RDB & SQL

Session 8

Set Operators

I've completed the **pre-class** content?

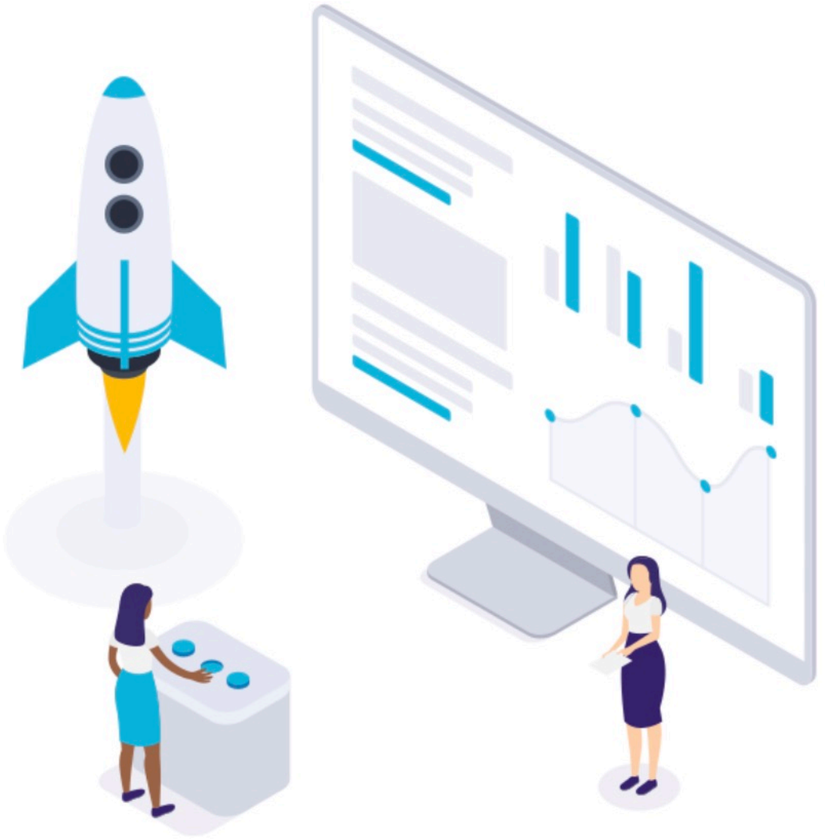
Students choose an option

Pear Deck Interactive Slide
Do not remove this bar

No Multiple Choice Response
You didn't answer this question

Table of Contents

1 Introduction



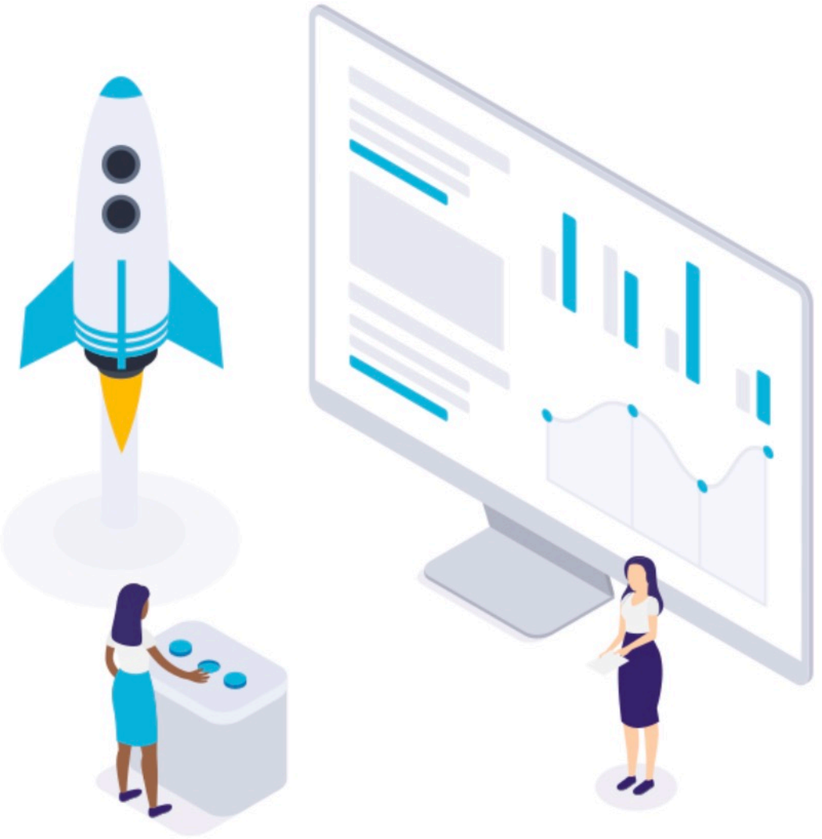
Introduction

- ★ Set operations allow the results of multiple queries to be combined into a single result set.
- ★ Set operators include UNION, UNION ALL, INTERSECT, and EXCEPT for MS SQL Server.

Important!

