


PRAKTIKUM SISTEM BASIS DATA

BAB	: OPERATOR	
NAMA	: CANNINO ALBY DARMAWAN	
NIM	: 215150200111018	
ASISTEN	: FEMI NOVIA LINA	
	: QOLANDAR ANNURI	
TGL PRAKTIKUM	: 31/10/2022	TGL PENGUMPULAN : 07/10/2022

Pertanyaan

1. Tampilkan data dari tabel **course** yang memiliki **credits** selain 3!

Implementasi

```
SELECT * FROM course
WHERE credits != 3;
```

Tampilan Keluaran

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The central pane shows a SQL query in the 'SQLQuery1.sql' file:

```
use University  
  
SELECT * FROM course  
WHERE credits != 3;
```

The 'Object Explorer' on the left shows the 'University' database selected. The 'Results' pane at the bottom displays the output of the query, which is a table with 4 columns: 'course_id', 'title', 'dept_name', and 'credits'. The table contains 19 rows of data. A status bar at the bottom indicates 'Query executed successfully.' and '92 rows'.

An 'Untitled - Notepad' window is open in the foreground, displaying the text:

```
Cannino Alby Darmawan  
215150200111018
```

Pembahasan dan Analisis

Perintah `SELECT * FROM` digunakan untuk mengambil semua data dan kolom pada tabel `course`. Kemudian menggunakan `WHERE` untuk menentukan constraint berupa nilai `credits` tidak boleh 3.

Pertanyaan

2. Tampilkan data dari tabel **course** yang memiliki **course_id** mulai 140 sampai ke atas!

Implementasi

```
SELECT * FROM course
WHERE course_id >= 140;
```

Tampilan Keluaran

The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows the following SQL code:

```
use University

SELECT * FROM course
WHERE credits != 3;

SELECT * FROM course
WHERE course_id >= 140;
```

The Results pane at the bottom shows the output of the second query, displaying a table with 19 rows of course data. The status bar at the bottom indicates "Query executed successfully." and "192 rows".

	course_id	title	dept_name	credits
1	158	Elastic Structures	Cybernetics	3
2	165	Marine Mammals	Elec. Eng.	3
3	190	Romantic Literature	Civil Eng.	3
4	192	Drama	Languages	4
5	195	Numerical Methods	Geology	4
6	200	The Music of the Ramones	Accounting	4
7	209	International Trade	Cybernetics	4
8	224	International Finance	Athletics	3
9	227	Elastic Structures	Languages	4
10	235	International Trade	Math	3
11	236	Design and Analysis of Algorithms	Mech. Eng.	3
12	237	Surfing	Cybernetics	3
13	238	The Music of Donovan	Mech. Eng.	3
14	239	The Music of the Ramones	Physics	4
15	241	Biostatistics	Geology	3
16	242	Rock and Roll	Marketing	3
17	254	Security	Cybernetics	3
18	258	Colloid and Surface Chemistry	Math	3
19	265	Thermal Physics	Cybernetics	4

Pembahasan dan Analisis

Perintah `SELECT * FROM` digunakan untuk mengambil semua data dan kolom pada tabel `course`. Kemudian menggunakan `WHERE` untuk menentukan constraint berupa nilai `course_id` 140 atau lebih.

Pertanyaan

3. Tampilkan data (dari *join*) `student_id`, `nama`, `course`, `grade` (dari tabel `takes`) dengan syarat `grade` mengandung huruf “C” (bisa C/C+/C-) !

Implementasi

```
SELECT student.ID, student.name, takes.course_id,  
takes.grade  
FROM takes  
JOIN student  
ON takes.ID = student.ID  
WHERE takes.grade like 'C%';
```

Tampilan Keluaran

The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a SQL query in the query editor, which is executed against the 'University' database. The query is as follows:

```
use University

SELECT * FROM course
WHERE credits != 3;

SELECT * FROM course
WHERE course_id >= 140;

SELECT student.ID, student.name, takes.course_id, takes.grade
FROM takes
JOIN student
ON takes.ID = student.ID
WHERE takes.grade like 'C%';
```

The 'Results' pane at the bottom shows the output of the query, which is a table with 4 columns: ID, name, course_id, and grade. The table contains 19 rows of data. A 'Messages' pane is also visible, showing the status 'Query executed successfully.'.

