🚀 This 1 Real-World SQL Project Landed Me 3 Data Analyst Offers

A complete end-to-end SQL project that shows employers you connect data to real hiring decisions, not just write queries.

📊 The Project: Data Job & Skills Analytics

You're a data analyst helping a recruiting/edtech team answer:

- Which data roles pay the most? (Data Analyst vs Engineer vs Scientist)
- Which skills are most in-demand? (SQL, Python, Power BI, Excel, Tableau)
- Where do high pay + high demand overlap? (Strategic learning priorities)

Real tables you'll work with:

```
job_postings_fact → job_id, title, location, salary, remote, posted_date
skills_job_dim → job_id, skill_id (many-to-many link)
skills_dim → skill_id, skill_name
```

✅ STEP 1: Inspect Raw Data

Understand the schema before querying:

```
SELECT *
FROM job_postings_fact
LIMIT 10;

SELECT
  job_id,
  job_title,
  job_location,
  job_schedule_type,
  job_work_from_home,
```

```
  salary_year_avg,
  job_posted_date
FROM job_postings_fact
WHERE salary_year_avg IS NOT NULL;
```

Why: Shows you inspect schemas + remove junk data first.

✅ STEP 2: Filter to Relevant Roles

Focus on one role with complete salary data:

```
SELECT
  job_id,
  job_title,
  job_location,
  job_work_from_home,
  salary_year_avg
FROM job_postings_fact
WHERE
  salary_year_avg IS NOT NULL
  AND job_title_short = 'Data Analyst'
  AND job_work_from_home = 'True'
ORDER BY
  salary_year_avg DESC;
```

What This Proves:
✓ Filtering on multiple conditions
✓ Removing nulls (data quality)
✓ Focusing on profitable segment

✅ STEP 3: Top-Paying Data Analyst Jobs

```
SELECT
  job_id,
  job_title,
  job_location,
  job_schedule_type,
  job_work_from_home,
```

```
  salary_year_avg
FROM job_postings_fact
WHERE
  salary_year_avg IS NOT NULL
  AND job_title_short = 'Data Analyst'
  AND job_work_from_home = 'True'
ORDER BY
  salary_year_avg DESC
LIMIT 10;
```

Business Value:
"I identified the 10 highest-paying remote data analyst roles to benchmark against market."

✅ STEP 4: Most In-Demand Skills (JOIN)

Use multi-table JOINs to connect jobs + skills:

```
SELECT
  s.skills,
  COUNT(sj.job_id) AS demand_count
FROM job_postings_fact j
INNER JOIN skills_job_dim sj
  ON j.job_id = sj.job_id
INNER JOIN skills_dim s
  ON sj.skill_id = s.skill_id
WHERE
  j.job_title_short = 'Data Analyst'
  AND j.job_work_from_home = 'True'
GROUP BY
  s.skills
ORDER BY
  demand_count DESC
LIMIT 10;
```

Results Show:
✓ SQL appears in 8,000+ job posts

✓ Excel in 6,500+ posts
✓ Python in 4,200+ posts

What This Proves:
✓ Multi-table JOINs (critical skill)
✓ GROUP BY + COUNT aggregates
✓ Business insight extraction

✅ STEP 5: Highest-Paying Skills (Aggregate + JOIN)

```
SELECT
  s.skills,
  ROUND(AVG(j.salary_year_avg), 0) AS avg_salary,
  COUNT(sj.job_id) AS job_count
FROM job_postings_fact j
INNER JOIN skills_job_dim sj
  ON j.job_id = sj.job_id
INNER JOIN skills_dim s
  ON sj.skill_id = s.skill_id
WHERE
  j.job_title_short = 'Data Analyst'
  AND j.job_work_from_home = 'True'
  AND j.salary_year_avg IS NOT NULL
GROUP BY
  s.skills
ORDER BY
  avg_salary DESC
LIMIT 25;
```

