-- Project Questions

------Steve-------

-- 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

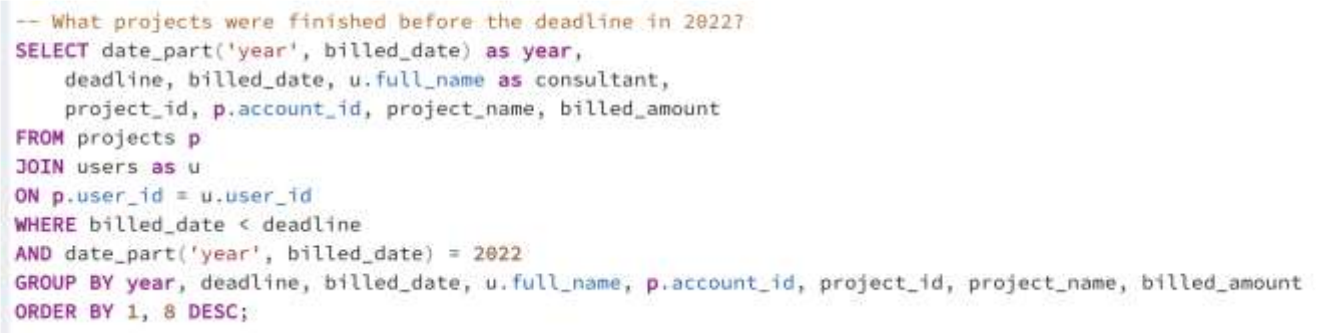
ON p.user_id = u.user_id

WHERE billed_date < deadline

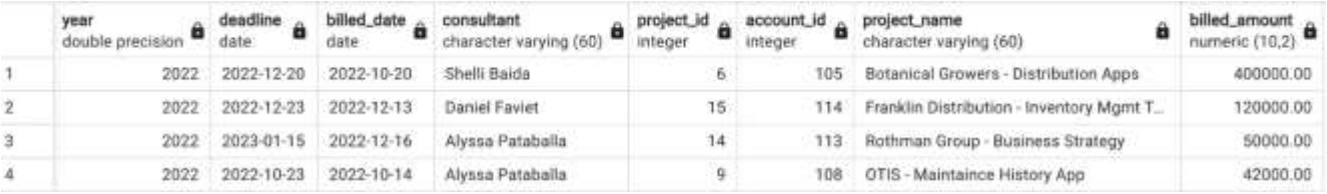
AND date_part('year', billed_date) = 2022

GROUP BY year, deadline, billed_date, u.full_name, p.account_id, project_id, project_name, billed_amount

ORDER BY 1, 8 DESC;



```
-- 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

-- 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

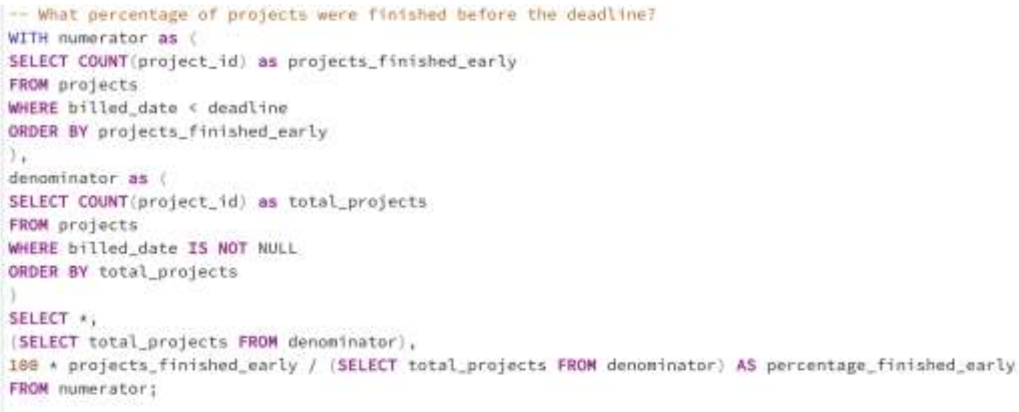
ORDER BY total_projects

)

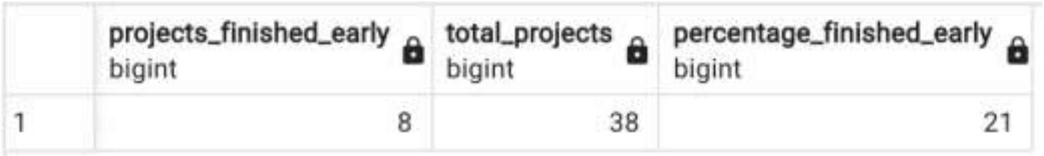
SELECT *,

100 * projects_finished_early / (SELECT total_projects FROM denominator) AS percentage_finished_early

FROM numerator;



