BAN5501 Final Project: Project Management Database

By Nathan Minkwitz, Noah Skole, & Steven Caione

Project Overview

Introduction

- □ **Industry:** Consulting
- □ **Business Type:** Multiple-Branch Business Consulting Company
- □ **Company Information:**

  We are the new executives for DevKings Consulting Group, a US-based consulting company that has just scaled operations to service all four time zones in the US. DevKings was founded in 2019 in Boston, MA.

  - □ The company just expanded operations, opening three more offices in Chicago, IL, Denver, CO, and Los Angeles, CA,

DK

DEVKINGS
CONSULTING GROUP

Business Situation

❑ Our existing database works for managing projects for our original office in Boston, MA.

❑ It will not be able to maintain data integrity and security across an additional three offices in Chicago, IL, Denver, CO, and Los Angeles, CA.

❑ We need our Dev Team to build a new project management database that enables the executives, managers, and consultants at DevKings to operate successfully at scale.

WASHINGTON
MONTANA
NORTH DAKOTA
MIN NES OTA
OREGON
IDAHO
SOUTH DAKOTA
WISCO NSIN
MI CHI GAN
NEW YORK
WYOMING
PENNSYLVANIA
CALIFORNIA
NEVADA
NEBRASKA
IOWA
IL NOIS
INDI ANA
OHIO
UTAH
COLORADO
KANSAS
MISS OURI
KENTUCKY
VIRGINIA
NORTH CAROLINA
TENNESSEE
ARIZONA
NEW MEXICO
OKLAHOMA
ARKA NSAS
SOUTH CAROLINA
MIS SIS SIPPI
ALA BAMA
GEO RGIA
TEXAS
LOU ISI ANA
FLORIDA
ALASKA
DK
DEVKINGS CONSULTING GROUP

Database Considerations

- ❑ Preliminary
    - ❑ What industry are we looking at?
    - ❑ What does our company do?
    - ❑ How many employees do we have
    - ❑ Where are they located?
    - ❑ What is the current situation?
    - ❑ What are the business requirements?
- ❑ Secondary
    - ❑ How many tables do we need?
    - ❑ What type of information will they hold?
    - ❑ Which are our fact tables? Lookup?
    - ❑ Are there any relationships?
        - ❑ PK or FK?

Database Design

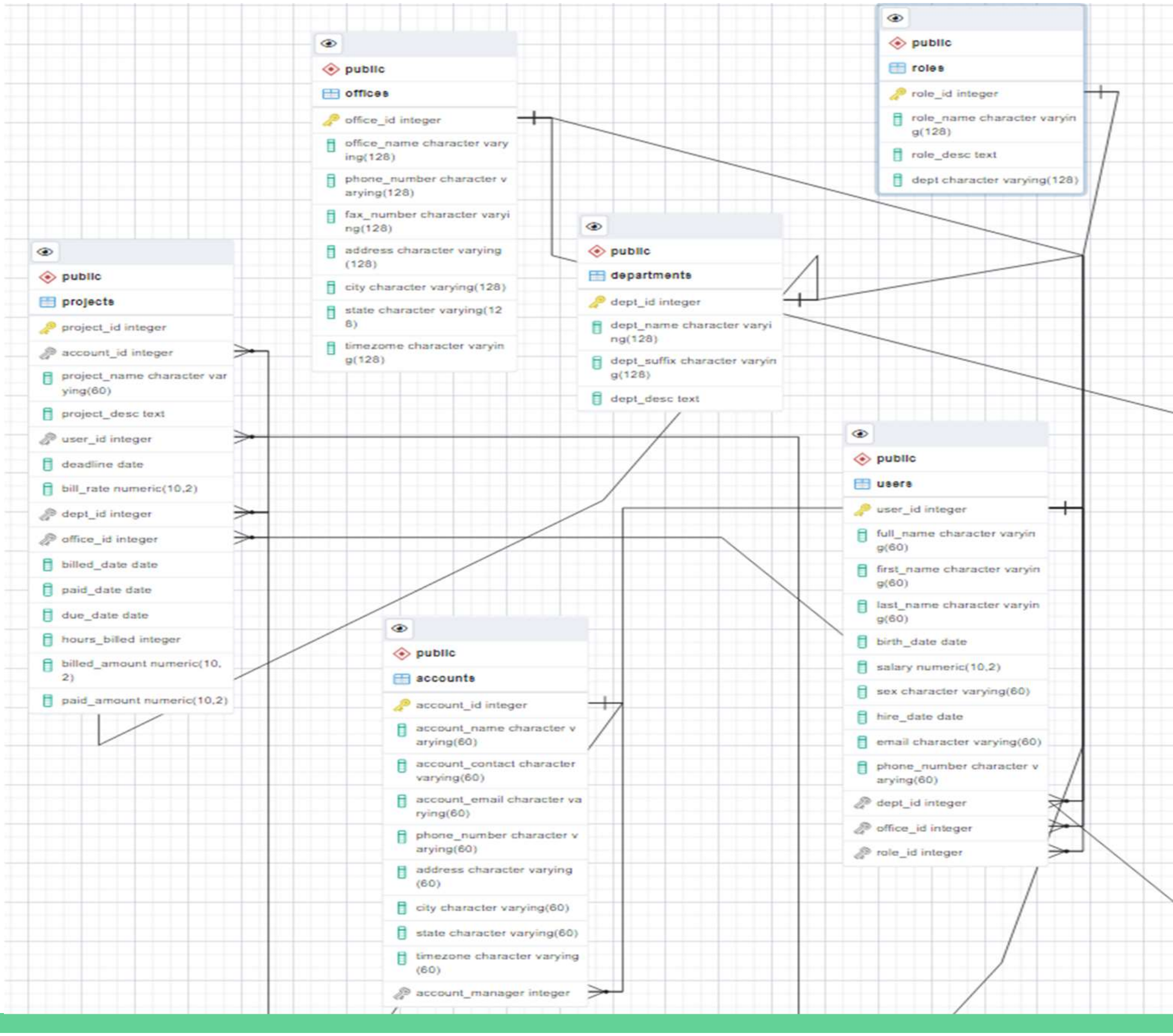
Database Schema: Fact Tables

- ❑ **Roles**
  - ❑ **Primary Key:** role_id
  - ❑ **Fields:** role_name, role_description, dept
- ❑ **Departments**
  - ❑ **Primary Key:** dept_id
  - ❑ **Fields:** dept_name, dept_suffix, dept_description
- ❑ **Offices**
  - ❑ **Primary Key:** office_id
  - ❑ **Fields:** office_name, phone_number, fax_number, address, city, state, timezone
- ❑ **Accounts**
  - ❑ **Primary Key:** account_id
  - ❑ **Fields:** account_name, account_manager (user_id), account_contact, account_email, account_phone_number, city, state, timezone

Database Schema: Lookup Tables

- ❑ **Users**
  - ❑ **Primary Key:** user_id
  - ❑ **Foriegn Keys**: dept_id, office_id, role_id
  - ❑ **Fields:** full_name, first_name, last_name, birthdate, salary, sex, hire_date, address, email, phone number
- ❑ **Projects**
  - ❑ **Primary Key:** project_id
  - ❑ **Foreign Keys:** account_id, dept_id, office_id, project_consultant(user_id)
  - ❑ **Fields:** project_name, project_description, deadline, bill_rate, bill_date, paid_date, due_date, hours_billed, billed_amount, paid_amount

Database Schema: Entity Relationship Diagram (ERD)



public
offices
- office_id integer
- office_name character varying(128)
- phone_number character varying(128)
- fax_number character varying(128)
- address character varying (128)
- city character varying(128)
- state character varying(128)
- timezome character varying(128)

public
roles
- role_id integer
- role_name character varying(128)
- role_desc text
- dept character varying(128)

public
departments
- dept_id integer
- dept_name character varying(128)
- dept_suffix character varying(128)
- dept_desc text

public
projects
- project_id integer
- account_id integer
- project_name character varying(60)
- project_desc text
- user_id integer
- deadline date
- bill_rate numeric(10,2)
- dept_id integer
- office_id integer
- billed_date date
- paid_date date
- due_date date
- hours_billed integer
- billed_amount numeric(10, 2)
- paid_amount numeric(10,2)