ID	name	course_id	grade	
1	1000	Manber	239	C
2	1000	Manber	571	C+
3	1000	Manber	642	C-
4	1000	Manber	663	C+
5	1000	Manber	802	C+
6	10033	Zelty	334	C-
7	10033	Zelty	338	C
8	10033	Zelty	408	C-
9	10033	Zelty	443	C-
10	10033	Zelty	445	C
11	10033	Zelty	457	C-
12	10033	Zelty	486	C
13	10033	Zelty	493	C-
14	10033	Zelty	791	C
15	10033	Zelty	960	C+
16	10033	Zelty	972	C+
17	10076	Duan	105	C-
18	10076	Duan	629	C
19	10076	Duan	962	C+

A 'Notepad' window is also open, showing the text 'Cannino Alby Darmawan' and '215150200111018'.

Pembahasan dan Analisis

Perintah SELECT untuk mengambil data dari dua tabel, yaitu ID dan name dari tabel student, serta course_id dan grade dari tabel takes. FROM untuk mengambil data dari tabel takes kemudian JOIN untuk menggabungkan tabel takes dan student. ON digunakan untuk menggabungkan student dengan takes berdasarkan ID. WHERE untuk memberikan constraint berupa grade harus mengandung huruf 'C' dengan menggunakan likes.