```sql
-- 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

-- 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

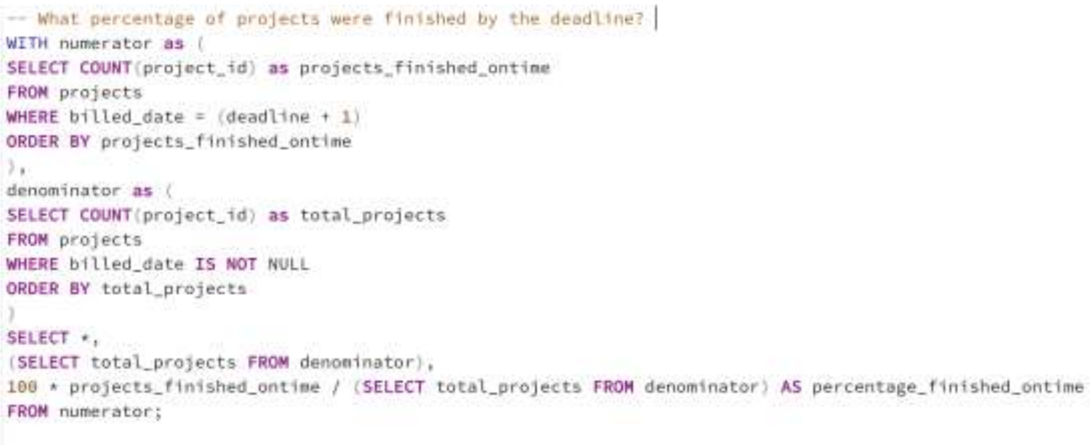
)

SELECT *,

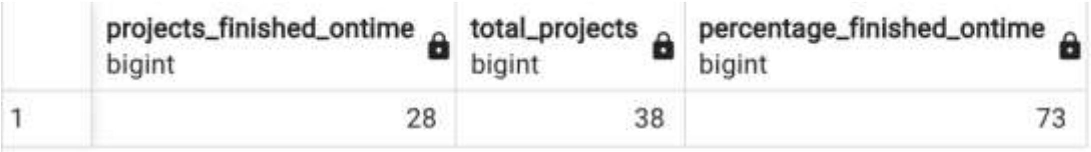
(SELECT total_projects FROM denominator),

100 * projects_finished_ontime / (SELECT total_projects FROM denominator) AS percentage_finished_ontime

FROM numerator;



```
-- 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

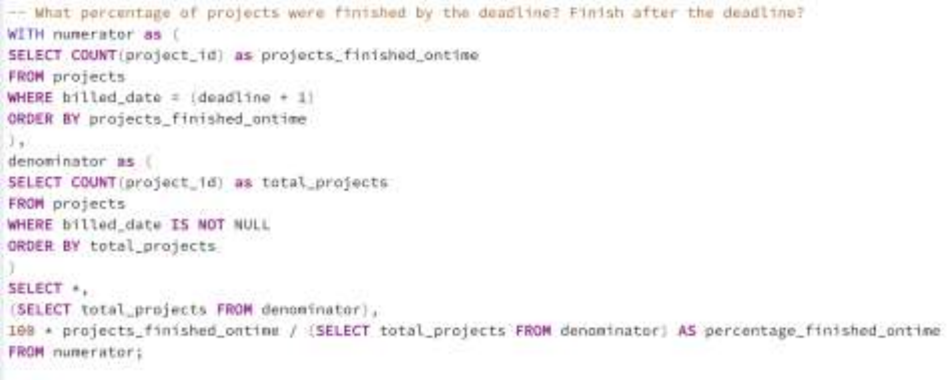
-- What percentage of projects were finished after the deadline?

WITH numerator as (

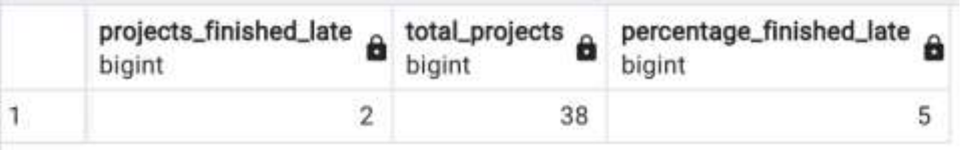
SELECT COUNT(project_id) as projects_finished_late

FROM projects