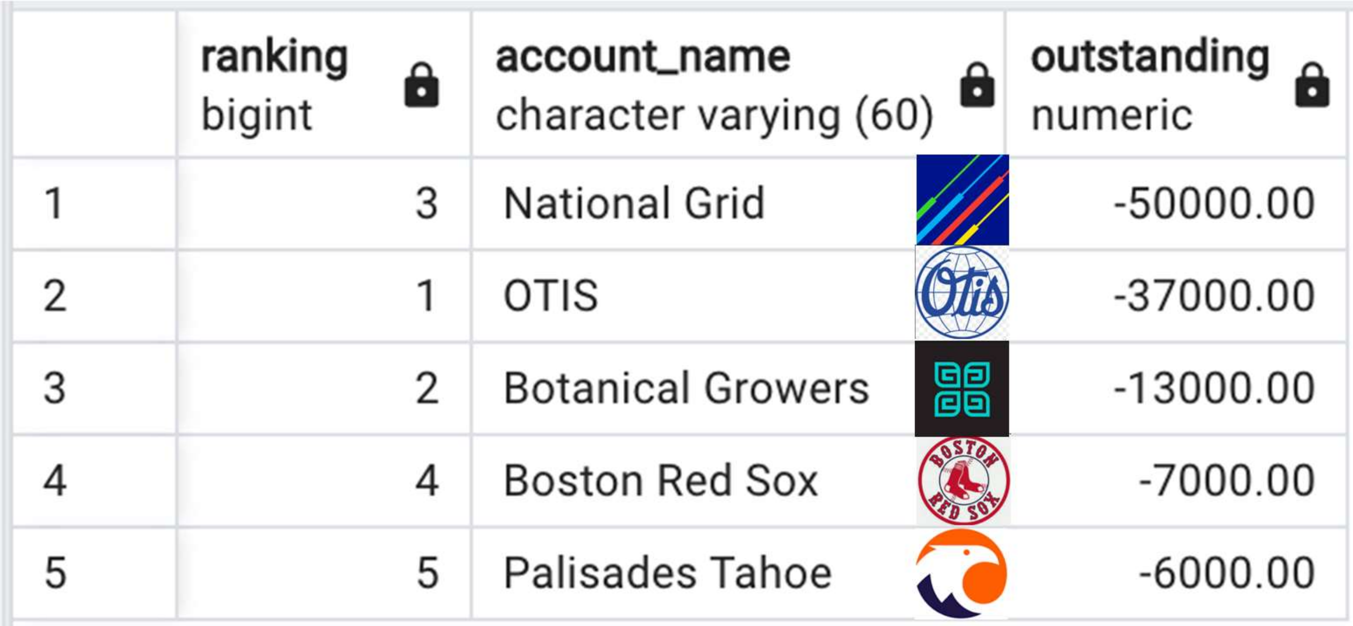
public
users
- user_id integer
- full_name character varying(60)
- first_name character varying(60)
- last_name character varying(60)
- birth_date date
- salary numeric(10,2)
- sex character varying(60)
- hire_date date
- email character varying(60)
- phone_number character varying(60)
- dept_id integer
- office_id integer
- role_id integer

public
accounts
- account_id integer
- account_name character varying(60)
- account_contact character varying(60)
- account_email character varying(60)
- phone_number character varying(60)
- address character varying (60)
- city character varying(60)
- state character varying(60)
- timezone character varying (60)
- account_manager integer

Application: SQL Queries

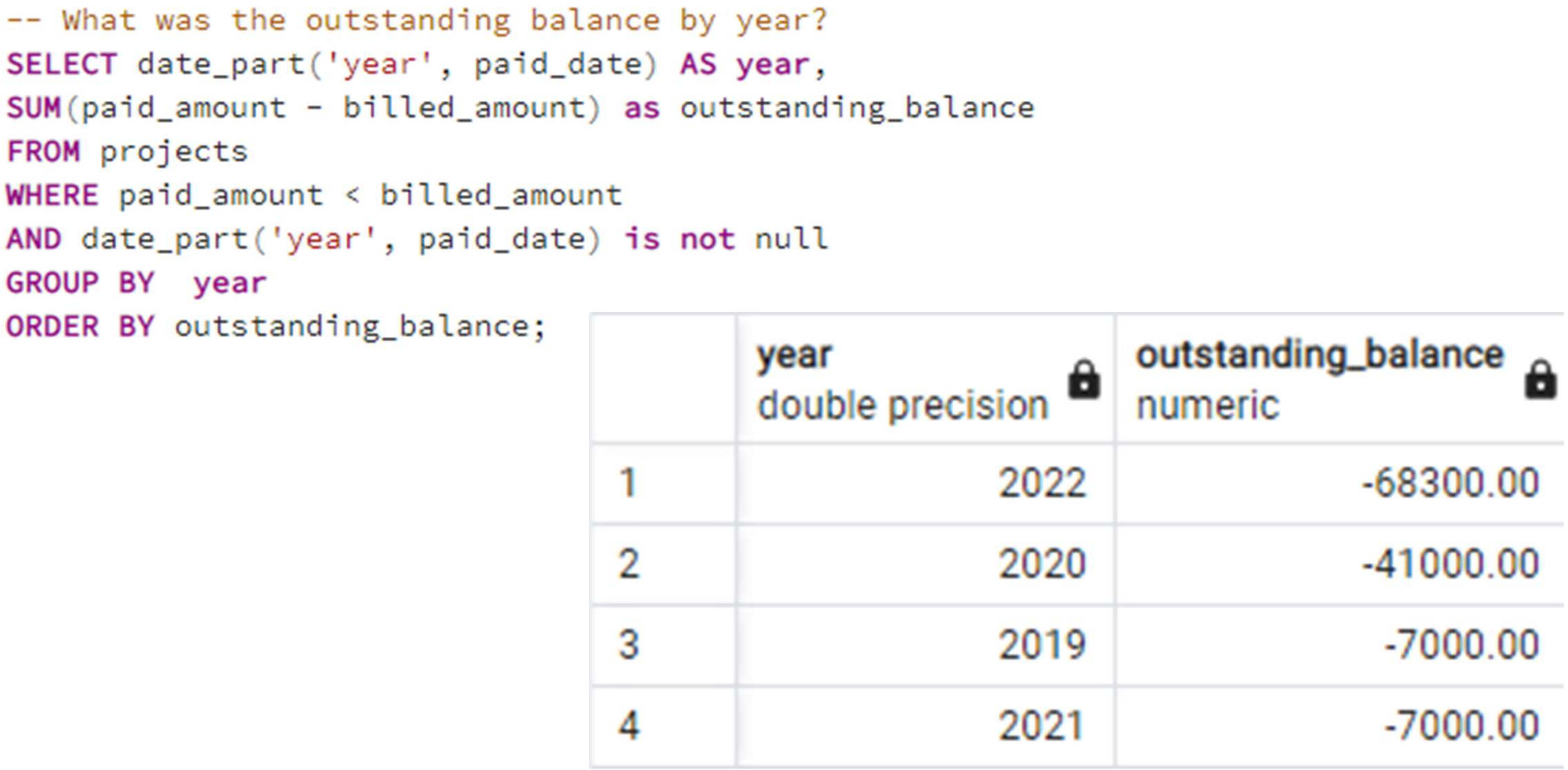
Query 1.0: Ranking Accounting Analysis

```sql
-- Outstanding Balance by Account?
SELECT RANK() OVER(ORDER BY SUM(p.paid_amount) DESC) AS ranking,
    a.account_name,
    SUM(p.paid_amount - p.billed_amount) AS outstanding
FROM projects p
JOIN accounts a
ON p.account_id = a.account_id
where paid_amount < billed_amount
GROUP BY account_name
LIMIT 5;
```



	ranking bigint	account_name character varying (60)	outstanding numeric
1	3	National Grid	-50000.00
2	1	OTIS	-37000.00
3	2	Botanical Growers	-13000.00
4	4	Boston Red Sox	-7000.00
5	5	Palisades Tahoe	-6000.00

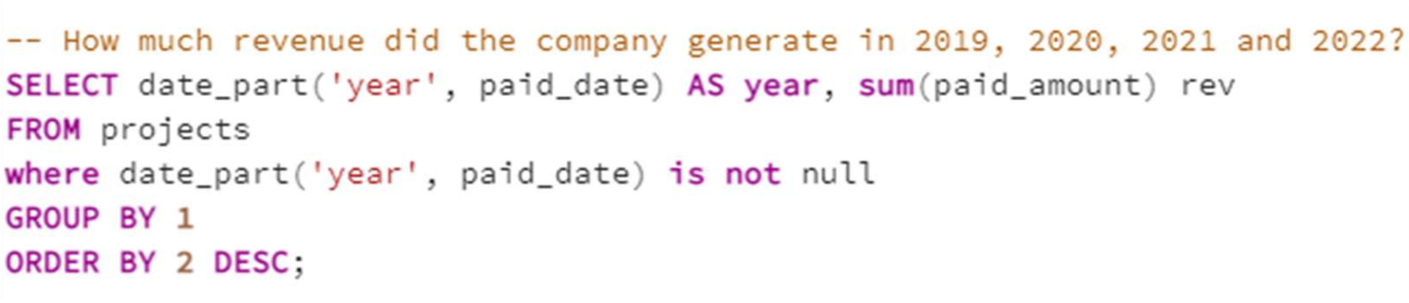
Query 1.1: YoY Accounting Analysis



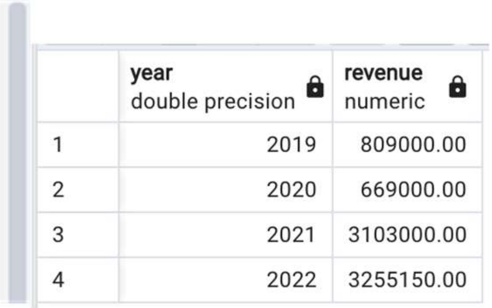
```
-- What was the outstanding balance by year?
SELECT date_part('year', paid_date) AS year,
SUM(paid_amount - billed_amount) as outstanding_balance
FROM projects
WHERE paid_amount < billed_amount
AND date_part('year', paid_date) is not null
GROUP BY  year
ORDER BY outstanding_balance;
```

	year double precision	outstanding_balance numeric
1	2022	-68300.00
2	2020	-41000.00
3	2019	-7000.00
4	2021	-7000.00