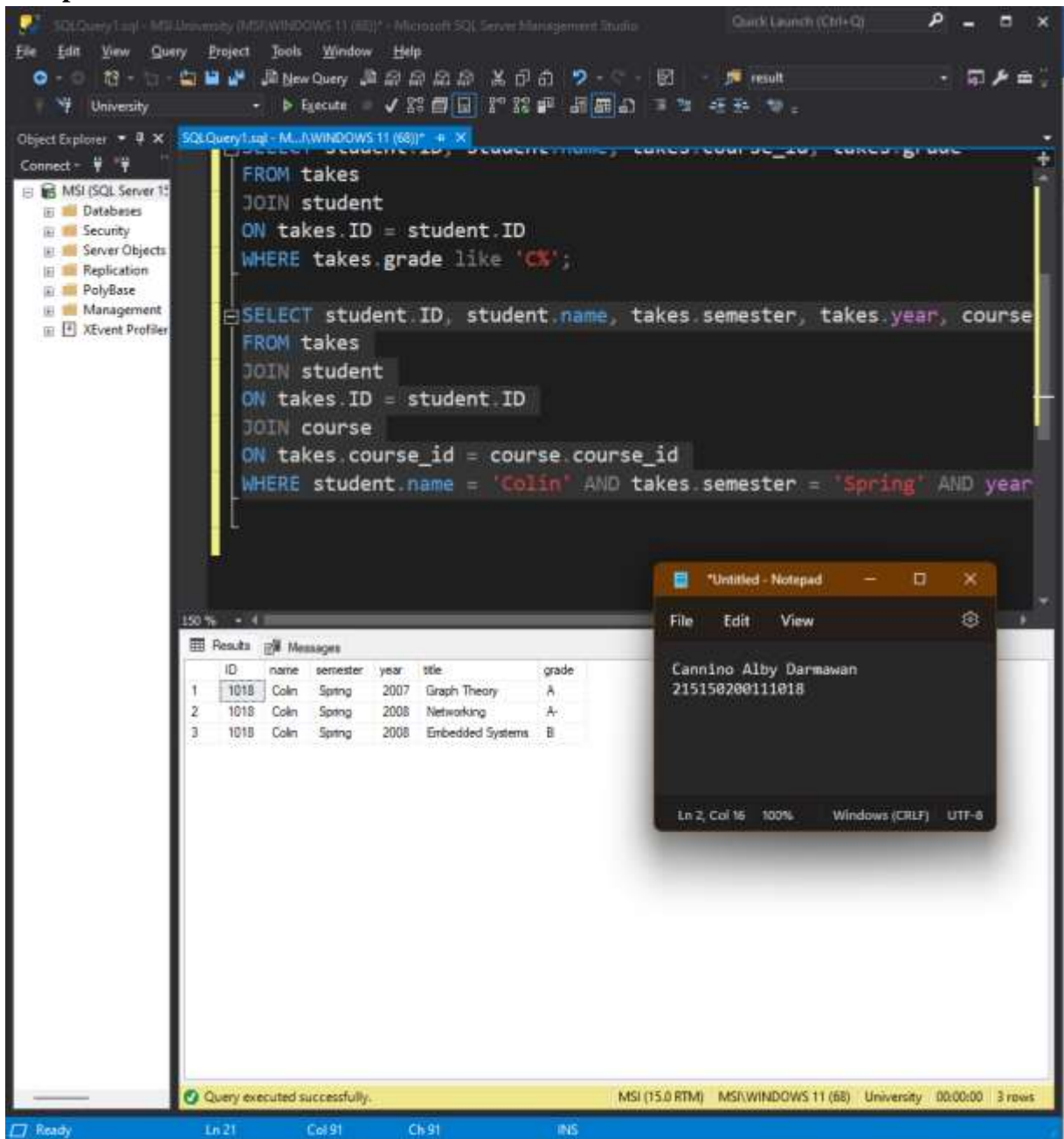
Pertanyaan

4. Modifikasi *query* untuk menampilkan semua nilai **Colin** semester **spring** dan tahun **2007** dan **2008**!

Implementasi

```
SELECT student.ID, student.name, takes.semester,  
takes.year, course.title, takes.grade  
FROM takes  
JOIN student  
ON takes.ID = student.ID  
JOIN course  
ON takes.course_id = course.course_id  
WHERE student.name = 'Colin' AND takes.semester =  
'Spring' AND year BETWEEN 2007 AND 2008;
```

Tampilan Keluaran



The screenshot displays the Microsoft SQL Server Management Studio interface. The SQL Query Editor shows a query that joins the `student`, `takes`, and `course` tables. The query filters for students named 'Colin' and takes from the 'Spring' semester. The Results pane shows the output of the query, which consists of three rows of data. A Notepad window is overlaid on the results, displaying the text: Cannino Alby Darmawan 215150200111018.

```
SELECT student.ID, student.name, takes.semester, takes.year, course.title, takes.grade
FROM takes
JOIN student
ON takes.ID = student.ID
WHERE takes.grade like 'C%';

SELECT student.ID, student.name, takes.semester, takes.year, course.title, takes.grade
FROM takes
JOIN student
ON takes.ID = student.ID
JOIN course
ON takes.course_id = course.course_id
WHERE student.name = 'Colin' AND takes.semester = 'Spring' AND year = 2008
```

ID	name	semester	year	title	grade
1018	Colin	Spring	2007	Graph Theory	A
1018	Colin	Spring	2008	Networking	A-
1018	Colin	Spring	2008	Embedded Systems	B

Cannino Alby Darmawan
215150200111018

Pembahasan dan Analisis

SELECT digunakan untuk mengambil data dari 3 tabel, yaitu ID dan name dari tabel student, semester, year, dan grade dari tabel takes, serta title dari tabel course. FROM digunakan untuk mengambil data dari tabel takes. JOIN student untuk menggabungkan tabel takes dan student dengan menggunakan ON sebagai persyaratan bahwa ID dari takes harus sama dengan yang ada di student. Kemudian menggunakan JOIN lagi untuk menggabungkan dengan tabel course dengan menggunakan ON sebagai persyaratan bahwa course_id dari takes harus sama dengan

yang ada di course. WHERE digunakan untuk memberi constraint yaitu nama dari student harus 'Colin', semester dari takes harus 'Spring', dan harus berada antara tahun 2007 dan 2008.

