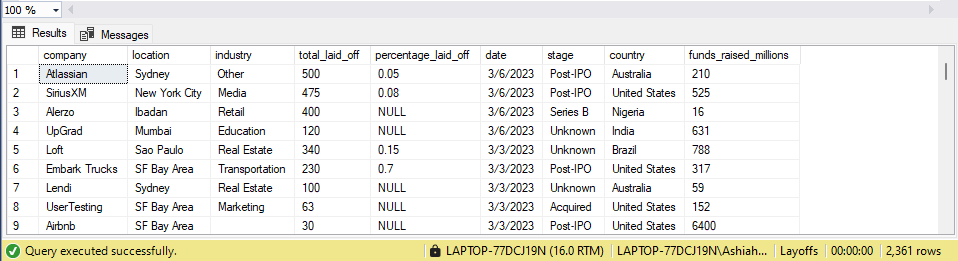
**SQL SERVER DATA CLEANING PROJECT**

This project will be using the layoffs dataset which contains data about employee layoffs from an array of companies and industries. The data has not been cleaned in anyway prior.

SELECT \* FROM layoffs



* Creating a staging dataset, a copy of our original raw dataset as a precaution in the event of making an error on the entire original dataset.

SELECT \* INTO layoffs\_staging

FROM layoffs ;

SELECT \* FROM layoffs\_staging;

* There isn't a primary key in the dataset so we are going to create one so we can identify duplicates

WITH duplicate\_CTE AS

(

SELECT \*, ROW\_NUMBER() OVER(

PARTITION BY company, location, industry, total\_laid\_off,percentage\_laid\_off, date, stage, country, funds\_raised\_millions ORDER BY company)

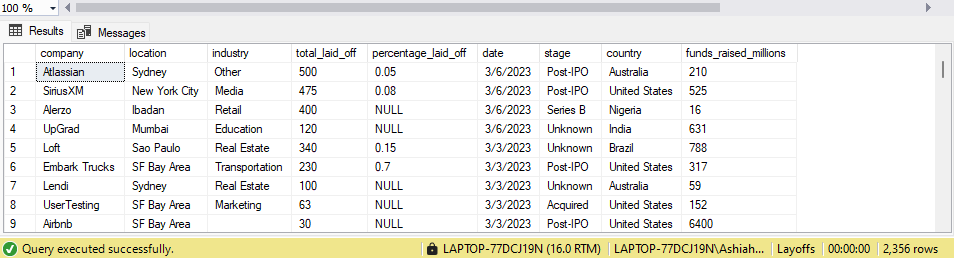
AS row\_num FROM layoffs\_staging

)

DELETE

FROM duplicate\_CTE

WHERE row\_num > 1;



* Now that we have identified and deleted the duplicates, we will now standardize the data.

SELECT company, TRIM(company)

FROM layoffs\_staging;

* The Trim function removes all unwanted characters (usually spaces) from the front and end of strings (In this case we're just doing it for the front.)

UPDATE layoffs\_staging

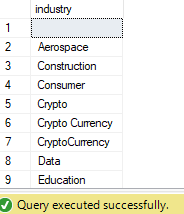
SET company= TRIM(company);

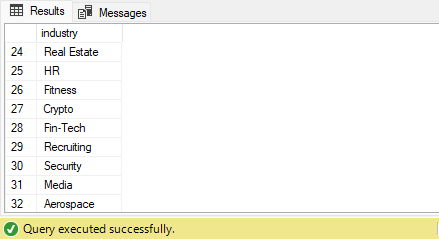
SELECT DISTINCT(industry)

FROM layoffs\_staging

ORDER BY 1;

* We notice that Crypto, Crypto Currency and CryptoCurrency are being profiled as three unique industries and we want to update them to have the same industry name.





SELECT \*

FROM layoffs\_staging

WHERE industry LIKE 'Crypto%';

UPDATE layoffs\_staging

SET industry = 'Crypto'

WHERE industry LIKE 'Crypto%'

SELECT DISTINCT industry

FROM layoffs\_staging

SELECT DISTINCT (location)

FROM Layoffs\_staging

ORDER BY 1;

SELECT DISTINCT (country)

FROM Layoffs\_staging

ORDER BY 1;

* We notice there's two entries for 'United states'; United States and United States.

SELECT DISTINCT (country), TRIM(TRAILING '.' FROM country)

FROM Layoffs\_staging

ORDER BY 1;

UPDATE Layoffs\_staging

SET country = TRIM(TRAILING '.' FROM country)

WHERE country LIKE 'United States%';

We utilize the **sp\_help** function to find out information about the structure of the layoffs\_staging table. The Date column has a string datatype and we want our data column to have a date datatype. We use the **sp\_rename** function to change the name of the column to avoid it being the name of a reserved keyword and also give a more descriptive name. We then create a new column called 'DATE' and assign it the date datatype. We use the update command to then copy all the records from the EventDate column into the DATE column. Lastly we remove the EventDate column from the table.

EXEC sp\_help layoffs\_staging;

EXEC sp\_rename 'layoffs\_staging.[date]', 'EventDate','COLUMN';

ALTER TABLE layoffs\_staging ADD DateColumn DATE;

UPDATE layoffs\_staging

SET DateColumn = TRY\_CONVERT(DATE, [EventDate], 101)

WHERE TRY\_CONVERT(DATE, [EventDate], 101) IS NOT NULL;

ALTER TABLE Layoffs\_staging DROP COLUMN EventDate

--Dealing with Null or blank values

SELECT \*

FROM layoffs\_staging

WHERE total\_laid\_off ='NULL';

SELECT \*

FROM layoffs\_staging

WHERE industry ='NULL'

OR industry = ' ';

SELECT \*

FROM layoffs\_staging t1

JOIN layoffs\_staging t2

ON t1.company = t2.company

WHERE (t1.industry IS NULL OR LTRIM(RTRIM( t1.industry)) = '')

AND t2.industry IS NOT NULL

UPDATE layoffs\_staging

SET industry = NULL

WHERE industry = ' ';

UPDATE t1

SET t1.industry = t2.industry

FROM layoffs\_staging t1

INNER JOIN layoffs\_staging t2

ON t1.company = t2.company

WHERE t1.industry IS NULL

AND t2.industry IS NOT NULL;

Lastly we deleted these rows as they cannot provide us with layoff or layoff percentage data.

DELETE

FROM layoffs\_staging

WHERE total\_laid\_off IS NULL

AND percentage\_laid\_off IS NULL ;

That ends our datacleaning process in SQL.