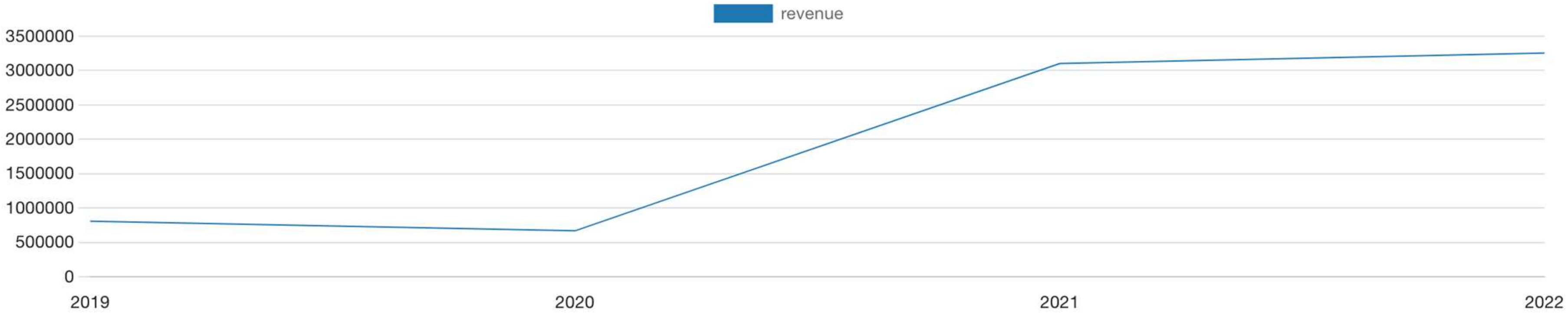
Query 2: Analyzing YoY Revenue Growth of DevKings Consulting Group



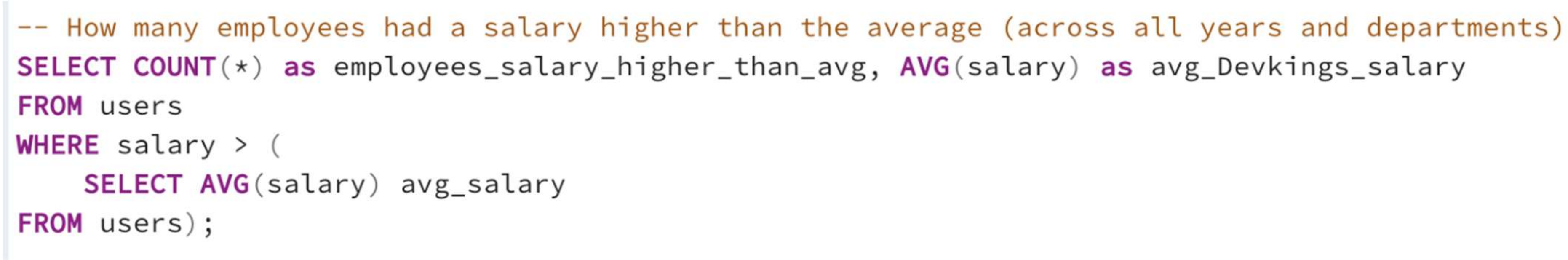
```
-- How much revenue did the company generate in 2019, 2020, 2021 and 2022?
SELECT date_part('year', paid_date) AS year, sum(paid_amount) rev
FROM projects
where date_part('year', paid_date) is not null
GROUP BY 1
ORDER BY 2 DESC;
```



	year double precision 🔒	revenue numeric 🔒
1	2019	809000.00
2	2020	669000.00
3	2021	3103000.00
4	2022	3255150.00

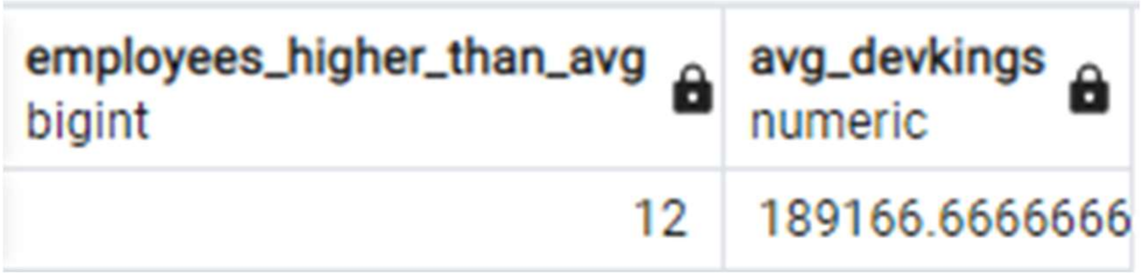


Query 3: Average Salary Analysis



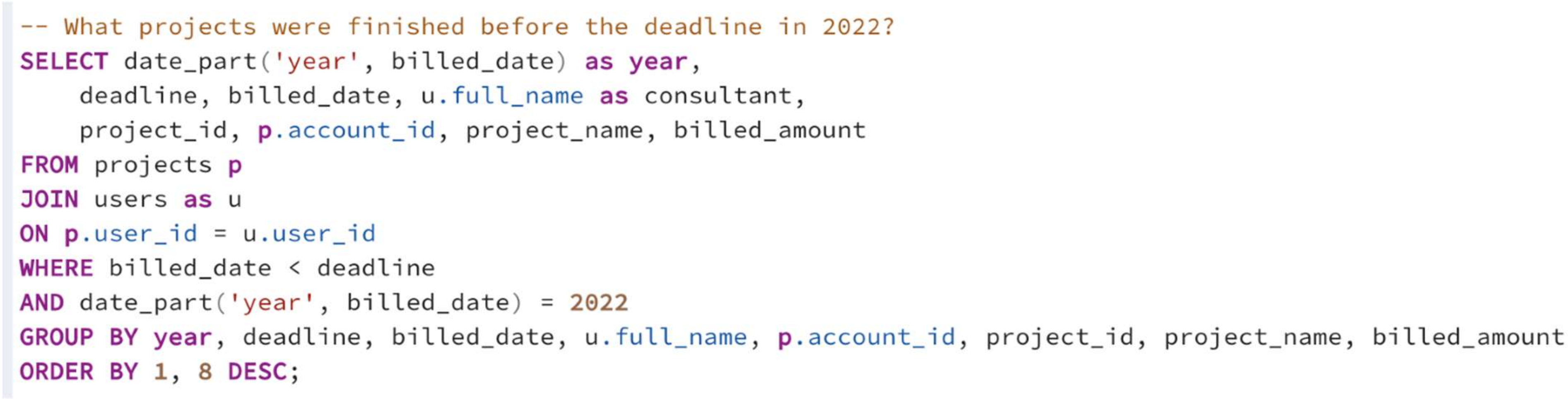
```sql
-- How many employees had a salary higher than the average (across all years and departments)
SELECT COUNT(*) as employees_salary_higher_than_avg, AVG(salary) as avg_Devkings_salary
FROM users
WHERE salary > (
    SELECT AVG(salary) avg_salary
FROM users);
```

