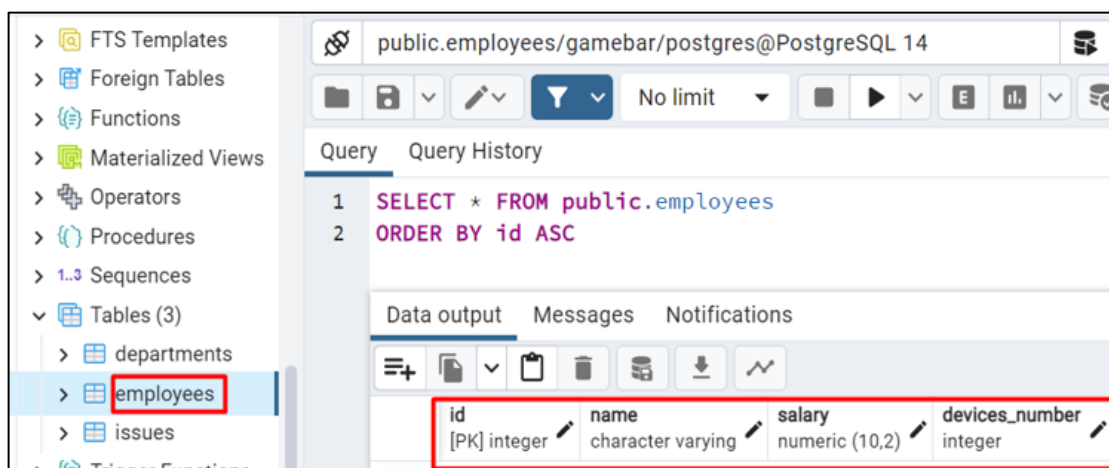
Table "issues":

- **id** – serial, primary key, unique;
- **description** – character varying, max length 150;
- **date** – date;
- **start** – timestamp without time zone

Columns								+
		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
		id	serial v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		description	character varying v	150		<input type="checkbox"/>	<input type="checkbox"/>	
		date	date v			<input type="checkbox"/>	<input type="checkbox"/>	
		start	timestamp without time... v			<input type="checkbox"/>	<input type="checkbox"/>	

5. Insert Data in Tables

Now we can start adding some records to our newly created tables. First, select the "employees" table to see all rows:



The screenshot shows a database management interface. On the left, a sidebar lists various database objects, with 'employees' highlighted under 'Tables (3)'. The main area displays the 'public.employees/gamebar/postgres@PostgreSQL 14' database. A query is entered: `SELECT * FROM public.employees ORDER BY id ASC`. Below the query, the 'Data output' tab shows the table structure with columns: **id** [PK] integer, name character varying, salary numeric (10,2), and devices_number integer. The 'employees' table name is highlighted with a red box in the sidebar, and the table structure details are also highlighted with a red box.

Select the  button to add a new record.

Fill in the fields with values by double-clicking inside the desired field. Create 3 records in each table. **Save** data by clicking on the **database icon**.

	id [PK] integer	name character varying	salary numeric (10,2)	devices_number integer
1+	[default]	[null]	[null]	[null]

6. Editing Data

Data in tables can easily be edited with the GUI. Now that we've populated our tables with values, we can edit them by **double-clicking** on the **value** field. **Save** data by clicking on the **database icon**.

Data output

Messages

Notifications

	id [PK] integer	name character varying	salary numeric (10,2)	devices_number integer
1	4	Mary Lu	2830.22	7
2	3	Steven Doe	2360.00	2
3	1	Peter Mayer	1250.33	5

7. Deleting Data

Data deletion is easy too. We can just select the row we are about to delete and click on the **bin icon**. Do not forget to **save** your changes.

Delete **all rows** from table **employees**.

Data output

Messages

Notifications

≡ +

📄

▼

📋

🗑

🗄

⬇

📈

	id [PK] integer	name character varying	salary numeric (10,2)	devices_number integer
1	4	Mary Lu	2830.22	7
2	3	Steven Doe	2360.00	2
3	1	Peter Mayer	1250.33	5

8. Modifying Columns and Adding Constraints

Select the table "**employees**". You can modify the column named "**name**", so you change it to **first_name**, then **add** a new column **last_name**. Add a new column **hiring_date** as well.