- Both SELECT statements must contain the **same number of columns**.
- In the SELECT statements, the corresponding columns must have the **same data type**.
- Positional ordering must be used to sort the result set. The individual result set ordering is not allowed with Set operators. **ORDER BY can appear once at the end of the query.**
- **UNION and INTERSECT** operators are commutative, i.e. **the order of queries is not important**; it doesn't change the final result.
- **Performance-wise, UNION ALL** shows better performance as compared to **UNION** because resources are **not wasted in filtering duplicates and sorting** the result set.
- Set operators can be the **part of subqueries.**

2 UNION



Introduction

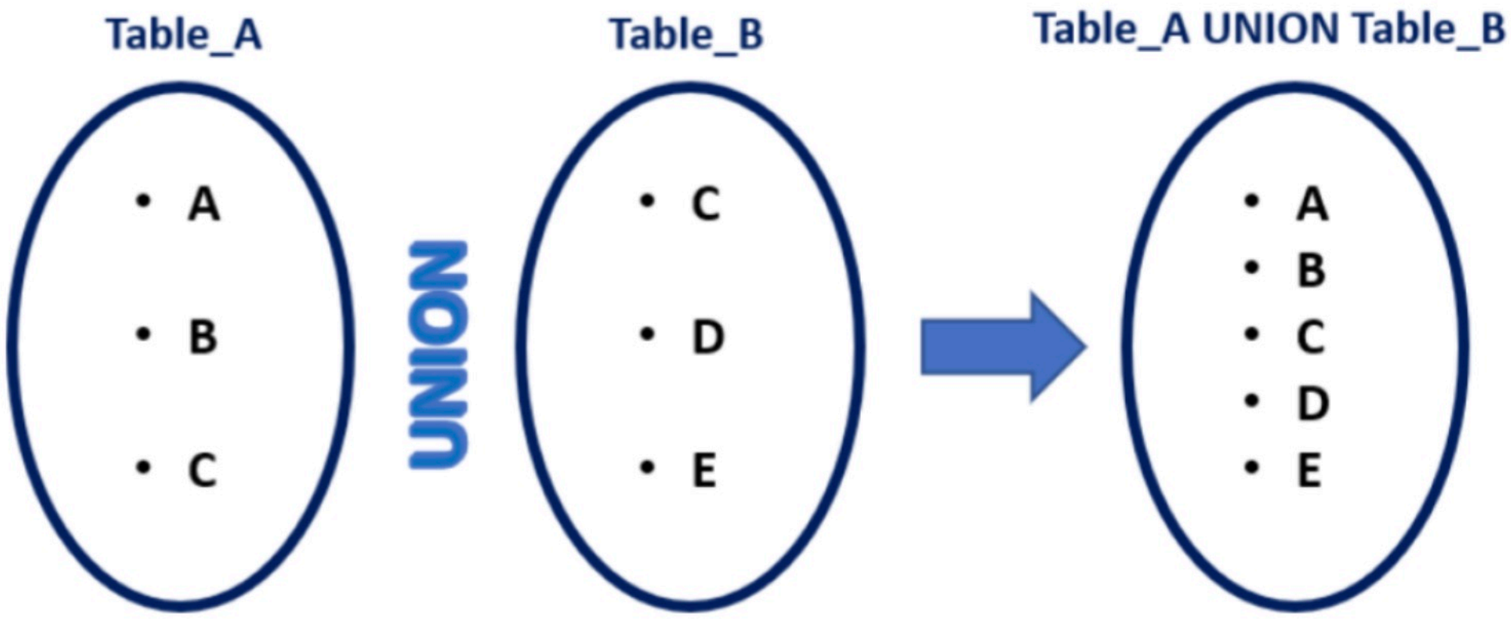
In some cases, you may need to **combine data from two or more tables** into a result set. **Union** clause is used to perform this operation.

The tables that you need to combine can be tables with similar data in the same database, or in different databases.

Syntax

```
SELECT column1, column2, ...
  FROM table_A
UNION
SELECT column1, column2, ...
  FROM table_B
```

UNION



Sample Tables

"employees_A" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
17679       Robert      Gilmore     110000      Operations Director  Male
26650       Elvis       Ritter      86000       Sales Manager        Male
30840       David       Barrow      85000       Data Scientist       Male
49714       Hugo        Forester    55000       IT Support Speciali  Male
51821       Linda       Foster      95000       Data Scientist       Female
67323       Lisa        Wiener      75000       Business Analyst     Female
```

"employees_B" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
49714       Hugo        Forester    55000       IT Support Specialist  Male
67323       Lisa        Wiener      75000       Business Analyst       Female
70950       Rodney      Weaver      87000       Project Manager        Male
71329       Gayle       Meyer       77000       HR Manager             Female
76589       Jason       Christian   99000       Project Manager        Male
97927       Billie      Lanning     67000       Web Developer          Female
```

