

**TUGAS SECTION-01**  
**FUNDAMENTAL ORACLE SQL**



Oleh :

Shinta Anggreina

[shintaanggreina@gmail.com](mailto:shintaanggreina@gmail.com)

ID Kegiatan: 9267714

**DATA ENGINEER WITH DATABASE SQL APPROACH**  
**METRODATA ACADEMY**  
**STUDI INDEPENDEN BERSERTIFIKAT KAMPUS MERDEKA**  
**2024**

## KETENTUAN TUGAS

### Tugas :

1. Silakan tampilkan semua data pada kolom **COL\$**.
2. Silakan tampilkan hanya kolom **COL#**, **Name** dan **Length** dan tolong di rename menjadi **COLUMN**, **NAME COL\$** dan **LENGTH\_COL\$**.
3. **Melanjutkan dari soal no.2** silakan tampilkan jumlahkan total **NAME** yang sama dengan kondisi **COL# = 10** dan Total **NAME** lebih besar dari 5. Lalu order **Total NAME** secara Descending.
4. Buat sebuah table dengan nama **"TBL\_NUMERIC"** dengan nama field **"myNumeric"** numeric dan tentukan berdasarkan angka precision terbesar dan scale 2 berikut;

- 154896.245
- 12416.223
- 124.24565

Setelah menginputkan data silakan tampilkan, hapus semua data di dalam table dan drop table yang telah dibuat.

5. Buat sebuah table dengan nama **"TBL\_STRING"** dengan nama field **"myString"** varchar2 dan tentukan panjang karakter berdasarkan unicode byte terbesar berikut;

- 我喜欢编码
- أنا أحب الترميز
- I like coding

6. Silakan buat select statement berikut **"Hello im a billionaire \*\*\*\*\*"** silakan gunakan String function agar hasilnya sebagai berikut **"Hello im a TRILLIONAIRE"**
7. Silakan ubah String ini **"£5,280,450.89"** menjadi Number dengan output **5280450.89**
8. Silakan ubah Number **128450000** ini menjadi string dengan output sebagai berikut **"The price of this laptop Rp.128,450,000.00"** perlu diperhatikan **"Rp."** harus menjadi format Lokal currency.
9. Buat select statement tampilkan **current\_date** dan dari current\_date tampilkan saat kalian mengerjakan tugasnya kita sedang berada diminggu ke berapa dalam bulan ini.
10. Silakan buat table dengan nama **"TBL\_DATE"** dan Filed **"MYDATE"** lalu inputkan data berikut;

- 20-Mar-2024
- 2024-02-15
- 12-Jun-2024

Lalu tampilkan First Day berdasarkan inputan yang ada sebelumnya.

Ketentuan Pengumpulan:

- **!!~Screenshot code, Result Query & penjelasan~!!**
- File dalam bentuk PDF.
- File name : SIBKM\_Tugas\_Section-01

Shinta - anggreina

## QUERY TUGAS

- No 1: Menampilkan semua data dari tabel COL\$

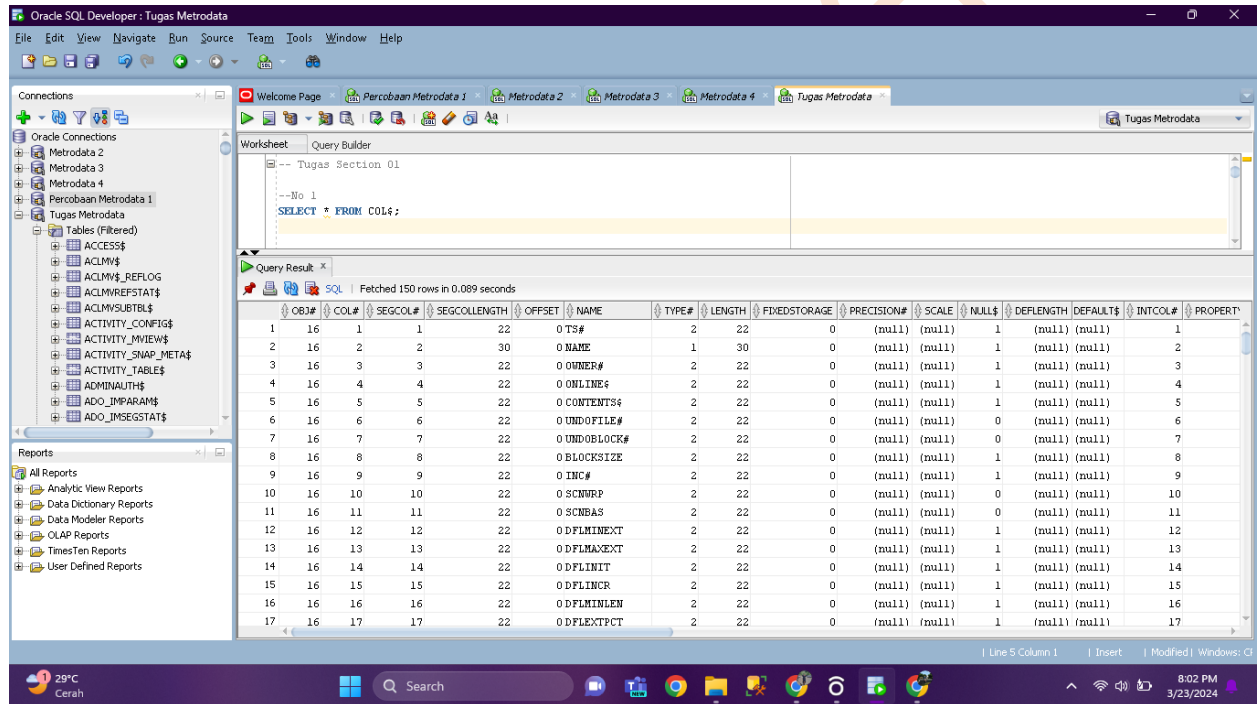
Query:

```
SELECT * FROM COL$;
```

Penjelasan query:

Query tersebut adalah perintah SQL untuk menampilkan semua data yang tersimpan dalam tabel yang bernama 'COL\$'. Kemudian, 'SELECT \*' digunakan untuk memilih semua kolom dalam sebuah tabel, sedangkan 'FROM COL\$' menunjukkan bahwa data akan diambil dari tabel 'COL\$'. Jadi, keseluruhan perintah ini akan menampilkan semua baris dan kolom yang ada dalam tabel 'COL\$'.

Dokumentasi:



The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' window displays the query: `SELECT * FROM COL$;`. The 'Query Result' window shows the results of the query, which are 17 rows of data. The columns in the result are: COL#, COL\$, SEGCOL#, SEGCOLLENGTH, OFFSET, NAME, TYPE#, LENGTH, FIXED STORAGE, PRECISION#, SCALE, NULL\$, DEFLLENGTH, DEFAULT\$, INTCOL#, and PROPERTY#.

COL#	COL\$	SEGCOL#	SEGCOLLENGTH	OFFSET	NAME	TYPE#	LENGTH	FIXED STORAGE	PRECISION#	SCALE	NULL\$	DEFLLENGTH	DEFAULT\$	INTCOL#	PROPERTY#
1	16	1	1	22	0 TS#	2	22	0	(null)	(null)	1	(null)	(null)	1	
2	16	2	2	30	0 NAME	1	30	0	(null)	(null)	1	(null)	(null)	2	
3	16	3	3	22	0 OWNER#	2	22	0	(null)	(null)	1	(null)	(null)	3	
4	16	4	4	22	0 ONLINE\$	2	22	0	(null)	(null)	1	(null)	(null)	4	
5	16	5	5	22	0 CONTENTS\$	2	22	0	(null)	(null)	1	(null)	(null)	5	
6	16	6	6	22	0 UNDOFILE#	2	22	0	(null)	(null)	0	(null)	(null)	6	
7	16	7	7	22	0 UNDOBLOCK#	2	22	0	(null)	(null)	0	(null)	(null)	7	
8	16	8	8	22	0 BLOCKSIZE	2	22	0	(null)	(null)	1	(null)	(null)	8	
9	16	9	9	22	0 INC#	2	22	0	(null)	(null)	1	(null)	(null)	9	
10	16	10	10	22	0 SCNRWP	2	22	0	(null)	(null)	0	(null)	(null)	10	
11	16	11	11	22	0 SCNBAS	2	22	0	(null)	(null)	0	(null)	(null)	11	
12	16	12	12	22	0 DFLMINEXT	2	22	0	(null)	(null)	1	(null)	(null)	12	
13	16	13	13	22	0 DFLMAXEXT	2	22	0	(null)	(null)	1	(null)	(null)	13	
14	16	14	14	22	0 DFLINIT	2	22	0	(null)	(null)	1	(null)	(null)	14	
15	16	15	15	22	0 DFLINCR	2	22	0	(null)	(null)	1	(null)	(null)	15	
16	16	16	16	22	0 DFLMINLEN	2	22	0	(null)	(null)	1	(null)	(null)	16	
17	16	17	17	22	0 DFLXPCT	2	22	0	(null)	(null)	1	(null)	(null)	17	

- No 2: Menampilkan kolom COL#, Name, dan Length dari tabel COL\$, dengan penggantian nama kolom sesuai permintaan

Query:

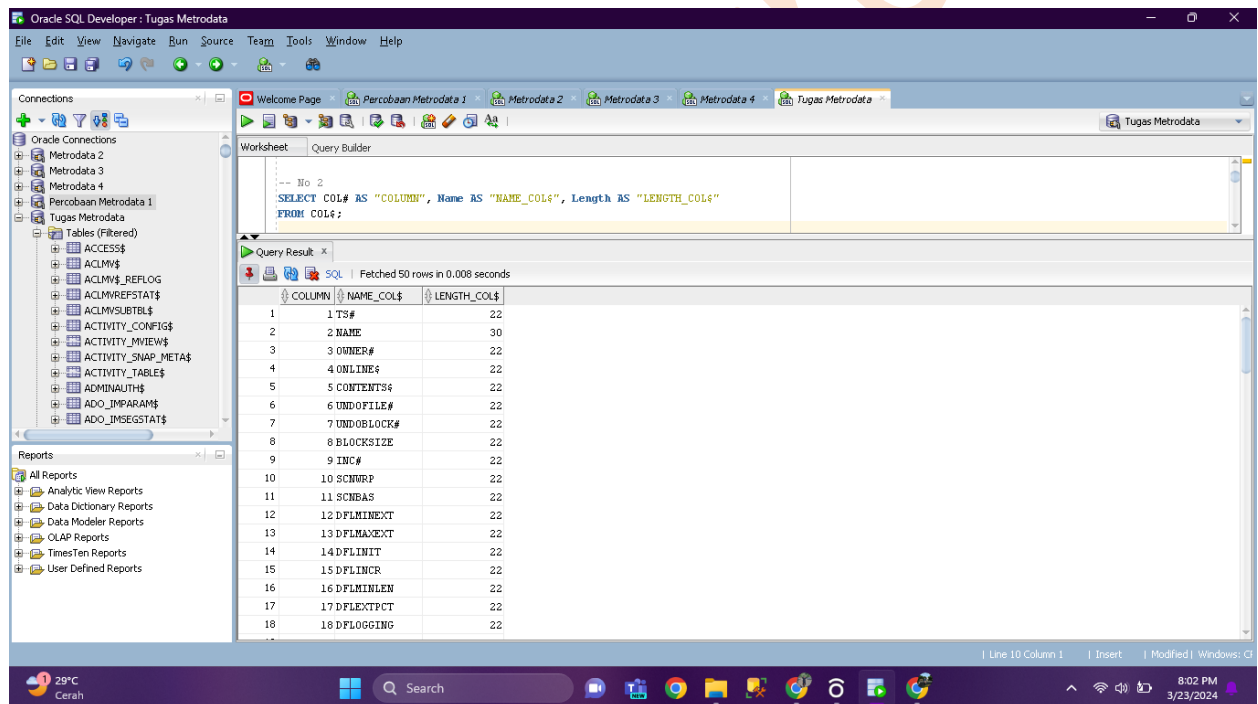
```
SELECT COL# AS "COLUMN", Name AS "NAME_COL$", Length AS "LENGTH_COL$" FROM COL$;
```

Penjelasan query:

Perintah ini adalah query SQL yang digunakan untuk menampilkan kolom-kolom tertentu dari tabel 'COL\$', dengan mengganti nama kolom sesuai dengan keinginan. Berikut adalah penjelasan untuk setiap bagian dari query tersebut:

- 'SELECT': Ini adalah klausa dalam SQL yang digunakan untuk memilih data dari tabel database.
  - 'COL# AS "COLUMN"': Menggunakan nilai dalam kolom 'COL#' dari tabel 'COL\$', tetapi mengganti nama kolom tersebut menjadi "COLUMN" pada hasil output.
  - 'Name AS "NAME\_COL\$"' : Menggunakan nilai dalam kolom 'Name' dari tabel 'COL\$', tetapi mengganti nama kolom tersebut menjadi "NAME\_COL\$" pada hasil output.
  - 'Length AS "LENGTH\_COL\$"' : Menggunakan nilai dalam kolom 'Length' dari tabel 'COL\$', tetapi mengganti nama kolom tersebut menjadi "LENGTH\_COL\$" pada hasil output.
  - 'FROM COL\$': Menunjukkan bahwa data akan diambil dari tabel 'COL\$'.
- Jadi, secara keseluruhan, query tersebut akan menghasilkan output yang menampilkan nilai dari kolom 'COL#' dengan nama "COLUMN", nilai dari kolom 'Name' dengan nama "NAME\_COL\$", dan nilai dari kolom 'Length' dengan nama "LENGTH\_COL\$".

### Dokumentasi:



- No 3: Menghitung jumlah nama yang sama dimana COL# = 10, hanya menampilkan yang jumlahnya lebih dari 5, dan diurutkan secara descending

### Query:

```
SELECT COUNT(Name) AS Total_Name
FROM COL$
WHERE COL# = 10
GROUP BY Name
HAVING COUNT(Name) > 5
```

ORDER BY Total\_Name DESC;

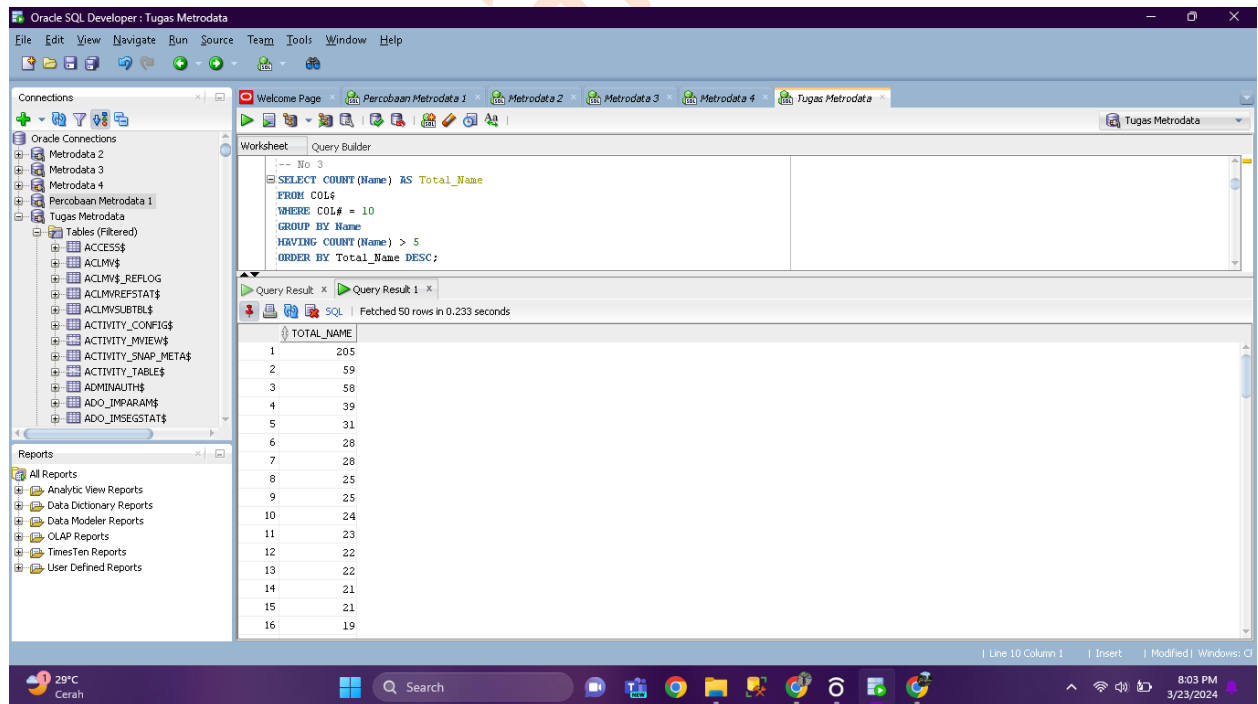
Penjelasan query:

