Karol Harasim

Al Sallami

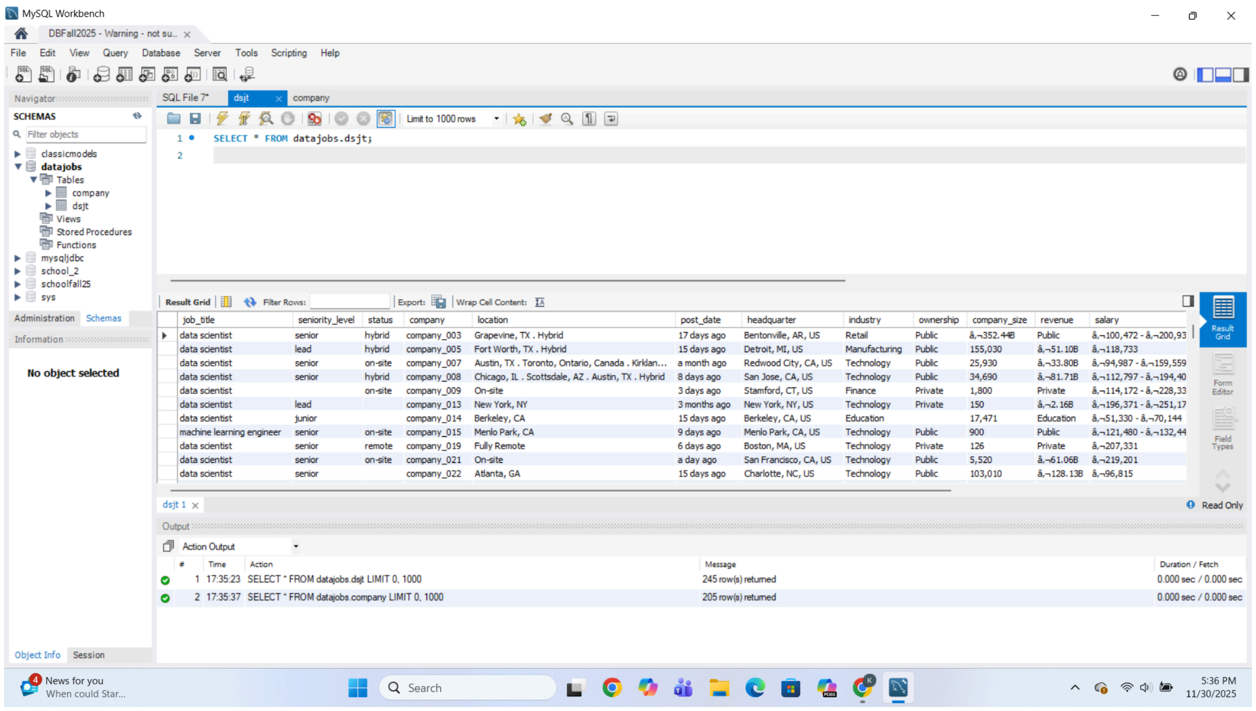
CS286

12/12/25

Data Normalization Summary

For my project I was interested in data involving data science careers. I chose a table of data from Kaggle called Data Science Careers & Salaries 2025. I liked this table because it listed over 200 jobs in the data science field with thirteen attributes describing each listing. My plan for this table was to normalize its data into three separate tables and create queries to draw insightful conclusions about careers in data science.

I started this project by downloading the table from Kaggle as a csv file. I then imported the csv file into MySQL Workbench in a new schema to create my first table.

