Lab 03: SQL the Oracle Way

Learning Outcomes

- TECHNICAL KNOWLEDGE –You will achieve deep technical knowledge and comprehension about database management systems. You will develop the ability to apply these technologies to solve information problems at the individual and organizational levels. After completing lab, you will be able to:

1. Create database tables, keys and constraints in Oracle 11g
2. Insert update, and delete data in tables
3. Use the select statement to query data.

- MANAGEMENT OF SOLUTION DEVELOPMENT - You will achieve a deep level of knowledge and comprehension of the disciplines used in the development of information system solutions. You will develop the ability to apply these disciplines to the solution of organizational and business problems. After completing this course, you will be able to:

1. Design, construct and maintain a database and various database objects using SQL language constructs

Specifically, after completing this lab, you will be able to:

- Explain the schema concept.
- Use Oracle SQL to create and manage database objects from a design specification.
- Understand the proper approach for using Oracle data types.

Important:

Please complete the previous lab prior to beginning this one.

Part I – Understanding Oracle SQL

In this part we'll dive into Oracle SQL, by creating tables to support a student roster. This should example should seem very familiar to you! ☺

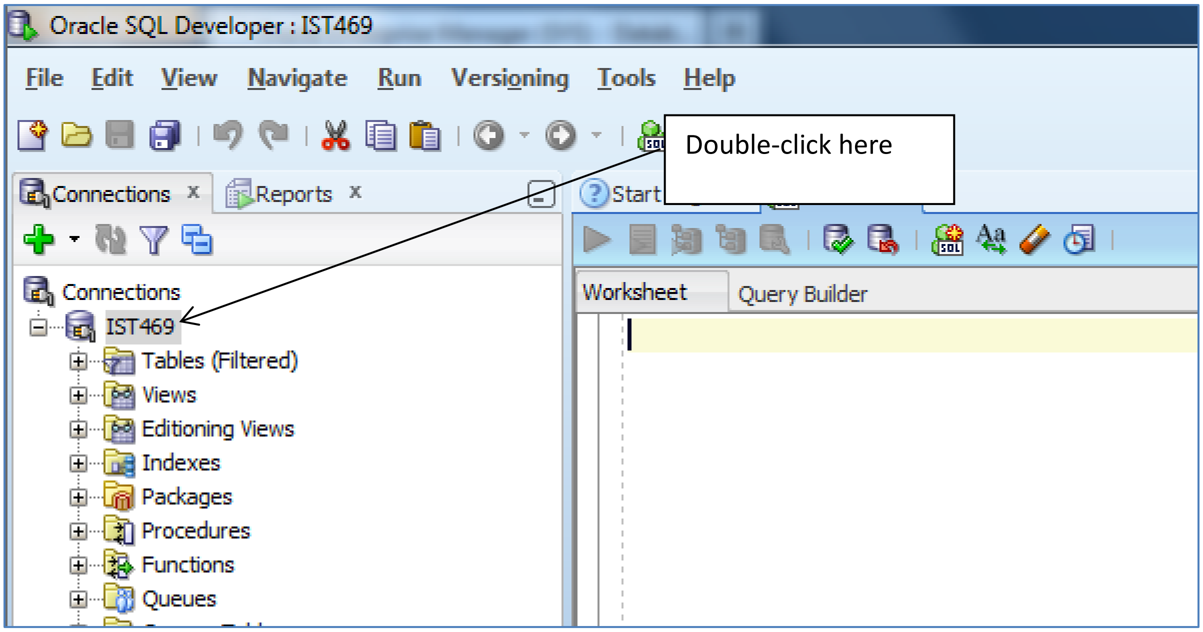
Step 1: Connect to Oracle as IST469

First we'll need to connect to out Oracle Database as IST469. Remember the username and password should be **IST469** / **SU2orange**.

Do This: Launch SQL Developer from your computer:

Once SQL Developer loads, open your IST469 Connection

Do This: Menu **File** → **new**

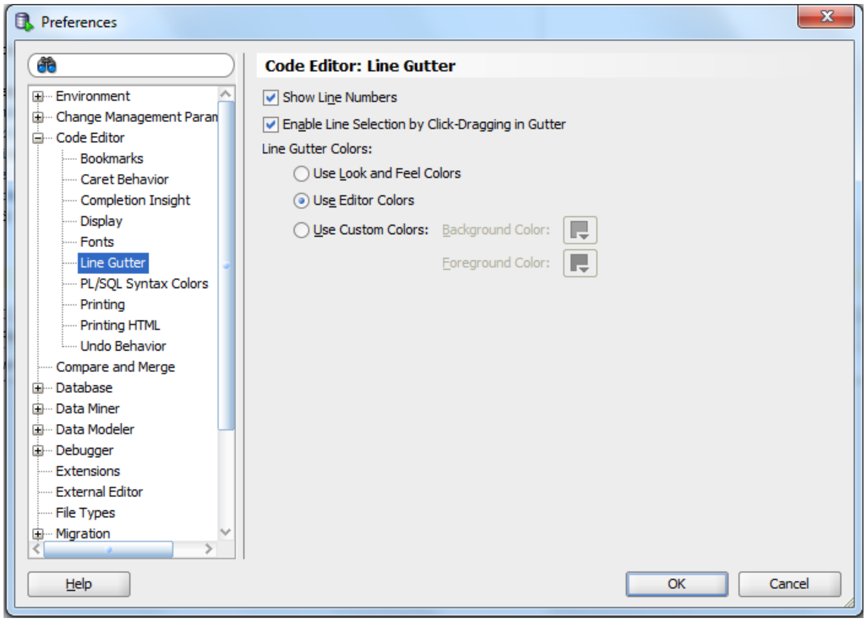


Double-click here

Finally let's turn on line numbers, for reference:

Do This: Menu **Tools → Preferences → Code Editor → Line Gutter →**

Check show line numbers as seen here:



Code Editor: Line Gutter

☑ Show Line Numbers

☑ Enable Line Selection by Click-Dragging in Gutter

Line Gutter Colors:

○ Use Look and Feel Colors

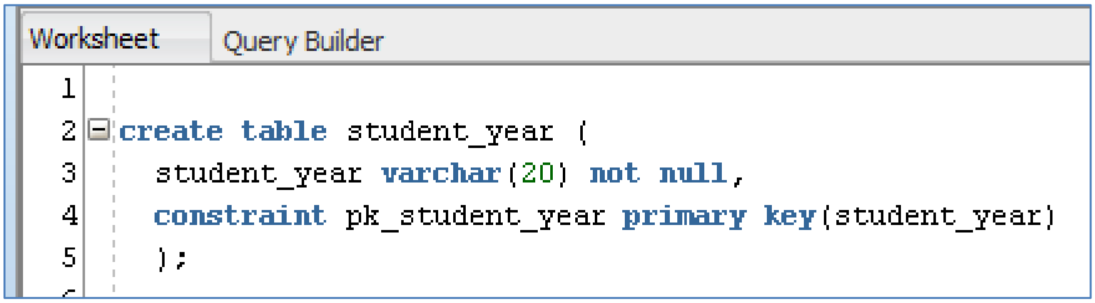
● Use Editor Colors

○ Use Custom Colors: Background Color: Foreground Color:

Step 2: SQL Developer – Understanding the Basics

Now that we're connected, we're going to learn our way around using SQL developer. We will learn how to execute code, save queries, fix errors and verify our code works.

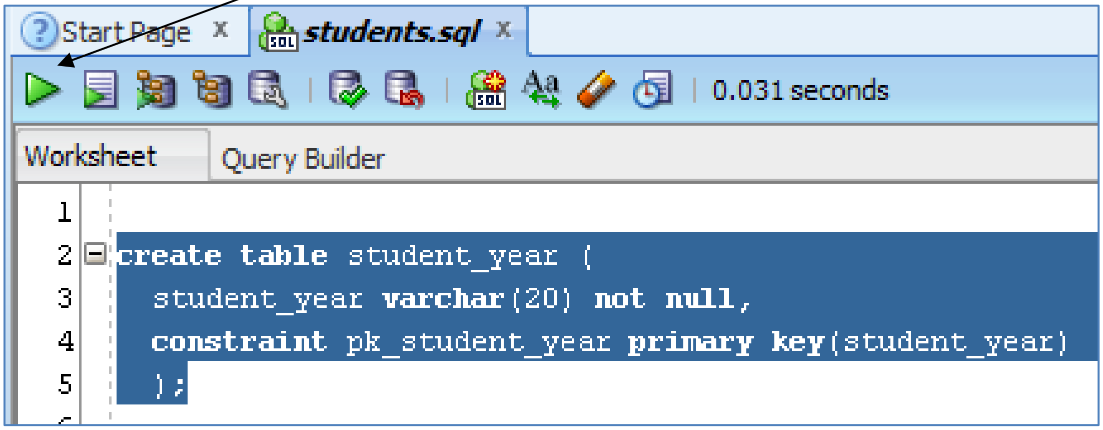
First let's create the lookup table to store the **student years** (Freshman, Sophomore, Junior, Senior, etc…)
Do This: Type the following create table command:



```

create table student_year (
   student_year varchar(20) not null,
   constraint pk_student_year primary key(student_year)
   );
```

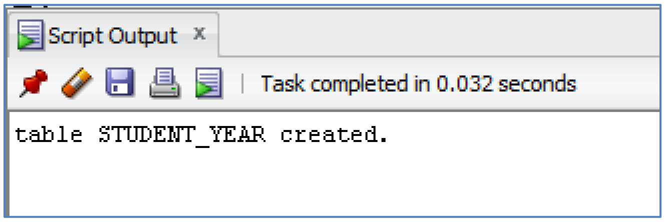
To execute this code, and thus create the table, you should **highlight** the code you wrote and then click the **run statement** button ▶ (or press **Ctrl+Enter**). **From this point forward, I will call this "Highlight and Execute"**