Perintah yang digunakan untuk menghitung jumlah kemunculan nama yang sama dari kolom 'Name' dalam tabel 'COL\$', dengan kondisi bahwa nilai kolom 'COL#' adalah 10. Berikut adalah penjelasan singkat untuk setiap bagian dari query tersebut:

- 'SELECT COUNT(Name) AS Total\_Name': Menghitung jumlah kemunculan nama yang sama dalam kolom 'Name' dan menamai hasilnya sebagai "Total\_Name".
- 'FROM COL\$': Menunjukkan bahwa data akan diambil dari tabel 'COL\$'.
- 'WHERE COL# = 10': Menyaring baris-baris di mana nilai kolom 'COL#' adalah 10.
- 'GROUP BY Name': Mengelompokkan hasil berdasarkan nilai yang ada dalam kolom 'Name'.
- 'HAVING COUNT(Name) > 5': Menerapkan filter setelah pengelompokkan, hanya mempertahankan kelompok yang memiliki jumlah nama lebih dari 5.
- 'ORDER BY Total\_Name DESC': Mengurutkan hasil berdasarkan total nama secara descending.

Jadi, secara keseluruhan, perintah ini akan menghasilkan daftar nama yang memiliki jumlah kemunculan lebih dari 5, diurutkan berdasarkan jumlah kemunculan nama tersebut secara descending.

Dokumentasi:



The screenshot displays the Oracle SQL Developer interface. The 'Query Builder' window shows the following SQL query:

```
-- No 3
SELECT COUNT(Name) AS Total_Name
FROM COL$
WHERE COL# = 10
GROUP BY Name
HAVING COUNT(Name) > 5
ORDER BY Total_Name DESC;
```

The 'Query Result' window shows the results of the query, fetched 50 rows in 0.233 seconds. The results are as follows:

	TOTAL_NAME
1	205
2	59
3	58
4	39
5	31
6	28
7	28
8	25
9	25
10	24
11	23
12	22
13	22
14	21
15	21
16	19

- No 4: Membuat tabel TBL\_NUMERIC, mengisi data, menampilkan data, menghapus semua data, dan menghapus tabel

Query:

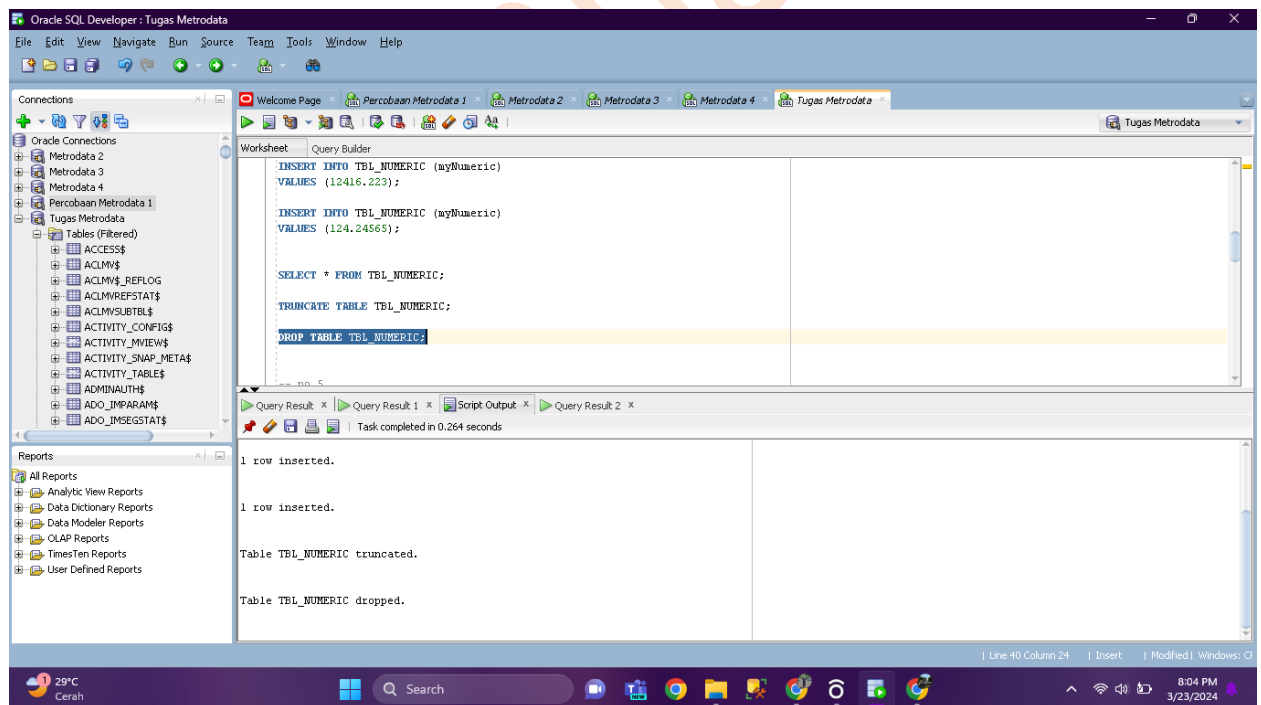
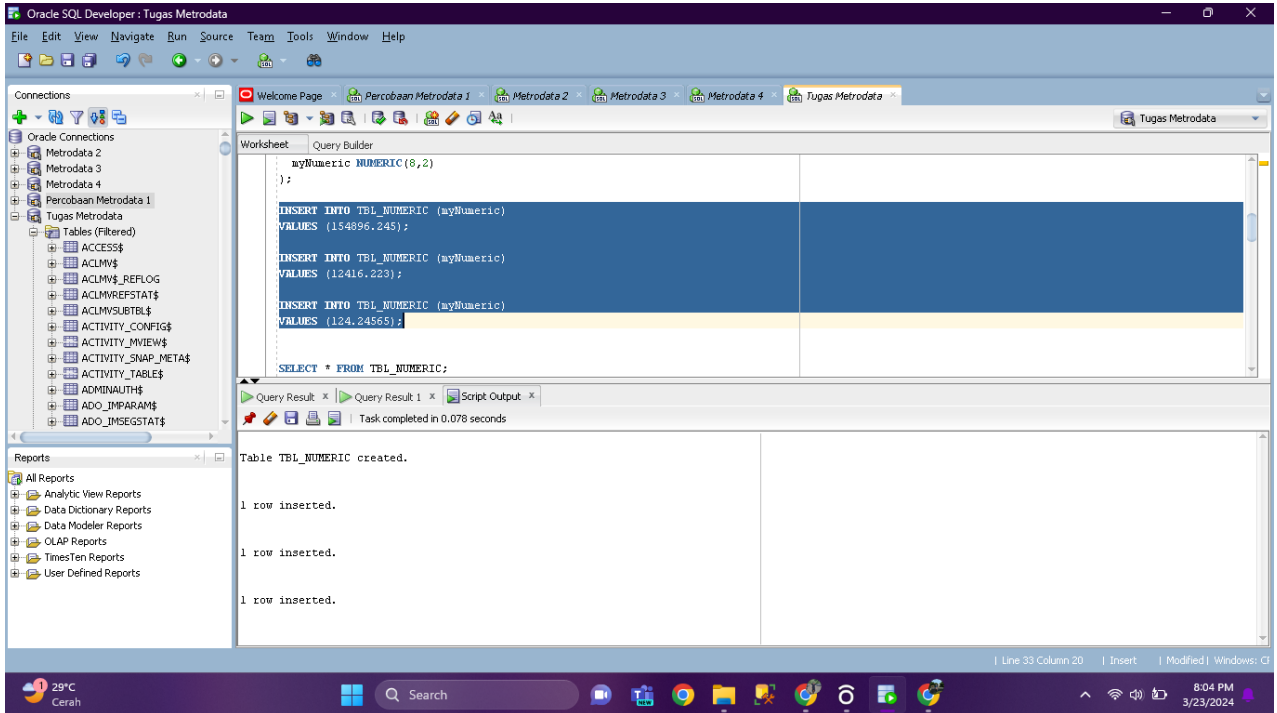
```
CREATE TABLE TBL_NUMERIC (  
    myNumeric NUMERIC(8,2)  
);  
INSERT INTO TBL_NUMERIC (myNumeric)  
VALUES (154896.245);  
INSERT INTO TBL_NUMERIC (myNumeric)  
VALUES (12416.223);  
INSERT INTO TBL_NUMERIC (myNumeric)  
VALUES (124.24565); SELECT * FROM TBL_NUMERIC;
```

```
TRUNCATE TABLE TBL_NUMERIC;  
DROP TABLE TBL_NUMERIC;
```

Penjelasan query:

- CREATE TABLE TBL\_NUMERIC ( myNumeric NUMERIC(8,2) ): Membuat tabel baru bernama TBL\_NUMERIC dengan satu kolom bernama myNumeric yang memiliki tipe data NUMERIC dengan panjang maksimum 8 digit dan 2 digit di belakang koma.
- INSERT INTO TBL\_NUMERIC (myNumeric) VALUES (154896.245): Memasukkan nilai 154896.245 ke dalam kolom myNumeric dalam tabel TBL\_NUMERIC.
- SELECT \* FROM TBL\_NUMERIC: Menampilkan semua data yang ada dalam tabel TBL\_NUMERIC.
- TRUNCATE TABLE TBL\_NUMERIC: Menghapus semua data dalam tabel TBL\_NUMERIC tetapi mempertahankan struktur tabel.
- DROP TABLE TBL\_NUMERIC: Menghapus tabel TBL\_NUMERIC beserta semua data dan strukturnya dari database.

