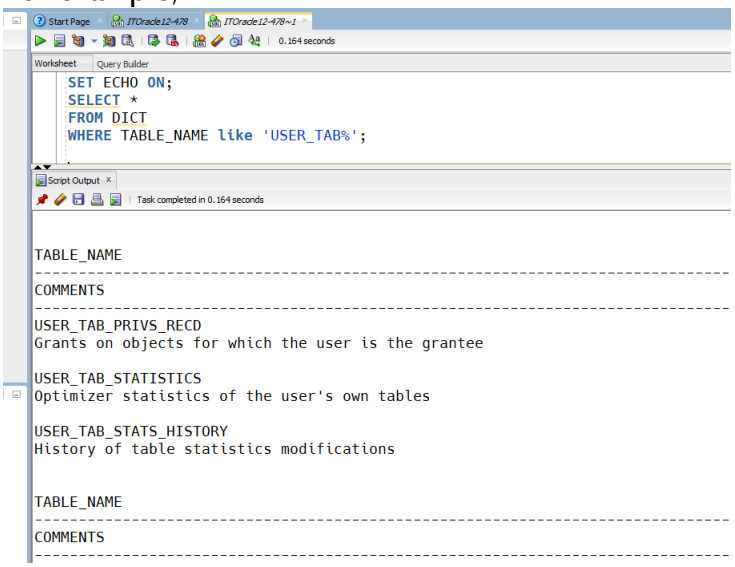
Lab #9 Assignment Tips

As some of you have noticed, a few of the queries are UGLY!

For example,



This ugly output is caused by some of the very large column sizes of the systems tables we are querying.

Let's look at the **DICT**ionary table

```
SQL> DESC dict;
Name        Null? Type
---------- ----- --------------
TABLE_NAME        VARCHAR2(128)
COMMENTS          VARCHAR2(4000)
```

We can use the **SHOW LINESIZE** to see that our line is much smaller than our data

SQL> SHOW LINESIZE;
linesize 1566

My LINESIZE is set to 1566. Your LINESIZE may ne even smaller. You can test it to find out.

So, what should we do to "de-uglify" our output.

Option 1: Nothing. Really, this is perfectly acceptable. The assignment is about exploring the USER_* and ALL_* tables, not about "prettifying" your output. BUT, I know many of you will reject this option.

Option 2: Format the Columns. This is an SQL*PLUS trick. But, it also works with **SQL Developer.** In the example below, I have formatted the **TABLE_NAME** and the **COMMENTS** columns to be 20 characters long (**A20**). NOTE: I needed a format command for each column I wanted to format, and that I **did not use semicolons** at the end of the lines.

```
COLUMN TABLE_NAME FORMAT A20
COLUMN COMMENTS FORMAT A20
SELECT *
FROM DICT
WHERE TABLE_NAME like 'USER_TAB%';
```

```
SQL> COLUMN TABLE_NAME FORMAT A20
SQL> COLUMN COMMENTS FORMAT A20
SQL> SELECT *
  2  FROM DICT
  3  WHERE TABLE_NAME like 'USER_TAB%';

TABLE_NAME           COMMENTS
-------------------- --------------------
USER_TABLES          Description of the u
                     ser's own relational
                      tables

USER_TABLESPACES     Description of acces
                     sible tablespaces

USER_TAB_COLS        Columns of user's ta
                     bles, views and clus
                     ters
```

But, the word wrapping in the **COMMENTS** column is a bit ugly.

I can fix this with the **WORD_WRAPPED** command. See below.

```
COLUMN TABLE_NAME FORMAT A20
COLUMN COMMENTS FORMAT A20 WORD_WRAPPED
SELECT *
FROM DICT
WHERE TABLE_NAME like 'USER_TAB%';
```

```
TABLE_NAME           COMMENTS
-------------------- --------------------
USER_TAB_PRIVS_RECD  Grants on objects
                     for which the user
                     is the grantee

USER_TAB_STATISTICS  Optimizer statistics
                     of the user's own
                     tables

USER_TAB_STATS_HISTO History of table
RY                   statistics
```

NOTE: The **CLEAR COLUMNS** command clears the above column formatting.