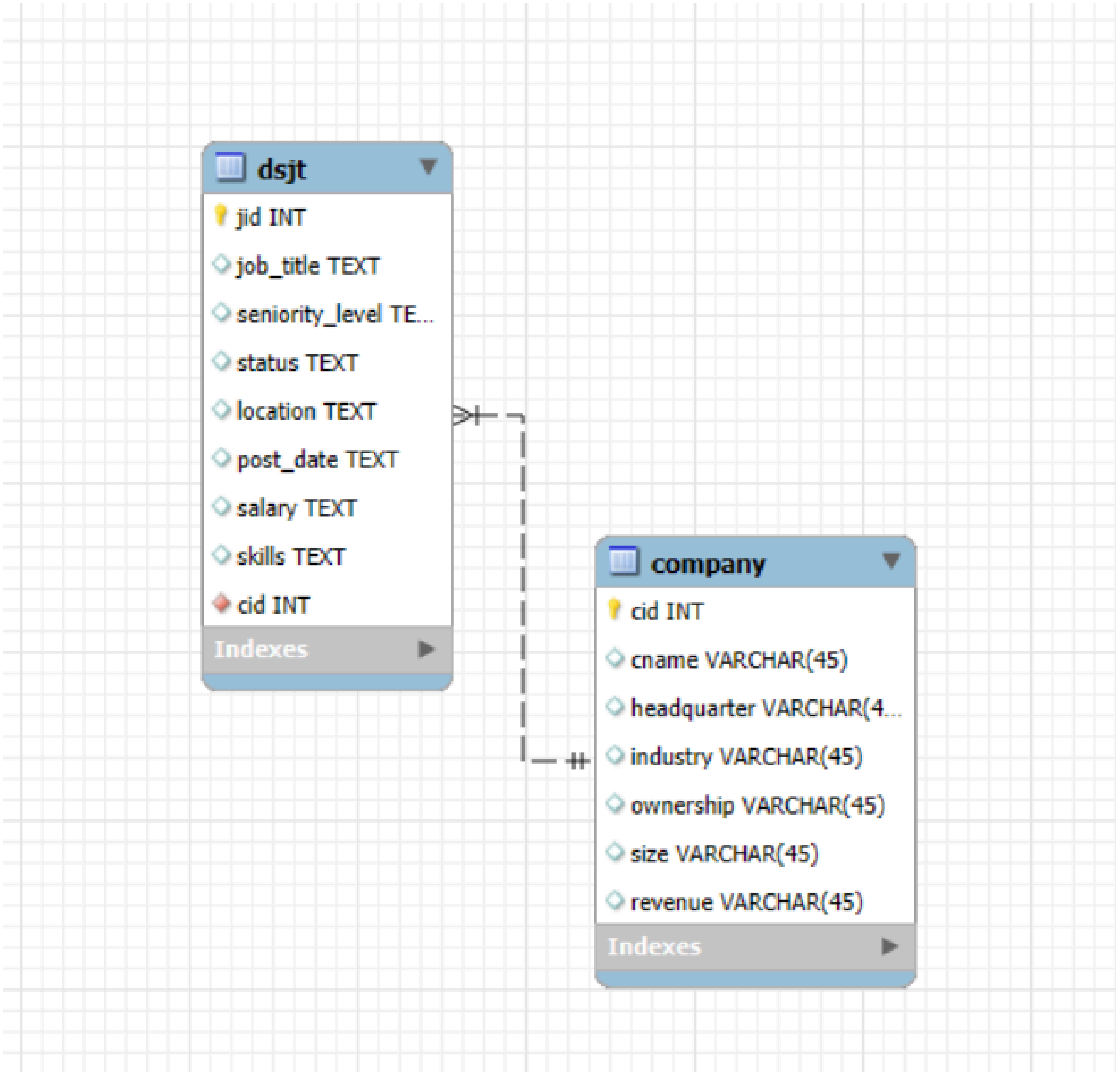


```
SELECT * FROM datajobs.dsjt;
```

job_title	seniority_level	status	company	location	post_date	headquarter	industry	ownership	company_size	revenue	salary
data scientist	senior	hybrid	company_003	Grapevine, TX . Hybrid	17 days ago	Bentonville, AR, US	Retail	Public	â,¬352.44B	Public	â,¬100,472 - â,¬200,93
data scientist	lead	hybrid	company_005	Fort Worth, TX . Hybrid	15 days ago	Detroit, MI, US	Manufacturing	Public	155,030	â,¬51.10B	â,¬118,733
data scientist	senior	on-site	company_007	Austin, TX . Toronto, Ontario, Canada . Kirklan...	a month ago	Redwood City, CA, US	Technology	Public	25,930	â,¬33.80B	â,¬94,987 - â,¬159,559
data scientist	senior	hybrid	company_008	Chicago, IL . Scottsdale, AZ . Austin, TX . Hybrid	8 days ago	San Jose, CA, US	Technology	Public	34,690	â,¬81.71B	â,¬112,797 - â,¬194,40
data scientist		on-site	company_009	On-site	3 days ago	Stamford, CT, US	Finance	Private	1,800	Private	â,¬114,172 - â,¬228,33
data scientist	lead		company_013	New York, NY	3 months ago	New York, NY, US	Technology	Private	150	â,¬2.16B	â,¬196,371 - â,¬251,17
data scientist	junior		company_014	Berkeley, CA	15 days ago	Berkeley, CA, US	Education		17,471	Education	â,¬51,330 - â,¬70,144
machine learning engineer	senior	on-site	company_015	Menlo Park, CA	9 days ago	Menlo Park, CA, US	Technology	Public	900	Public	â,¬121,480 - â,¬132,44
