ASSIGNMENT 5
Create, Alter

In this lab you will use SQL statements that fall in both the DDL and DML category. In the previous labs you were retrieving information that was already stored in the database. In this lab you will be storing new information in the database.

You must execute the statements in the order in which the questions are being asked.

Suggestions:

1) Do not create a spool file. This lab will probably take several days. Since you cannot guarantee that the work that you did on my home computer or the lab computers on campus will be there the next time you open up the SQLPlus session, I would make the following suggestion: Store all your SQL statements in a text file. Then you can just copy and paste your SQL statements into the SQLPlus session and get back to where you left off.

2) I would also suggest that you drop all your tables in the beginning of the text file just in case the tables are still there so that you don't get any error messages

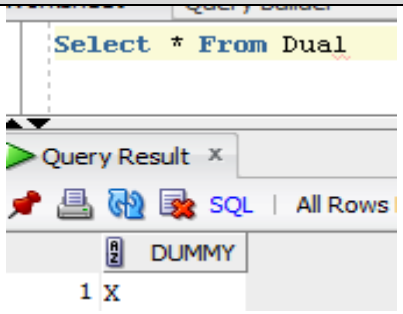
All the tables that you create should be prefixed with the first five letters of your lastname such as **sabze_patient**

What to turn in:

1) You will turn in this word document only. I do not want any other files
2) Paste a printscreen of either the **SQLPlus session** or **SQL Developer** showing only the SQL command and the results from the database engine. Some of the SQL statements that you issue may cause an error and may actually be the expected result. Do not assume that just because you are not getting an error message, everything is okay.
3) When typing in your SQL statements, make sure that the keywords are all in uppercase. The identifiers that you come up with such as table names, column names or constraint names should all be in lower case.
4) Make sure that you prefix your table names with the first five letters of your last name.
5) Make sure that you only provide a printscreen of the snippet that pertains to the question (NOTHING MORE).

Suggestion: you can use the snipping tool in windows 7 or you can download this open source program http://getgreenshot.org/ for printscreens. Provide only the printscreen that pertains to the question. **I do not want to see your trial and errors or things that pertain to other questions.**

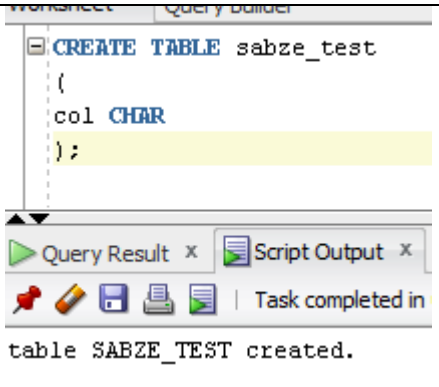
SQLPlus or SQLDeveloper (Your choice)

Example	Display the contents of the dual table
	`SQL> SELECT * FROM dual;` `D` `-` `X` `1 row selected.` OR 
Next Example	Create a table called test

```
SQL> CREATE TABLE sabze_test
  2  (
  3  col    CHAR
  4  );

Table created.
```

OR



All the tables that you create must be prefixed with the first five letters of your last name such as sabze_student.

1A	Create a **student** table that will hold the following data. Make sure you assign a **primary key**. SSN text lname text fname text age numeric salary numeric dob numeric
	`DROP TABLE mendo_student;` `DROP TABLE mendo_class;` `CREATE TABLE mendo_student` `(` `ssn VARCHAR2(20),` `lname VARCHAR2(20),` `fname VARCHAR2(20),` `age NUMBER,` `salary NUMBER,` `dob NUMBER,` `CONSTRAINT mendo_student_ssn_pk PRIMARY KEY (ssn)` `);`
1B	**After** the table has been created add a **candidate key** based on lname and fname. **Note: Candidate and unique key are the same thing**
	`ALTER TABLE mendo_student ADD UNIQUE (fname, lname);`
1C	**After** the table has been created add a **check constraint** such that the age is greater than 10 but less than 50. Provide a name for the check constraint.
	`ALTER TABLE mendo_student MODIFY age CONSTRAINT mendo_student_age_ck` `CHECK (age<10 AND age>50);`
1D	**After** the table has been created **add a column** called address.
	`ALTER TABLE mendo_student ADD address VARCHAR2(20);`
1E	**After** the table has been created, modify the dob column to be of datatype **date** and also **not null**
	`ALTER TABLE mendo_student MODIFY dob DATE NOT NULL;`
1F	Create a composite **index on** ssn and dob
	`CREATE INDEX mendo_student_ssn_dob_idx ON mendo_student (ssn, dob);`
1G	After the table has been created **add a column** called **transferable** with a **not null** constraint. Do not assign a name to the constraint

	`` ALTER TABLE mendo_student ADD transferable CHAR; ALTER TABLE mendo_student MODIFY transferable CHAR NOT NULL; ```
1H	After the table has been created add a **check constraint** on the column t**ransferable** to allow only 'y','Y','n','N'. Give the constraint a name.
	`` ALTER TABLE mendo_student MODIFY transferable CONSTRAINT mendo_student_transferable_ck CHECK (transferable = 'y' or transferable = 'Y' or transferable = 'n' or transferable = 'N'); ```
1I	Drop the age column
	`` ALTER TABLE mendo_student DROP (age); ```

2A	Create a second table called **class** that will hold the following data. You decide what the data types are going to be. Class code Class description In the create table statement make **Class description the candidate key** and also make it is **not null. NOTE: Candidate and unique keys are the same thing**
	`` CREATE TABLE mendo_class (code NUMBER, description VARCHAR2(100) NOT NULL, CONSTRAINT mendo_class_description_uk UNIQUE (description)); ```
2B	**After** the table has been created add the **primary key**. Give the constraint a name
	`` ALTER TABLE mendo_class ADD PRIMARY KEY (code); ```
2C	Create an **index** on **class description**
	`` CREATE INDEX mendo_class_decription_idx ON mendo_class (description); ```

3A	Create a third table called **student_class.** This table is an association table that contains information on the different class that the students are taking. You figure out what the columns should be. It should contain only two columns.
	`` CREATE TABLE mendo_student_class(ssn NUMBER; code NUMBER;); ```
3B	**After** the table has been created add the **primary key** constraint (Name the constraint)
3C	**After** the table has been created add the **foreign key** constraint(**s**) (Name the constraint(s))