- □ 12 employees had higher than the average salary of $189,167

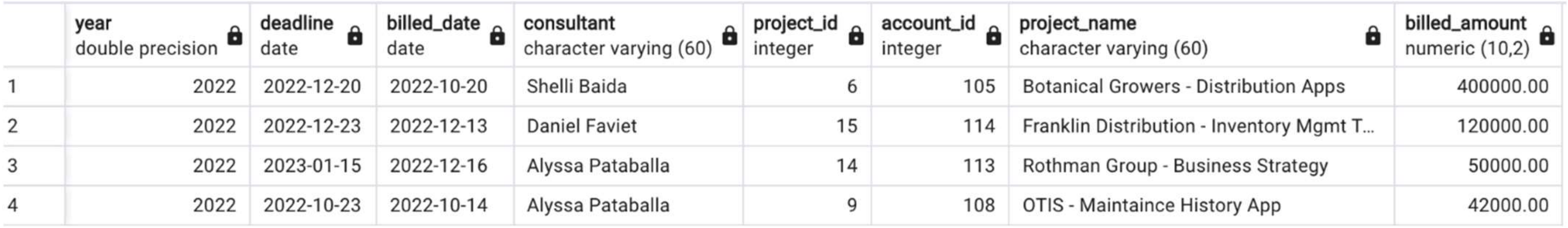


employees_higher_than_avg bigint	avg_devkings numeric
12	189166.6666666

Query 4: Consultant Productivity Analysis

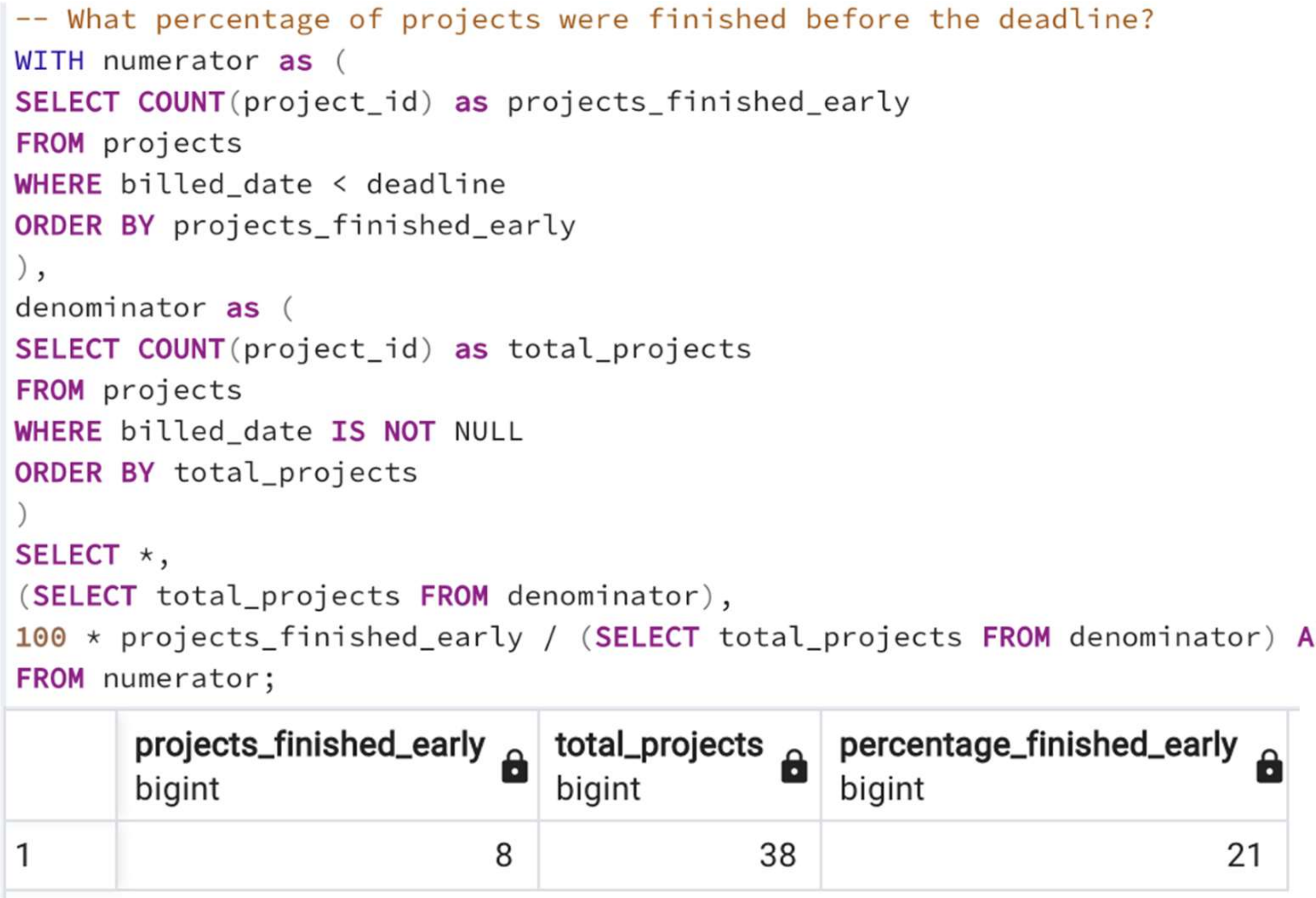


```
-- What projects were finished before the deadline in 2022?
SELECT date_part('year', billed_date) as year,
    deadline, billed_date, u.full_name as consultant,
    project_id, p.account_id, project_name, billed_amount
FROM projects p
JOIN users as u
ON p.user_id = u.user_id
WHERE billed_date < deadline
AND date_part('year', billed_date) = 2022
GROUP BY year, deadline, billed_date, u.full_name, p.account_id, project_id, project_name, billed_amount
ORDER BY 1, 8 DESC;
```



	year double precision	deadline date	billed_date date	consultant character varying (60)	project_id integer	account_id integer	project_name character varying (60)	billed_amount numeric (10,2)
1	2022	2022-12-20	2022-10-20	Shelli Baida	6	105	Botanical Growers - Distribution Apps	400000.00
2	2022	2022-12-23	2022-12-13	Daniel Faviet	15	114	Franklin Distribution - Inventory Mgmt T...	120000.00
3	2022	2023-01-15	2022-12-16	Alyssa Pataballa	14	113	Rothman Group - Business Strategy	50000.00
4	2022	2022-10-23	2022-10-14	Alyssa Pataballa	9	108	OTIS - Maintaince History App	42000.00

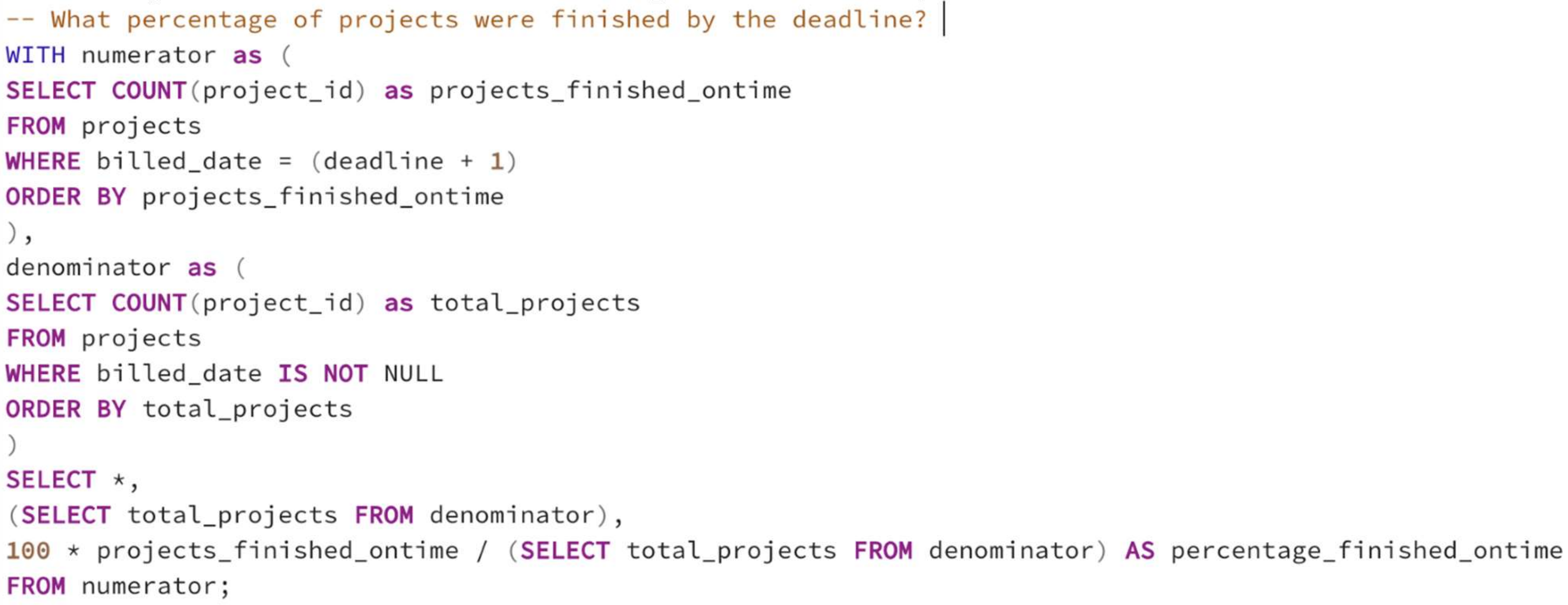
Query 5.0: Early Project Completion Rate



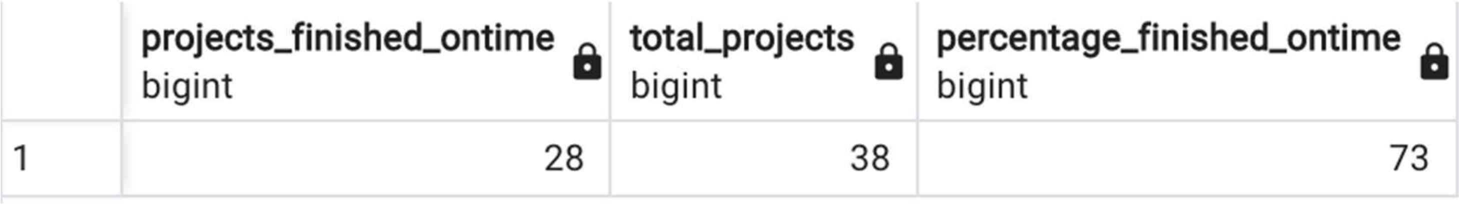
```
-- What percentage of projects were finished before the deadline?
WITH numerator as (
SELECT COUNT(project_id) as projects_finished_early
FROM projects
WHERE billed_date < deadline
ORDER BY projects_finished_early
),
denominator as (
SELECT COUNT(project_id) as total_projects
FROM projects
WHERE billed_date IS NOT NULL
ORDER BY total_projects
)
SELECT *,
(SELECT total_projects FROM denominator),
100 * projects_finished_early / (SELECT total_projects FROM denominator) AS percentage_finished_early
FROM numerator;
```

	projects_finished_early bigint	total_projects bigint	percentage_finished_early bigint
1	8	38	21

