

Homework step 1

Diagrams Text

departments

-

dept_no varchar(10) PK

dept_name" varchar(100) NOT NULL,

last_updated timestamp

dept_emp

-

id serial PK

emp_no int FK >- employees.emp_no

dept_no varchar(10) FK >- departments.dept_no

last_updated timestamp

dept_manager

-

id serial PK

dept_no varchar(10) FK >- departments.dept_no

emp_no int FK >- employees.emp_no

last_updated timestamp

employees

-

emp_no int PK

emp_title_id varchar(10) FK >- titles.title_id

birth_date date

first_name varchar(100)

last_name varchar(100)

sex varchar(50)

hire_date date

last_update timestamp

titles

-

title_id varchar(50) PK

title varchar(100)

salaries

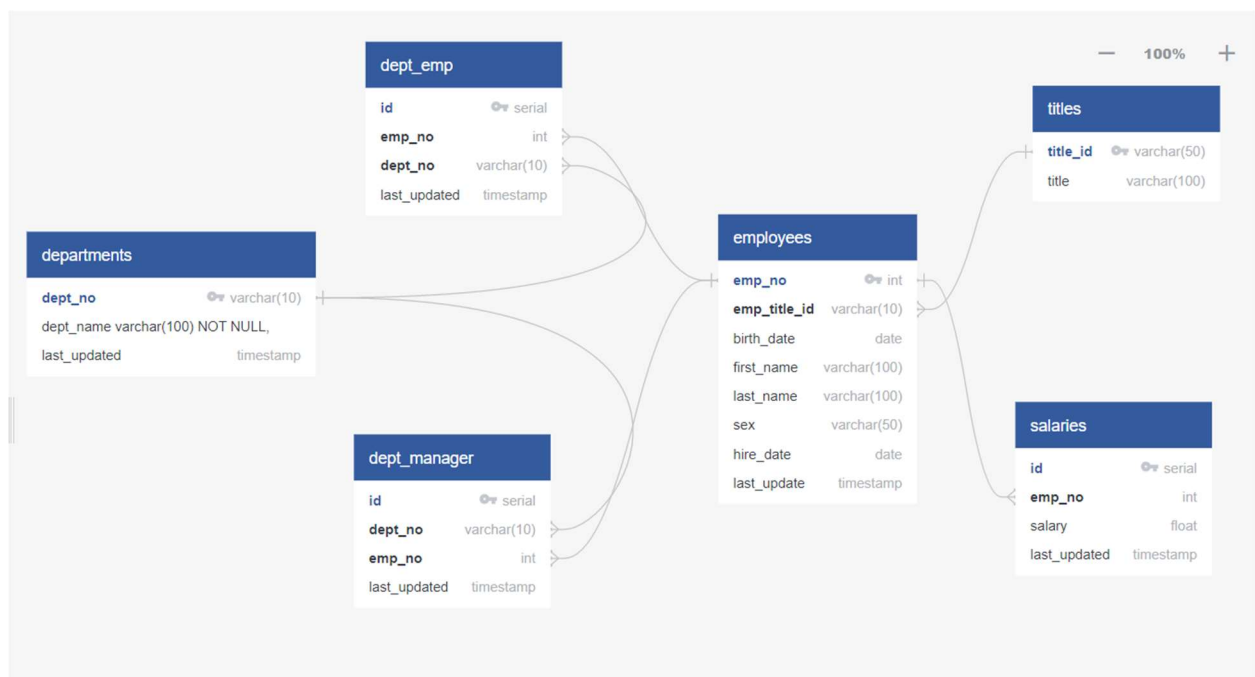
-

id serial PK

emp_no int FK >- employees.emp_no

salary float

last_updated timestamp



Step Export to pgAdmin

-- Exported from QuickDBD: <https://www.quickdatabasediagrams.com/>

-- NOTE! If you have used non-SQL datatypes in your design, you will have to change these here.

```
CREATE TABLE "departments" (  
    "dept_no" varchar(10) NOT NULL,  
    "dept_name" varchar(100) NOT NULL,  
    "last_updated" timestamp NOT NULL,  
    CONSTRAINT "pk_departments" PRIMARY KEY (  
        "dept_no"  
    )  
);
```

```
CREATE TABLE "dept_emp" (  
    "id" serial NOT NULL,  
    "emp_no" int NOT NULL,  
    "dept_no" varchar(10) NOT NULL,  
    "last_updated" timestamp NOT NULL,  
    CONSTRAINT "pk_dept_emp" PRIMARY KEY (  
        "id"  
    )  
);
```

```
CREATE TABLE "dept_manager" (  
    "id" serial NOT NULL,  
    "dept_no" varchar(10) NOT NULL,  
    "emp_no" int NOT NULL,
```

```
"last_updated" timestamp NOT NULL,  
CONSTRAINT "pk_dept_manager" PRIMARY KEY (  
    "id"  
)  
);
```

```
CREATE TABLE "employees" (  
    "emp_no" int NOT NULL,  
    "emp_title_id" varchar(10) NOT NULL,  