Do This: Highlight and execute the code as shown:



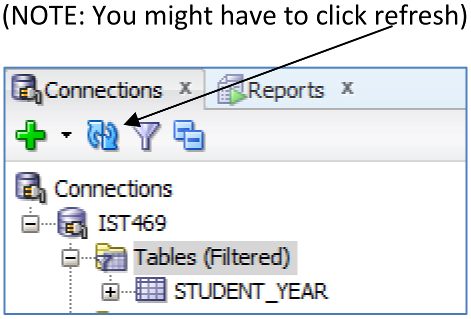
```
create table student_year (
  student_year varchar(20) not null,
  constraint pk_student_year primary key(student_year)
  );
```

If you do this successfully, you will see the following in your script output area:



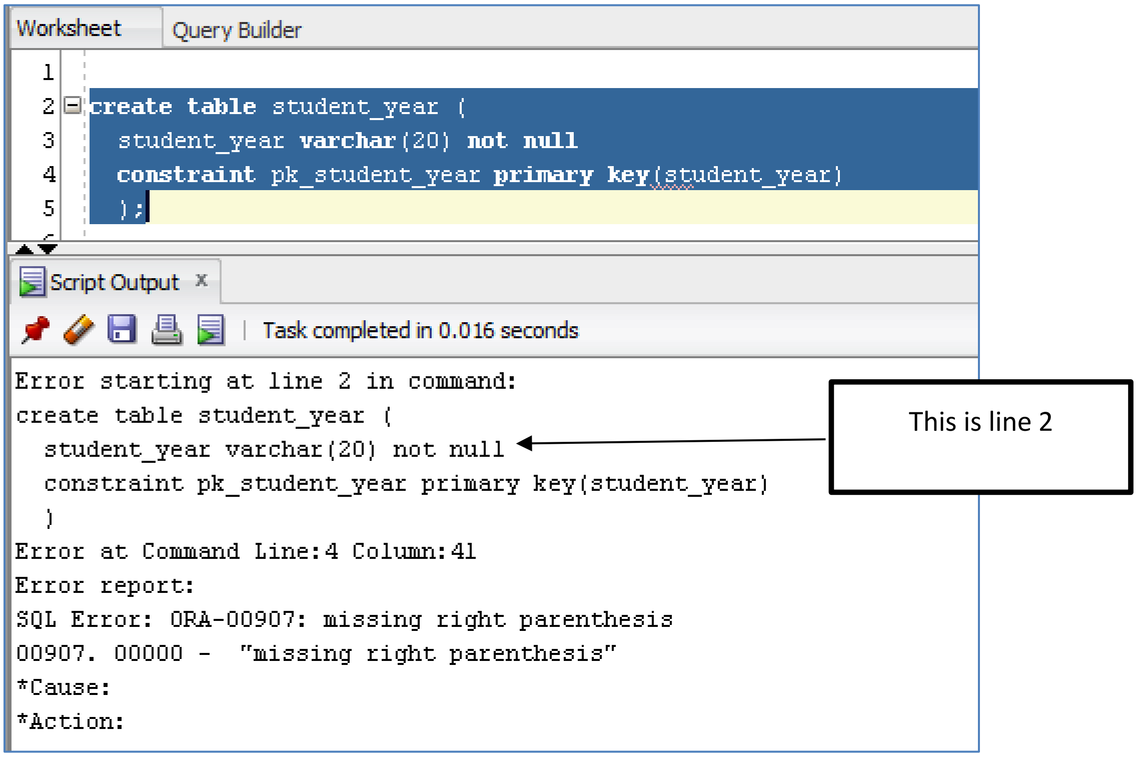
```
table STUDENT_YEAR created.
```

And you should see STUDENT_YEAR listed under Tables section the connection tab:
(NOTE: You might have to click refresh)



But What If I have a Syntax Error?

If you have a syntax error, make sure you fix it and try to execute your code again. Here's an example of my code execution with a syntax error. Can you find the mistake I made?



```
1
2 create table student_year (
3   student_year varchar(20) not null
4   constraint pk_student_year primary key(student_year)
5   );
```

Task completed in 0.016 seconds

```
Error starting at line 2 in command:
create table student_year (
  student_year varchar(20) not null
  constraint pk_student_year primary key(student_year)
  )
Error at Command Line:4 Column:41
Error report:
SQL Error: ORA-00907: missing right parenthesis
00907. 00000 -  "missing right parenthesis"
*Cause:
*Action:
```

Oracle's parser is not the most user-friendly and it will take you some time to "understand" exactly where the error is.

But What If I need to execute the create table again?

You might recall, you can't re-create a table. You need to drop the table before you can create it again.

