

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Практическая работа №2

Выполнил:

Доценников Никита Андреевич

Группа: КЗ221

Проверила:

Татьяна Евгеньевна Войтюк

Цель работы

Задачи, решаемые при выполнении работы

Исходные данные

Выполнение работы

Задание 1. Описание структуры таблицы, выборка данных из таблицы, задание имен столбцов, сортировка строк с помощью предложения ORDER BY

1.1 Будет ли успешна эта команда SELECT ?

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

Нет, не будет, так как для корректного выполнения данной команды, нужно добавить кавычки у JOB_GRADES (Рис. 1).

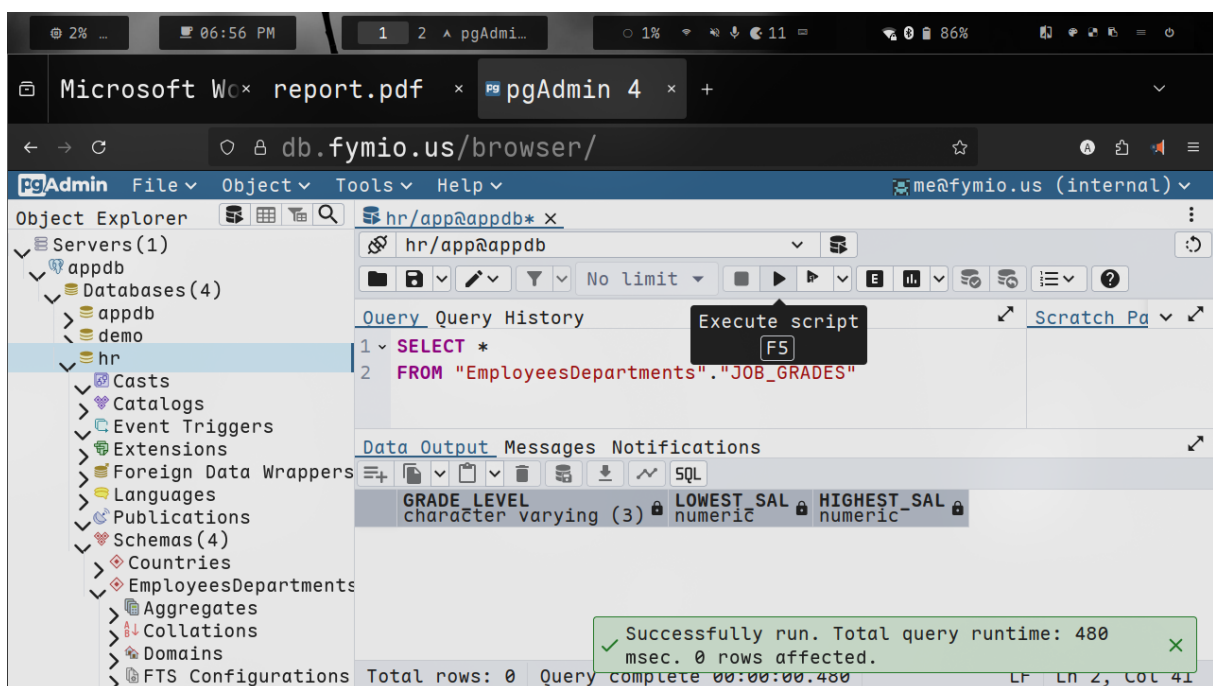


Рис. 1: Исполнение исправленного запроса.

1.2 Команда SELECT содержит 4 ошибки. Укажите их.

```
SELECT "EMPLOYEE_ID", "LAST_NAME"  
"SAL" x 12 ANNUAL SALARY  
FROM "EmployeesDepartments"."EMPLOYEES"
```

- Нет запятой перед "SAL".
- Использовать * вместо x.
- Неправильно задан псевдоним: нужен AS и, раз в псевдониме пробел, - кавычки (например AS "ANNUAL SALARY").
- Неправильное название столбца: вместо SAL должно быть SALARY.

Исправленная версия (Рис. 2):

```
SELECT "EMPLOYEE_ID", "LAST_NAME", "SALARY" * 12 AS "ANNUAL SALARY"
FROM "EmployeesDepartments"."EMPLOYEES";
```

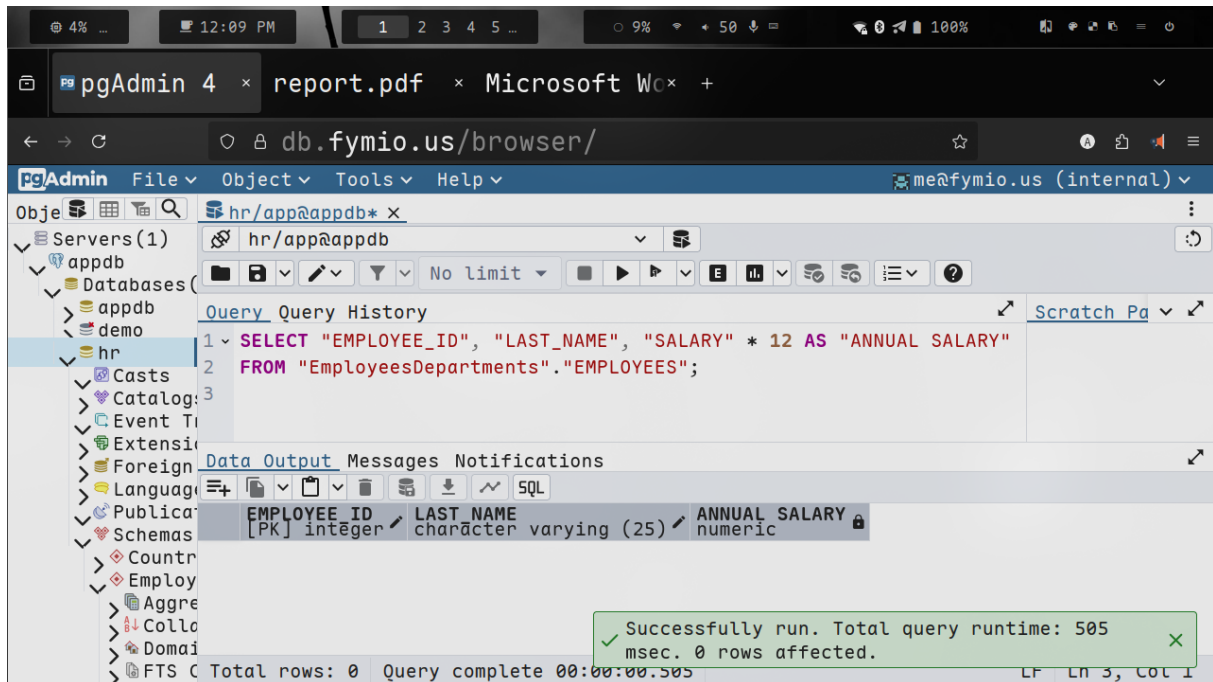


Рис. 2: Исполнение исправленного запроса.

1.3 Напишите запрос, который отображает структуру таблицы **DEPARTMENTS**, представленную на таблице 1. Сформируйте запрос на выборку данных из нее, результат должен соответствовать таблице 2.

column_name	is_nullable	Type
DEPARTMENT_ID	NO	smallint
DEPARTMENT_NAME	NO	character varying
MANAGER_ID	YES	integer
LOCATION_ID	YES	smallint

Табл. 1: Структура таблицы **DEPARTMENTS**.

```
SELECT column_name,
       CASE WHEN is_nullable = 'YES' THEN 'YES' ELSE 'NO' END AS is_nullable,
       data_type
FROM information_schema.columns
WHERE table_name = 'DEPARTMENTS'
      AND table_schema IN ('EmployeesDepartments', 'public')
ORDER BY ordinal_position;
```

Этот запрос извлекает данные таблицы **DEPARTMENTS** из представления **information_schema.columns**.

- **column_name** — имя столбца;
- **is_nullable** — может ли столбец содержать null;
- **data_type** — тип данных столбца.

Фильтрация по `table_name` и `table_schema` позволяет получить только нужную таблицу. Сортировка по `ordinal_position` отображает столбцы в порядке их создания в таблице. Результат выполнения скрипта представлен на рисунке 3.