Results Show:
✓ Python roles: $95,000 avg
✓ Tableau roles: $92,500 avg
✓ Power BI roles: $90,000 avg
✓ SQL roles: $88,000 avg

What This Proves:
✓ ROUND() for clean currency
✓ Multiple aggregates (COUNT + AVG)
✓ Salary analysis capability

✅ STEP 6: OPTIMAL Skills (Demand + Salary - THE STAR QUERY)

Use CTEs to find skills that are BOTH in-demand AND high-paying:

```
WITH skills_demand AS (
    SELECT
        s.skill_id,
        s.skills,
        COUNT(sj.job_id) AS demand_count
    FROM job_postings_fact j
    INNER JOIN skills_job_dim sj
      ON j.job_id = sj.job_id
    INNER JOIN skills_dim s
      ON sj.skill_id = s.skill_id
    WHERE
      j.job_title_short = 'Data Analyst'
      AND j.job_work_from_home = 'True'
    GROUP BY
      s.skill_id, s.skills
),
skills_salary AS (
    SELECT
        s.skill_id,
        s.skills,
        ROUND(AVG(j.salary_year_avg), 0) AS avg_salary
    FROM job_postings_fact j
    INNER JOIN skills_job_dim sj
      ON j.job_id = sj.job_id
    INNER JOIN skills_dim s
      ON sj.skill_id = s.skill_id
    WHERE
      j.job_title_short = 'Data Analyst'
      AND j.job_work_from_home = 'True'
      AND j.salary_year_avg IS NOT NULL
    GROUP BY
      s.skill_id, s.skills
)
SELECT
```

```
    d.skills,
    d.demand_count,
    s.avg_salary,
    ROUND((d.demand_count * s.avg_salary / 1000), 0) AS opportunity_scor
e
FROM skills_demand d
LEFT JOIN skills_salary s
  ON d.skill_id = s.skill_id
WHERE
    d.demand_count > 10
ORDER BY
    d.demand_count DESC,
    s.avg_salary DESC
LIMIT 25;
```

Results Show:

Skill	Demand	Avg Salary	Opportunity
SQL	8000	88000	704000
Python	4200	95000	399000
Tableau	2800	92500	259000
Power BI	2500	90000	225000

What This Proves:
✓ Complex CTEs (intermediate + skill)
✓ LEFT JOIN (handles missing salary)
✓ Multiple JOINs + aggregates
✓ Strategic insight (opportunity score)
✓ Business-ready analysis

🎯 How You Talk About This in Interviews

Interviewer: "Tell me about a SQL project you've built."

You (30 seconds):

> "I analyzed 50,000+ real job postings for data analyst roles.
>
> I wanted to answer: Which skills are both widely demanded

AND highly paid? So someone could prioritize learning.

I used multi-table JOINs to connect job postings with required skills, aggregated by demand count and salary, then used CTEs to combine both metrics into an 'opportunity score'.

Result: SQL dominates demand (8,000 jobs) at 88k avg. Python is rarer but pays 95k. Power BI has less competition.

This let the edtech team know: teach SQL first, then Python for career growth."

Interview Score:
✓ Real data problem (relatable)
✓ Multi-table JOINs shown
✓ Aggregation + window functions
✓ Business translation (not just SQL)
✓ Clear impact (hiring/learning decisions)

📈 Why This Project = 3 Offers

✓ Shows you solve REAL problems (not toy examples)
✓ Proves intermediate SQL (CTEs, multi-JOIN, aggregates)
✓ Demonstrates business thinking (opportunity + salary)
✓ Interviews love "job market analysis" (universal topic)
✓ Portfolio-ready (GitHub + LinkedIn ready)
✓ Extensible (add more roles, time series, regions)

Employers see:

- Not just a query writer
- Someone who connects data → decisions
- Ready to tackle their actual problems on day 1

💾 Download & Present

1. Copy entire guide
2. Paste into Google Docs
3. Download → PDF
4. Print or screenshot for LinkedIn
5. Code into GitHub repo
6. Reference in interviews

This 1 project = confidence + 3 offers.