```
WHERE billed_date > (deadline + 1)

ORDER BY projects_finished_late

),

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_late / (SELECT total_projects FROM denominator) AS percentage_finished_late

FROM numerator;
```



```
-- 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

```
-- 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),

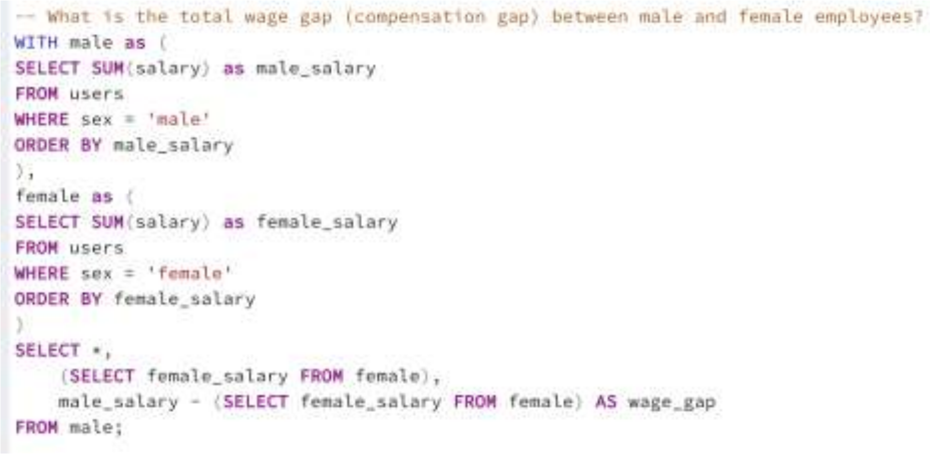
male_salary - (SELECT female_salary FROM female) AS wage_gap

FROM male;

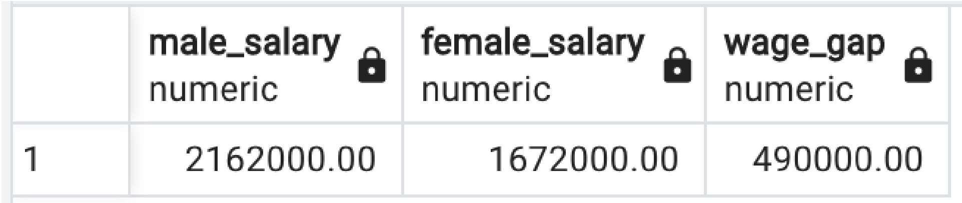
Total Male Compensation: $2,162,000

Total Female Compensation: $1,672,000

Compensation Gap: $490,000 (more for male employees)



```
-- 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;
```



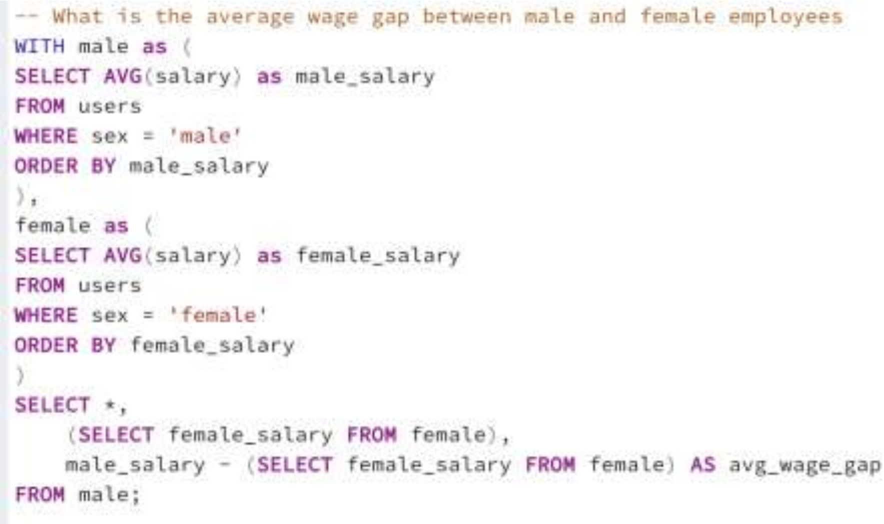
	male_salary numeric	female_salary numeric	wage_gap numeric
1	2162000.00	1672000.00	490000.00

-- What is the average wage gap between male and female employees

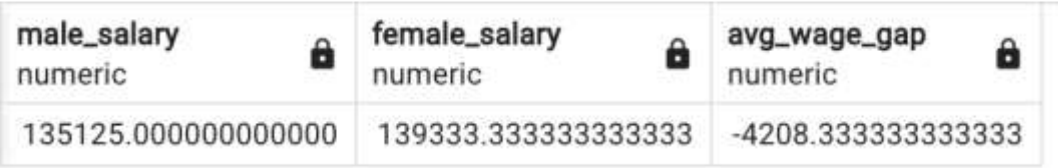
WITH male as (

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;
```



