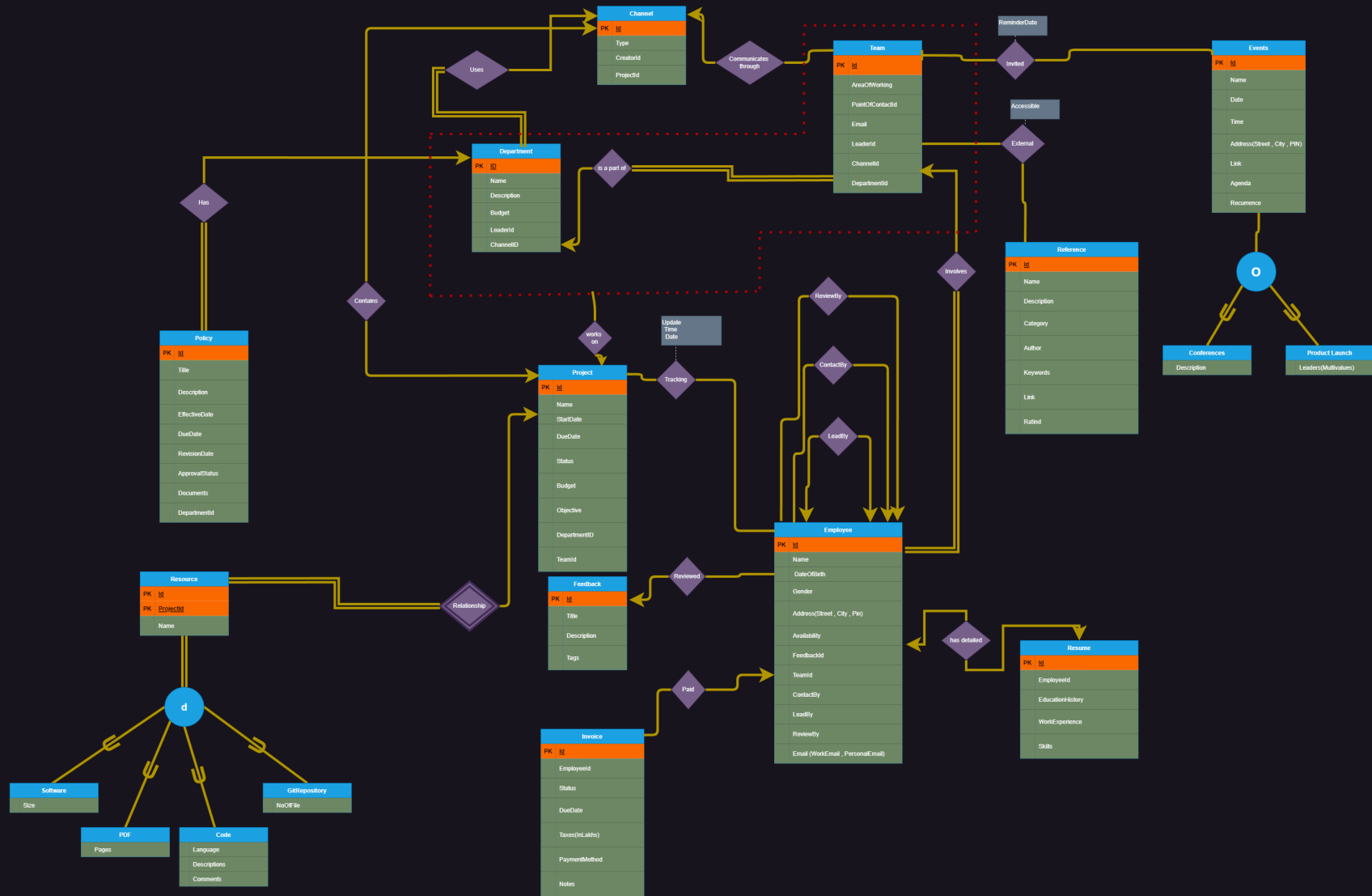