Saving

If you're like me, then you probably don't like to re-type anything. It's a good idea to save our code so we won't need to re-key it to execute it later.

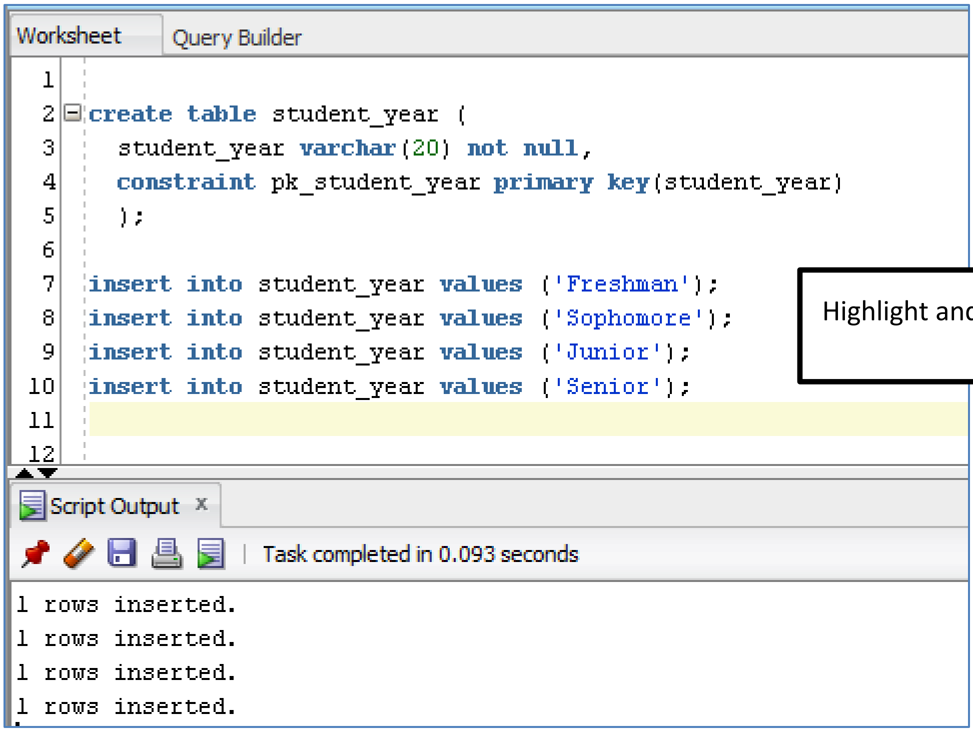
<u>**Do This:**</u> Menu → File → Save → type **students.sql** for the file name. From this point on, use your best judgment and save your code when you when you think it makes sense to do so.

Step 3: In Oracle *Everything* is a Database Transaction

The title says it all. None of your INSERTS, UPDATES or DELETES persist in the database until you execute a COMMIT statement. You can explicitly “undo” your data changes with a ROLLBACK statement. If you have uncommitted changes when your Oracle database restarts you will lose them.

Why not. Let’s give that a try. ☺

Do This: Type the following code into your **students.sql** file, save the file, and then highlight and execute your inserts.



```
1
2  create table student_year (
3    student_year varchar(20) not null,
4    constraint pk_student_year primary key(student_year)
5    );
6
7  insert into student_year values ('Freshman');
8  insert into student_year values ('Sophomore');
9  insert into student_year values ('Junior');
10 insert into student_year values ('Senior');
11
12
```

Task completed in 0.093 seconds

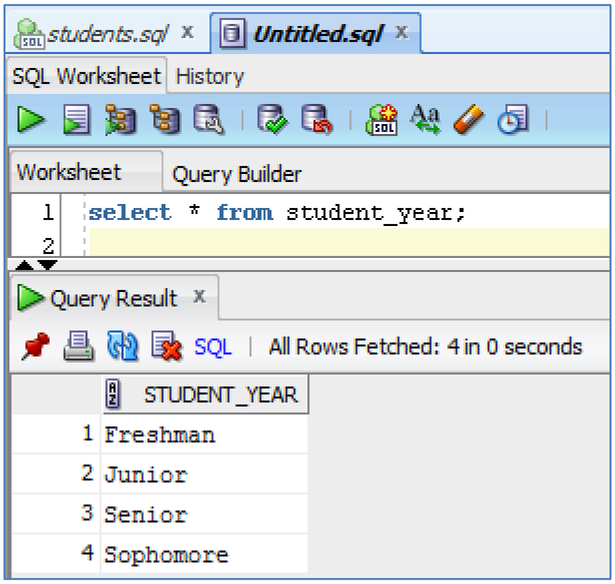
```
1 rows inserted.
1 rows inserted.
1 rows inserted.
1 rows inserted.
```

Highlight and execute lines 7-10

Next, create a new SQL file.