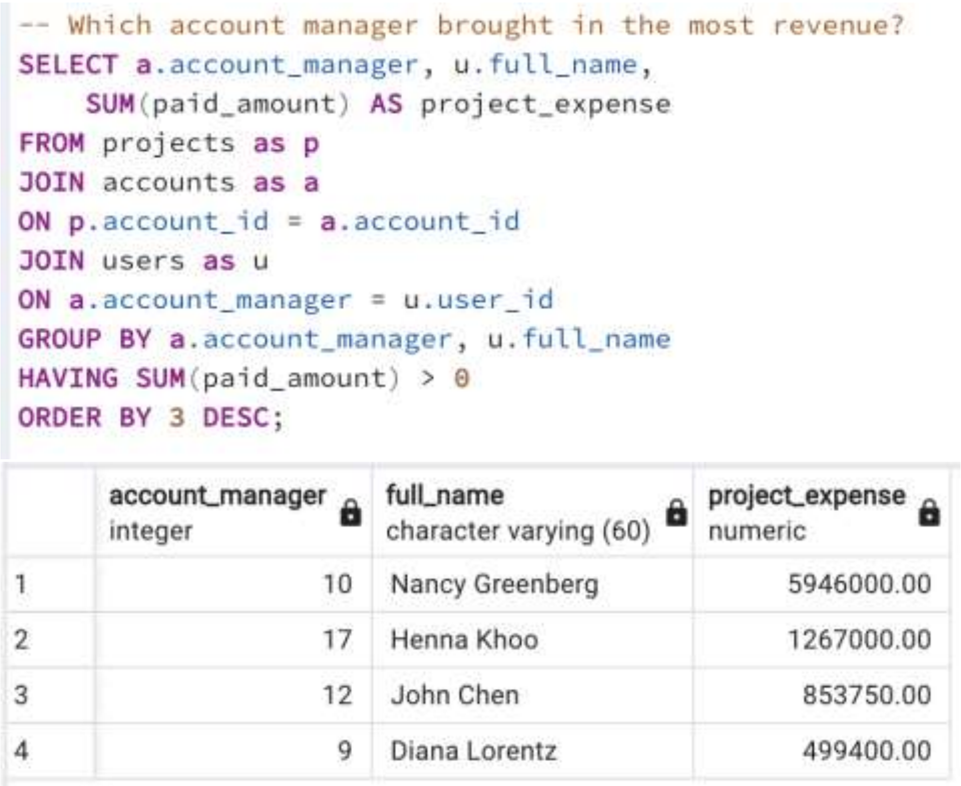
```
-- What is the average wage gap between male and female employees
WITH male as (
SELECT AVG(salary) as male_salary
FROM users
WHERE sex = 'male'
ORDER BY male_salary
),
female as (
SELECT AVG(salary) as female_salary
FROM users
WHERE sex = 'female'
ORDER BY female_salary
)
SELECT *,
    (SELECT female_salary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

-- 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;
```

