

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Практическая работа №1.2

Установка и первоначальная настройка субд postgresql.

Выполнил:

Доценников Никита Андреевич

Группа: К3221

Проверила:

Татьяна Евгеньевна Войтюк

Санкт-Петербург

2025

Цель работы:

Изучение принципов проектирования и администрирования реляционных баз данных в среде postgresql с использованием графического интерфейса pgadmin и языка sql. Освоение создания, изменения и связи таблиц, настройки ограничений целостности, индексов и построения ER-диаграмм. Формирование практических навыков работы с объектами базы данных, добавления и проверки данных, а также документирования структуры базы данных.

Задачи, решаемые при выполнении работы.

- Освоить создание таблиц в базе данных с использованием графического интерфейса pgadmin и sql запросов.
- Научиться задавать ключевые параметры таблиц: первичные ключи, автоинкрементные поля, значения по умолчанию и ограничения `NOT NULL`.
- Изучить применение команды `ALTER TABLE` для изменения структуры таблицы без потери данных.
- Создать связи между таблицами (внешние ключи) как через интерфейс pgadmin, так и с помощью sql кода.
- Научиться создавать индексы для оптимизации поиска и сортировки данных.
- Освоить построение и редактирование er диаграмм для визуализации структуры базы данных.
- Научиться добавлять, изменять и проверять данные в таблицах с помощью интерфейса pgadmin и dml запросов.
- Изучить назначение и реализацию ограничений целостности.
- Научиться очищать таблицы с помощью команды `TRUNCATE` и выполнять внешние sql скрипты для заполнения и модификации базы.
- Сформировать целостное представление о создании и сопровождении базы данных на примере модели hr.

Исходные данные.

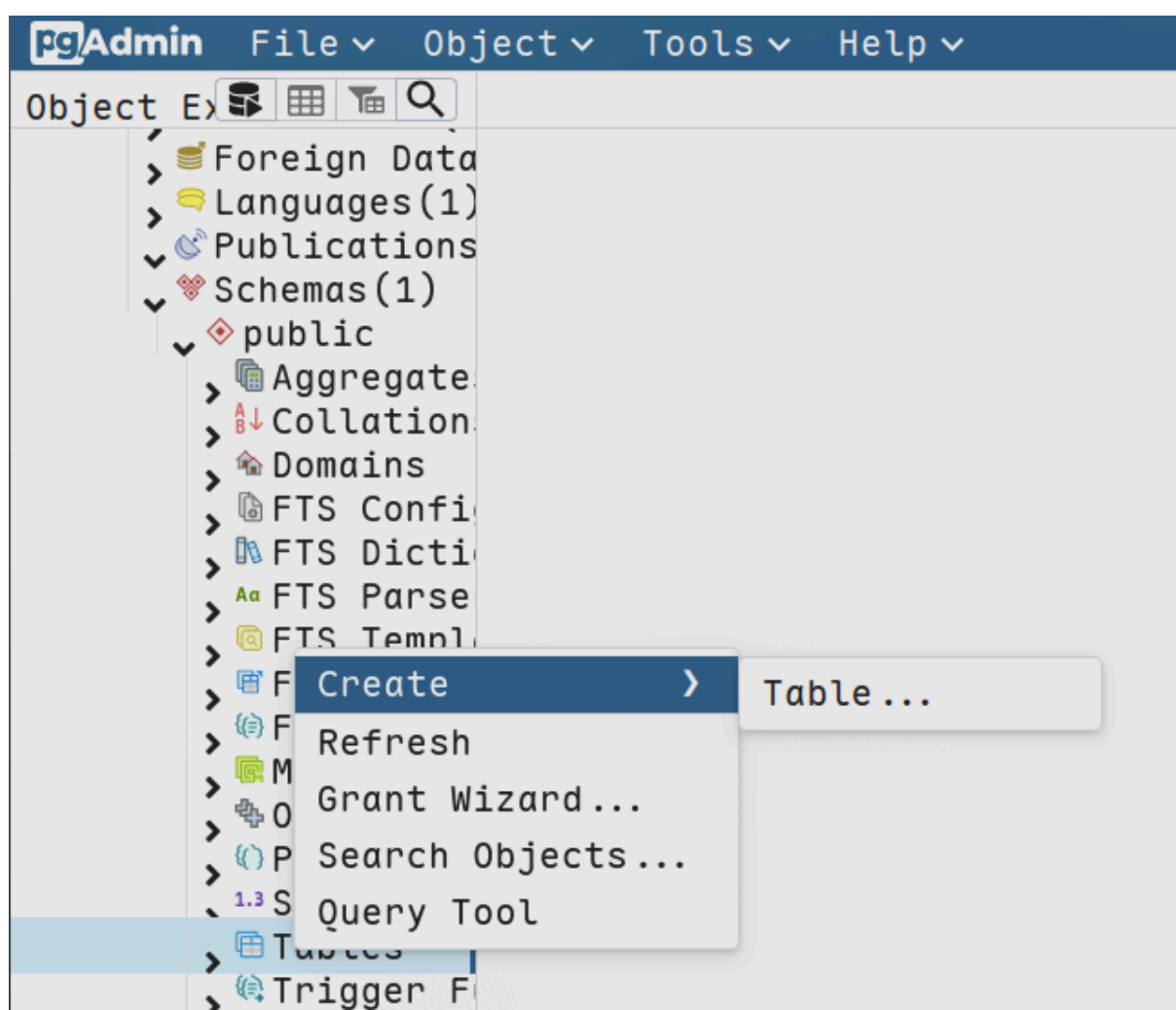
- субд postgresql и графическая оболочка pgadmin.
- бд hr, содержащая схемы и объекты для работы.
- схема employeesDepartments, созданная на предыдущем этапе.
- sql-скрипты, предоставленные в методических материалах:

- набор заданий, предполагающих создание таблиц employees, departments, locations и других таблиц базы данных hr.
- примерные структуры таблиц и параметры столбцов.
- методические указания по выполнению лабораторных работ и пошаговые инструкции с примерами sql команд и интерфейсных действий.

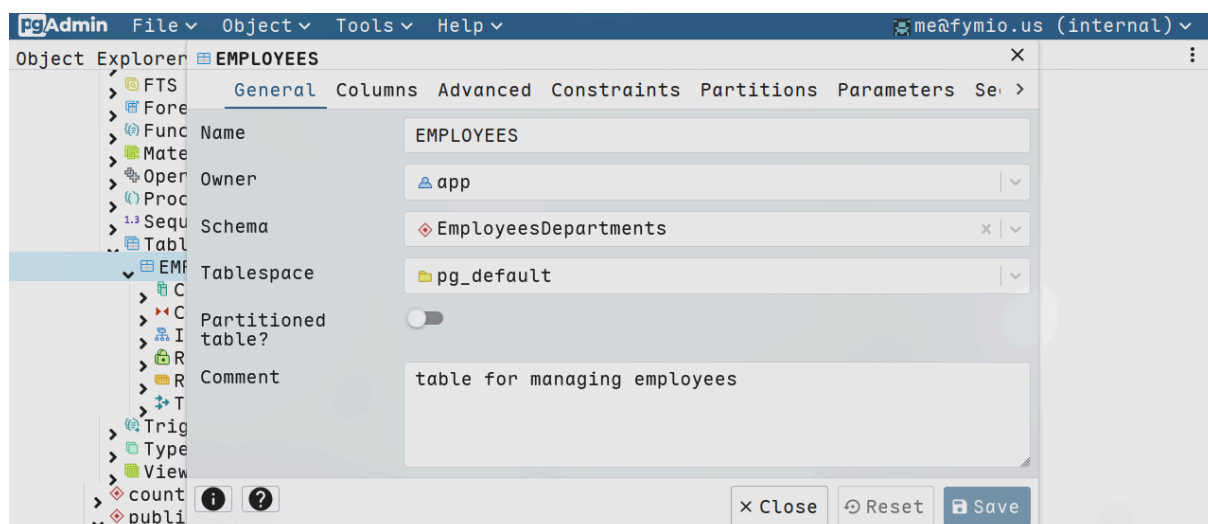
Выполнение работы.

Задание 1. Создание таблицы в графической среде pgadmin.

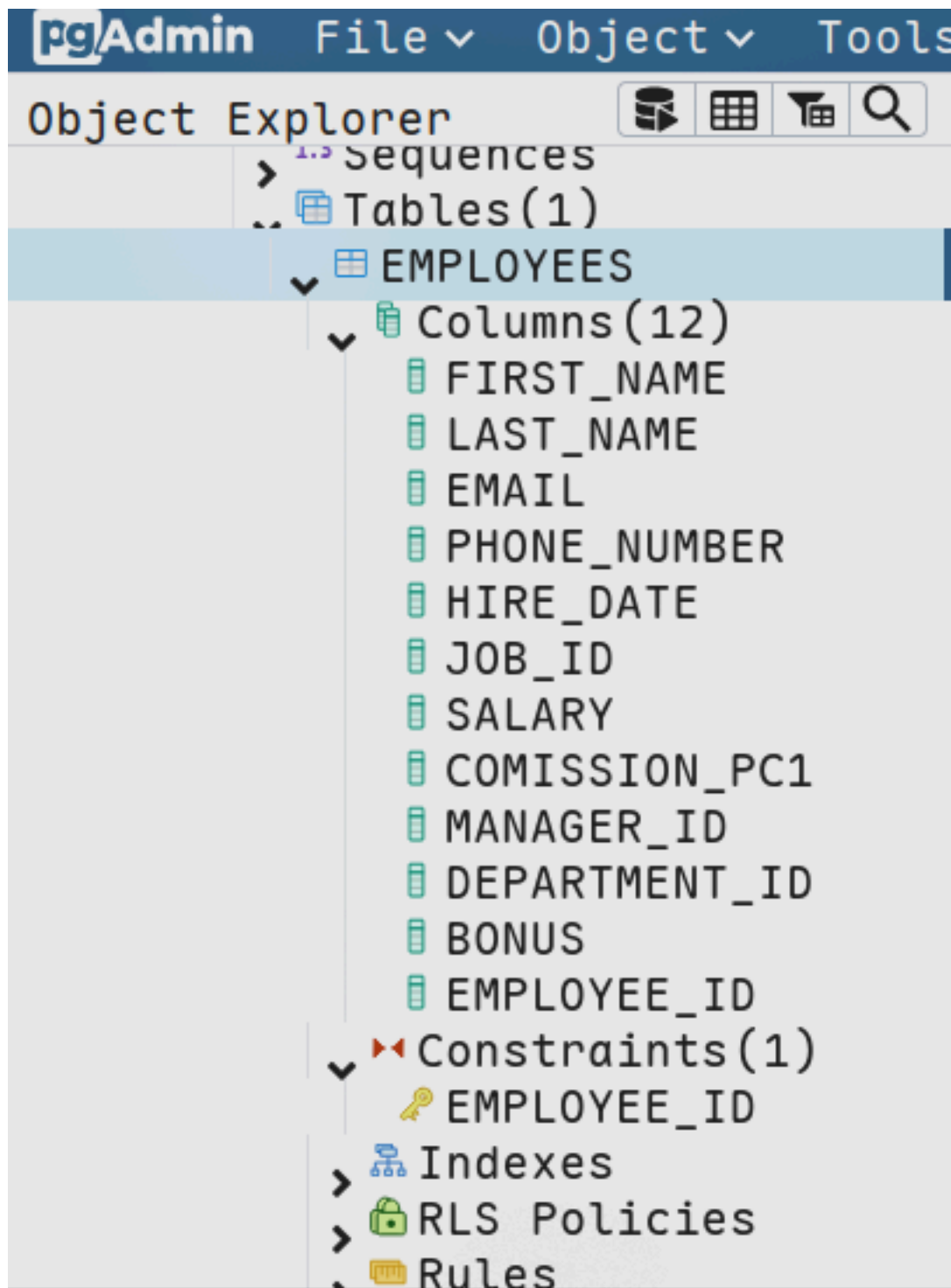
Я раскрыл базу hr до узла tables и выбрал опцию create:



