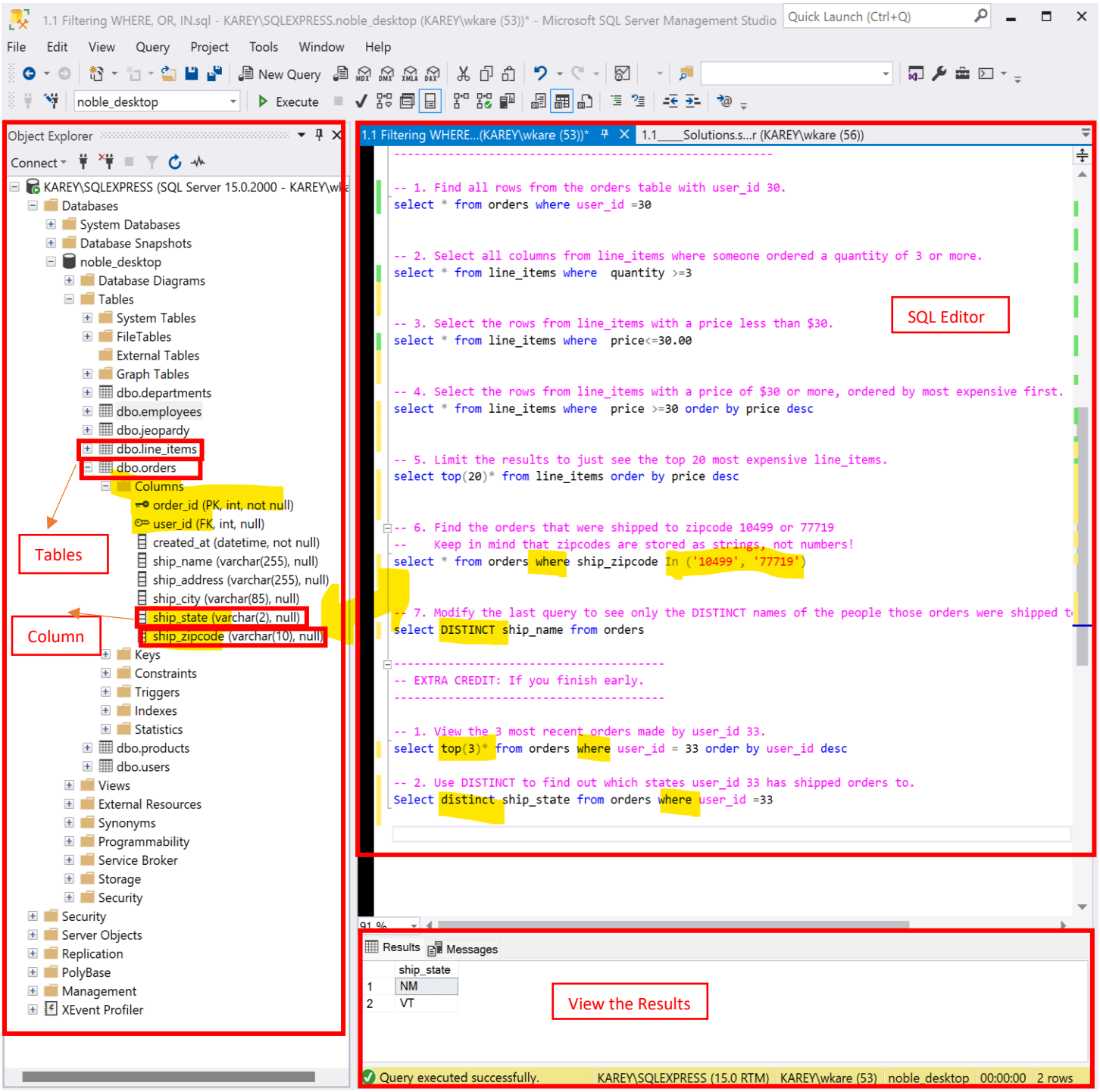