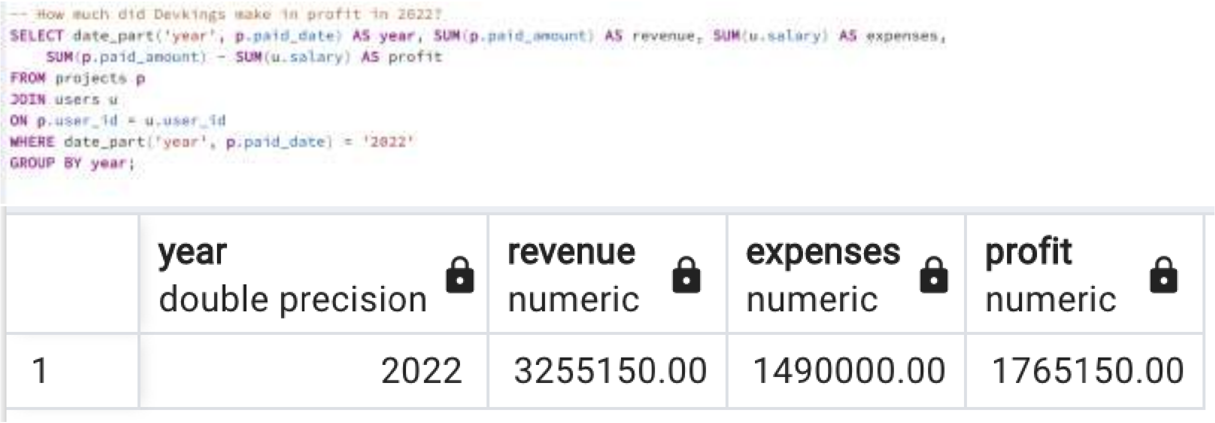


```
-- Which account manager brought in the most revenue?
SELECT a.account_manager, u.full_name,
    SUM(paid_amount) AS project_expense
FROM projects as p
JOIN accounts as a
ON p.account_id = a.account_id
JOIN users as u
ON a.account_manager = u.user_id
GROUP BY a.account_manager, u.full_name
HAVING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

------------- Noah -------------

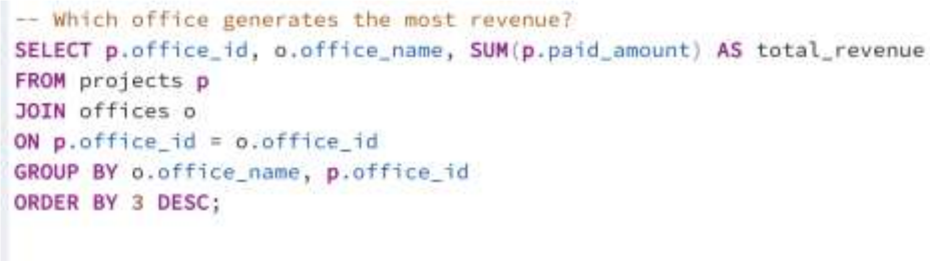
```
-- How much did Devkings make in profit in 2022?

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;
```



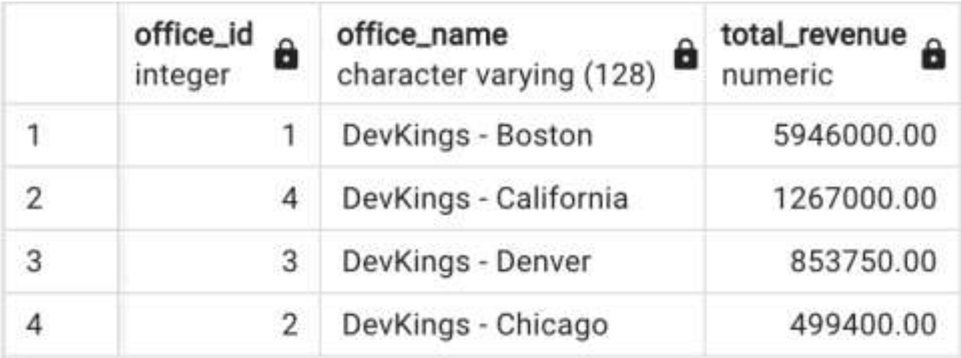
```
-- How much did Devkings make in profit in 2022?
SELECT date_part('year', p.paid_date) AS year, SUM(p.paid_amount) AS revenue, SUM(u.salary) AS expenses,
    SUM(p.paid_amount) - SUM(u.salary) AS profit
FROM projects p
JOIN users u
ON p.user_id = u.user_id
WHERE date_part('year', p.paid_date) = '2022'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