И создал таблицу EMPLOYEES :



Затем я заполнил вкладку `columns` как показано на рисунке в условии:



Для поля `employee_id`, я выставил следующие параметры:

EMPLOYEE_ID

GeneralDefinitionConstraints

Default

Not NULL?

☒

Type

NONE

IDENTITY

GENERATED

Identity

ALWAYS

Increment

1

Start

1

Minimum

i?

Close

Reset

Save

Для столбца `hire_date` я поставил значение по умолчанию `current_date`.

HIRE_DATE

×

General

Definition

Constraints

>

Default

current_date

Not NULL?

☒

Type

✓ NONE

IDENTITY

GENERATED

i

?

×

Close

↺

Reset

💾

Save

После этого, я сохранил таблицу.

Задание 2. Создание таблицы в Query Editor.

Я открыл query tool для hr и вписал туда следующий скрипт:

```
CREATE TABLE "EmployeesDepartments"."DEPARTMENTS" (  
    DEPARTMENT_ID integer GENERATED ALWAYS AS IDENTITY  
(INCREMENT 1 START 1) PRIMARY KEY NOT NULL,  
    DEPARTMENT_NAME character varying(30)[] NOT NULL,  
    MANAGER_ID bigint NULL,  
    LOCATION_ID integer NULL  
);
```

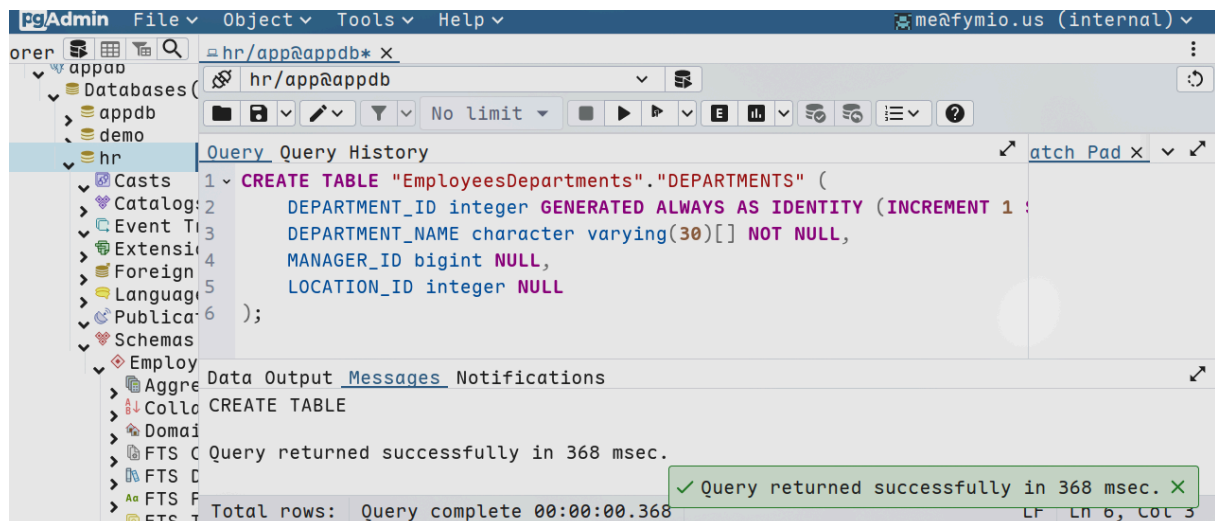
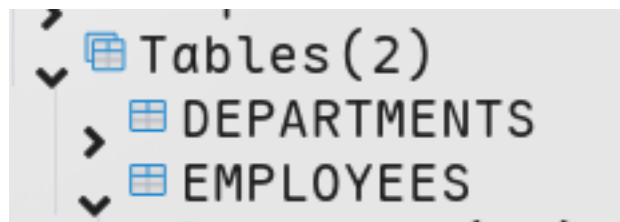
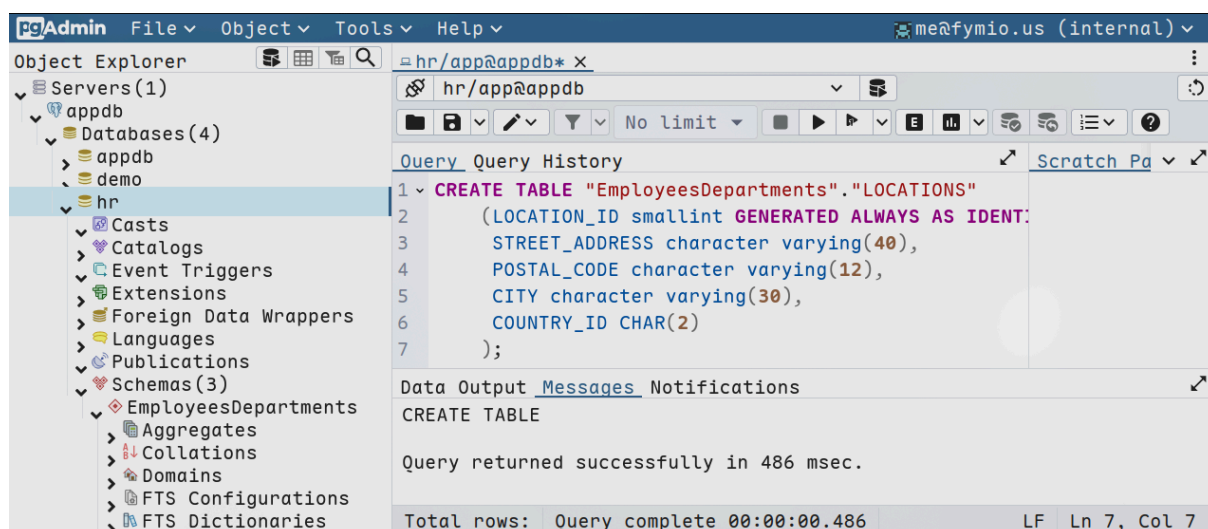


Таблица появилась в object explorer:

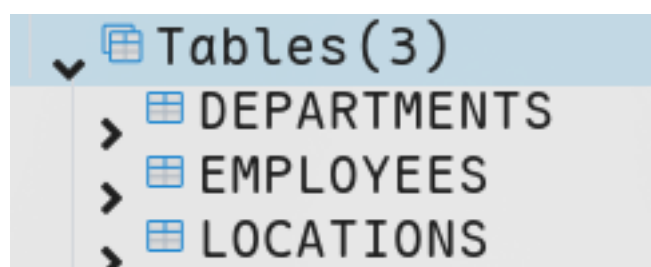


Затем я создал еще одну таблицу скриптом:

```
CREATE TABLE "EmployeesDepartments"."LOCATIONS"
  (LOCATION_ID smallint GENERATED ALWAYS AS IDENTITY NOT
  NULL,
  STREET_ADDRESS character varying(40),
  POSTAL_CODE character varying(12),
  CITY character varying(30),
  COUNTRY_ID CHAR(2)
  );
```

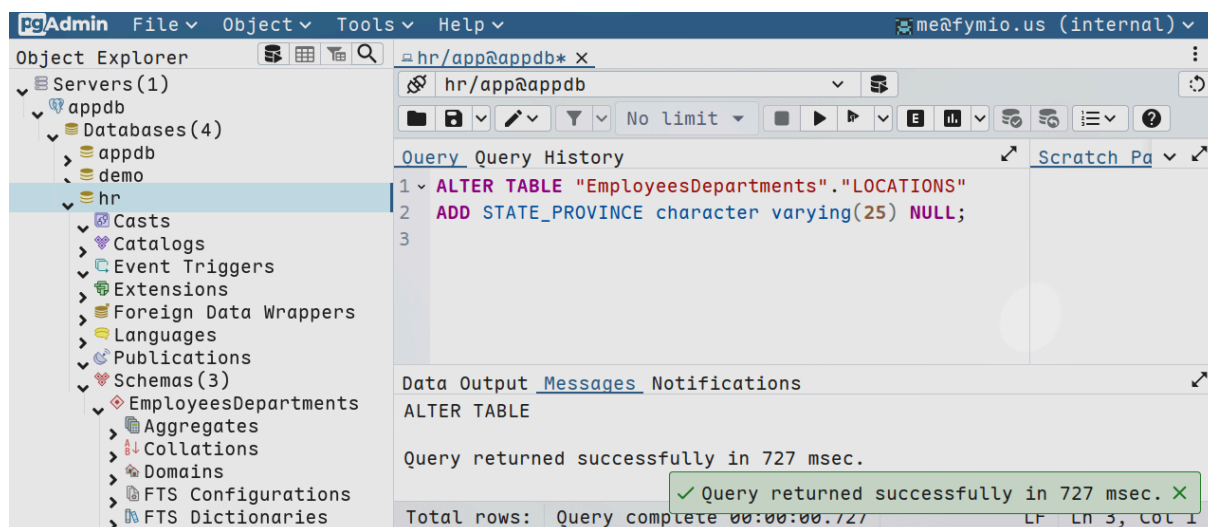
Как можно видеть, таблица была создана:



Задание 3. Изменение таблицы.

