


PRAKTIKUM BASIS DATA

BAB	: SUBQUERY SQL	
NAMA	: DANI ADRIAN	
NIM	: 225150201111009	
ASISTEN	: NATHAN DAUD	
	RAKHA HANIF MAHESWARA	
	NAUFAL PUTRA SUTRISNA	
	STANISLAUS FRANS BERNADO	
TGL PRAKTIKUM	: 1511/2023	TGL PENGUMPULAN :16/11/2023

Pertanyaan

1. Menampilkan *instructor* yang belum pernah mengajar!

Implementasi

<pre>USE DBSS_009 SELECT ID, name FROM instructor WHERE ID NOT IN (SELECT ID FROM teaches)</pre>

Tampilan Keluaran

```
1 USE DBSS_009
2
3 SELECT ID, name
4 FROM instructor
5 WHERE ID NOT IN (SELECT ID FROM teaches)
```

133 %

Results Messages

	ID	name
1	16807	Yazdi
2	31955	Moreira
3	35579	Soisalon-Soininen
4	37687	Arias
5	4034	Murata
6	50885	Konstantinides
7	52647	Bancilhon
8	57180	Hau
9	58558	Dusserre
10	59795	Desyl
11	63395	McKinnon
12	64871	Gutierrez
13	72553	Yin
14	74426	Kenje
15	78699	Pingr
16	79653	Levine
17	95030	Arinb
18	96895	Mird
19	97302	Bertolino

Dani Adrian.txt

File Edit View

Dani Adrian
225150201111009

Ln 1, Col 1

Pembahasan dan Analisis

Nama : Dani Adrian

NIM : 225150201111009

Perintah `SELECT (atribut) FROM (tabel) WHERE (kondisi / syarat) NOT IN (subquery)` digunakan Untuk menampilkan Instructor yang belum pernah mengajar.

`NOT IN` merupakan salah satu operator yang menghubungkan sebuah query dengan subquery yang data query tersebut tidak ada di dalam subquery yang ditentukan.

Subquery sendiri terdiri dari perintah `SELECT (atribut) FROM (tabel)` yang mengambil atribut ID dari tabel teaches.

Dapat diartikan bahwa keseluruhan perintah adalah mengambil dan menampilkan daftar instructor yang belum pernah mengajar

Pertanyaan

2. Menampilkan *course* yang pernah diajar setidaknya 2 *instructor*!

Implementasi

```
USE DBSS_009

SELECT c.course_id, title, COUNT(ID) AS
num_of_instructor
FROM course c
JOIN teaches t ON c.course_id = t.course_id
WHERE c.course_id IN (SELECT course_id FROM teaches
GROUP BY course_id HAVING COUNT(ID) >= 2)
GROUP BY title, c.course_id
```

Tampilan Keluaran

SQLQuery3.sql - LA....DBSS_009 (sa (73))
SQLQuery2.sql - LA....DBSS_009 (sa (72))
SQLQuery1.sql - LA....DBSS_009 (sa (71))

```
1 USE DBSS_009
2
3 SELECT c.course_id, title, COUNT(ID) AS num_of_instructor
4 FROM course c
5 JOIN teaches t ON c.course_id = t.course_id
6 WHERE c.course_id IN (SELECT course_id FROM teaches GROUP BY course_id HAVING COUNT(ID) >= 2)
7 GROUP BY title, c.course_id
8
```

133 %

Results Messages

	course_id	title	num_of_instructor
1	105	Image Processing	2
2	158	Elastic Structures	2
3	169	Marine Mammals	2
4	200	The Music of the Ramones	2
5	237	Surfing	2
6	338	Graph Theory	2
7	362	Embedded Systems	3
8	400	Visual BASIC	2
9	408	Bankruptcy	2
10	443	Journalism	2
11	468	Fractal Geometry	2
12	735	Greek Tragedy	2
13	867	The IBM 360 Architecture	2
14	960	Tort Law	2

Dani Adrian.txt

File Edit View

Dani Adrian
225150201111009

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Pembahasan dan Analisis

Nama : Dani Adrian

NIM : 225150201111009

perintah SELECT (atribur) FROM (tabel 1) JOIN (tabel 2) ON (Persamaan pk dengan fk) WHERE (Kondisi / syarat) IN (subquery) GROUP BY (atribur) digunakan Untuk menampilkan course yang Pernah diajar

Setidaknya 2 Instructor

perintah FROM (tabel 1) JOIN (tabel 2) ON (persamaan PK dengan FK) digunakan Untuk menggabungkan 2 tabel, Course dan teaches Untuk menampilkan Course-id serta title dari Course yang ada.

IN merupakan Operator yang menghubungkan Query dengan Subquery.

Perintah Subquery SELECT (atribur) FROM (tabel) GROUP BY (atribur) HAVING (syarat dengan fungsi agregat) digunakan untuk mengambil atribut course-id dari tabel teaches yang dikelompokkan menggunakan perintah GROUP BY berdasarkan atribut course-id dan melakukan perhitungan banyak ID pada setiap course-id dan

No. :

Perintah HAVING Untuk menunjukkan syarat bahwa data yang diambil memiliki ID lebih dari sama dengan 2 pada setiap course-id.

Seluruh perintah berfungsi mengambil dan menampilkan data dari tabel course dimana course-id memiliki lebih dari sama dengan 2 ID dari tabel instructor.