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

```
-- 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;
```



```sql
-- 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

-- 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

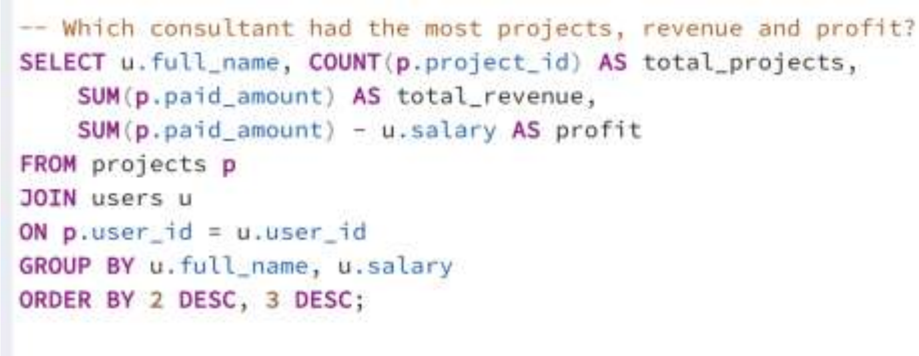
FROM projects p

JOIN users u

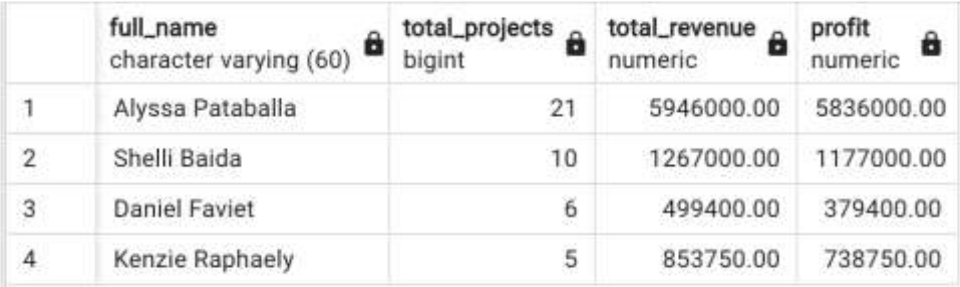
ON p.user_id = u.user_id

GROUP BY u.full_name, u.salary

ORDER BY 2 DESC, 3 DESC;