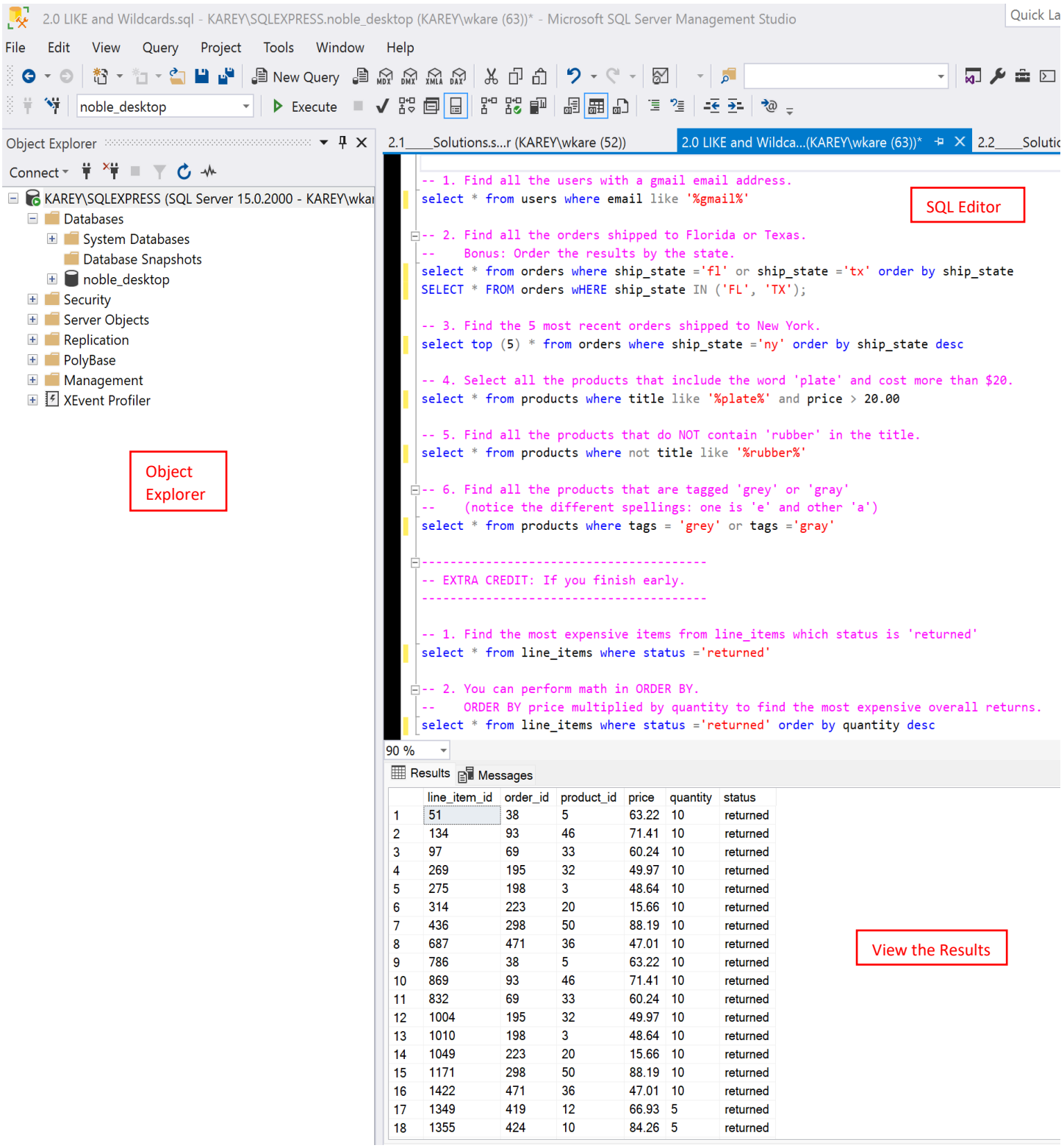
Where/ Distinct