## Dokumentasi:





- No 5: Membuat tabel TBL\_STRIIING, mengisi data, menampilkan data dan panjang byte dari string, dan menghapus tabel

Query:

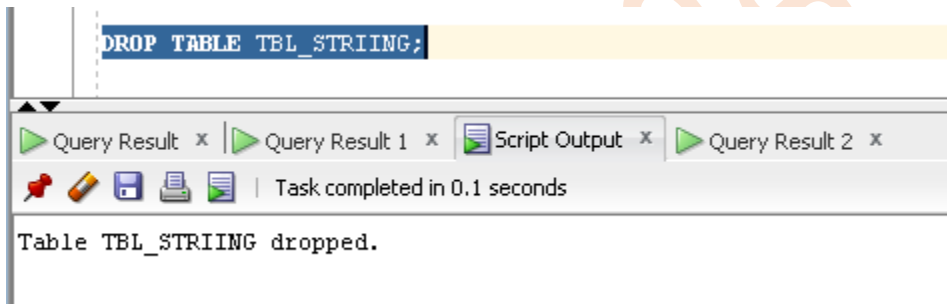
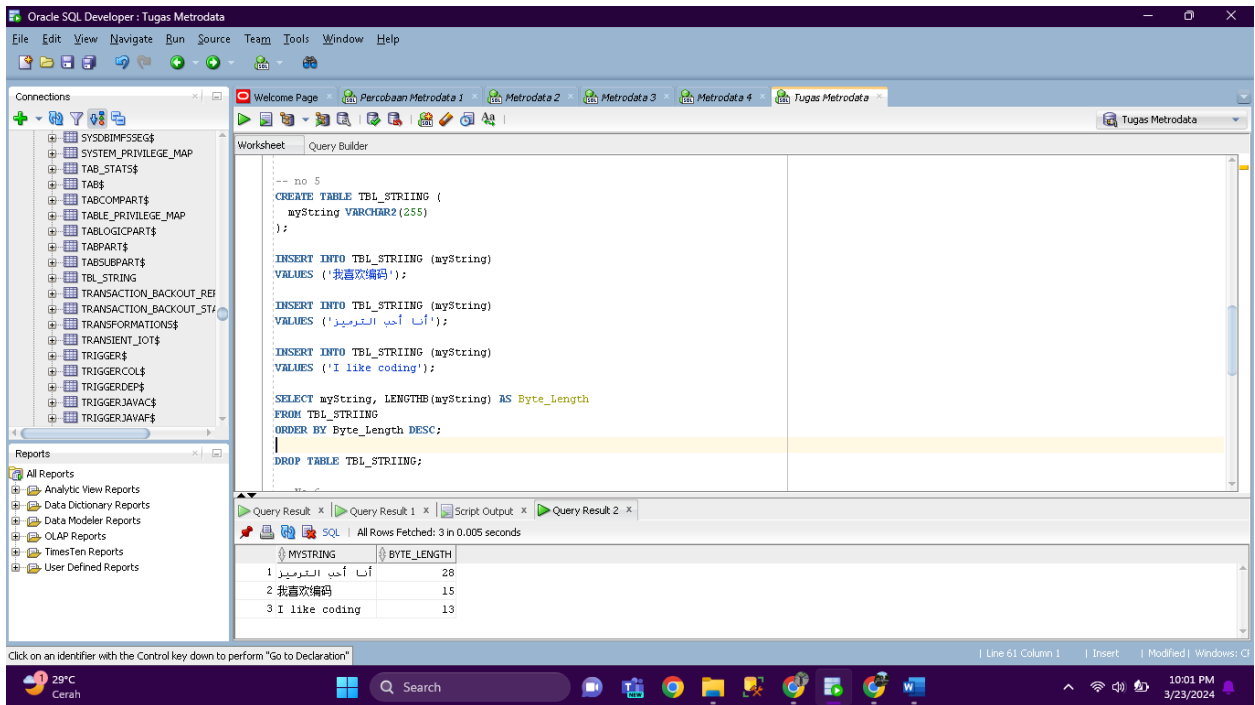
```
CREATE TABLE TBL_STRIIING (
  myString VARCHAR2(255)
);
INSERT INTO TBL_STRIIING (myString)
VALUES ('我喜欢编码');
INSERT INTO TBL_STRIIING (myString)
VALUES ('أنا أحب الترميز');
INSERT INTO TBL_STRIIING (myString)
VALUES ('I like coding');
SELECT myString, LENGTHB(myString) AS Byte_Length
FROM TBL_STRIIING
ORDER BY Byte_Length DESC;
```

DROP TABLE TBL\_STRIIING;

Penjelasan query:

- CREATE TABLE TBL\_STRIIING ( myString VARCHAR2(255) ): Membuat tabel baru bernama TBL\_STRIIING dengan satu kolom bernama myString yang memiliki tipe data VARCHAR2 dengan panjang maksimum 255 karakter.
- INSERT INTO TBL\_STRIIING (myString) VALUES ('我喜欢编码'): Memasukkan nilai '我喜欢编码' ke dalam kolom myString dalam tabel TBL\_STRIIING.
- SELECT myString, LENGTHB(myString) AS Byte\_Length FROM TBL\_STRIIING ORDER BY Byte\_Length DESC: Menampilkan nilai dari kolom myString beserta panjang byte dari setiap string dalam tabel TBL\_STRIIING, diurutkan secara descending berdasarkan panjang byte.
- DROP TABLE TBL\_STRIIING: Menghapus tabel TBL\_STRIIING beserta semua data dan strukturnya dari database.

