

**Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”
ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5
«Выполнение запросов в PostgreSQL»**

Специальность 09.02.07 «Информационные системы и программирование»

Дисциплина «Основы проектирования баз данных»

Преподаватель:

Говоров А.И. _____

«__» _____ 2020 г.

Оценка _____

Выполнил:

студент группы У2437

Пировский В.И.

Санкт-Петербург
2019/2020

Цель работы: овладеть практическими навыками проведения анализа данных системы и построения инфологической модели данных БД.

Практическое задание: проанализировать предметную область согласно варианту задания. Выполнить инфологическое моделирование базы данных по заданной предметной области с использованием метода ER-диаграмм («сущность-связь») в нотации Питера Чена.

Индивидуальное задание: Создать программную систему, предназначенную для управления договорами страхования с физическими лицами и юридическими организациями. Страховая организация заключает договоры. Для организации оформляется коллективный договор, в котором перечислены страхуемые сотрудники: ФИО, возраст, категория риска (первая, вторая, высшая и т.п.). О предприятии хранится следующая информация: код, полное наименование, краткое наименование, адрес, банковские реквизиты (номер банка), специализация предприятия (медицинское учреждение, автотранспортное предприятие, учебное заведение и т.п.). В заключаемом коллективном договоре указывается дата заключения, срок договора (начало и конец действия договора), сумма выплат по каждой категории сотрудников, выплаты по страховым случаям. Выплаты зависят от категории сотрудника. Необходимо также хранить информацию о страховом агенте, заключившем договор (ФИО, паспортные данные, контактные данные). Каждый агент может заключить много договоров, в каждом договоре может быть оформлено несколько сотрудников. С физическим лицом заключается индивидуальный договор. Каждый конкретный договор может быть заключен только одним агентом. При возникновении страхового случая необходима информация о его дате, причине, решении о выплате страховой суммы и размере выплаты. Директор компании должен иметь возможность принять и уволить на работу страхового агента. Поэтому должна сохраняться информация о заключенных с ними трудовых договорах.