```sql
-- 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

-- 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

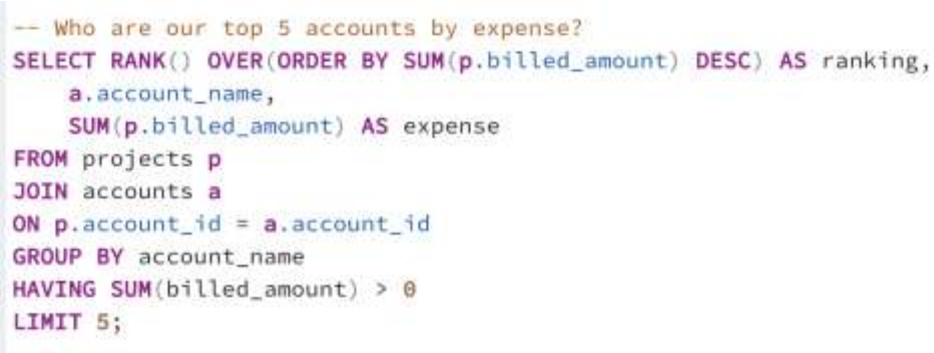
JOIN accounts a

ON p.account_id = a.account_id

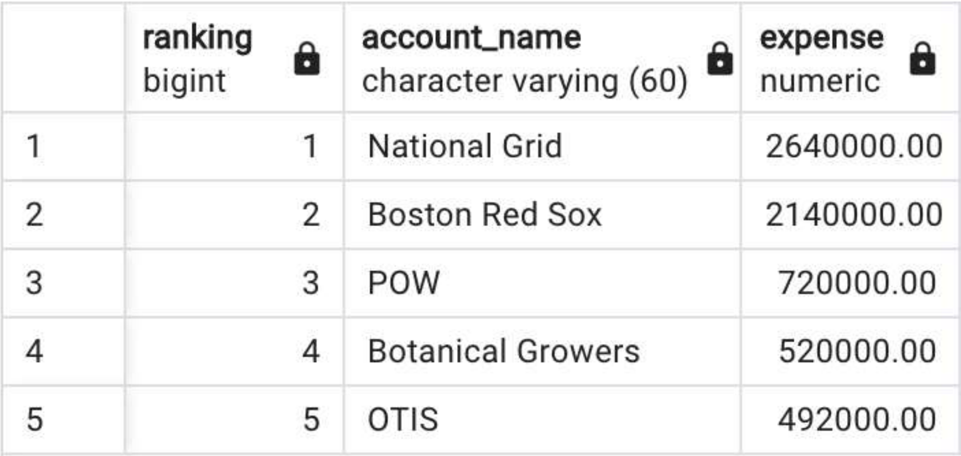
GROUP BY account_name

HAVING SUM(billed_amount) > 0

LIMIT 5;



```
-- 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

--------- Nate ---------

-- What was the outstanding balance by year?

SELECT date_part('year', paid_date) AS year,

SUM(paid_amount - billed_amount) AS outstanding_balance

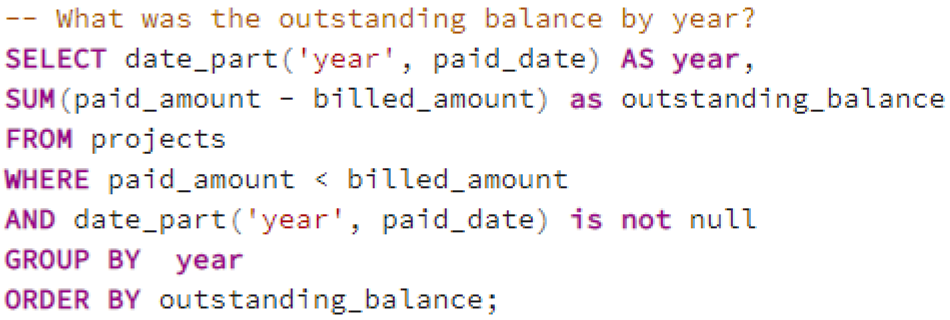
FROM projects

WHERE paid_amount < billed_amount

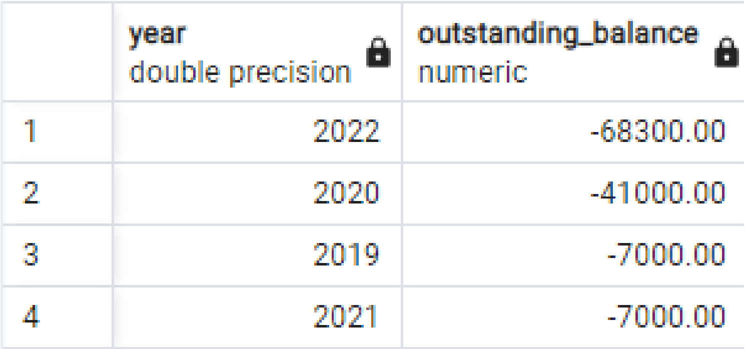
AND date_part('year', paid_date) IS NOT NULL

GROUP BY year

ORDER BY year;



```
-- 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

-- What is the Outstranding 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