В query tool я ввел следующее:

```
ALTER TABLE "EmployeesDepartments"."LOCATIONS"
ADD STATE_PROVINCE character varying(25) NULL;
```



Изменил столбец CITY следующим кодом:

```
ALTER TABLE "EmployeesDepartments"."LOCATIONS"
ADD CONSTRAINT "PK_LokationId" PRIMARY KEY (LOCATION_ID);
```

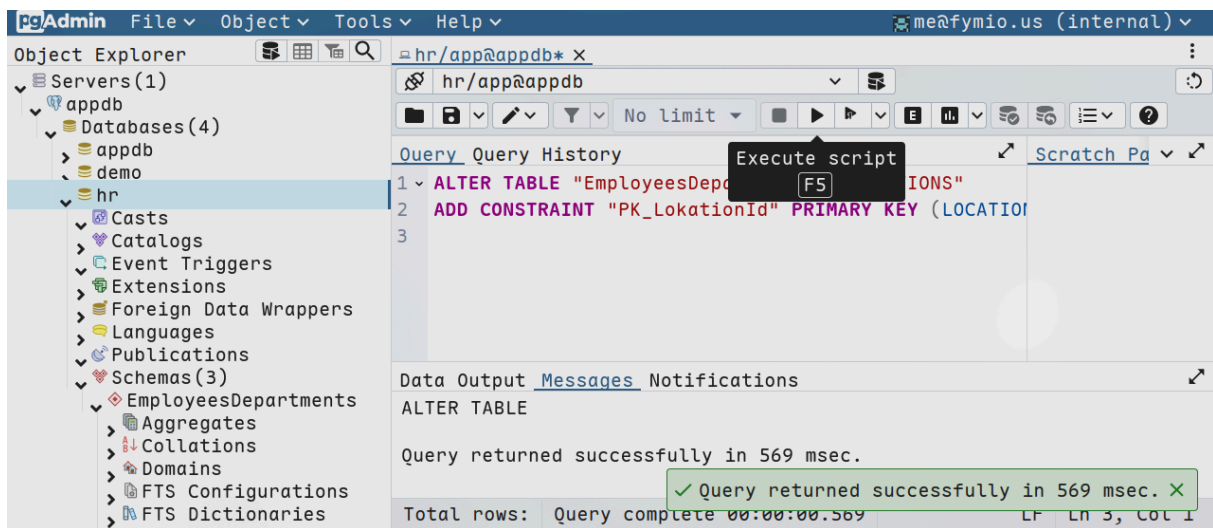
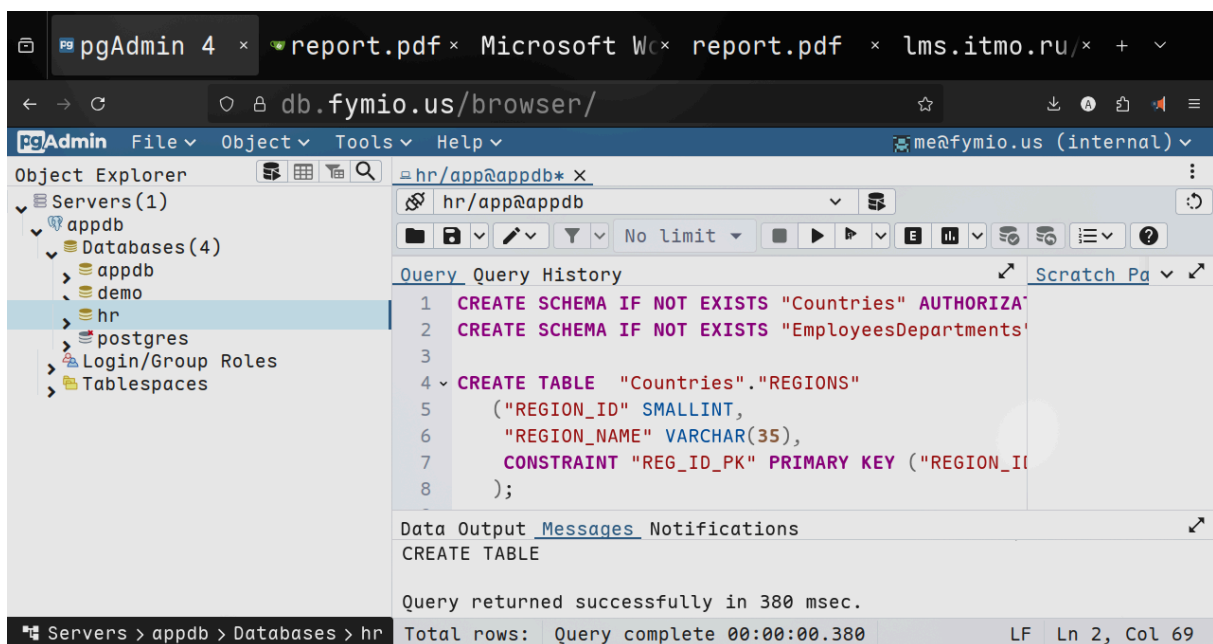
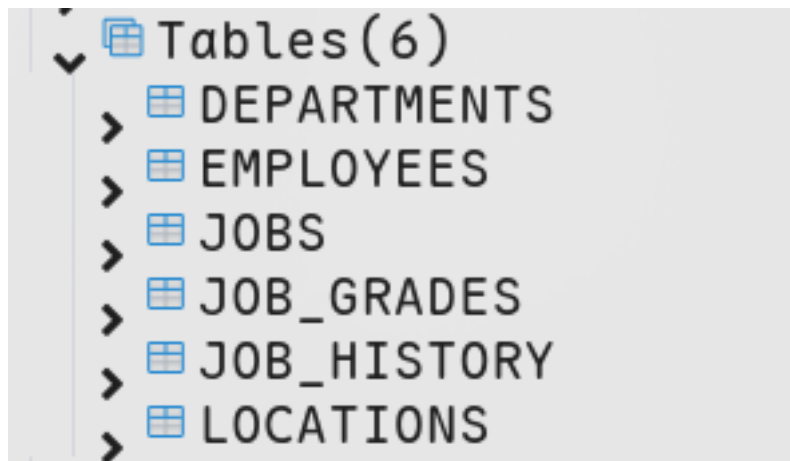


Таблица locations:

location_id	street_address	postal_code	city
[PK] smallint	character varying (40)	character varying (12)	character varying (30)

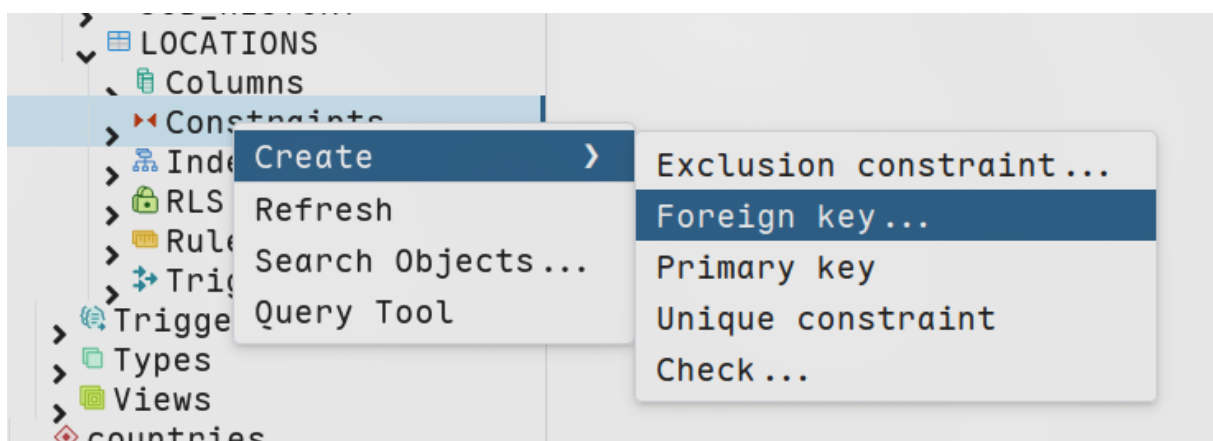
Затем я прописал текст скрипта script1_create.sql:





Задание 4. Создание отношения графическим интерфейсом pgadmin.

Я нажал ПКМ по таблице EmployeesDepartments.LOCATIONS и выбрал меню создания внешнего ключа:



Затем я ввел задание название отношения:

 **Create - Foreign key** ×

General Definition Columns Acti >

Name

Comment

 Please specify columns for Foreign key. ×

  × Close ↺ Reset 💾 Save

После, я добавил столбец кнопкой add:

Create - Foreign key
 ×

General
 Definition
 Columns
 Acti >

Columns

Local column

Select an item... | ▾

References

Select an item... | ▾

Referencing

Select an item... | ▾

Add

	Local	Referenced	Referenced...
	LOCATION_ID	COUNTRY_ID	"Countries".

i

?