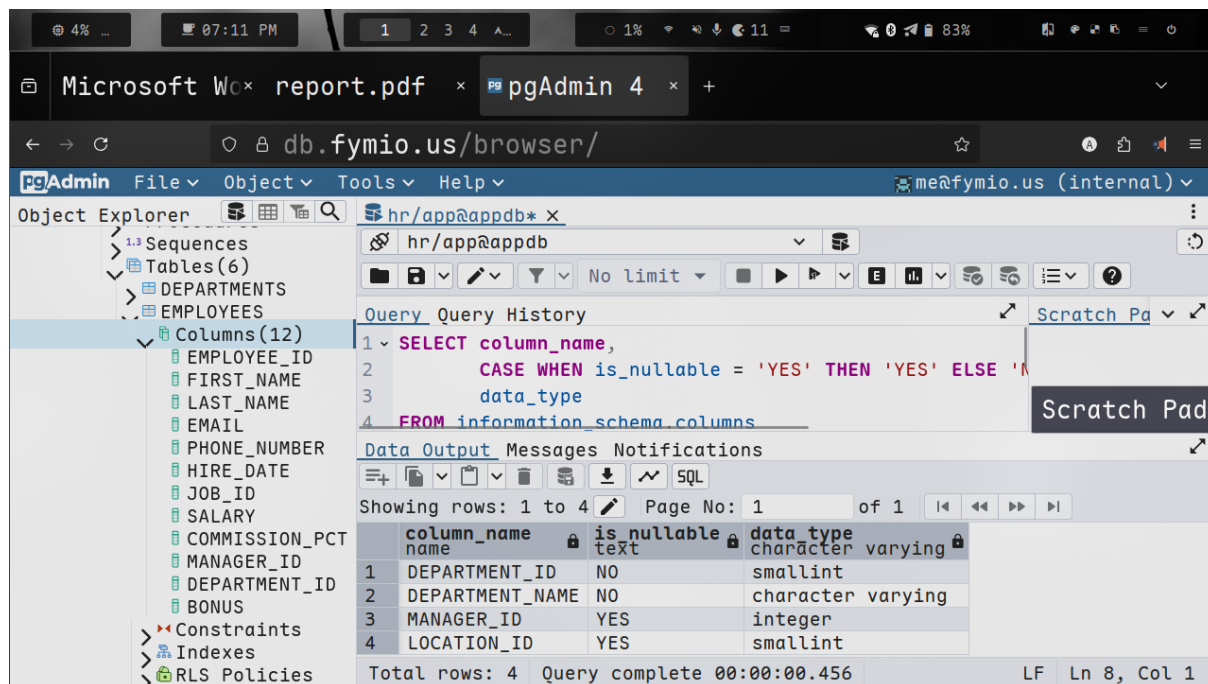


Рис. 3: Результат выполнения запроса.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales - Europe	149	2500
85	Sales - Americas	149	2100
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

Табл. 2: Результат выполнения запроса к таблице `DEPARTMENTS`.

```
SELECT "DEPARTMENT_ID", "DEPARTMENT_NAME", "MANAGER_ID", "LOCATION_ID"
FROM "EmployeesDepartments"."DEPARTMENTS"
ORDER BY "DEPARTMENT_ID";
```

Этот запрос выбирает все данные из таблицы `DEPARTMENTS` в схеме `EmployeesDepartments`. Сортировка по `DEPARTMENT_ID` обеспечивает упорядоченный вывод строк. Результат выполнения запроса представлен на рисунке 4.

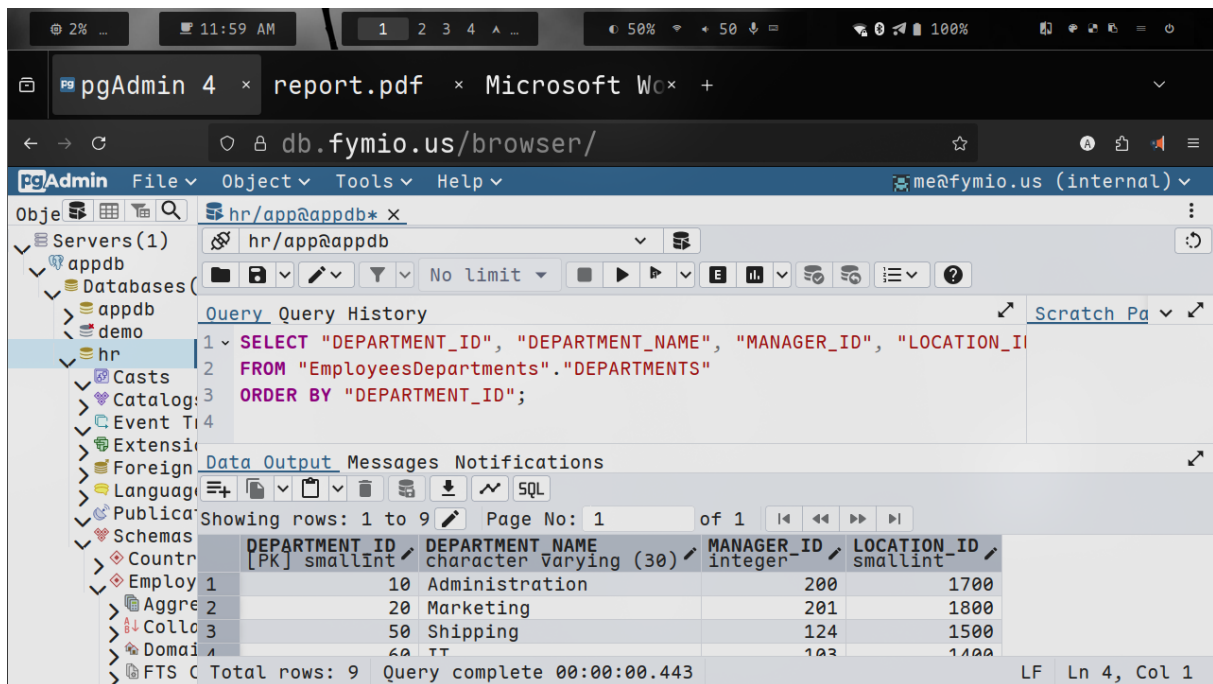


Рис. 4: Результат выполнения запроса.

1.4 Напишите запрос, который отображает структуру таблицы **EMPLOYEES**, представленную в таблице 3.

Name	Null?	Type
HIRE_DATE	NO	date
SALARY	YES	numeric
COMMISSION_PCT	YES	numeric
MANAGER_ID	YES	integer
DEPARTMENT_ID	YES	smallint
EMPLOYEE_ID	NO	integer
BONUS	YES	character varying
FIRST_NAME	YES	character varying
LAST_NAME	NO	character varying
EMAIL	NO	character varying
PHONE_NUMBER	YES	character varying
JOB_ID	NO	character varying

Табл. 3: Результат выполнения запроса к таблице **DEPARTMENTS**.

```
SELECT column_name AS "Name",
       CASE WHEN is_nullable = 'YES' THEN 'YES' ELSE 'NO' END AS "Null?",
       data_type AS "Type"
FROM information_schema.columns
WHERE table_schema = 'EmployeesDepartments'
      AND table_name = 'EMPLOYEES'
ORDER BY ordinal_position;
```

Этот запрос извлекает информацию о столбцах таблицы **EMPLOYEES** :

- **Name** — имя столбца;
- **Null?** — может ли столбец содержать NULL;
- **Type** — тип данных столбца.

Фильтрация по **table_schema** и **table_name** позволяет получить данные только для нужной таблицы. Сортировка по **ordinal_position** отображает столбцы в том порядке, в котором они были созданы в таблице. Результат запроса приведен на рисунке 5.

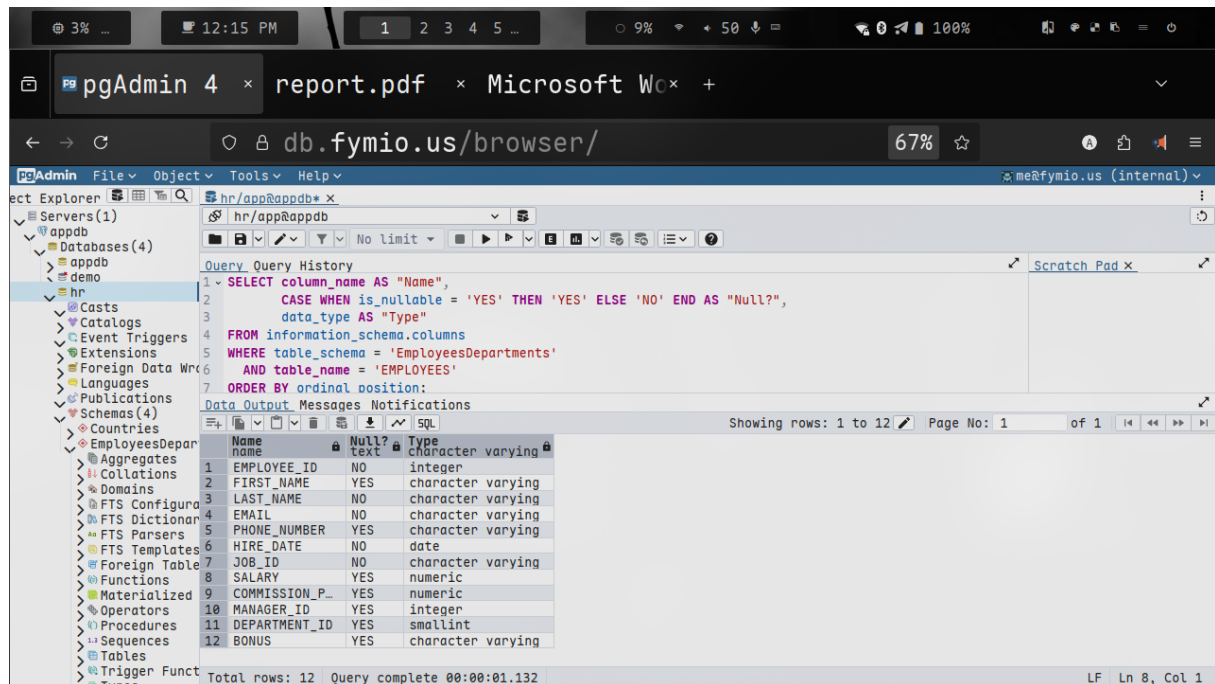


Рис. 5: Результат выполнения запроса.