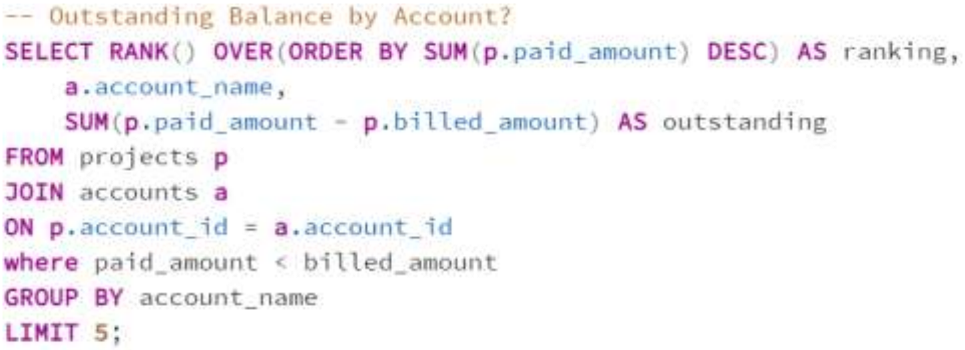
JOIN accounts a

ON p.account_id = a.account_id

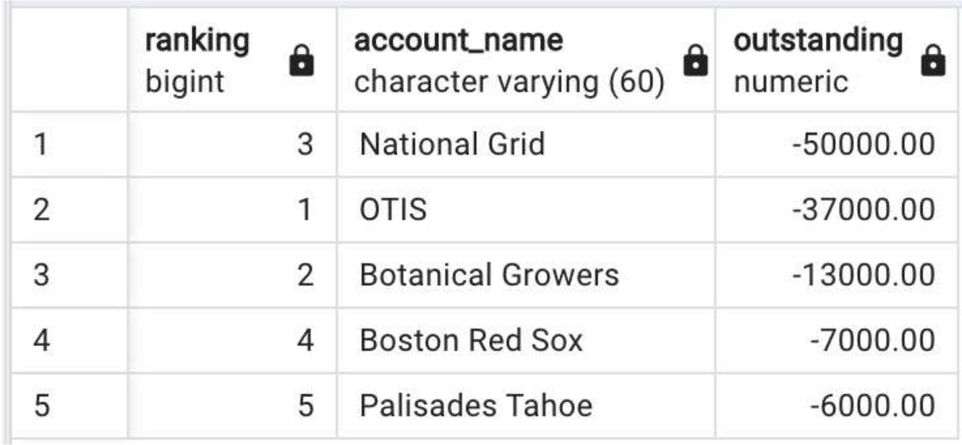
where paid_amount < billed_amount

GROUP BY account_name

LIMIT 5;



```
-- 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

-- Count the number of employees whose salary is

-- higher than the average of that year (across all departments all years)

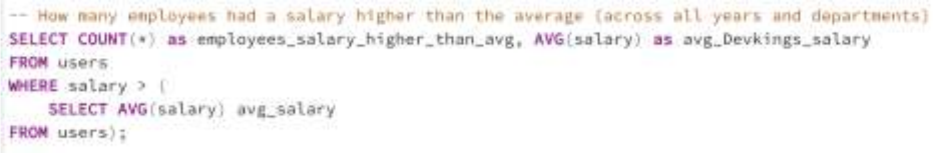
SELECT user_id, first_name, last_name

FROM users

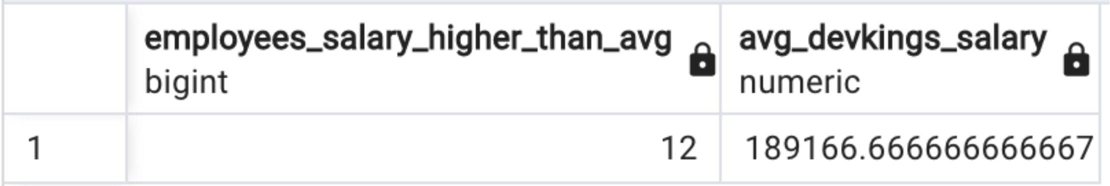
WHERE salary > (

SELECT AVG(salary) avg_salary

FROM users);



```
-- 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);
```



	employees_salary_higher_than_avg bigint	avg_devkings_salary numeric
1	12	189166.666666666667

-- How much revenue did the company generate in 2019, 2020, 2021 and 2022?

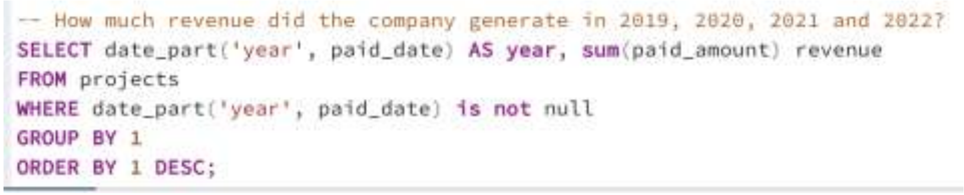
SELECT date_part('year', paid_date) AS year, sum(paid_amount) rev

FROM projects

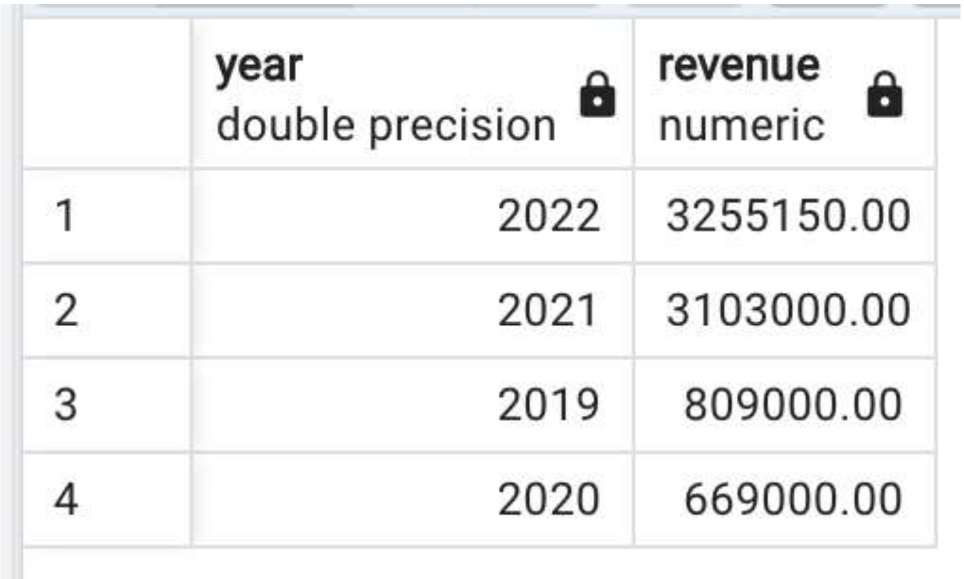
where date_part('year', paid_date) is not null

GROUP BY 1

ORDER BY 2 DESC;



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



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