Query 5.1: On-time Project Completion Rate

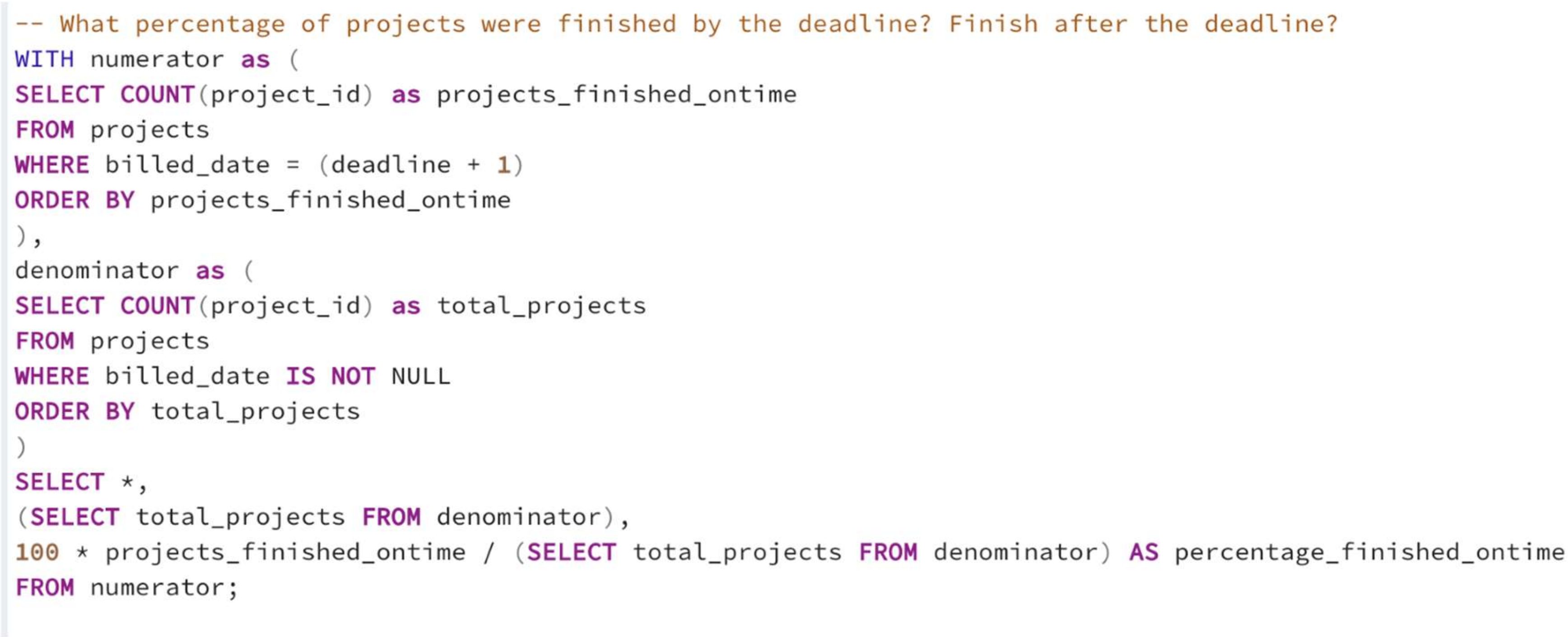


```sql
-- What percentage of projects were finished by the deadline?
WITH numerator as (
SELECT COUNT(project_id) as projects_finished_ontime
FROM projects
WHERE billed_date = (deadline + 1)
ORDER BY projects_finished_ontime
),
denominator as (
SELECT COUNT(project_id) as total_projects
FROM projects
WHERE billed_date IS NOT NULL
ORDER BY total_projects
)
SELECT *,
(SELECT total_projects FROM denominator),
100 * projects_finished_ontime / (SELECT total_projects FROM denominator) AS percentage_finished_ontime
FROM numerator;
```

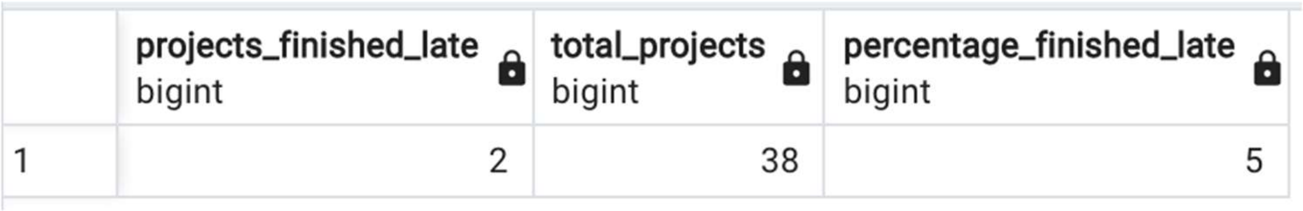


	projects_finished_ontime bigint	total_projects bigint	percentage_finished_ontime bigint
1	28	38	73

Query 5.2: Late Project Completion Rate

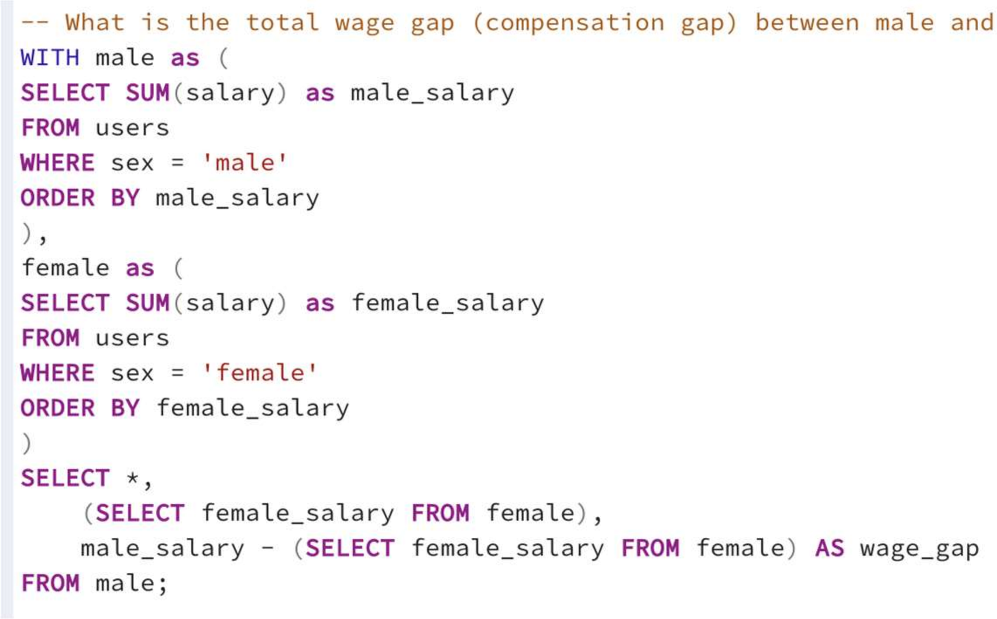


```
-- What percentage of projects were finished by the deadline? Finish after the deadline?
WITH numerator as (
SELECT COUNT(project_id) as projects_finished_ontime
FROM projects
WHERE billed_date = (deadline + 1)
ORDER BY projects_finished_ontime
),
denominator as (
SELECT COUNT(project_id) as total_projects
FROM projects
WHERE billed_date IS NOT NULL
ORDER BY total_projects
)
SELECT *,
(SELECT total_projects FROM denominator),
100 * projects_finished_ontime / (SELECT total_projects FROM denominator) AS percentage_finished_ontime
FROM numerator;
```

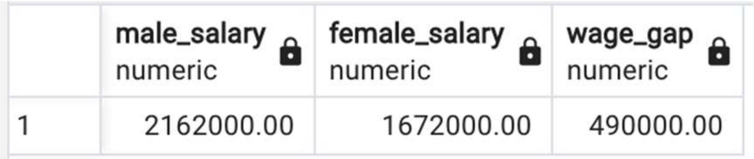


	projects_finished_late bigint	total_projects bigint	percentage_finished_late bigint
1	2	38	5

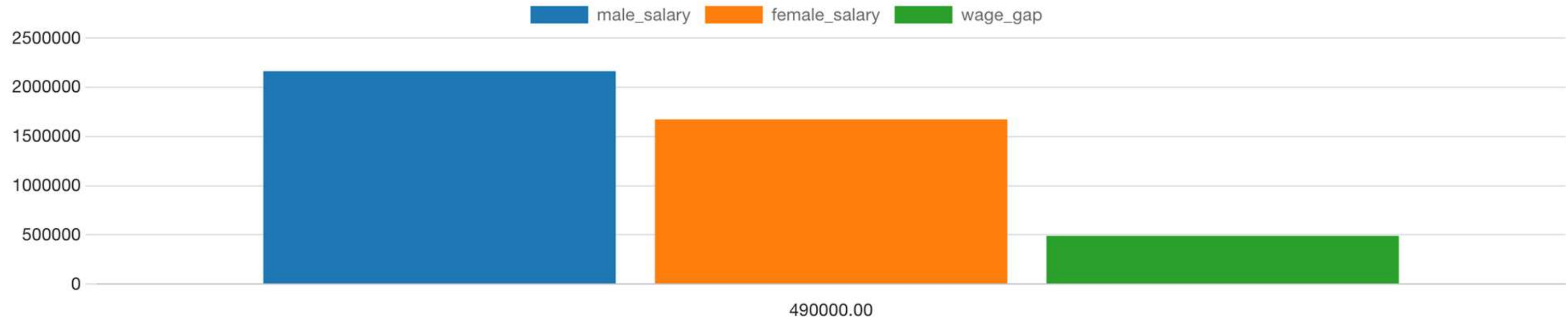
Query 6.0: Investigating Compensation Gaps