1.5 Составьте запрос для вывода фамилии каждого служащего, должности, даты найма и номера. Номер служащего должен быть первым. Результат запроса должен быть схож с таблицей 4.

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
100	King	AD_PRES	2002-06-17
101	Kochhar	AD_VP	2004-09-21
102	De Haan	AD_VP	2008-01-13
103	Hunold	IT_PROG	2005-01-03
104	Ernst	IT_PROG	2006-05-21
107	Lorentz	IT_PROG	2014-02-07
124	Mourgos	ST_MAN	2014-11-16
141	Rajs	ST_CLERK	2010-10-17
142	Davies	ST_CLERK	2012-01-29
143	Matos	ST_CLERK	2013-03-15
144	Vargas	ST_CLERK	2013-07-09
...	...	...	...

Табл. 4: Часть результата выполнения запроса из пункта 1.5.

```
SELECT "EMPLOYEE_ID",
       "LAST_NAME",
       "JOB_ID",
       "HIRE_DATE"
FROM "EmployeesDepartments"."EMPLOYEES"
ORDER BY "EMPLOYEE_ID";
```

В этом запросе выбираются четыре столбца таблицы EMPLOYEES :

- EMPLOYEE_ID — идентификатор сотрудника, выводится первым;
- LAST_NAME — фамилия сотрудника;
- JOB_ID — код должности;
- HIRE_DATE — дата найма.

Сортировка по EMPLOYEE_ID обеспечивает упорядоченный вывод. Использование точных имён столбцов с кавычками гарантирует корректное выполнение запроса в PostgreSQL с регистрозависимыми именами. Результат выполнения запроса представлен на рисунке 6.

The screenshot shows the pgAdmin 4 interface. The query window displays the following SQL query:

```
1 SELECT "EMPLOYEE_ID",
2       "LAST_NAME",
3       "JOB_ID",
4       "HIRE_DATE"
5 FROM "EmployeesDepartments"."EMPLOYEES"
6 ORDER BY "EMPLOYEE_ID";
```

The Data Output tab shows the results of the query, displaying 11 rows. The columns are EMPLOYEE_ID (integer), LAST_NAME (character varying (25)), JOB_ID (character varying (10)), and HIRE_DATE (date). The status bar indicates 'Total rows: 11 Query complete 00:00:00.445'.

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
100	King	AD_PRES	2002-06-17
101	Kochhar	AD_VP	2004-09-21
102	De Haan	AD_VP	2008-01-13
103	Hunold	IT_PROG	2005-01-03
104	Ernst	IT_PROG	2006-05-21
107	Lorentz	IT_PROG	2014-02-07
124	Mourgos	ST_MAN	2014-11-16
141	Rajs	ST_CLERK	2010-10-17
142	Davies	ST_CLERK	2012-01-29
143	Matos	ST_CLERK	2013-03-15
144	Vargas	ST_CLERK	2013-07-09

Рис. 6: Результат выполнения запроса.

1.6 Составьте запрос для вывода неповторяющихся должностей из таблицы EMPLOYEES, результат должен соответствовать таблице 5.

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
IT_PROG
MK_MAN
MK_REP
SA_MAN
SA_REP
SR_MK_REP
SR_SA_REP
SR_ST_CLRK
ST_CLERK
ST_MAN

Табл. 5: Результат выполнения запроса для вывода неповторяющихся должностей.

```
SELECT DISTINCT "JOB_ID"  
FROM "EmployeesDepartments"."EMPLOYEES"  
ORDER BY "JOB_ID";
```

В этом запросе используется ключевое слово `DISTINCT` для того, чтобы выбрать только уникальные значения столбца `JOB_ID` из таблицы `EMPLOYEES`. Сортировка по `JOB_ID` упорядочивает результат по алфавиту. Результат выполнения запроса представлен на рисунке 7.

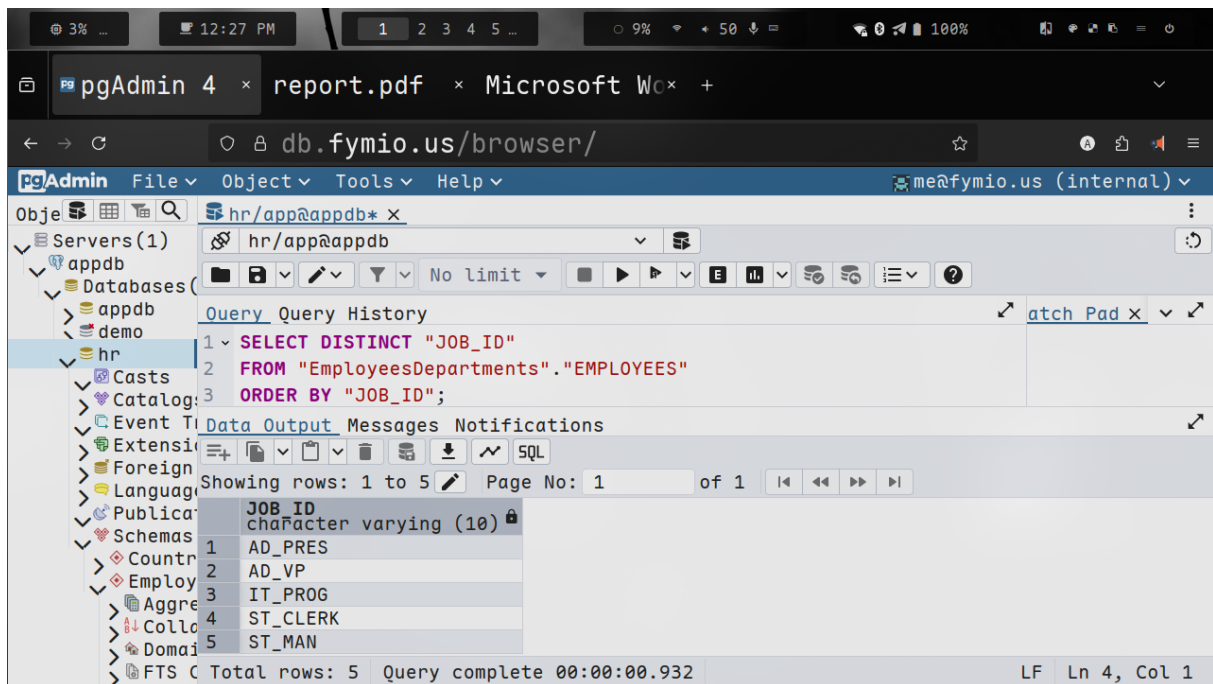


Рис. 7: Результат выполнения запроса.

1.7 Выведите на экран фамилию, соединенную с идентификатором должности через запятую и пробел. Назовите новый столбец Employee and Title. Результат запроса должен быть схож с таблицей 6.

Employee and Title
King, AD_PRE
Kochhar, AD_VP
De Haan, AD_VP
Hunold, IT_PROG
Ernst, IT_PROG
Lorentz, IT_PROG
Mourgos, ST_MAN
Rajs, ST_CLERK
Davies, ST_CLERK
Matos, ST_CLERK
Vargas, ST_CLERK
...

Табл. 6: Результат выполнения запроса для вывода фамилии, соединённой с идентификатором должности.

```
SELECT "LAST_NAME" || ', ' || "JOB_ID" AS "Employee and Title"
FROM "EmployeesDepartments"."EMPLOYEES"
ORDER BY "EMPLOYEE_ID";
```

В этом запросе используется оператор `||` для конкатенации строк:

- сначала `LAST_NAME`,
- затем запятая и пробел `', '`,

- затем `JOB_ID` .

Результат выводится под новым именем столбца `Employee and Title` . Результат выполнения запроса представлен на рисунке 8.