×

Close

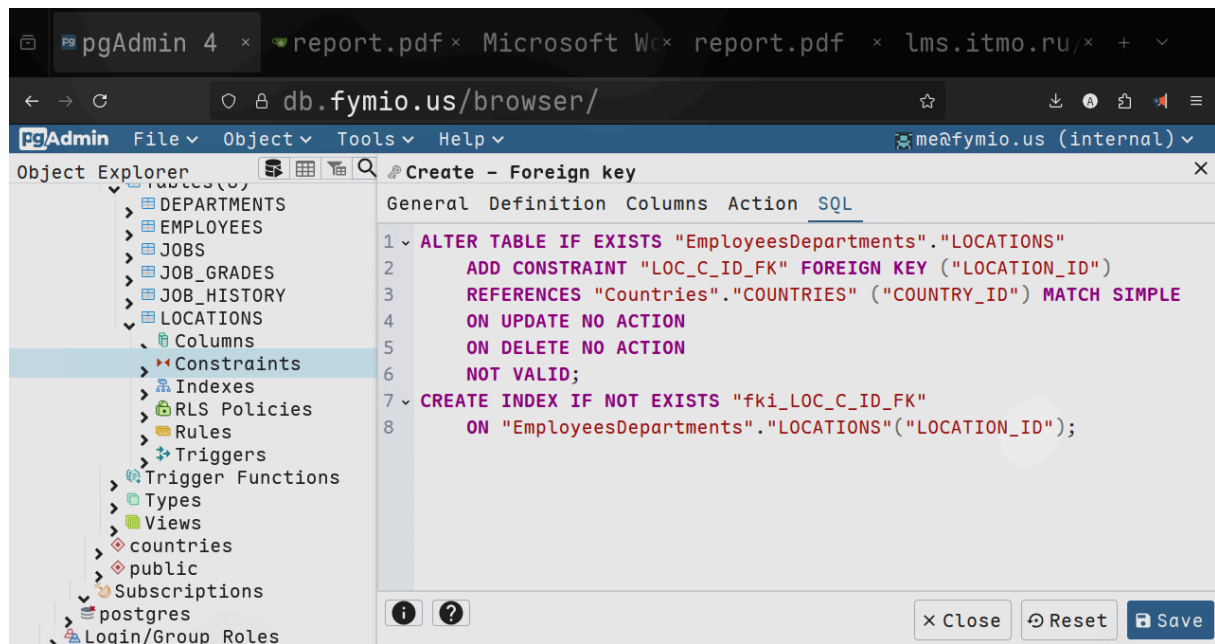
↺

Reset

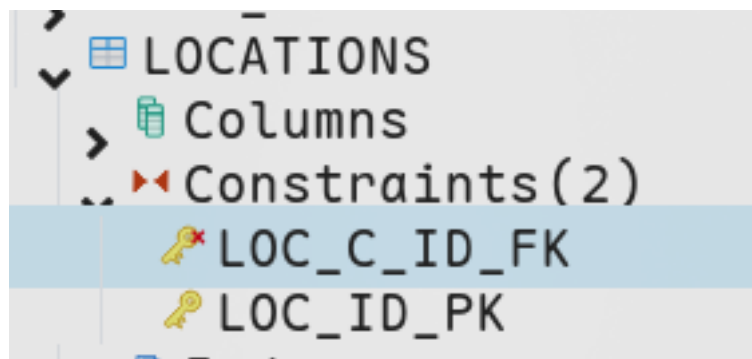
💾

Save

Вкладка sql:

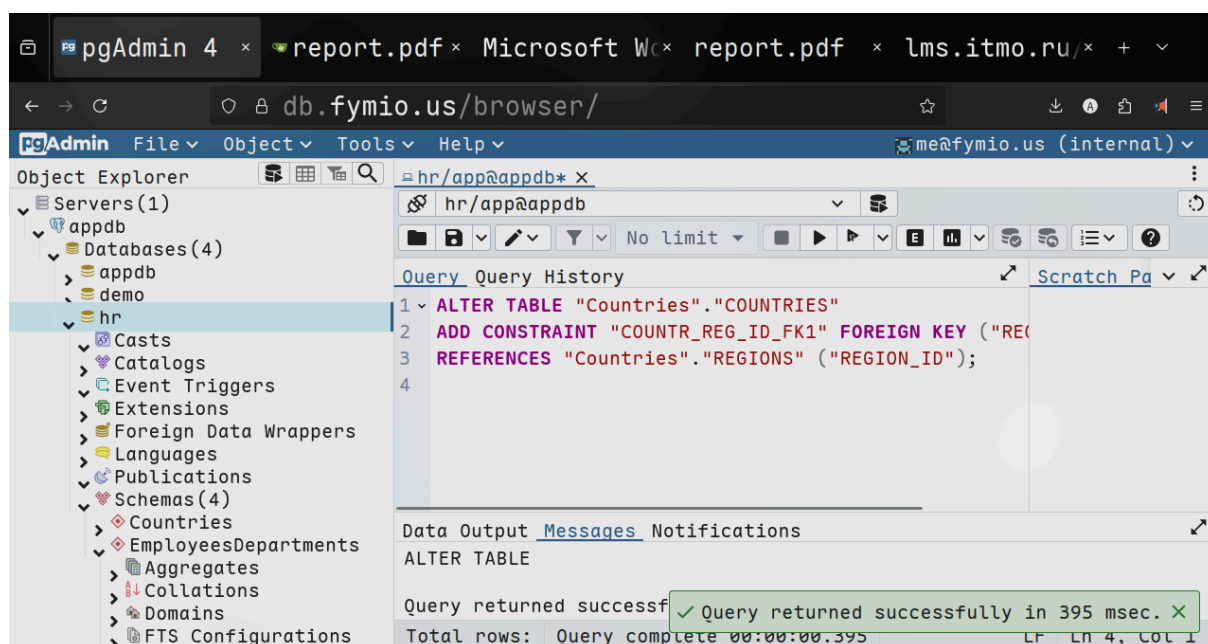


Ограничения locations:



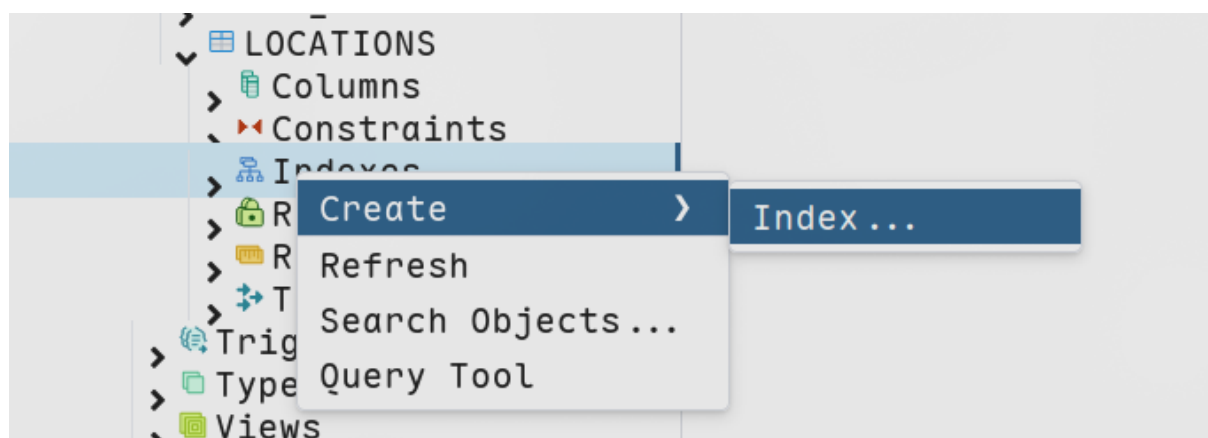
Затем я создал отношение с помощью ddl оператора alter table:

```
ALTER TABLE "Countries"."COUNTRIES"
ADD CONSTRAINT "COUNTR_REG_ID_FK1" FOREIGN KEY ("REGION_ID")
REFERENCES "Countries"."REGIONS" ("REGION_ID");
```



Задание 5. Создания индекса с помощью графического интерфейса pgadmin.

В таблице EmployeesDepartments.LOCATIONS, я выбрал раздел меню для создания индекса:



Я в диалоге создания индекса я передал следующие параметры:

Create - Index

General

Definition

Columns

SQL

Name

LOC_CITY_IX

Tablespace

Select an item...

Comment

You must specify at least one column/expression.

i

?

Close

Reset

Save

Create - Index

General

Definition

Columns

SQL

Access Method

btree

With

Fill factor

Gin pending list limit

This value is specified in kilobytes.

Pages per range

Number of table blocks that make up one block range for each entry of a BRIN index.

You must specify at least one column/expression.

i

?

Close

Reset

Save

Create - Index [X]

General Definition Columns SQL

Columns/Expressions

Is expression ☒

Column

Expression

Add

	Col/Exp	Operator ...	Sort order	NULLs	Collation
	C CITY	Select v	ASC v	LAST v	Select v

Include columns

[i] [?] [X Close] [↺ Reset] [Save]

sql:

Create - Index [X]

General Definition Columns SQL

```

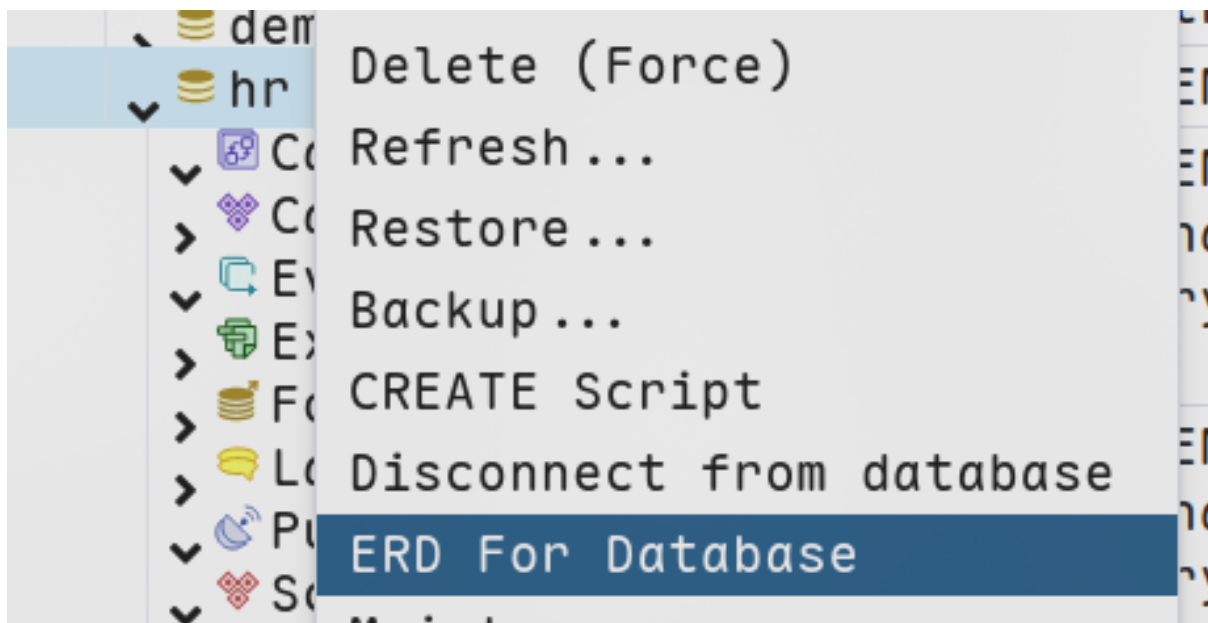
1 CREATE INDEX "LOC_CITY_IX"
2   ON "EmployeesDepartments"."LOCATIONS" USING btree
3   ("CITY")
4   WITH (deduplicate_items=True)
5 ;