Example

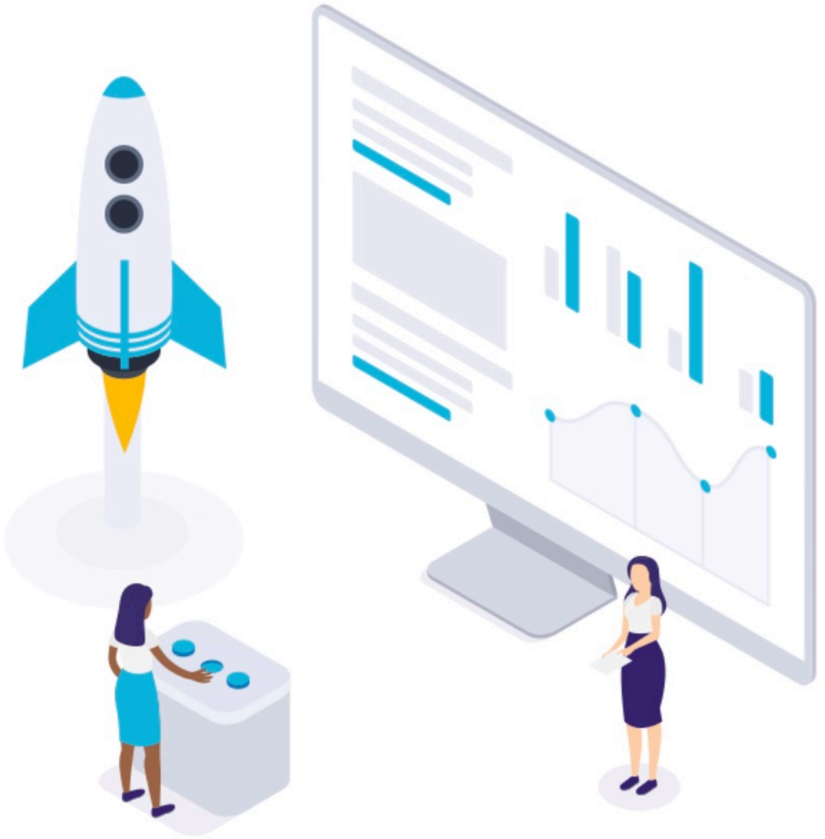
query :

```
1  SELECT emp_id, first_name, last_name, job_title
2    FROM employees_A
3  UNION
4  SELECT emp_id, first_name, last_name, job_title
5    FROM employees_B;
```

output:

```
1  emp_id      first_name  last_name   job_title
2  ----------  ----------  ----------  ------------------
3  17679       Robert      Gilmore     Operations Director
4  26650       Elvis       Ritter      Sales Manager
5  30840       David       Barrow      Data Scientist
6  49714       Hugo        Forester    IT Support Speciali
7  51821       Linda       Foster      Data Scientist
8  67323       Lisa        Wiener      Business Analyst
9  70950       Rodney      Weaver      Project Manager
10 71329       Gayle       Meyer       HR Manager
11 76589       Jason       Christian   Project Manager
12 97927       Billie      Lanning     Web Developer
```

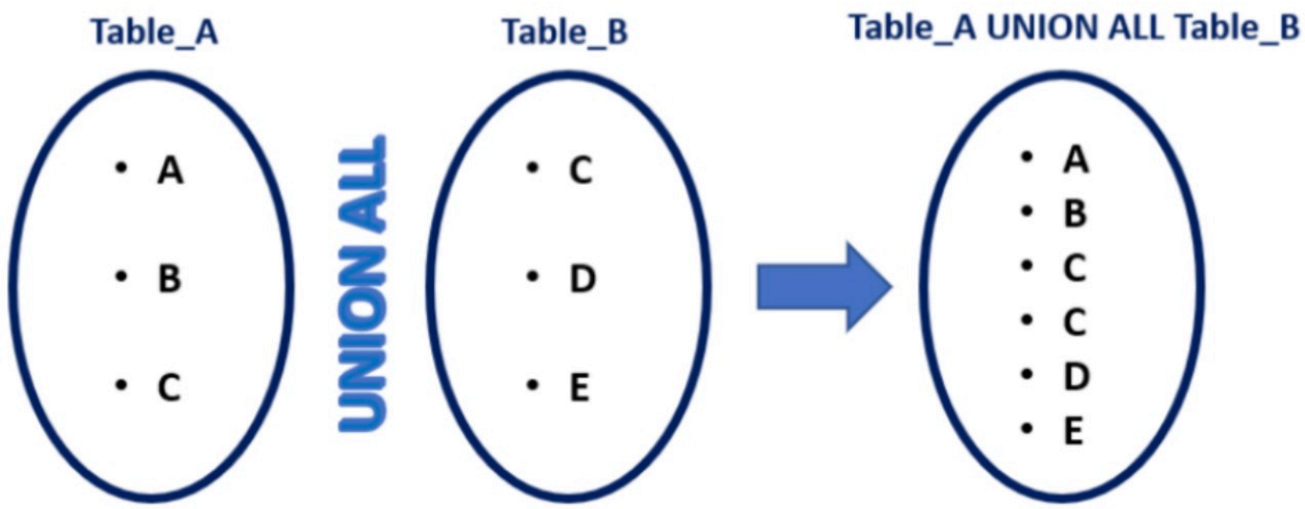
3 UNION ALL



Introduction

Syntax

```sql
SELECT column1, column2, ...
  FROM table_A
UNION ALL
SELECT column1, column2, ...
  FROM table_B
```



Sample Tables

"employees_A" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
17679       Robert      Gilmore     110000      Operations Director  Male
26650       Elvis       Ritter      86000       Sales Manager        Male
30840       David       Barrow      85000       Data Scientist       Male
49714       Hugo        Forester    55000       IT Support Speciali  Male
51821       Linda       Foster      95000       Data Scientist       Female
67323       Lisa        Wiener      75000       Business Analyst     Female
```

"employees_B" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
49714       Hugo        Forester    55000       IT Support Specialist  Male
67323       Lisa        Wiener      75000       Business Analyst       Female
70950       Rodney      Weaver      87000       Project Manager        Male
71329       Gayle       Meyer       77000       HR Manager             Female
76589       Jason       Christian   99000       Project Manager        Male
97927       Billie      Lanning     67000       Web Developer          Female
```

