

PostgreSQL CTE

Query Editor	Query History	Explain	Messages	Notifications
<pre>1 --We will use the film and rental tables from the sample database for the demonstration. 2 WITH cte_film AS (3 SELECT 4 film_id, 5 title, 6 (CASE 7 WHEN length < 30 THEN 'Short' 8 WHEN length < 90 THEN 'Medium' 9 ELSE 'Long' 10 END) length 11 FROM 12 film 13)</pre>				
Data Output				
film_id [PK] integer	title character varying (255)	length text		
1	4 Affair Prejudice	Long		
2	5 African Egg	Long		
3	6 Agent Truman	Long		
4	9 Alabama Devil	Long		
5	11 Alamo Videotape	Long		
6	12 Alaska Phantom	Long		
7	13 Ali Forever	Long		
8	14 Alice Fantasia	Long		
9	16 Alley Evolution	Long		
10	19 Amadeus Holy	Long		
11	21 American Circus	Long		
12	23 Anaconda Confessions	Long		

Query Editor	Query History	Explain	Messages	Notifications
<pre>1 --we will use the rental and staff tables: 2 WITH cte_rental AS (3 SELECT staff_id, 4 COUNT(rental_id) rental_count 5 FROM 6 rental 7 GROUP BY staff_id 8) 9 SELECT s.staff_id, 10 first_name, 11 last_name, 12 rental_count 13 FROM 14 staff s 15 INNER JOIN cte_rental USING (staff_id);</pre>				
Data Output				
staff_id [PK] integer	first_name character varying (45)	last_name character varying (45)	rental_count bigint	
1	1 Mike	Hillyer	8040	
2	2 Jon	Stephens	8004	

<pre>1 --The following statement illustrates how to use the CTE with the RANK() window function: 2 3 WITH cte_film AS (4 SELECT film_id, 5 title, 6 rating, 7 length, 8 RANK() OVER (9 PARTITION BY rating 10 ORDER BY length DESC) 11 length_rank 12 FROM 13 film</pre>				
Data Output				
film_id [PK] integer	title character varying (255)	rating mpaa_rating	length smallint	length_rank bigint
1	182 Control Anthem	G	185	1
2	212 Darn Forrester	G	185	1
3	609 Muscle Bright	G	185	1
4	991 Worst Banger	PG	185	1
5	141 Chicago North	PG-13	185	1
6	349 Gangs Pride	PG-13	185	1
7	690 Pond Seattle	PG-13	185	1
8	817 Soldiers Evolution	R	185	1
9	426 Home Pity	R	185	1
10	872 Sweet Brotherhood	R	185	1
11	821 Security Queue	NC-17	184	1