```

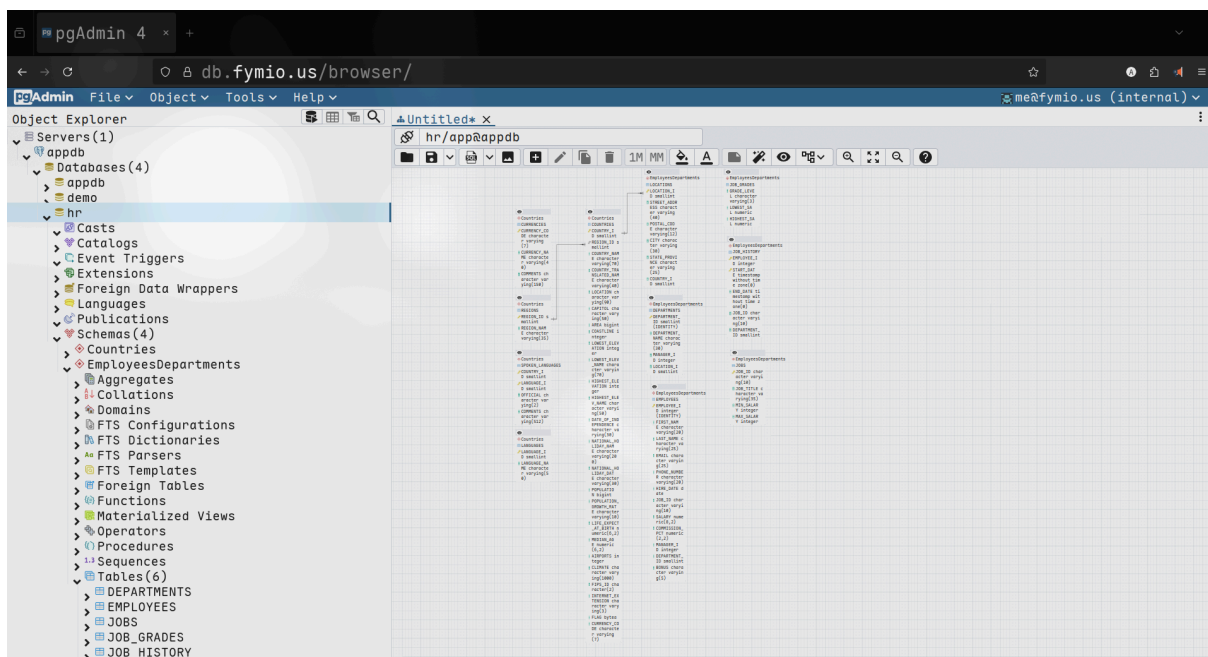
[i] [?] [X Close] [↺ Reset] [Save]

Задание 6. Построение диаграмм базы данных.

Для создания диаграммы я нажал ПКМ на узел базы данных и выбрал параметр “ERD For Database”:

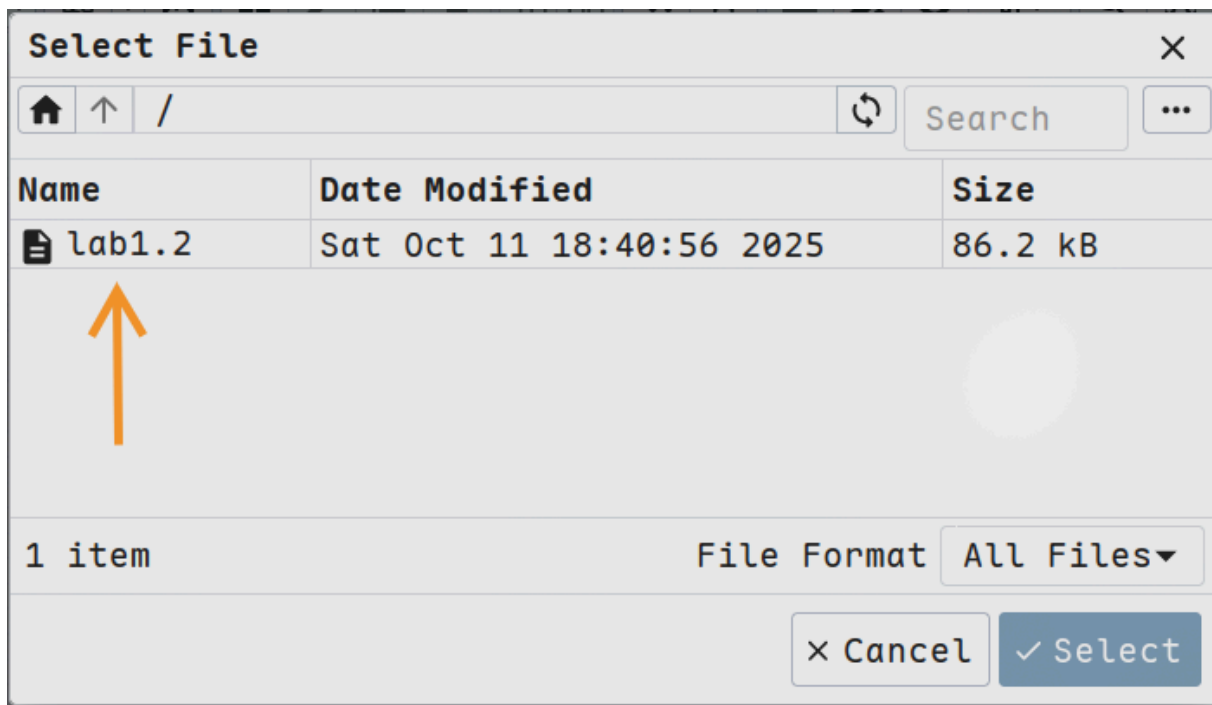


Затем я скорректировал вид своей диаграммы в соответствии с условием:



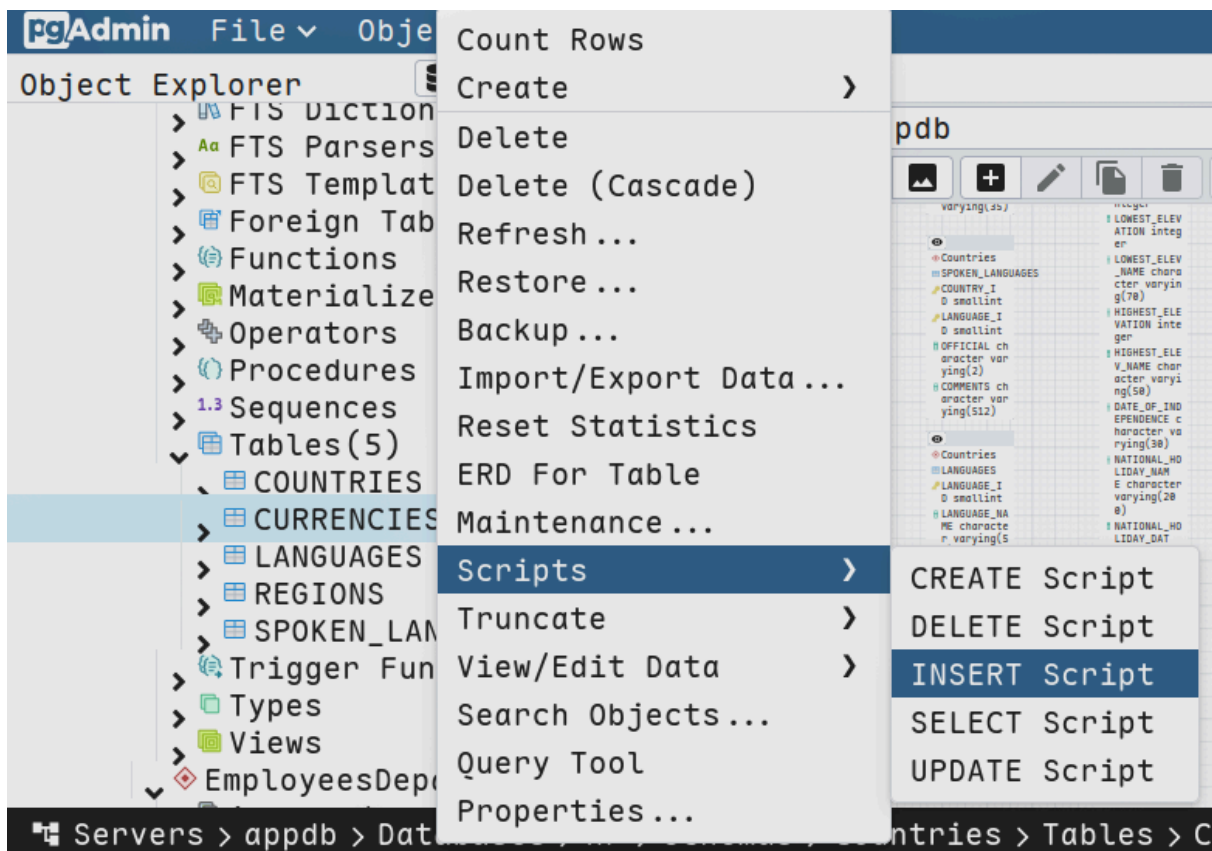
Я экспортировал снимок диаграммы, нажав на соответствующую кнопку. Результат:

Сохранил проект диаграммы:



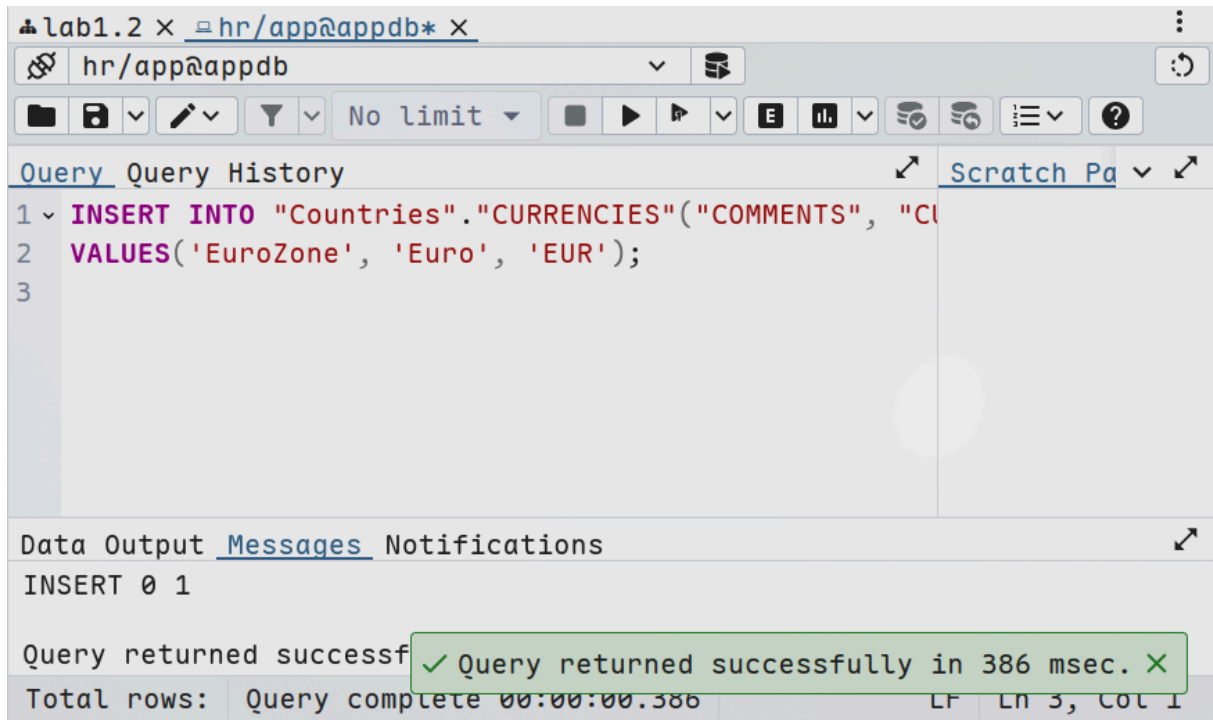
Задание 7. Добавление данных в таблицу.

Я раскрыл меню для генерирования скрипта по вставке данных в таблицу Countries.CURRENCIES:



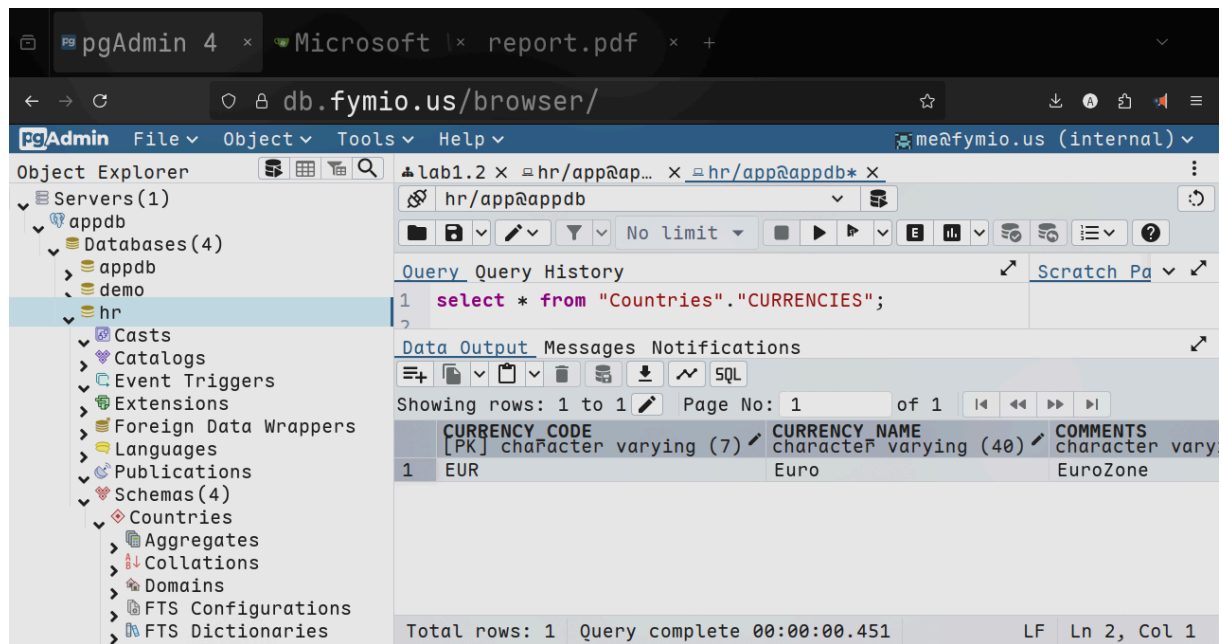
Затем я добавил следующий код в скрипт:

```
INSERT INTO "Countries"."CURRENCIES" ("COMMENTS",  
"CURRENCY_NAME", "CURRENCY_CODE")  
VALUES ('EuroZone', 'Euro', 'EUR');
```



После запуска скрипта проверил при помощи:

```
select * from "Countries"."CURRENCIES";
```



В столбце EMPLOYEE_ID я заменил identity с always на by default.

EMPLOYEE_ID

General

Definition

Constraints

Default

Not NULL?

☒

Type

NONE

✓ IDENTITY

GENERATED

Identity

BY DEFAULT

Increment

1

Start

1

Minimum

1

i

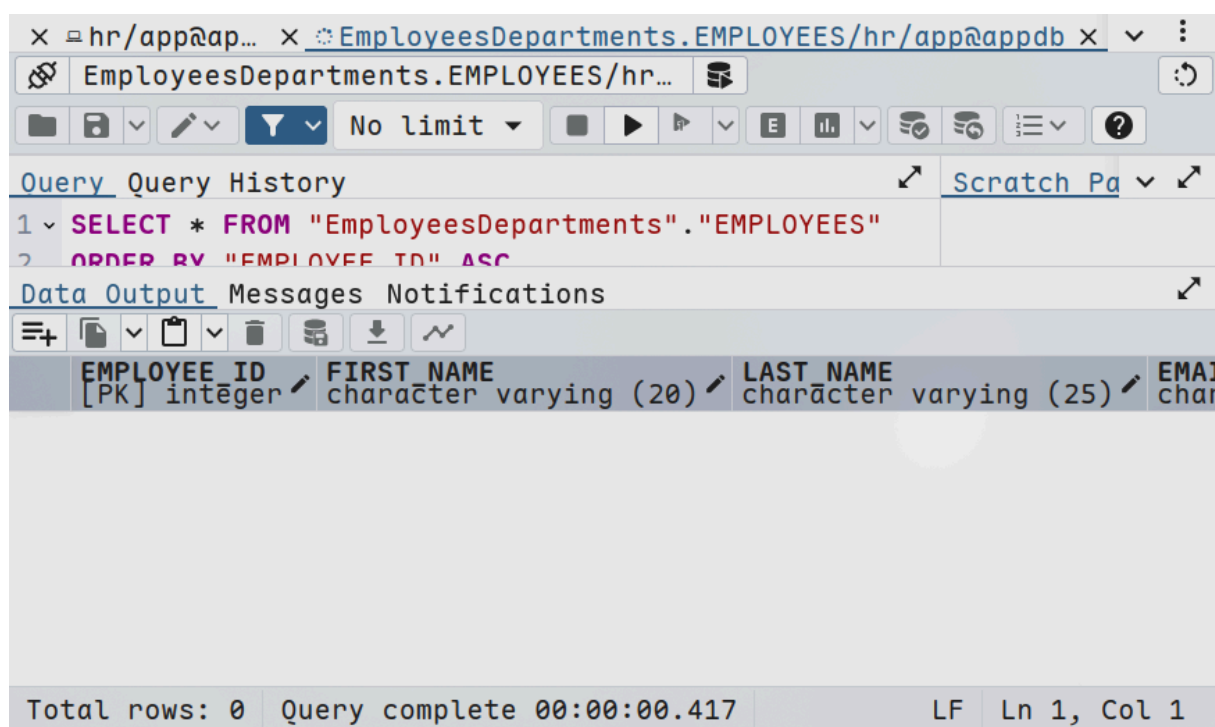
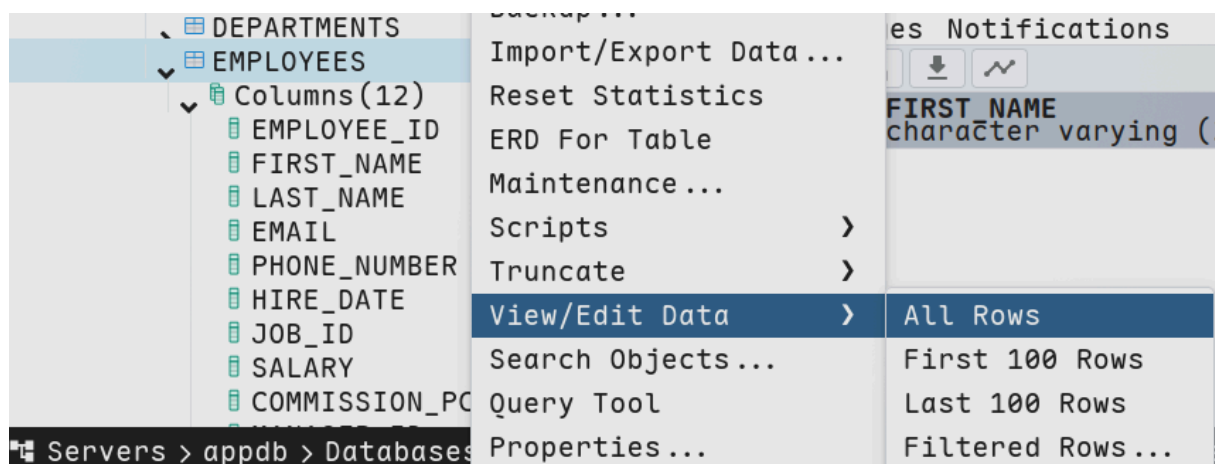
?