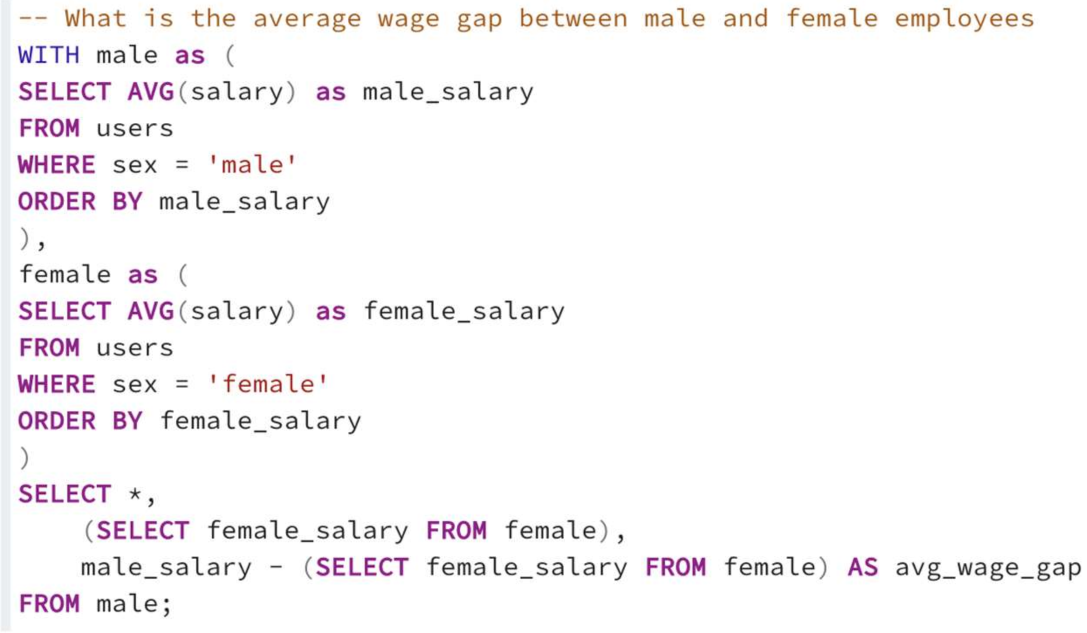
```
-- What is the total wage gap (compensation gap) between male and female employees?
WITH male as (
SELECT SUM(salary) as male_salary
FROM users
WHERE sex = 'male'
ORDER BY male_salary
),
female as (
SELECT SUM(salary) as female_salary
FROM users
WHERE sex = 'female'
ORDER BY female_salary
)
SELECT *,
    (SELECT female_salary FROM female),
    male_salary - (SELECT female_salary FROM female) AS wage_gap
FROM male;
```



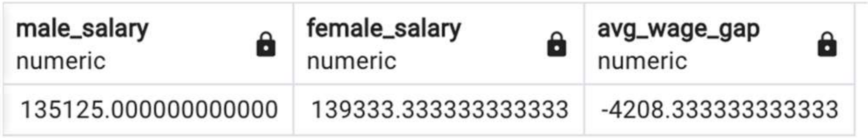
	male_salary numeric	female_salary numeric	wage_gap numeric
1	2162000.00	1672000.00	490000.00



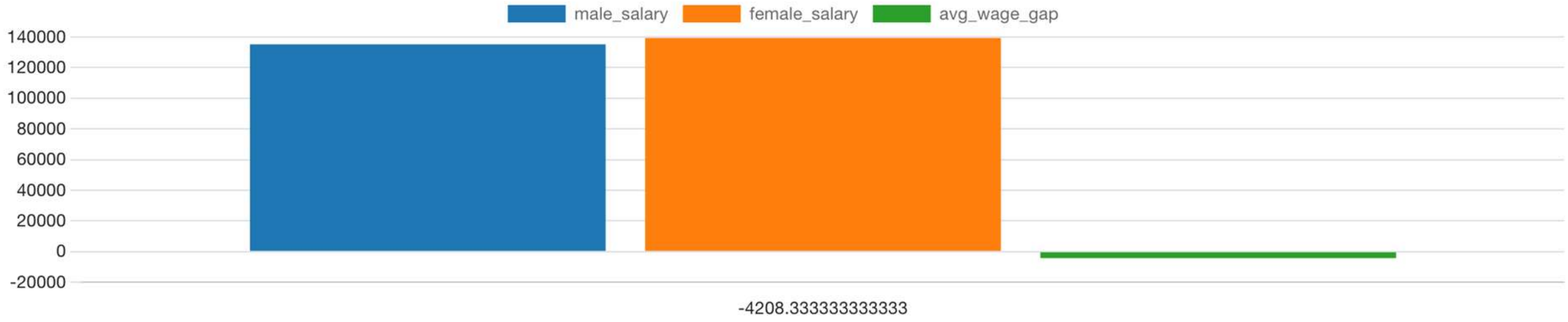
Query 6.1: Average Salary Gap



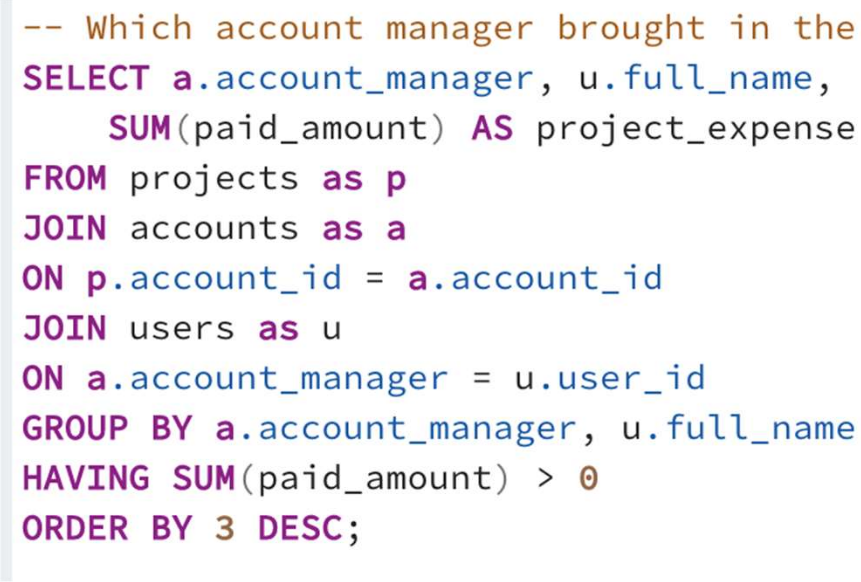
```
-- What is the average wage gap between male and female employees
WITH male as (
SELECT AVG(salary) as male_salary
FROM users
WHERE sex = 'male'
ORDER BY male_salary
),
female as (
SELECT AVG(salary) as female_salary
FROM users
WHERE sex = 'female'
ORDER BY female_salary
)
SELECT *,
    (SELECT female_salary FROM female),
    male_salary - (SELECT female_salary FROM female) AS avg_wage_gap
FROM male;
```



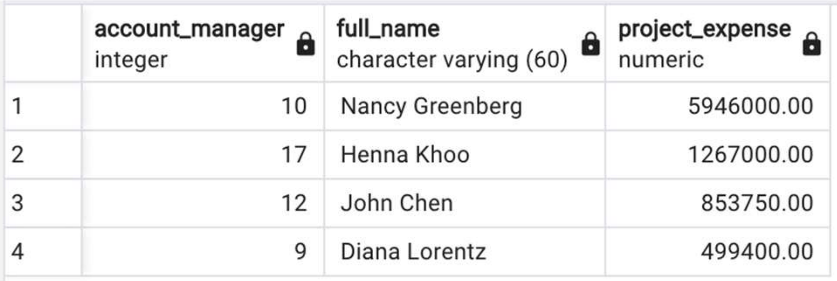
male_salary numeric	female_salary numeric	avg_wage_gap numeric
135125.000000000000	139333.333333333333	-4208.333333333333



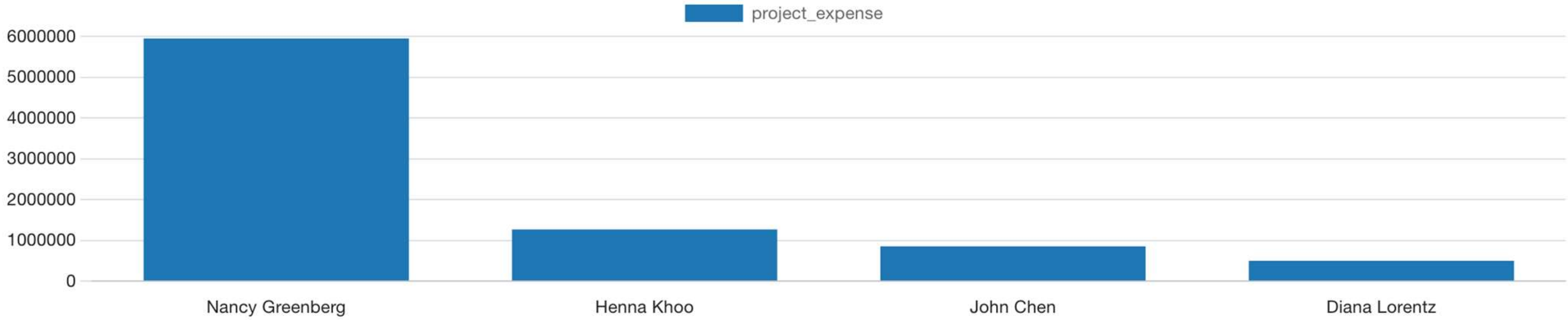
Query 7: Analyzing Account Manager Performance



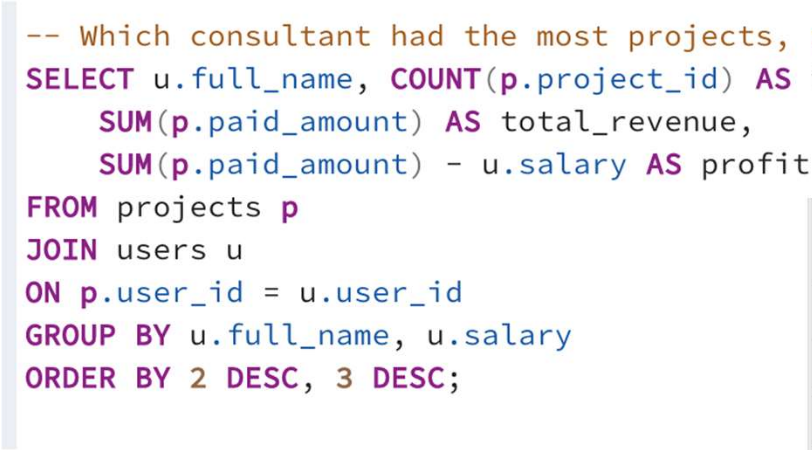
```
-- Which account manager brought in the most revenue?
SELECT a.account_manager, u.full_name,
    SUM(paid_amount) AS project_expense
FROM projects as p
JOIN accounts as a
ON p.account_id = a.account_id
JOIN users as u
ON a.account_manager = u.user_id
GROUP BY a.account_manager, u.full_name
HAVING SUM(paid_amount) > 0
ORDER BY 3 DESC;
```