data scientist	senior	remote	company_019	Fully Remote	6 days ago	Boston, MA, US	Technology	Private	126	Private	â,¬207,331
data scientist	senior	on-site	company_021	On-site	a day ago	San Francisco, CA, US	Technology	Public	5,520	â,¬61.06B	â,¬219,201
data scientist	senior		company_022	Atlanta, GA	15 days ago	Charlotte, NC, US	Technology	Public	103,010	â,¬128.13B	â,¬96,815

```
#  Time      Action                                       Message                Duration / Fetch
1  17:35:23  SELECT * FROM datajobs.dsjt LIMIT 0, 1000    245 row(s) returned    0.000 sec / 0.000 sec
2  17:35:37  SELECT * FROM datajobs.company LIMIT 0, 1000 205 row(s) returned    0.000 sec / 0.000 sec
```

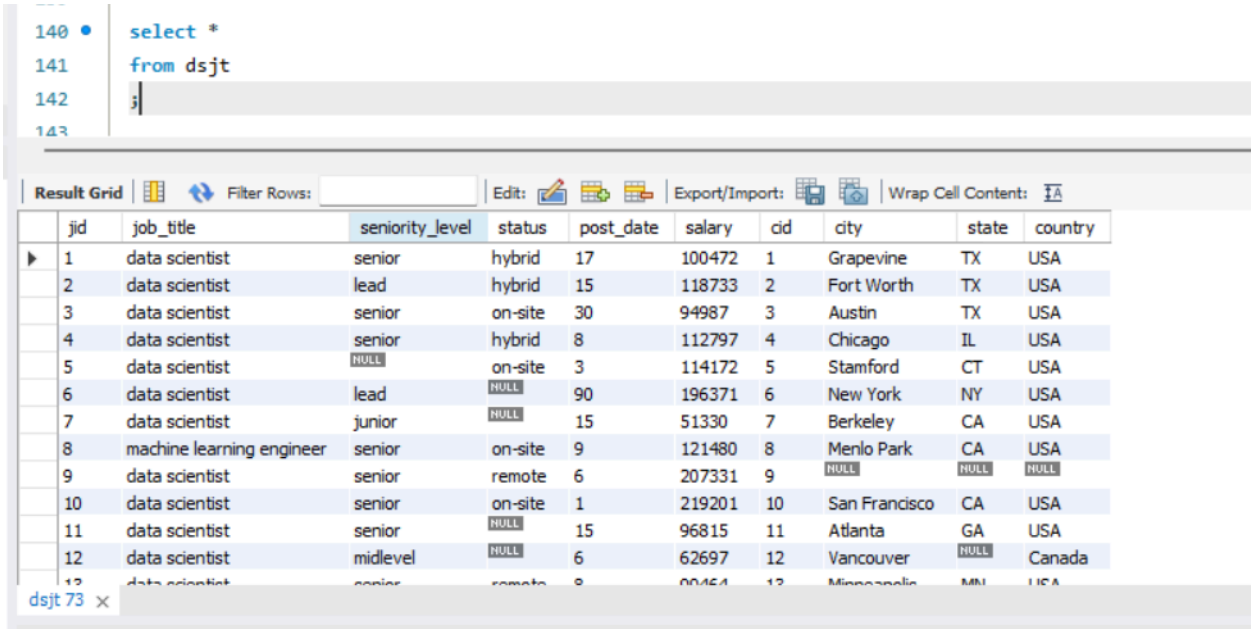
I started by creating a job id for each individual job. Then I created a new table called company with all of the columns from the jobs table related to companies. I also added a company id column to this table for each unique company. I first inserted each unique company name into the company table. I then linked the two tables on the company name columns and added the rest of the data to the company table. I then added a company id column to the jobs table as a foreign key to link the two tables together with a simple integer identifier.