Do This: Menu → File → New → SQL File → OK → Untitled.sql → OK

Switch to your **untitled.sql** tab then highlight and execute this command to see the data in the table.

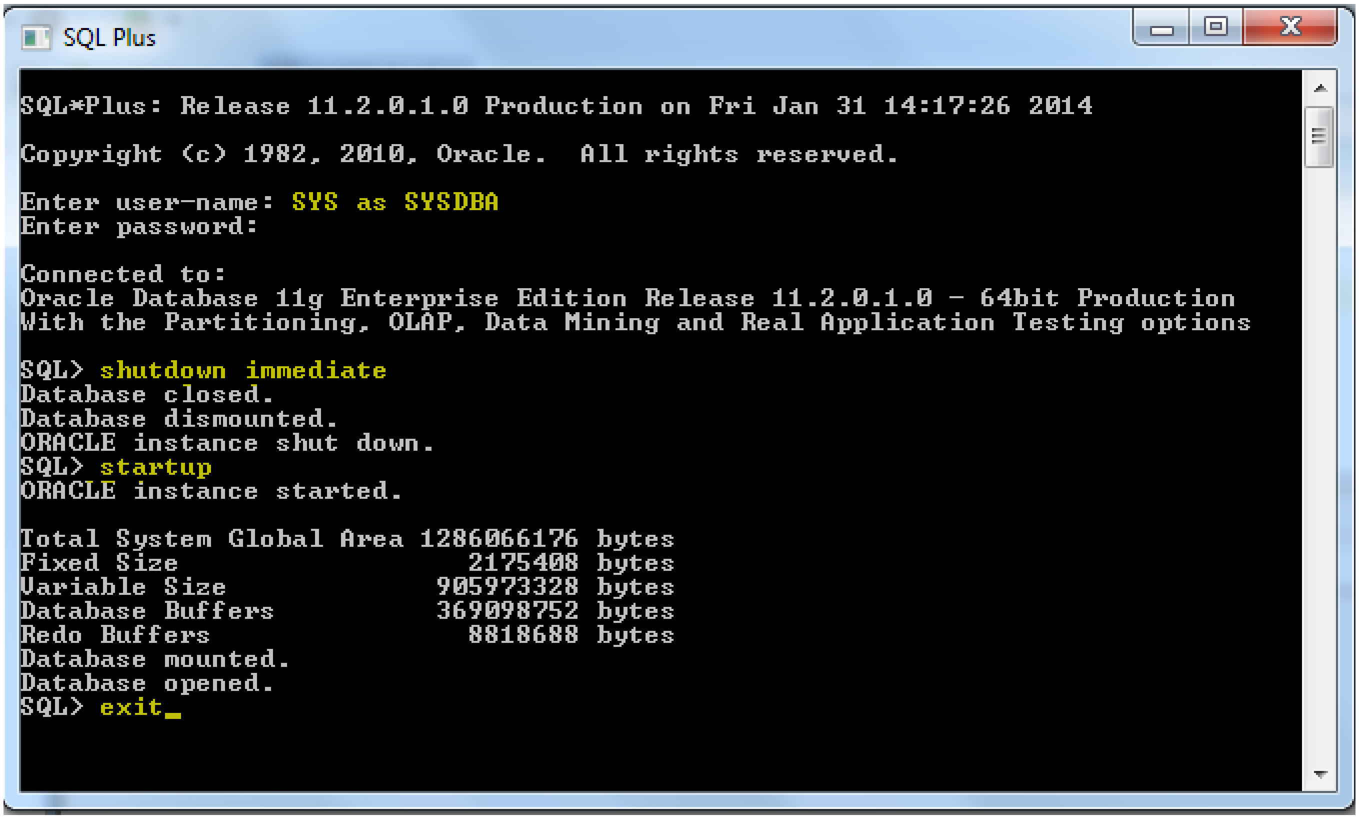


```
select * from student_year;
```

	STUDENT_YEAR
1	Freshman
2	Junior
3	Senior
4	Sophomore

Next, let's prove these inserts ***are not permanent*** by restarting our database instance!