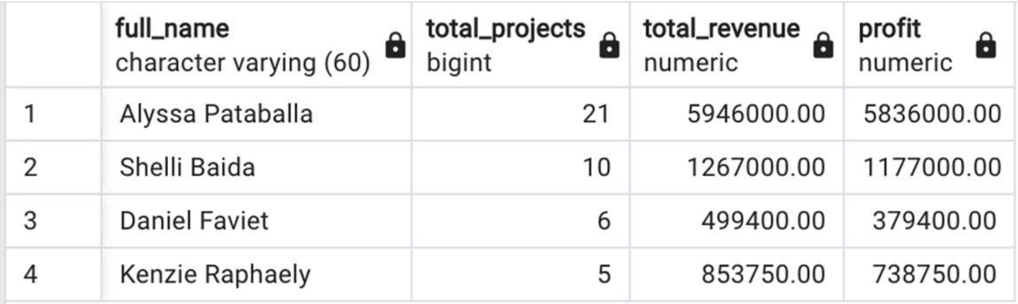
	account_manager integer	full_name character varying (60)	project_expense numeric
1	10	Nancy Greenberg	5946000.00
2	17	Henna Khoo	1267000.00
3	12	John Chen	853750.00
4	9	Diana Lorentz	499400.00



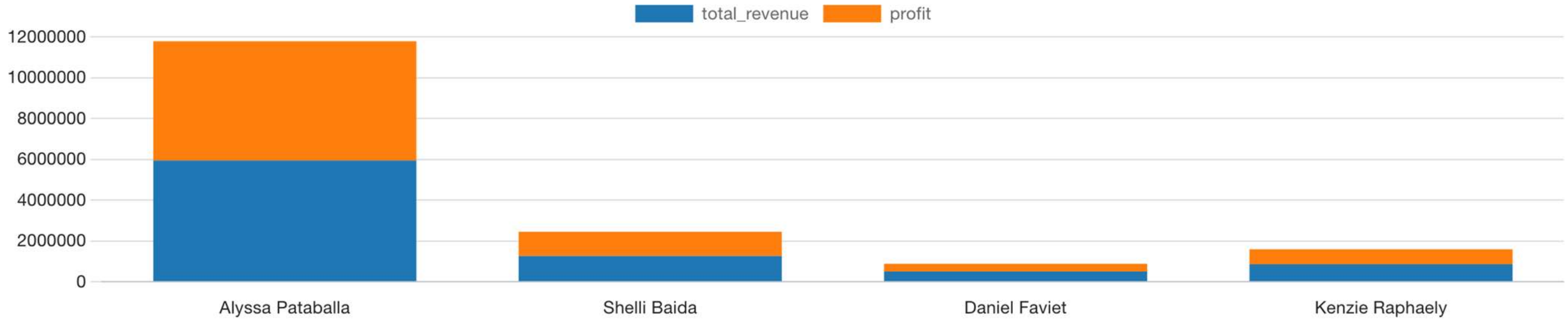
Query 8: Analyzing Consultant Profitability



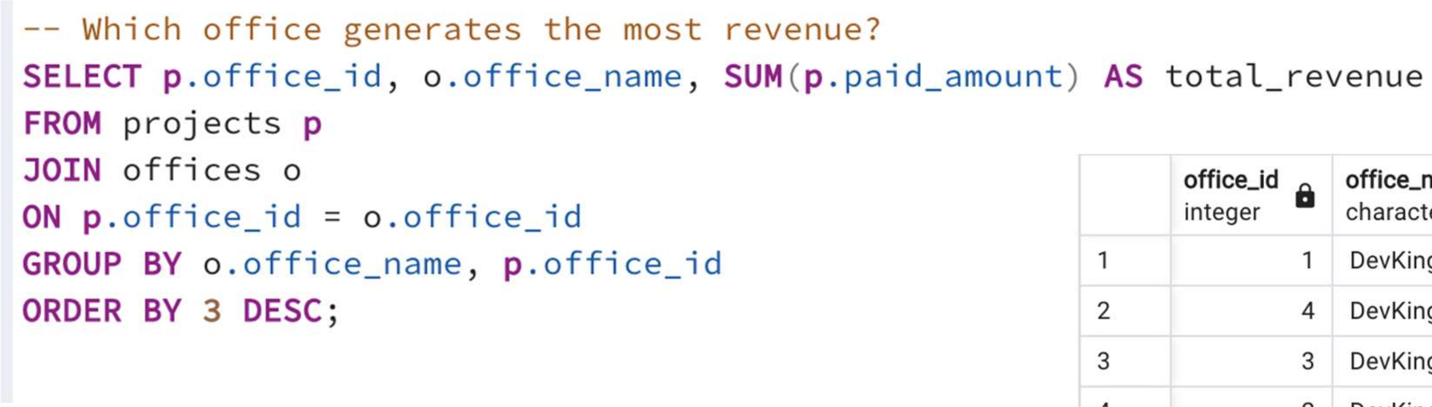
```
-- Which consultant had the most projects, revenue and profit?
SELECT u.full_name, COUNT(p.project_id) AS total_projects,
    SUM(p.paid_amount) AS total_revenue,
    SUM(p.paid_amount) - u.salary AS profit
FROM projects p
JOIN users u
ON p.user_id = u.user_id
GROUP BY u.full_name, u.salary
ORDER BY 2 DESC, 3 DESC;
```



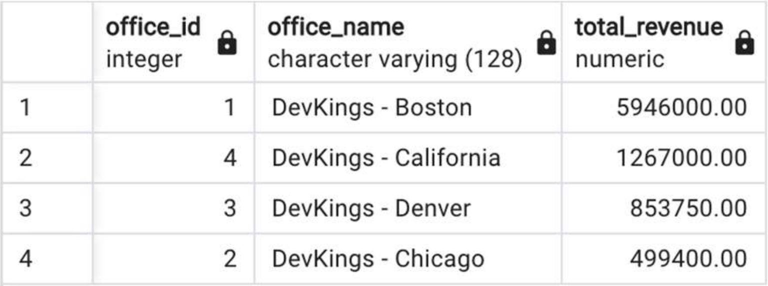
	full_name character varying (60)	total_projects bigint	total_revenue numeric	profit numeric
1	Alyssa Pataballa	21	5946000.00	5836000.00
2	Shelli Baida	10	1267000.00	1177000.00
3	Daniel Faviet	6	499400.00	379400.00
4	Kenzie Raphaely	5	853750.00	738750.00



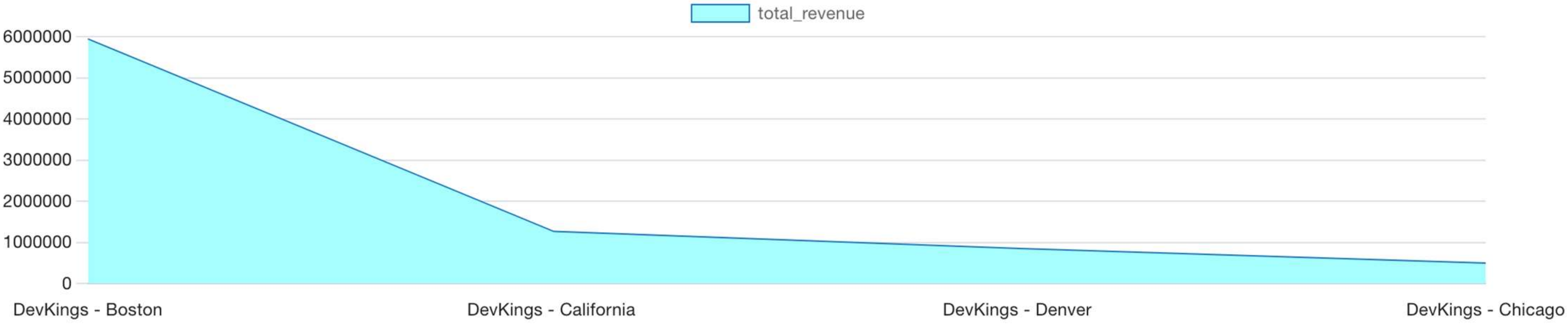
Query 9: Analyzing Office Profitability



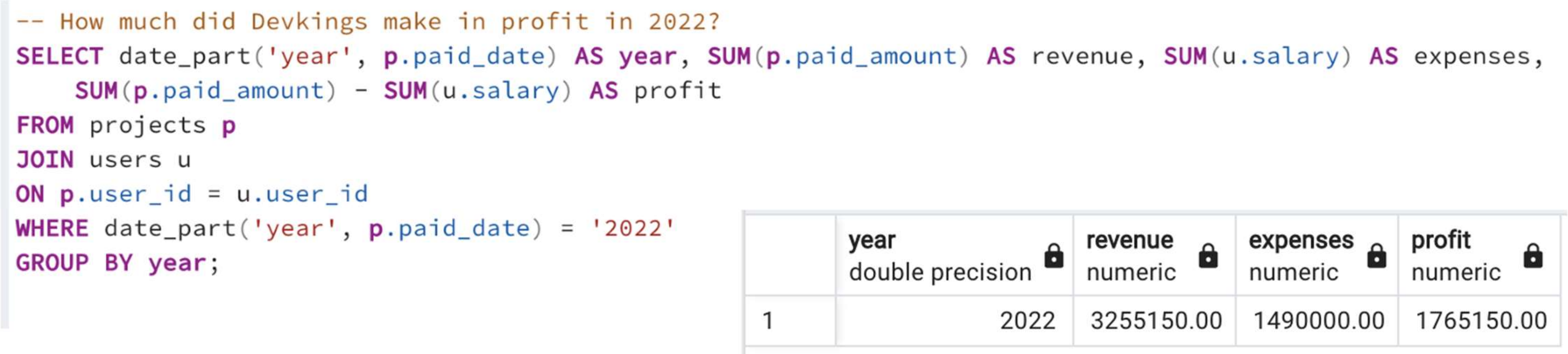
```
-- Which office generates the most revenue?
SELECT p.office_id, o.office_name, SUM(p.paid_amount) AS total_revenue
FROM projects p
JOIN offices o
ON p.office_id = o.office_id
GROUP BY o.office_name, p.office_id
ORDER BY 3 DESC;
```



	office_id integer	office_name character varying (128)	total_revenue numeric
1	1	DevKings - Boston	5946000.00
2	4	DevKings - California	1267000.00
3	3	DevKings - Denver	853750.00
4	2	DevKings - Chicago	499400.00