    "birth_date" date NOT NULL,  
    "first_name" varchar(100) NOT NULL,  
    "last_name" varchar(100) NOT NULL,  
    "sex" varchar(50) NOT NULL,  
    "hire_date" date NOT NULL,  
    "last_update" timestamp NOT NULL,  
    CONSTRAINT "pk_employees" PRIMARY KEY (  
        "emp_no"  
    )  
);
```

```
CREATE TABLE "titles" (  
    "title_id" varchar(50) NOT NULL,  
    "title" varchar(100) NOT NULL,  
    CONSTRAINT "pk_titles" PRIMARY KEY (  
        "title_id"  
    )  
);
```

```
CREATE TABLE "salaries" (  
    "emp_no" int NOT NULL,  
    "salary" int NOT NULL,  
    "last_update" timestamp NOT NULL,  
    CONSTRAINT "pk_salaries" PRIMARY KEY (  
        "emp_no"  
    )  
);
```

```
"id" serial NOT NULL,  
"emp_no" int NOT NULL,  
"salary" float NOT NULL,  
"last_updated" timestamp NOT NULL,  
CONSTRAINT "pk_salaries" PRIMARY KEY (  
    "id"  
)  
);
```

```
ALTER TABLE "dept_emp" ADD CONSTRAINT "fk_dept_emp_emp_no" FOREIGN KEY("emp_no")  
REFERENCES "employees" ("emp_no");
```

```
ALTER TABLE "dept_emp" ADD CONSTRAINT "fk_dept_emp_dept_no" FOREIGN KEY("dept_no")  
REFERENCES "departments" ("dept_no");
```

```
ALTER TABLE "dept_manager" ADD CONSTRAINT "fk_dept_manager_dept_no" FOREIGN KEY("dept_no")  
REFERENCES "departments" ("dept_no");
```

```
ALTER TABLE "dept_manager" ADD CONSTRAINT "fk_dept_manager_emp_no" FOREIGN KEY("emp_no")  
REFERENCES "employees" ("emp_no");
```

```
ALTER TABLE "employees" ADD CONSTRAINT "fk_employees_emp_title_id" FOREIGN  
KEY("emp_title_id")  
REFERENCES "titles" ("title_id");
```

```
ALTER TABLE "salaries" ADD CONSTRAINT "fk_salaries_emp_no" FOREIGN KEY("emp_no")  
REFERENCES "employees" ("emp_no");
```

Step 4 Codes for Questions

--List the following details of each employee: employee number, last name, first name, sex, and salary.