Название создаваемой БД – Insurance

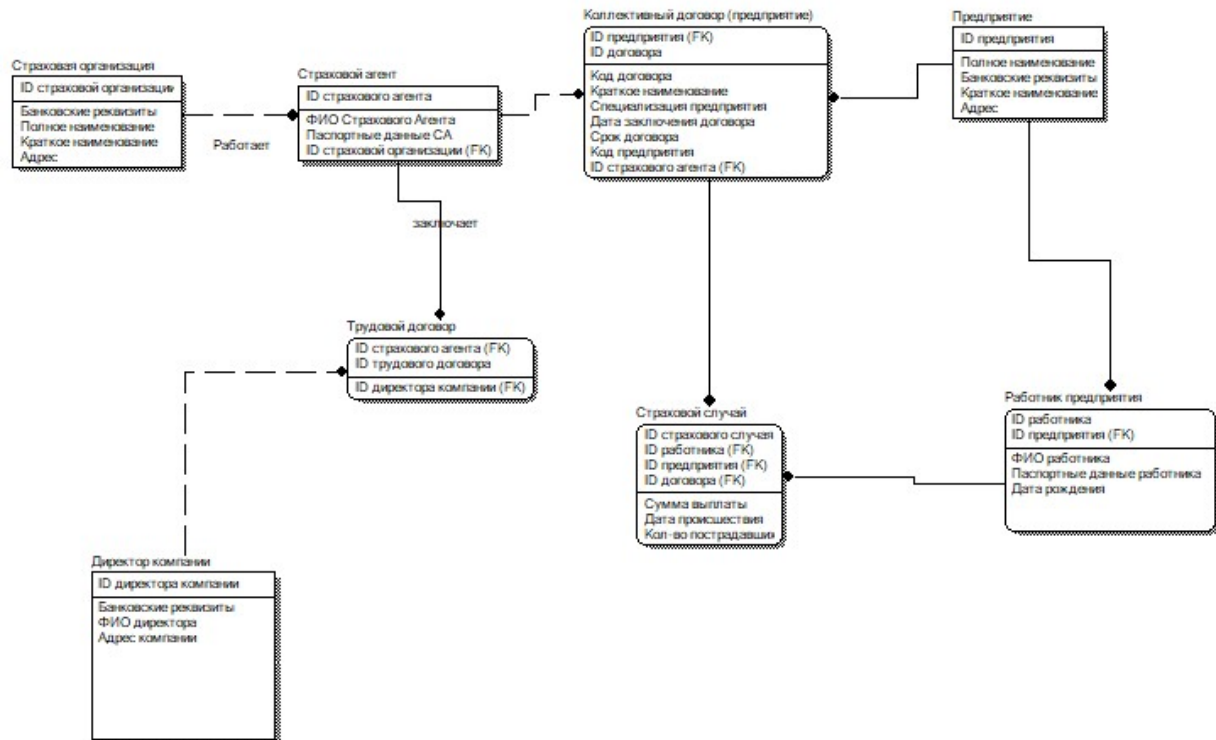
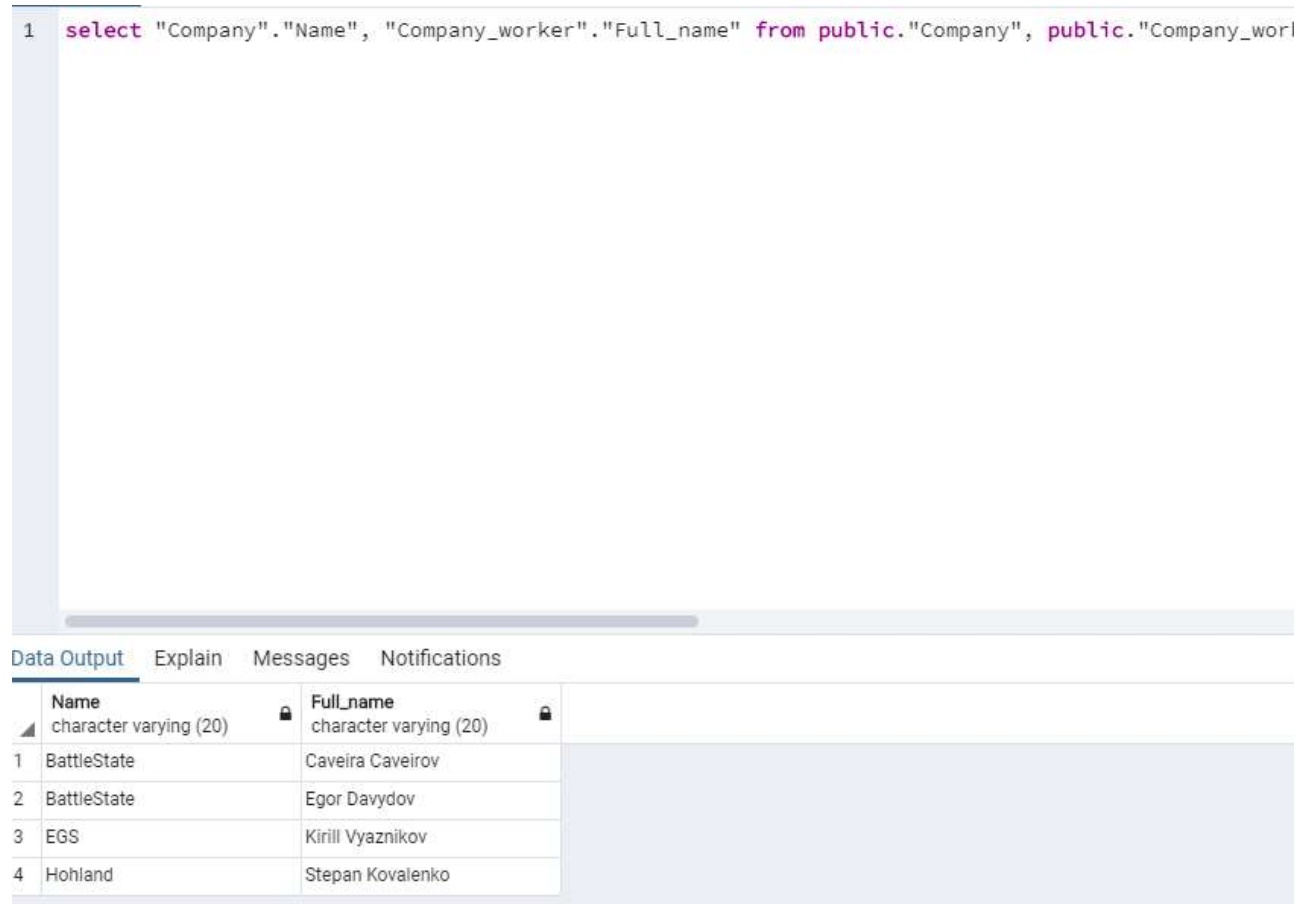


Рисунок 1. Схема инфологической модели данных БД, выполненная в среде CA ERwin Data Modeler.

1. Выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;

```
select "Company"."Name", "Company_worker"."Full_name" from public."Company", public."Company_worker" where "Company"."ID_company"="Company_worker"."ID_company" order by "Company_worker"."Full_name"
```



The screenshot shows a SQL query execution interface. At the top, the query is entered in a text area. Below the query, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, displaying a table with two columns: 'Name' and 'Full_name'. The table contains four rows of data.

	Name character varying (20)	Full_name character varying (20)
1	BattleState	Caveira Caveirov
2	BattleState	Egor Davydov
3	EGS	Kirill Vyaznikov
4	Hohland	Stepan Kovalenko

2. Использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;

select * from public."Company_worker" where ("ID_Company_worker">1
and "ID_company"<3)

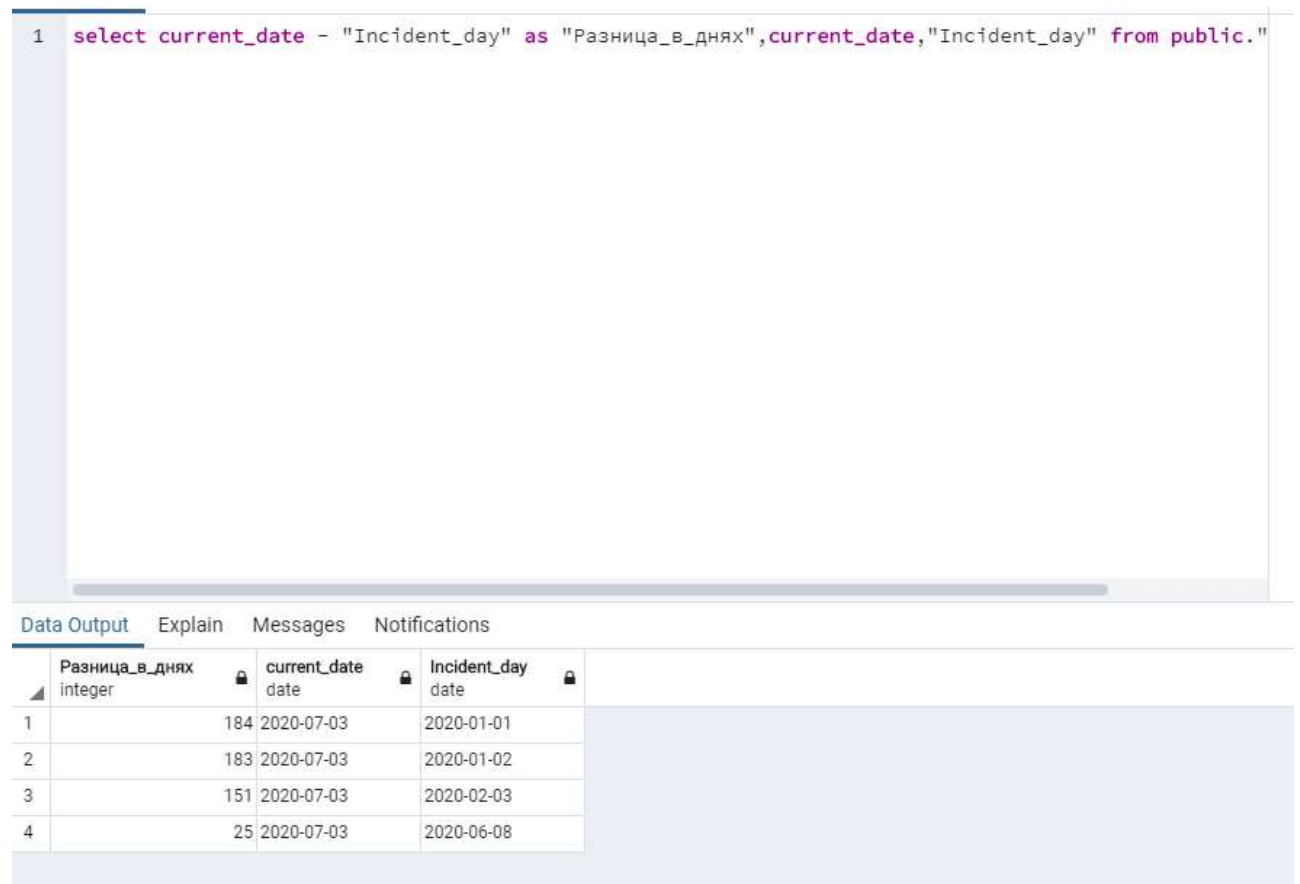
```
1 select * from public."Company_worker" where ("ID_Company_worker">1 and "ID_company"<3)
```