× Close

↺ Reset

Save

Затем при помощи графического интерфейса pgadmin, я добавил запись в таблицу “EmployeesDepartments”.“EMPLOYEES”:



Затем я добавил строку и сохранил данные:

	EMPLOYEE_ID [PK] integer	FIRST_NAME character varying (20)	LAST_NAME character varying (25)	EMAIL character varying (25)	PHONE_NUMBER character varying (15)	HIRE_DATE date	JOB_ID character varying (10)	SALARY numeric (8,2)
0+	1	Steven	King	SKING	515.123.4567	2025-05-11	AD_PRES	24000.00

COMMISSION_PCT numeric (3,2)	MANAGER integer	DEPARTMENT_ID small integer	BONUS character varying (25)
0.23	[null]	90	[null]

	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMI	MANAGER	DEP/	BONUS
	character var	date	character	numeric (	number	integer	small	character
0	515.123.4567	2025-05-11	AD_PRES	24000.00	0.23	[null]	90	[null]

✓ Data saved successfully. ✕

Total rows: 0 Query complete 00:00:00.417 LF LN 1, Col 1

Затем, в query tool я ввел следующий код:

```
INSERT INTO "EmployeesDepartments"."EMPLOYEES" ("FIRST_NAME",
"LAST_NAME", "EMAIL", "PHONE_NUMBER", "JOB_ID", "SALARY",
"COMMISSION_PCT", "MANAGER_ID", "DEPARTMENT_ID")
VALUES(2, 'Steven', 'King', 'SKING', '515.123.4567',
'AD_PRES', 24000, null, null, 90);
```

hr/app@apppdb* X

hr/app@apppdb

No limit

Query Query History

Scratch Pa

1 INSERT INTO "EmployeesDepartments"."EMPLOYEES"

2 ("EMPLOYEE_ID", "FIRST_NAME", "LAST_NAME", "EMAIL",

3 VALUES (2, 'Steven', 'King', 'SKING', '515.123.4567

4

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 518 msec.

✓ Query returned successfully in 518 msec. ✕

Total rows: Query complete 00:00:00.518 LF LN 3, Col 14

Проверим:

```
SELECT * FROM "EmployeesDepartments"."EMPLOYEES";
```

hr/app@appdb* X				
hr/app@appdb				
No limit				
Query Query History				
<pre> 1 SELECT * FROM "EmployeesDepartments"."EMPLOYEES"; 2 </pre>				
Scratch Pa				
Data Output Messages Notifications				
Showing rows: 1 to 2 Page No: 1 of 1				
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	
[PK] integer	character varying (20)	character varying (25)	char	
1	Steven	King	SKIN	
2	Steven	King	SKIN	
Total rows: 2 Query complete 00:00:00.407 LF Ln 2, Col 1				

Задание 8. Ограничения для столбцов.

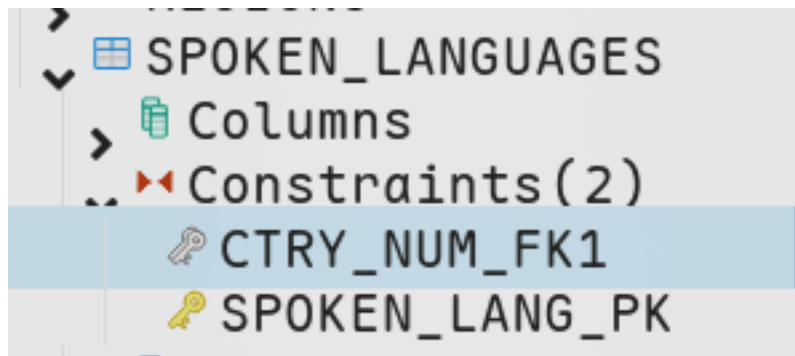
Я добавил ограничения к таблицам "Countries"."SPOKEN_LANGUAGES" и "EmployeesDepartments"."EMPLOYEES" при помощи следующего кода:

```

ALTER TABLE "Countries"."SPOKEN_LANGUAGES"
ADD CONSTRAINT "CTRY_NUM_FK1" FOREIGN KEY ("COUNTRY_ID")
REFERENCES "Countries"."COUNTRIES" ("COUNTRY_ID");

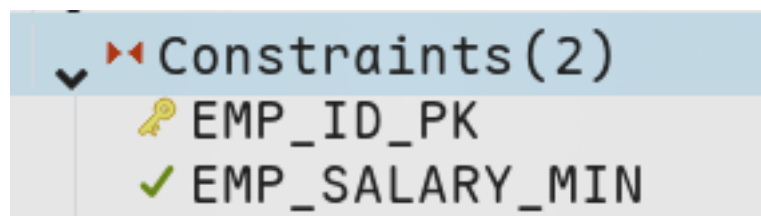
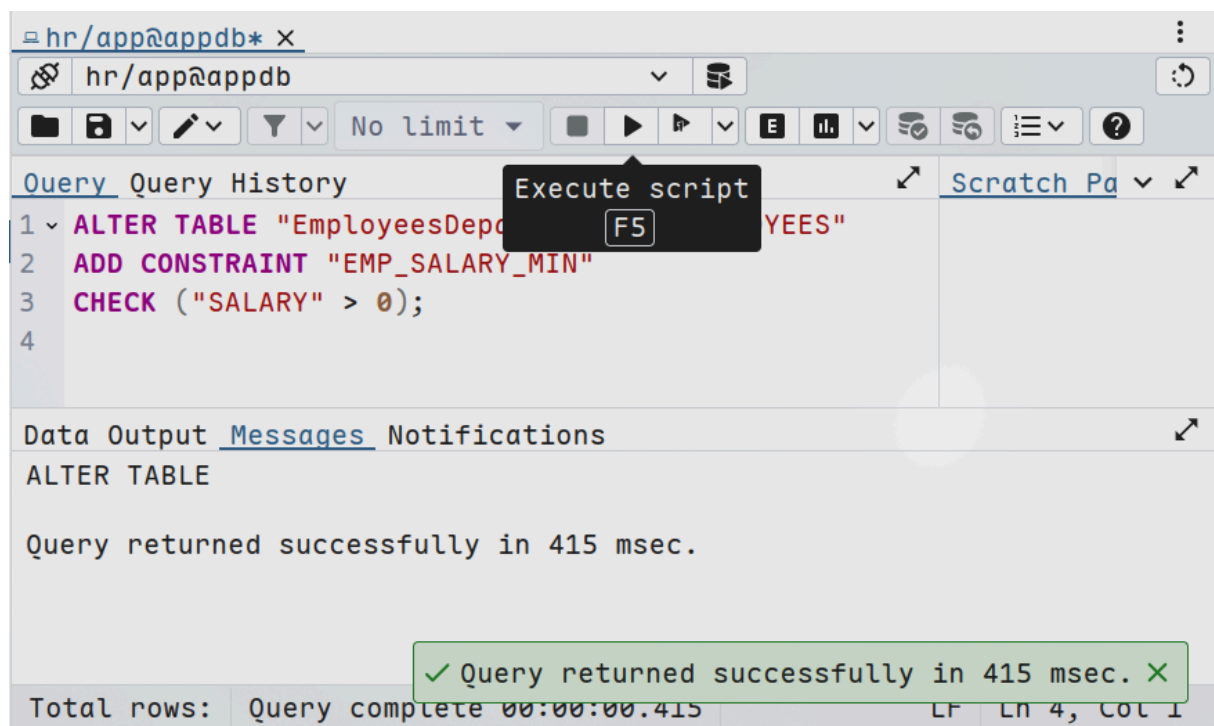
```

Query Query History				
Scratch Pa				
<pre> 1 ALTER TABLE "Countries"."SPOKEN_LANGUAGES" 2 ADD CONSTRAINT "CTRY_NUM_FK1" FOREIGN KEY ("COUNTRY_ID") 3 REFERENCES "Countries"."COUNTRIES" ("COUNTRY_ID"); 4 </pre>				
Data Output Messages Notifications				
ALTER TABLE				
Query returned successfully in 368 msec.				
Total rows: Query complete 00:00:00.368 LF Ln 4, Col 1				



И

```
ALTER TABLE "EmployeesDepartments"."EMPLOYEES"
ADD CONSTRAINT "EMP_SALARY_MIN"
CHECK ("SALARY" > 0);
```



Затем я добавил еще одно ограничение для таблицы JOB_HISTORY. Для этого в свойствах таблицы в разделе constraints добавил check с условием, что "END_DATE" > "START_DATE":

JOB_HISTORY [X]

General Columns Advanced Constraints Partitions Parameters Se >

Primary Key Foreign Key Check Unique Exclude

	Name	Check
 	JHIST_DATE_INTERVAL	"END_DATE" > "START_DATE"

[i] [?] [X Close] [↺ Reset] [Save]

```

Constraints(2)
  ✓ JHIST_DATE_INTERVAL
  🔑 JHIST_EMP_ID_ST_DATE_PK

```

Я включил это ограничение выключив соответствующую опцию:

✓ JHIST_DATE_INTERVAL

General Definition SQL

Check

"END_DATE" > "START_DATE"

No inherit?

☐

Don't validate?

☐

i?