SELECT

e.emp_no,
e.first_name,
e.last_name,
e.sex,
s.salary

FROM

employees e

JOIN

salaries s

ON e.emp_no = s.emp_no;

4 e.first_name,

5 e.last_name,

Data Output Messages Notifications

	emp_no integer	first_name character varying (100)	last_name character varying (100)	sex character varying (50)	salary double precision	
1	10005	Kyoichi	Maliniak	M	78228	
2	10010	Duangkaew	Piveteau	F	72488	
3	10011	Mary	Sluis	F	42365	
4	10013	Eberhardt	Terkki	M	40000	
5	10017	Cristinel	Bouloucos	F	71380	
6	10035	Alain	Chappelet	M	41538	
7	10037	Pradeep	Makrucki	M	40000	
8	10045	Moss	Shanbhogue	M	41971	
9	10051	Hidefumi	Caine	M	48817	
10	10058	Berhard	McFarlin	M	52787	
11	10078	Danel	Mondadori	F	47280	
12	10085	Kenroku	Malabarba	M	40000	
13	10096	Jayson	Mandell	M	61395	
14	10117	Kiyotoshi	Blokdijk	F	52284	
15	10156	Sumali	Fargier	M	49879	
16	10167	Duangkaew	Rassart	M	56990	
17	10177	Pragnesh	Iisaka	M	48041	
18	10179	Deniz	Duclos	F	40000	
19	10219	Genta	Kolvik	M	40000	
20	10223	Carrsten	Schmiedel	F	62922	
21	10228	Karoline	Cesareni	F	79474	
22	10244	Foong	Flasterstein	M	56292	
23	10248	Frederique	Tempesti	F	58918	
24	10250	Serap	Etalle	M	50330	

Total rows: 1000 of 300024 Query complete 00:00:00.335

--List first name, last name, and hire date for employees who were hired in 1986.