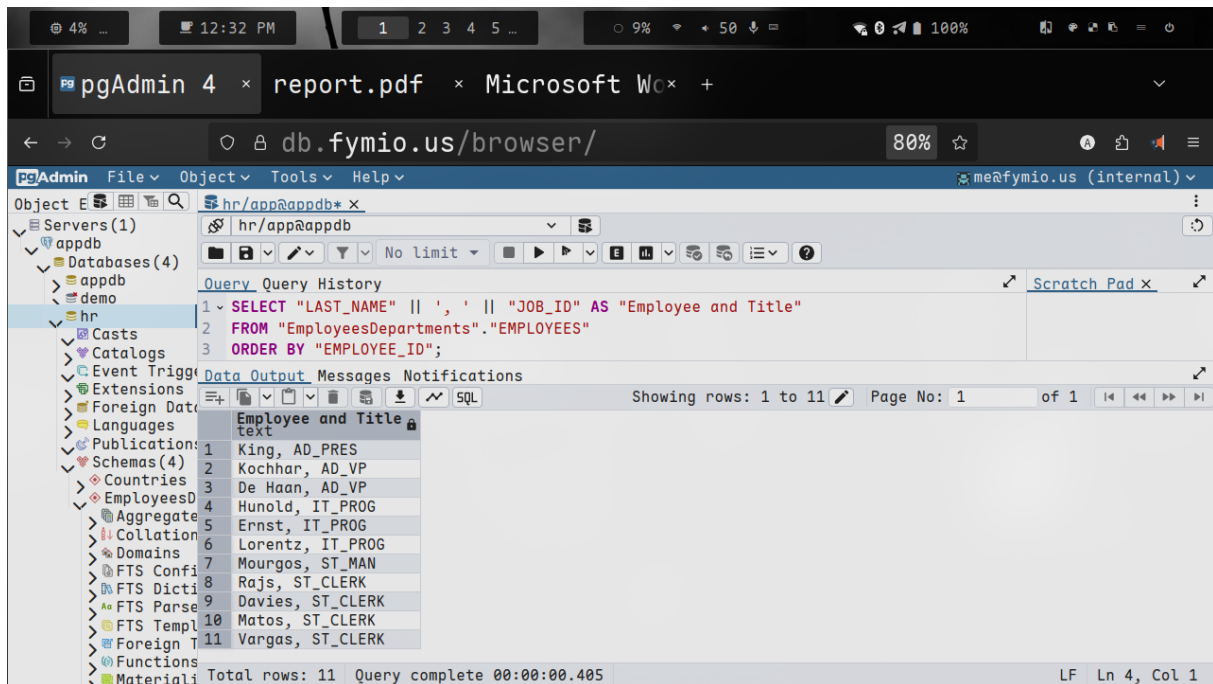


Рис. 8: Результат выполнения запроса.

Задание 2. Выборка данных и изменение последовательности вывода строк, ограничение количества возвращаемых строк с помощью предложения `WHERE` , сортировка строк с помощью предложения `ORDER BY` .

2.1 Создайте запрос для вывода фамилии и заработной платы служащих, зарабатывающих более 12000. Результат выполнения запроса должен соответствовать таблице 7.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hartstein	13000

Рис. 7: Результат выполнения запроса для вывода фамилии и заработной платы служащих

```
SELECT "LAST_NAME", "SALARY"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "SALARY" > 12000
ORDER BY "SALARY" DESC;
```

- выбираются столбцы `LAST_NAME` и `SALARY` из таблицы `EMPLOYEES` ;
- фильтруются только те строки, где заработная плата больше 12000;
- строки сортируются по убыванию зарплаты, чтобы сначала отображались сотрудники с наибольшей зарплатой.

Результат выполнения запроса представлен на рисунке 9.

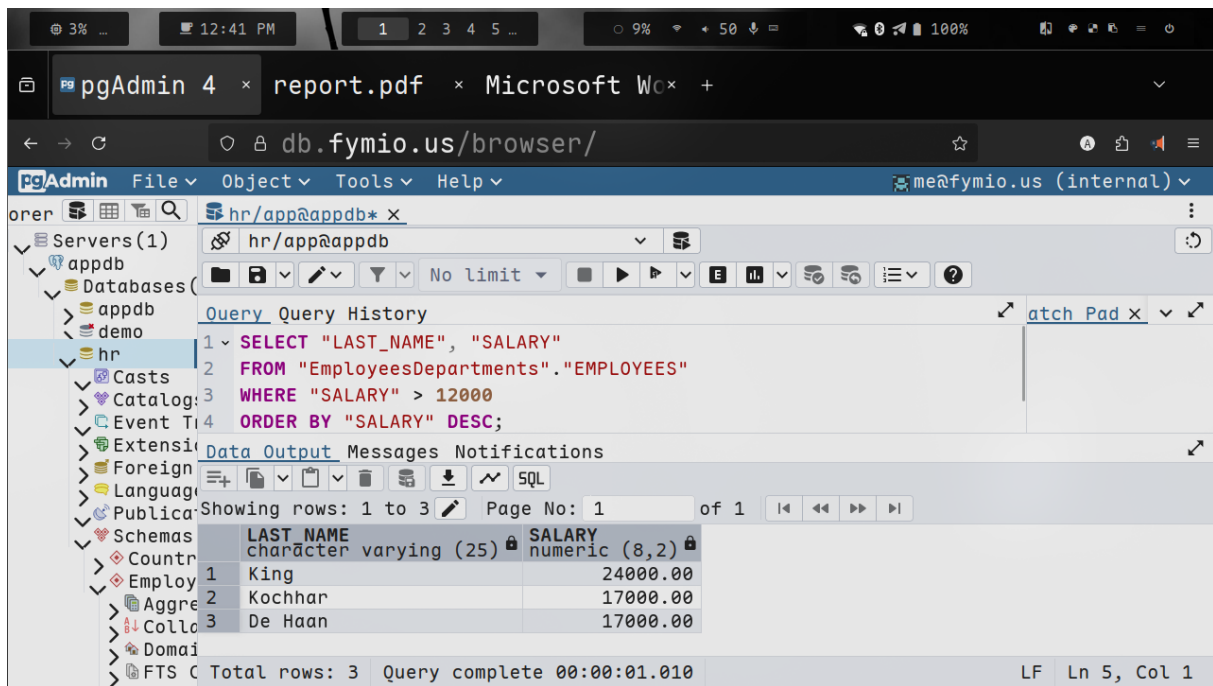


Рис. 9: Результат выполнения запроса.

2.2 Создайте запрос для вывода фамилии и номера отдела служащего под номером 176. Результат выполнения запроса должен соответствовать таблице 7.

LAST_NAME	DEPARTMENT_ID
Taylor	80

Табл. 8: Результат выполнения запроса для служащего под номером 176.

```
SELECT "LAST_NAME", "DEPARTMENT_ID"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "EMPLOYEE_ID" = 176;
```

- выбираются столбцы `LAST_NAME` и `DEPARTMENT_ID` из таблицы `EMPLOYEES` ;
- фильтрация выполняется по точному значению `EMPLOYEE_ID` с помощью условия `WHERE` ;

Результат выполнения запроса представлен на рисунке 10.

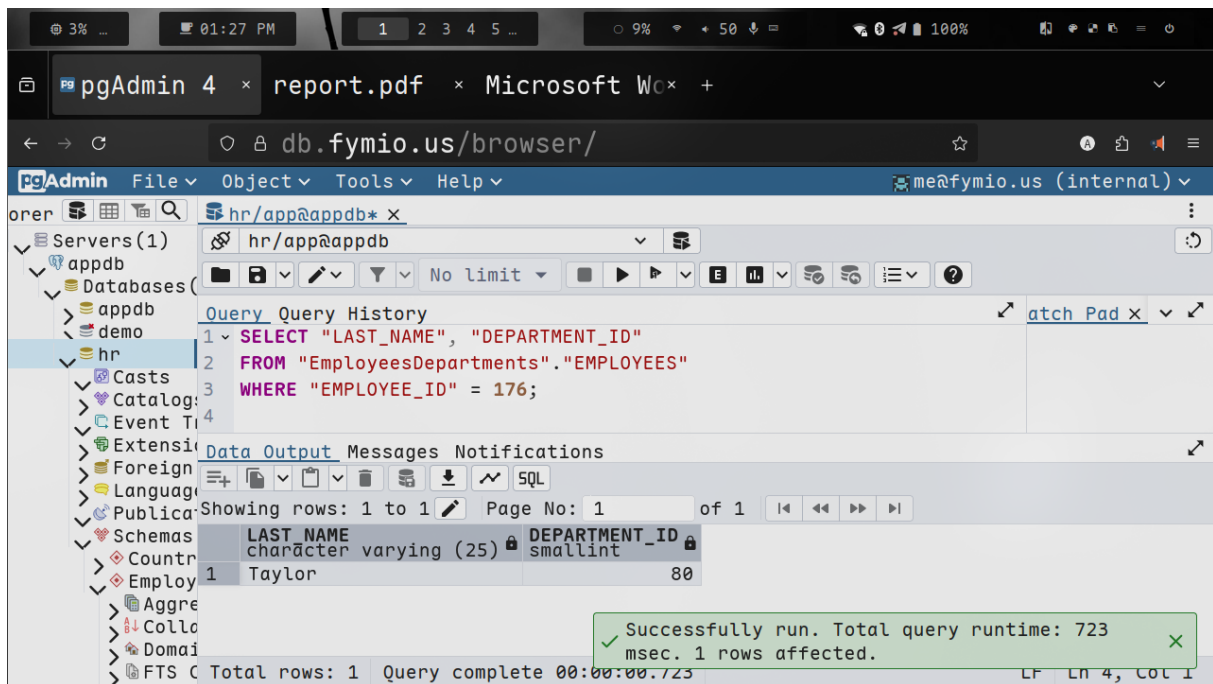


Рис. 10: Результат выполнения запроса.