Pertanyaan

3. Menampilkan *student* yang paling banyak mendapat nilai A!

Implementasi

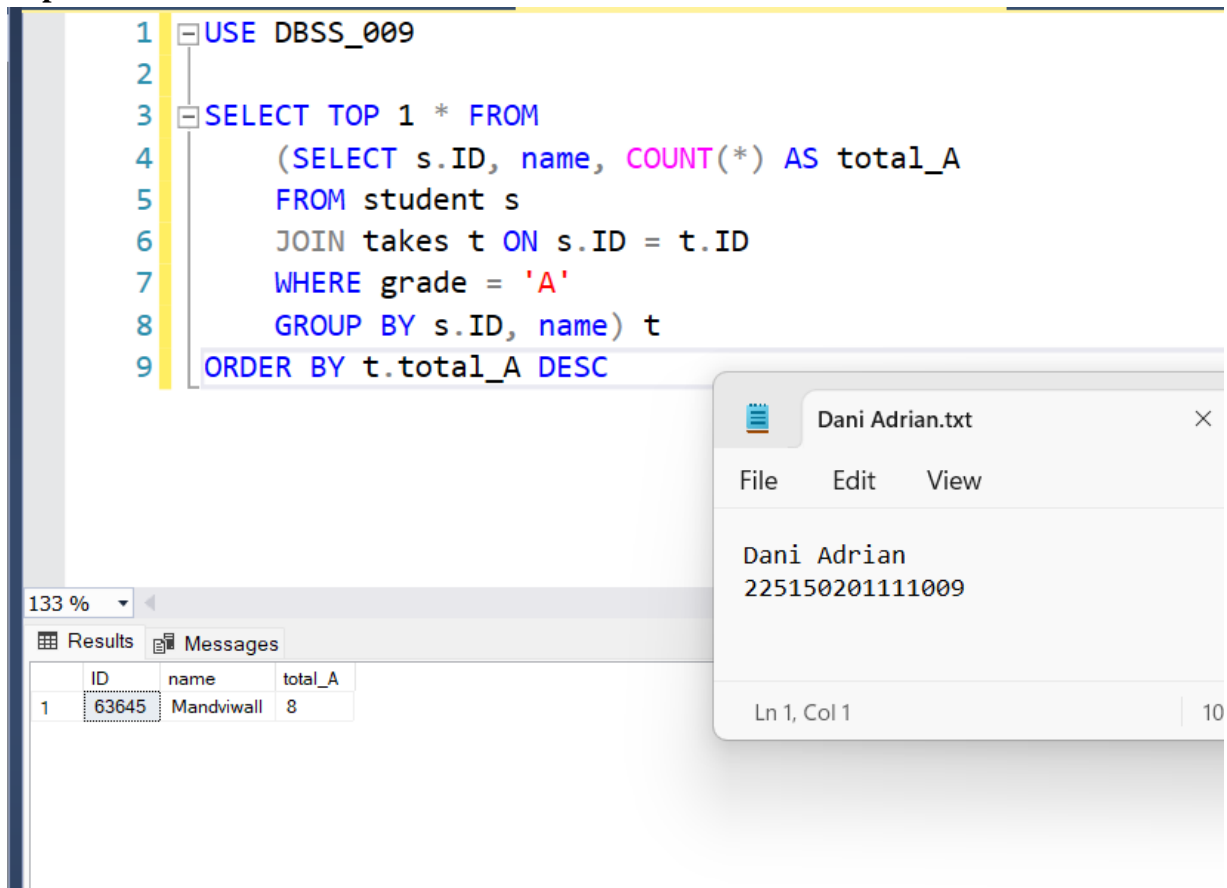
```

USE DBSS_009

SELECT TOP 1 * FROM
    (SELECT s.ID, name, COUNT(*) AS total_A
    FROM student s
    JOIN takes t ON s.ID = t.ID
    WHERE grade = 'A'
    GROUP BY s.ID, name) t
ORDER BY t.total_A DESC

```

Tampilan Keluaran



The screenshot shows a SQL IDE interface. The script editor contains the following SQL query:

```

1 USE DBSS_009
2
3 SELECT TOP 1 * FROM
4     (SELECT s.ID, name, COUNT(*) AS total_A
5     FROM student s
6     JOIN takes t ON s.ID = t.ID
7     WHERE grade = 'A'
8     GROUP BY s.ID, name) t
9 ORDER BY t.total_A DESC

```

Below the script editor, the results are displayed in a table:

	ID	name	total_A
1	63645	Mandviwall	8

In the foreground, a text editor window titled "Dani Adrian.txt" is open, displaying the following text:

```

Dani Adrian
225150201111009

```

The text editor window also shows a menu bar with "File", "Edit", and "View" options, and a status bar at the bottom indicating "Ln 1, Col 1" and a page count of "10".

Pembahasan dan Analisis

Nama : Dani Adrian

NIM : 225150201111009

Perintah `SELECT TOP 1 * FROM (subquery) ORDER BY (atribut) DESC` digunakan untuk menampilkan student yang paling banyak mendapat nilai A.

perintah `SELECT TOP 1 * FROM (subquery)` digunakan untuk menampilkan seluruh data dari nilai teratas yang berasal dan hasil dari subquery yang ditentukan

perintah subquery `SELECT (atribut) FROM (tabel 1) JOIN (tabel 2) ON (persamaan pk dengan fk) WHERE (kondisi / syarat) GROUP BY (atribut).`

perintah `FROM (tabel 1) JOIN (tabel 2) ON (persamaan pk dengan fk)` digunakan untuk menggabungkan 2 tabel untuk menampilkan ID, name, dan `COUNT (*)` yang dimana name hanya tersedia di bagian tabel student saja, Sedangkan untuk menghitung jumlah nilai A hanya tersedia di bagian tabel takes saja.

perintah `GROUP BY (atribut)` digunakan karena menggunakan fungsi agregat.

Seluruh perintah berfungsi untuk mengambil dan menampilkan data dari student yang paling banyak mendapatkan nilai A.