SELECT

e.first_name,

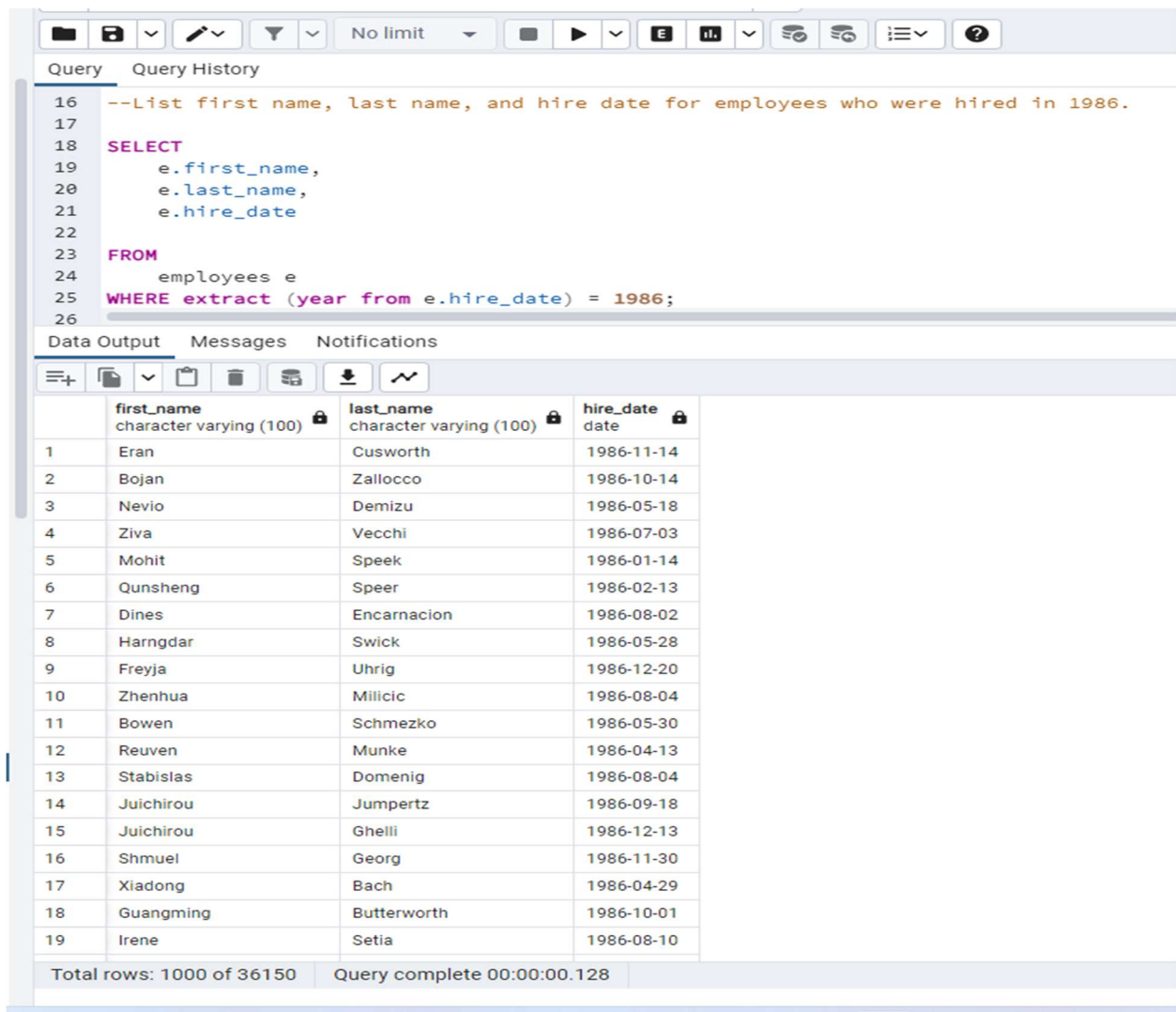
e.last_name,

e.hire_date

FROM

employees e

WHERE extract (year from e.hire_date) = 1986;



The screenshot shows a SQL IDE interface. The top toolbar contains icons for file operations, query execution, and settings. Below the toolbar, the 'Query' tab is active, displaying the following SQL code:

```
16 --List first name, last name, and hire date for employees who were hired in 1986.
17
18 SELECT
19     e.first_name,
20     e.last_name,
21     e.hire_date
22
23 FROM
24     employees e
25 WHERE extract (year from e.hire_date) = 1986;
26
```

Below the query editor, the 'Data Output' tab is active, showing a table with 3 columns: first_name, last_name, and hire_date. The table contains 19 rows of data, representing employees hired in 1986. The status bar at the bottom indicates 'Total rows: 1000 of 36150' and 'Query complete 00:00:00.128'.

	first_name character varying (100)	last_name character varying (100)	hire_date date
1	Eran	Cusworth	1986-11-14
2	Bojan	Zalocco	1986-10-14
3	Nevio	Demizu	1986-05-18
4	Ziva	Vecchi	1986-07-03
5	Mohit	Speek	1986-01-14
6	Qunsheng	Speer	1986-02-13
7	Dines	Encarnacion	1986-08-02
8	Harnghdar	Swick	1986-05-28
9	Freyja	Uhrig	1986-12-20
10	Zhenhua	Milicic	1986-08-04
11	Bowen	Schmezko	1986-05-30
12	Reuven	Munke	1986-04-13
13	Stabslas	Domenig	1986-08-04
14	Juichirou	Jumpertz	1986-09-18
15	Juichirou	Ghelli	1986-12-13
16	Shmuel	Georg	1986-11-30
17	Xiadong	Bach	1986-04-29
18	Guangming	Butterworth	1986-10-01
19	Irene	Setia	1986-08-10

Total rows: 1000 of 36150 Query complete 00:00:00.128

Query

Query History

```

28
29 --List the manager of each department with the following information: department number, department name, the manager's
30 SELECT
31     d.dept_no,
32     d.dept_name,
33     dm.emp_no,
34     e.first_name,
35     e.last_name
36
37
38 FROM dept_manager dm
39 INNER JOIN departments d on dm.dept_no = d.dept_no
40 INNER JOIN employees e on dm.emp_no=e.emp_no
41
42