SELECT * → DISTINCT → Top ( )* → From → Join → Where → Group By → Having → Order By



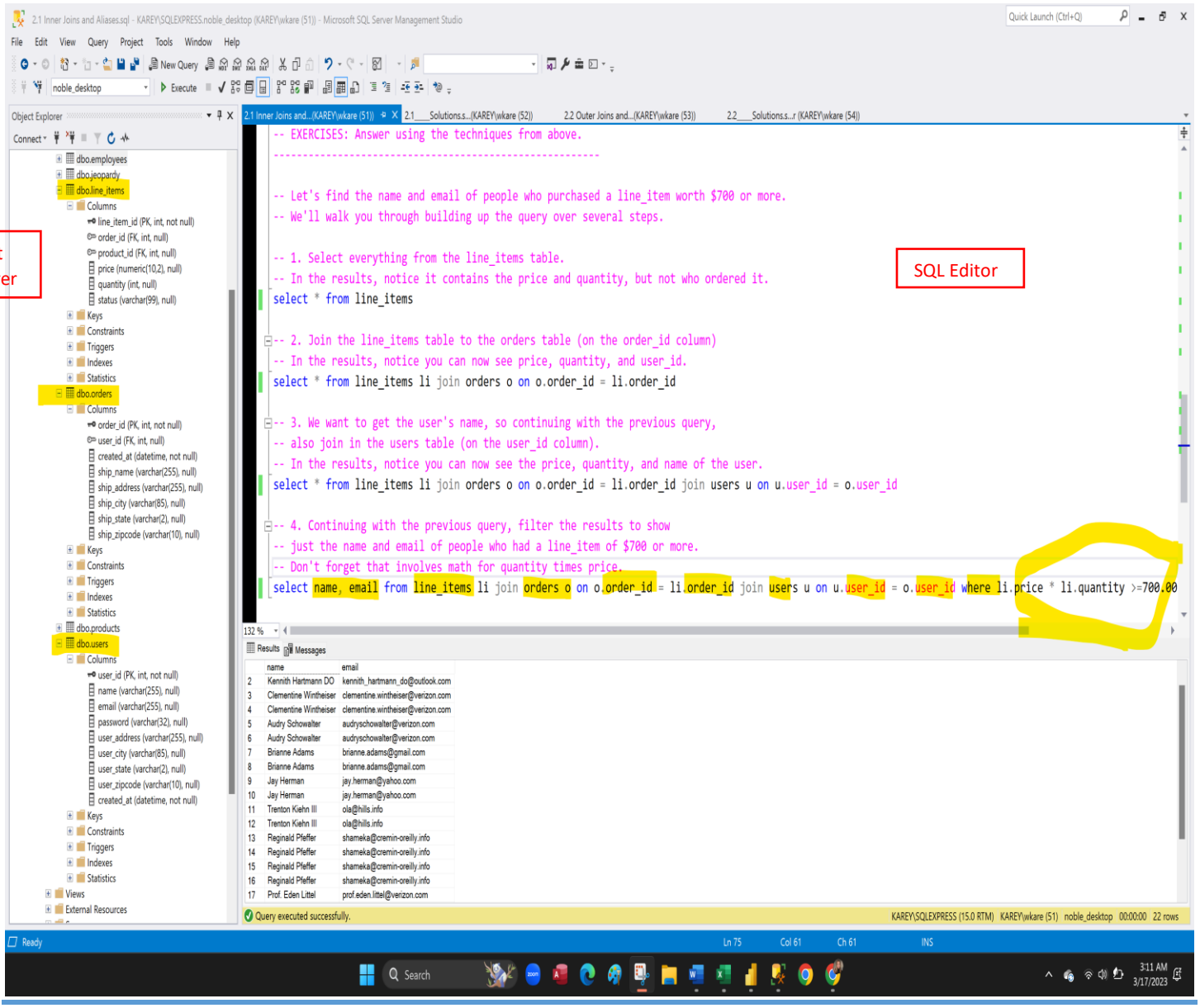
Wildcards/ Like

SELECT * → DISTINCT → Top ( )* → From → Join → Where → Group By → Having → Order By



Inner Join/ Alias

SELECT * → DISTINCT → Top ( )* → From → Join → Where → Group By → Having → Order By



Outer Joins/ Nulls

SELECT * → DISTINCT → Top ( )* → From → Join → Where → Group By → Having → Order By