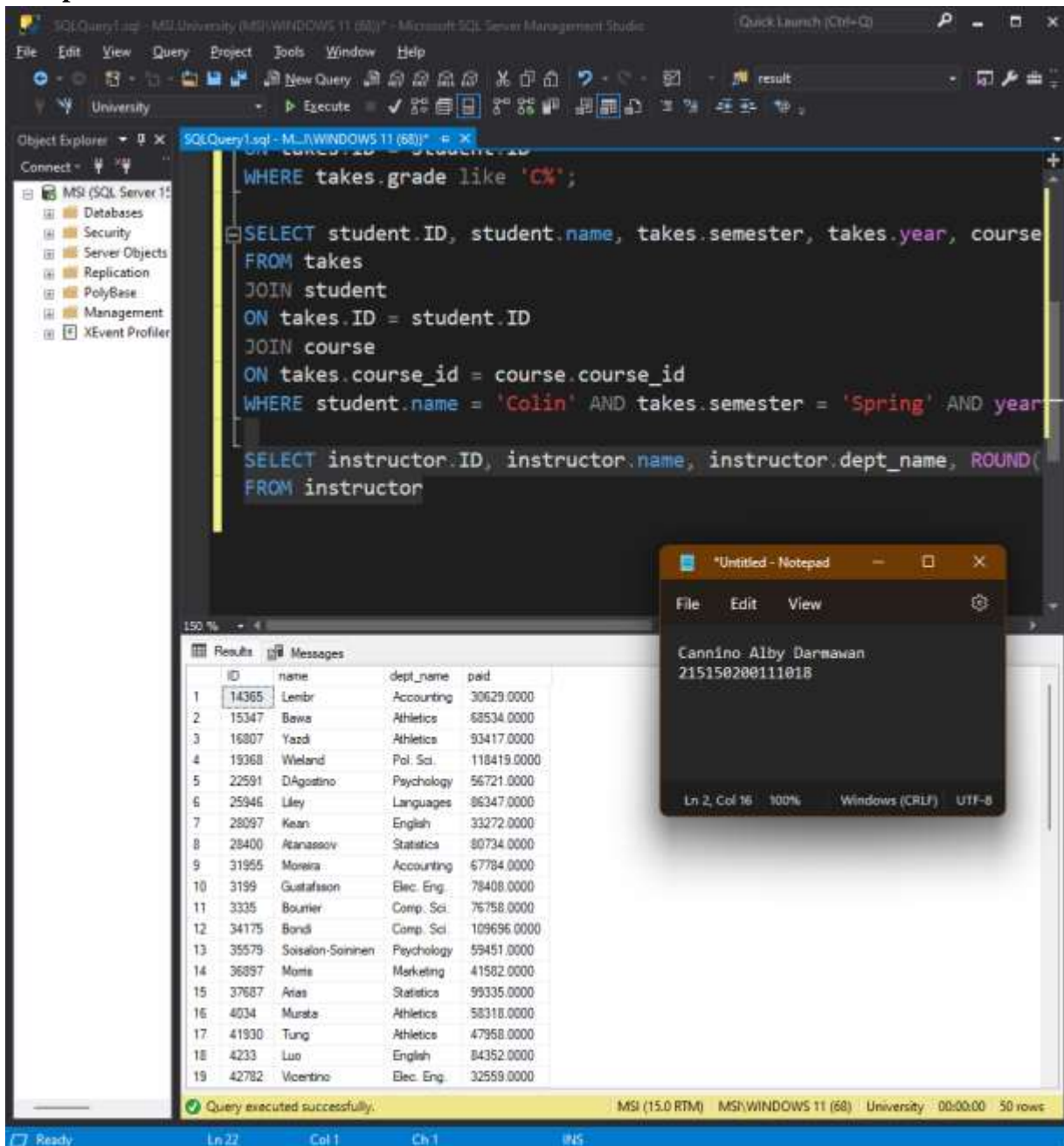
Pertanyaan

5. Tampilkan gaji bersih **isnstructor** dengan **pembulatan**!

Implementasi

```
SELECT instructor.ID, instructor.name,  
instructor.dept_name, ROUND(salary - (salary * 0.05),  
0) as paid  
FROM instructor
```


Tampilan Keluaran



The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows a SQL query that selects student information and instructor salary. The query is as follows:

```
WHERE takes.grade like 'CX';

SELECT student.ID, student.name, takes.semester, takes.year, course
FROM takes
JOIN student
ON takes.ID = student.ID
JOIN course
ON takes.course_id = course.course_id
WHERE student.name = 'Colin' AND takes.semester = 'Spring' AND year

SELECT instructor.ID, instructor.name, instructor.dept_name, ROUND(
FROM instructor
```

The Results pane shows the output of the query, which is a list of instructors with their IDs, names, departments, and salaries. The data is as follows:

ID	name	dept_name	paid
14365	Lembr	Accounting	30629.0000
15347	Bawa	Athletics	68534.0000
16807	Yazd	Athletics	93417.0000
19368	Wieland	Pol. Sci.	118419.0000
22591	DiAgostino	Psychology	56721.0000
25946	Liley	Languages	86347.0000
28097	Kean	English	33272.0000
28400	Atanassov	Statistics	80734.0000
31955	Moreira	Accounting	67784.0000
3199	Gustafsson	Elec. Eng.	78408.0000
3335	Bourier	Comp. Sci.	76758.0000
34175	Bond	Comp. Sci.	109696.0000
35579	Saisalon-Soininen	Psychology	59451.0000
36897	Monte	Marketing	41582.0000
37687	Arias	Statistics	99335.0000
4034	Murata	Athletics	58318.0000
41930	Tung	Athletics	47958.0000
4233	Luo	English	84352.0000
42782	Vicentino	Elec. Eng.	32559.0000

A Notepad window is also open, showing the following text:

```
Cannino Alby Darmawan
215150200111018
```

Pembahasan dan Analisis

SELECT untuk mengambil data, yaitu ID, name, dept_name, salary dan perintah FROM untuk ,mengambil data dari tabel instructor. ROUND digunakan untuk pembulatan gaji bersih dari instructor.