```

Data Output

Messages

Notifications

+

📄

▼

🗑️

🔄

📥

📈

	dept_no character varying (10)	dept_name character varying (100)	emp_no integer	first_name character varying (100)	last_name character varying (100)
1	d001	Marketing	110022	Margareta	Markovitch
2	d001	Marketing	110039	Vishwani	Minakawa
3	d002	Finance	110085	Ebru	Alpin
4	d002	Finance	110114	Isamu	Legleitner
5	d003	Human Resources	110183	Shirish	Ossenbruggen
6	d003	Human Resources	110228	Karsten	Sigstam
7	d004	Production	110303	Krassimir	Wegerle
8	d004	Production	110344	Rosine	Cools
9	d004	Production	110386	Shem	Kieras
10	d004	Production	110420	Oscar	Ghazalie
11	d005	Development	110511	DeForest	Hagimont
12	d005	Development	110567	Leon	DasSarma
13	d006	Quality Management	110725	Peternela	Onuegbe
14	d006	Quality Management	110765	Rutger	Hofmeyr
15	d006	Quality Management	110800	Sanjoy	Quadeer
16	d006	Quality Management	110854	Dung	Pesch
17	d007	Sales	111035	Przemyslaw	Kaelbling

Total rows: 24 of 24

Query complete 00:00:00.043

--List the department of each employee with the following information: employee number, last name, first name, and department name.

SELECT

e.emp_no,
e.first_name,
e.last_name,
d.dept_name

FROM employees e

INNER JOIN dept_emp de on e.emp_no = de.emp_no

INNER JOIN departments d on d.dept_no=de.dept_no;

```
3 --List the department of each employee with the following information: employee number, last name, first name, and dep
4
5 SELECT
6     e.emp_no,
7     e.first_name,
8     e.last_name,
9     d.dept_name
10
11
12 FROM employees e
13 INNER JOIN dept_emp de on e.emp_no = de.emp_no
14 INNER JOIN departments d on d.dept_no=de.dept_no
15 --List first name, last name, and sex for employees whose first name is "Hercules" and last names begin with "B."
16
```

ata Output Messages Notifications

	emp_no integer	first_name character varying (100)	last_name character varying (100)	dept_name character varying (100)
	10005	Kyoichi	Maliniak	Human Resources
	10010	Duangkaew	Piveteau	Production
	10010	Duangkaew	Piveteau	Quality Management
	10011	Mary	Sluis	Customer Service
	10013	Eberhardt	Terkki	Human Resources
	10017	Cristinel	Bouloucos	Marketing
	10035	Alain	Chappelet	Production
	10037	Pradeep	Makrucki	Development
	10045	Moss	Shanbhogue	Production
0	10051	Hidefumi	Caine	Production
1	10058	Berhard	McFarlin	Marketing
2	10078	Danel	Mondadori	Development
3	10085	Kenroku	Malabarba	Production
4	10096	Jayson	Mandell	Production
5	10117	Kiyotoshi	Blokdijs	Production
6	10156	Sumali	Fardier	Quality Management

total rows: 1000 of 331603 Query complete 00:00:00.387

--List first name, last name, and sex for employees whose first name is "Hercules" and last names begin with "B."