Next I created the skills table. Before I inserted skills from the jobs table into the skills table I had to reformat the skills values so they would be easier to work with. I split the skills into individual substrings and changed the skill column to contain arrays which contained the substrings. Then I was able to select unique skill values and insert them into the skills table with a corresponding skill id. Next I created the job skills table which is an associative table to link the many to many relationship between the jobs and skills tables.

Importing the table and creating new tables was not too difficult because of previous practice of doing so in assignments. I found it difficult to insert data into the new tables, especially the skills table. Working with arrays in SQL is a new topic and I had to learn this on my own, so this part of the project took longer than compared to others.

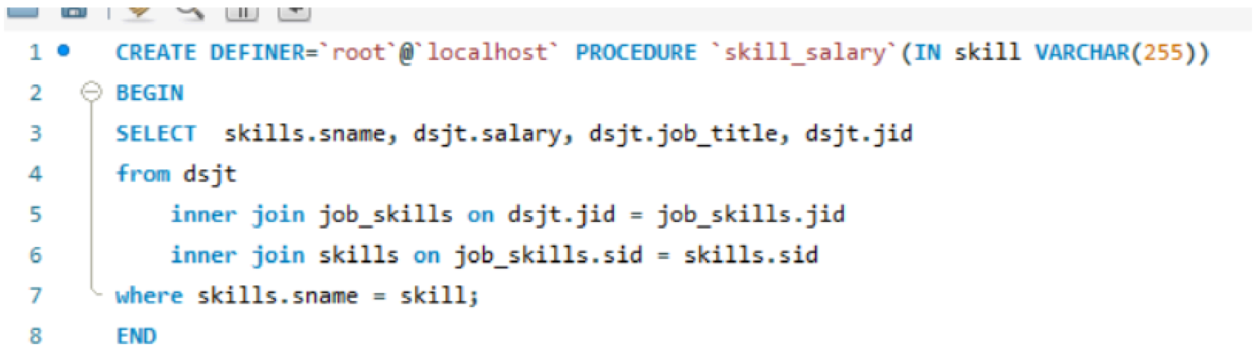
For the next part in my project I started cleaning the data. I looked for some common inconsistencies in the data in every column and removed them. For example in the revenue column there were monetary values instead of integer values. I then made a query to select the values with dollar signs and set them to null. I repeated this process on all the other columns. Some salary instances gave a range of salaries instead of a single salary. So I split the values into substrings and took out all of the non numerical characters. I then chose to keep the lowest values from the salary ranges for consistency. I did the same with the revenue column and finished by turning the data types of both columns from text to integer. Next I fixed the post date column so all of the values were integers representing the amount of days. This column was easy to fix because it just required removing non numeric characters from each value.



