

21 DAYS SQL CHALLENGE

CHALLENGE STARTS FROM

3RD NOVEMBER 2025

REGISTRATION IS
LIVE

SCAN HERE



#SQLWithIDC

Day 11 (14/11): DISTINCT and Handling Duplicates

🎯 Objective

Learn how to use the DISTINCT keyword to eliminate duplicate entries, display only unique values, and ensure data accuracy when querying SQL tables.

🔍 Topics Covered

- DISTINCT keyword
- Removing duplicate records
- Retrieving unique values from data

DISTINCT CLAUSE :-

The DISTINCT keyword is used in SQL to **remove duplicate rows** from the result and return only **unique values**.

Syntax :-

```
SELECT DISTINCT column,  
               column2, ...  
FROM table_name;
```

DISTINCT vs GROUP BY –

Both return unique results, but used in different contexts

Example 1 – Using DISTINCT

```
SELECT  
DISTINCT service, event  
FROM service_weekly;
```

This shows all **unique pairs** of (service, event) – no duplicates.

DISTINCT → simpler, used when you only want **unique rows**.

Example 2 – Using GROUP BY

```
SELECT service, event  
FROM service_weekly  
GROUP BY service, event;
```

This also returns **unique combinations** of service and event

GROUP BY → does the same, and **can be extended** to include **aggregate functions** (like COUNT, SUM, etc.).

	service	event
▶	emergency	none
	surgery	flu
	general_medicine	flu
	ICU	flu
	surgery	none
	ICU	donation
	emergency	flu
	general_medicine	none
	ICU	none
	surgery	donation
	general_medicine	donation
	surgery	strike
	emergency	donation
	general_medicine	strike
	emergency	strike

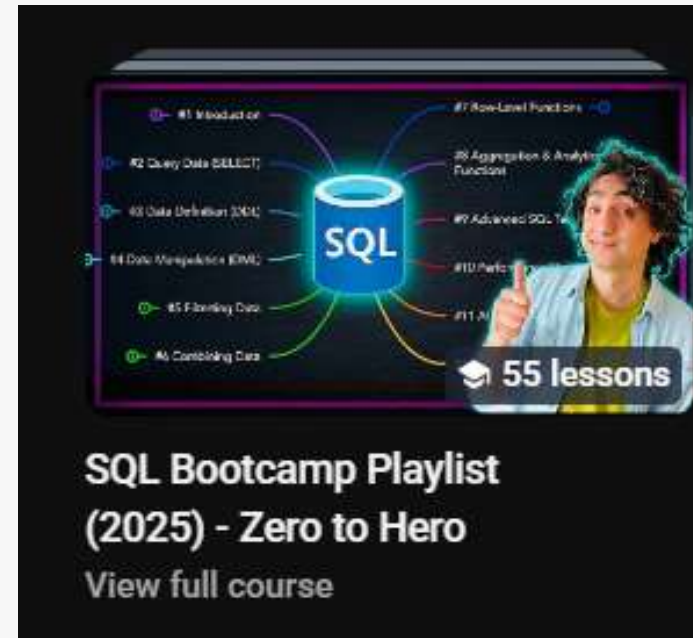
Resources:



SQL Beginner to Advanced For Data...


★ 4.9 (1308) 👤 9032 Enrolled

Beginners to Advanced SQL







SQL Bootcamp Playlist (2025) - Zero to Hero

[View full course](#)

 Data with Baraa **SQL DISTINCT - Remove Duplicate Rows - SQL Tutorial #12**

Practice Questions:

```
1  -- List all unique services in the patients table.
2 • SELECT
3      DISTINCT service AS services
4  FROM patients;
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	services				
▶	ICU				
	general_medicine				
	surgery				
	emergency				

- **SELECT DISTINCT service** → retrieves all unique service names by removing duplicates from the *service* column.
- **AS services** → renames the output column to *services* for a cleaner display.
- **FROM patients** → specifies the *patients* table as the data source.

Practice Questions:

```
6  -- Find all unique staff roles in the hospital.
7 • SELECT
8      DISTINCT role AS staff_role
9  FROM staff;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
staff_role			
▶ doctor			
nursing_assistant			
nurse			

- **SELECT DISTINCT role** → retrieves all unique staff roles by removing duplicates from the **role** column.
- **AS staff_role** → renames the output column to **staff_role** for better readability.
- **FROM staff** → specifies that data is being fetched from the **staff** table.

Practice Questions:

```
11  -- Get distinct months from the services_weekly table.
12 • SELECT
13     DISTINCT month AS month
14 FROM services_weekly;
```

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

	month
▶	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12

- **SELECT DISTINCT** → Returns only unique values, removing duplicates.
- **month AS month** → Selects the month column and renames it as month.
- **FROM services_weekly** → Reads data from the services_weekly table.

Daily Challenge:

```

16  -- Find all unique combinations of service and event type from the services_weekly
17  -- where events are not null or none,
18  -- along with the count of occurrences for each combination.
19  -- Order by count descending.
20 • SELECT
21      service,
22      event,
23      COUNT(*) AS occurrence_count
24  FROM services_weekly
25  WHERE event IS NOT NULL
26        AND event <> 'none'
27  GROUP BY service, event
28  ORDER BY occurrence_count DESC;

```

service	event	occurrence_count
general_medicine	flu	6
ICU	flu	5
emergency	flu	5
surgery	donation	5
surgery	strike	4
emergency	donation	4
emergency	strike	4
surgery	flu	3
general_medicine	donation	3
general_medicine	strike	3
ICU	donation	2

- **SELECT service, event, COUNT(*)** → Retrieves each service-event pair and counts how many times it occurs.
- **FROM services_weekly** → Reads data from the table.
- **WHERE event IS NOT NULL AND event <> 'none'** → Excludes empty or “none” events.
- **GROUP BY service, event** → Groups records by both columns to calculate the count per pair.
- **ORDER BY occurrence_count DESC** → Sorts results so the most frequent combinations appear first.