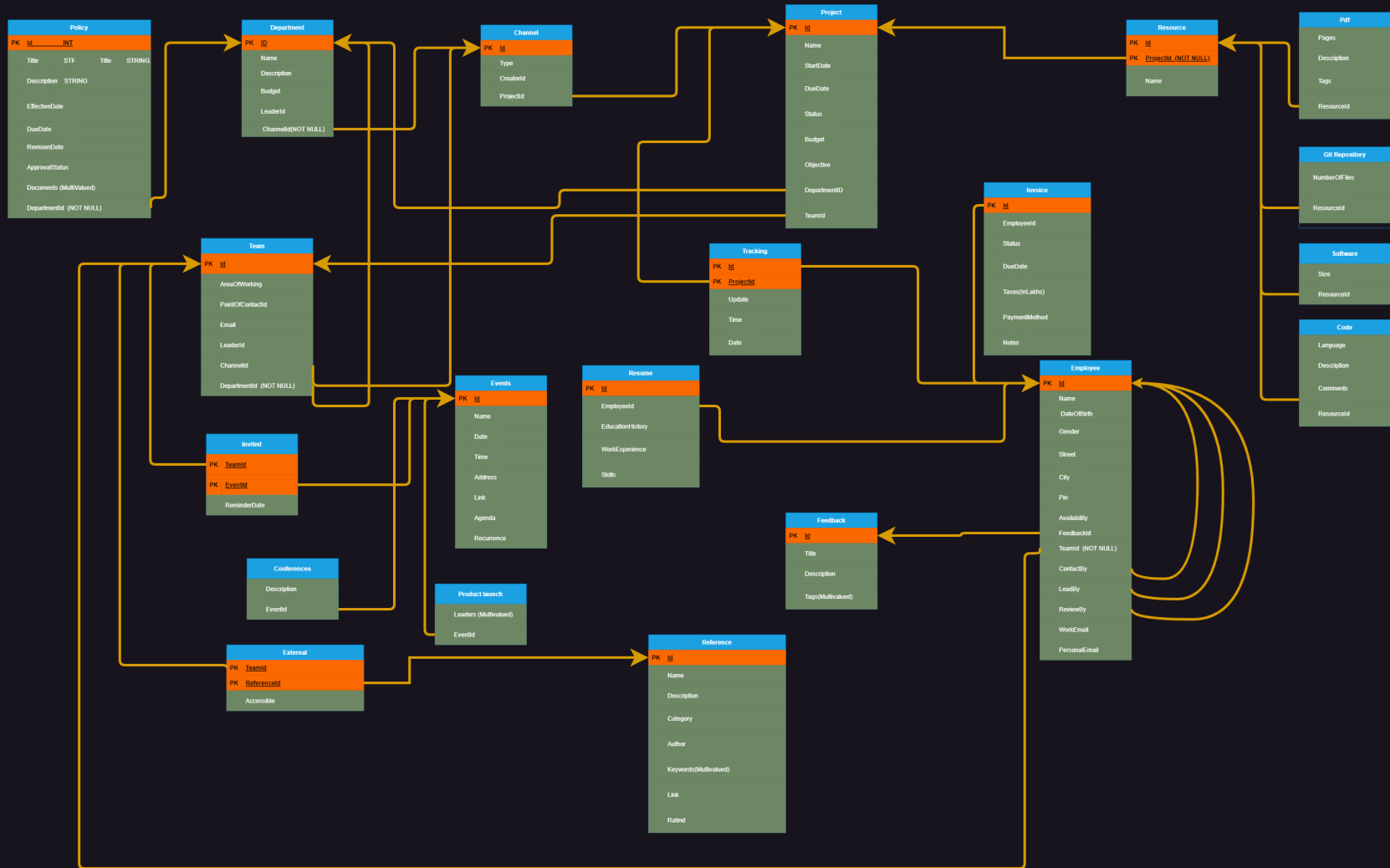