Data Output Explain Messages Notifications

	ID_Company_worker [PK] integer	ID_company integer	Full_name character varying (20)	Passport_data character varying (20)
1		2	1 Caveira Cavelrov	890-09-70
2		3	2 Kirill Vyaznikov	456-87-23

3. Использование функций для работы с датами – от 2 баллов;

`select current_date - "Incident_day" as "Разница_в_днях",current_date,"Incident_day" from public."Insurance_case"`



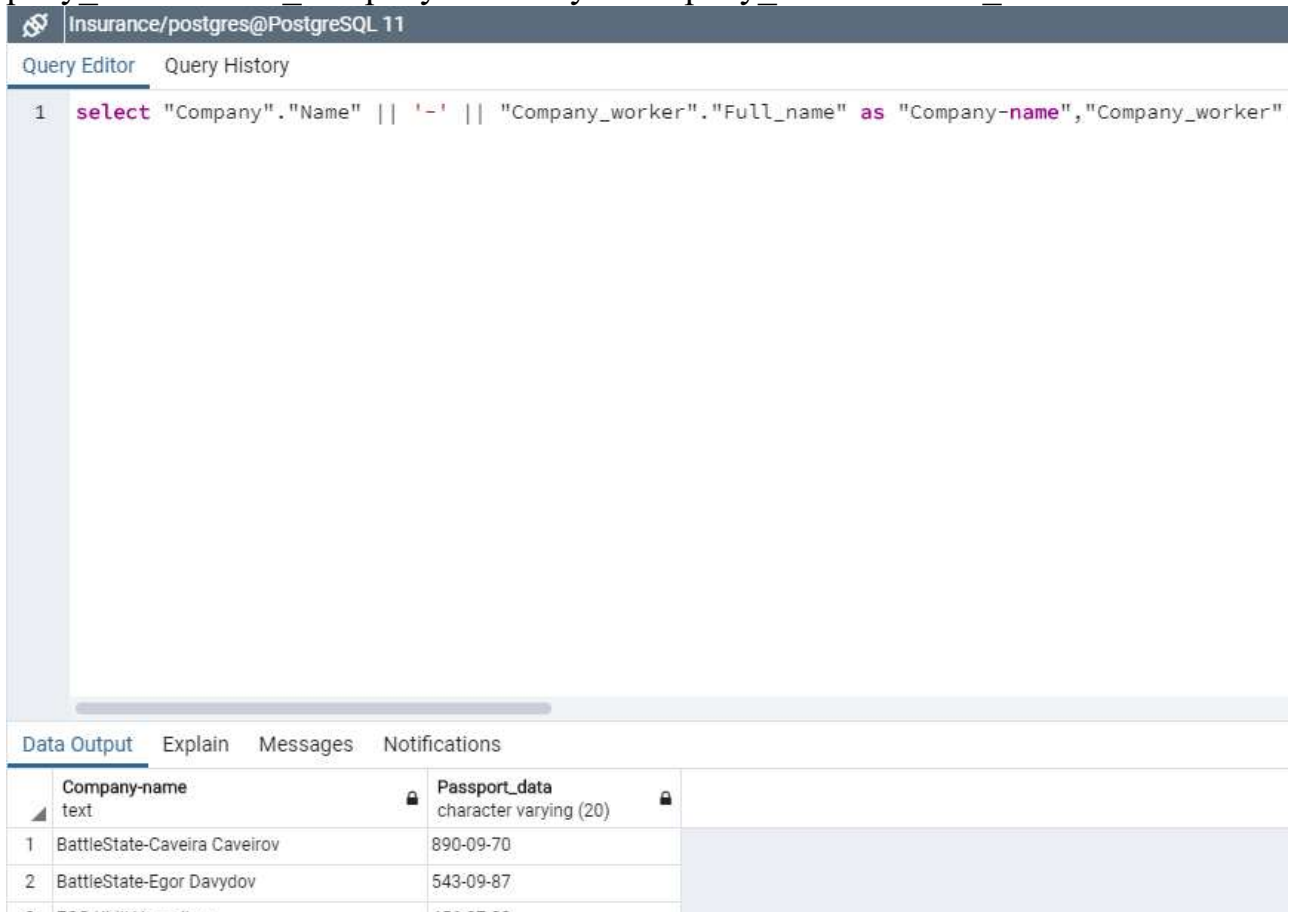
The screenshot shows a SQL query execution interface. At the top, the query is entered in a text area: `1 select current_date - "Incident_day" as "Разница_в_днях",current_date,"Incident_day" from public."`. Below the query area, there are tabs for "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected, displaying a table with the results of the query. The table has four columns: "Разница_в_днях" (integer), "current_date" (date), and "Incident_day" (date). The results are as follows:

	Разница_в_днях integer	current_date date	Incident_day date
1	184	2020-07-03	2020-01-01
2	183	2020-07-03	2020-01-02
3	151	2020-07-03	2020-02-03
4	25	2020-07-03	2020-06-08

с."Book"

4. Использование строковых функций – от 3 баллов;

```
select "Company"."Name" || '-' || "Company_worker"."Full_name" as "Company-name", "Company_worker"."Passport_data" from public."Company", public."Company_worker" where "Company"."ID_company"="Company_worker"."ID_company" order by "Company_worker"."Full_name"
```



The screenshot shows a PostgreSQL Query Editor interface. The top bar indicates the connection is to 'Insurance/postgres@PostgreSQL 11'. Below the bar are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the SQL query from the previous block. Below the query editor is a horizontal bar with tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'Company-name' (text) and 'Passport_data' (character varying (20)). There are two rows of data.

	Company-name	Passport_data
1	BattleState-Caveira Caveirov	890-09-70
2	BattleState-Egor Davydov	543-09-87

5. запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 3 баллов);

```
select * from public."Company_worker" where ("ID_Company_worker">1  
and "ID_company" in (select "ID_company" from public."Company" where  
"ID_company"<3))
```

Insurance/postgres@PostgreSQL 11

Query Editor Query History

```
1 ny_worker">1 and "ID_company" in (select "ID_company" from public."Company" where "ID_company"<3))
```

Data Output Explain Messages Notifications

	ID_Company_worker [PK] integer	ID_company integer	Full_name character varying (20)	Passport_data character varying (20)	
1		2	1 Caveira Caveirov	890-09-70	
2		3	2 Kirill Vyaznikov	456-87-23	

6. вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 3 баллов);

Select distinct max("ID_company") from public."Company";

Query Editor

Query History

1

Select distinct max("ID_company") from public."Company";