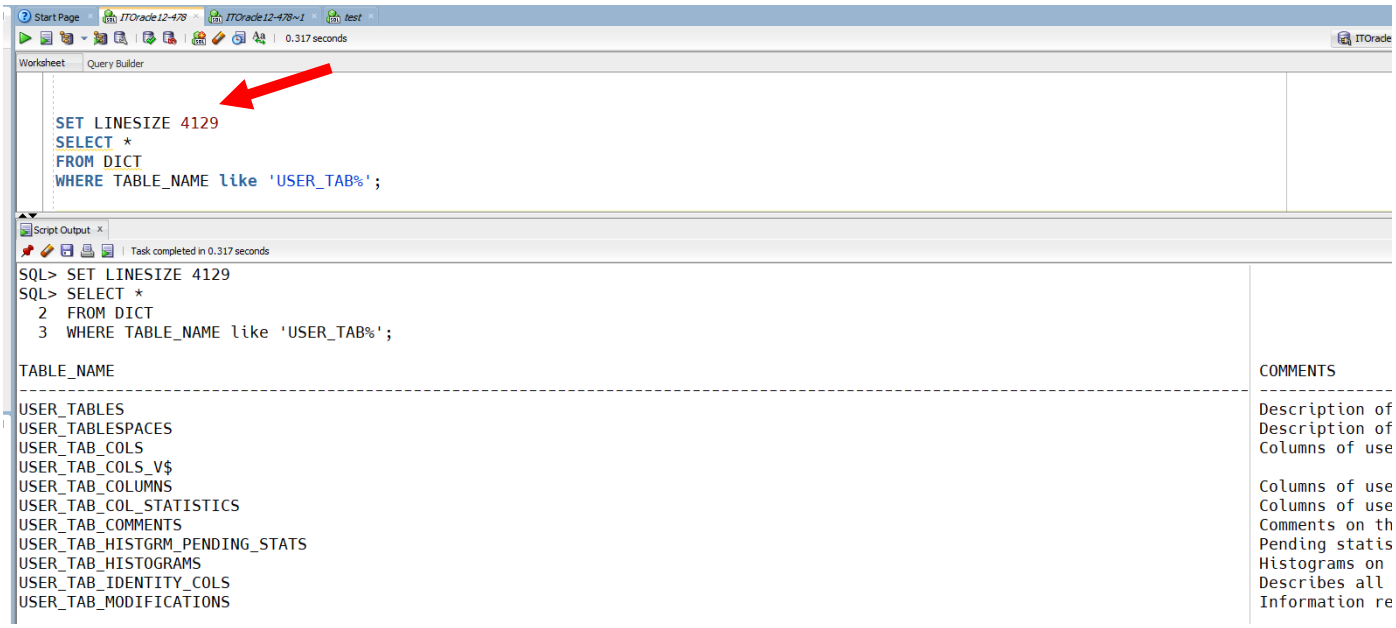
Option 3: Change the **LINESIZE**.

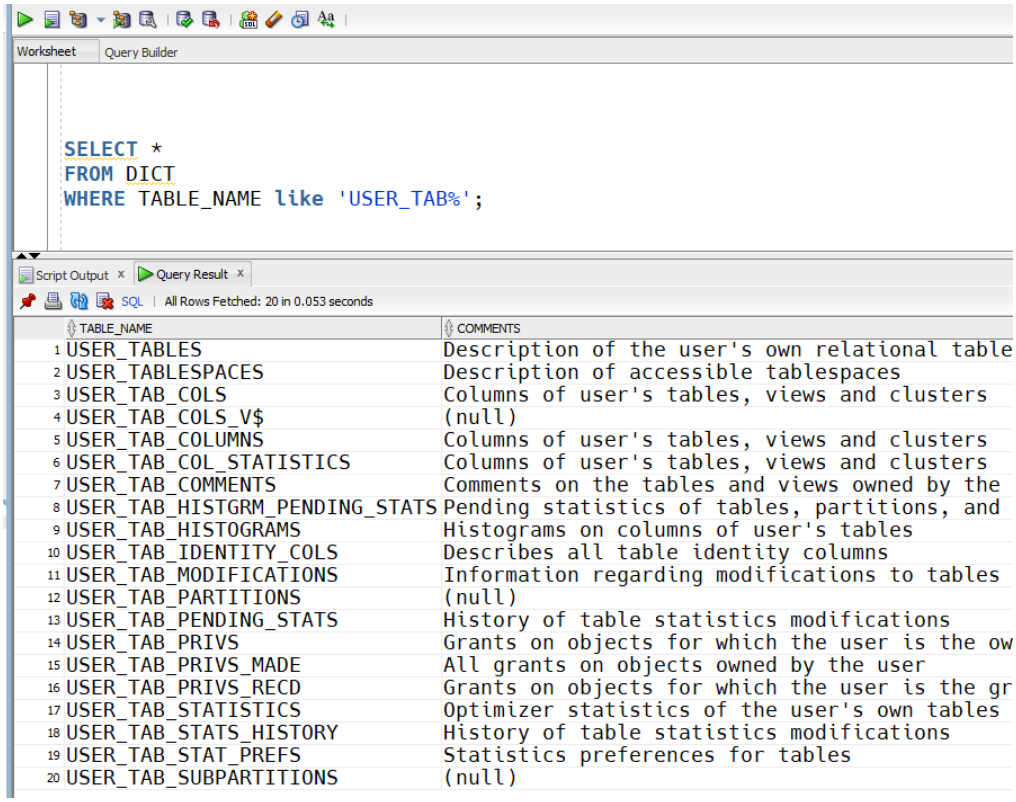
```
SQL> DESC dict;

Name        Null? Type
---------- ----- --------------
TABLE_NAME        VARCHAR2(128)
COMMENTS          VARCHAR2(4000)
```

128 + 1 + 4000 = 4129 (There is a pace between each column)

So, setting the **LINESIZE** solves the ugliness problem (sort of). Again, this line does not need a semicolon. See below.





Final note: If you do not get any results for the last question, it is probably because you didn't do the first question correctly.

Here is the text from question 1:

1. a. Use SQL to create the tables **`ch10_suppliers`** and **`ch10_subjects`**. Once created, the two tables should have the same column names, null? and type listed below. In addition, **`SUBJECT_ID`**, **`SUBJECT_NAME`** and **`SUPPLIER_ID`** should all be created with the **`UNIQUE`** constraint.

A few people have missed this highlighted part. If you add the **UNIQUE** constraints for the three columns. **You will get results for the final question.**

In Oracle 12c, users can create indexes with the CREATE INDEX command), but Oracle automatically creates indexes for **UNIQUE** and **PRIMARY KEY** constraints. I will be talking more about constraints and indexes in future lectures.