Example

query :

```sql
SELECT 'Employees A' AS Type, emp_id, first_name, last_name, job_title
  FROM employees_A
UNION ALL
SELECT 'Employees B' AS Type, emp_id, first_name, last_name, job_title
  FROM employees_B;
```

output:

```
Type         emp_id      first_name  last_name   job_title
-----------  ----------  ----------  ----------  -------------------
Employees A  17679       Robert      Gilmore     Operations Director
Employees A  26650       Elvis       Ritter      Sales Manager
Employees A  30840       David       Barrow      Data Scientist
Employees A  49714       Hugo        Forester    IT Support Speciali
Employees A  51821       Linda       Foster      Data Scientist
Employees A  67323       Lisa        Wiener      Business Analyst
Employees B  49714       Hugo        Forester    IT Support Speciali
Employees B  67323       Lisa        Wiener      Business Analyst
Employees B  70950       Rodney      Weaver      Project Manager
Employees B  71329       Gayle       Meyer       HR Manager
Employees B  76589       Jason       Christian   Project Manager
Employees B  97927       Billie      Lanning     Web Developer
```

Query Time

Question: List the products sold in the cities of Charlotte and Aurora

Query Time For You

Question: Write a query that returns all customers whose first or last name is Thomas. (don't use 'OR')

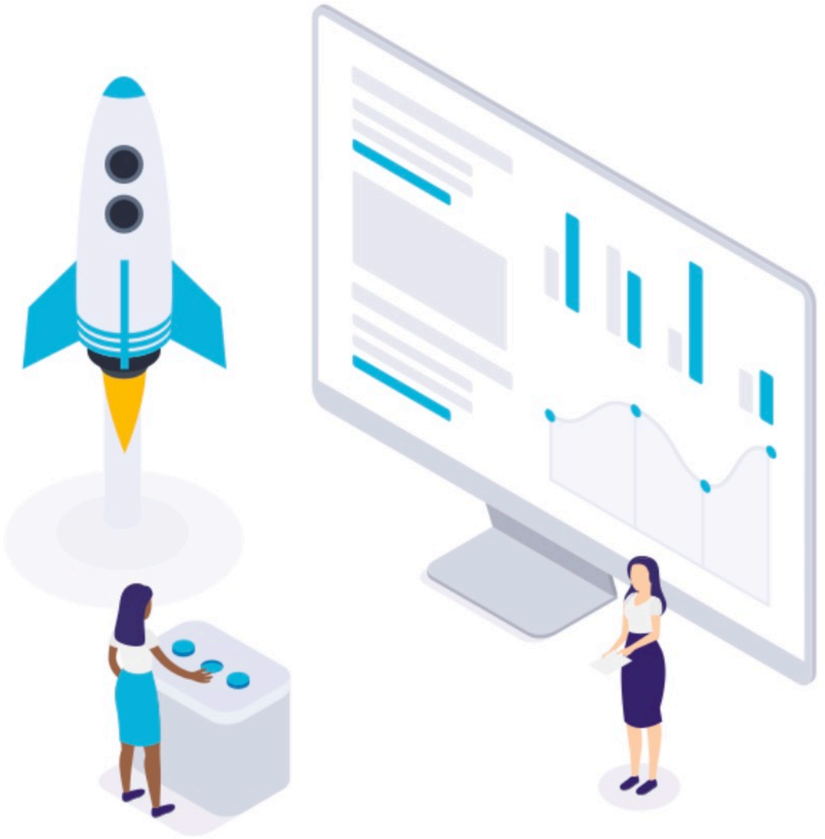
Expected Output:

Results | Messages

	first_name	last_name
1	Thomas	Rogers
2	Thomas	Chan
3	Thomas	Jefferson
4	Thomas	Moore
5	Thomas	Erickson
6	Thomas	Newman
7	Thomas	Mcdaniel
8	Thomas	Davis
9	Thomas	Webb
10	Thomas	Little
11	James	Thomas
12	Lidia	Thomas
13	Hildegar...	Thomas

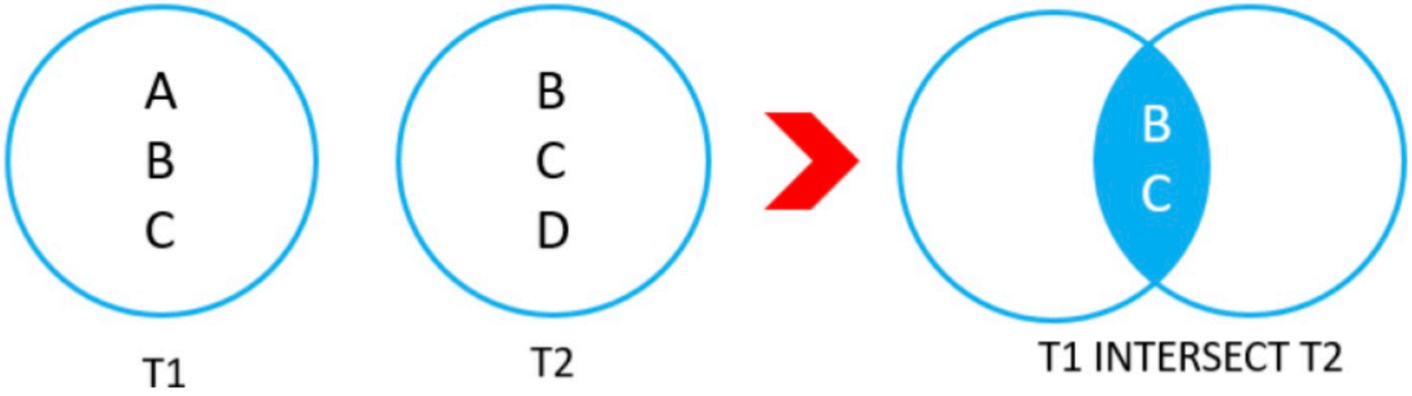
KTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 37 rows

4 Intersect



Introduction

INTERSECT operator compares the result sets of two queries and returns distinct rows that are output by both queries.