CollabNet

One stop destination for collaboration and networking for a company.

ER DIAGRAM

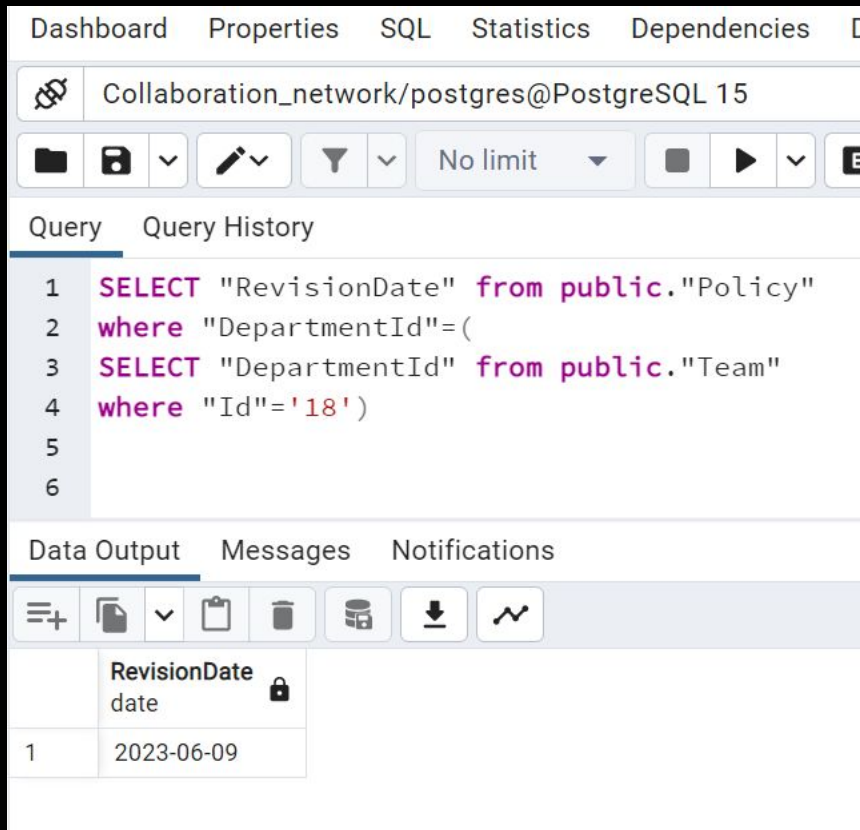


Relational Schema



Queries

Select the Revision Date of Policies of department in which team 18 exists.



The screenshot shows a database interface with a query editor and a results table. The query editor displays a SQL query that selects the RevisionDate from the Policy table, filtered by DepartmentId, which is further filtered by TeamId = 18. The results table shows a single row with the RevisionDate 2023-06-09.

Dashboard Properties SQL Statistics Dependencies D

Collaboration_network/postgres@PostgreSQL 15

No limit

Query Query History

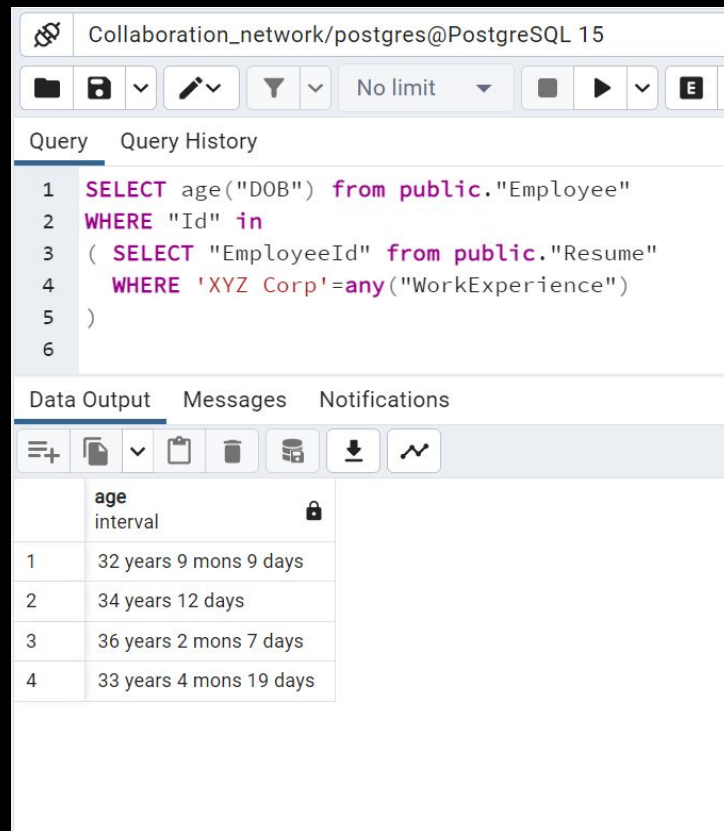
```
1 SELECT "RevisionDate" from public."Policy"
2 where "DepartmentId"=(
3 SELECT "DepartmentId" from public."Team"
4 where "Id"='18')
5
6
```

Data Output Messages Notifications

	RevisionDate date
1	2023-06-09

Queries

Print the age of all the employees who
have worked for 'XYZ Corp' in the
past.