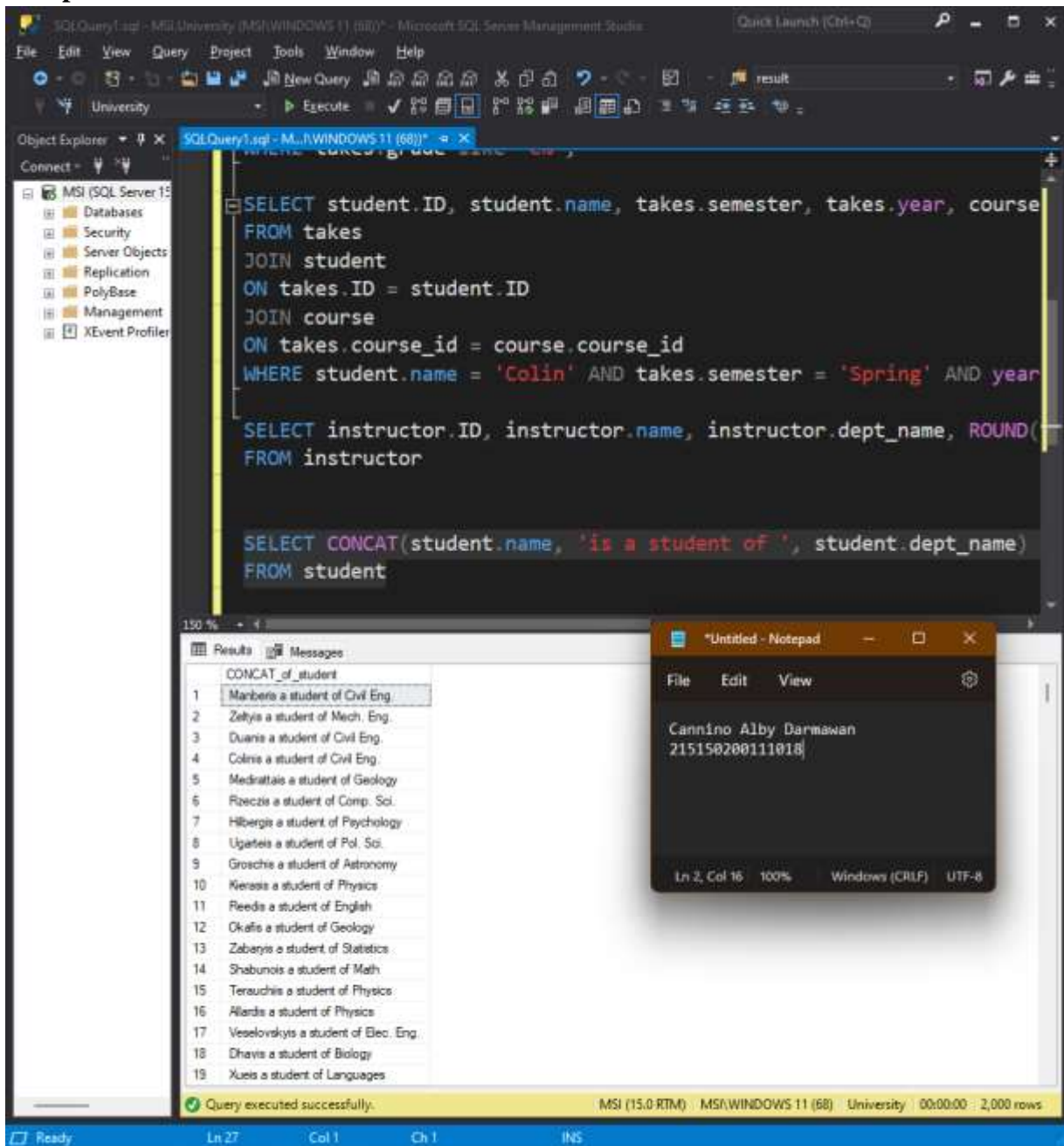
Pertanyaan

6. Tampilkan gabungan *field* **name** dan **dept_name** dari student menjadi bentuk “**name is a student of departement**” (contoh **Colin is student of Civil eng.**) dengan salah satu fungsi *string*!

Implementasi

```
SELECT CONCAT(student.name, 'is a student of ',  
student.dept_name) as CONCAT_of_student  
FROM student
```

Tampilan Keluaran



Pembahasan dan Analisis

Perintah SELECT untuk mengambil data name, dept_name kemudian FROM untuk mengambil data dari student. Fungsi CONCAT untuk menggabungkan string.