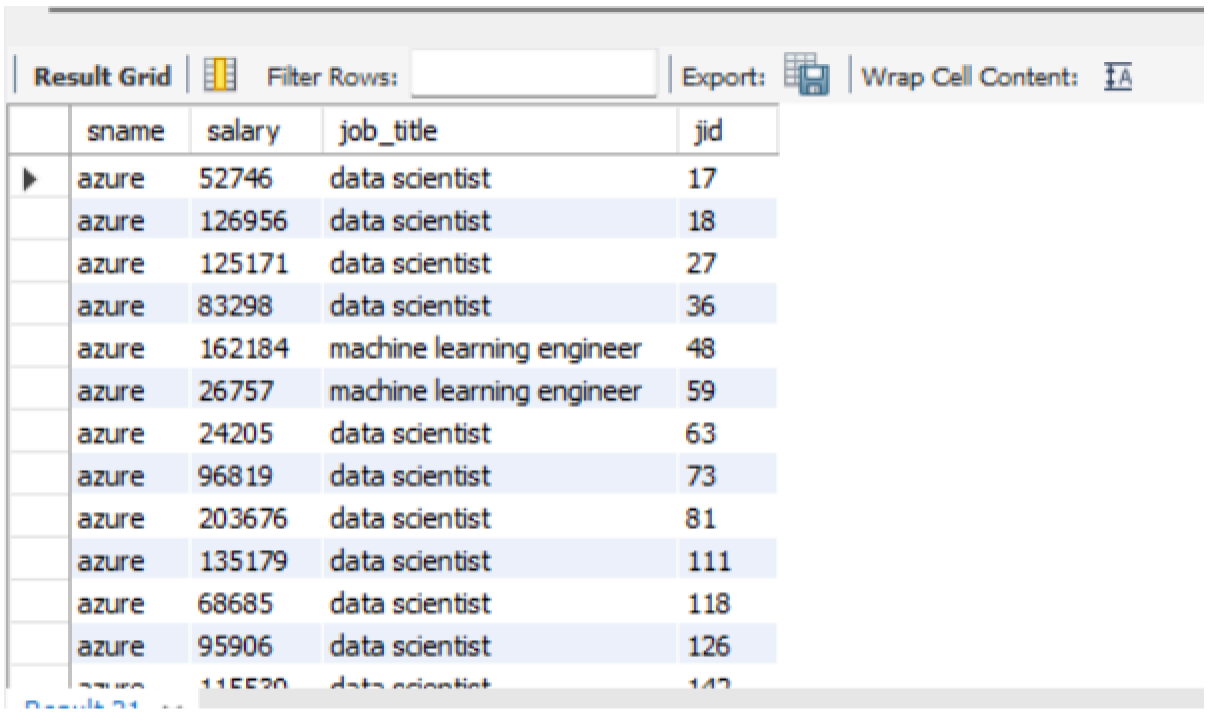
```
select *
from dsjt
;
```

jid	job_title	seniority_level	status	post_date	salary	cid	city	state	country
1	data scientist	senior	hybrid	17	100472	1	Grapevine	TX	USA
2	data scientist	lead	hybrid	15	118733	2	Fort Worth	TX	USA
3	data scientist	senior	on-site	30	94987	3	Austin	TX	USA
4	data scientist	senior	hybrid	8	112797	4	Chicago	IL	USA
5	data scientist	NULL	on-site	3	114172	5	Stamford	CT	USA
6	data scientist	lead	NULL	90	196371	6	New York	NY	USA
7	data scientist	junior	NULL	15	51330	7	Berkeley	CA	USA
8	machine learning engineer	senior	on-site	9	121480	8	Menlo Park	CA	USA
9	data scientist	senior	remote	6	207331	9	NULL	NULL	NULL
10	data scientist	senior	on-site	1	219201	10	San Francisco	CA	USA
11	data scientist	senior	NULL	15	96815	11	Atlanta	GA	USA
12	data scientist	midlevel	NULL	6	62697	12	Vancouver	NULL	Canada

Next I created stored procedures. The first procedure took in a skill as a parameter and returned all of the possible salaries of jobs that require the skill. I did this by joining the skills, job skills, and jobs table together with skill id and job id.



```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `skill_salary`(IN skill VARCHAR(255))
BEGIN
SELECT  skills.sname, dsjt.salary, dsjt.job_title, dsjt.jid
from dsjt
    inner join job_skills on dsjt.jid = job_skills.jid
    inner join skills on job_skills.sid = skills.sid
where skills.sname = skill;
END
```