The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is to 'Collaboration_network/postgres@PostgreSQL 15'. The 'Query' tab is active, displaying the following SQL query:

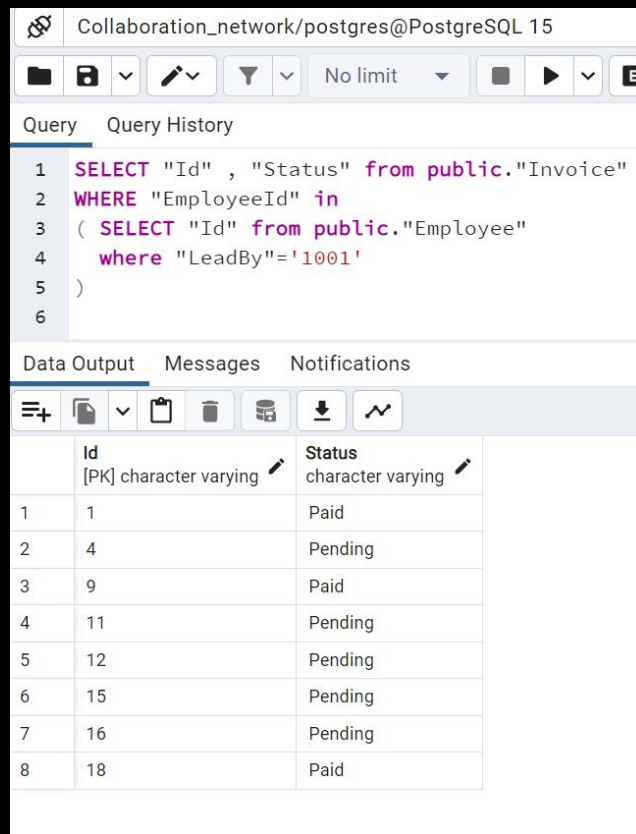
```
1 SELECT age("DOB") from public."Employee"  
2 WHERE "Id" in  
3 ( SELECT "EmployeeId" from public."Resume"  
4   WHERE 'XYZ Corp'=any("WorkExperience")  
5 )  
6
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'age' and 'interval'. The 'age' column is highlighted, and the 'interval' column is shown as a lock icon. The table contains four rows of data:

	age	interval
1	32 years 9 mons 9 days	
2	34 years 12 days	
3	36 years 2 mons 7 days	
4	33 years 4 mons 19 days	

Queries

Print the Id and Status of the Invoice of
the employees who are lead by
Employee with Id “1001”.



The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is to 'Collaboration_network/postgres@PostgreSQL 15'. The 'Query' tab is active, displaying the following SQL query:

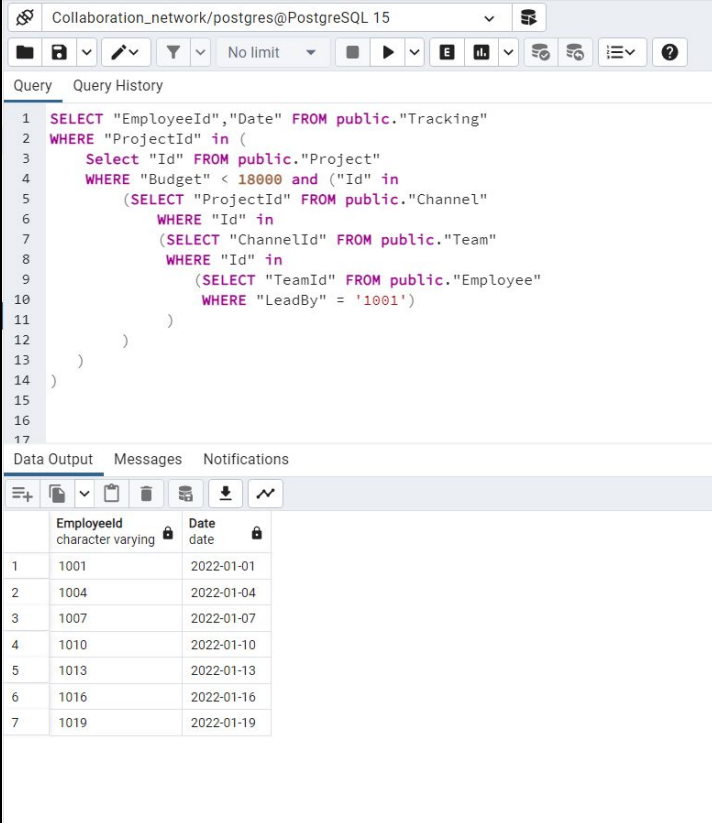
```
1 SELECT "Id" , "Status" from public."Invoice"  
2 WHERE "EmployeeId" in  
3 ( SELECT "Id" from public."Employee"  
4   where "LeadBy"='1001'  
5 )  
6
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'Id' (labeled as [PK] character varying) and 'Status' (character varying). There are 8 rows of data.