Dokumentasi:



- No 6: Mengganti kata "billionaire" dengan "TRILLIONAIRE" dalam string

Query:

```
SELECT REPLACE(REPLACE('Hello im a billionaire **', 'billionaire', 'TRILLIONAIRE'), '
**', '')
```

AS Result  
FROM DUAL;

Penjelasan query:

Query diatas mengganti dua bagian teks dalam string 'Hello im a billionaire \*\*'. Pertama, mengganti kata 'billionaire' dengan 'TRILLIONAIRE'. Kedua, menghilangkan ' \*\*' dari string tersebut. Hasil akhir dari perintah tersebut adalah 'Hello im a TRILLIONAIRE'. Perintah tersebut dieksekusi pada tabel DUAL, yang merupakan tabel sistem yang umumnya digunakan untuk operasi temporer dalam database Oracle.

Dokumentasi:

```
-- No 6
SELECT REPLACE(REPLACE('Hello im a billionaire **', 'billionaire', 'TRILLIONAIRE'), ' **', '')
AS Result
FROM DUAL;
```

Query Result x Query Result 1 x Script Output x Query Result 2 x

SQL | All Rows Fetched: 1 in 0.001 seconds

RESULT
1 Hello im a TRILLIONAIRE

- No 7: Mengubah string "£5,280,450.89" menjadi angka

Query:

```
SELECT TO_NUMBER(REPLACE('£5,280,450.89', '£', ''), '9999999990.99') AS Result
FROM DUAL;
```

Penjelasan query:

Perintah ini mengkonversi string '£5,280,450.89' menjadi tipe data numerik dengan menghilangkan simbol poundsterling (£) terlebih dahulu. Kemudian, nilai numerik yang dihasilkan dari string tersebut disertakan dengan format yang diberikan dalam parameter kedua, yaitu '9999999990.99'. Hasilnya adalah nilai numerik 5280450.89. Perintah tersebut dieksekusi pada tabel DUAL, yang merupakan tabel sistem yang umumnya digunakan untuk operasi temporer dalam database Oracle.

Dokumentasi:

```
-- No 7
SELECT TO_NUMBER(REPLACE('£5,280,450.89', '£', ''), '9999999990.99') AS Result FROM DUAL;
```

Query Result x Query Result 1 x Script Output x Query Result 2 x

SQL | All Rows Fetched: 1 in 0.004 seconds

RESULT
1 5280450.89

- No 8: Mengubah angka menjadi string dengan format mata uang lokal

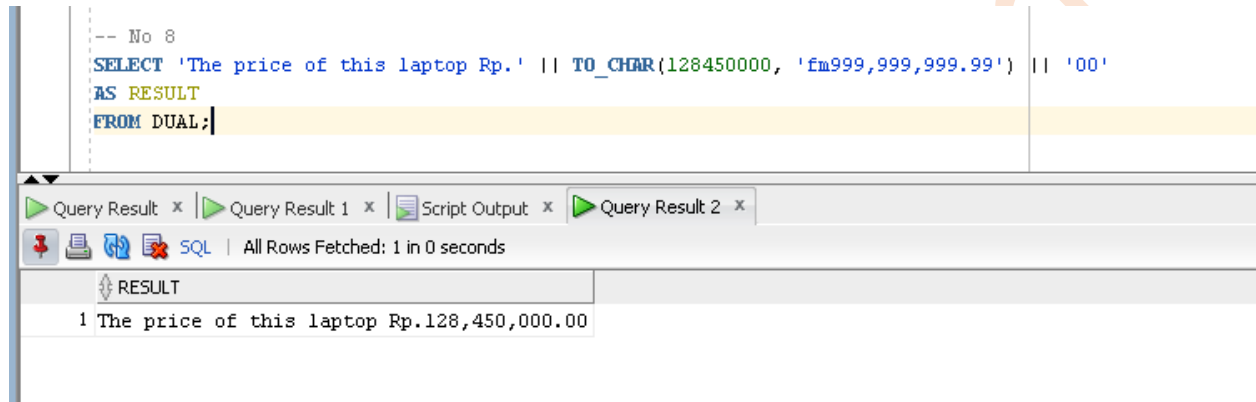
Query:

```
SELECT 'The price of this laptop Rp.' || TO_CHAR(128450000, 'fm999,999,999.99') || '00'
AS RESULT
FROM DUAL;
```

Penjelasan query:

Perintah ini menggabungkan beberapa elemen menjadi satu string menggunakan operator penggabungan ||. Elemen pertama adalah string 'The price of this laptop Rp.', diikuti oleh hasil konversi angka 128450000 menjadi string dengan format mata uang lokal menggunakan fungsi TO\_CHAR dengan format 'fm999,999,999.99', dan diakhiri dengan string '00'. Hasil akhirnya adalah string 'The price of this laptop Rp.128,450,000.00'. Perintah tersebut dieksekusi pada tabel DUAL, yang merupakan tabel sistem yang umumnya digunakan untuk operasi temporer dalam database Oracle.