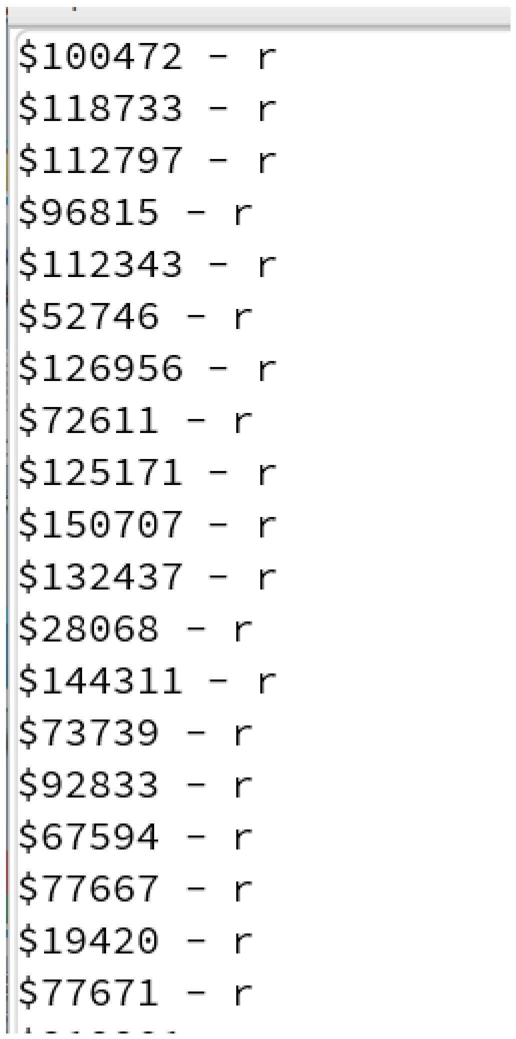


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sname	salary	job_title	jid
azure	52746	data scientist	17
azure	126956	data scientist	18
azure	125171	data scientist	27
azure	83298	data scientist	36
azure	162184	machine learning engineer	48
azure	26757	machine learning engineer	59
azure	24205	data scientist	63
azure	96819	data scientist	73
azure	203676	data scientist	81
azure	135179	data scientist	111
azure	68685	data scientist	118
azure	95906	data scientist	126

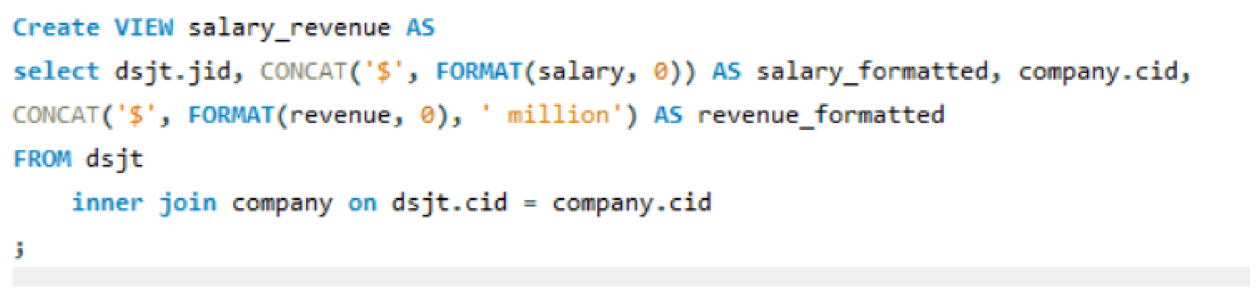
Another procedure I made takes a status as a parameter and returns all of the jobs that have that status. This procedure only required one table so it was pretty simple to make. The last stored procedure I made took in an industry as a parameter and returned jobs that are in that industry. This procedure required me to join the job and company tables together since the industry column in the job table. The stored procedures were not hard to make and I could definitely make more of them to retrieve more unique data.

I linked the skill-salary stored procedure to java. I did this by using jdbc to connect my java code to my database. This process was similar to assignment four where we used jdbc to connect java code to a database. I found this part a little difficult because I have not learned much java so I had to guess and check my code multiple times before I finally got it to work.