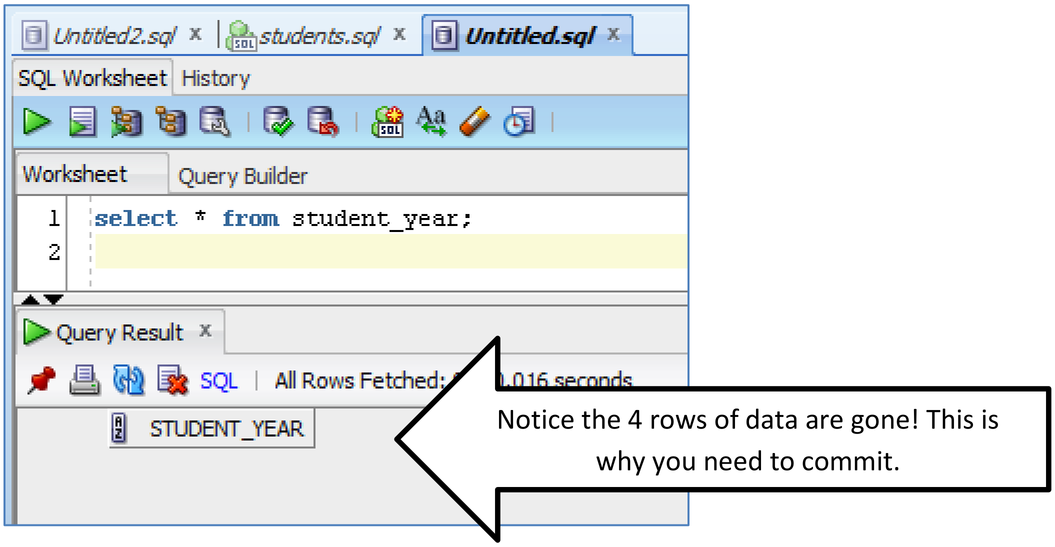
Do This: Start SQL Plus, then enter the following commands (highlighted in yellow) to restart your database:



```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Jan 31 14:17:26 2014

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: SYS as SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1286066176 bytes
Fixed Size                  2175408 bytes
Variable Size             905973328 bytes
Database Buffers          369098752 bytes
Redo Buffers                8818688 bytes
Database mounted.
Database opened.
SQL> exit_
```

Next go back to SQL Developer. Because we just restarted the database, you'll have to re-connect:

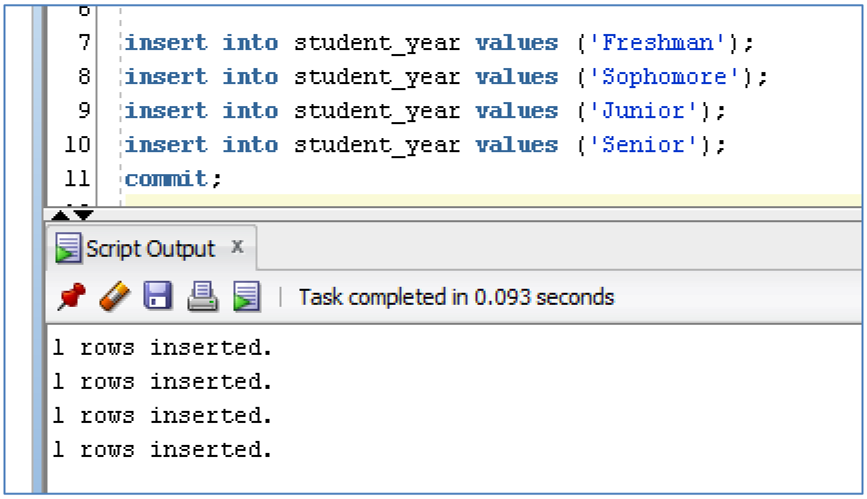
<u>Do This:</u> Right-Click on the **IST469** connection, and select **Reconnect**. When you execute the query again you'll see the inserts are gone:



```
select * from student_year;
```

Notice the 4 rows of data are gone! This is why you need to commit.

Now go back to your **students.sql** script and add a commit line after the inserts. Save and execute lines 7-11 to perform the inserts and then commit the transaction.



```
 7  insert into student_year values ('Freshman');
 8  insert into student_year values ('Sophomore');
 9  insert into student_year values ('Junior');
10  insert into student_year values ('Senior');
11  commit;
```

```
1 rows inserted.
1 rows inserted.
1 rows inserted.
1 rows inserted.
```

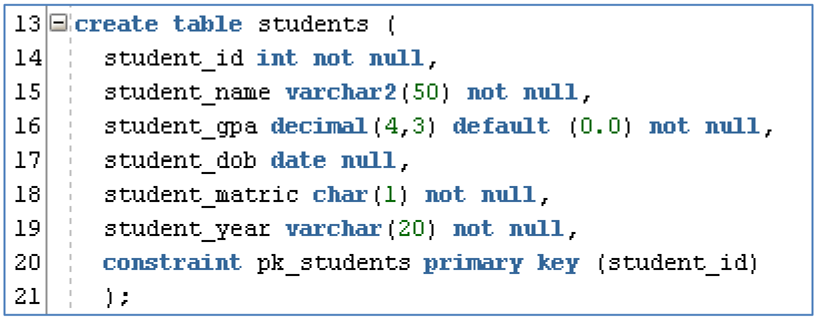
Step 3: Sequences, Dates, and Constraints

Now we will create the students table. This table will have a variety of constraints to ensure valid data is entered.

Let's make the table.