	Id [PK] character varying	Status character varying
1	1	Paid
2	4	Pending
3	9	Paid
4	11	Pending
5	12	Pending
6	15	Pending
7	16	Pending
8	18	Paid

Queries

Find the projects on which those employees have worked who have been lead by the employee with id '1001'. Among those projects , select the ones having budget less than 18000. Then , print the EmployeeId of those employees who have updated that project and also print the date of updation.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

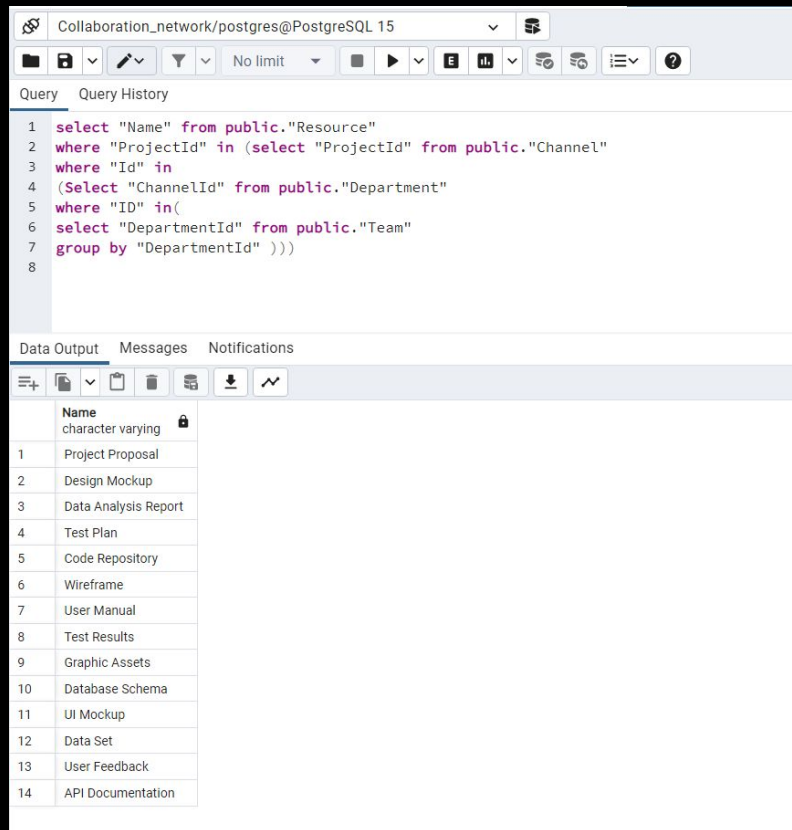
```
1 SELECT "EmployeeId","Date" FROM public."Tracking"
2 WHERE "ProjectId" in (
3     Select "Id" FROM public."Project"
4     WHERE "Budget" < 18000 and ("Id" in
5         (SELECT "ProjectId" FROM public."Channel"
6             WHERE "Id" in
7                 (SELECT "ChannelId" FROM public."Team"
8                     WHERE "Id" in
9                         (SELECT "TeamId" FROM public."Employee"
10                            WHERE "LeadBy" = '1001')
11                     )
12                 )
13             )
14         )
15     )
16 )
17 )
```

The results are displayed in a table with the following data:

	EmployeeId character varying	Date date
1	1001	2022-01-01
2	1004	2022-01-04
3	1007	2022-01-07
4	1010	2022-01-10
5	1013	2022-01-13
6	1016	2022-01-16
7	1019	2022-01-19

Queries

Print the name of the resources of any project on which a department's teams are working.