× Close

↺ Reset

Save

⚡	Constraints (2)
✓	JHIST_DATE_INTERVAL
🔑	JHIST_EMP_ID_ST_DATE_PK

Также я проверил работоспособность ограничения:

```
INSERT INTO "EmployeesDepartments"."JOB_HISTORY"  
("EMPLOYEE_ID", "START_DATE", "END_DATE", "JOB_ID",  
"DEPARTMENT_ID")  
VALUES(200, '06-17-1993', '09-17-1987', 'AD_ASST', 90);
```

hr/app@appdb* x

hr/app@appdb

No limit

Query Query History Scratch Pa

```

1 INSERT INTO "EmployeesDepartments"."JOB_
2 ("EMPLOYEE_ID", "START_DATE", "END_DATE"
3 VALUES(200, '06-17-1993', '09-17-1987',
4

```

Data Output Messages Notifications

ERROR: Failing row contains (200, 1993-06-17 00:00:00, 1987-09-17 00:00:00, AD ASST, 90).new row for relation "JOB_HISTORY" violates check constraint "JHIST_DATE_INTERVAL"

Total rows: Query complete 00:00:00.396

LF Ln 4, Col 1

II

```

INSERT INTO "EmployeesDepartments"."JOB_HISTORY"
("EMPLOYEE_ID", "START_DATE", "END_DATE", "JOB_ID",
"DEPARTMENT_ID")
VALUES(200, '06-17-1981', '06-17-1993', 'AD_ASST', 90);

```

The screenshot shows the SQL Developer interface. The top pane contains the following SQL query:

```
1 INSERT INTO "EmployeesDepartments"."JOB_  
2 ("EMPLOYEE_ID", "START_DATE", "END_DATE"  
3 VALUES(200, '06-17-1981', '06-17-1993',  
4
```

The bottom pane shows the execution results:

```
Data Output Messages Notifications  
INSERT 0 1  
  
Query returned successfully in 378 msec.  
  
Total rows: ✓ Query returned successfully in 378 msec. ✕
```

The status bar at the bottom indicates "LF Ln 4, Col 1".

Задание 9. Очистка таблиц и завершение создания БД HR.

Я произвел очистку таблицы EMPLOYEES с помощью DDL команды TRUNCATE :

```
TRUNCATE "EmployeesDepartments"."EMPLOYEES";
```

The screenshot shows the SQL Developer interface with the connection "hr/app@apppdb" selected. The top pane contains the following SQL query:

```
1 TRUNCATE "EmployeesDepartments"."EMPLOYE  
2
```

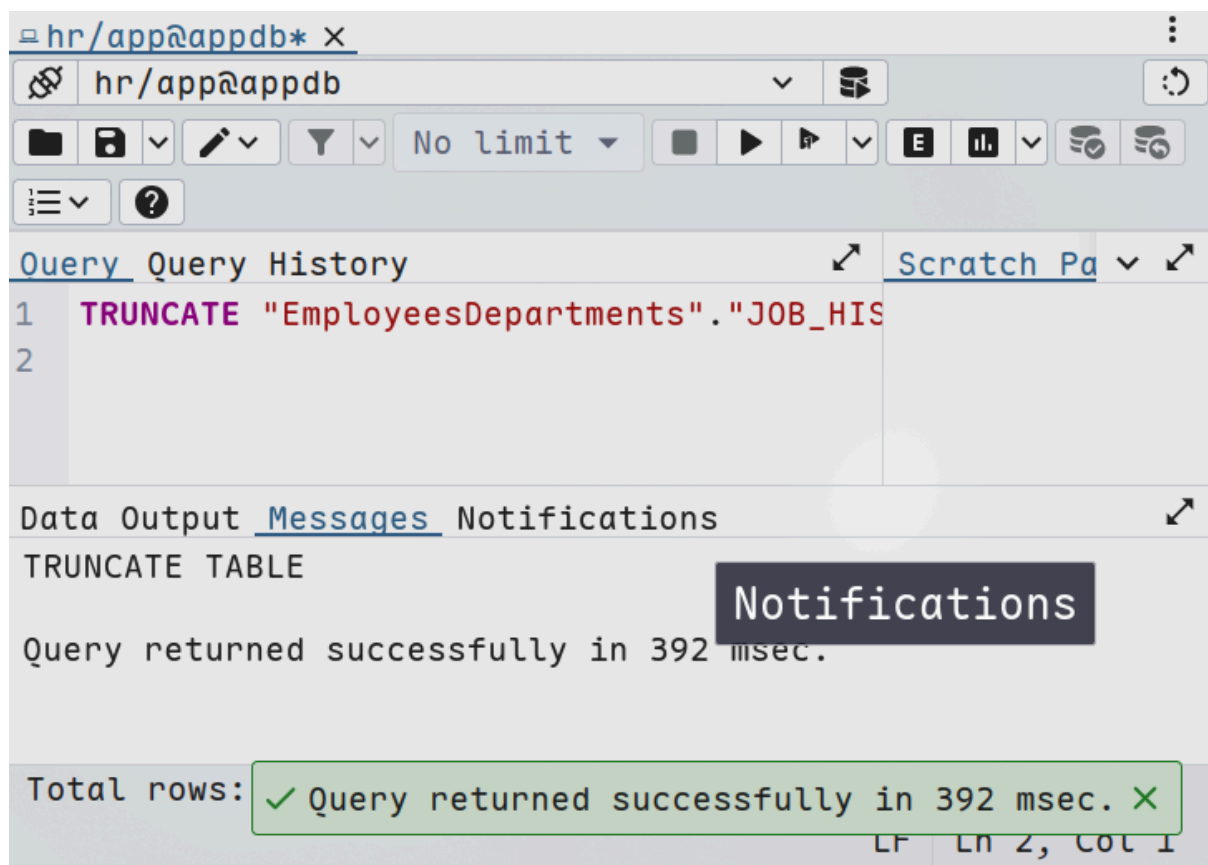
The bottom pane shows the execution results:

```
Data Output Messages Notifications  
TRUNCATE TABLE  
  
Query returned successfully in 401 msec.  
  
Total rows: Query complete 00:00:00.401
```

The status bar at the bottom indicates "LF Ln 2, Col 1".

Также я очистил JOB_HISTORY и CURRENCIES:

```
TRUNCATE "EmployeesDepartments"."JOB_HISTORY";
```



И

```
TRUNCATE "Countries"."CURRENCIES";
```


The screenshot shows the SQL Developer interface with the connection 'hr/app@appdb' selected. The 'Query' tab is active, displaying a SQL script with two lines: `1 TRUNCATE "Countries"."CURRENCIES";` and `2`. The 'Messages' tab is also visible, showing the output: `TRUNCATE TABLE` and `Query returned successfully in 394 msec.`. The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.394' and the cursor position 'LF Ln 2, Col 1'.

После очистки я выполнил код из файла `script2_alter.sql`:

The screenshot shows the SQL Developer interface with the connection 'hr/app@appdb' selected. The 'Query' tab is active, displaying a SQL script with three lines: `47`, `48 CREATE INDEX "JHIST_DEPARTMENT_IX" ON`, `49 CREATE INDEX "JHIST_EMPLOYEE_IX" ON "`, and `50 CREATE INDEX "JHIST_JOB_IX" ON "Emplc`. The 'Messages' tab is also visible, showing the output: `CREATE INDEX` and `Query returned successfully in 469 msec.`. The status bar at the bottom indicates 'Total rows: ✓ Query returned successfully in 469 msec. ✗' and the cursor position 'LF Ln 50, Col 82'.

И для добавления данных, я выполнил скрипт `script3_insert.sql`:

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations, editing, and execution. The main window displays a SQL script with the following content:

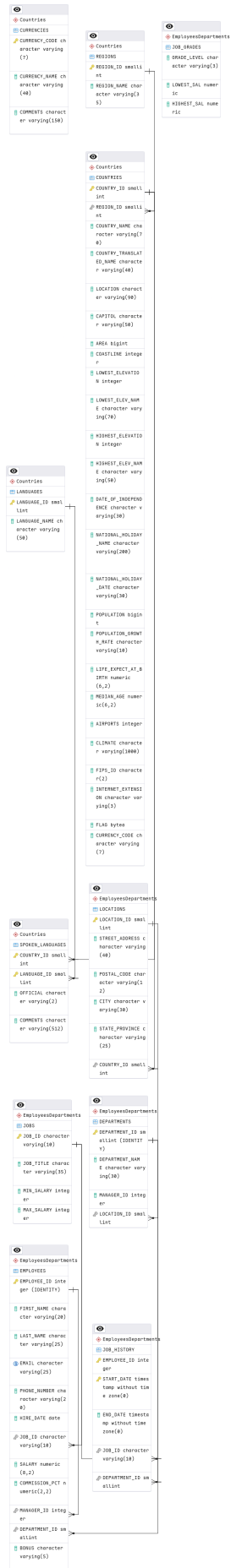
```
1421 REFERENCES "EmployeesDepartments"  
1422  
1423 ALTER SEQUENCE "EmployeesDepartments"  
1424 RESTART WITH 236;  
1425  
1426  
1427
```

Below the script, the "Data Output" tab is active, showing the following messages:

```
Query returned successfully in 901 msec.  
Total rows: Query complete 00:00:00.901
```

The status bar at the bottom indicates the cursor is at line 1427, column 1.

Результат:



Выводы.

В ходе работы были поставлены цели изучить принципы проектирования и администрирования реляционных баз данных в postgresql, освоить создание таблиц, настройку ключей и ограничений, добавление и проверку данных, а также построение ег диаграмм.

Что должны были достичь:

- Создать таблицы с корректной структурой и типами данных.
- Настроить первичные и внешние ключи, ограничения `NOT NULL` и `CHECK`.
- Добавить тестовые данные через pgadmin и sql запросы.
- Создать индексы и визуализировать структуру базы данных через ег диаграмму.

Что фактически достигнуто:

- Все требуемые таблицы созданы.
- Настроены первичные и внешние ключи, значения по умолчанию и автоинкрементные поля.
- Добавлены данные, проверены ограничения целостности.
- Построена и сохранена ег диаграмма базы.

Трудности и решения:

- Появлялись ошибки дублирования ключей при вставке данных, решено изменением генерации `IDENTITY` с `ALWAYS` на `BY DEFAULT`.
- Нарушение внешнего ключа при добавлении записей решалось последовательным заполнением родительских таблиц перед зависимыми.
- Ошибки при создании ограничений и индексов устранялись проверкой существующих объектов и корректировкой sql запросов.

Работа позволила закрепить навыки создания и администрирования баз данных в postgresql, понять логику ограничений и связей между таблицами, а также научиться эффективно добавлять и проверять данные с соблюдением целостности базы. Все задачи лабораторной выполнены с учетом практических рекомендаций и ошибок, которые возникали в процессе работы.