Set all constraints as follows:

- **first_name** - character varying, max length 30, Not NULL;
- **last_name** - character varying, max length 50, Not NULL;
- **hiring_date** - date, default '2023-01-01';

employees

General **Columns** Advanced **Constraints** Parameters Security SQL

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('en
name	character varying			<input type="checkbox"/>	<input type="checkbox"/>	
salary	numeric	10	2	<input type="checkbox"/>	<input type="checkbox"/>	
devices_number	integer			<input type="checkbox"/>	<input type="checkbox"/>	

General Definition **Constraints** Variables Security

Name **name**

Comment

Your **modified** table **employees** should look like the following now:

employees

General **Columns** Advanced Constraints Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('emp
first_name	character varying	30		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
salary	numeric	10	2	<input type="checkbox"/>	<input type="checkbox"/>	
devices_number	integer			<input type="checkbox"/>	<input type="checkbox"/>	
last_name	character varying	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
hiring_date	date			<input type="checkbox"/>	<input type="checkbox"/>	'2023-01-01'

Close Reset **Save**

public.employees/gamebar/postgres@PostgreSQL 14

No limit

Query Query History

```
1 SELECT * FROM public.employees
2 ORDER BY id ASC
```

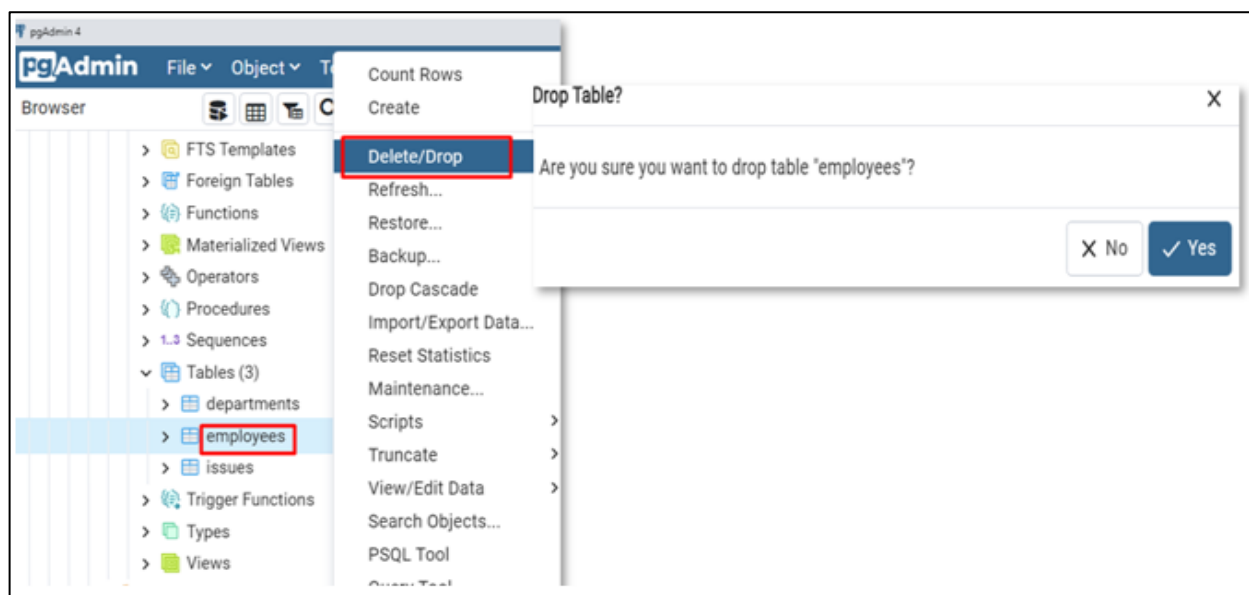
Data output Messages Notifications

id	first_name	salary	devices_number	last_name	hiring_date
[PK] integer	character varying (30)	numeric (10,2)	integer	character varying (50)	date

9. Dropping Tables

We can delete the whole table, by selecting the one we want to delete, right-clicking, and selecting "Delete/Drop".
You cannot undo this action.

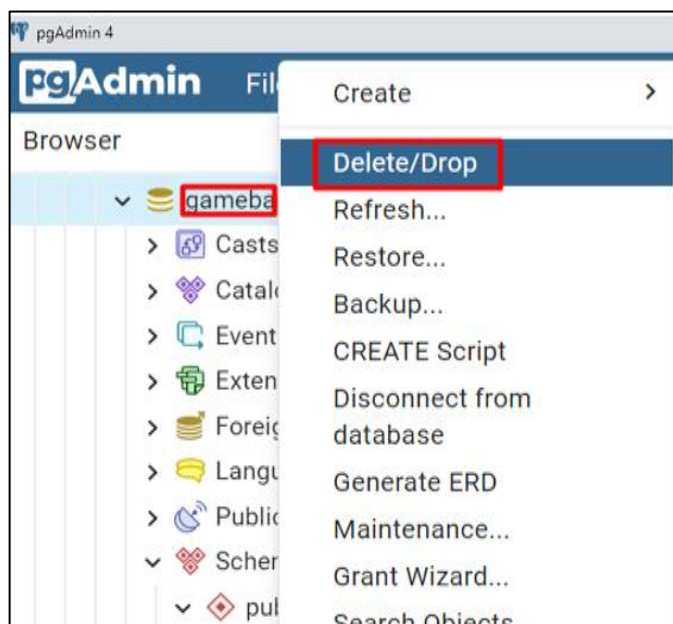
Now drop your 3 tables from your database.