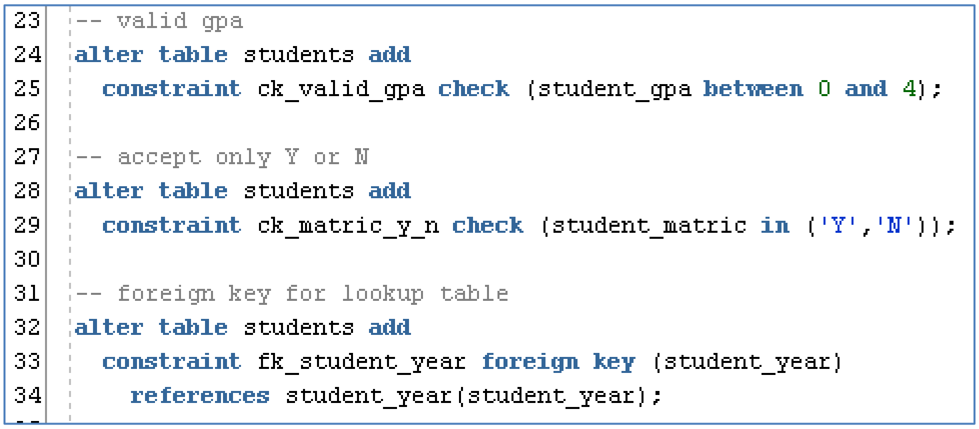
Making the Students Table

Do This: Inside **students.sql** type, highlight and execute this create table statement:



```
create table students (
  student_id int not null,
  student_name varchar2(50) not null,
  student_gpa decimal(4,3) default (0.0) not null,
  student_dob date null,
  student_matric char(1) not null,
  student_year varchar(20) not null,
  constraint pk_students primary key (student_id)
  );
```

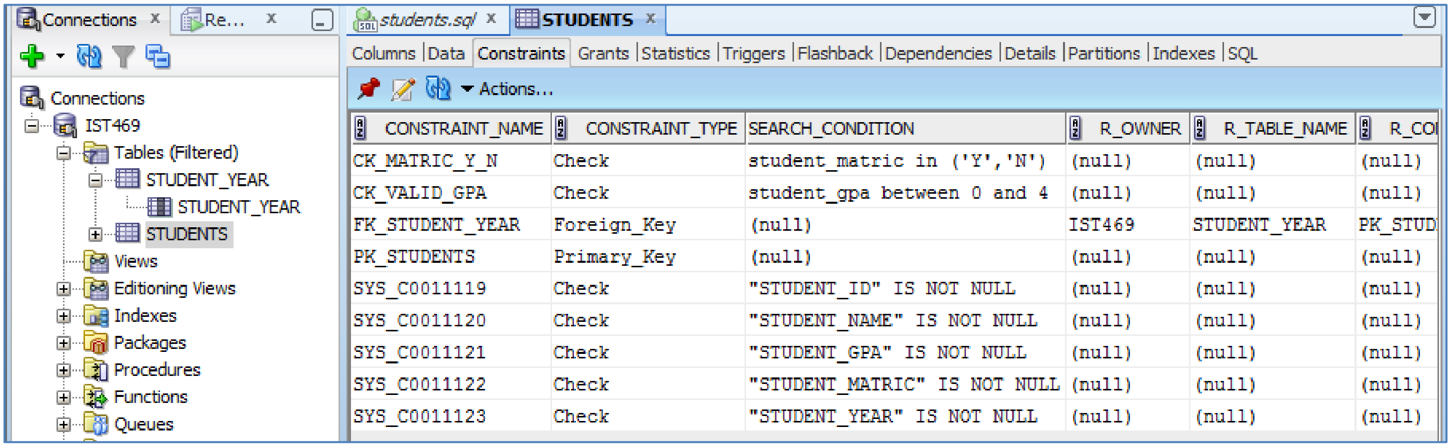
After you create the table, let's add some check constraints. Again type, highlight and execute each of these:



```
-- valid gpa
alter table students add
  constraint ck_valid_gpa check (student_gpa between 0 and 4);

-- accept only Y or N
alter table students add
  constraint ck_matric_y_n check (student_matric in ('Y','N'));

-- foreign key for lookup table
alter table students add
  constraint fk_student_year foreign key (student_year)
    references student_year(student_year);
```

Let's verify your constraints are here:

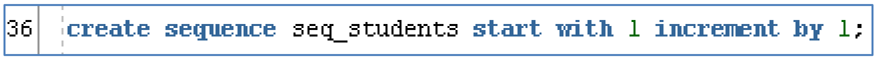
Do This: Under Connections → IST469 → Tables -> Click STUDENTS → Click on Constraints tab. You should see:



CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CO
CK_MATRIC_Y_N	Check	student_matric in ('Y','N')	(null)	(null)	(null)
CK_VALID_GPA	Check	student_gpa between 0 and 4	(null)	(null)	(null)
FK_STUDENT_YEAR	Foreign_Key	(null)	IST469	STUDENT_YEAR	PK_STUD
PK_STUDENTS	Primary_Key	(null)	(null)	(null)	(null)
SYS_C0011119	Check	"STUDENT_ID" IS NOT NULL	(null)	(null)	(null)
SYS_C0011120	Check	"STUDENT_NAME" IS NOT NULL	(null)	(null)	(null)
SYS_C0011121	Check	"STUDENT_GPA" IS NOT NULL	(null)	(null)	(null)
SYS_C0011122	Check	"STUDENT_MATRIC" IS NOT NULL	(null)	(null)	(null)
SYS_C0011123	Check	"STUDENT_YEAR" IS NOT NULL	(null)	(null)	(null)

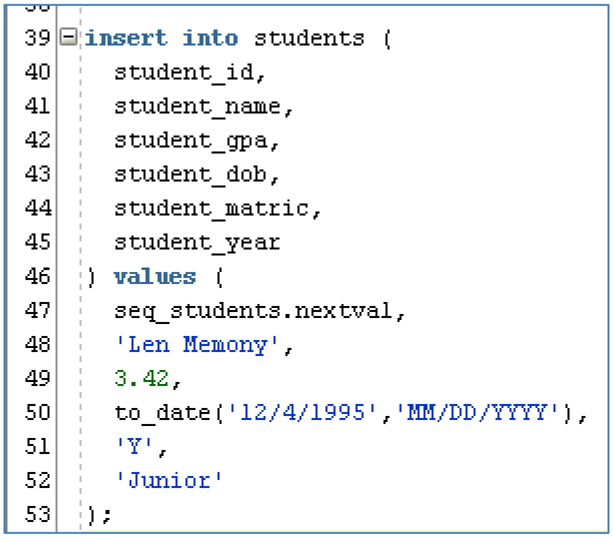
Inserting using, sequences and dates.

Next, let’s make a sequence to insert surrogate key value, **Highlight and Execute This:**



```
create sequence seq_students start with 1 increment by 1;
```

And now let’s insert a student **Highlight and Execute This:**



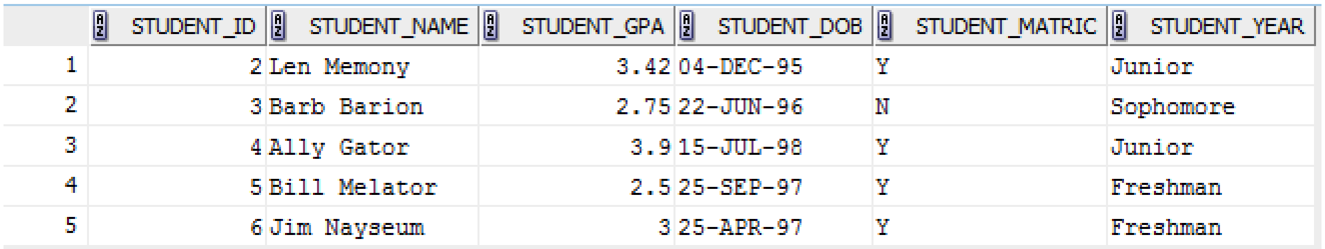
```
insert into students (
    student_id,
    student_name,
    student_gpa,
    student_dob,
    student_matric,
    student_year
) values (
    seq_students.nextval,
    'Len Memony',
    3.42,
    to_date('12/4/1995','MM/DD/YYYY'),
    'Y',
    'Junior'
);
```

Part II – On Your Own

In this part of the lab, you will take what you've learned and create various SQL structures in Oracle to extend our "Students" schema:

How to Hand in: Paste your answers into a MS Word document and then upload into Blackboard.

1. Write a Series of SQL INSERT statements to add students to the table. Make sure you include a commit statement to save them permanently. Don't worry about matching my STUDENT_ID value, obviously.



	STUDENT_ID	STUDENT_NAME	STUDENT_GPA	STUDENT_DOB	STUDENT_MATRIC	STUDENT_YEAR
1	2	Len Memony	3.42	04-DEC-95	Y	Junior
2	3	Barb Barion	2.75	22-JUN-96	N	Sophomore
3	4	Ally Gator	3.9	15-JUL-98	Y	Junior
4	5	Bill Melator	2.5	25-SEP-97	Y	Freshman
5	6	Jim Nayseum	3	25-APR-97	Y	Freshman

2. Write SQL to create the following table **majors** with the following columns and constraints. Implement the constraints as you see fit.

Column Name	Data Type	Constraints
major_id	Surrogate key	None
major_name	No more than 50 characters	Required.
major_school	No more than 50 characters	One of the following 'Arts & Sciences', 'iSchool', 'Whitman' or 'Newhouse'

3. Write SQL to insert at least 5 majors into the table. Make them up. ;-)
4. Write SQL to create a bridge table **student_majors** to satisfy the M-M relationship between students and majors. Remember a bridge table should have a two column primary key and each column should be a FK back to the outer table.
5. Write SQL to insert at least 5 rows into the **student_majors** table