Dokumentasi:



The screenshot shows the Oracle SQL Developer interface. The top pane contains the following SQL query:

```
-- No 8
SELECT 'The price of this laptop Rp.' || TO_CHAR(128450000, 'fm999,999,999.99') || '00'
AS RESULT
FROM DUAL;
```

The bottom pane shows the query result:

RESULT
1 The price of this laptop Rp.128,450,000.00

- No 9: Menampilkan tanggal sekarang dan minggu ke berapa dalam bulan ini

Query:

```
SELECT SYSDATE AS Current_Date, TO_NUMBER(TO_CHAR(SYSDATE, 'W')) AS
Week_Of_Month
FROM DUAL;
```

Penjelasan query:

Perintah ini menampilkan tanggal saat ini dengan menggunakan fungsi SYSDATE dan memberikan alias "Current\_Date" untuk hasilnya. Selain itu, fungsi TO\_CHAR digunakan untuk mendapatkan minggu ke dalam bulan ini dari tanggal saat ini, dan kemudian hasilnya dikonversi menjadi tipe data numerik menggunakan fungsi TO\_NUMBER. Hasilnya disertakan dengan alias "Week\_Of\_Month". Perintah tersebut dieksekusi pada tabel DUAL, yang merupakan tabel sistem yang umumnya digunakan untuk operasi temporer dalam database Oracle.

Dokumentasi:

-- No 9	
<pre>SELECT SYSDATE AS Current_Date, TO_NUMBER(TO_CHAR(SYSDATE, 'W')) AS Week_Of_Month FROM DUAL;</pre>	
Query Result x Query Result 1 x Script Output x Query Result 2 x Query Result 3 x	
All Rows Fetched: 1 in 0.002 seconds	
CURRENT_DATE	WEEK_OF_MONTH
1 23-MAR-24	4

- No 10: Membuat tabel TBL\_DATE, mengisi data dengan beberapa tanggal, menampilkan tanggal pertama dari setiap bulan, dan menghapus tabel

Query:

```
CREATE TABLE TBL_DATE (
  MYDATE DATE
);
```

```
INSERT INTO TBL_DATE VALUES('20-MAR-2024');
INSERT INTO TBL_DATE VALUES(DATE'2024-02-15');
INSERT INTO TBL_DATE VALUES('12-Jun-2024');
```

```
SELECT MYDATE, TRUNC(MYDATE, 'MM') AS First_Day
FROM TBL_DATE;
```

```
DROP TABLE TBL_DATE;
```

Penjelasan query:

- CREATE TABLE TBL\_DATE ( MYDATE DATE ): Membuat tabel baru bernama TBL\_DATE dengan satu kolom bernama MYDATE yang memiliki tipe data DATE. Kolom ini akan digunakan untuk menyimpan tanggal.
- INSERT INTO TBL\_DATE VALUES('20-MAR-2024');: Memasukkan tanggal '20-MAR-2024' ke dalam tabel TBL\_DATE.
- INSERT INTO TBL\_DATE VALUES(DATE'2024-02-15');: Memasukkan tanggal '2024-02-15' ke dalam tabel TBL\_DATE. Penggunaan DATE'...' adalah cara literal untuk menentukan nilai tanggal dalam SQL.
- INSERT INTO TBL\_DATE VALUES('12-Jun-2024');: Memasukkan tanggal '12-Jun-2024' ke dalam tabel TBL\_DATE.
- SELECT MYDATE, TRUNC(MYDATE, 'MM') AS First\_Day FROM TBL\_DATE;: Menampilkan semua data dari kolom MYDATE dalam tabel TBL\_DATE, serta menghitung tanggal pertama dari setiap bulan menggunakan fungsi TRUNC dengan argumen 'MM'. Hasilnya disertakan dengan alias "First\_Day".

→ DROP TABLE TBL\_DATE;; Menghapus tabel TBL\_DATE beserta semua data dan strukturnya dari database setelah selesai penggunaannya.

Dokumentasi:

```
-- No 10
CREATE TABLE TBL_DATE (
  MYDATE DATE
);
```

Task completed in 0.056 seconds

Table TBL\_DATE created.

```
CREATE TABLE TBL_DATE (
  MYDATE DATE
);

INSERT INTO TBL_DATE VALUES ('20-MAR-2024');
INSERT INTO TBL_DATE VALUES (DATE '2024-02-15');
INSERT INTO TBL_DATE VALUES ('12-Jun-2024');

SELECT MYDATE, TRUNC(MYDATE, 'MM') AS First_Day
FROM TBL_DATE;

DROP TABLE TBL_DATE;
```

Task completed in 0.158 seconds

Table TBL\_STRIIING dropped.

Table TBL\_DATE created.

1 row inserted.

1 row inserted.

1 row inserted.

```
SELECT MYDATE, TRUNC(MYDATE, 'MM') AS First_Day  
FROM TBL_DATE;
```

Query Result x Query Result 1 x Script Output x Query Result 2 x Query Result 3 x

SQL | All Rows Fetched: 3 in 0.006 seconds

	MYDATE	FIRST_DAY
1	20-MAR-24	01-MAR-24
2	15-FEB-24	01-FEB-24
3	12-JUN-24	01-JUN-24

```
DROP TABLE TBL_DATE;
```

Query Result x Query Result 1 x Script Output x

| Task completed in 0.073 seconds

Table TBL\_DATE dropped.