Syntax

```
SELECT column1, column2, ...
  FROM table_A
INTERSECT
SELECT column1, column2, ...
  FROM table_B
```

Sample Tables

"employees_A" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
17679       Robert      Gilmore     110000      Operations Director  Male
26650       Elvis       Ritter      86000       Sales Manager        Male
30840       David       Barrow      85000       Data Scientist       Male
49714       Hugo        Forester    55000       IT Support Speciali  Male
51821       Linda       Foster      95000       Data Scientist       Female
67323       Lisa        Wiener      75000       Business Analyst     Female
```

"employees_B" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
49714       Hugo        Forester    55000       IT Support Specialist  Male
67323       Lisa        Wiener      75000       Business Analyst       Female
70950       Rodney      Weaver      87000       Project Manager        Male
71329       Gayle       Meyer       77000       HR Manager             Female
76589       Jason       Christian   99000       Project Manager        Male
97927       Billie      Lanning     67000       Web Developer          Female
```

Example

query :

```
1  SELECT emp_id, first_name, last_name, job_title
2    FROM employees_A
3  INTERSECT
4  SELECT emp_id, first_name, last_name, job_title
5    FROM employees_B
6    ORDER BY emp_id;
```

output:

```
1  emp_id      first_name  last_name   job_title
2  ----------  ----------  ----------  --------------------
3  49714       Hugo        Forester    IT Support Specialist
4  67323       Lisa        Wiener      Business Analyst
```

Query Time

Question: Write a query that returns all brands with products for both 2018 and 2020 model year.

Expected Output:

Results | Messages

	brand_id	brand_name
1	1	Samsung
2	2	Apple
3	3	Sony
4	4	Logitech
5	5	Yamaha
6	6	Corsair
7	7	Jbl
8	8	Netgear
9	9	Sandisk
10	10	Asus
11	11	Razer
12	12	wd

)ESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 28 rows

Query Time For You

Question: Write a query that returns the first and the last names of the customers who placed orders in all of 2018, 2019, and 2020.

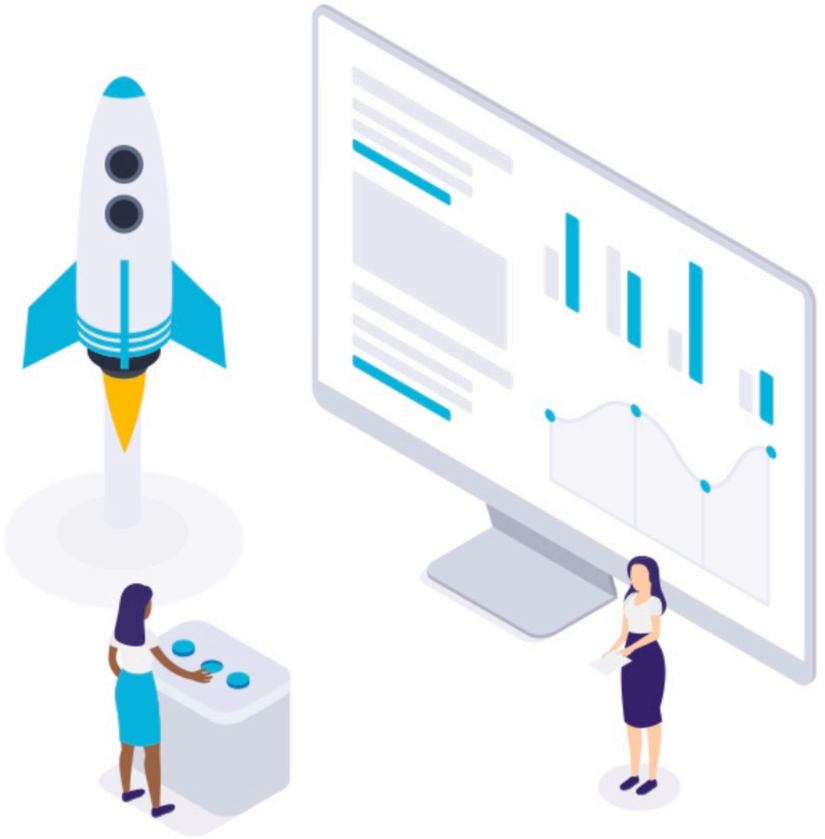
Expected Output:

Results | Messages

	first_name	last_name
1	Theresia	Schoeneck
2	Tomika	Bond
3	Daren	Rollins
4	Petronila	Higgins
5	Gilbert	Brown
6	Tajuana	Byers
7	Willow	Rulapaugh
8	Joshua	Mays
9	Joeann	Centini
10	Ronald	Gilliam
11	Jerald	Berray
12	Elizabeth	Levy