2.3 Измените запрос из задания 2.1 и выведите фамилии и оклады всех служащих, чей оклад не входит в диапазон от 5000 до 12000. Результат выполнения запроса должен быть схож с таблицей 8.

LAST_NAME	SALARY
King	24000.00
Kochhar	17000.00
De Haan	17000.00
Lorentz	4200.00
Rajs	3500.00
Davies	3100.00
Matos	2600.00
Vargas	2500.00
Whalen	4400.00
Hartstein	13000.00

Табл. 9: Результат выполнения запроса для служащих, чей оклад не входит в диапазон от 5000 до 12000.

```
SELECT "LAST_NAME", "SALARY"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "SALARY" < 5000 OR "SALARY" > 12000
ORDER BY "SALARY" DESC;
```

- выбираются столбцы `LAST_NAME` и `SALARY` из таблицы `EMPLOYEES`;
- фильтруются строки, где оклад не входит в диапазон от 5000 до 12000, с помощью условия `WHERE ... OR`;
- строки сортируются по убыванию оклада.

Результат выполнения запроса представлен на рисунке 11.

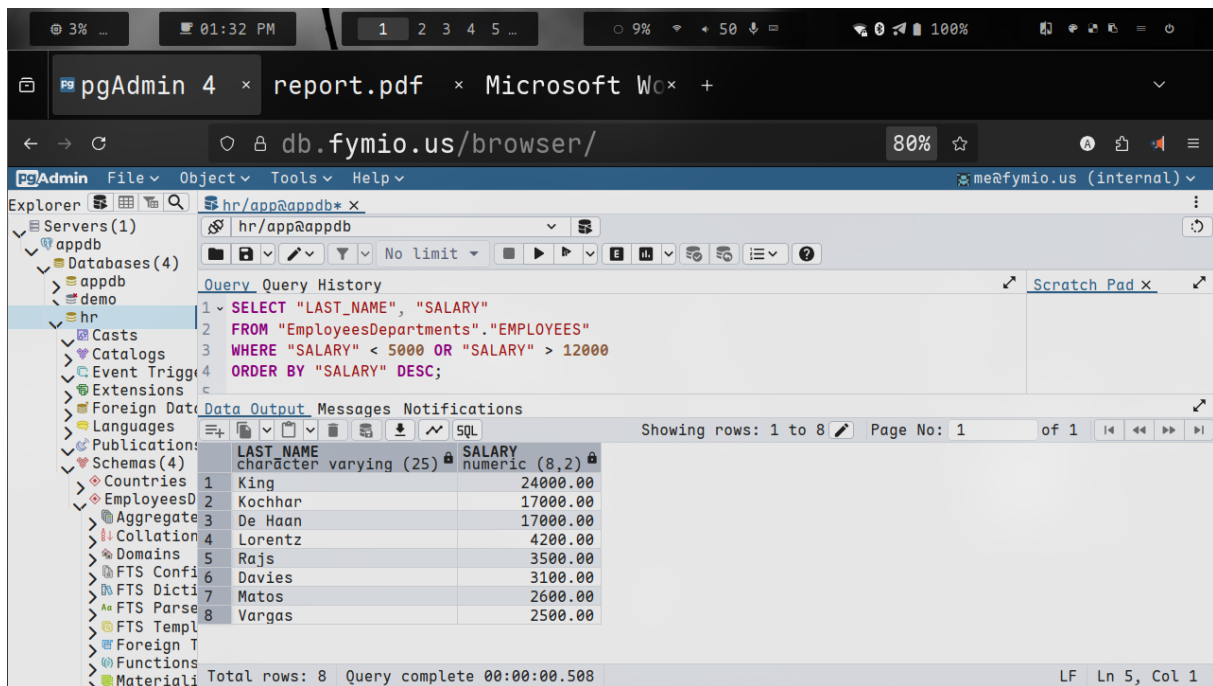


Рис. 11: Результат выполнения запроса.

2.4 Выведите фамилию, идентификатор должности и дату начала работы всех служащих, нанятых в период с 16 февраля 2011 по 12 мая 2011г. Отсортируйте данные в порядке возрастания даты найма. Результат выполнения запроса должен соответствовать таблице 9.

LAST_NAME	JOB_ID	HIRE_DATE
Hartstein	MK_MAN	2011-02-17
Abel	SA_REP	2011-05-11

Табл. 10: Результат выполнения запроса для служащих, нанятых в период с 16 февраля 2011 по 12 мая 2011г.

```
SELECT "LAST_NAME", "JOB_ID", "HIRE_DATE"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "HIRE_DATE" BETWEEN '2011-02-16' AND '2011-05-12'
ORDER BY "HIRE_DATE" ASC;
```

- выбираются столбцы `LAST_NAME`, `JOB_ID` и `HIRE_DATE` из таблицы `EMPLOYEES`;
- фильтрация выполняется по дате найма с помощью предложения `WHERE` и `BETWEEN`, чтобы выбрать сотрудников, нанятых с 16 февраля по 12 мая 2011 года включительно;
- сортировка по возрастанию даты найма.

Результат выполнения запроса представлен на рисунке 12.

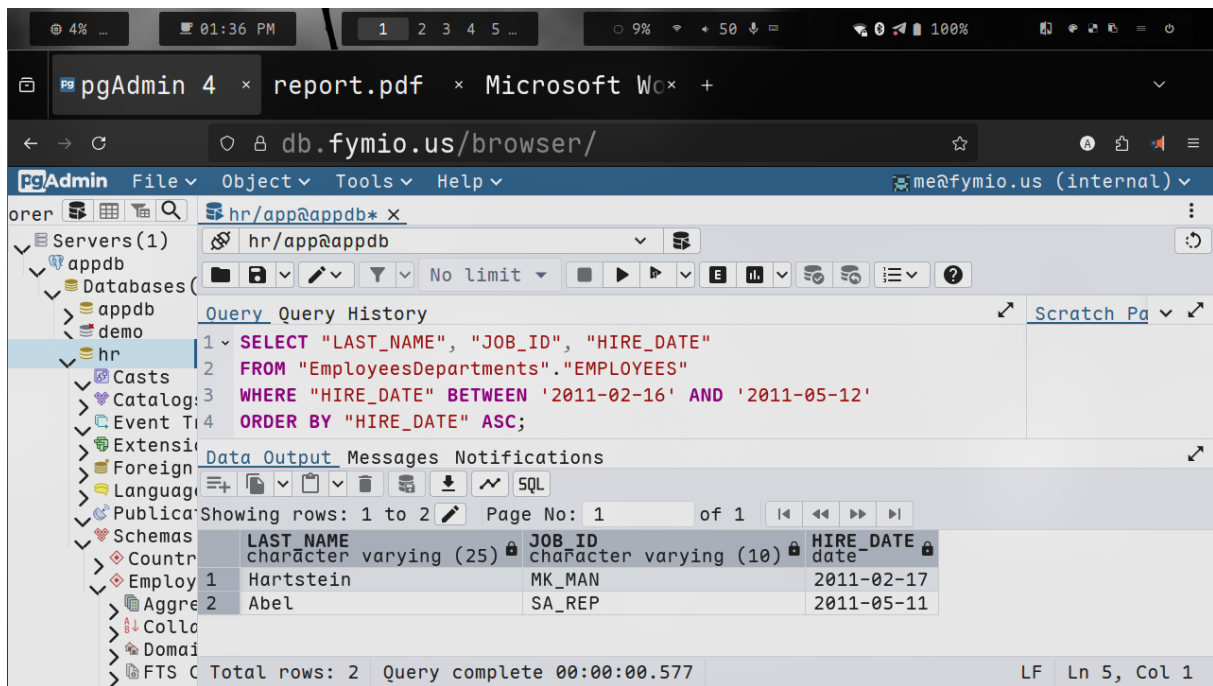


Рис. 12: Результат выполнения запроса.

2.5 Выведите фамилию и номер отдела всех служащих из отделов 20 и 50.

Отсортируйте данные по фамилиям в алфавитном порядке. Результат выполнения запроса должен быть схож с таблицей 10.