SELECT

e.first_name,

e.last_name,

e.sex

FROM employees e

WHERE e.first_name = 'Hercules'

and e.last_name like 'B%';

The screenshot shows a SQL IDE interface. The top pane displays a query with line numbers 53 to 65. The query is as follows:

```
53 INNER JOIN departments d on e.emp_no = de.emp_no
54 INNER JOIN departments d on d.dept_no=de.dept_no
55 --List first name, last name, and sex for employees whose first name is "Hercules" and last names begin with "B."
56
57 SELECT
58     e.first_name,
59     e.last_name,
60     e.sex
61 FROM employees e
62 WHERE e.first_name = 'Hercules'
63     and e.last_name like 'B%'
64
65
```

The bottom pane shows the 'Data Output' tab with a table of results. The table has three columns: first_name, last_name, and sex. There are 18 rows of data.

	first_name character varying (100)	last_name character varying (100)	sex character varying (50)
1	Hercules	Baer	M
2	Hercules	Biron	F
3	Hercules	Birge	F
4	Hercules	Berstel	F
5	Hercules	Bernatsky	M
6	Hercules	Bail	F
7	Hercules	Bodoff	M
8	Hercules	Benantar	F
9	Hercules	Basagni	M
10	Hercules	Bernardinello	F
11	Hercules	Baranowski	M
12	Hercules	Bisiani	F
13	Hercules	Benzmuller	M
14	Hercules	Bahr	M
15	Hercules	Biran	F
16	Hercules	Bain	F
17	Hercules	Brendel	F
18	Hercules	Buchter	M

At the bottom of the results pane, it says 'Total rows: 20 of 20' and 'Query complete 00:00:00.120'. The Windows taskbar is visible at the very bottom of the image.

Query

Query History

```

68 SELECT
69     e.emp_no,
70     e.first_name,
71     e.last_name,
72     d.dept_name
73
74
75 FROM employees e
76 INNER JOIN dept_emp de on e.emp_no = de.emp_no
77 INNER JOIN departments d on d.dept_no=de.dept_no
78 where d.dept_name = 'Sales'
79
80

```

Data Output

Messages

Notifications

≡

📄

▼

📋

🗑️

🔄

⬇️

📈

	emp_no integer	first_name character varying (100)	last_name character varying (100)	dept_name character varying (100)
1	10002	Bezalel	Simmel	Sales
2	10016	Kazuhiro	Cappelletti	Sales
3	10034	Bader	Swan	Sales
4	10041	Uri	Lenart	Sales
5	10050	Yinghua	Dredge	Sales
6	10053	Sanjiv	Zschoche	Sales
7	10060	Breanna	Billingsley	Sales
8	10061	Tse	Herber	Sales
9	10068	Charlene	Brattka	Sales
10	10087	Xinglin	Eugenio	Sales
11	10088	Jungsoo	Syrzycki	Sales
12	10089	Sudharsan	Flasterstein	Sales
13	10093	Sailaja	Desikan	Sales
14	10095	Hilari	Morton	Sales
15	10099	Valter	Sullins	Sales
16	10101	Perla	Heyers	Sales
17	10107	Dung	Baca	Sales
18	10125	Syozo	Hiltgen	Sales

--List all employees in the Sales and Development departments, including their employee number, last name, first name, and department name.

SELECT

e.emp_no,
e.first_name,
e.last_name,
d.dept_name

FROM employees e

INNER JOIN dept_emp de on e.emp_no = de.emp_no

INNER JOIN departments d on d.dept_no=de.dept_no

WHERE d.dept_name = 'Sales'

or d.dept_name = 'Development';

Query		Query history	
81	--List all employees in the Sales and Development departments, including their employee number, last name, first name		
82	SELECT		
83	e.emp_no,		
84	e.first_name,		
85	e.last_name,		
86	d.dept_name		
87			
88			
89	FROM employees e		
90	INNER JOIN dept_emp de on e.emp_no = de.emp_no		
91	INNER JOIN departments d on d.dept_no=de.dept_no		
92	where d.dept_name = 'Sales'		
93	or d.dept_name = 'Development'		
94			