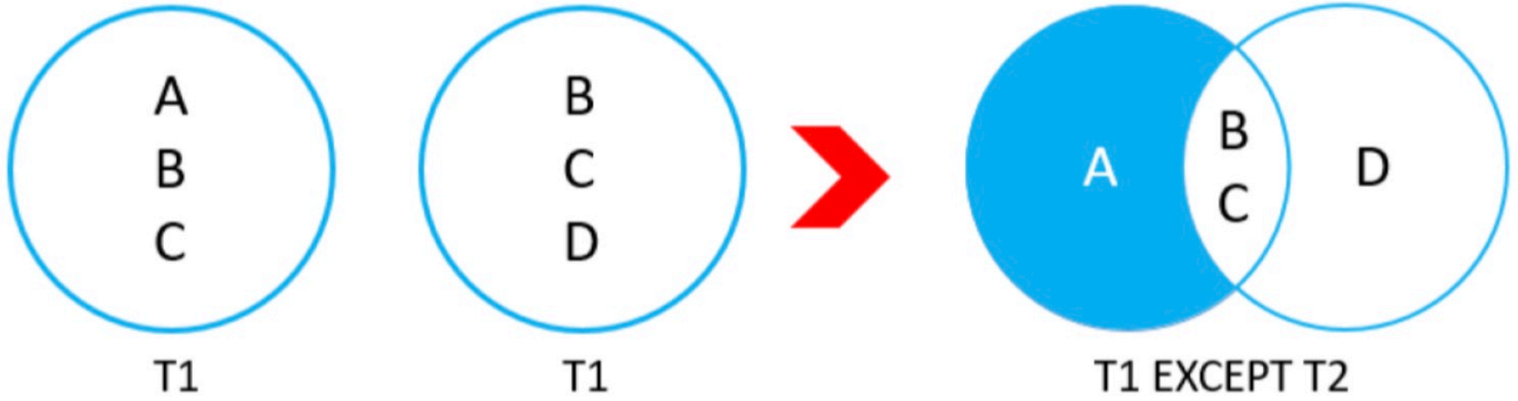
(TOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 12 rows

5 Except



Introduction

EXCEPT operator compares the result sets of the two queries and returns the rows of the previous query that differ from the next query.



Syntax

```
SELECT column1, column2, ...
  FROM table_A
EXCEPT
SELECT column1, column2, ...
  FROM table_B
```

Sample Tables

"employees_A" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
17679       Robert      Gilmore     110000      Operations Director  Male
26650       Elvis       Ritter      86000       Sales Manager        Male
30840       David       Barrow      85000       Data Scientist       Male
49714       Hugo        Forester    55000       IT Support Speciali  Male
51821       Linda       Foster      95000       Data Scientist       Female
67323       Lisa        Wiener      75000       Business Analyst     Female
```

"employees_B" table:

```
emp_id      first_name  last_name   salary      job_title            gender
----------  ----------  ----------  ----------  -------------------  ----------
49714       Hugo        Forester    55000       IT Support Specialist  Male
67323       Lisa        Wiener      75000       Business Analyst       Female
70950       Rodney      Weaver      87000       Project Manager        Male
71329       Gayle       Meyer       77000       HR Manager             Female
76589       Jason       Christian   99000       Project Manager        Male
97927       Billie      Lanning     67000       Web Developer          Female
```

Example

query :

```
1  SELECT emp_id, first_name, last_name, job_title
2    FROM employees_A
3  EXCEPT
4  SELECT emp_id, first_name, last_name, job_title
5    FROM employees_B;
```

output:

```
1  emp_id      first_name  last_name   job_title
2  ----------  ----------  ----------  -------------------
3  17679       Robert      Gilmore     Operations Director
4  26650       Elvis       Ritter      Sales Manager
5  30840       David       Barrow      Data Scientist
6  51821       Linda       Foster      Data Scientist
7
```

Query Time

Question: Write a query that returns the brands have 2018 model products but not 2019 model products.

Expected Output:

Results | Messages

	brand_id	brand_name
1	23	DENAQ
2	37	Western Digital

DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 2 rows

Query Time For You

Question: Write a query that contains only products ordered in 2019 (Result not include products ordered in other years)

Expected Output:

Results | Messages

	product_id	product_name
1	30	Sony Mini Digital Video Cassettes - DVC - 1 Hour
2	73	Logitech Focus Case with Integrated Keyboard for iPad ...
3	94	128GB iPod touch (Space Gray) (6th Generation)
4	83	Logitech Circle Black Portable Wifi Video Monitoring Ca...
5	74	32GB High Speed UHS-I SDHC U3 Memory Card (Class...

Query executed successfully. | (local) (15.0 RTM) | DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 5 rows

CASE Expression

Table of Contents

1 Introduction

Introduction

The **CASE** expression evaluates a list of conditions and returns a value when the first condition is met.

The CASE expression is similar to the **IF-ELSE** statement in other programming languages.

There are two kinds of **CASE** expression:

- **Simple** CASE
- **Searched** CASE

2 Simple CASE Expression

Simple CASE Expression

Syntax