LAST_NAME	DEPARTMENT_ID
Bell	50
Davies	50
Fay	20
Hartstein	20
Heiden	50
Matos	50
Mourgos	50

Табл. 11: Результат выполнения запроса для служащих из отделов 20 и 50.

```
SELECT "LAST_NAME", "DEPARTMENT_ID"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "DEPARTMENT_ID" IN (20, 50)
ORDER BY "LAST_NAME" ASC;
```

- выбираются столбцы `LAST_NAME` и `DEPARTMENT_ID` из таблицы `EMPLOYEES`;
- фильтруются сотрудники, работающие в отделах 20 и 50 с помощью условия `WHERE ... IN`;
- сортировка по фамилии.

Результат выполнения запроса представлен на рисунке 13.

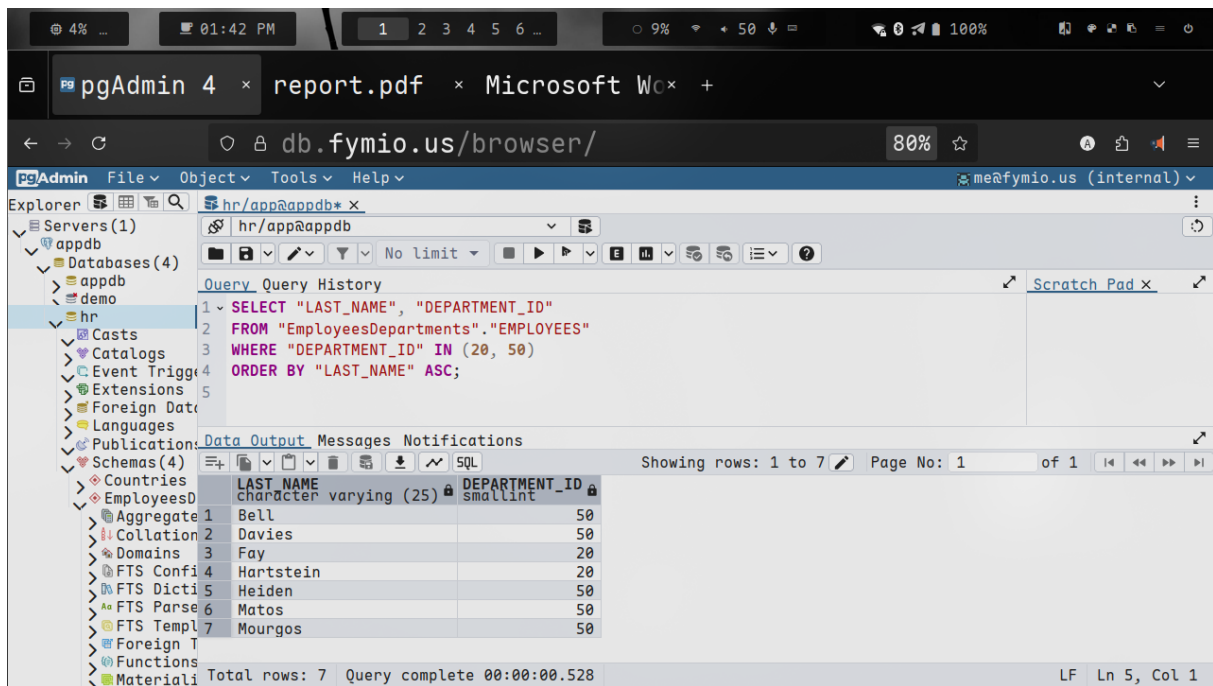


Рис. 13: Результат выполнения запроса.

2.6 Измените запрос из задания 2.3 для вывода фамилий и окладов служащих отделов 20 и 50, зарабатывающих от 5000 до 12000. Назовите столбцы Employee и Mounthly Salary, соответственно. Результат выполнения запроса должен соответствовать таблице 11.

EMPLOYEE	Mounthly Salary
Safwah	5000.00
Mourgos	5800.00
Steiner	8600.00

Табл. 12: Результат выполнения запроса для служащих из отделов 20 и 50, зарабатывающих от 5000 до 12000.

```
SELECT "LAST_NAME" AS "EMPLOYEE", "SALARY" AS "Mounthly Salary"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "DEPARTMENT_ID" IN (20, 50)
AND "SALARY" BETWEEN 5000 AND 12000
ORDER BY "LAST_NAME" ASC;
```

- выбираются столбцы `LAST_NAME` и `SALARY` из таблицы `EMPLOYEES`;
- столбцы переименовываются через `AS` в `EMPLOYEE` и `Mounthly Salary`;
- фильтруются сотрудники, работающие в отделах 20 и 50 с зарплатой от 5000 до 12000 с помощью `WHERE ... AND ... BETWEEN`;
- сортировка по фамилии.

Результат выполнения запроса представлен на рисунке 14.

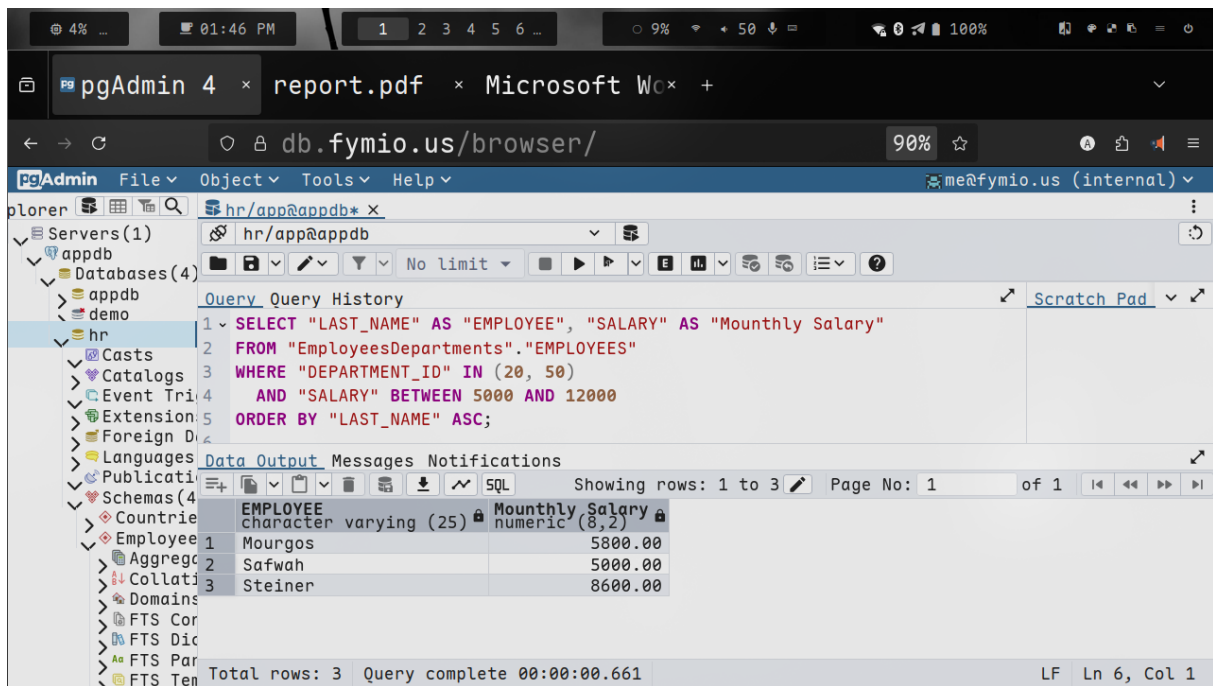


Рис. 14: Результат выполнения запроса.

2.7 Выведите фамилии и должности всех служащих, не имеющих менеджера.

Результат выполнения запроса должен соответствовать таблице 12.

LAST_NAME	JOB_ID
King	AD_PRES

Табл. 13: Результат выполнения запроса для служащих, не имеющих менеджера.

```

SELECT "LAST_NAME", "JOB_ID"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "MANAGER_ID" IS NULL;

```

- выбираются столбцы `LAST_NAME` и `JOB_ID` из таблицы `EMPLOYEES` ;
- фильтруются сотрудники, у которых нет менеджера, с помощью условия `WHERE "MANAGER_ID" IS NULL` ;
- результат содержит только тех сотрудников, которые не подчиняются никому.

Результат выполнения запроса представлен на рисунке 15.