FYIs

1. SQL Server case is insensitive

SQL Query Written Order

SQL	Purpose
SELECT	Specify the columns to show in result set
DISTINCT	Eliminate duplicate rows
TOP (SQL Server)	Limit the returned data to a specific number of rows
FROM	Get the base data from a table
JOIN	Obtain matching data from other table(s)
WHERE	Filter the base data
GROUP BY	Aggregate the base data (collect into groups)
HAVING	Filter the aggregated (grouped) data
ORDER BY	Sort the final data
LIMIT (Postgres)	Limit the returned data to a specific number of rows

2. Logical operators are used after a "WHERE" clause

Operator	Description
AND	Requires both specified conditions are met (true) for a record to be included in the result.
OR	Requires at least one of the specified conditions are met (true) for the record to be included in the result
NOT	Selects rows for the result which do **not** meet the specified criteria.

Instead of using "OR" clause, we can use "IN" clause

```
SELECT * FROM orders
WHERE ship_state = 'FL' OR ship_state = 'TX';
WHERE ship_state IN ('FL', 'TX');
```

```
SELECT * FROM products WHERE NOT title LIKE '%rubber%';
```

3. Alias- "AS" clause is optional when it comes to abbreviating table or column

Syntax for Aliases

COLUMN ALIASES:

- `emp_name AS Name`
- `emp_name Name`
- `emp_name AS "Employee Name"`
- `emp_name "Employee Name"`

TABLE ALIASES:

- `employees AS e`
- `employees e`

4. A foreign key in one table, refers to a primary key in another table. However, 1 table can only have 1 primary key (aka unique identifier) at the same time. When multiple fields are used as a primary key, it is called a composite key.

primary key

foreign key

emp_id	emp_name	dept_id	manager_id
E1	Amy Schumm	D1	E5
E2	Riley Heaney	D3	E5
E3	Tyler Monahan	D2	E6
E4	Chad Schmidt	NULL	E1
E5	Elma Stanton	D1	E6
E6	Bruce Rice	NULL	NULL

primary key

dept_id	dept_name
D1	Sales
D2	HR
D3	IT
D4	Support
D5	Administration

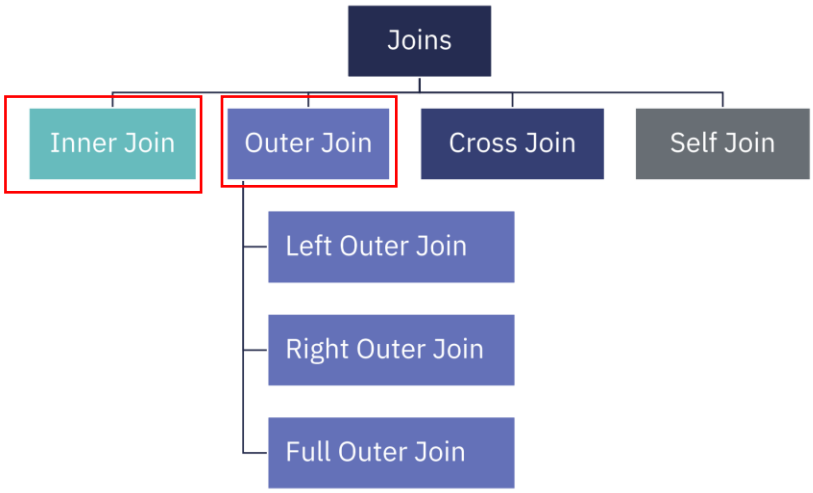
Inner Join

```
SELECT * FROM employees
JOIN departments
ON employees.dept_id = departments.dept_id;
```

Combine the tables, leaving out rows with NULL values

emp_id	emp_name	dept_id	manager_id	dept_id	dept_name
E1	Amy Schumm	D1	E5	D1	Sales
E2	Riley Heaney	D3	E5	D3	IT
E3	Tyler Monahan	D2	E6	D2	HR
E5	Elma Stanton	D1	E6	D1	Sales

5. A join combines data from multiple tables with or without using primary/foreign keys.

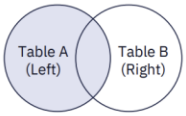


INNER JOIN: returns matching rows through the primary key (aka unique identifier)

OUTER JOIN: returns matching and non-matching rows (missing values appear as NULL).