Data Output

Explain

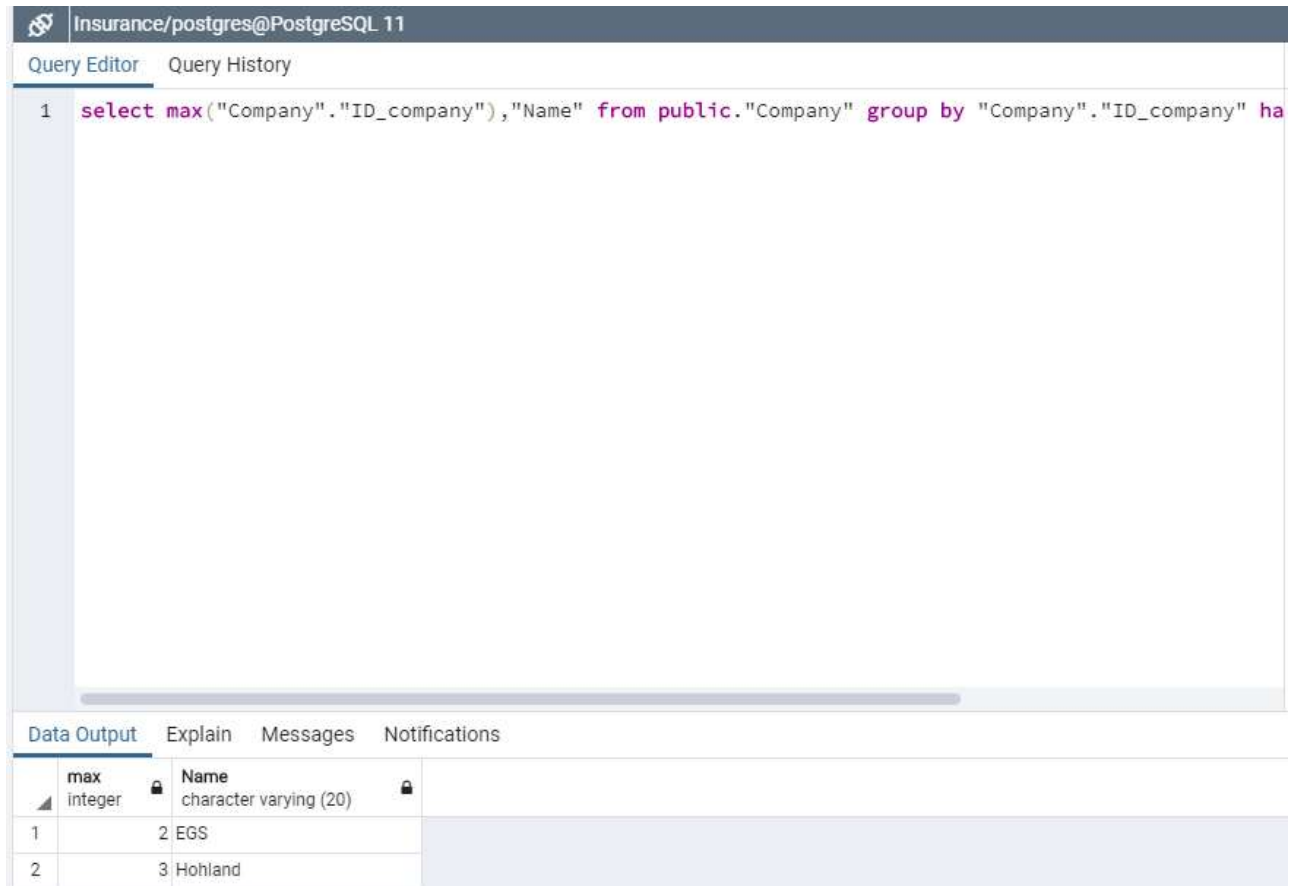
Messages

Notifications

	max integer	
1		3

7. вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов;

```
select max("Company"."ID_company"), "Name" from public."Company"  
group by "Company"."ID_company" having max("Company"."ID_company")  
> 1
```




The screenshot shows a PostgreSQL query editor interface. The top bar indicates the connection is to 'Insurance/postgres@PostgreSQL 11'. Below the bar are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query. Below the query editor is a horizontal bar with tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'max' (integer) and 'Name' (character varying (20)). The table contains two rows of data.

max	Name
1	EGS
2	Hohland

8. использование предикатов EXISTS, ALL, SOME и ANY - от 4 баллов;

```
select * from public."Company_worker" where ("ID_Company_worker">1  
and "ID_company" = any (select "ID_company" from public."Company"
```

where "ID_company"<3))

 Insurance/postgres@PostgreSQL 11

Query Editor

Query History

1 worker">1 and "ID_company" = any (select "ID_company" from public."Company" where "ID_company"<3))