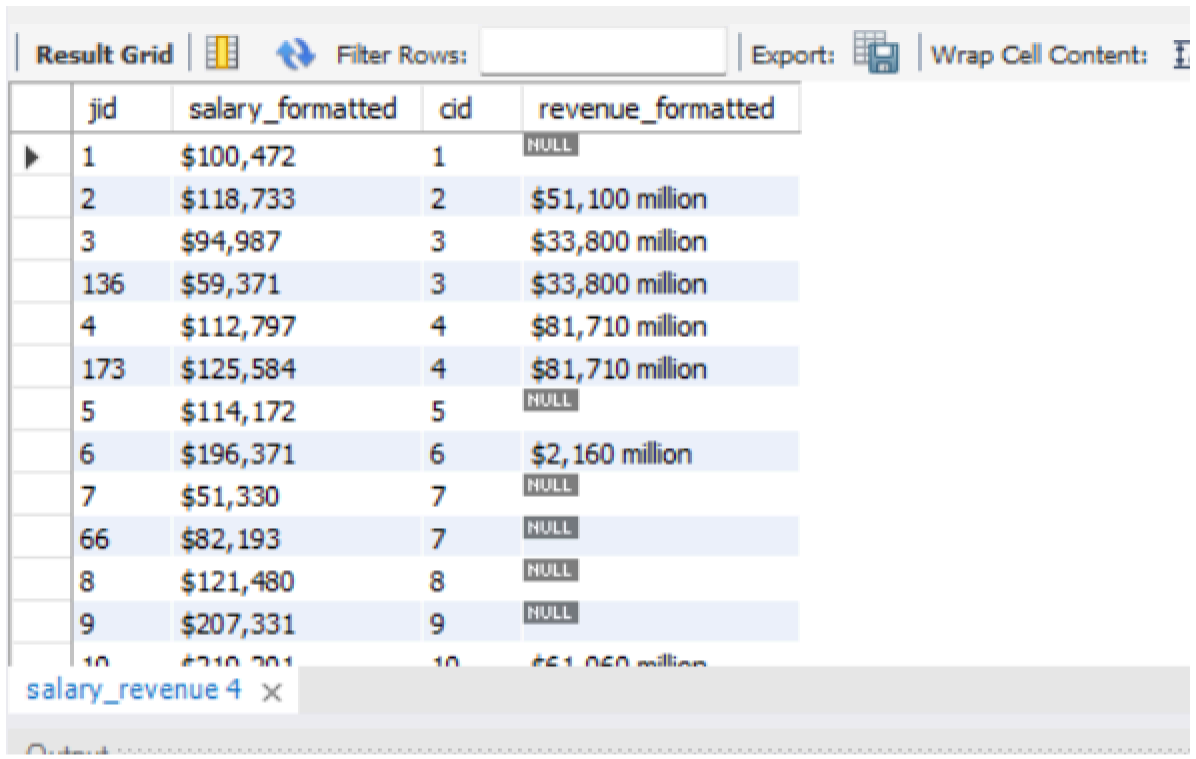


```
$100472 - r
$118733 - r
$112797 - r
$96815 - r
$112343 - r
$52746 - r
$126956 - r
$72611 - r
$125171 - r
$150707 - r
$132437 - r
$28068 - r
$144311 - r
$73739 - r
$92833 - r
$67594 - r
$77667 - r
$19420 - r
$77671 - r
```

My final addition was creating a view that shows the salary of each job and revenue of each company next to their respective id. I also formatted the values like, “$100,000” to make the value amount more obvious and easier to scan over. I made this view by joining the jobs and companies on company id so each job salary was next to its respective company.



```
Create VIEW salary_revenue AS
select dsjt.jid, CONCAT('$', FORMAT(salary, 0)) AS salary_formatted, company.cid,
CONCAT('$', FORMAT(revenue, 0), ' million') AS revenue_formatted
FROM dsjt
    inner join company on dsjt.cid = company.cid
;
```



jid	salary_formatted	cid	revenue_formatted
1	$100,472	1	NULL
2	$118,733	2	$51,100 million
3	$94,987	3	$33,800 million
136	$59,371	3	$33,800 million
4	$112,797	4	$81,710 million
173	$125,584	4	$81,710 million
5	$114,172	5	NULL
6	$196,371	6	$2,160 million
7	$51,330	7	NULL
66	$82,193	7	NULL
8	$121,480	8	NULL
9	$207,331	9	NULL
10	$210,201	10	$61,060 million

Overall this project was fun to make and was not as easy as I thought it would be. The initial parts of creating each table was easy along with the final parts of creating stored procedures and a view. The most difficult part for me was cleaning the data because it was not a topic that we covered in class. I was a little surprised how inconsistent the data was in the original table. If I had to do this project again I would definitely choose a data set with more consistent data which would make normalizing the data into multiple tables easier.

Works Cited

“Learn Mysql Fast, Easy and Fun 🐬.” *MySQL Tutorial*, 7 Jan. 2025, www.mysqltutorial.org/.

Nadeem, Aleesha. “Data Science Careers & Salaries 2025.” *Kaggle*, 2 Oct. 2025, www.kaggle.com/datasets/nalisha/data-science-careers-and-salaries-2025.