The **Left, Right, Outer Joins** are all the same. They simply change the side where we keep unmatched rows. See below.

Outer Left Join



primary key → foreign key → primary key

emp_id	emp_name	dept_id	manager_id
E1	Amy Schumm	D1	E5
E2	Riley Heaney	D3	E5
E3	Tyler Monahan	D2	E6
E4	Chad Schmidt	NULL	E1
E5	Elma Stanton	D1	E6
E6	Bruce Rice	NULL	NULL

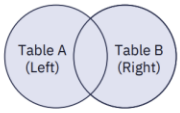
dept_id	dept_name
D1	Sales
D2	HR
D3	IT
D4	Support
D5	Administration

emp_id	emp_name	dept_id	manager_id	dept_id	dept_name
E1	Amy Schumm	D1	E5	D1	Sales
E2	Riley Heaney	D3	E5	D3	IT
E3	Tyler Monahan	D2	E6	D2	HR
E4	Chad Schmidt	NULL	E1	NULL	NULL
E5	Elma Stanton	D1	E6	D1	Sales
E6	Bruce Rice	NULL	NULL	NULL	NULL

```
SELECT * FROM employees e
LEFT JOIN departments d
ON e.dept_id = d.dept_id;
```

Combine the tables, keeping all rows from the left (first) table.

Full Outer Join



emp_id	emp_name	dept_id	manager_id
E1	Amy Schumm	D1	E5
E2	Riley Heaney	D3	E5
E3	Tyler Monahan	D2	E6
E4	Chad Schmidt	NULL	E1
E5	Elma Stanton	D1	E6
E6	Bruce Rice	NULL	NULL

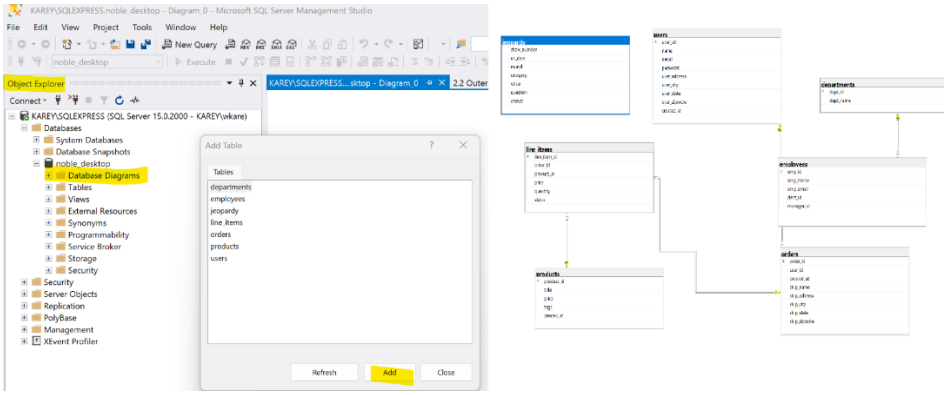
dept_id	dept_name
D1	Sales
D2	HR
D3	IT
D4	Support
D5	Administration

emp_id	emp_name	dept_id	manager_id	dept_id	dept_name
E1	Amy Schumm	D1	E5	D1	Sales
E2	Riley Heaney	D3	E5	D3	IT
E3	Tyler Monahan	D2	E6	D2	HR
E4	Chad Schmidt	NULL	E1	NULL	NULL
E5	Elma Stanton	D1	E6	D1	Sales
E6	Bruce Rice	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	D5	Administration
NULL	NULL	NULL	NULL	D4	Support

```
SELECT * FROM employees e
FULL JOIN departments d
ON e.dept_id = d.dept_id;
```

Combine the tables, keeping all rows from both tables.

6. Object Explorer→ Right Click Database→Create Database Diagram→Yes→Add->Close



Schema

- A database schema is a set of tables.
- The schema defines how data is organized, the relations among tables.