Data Output

Explain

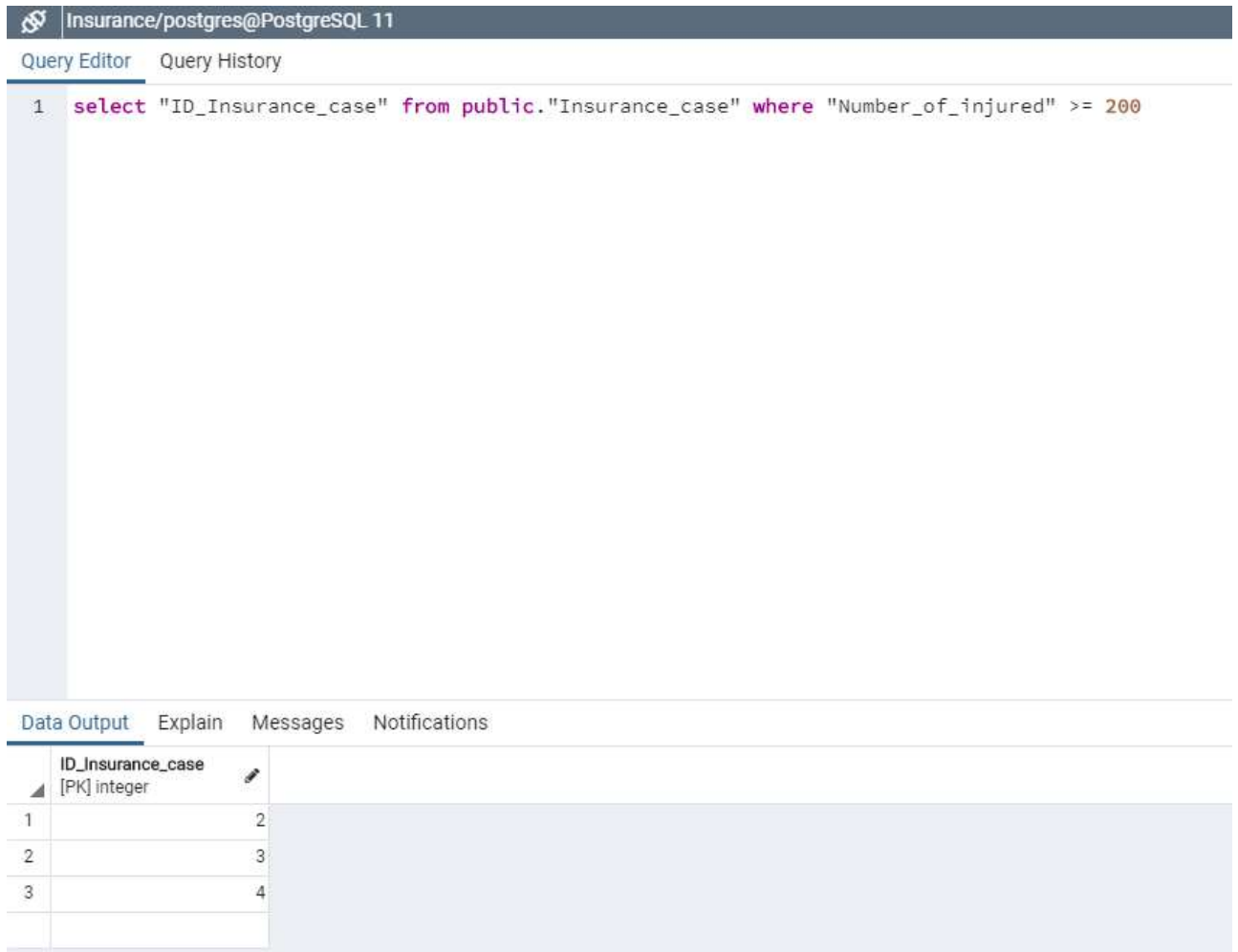
Messages

Notifications

	ID_Company_worker [PK] integer	ID_company integer	Full_name character varying (20)	Passport_data character varying (20)	
1		2	1 Caveira Caveirov	890-09-70	
2		3	2 Kirill Vyaznikov	456-87-23	

9. использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 3 баллов;

```
select "ID_Insurance_case" from public."Insurance_case" where "Number_of_injured" >= 200;
```