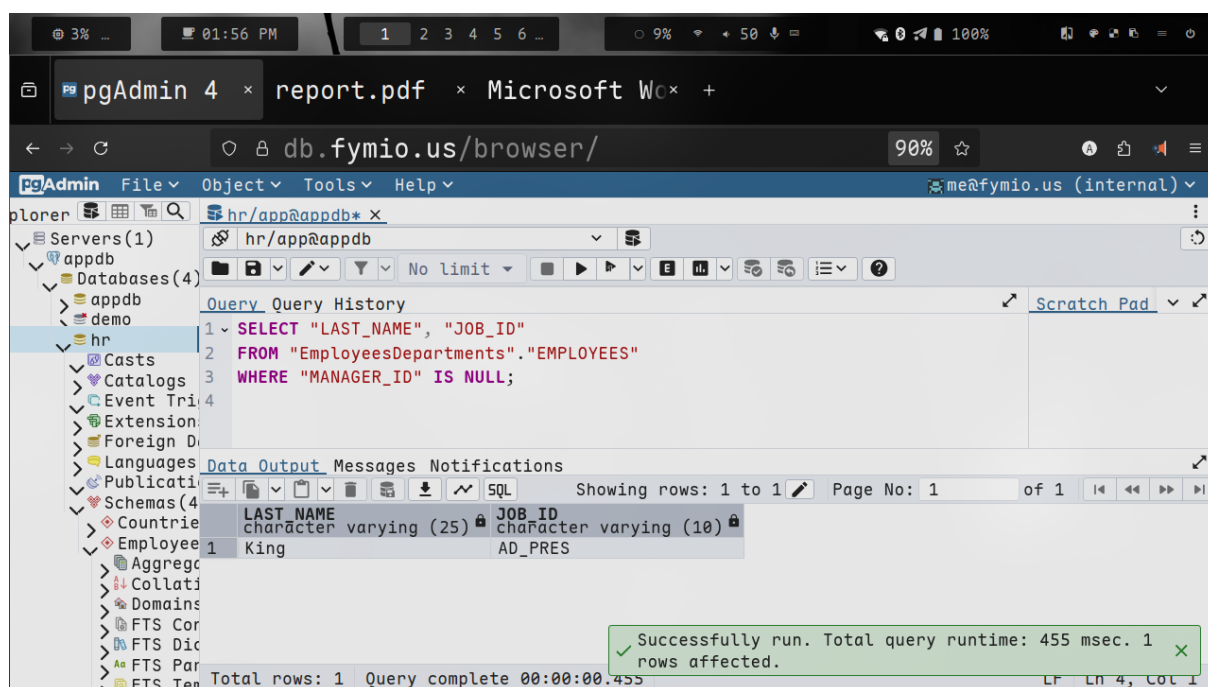


Рис. 15: Результат выполнения запроса.

2.8 Выведите фамилию, и комиссионные всех служащих, зарабатывающих комиссионные. Отсортируйте данные в порядке убывания окладов и комиссионных. Результат выполнения запроса должен соответствовать таблице 13.

LAST_NAME	SALARY	COMMISSION_PCT
Grant	7000.00	0.15
Alves Rocha	7300.00	0.15
Almeida Castro	7300.00	0.20
Silva Pinto	7500.00	0.15
Taylor	8600.00	0.20
Barbosa Souza	9500.00	0.20
Hooper	9600.00	0.20
Zlotkey	10500.00	0.20
Abel	11000.00	0.30

Табл. 14: Результат выполнения запроса для служащих, зарабатывающих комиссионные.

```
SELECT "LAST_NAME", "SALARY", "COMMISSION_PCT"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "COMMISSION_PCT" IS NOT NULL
ORDER BY "SALARY" DESC, "COMMISSION_PCT" DESC;
```

- WHERE "COMMISSION_PCT" IS NOT NULL — отбирает только сотрудников, которые получают комиссионные.
- ORDER BY "SALARY" DESC, "COMMISSION_PCT" DESC — сортировка по зарплате сначала от большей к меньшей, а при равной зарплате — по комиссионным.

Результат выполнения запроса представлен на рисунке 16.

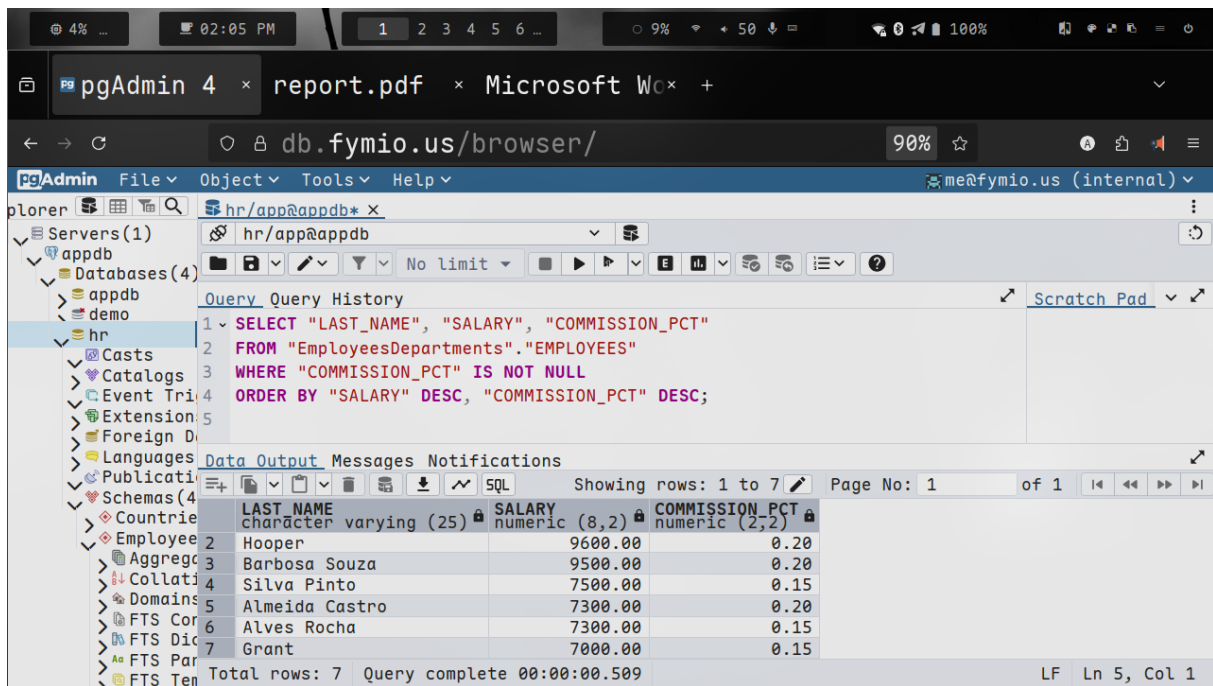


Рис. 16: Результат выполнения запроса.

2.9 Выведите все фамилии служащих БЕЗ использования строковых функций, в которых третья буква - а. Результат выполнения запроса должен соответствовать таблице 14.

LAST_NAME
Grant
Whalen

Табл. 15: Результат выполнения запроса (фамилии служащих, в которых третья буква а).

```
SELECT "LAST_NAME"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "LAST_NAME" LIKE '__a%'
ORDER BY "LAST_NAME" ASC;
```

- используется оператор `LIKE` для поиска шаблона в столбце `LAST_NAME` ;
- `'__a%'` означает:
 - первые две символа могут быть любыми,
 - третий символ — буква 'а',
 - `%` соответствует любому количеству оставшихся символов;
- сортировка по фамилии.

Результат выполнения запроса представлен на рисунке 17.

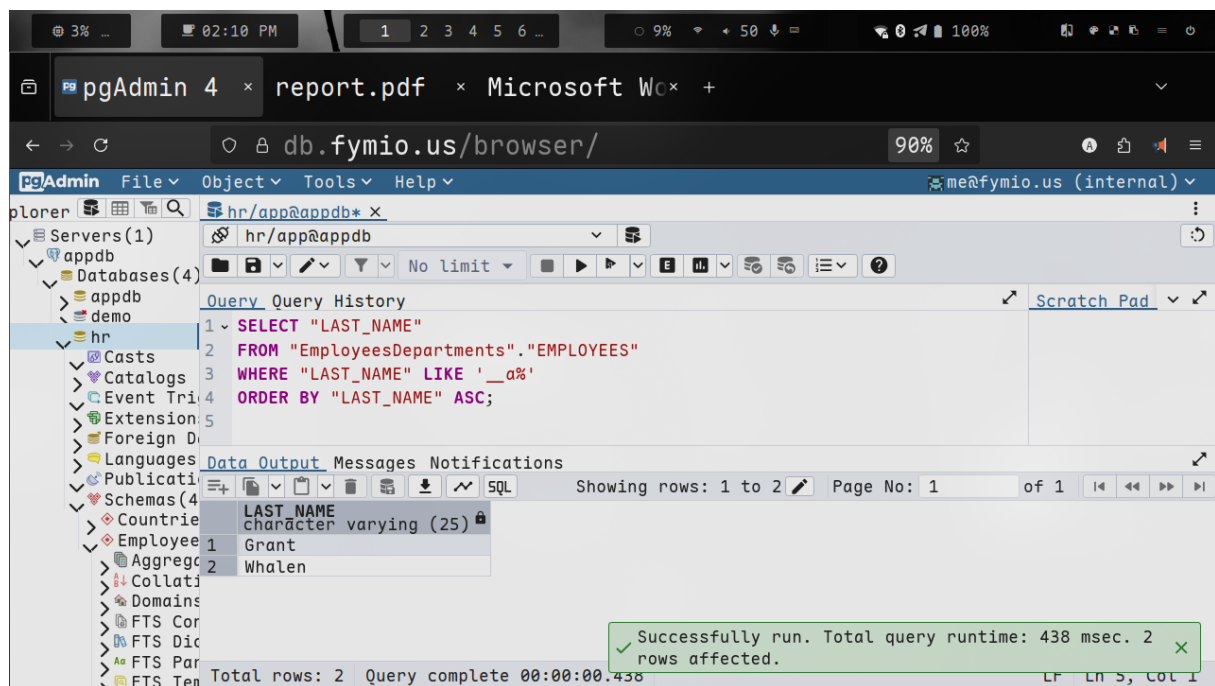


Рис. 17: Результат выполнения запроса.