The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is to 'Collaboration_network/postgres@PostgreSQL 15'. The query editor contains the following SQL query:

```
1 select "Name" from public."Resource"  
2 where "ProjectId" in (select "ProjectId" from public."Channel"  
3 where "Id" in  
4 (select "ChannelId" from public."Department"  
5 where "ID" in(  
6 select "DepartmentId" from public."Team"  
7 group by "DepartmentId" )))  
8
```

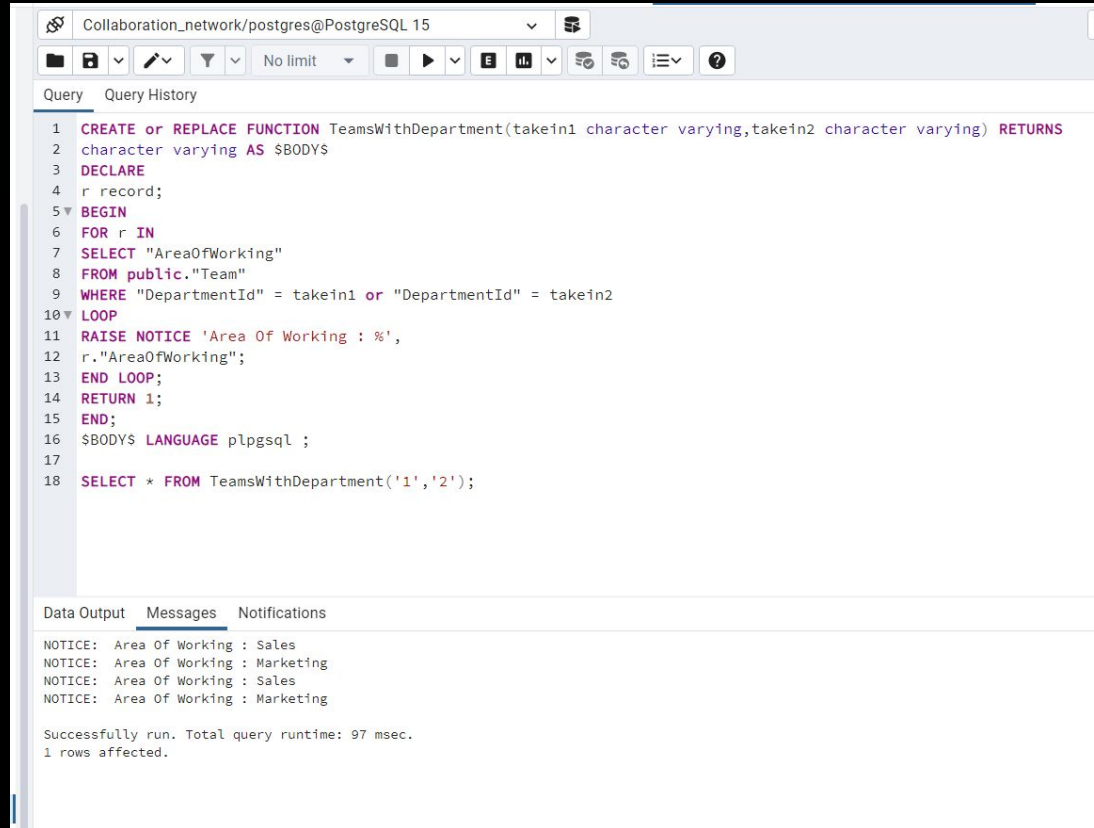
Below the query editor, the 'Data Output' tab is active, displaying a table with 14 rows of results. The table has a single column named 'Name' with a data type of 'character varying' and a lock icon. The results are as follows:

	Name
1	Project Proposal
2	Design Mockup
3	Data Analysis Report
4	Test Plan
5	Code Repository
6	Wireframe
7	User Manual
8	Test Results
9	Graphic Assets
10	Database Schema
11	UI Mockup
12	Data Set
13	User Feedback
14	API Documentation