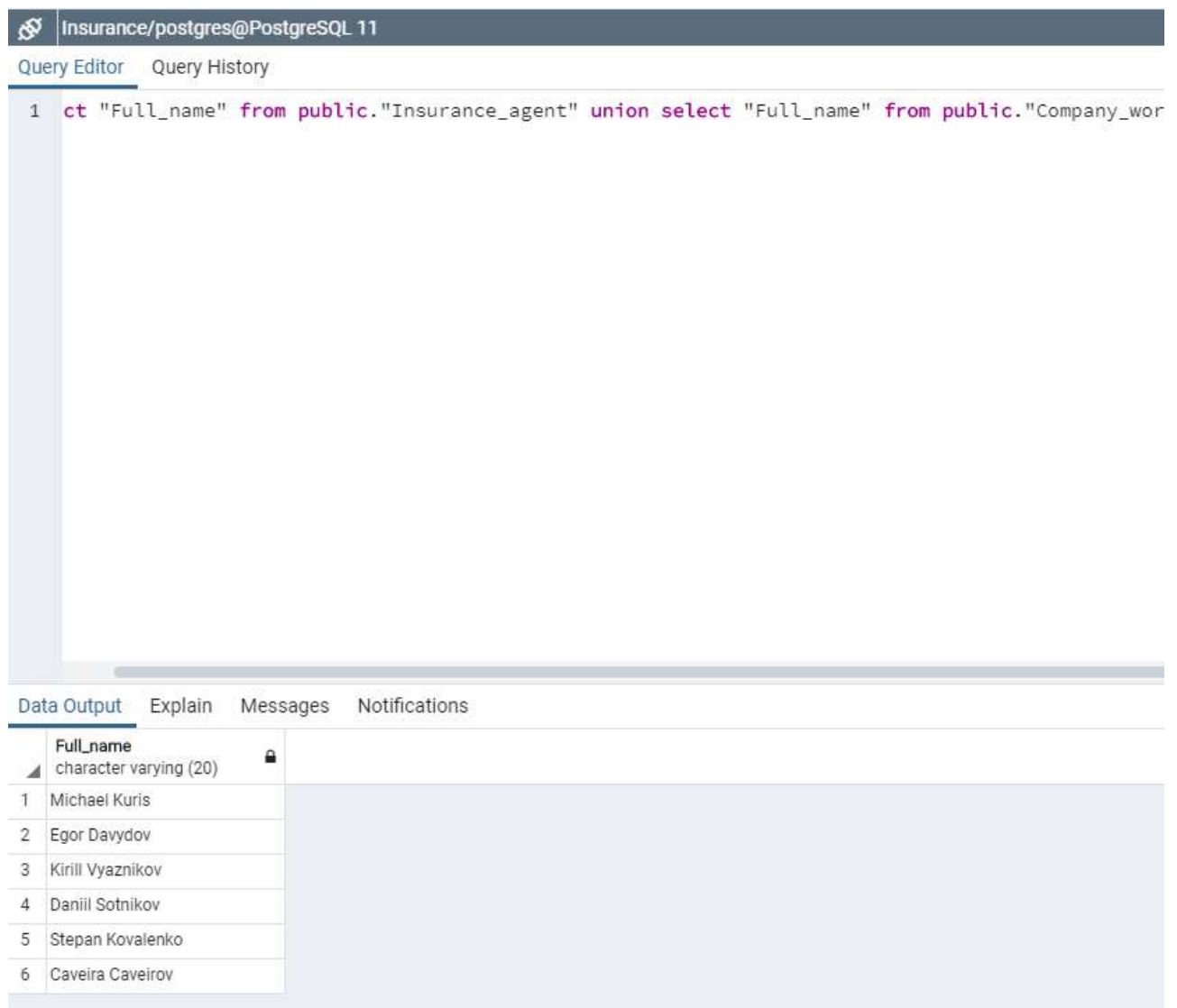


The screenshot shows a PostgreSQL query editor interface. The top bar indicates the connection is to 'Insurance/postgres@PostgreSQL 11'. Below the bar are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query: `1 select "ID_Insurance_case" from public."Insurance_case" where "Number_of_injured" >= 200`. Below the query editor are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'ID_Insurance_case' (labeled as '[PK] integer') and an unlabeled column. The table contains three rows of data: (1, 2), (2, 3), and (3, 4).

ID_Insurance_case [PK] integer	
1	2
2	3
3	4

10.использование объединений запросов (inner join и т.д.) - от 3 баллов.

```
select "Full_name" from public."Insurance_agent" union select "Full_name"  
from public."Company_worker"
```



The screenshot shows a PostgreSQL query editor interface. At the top, the connection is identified as 'Insurance/postgres@PostgreSQL 11'. Below this, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query that uses a UNION to combine results from two tables: 'Insurance_agent' and 'Company_worker'. The query is as follows:

```
1  ct "Full_name" from public."Insurance_agent" union select "Full_name" from public."Company_wor
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with one column, 'Full_name', which is a character varying (20) type. The table contains six rows of data:

	Full_name
1	Michael Kuris
2	Egor Davydov
3	Kirill Vyaznikov
4	Daniil Sotnikov
5	Stepan Kovalenko
6	Caveira Caveirov