Data Output		Messages		Notifications	
	emp_no	first_name	last_name	dept_name	
	integer	character varying (100)	character varying (100)	character varying (100)	
1	10001	Georgi	Facello	Development	
2	10002	Bezalel	Simmel	Sales	
3	10006	Anneke	Preusig	Development	
4	10008	Saniya	Kalloufi	Development	
5	10012	Patricio	Bridgland	Development	
6	10014	Berni	Genin	Development	
7	10016	Kazuhito	Cappelletti	Sales	
8	10018	Kazuhide	Peha	Development	
9	10021	Ramzi	Erde	Development	
10	10022	Shahaf	Famili	Development	
11	10023	Bojan	Montemayor	Development	
12	10025	Prasadram	Heyers	Development	
13	10027	Divier	Reistad	Development	
14	10028	Domenick	Tempesti	Development	
15	10031	Karsten	Joslin	Development	
16	10034	Bader	Swan	Sales	
17	10037	Pradeep	Makrucki	Development	

--List the frequency count of employee last names (i.e., how many employees share each last name) in descending order.

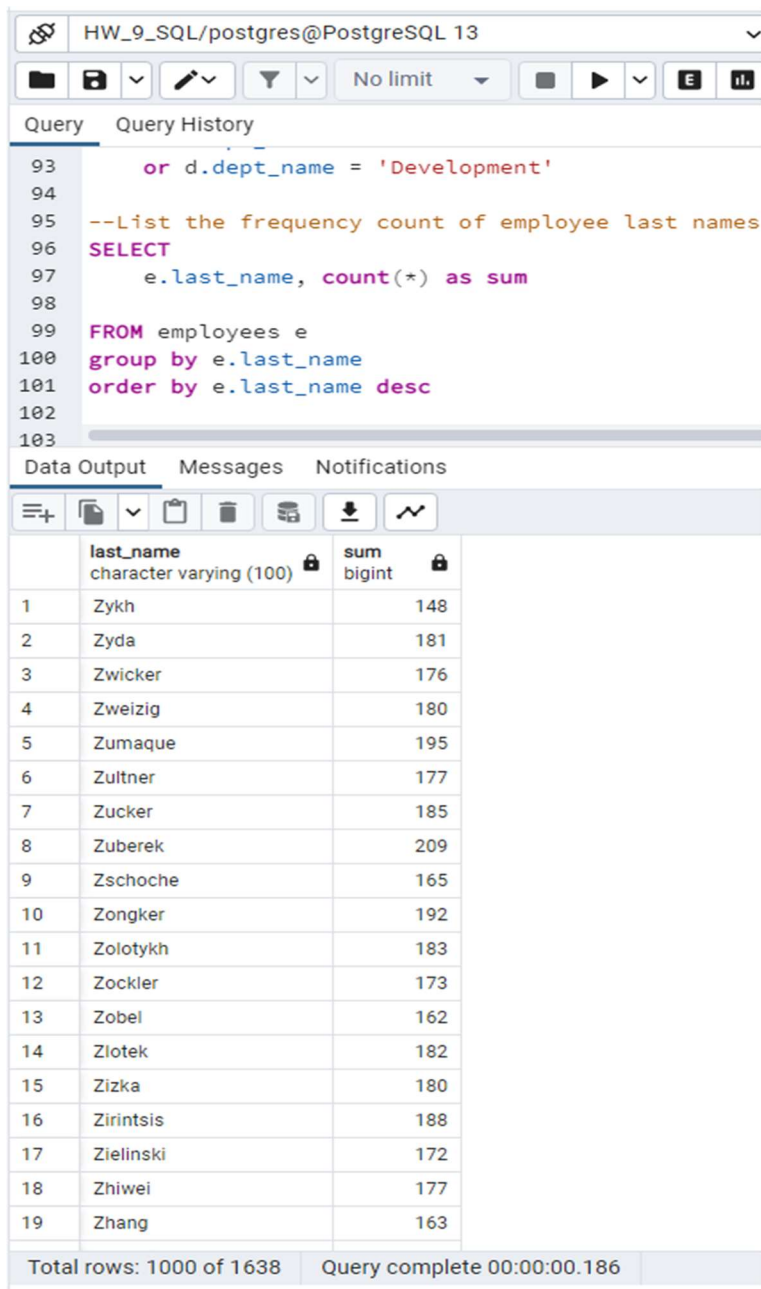
SELECT

 e.last_name, count(*) as sum

FROM employees e

GROUP BY e.last_name

ORDER BY e.last_name desc;