2.10 Выведите фамилии, должности и оклады всех служащих, работающих торговыми представителями (SA_REP) или клерками на складе (ST_CLERK), с окладом не равным 2600, 3100, 8600 и 11000. Результат выполнения запроса должен соответствовать таблице 15.

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500.00
Vargas	ST_CLERK	2500.00
Grant	SA_REP	7000.0
Silva Pinto	SA_REP	7500.00
Alves Rocha	SA_REP	7300.00
Almeida Castro	SA_REP	7300.00

Табл. 16: Результат выполнения запроса.

```
SELECT "LAST_NAME", "JOB_ID", "SALARY"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "JOB_ID" IN ('SA_REP', 'ST_CLERK')
AND "SALARY" NOT IN (2600, 3100, 8600, 11000)
ORDER BY "LAST_NAME" ASC;
```

- выбираются столбцы LAST_NAME, JOB_ID и SALARY из таблицы EMPLOYEES;
- фильтруются только сотрудники с должностями SA_REP или ST_CLERK с помощью условия WHERE ... IN;
- дополнительно исключаются строки с окладами 2600, 3100, 8600 и 11000 с помощью NOT IN;
- сортировка по фамилии.

Результат выполнения запроса представлен на рисунке 18.

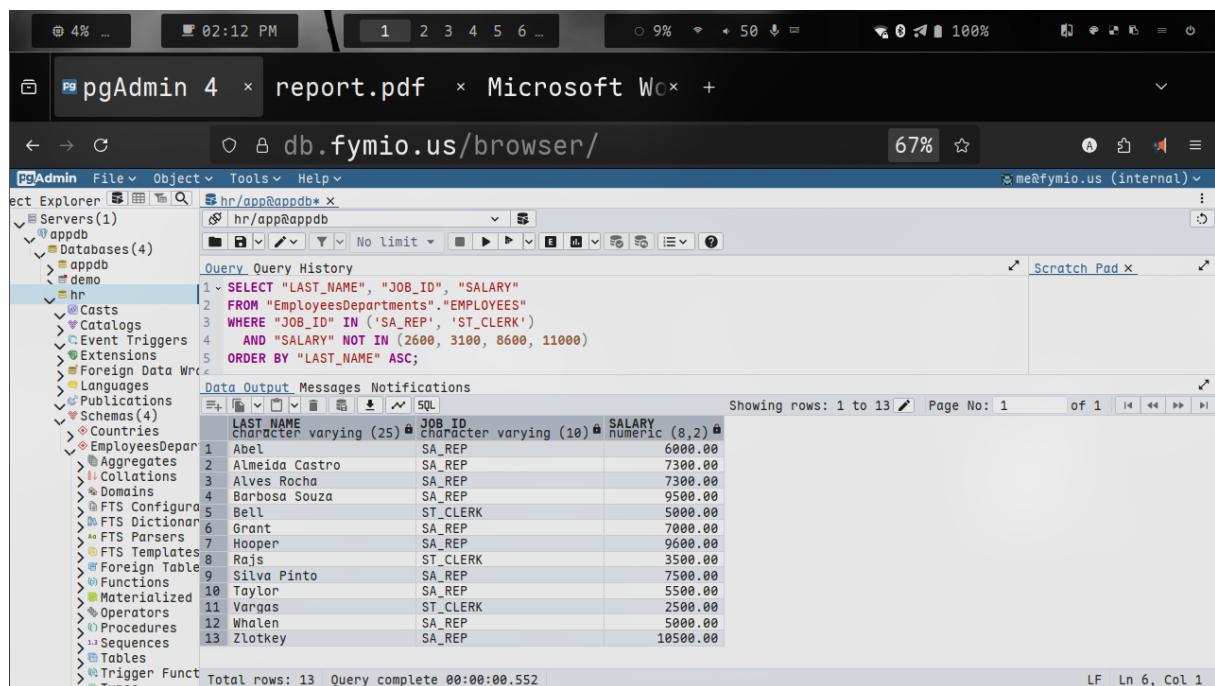


Рис. 18: Результат выполнения запроса.

2.11 Измените запрос 2.8 так, чтобы получить фамилии, оклады и комиссионные всех служащих, у которых сумма комиссионных равна или превышает 20%. Выполните запрос еще раз. Результат выполнения запроса должен соответствовать таблице 16.

Employee	Monthly Salary	COMMISSION_PCT
Almeida Castro	7300.00	0.20
Taylor	8600.00	0.20
Barbosa Souza	9500.00	0.20
Hooper	9600.00	0.20
Zlotkey	10500.00	0.20
Abel	11000.00	0.30

Табл. 17: Результат выполнения запроса.

```

SELECT "LAST_NAME" AS "Employee",
       "SALARY" AS "Monthly Salary",
       "COMMISSION_PCT"
FROM "EmployeesDepartments"."EMPLOYEES"
WHERE "COMMISSION_PCT" >= 0.20
ORDER BY "LAST_NAME" ASC;

```

- выбираются фамилии сотрудников `LAST_NAME`, месячная зарплата `SALARY` и процент комиссионных `COMMISSION_PCT`;
- фильтруются только те сотрудники, у которых `COMMISSION_PCT` ≥ 0.20 , то есть сумма комиссионных равна или превышает 20%;
- столбцы переименованы с помощью `AS`;
- сортировка по фамилии.

Результат выполнения запроса представлен на рисунке 19.

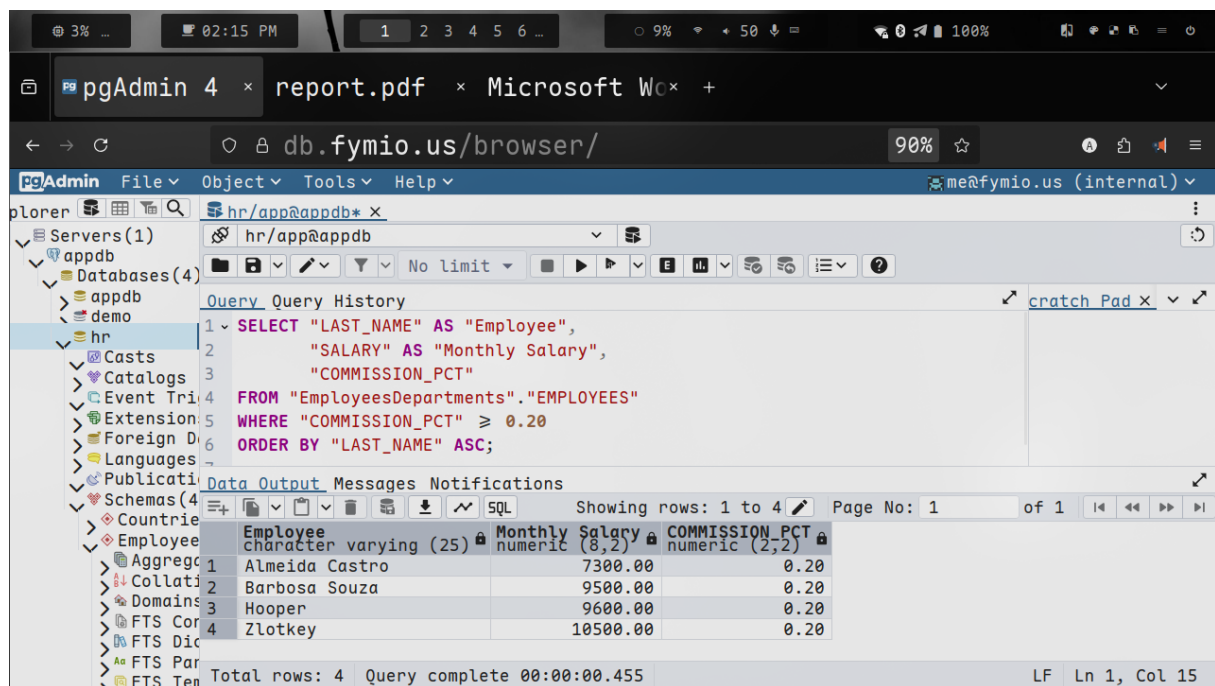


Рис. 19: Результат выполнения запроса.

Задание 3. Составление запросов, требующих использования числовых функций (TRUNC, ROUND и т.д.)

Выводы и анализ результатов работы