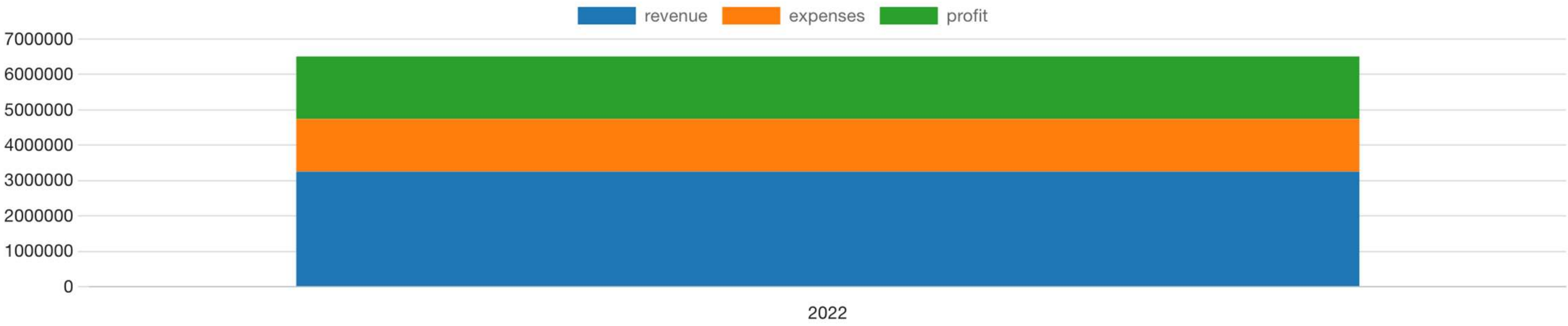


Query 10: DevKings Profit Analysis for 2022

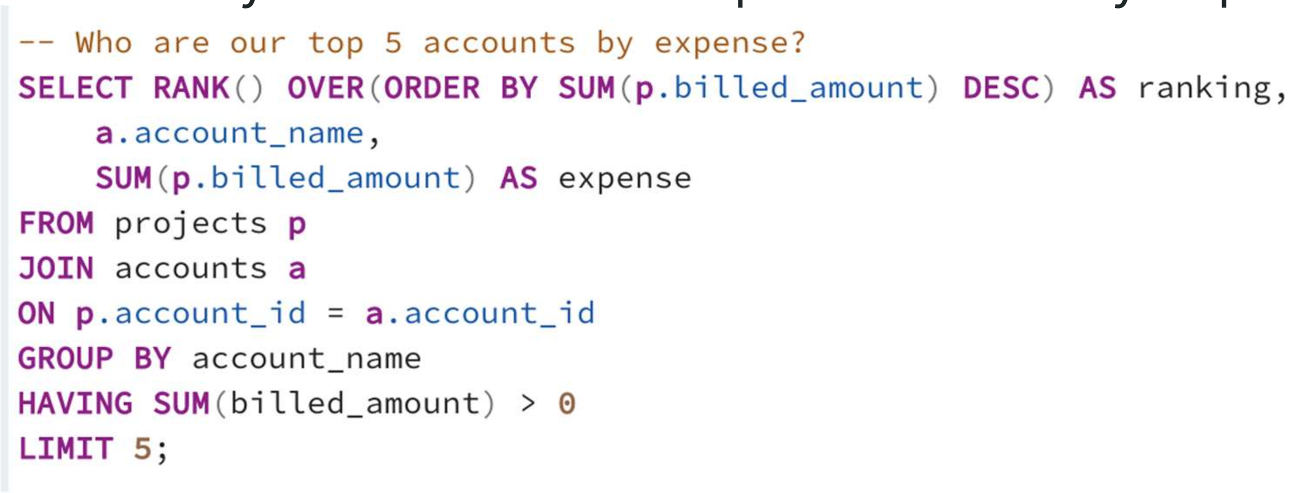


```
-- How much did Devkings make in profit in 2022?
SELECT date_part('year', p.paid_date) AS year, SUM(p.paid_amount) AS revenue, SUM(u.salary) AS expenses,
    SUM(p.paid_amount) - SUM(u.salary) AS profit
FROM projects p
JOIN users u
ON p.user_id = u.user_id
WHERE date_part('year', p.paid_date) = '2022'
GROUP BY year;
```

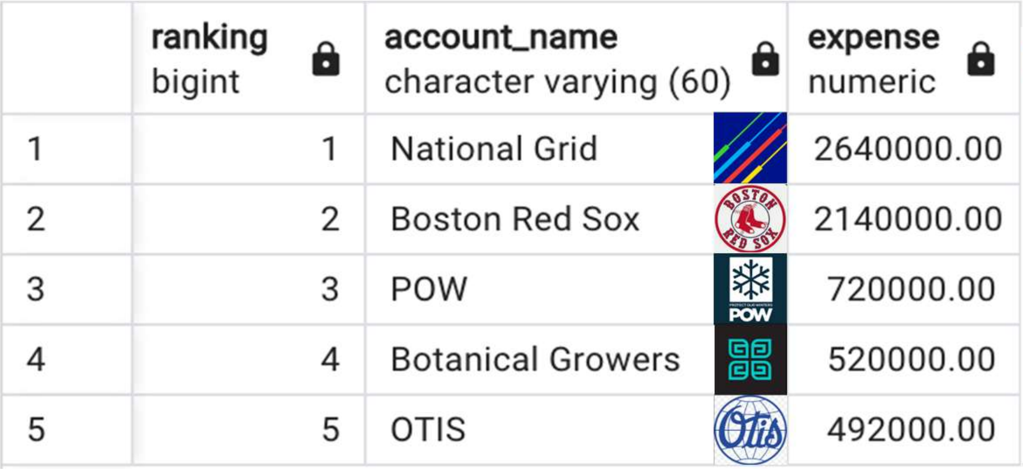
	year double precision 🔒	**revenue** numeric 🔒	**expenses** numeric 🔒	**profit** numeric 🔒
1	2022	3255150.00	1490000.00	1765150.00



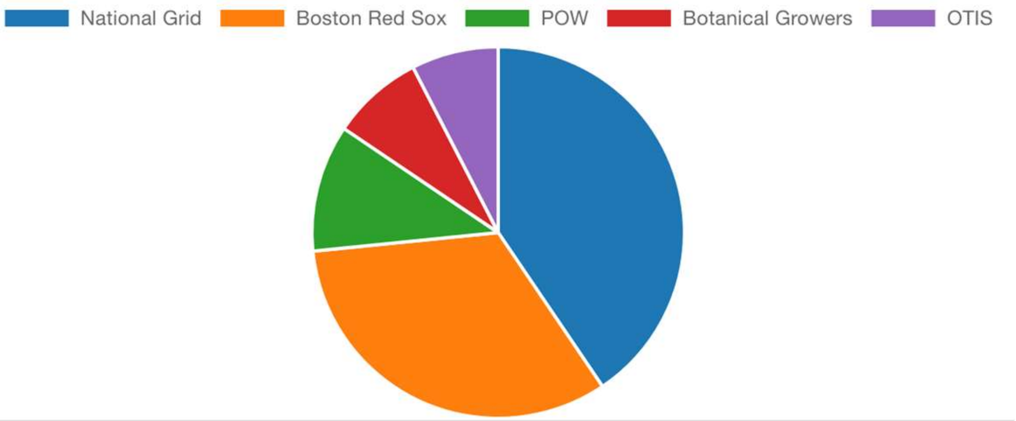
Query 11: Who are our top 5 accounts by expense?



```sql
-- Who are our top 5 accounts by expense?
SELECT RANK() OVER(ORDER BY SUM(p.billed_amount) DESC) AS ranking,
    a.account_name,
    SUM(p.billed_amount) AS expense
FROM projects p
JOIN accounts a
ON p.account_id = a.account_id
GROUP BY account_name
HAVING SUM(billed_amount) > 0
LIMIT 5;
```



	ranking bigint 🔒	account_name character varying (60) 🔒	expense numeric 🔒
1	1	National Grid	2640000.00
2	2	Boston Red Sox	2140000.00
3	3	POW	720000.00
4	4	Botanical Growers	520000.00
5	5	OTIS	492000.00



Thank you ☺

DK

DEVKINGS

CONSULTING GROUP

Questions?