The screenshot shows a PostgreSQL query editor window titled "HW_9_SQL/postgres@PostgreSQL 13". The query is as follows:

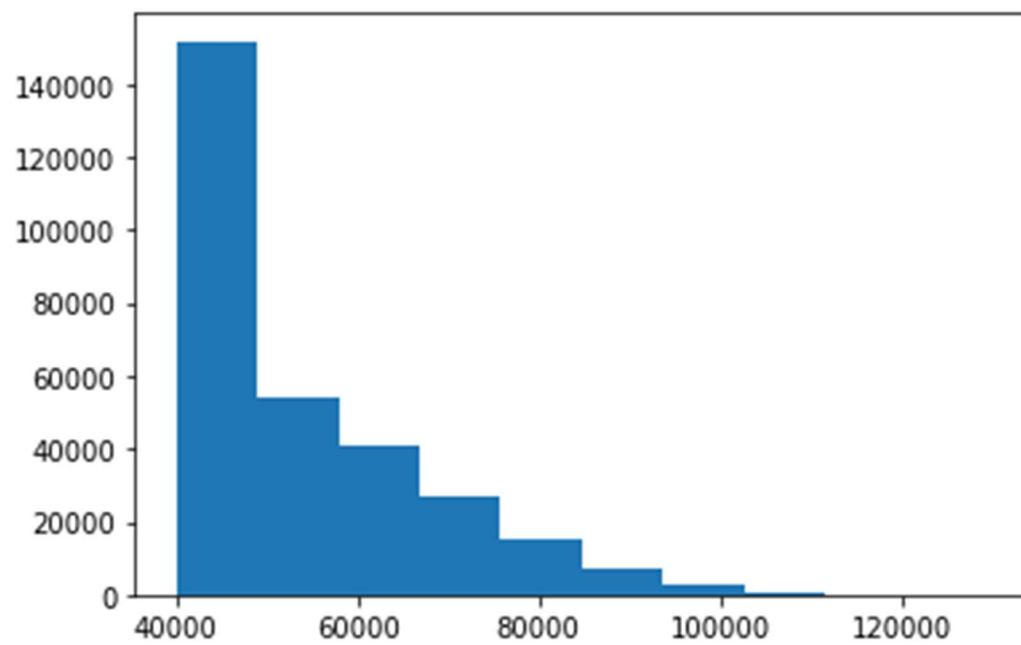
```
93     or d.dept_name = 'Development'
94
95 --List the frequency count of employee last names
96 SELECT
97     e.last_name, count(*) as sum
98
99 FROM employees e
100 group by e.last_name
101 order by e.last_name desc
102
103
```

The results are displayed in a table with the following columns: last_name (character varying (100)) and sum (bigint). The table contains 19 rows of data, sorted in descending order by the frequency count (sum).

	last_name character varying (100)	sum bigint
1	Zykh	148
2	Zyda	181
3	Zwicker	176
4	Zweizig	180
5	Zumaque	195
6	Zultner	177
7	Zucker	185
8	Zuberek	209
9	Zschoche	165
10	Zongker	192
11	Zolotykh	183
12	Zockler	173
13	Zobel	162
14	Zlotek	182
15	Zizka	180
16	Zirintsis	188
17	Zielinski	172
18	Zhiwei	177
19	Zhang	163

Total rows: 1000 of 1638 Query complete 00:00:00.186

Bonus Chart 1



Bonus Chart 2