10. Dropping the Database

As the table dropping, we can drop the database too. **This action cannot be undone too.**

Right-click the database you want to drop and select "Delete/Drop".

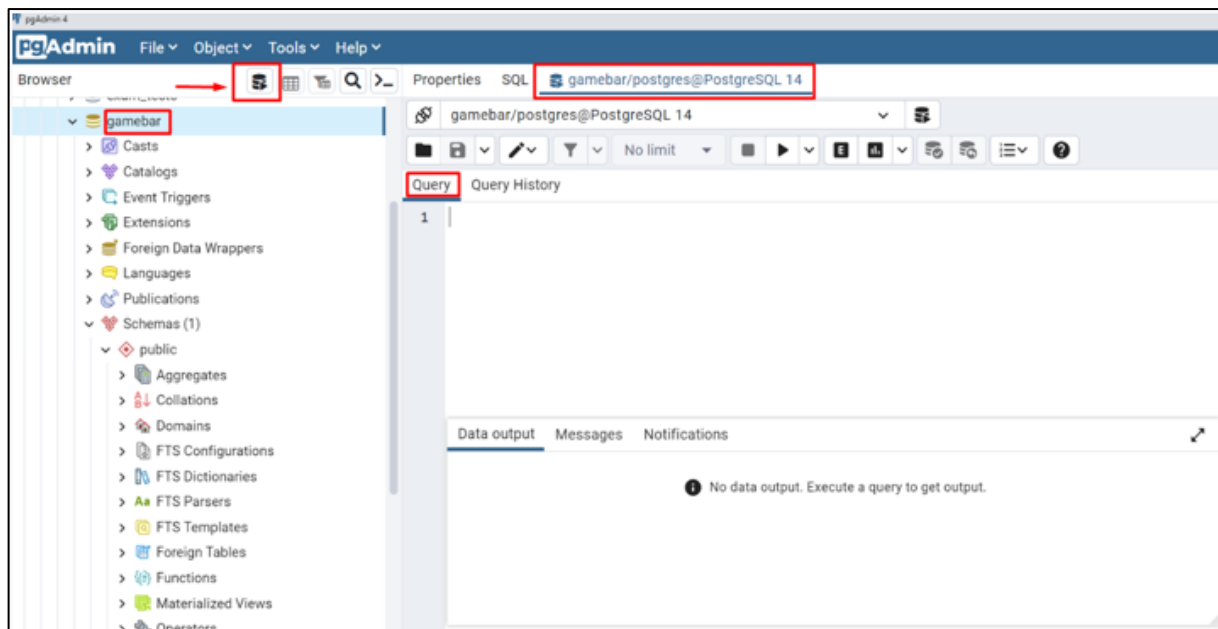


Part II - Simple Database Operations Using Queries

Now we are going to do the same steps from Part I, using simple SQL queries.

Exercises from this section should be submitted to [Judge Contest](#) (tasks 1, 3 to 7).

Queries are written in the "Query" tab.



Descriptions for Exercises in Judge System

1. Create New Database

Write a query that will create the "gamebar" database. Open its **Query Tool**

2. Create Tables

Table "employees":

- **id** – serial, primary key, Not NULL;
- **first_name** – character varying, max length 30;
- **last_name** – character varying, max length 50;
- **hiring_date** – date, default '2023-01-01';
- **salary** – numeric, specified to the second decimal place, and has 10 digits in total;
- **devices_number** – integer;

Create the "departments" and "issues" tables analogically:

Table "departments":

- **id** – serial, primary key, Not NULL;
- **name** – character varying, max length 50;
- **code** – character, fixed length 3;
- **description** – text;

Table "issues":

- **id** – serial, primary key, unique;
- **description** – character varying, max length 150;
- **date** – date;
- **start** – timestamp without time zone;

3. Insert Data in Tables

**This task is not included in the Judge Contest.*

Populate the "employees" table with 3 test values using pgAdmin UI.

4. Alter Tables

Altering the tables is done via the "ALTER TABLE" clause. Add a new column – "middle_name", "VARCHAR(50)" to the "employees" table.

5. Add Constraints

In the table "employees", set the salary column as **Not NULL** with a **default value of 0**. Set the hiring date column as **Not NULL** too.

6. Modify Columns

Change the property "VARCHAR(50)" to "VARCHAR(100)" for the middle_name column in "employees" table.

7. Truncate Tables

Truncate table "issues".

8. Drop Tables

Drop table "departments".