Stored Procedures

Print the Area of working of temas

belonging to the department 1 or 2



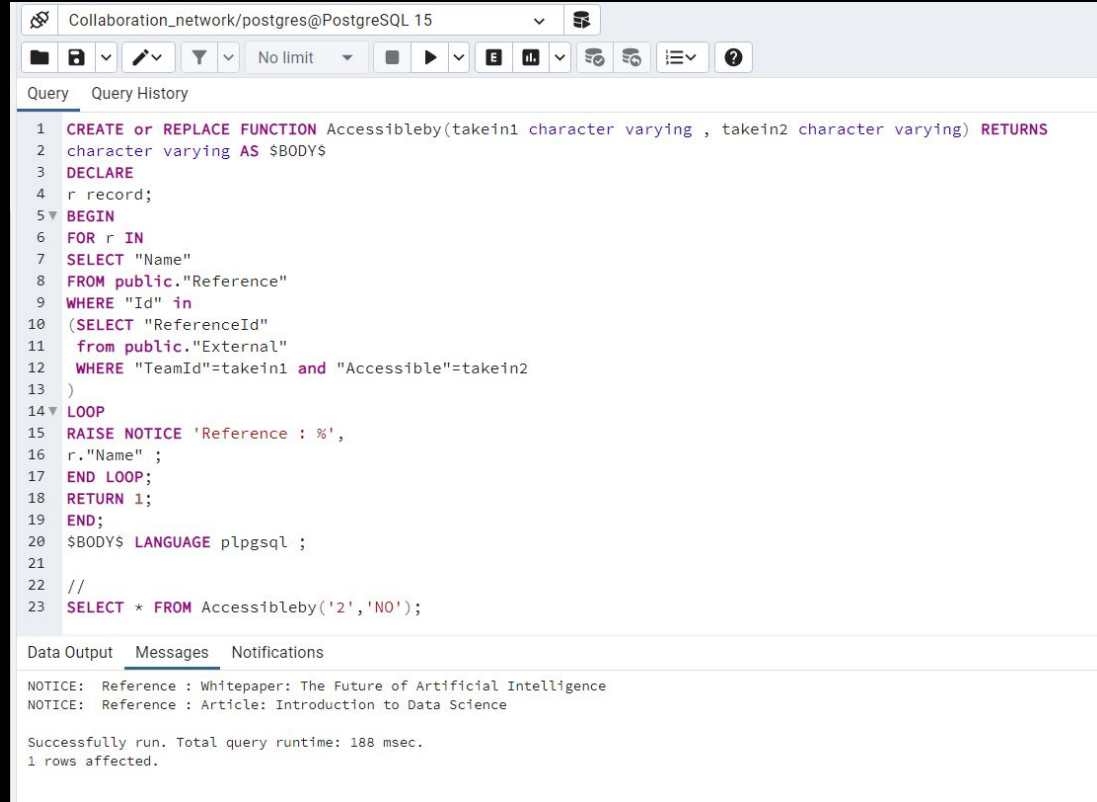
```
1 CREATE OR REPLACE FUNCTION TeamsWithDepartment(takein1 character varying,takein2 character varying) RETURNS
2 character varying AS $BODY$
3 DECLARE
4 r record;
5 BEGIN
6 FOR r IN
7 SELECT "AreaOfWorking"
8 FROM public."Team"
9 WHERE "DepartmentId" = takein1 or "DepartmentId" = takein2
10 LOOP
11 RAISE NOTICE 'Area Of Working : %',
12 r."AreaOfWorking";
13 END LOOP;
14 RETURN 1;
15 END;
16 $BODY$ LANGUAGE plpgsql ;
17
18 SELECT * FROM TeamsWithDepartment('1','2');
```

NOTICE: Area Of Working : Sales
NOTICE: Area Of Working : Marketing
NOTICE: Area Of Working : Sales
NOTICE: Area Of Working : Marketing

Successfully run. Total query runtime: 97 msec.
1 rows affected.

Stored Procedures

You will be given two inputs : One will be teamId and other will be a boolean value (YES or NO). Print the references' names that are accessible/not accessible by that Team number (according to the input being yes or no)



```
1 CREATE or REPLACE FUNCTION Accessibleby(takein1 character varying , takein2 character varying) RETURNS
2 character varying AS $BODY$
3 DECLARE
4 r record;
5 BEGIN
6 FOR r IN
7 SELECT "Name"
8 FROM public."Reference"
9 WHERE "Id" in
10 (SELECT "ReferenceId"
11  from public."External"
12  WHERE "TeamId"=takein1 and "Accessible"=takein2
13 )
14 LOOP
15 RAISE NOTICE 'Reference : %',
16 r."Name" ;
17 END LOOP;
18 RETURN 1;
19 END;
20 $BODY$ LANGUAGE plpgsql ;
21
22 //
23 SELECT * FROM Accessibleby('2','NO');
```

NOTICE: Reference : Whitepaper: The Future of Artificial Intelligence
NOTICE: Reference : Article: Introduction to Data Science

Successfully run. Total query runtime: 188 msec.
1 rows affected.