```
CASE case_expression
  WHEN when_expression_1 THEN result_expression_1
  WHEN when_expression_1 THEN result_expression_1
  ...
  [ ELSE else_result_expression ]
END
```

Simple CASE Expression

department table

	id	name	dept_name	salary
1	10238	Eric	Economics	72000
2	13378	Karl	Music	42000
3	23493	Jason	Philosophy	45000
4	26299	Jane	Computer Science	91000
5	30766	Jack	Economics	68000
6	40284	Mary	Psychology	78000
7	43087	Brian	Physics	93000
8	53695	Richard	Philosophy	54000
9	58248	Joseph	Political Science	58000
10	63172	David	Art History	65000
11	64378	Elvis	Physics	87000
12	96945	John	Computer Science	80000
13	99231	Santosh	Computer Science	74000

query:

```
1  SELECT dept_name,
2      CASE dept_name
3          WHEN 'Computer Science' THEN 'IT'
4          ELSE 'others'
5      END AS category
6  FROM department;
7
```

output:

```
1  dept_name   category
2  ----------  ----------
3  Economics   others
4  Music       others
5  Philosophy  others
6  Computer S  IT
7  Economics   others
8  Psychology  others
9  Physics     others
10 Philosophy  others
11 Political   others
12 Art Histor  others
13 Physics     others
14 Computer S  IT
15 Computer S  IT
16
```

Query Time

Question: Create a new column with the meaning of the values in the Order_Status column.

1 = Pending; 2 = Processing; 3 = Rejected; 4 = Completed

Expected Output:

Results | Messages

	order_id	order_status	order_status_desc
1	1	4	Completed
2	2	4	Completed
3	3	4	Completed
4	4	4	Completed
5	5	4	Completed
6	6	4	Completed
7	7	4	Completed
8	8	4	Completed
9	9	4	Completed
10	10	4	Completed

Q... | (local) (15.0 RTM) | DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 1.615 rows

Query Time For You

Question: Create a new column with the names of the stores to be consistent with the values in the store_ids column

1 = Davi techno Retail; 2 = The BFLO Store; 3 = Burkes Outlet

Expected Output:

Results Messages

	first_name	last_name	store_id	Store_name
1	James	Garcia	1	Davi techno Retail
2	Charles	Cussona	1	Davi techno Retail
3	Jhon	Setamento	1	Davi techno Retail
4	Davis	Thomas	1	Davi techno Retail
5	Williams	Destrey	2	The BFLO Store
6	Barbara	Rodriguez	2	The BFLO Store
7	Taylor	Mango	2	The BFLO Store
8	Elizabeth	Island	3	Burkes Outlet
9	Brown	Jackson	3	Burkes Outlet

Query executed successfully. (local) (15.0 RTM) DESKTOP-3E95HEO\DataSc... SampleRetail 00:00:00 10 rows

3 Searched CASE Expression

Searched CASE Expression

Syntax

```
CASE
  WHEN condition_1 THEN result_1
  WHEN condition_2 THEN result_2
  WHEN condition_N THEN result_N
  [ ELSE result ]
END
```

Searched CASE Expression

department table

	id	name	dept_name	salary
1	10238	Eric	Economics	72000
2	13378	Karl	Music	42000
3	23493	Jason	Philosophy	45000
4	26299	Jane	Computer Science	91000
5	30766	Jack	Economics	68000
6	40284	Mary	Psychology	78000
7	43087	Brian	Physics	93000
8	53695	Richard	Philosophy	54000
9	58248	Joseph	Political Science	58000
10	63172	David	Art History	65000
11	64378	Elvis	Physics	87000
12	96945	John	Computer Science	80000
13	99231	Santosh	Computer Science	74000

query:

```
1  SELECT first_name, last_name,
2      CASE
3          WHEN salary <= 55000 THEN 'Low'
4          WHEN salary > 55000 AND salary < 80000 THEN 'Middle'
5          WHEN salary >= 80000 THEN 'High'
6      END AS category
7  FROM employees;
8
```

output:

```
1  first_name  last_name   category
2  ----------  ----------  ----------
3  Robert      Gilmore     High
4  Elvis       Ritter      High
5  David       Barrow      High
6  Hugo        Forester    Low
7  Linda       Foster      High
8  Lisa        Wiener      Middle
9  Rodney      Weaver      High
10 Gayle       Meyer       Middle
11 Jason       Christian   High
12 Billie      Lanning     Middle
```

Query Time

Question: Create a new column with the meaning of the values in the Order_Status column. **(use searched case ex.)**

1 = Pending; 2 = Processing; 3 = Rejected; 4 = Completed

Expected Output:

Results | Messages

	order_id	order_status	order_status_desc
1	1	4	Completed
2	2	4	Completed
3	3	4	Completed
4	4	4	Completed
5	5	4	Completed
6	6	4	Completed
7	7	4	Completed
8	8	4	Completed
9	9	4	Completed
10	10	4	Completed

Q... | (local) (15.0 RTM) | DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 1.615 rows

Query Time For You

Question: Create a new column that shows which email service provider ("Gmail", "Hotmail", "Yahoo" or "Other").

Expected Output:

Results | Messages

	first_name	last_name	email	email_service_provider
1	Diane	Flosi	DianeFlosi@msn.com	Other
2	Nana	Gaines	NanaGaines@msn.com	Other
3	Teddy	Mills	TeddyMills@hotmail.com	Hotmail
4	Nicolas	Garrison	NicolasGarrison@aliexpress.com	Other
5	Tessie	Mullen	TessieMullen@aol.com	Other
6	Cyril	Kohl	CyrilKohl@msn.com	Other
7	William	Williams	WilliamWilliams@gmail.com	Gmail
8	Elizabeth	Levy	ElizabethLevy@yahoo.com	Yahoo

Query executed successfully. | (local) (15.0 RTM) | DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 2.000 rows

Query Time For You

Question: Write a query that gives the first and last names of customers who have ordered products from the computer accessories, speakers, and mp4 player categories in the same order.

Expected Output:

Results | Messages

	first_name	last_name
1	Barbara	Shealy
2	Gail	Pitts
3	Rima	Miller
4	Carita	Foreman
5	Charles	Ruta
6	Cira	Lane
7	Tamar	Schultz
8	Robert	Upthegr...
9	William	Jackson
10	Charles	Lawrence
11	Rosalie	Lapage
12	Ressie	Martinez

DESKTOP-3E95HEO\DataSc... | SampleRetail | 00:00:00 | 20 rows

Query Time

Question: By creating a new column, label the orders according to the instructions below:

1. Label the products as **'Not Shipped'** if they were NOT shipped.
2. Label the products as **'Fast'** if they were shipped on the day of order.
3. Label the products as **'Normal'** if they were shipped within two days of the order date.
4. Label the products as **'Slow'** if they were shipped later than two days after the order date.

Results | Messages

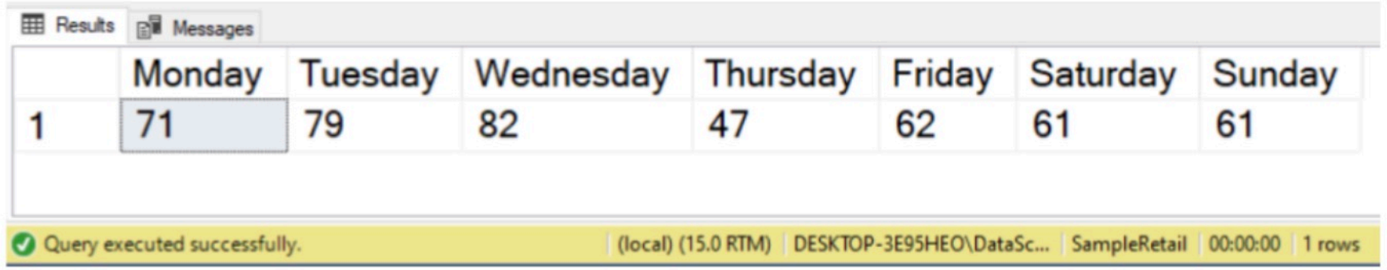
	order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id	ORDER_LABEL
166	1612	3	3	2020-10-21	2020-10-21	NULL	1	3	Not Shipped
167	1613	1	3	2020-11-18	2020-11-18	NULL	2	6	Not Shipped
168	1614	135	3	2020-11-28	2020-11-28	NULL	3	8	Not Shipped
169	1615	136	3	2020-12-28	2020-12-28	NULL	3	8	Not Shipped
170	1407	18	3	2020-02-26	2020-02-26	NULL	2	6	Not Shipped
171	98	1194	4	2018-02-28	2018-02-28	2018-02-28	2	6	Fast
172	102	336	4	2018-03-02	2018-03-05	2018-03-03	2	7	Normal
173	105	306	4	2018-03-03	2018-03-05	2018-03-04	2	7	Normal
174	106	422	4	2018-03-04	2018-03-05	2018-03-05	3	9	Normal
175	108	12	4	2018-03-06	2018-03-09	2018-03-07	2	6	Normal

Query executed successfully. (local) (15.0 RTM) DESKTOP-3E95HEO\DataSc... SampleRetail 00:00:00 1.615 rows

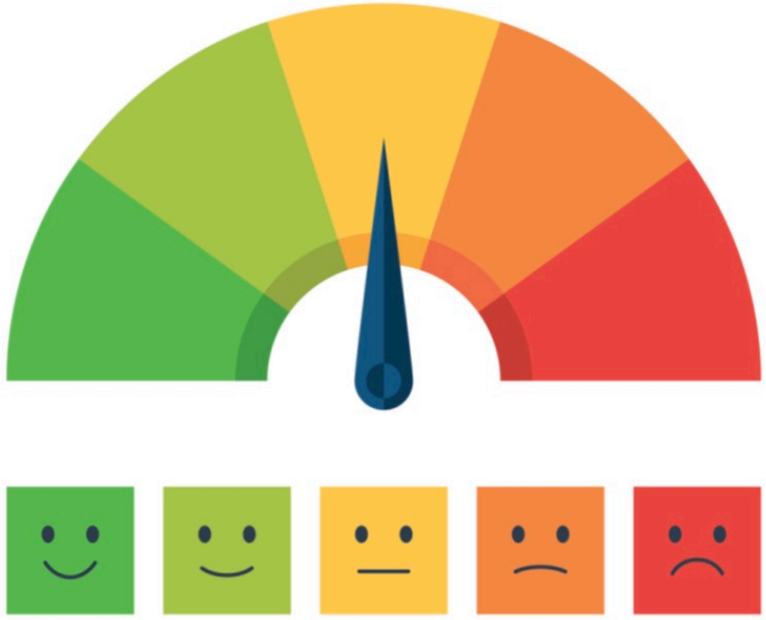
Query Time For You

Question: Write a query that returns the count of the orders day by day in a pivot table format that has been shipped two days later.

Expected Output:



Is everything clear so far?



Students choose an option

Pear Deck Interactive Slide
Do not remove this bar

No Multiple Choice Response
You didn't answer this question