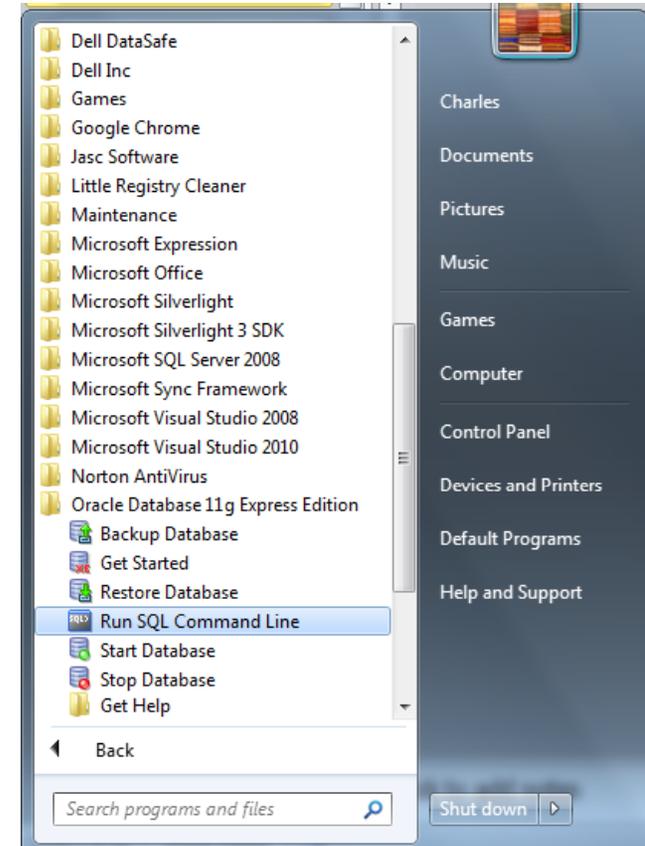


# Create a Table with SQL

October 29, 2012

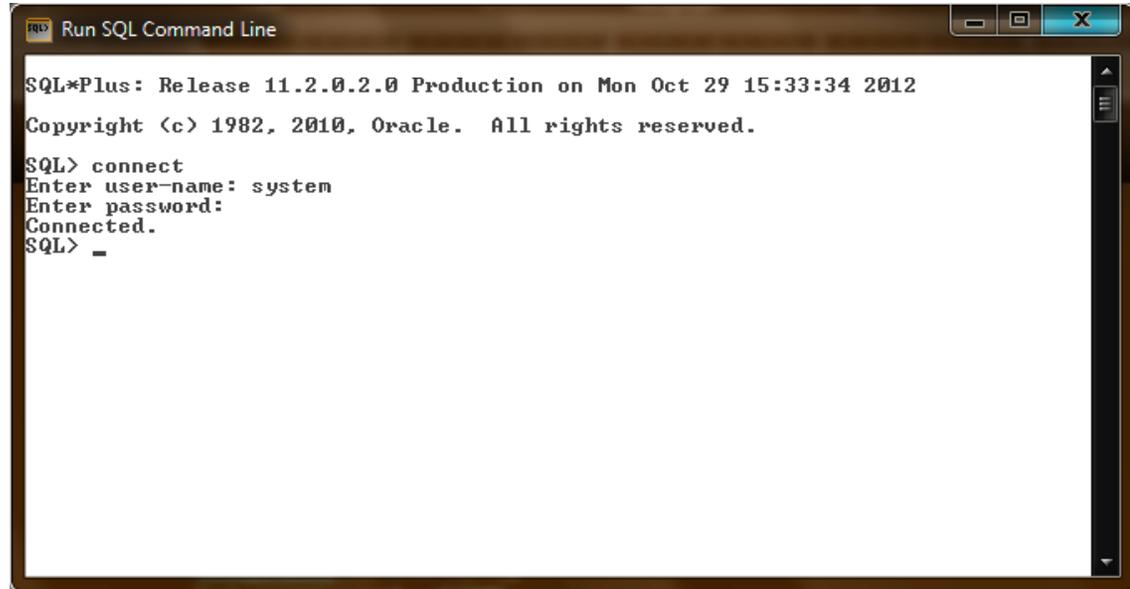
# Run SQL Command Line

We will begin this lesson by building a simple table. On the Start menu, select Programs, then Oracle Database Express Edition and Run SQL Command Line.



# Connect to Oracle

To connect to Oracle, we type connect when the Run SQL Command Line window appears. For the username, we input “system”. For our password, we use the same one we entered when we installed Oracle Database Express on our computer.

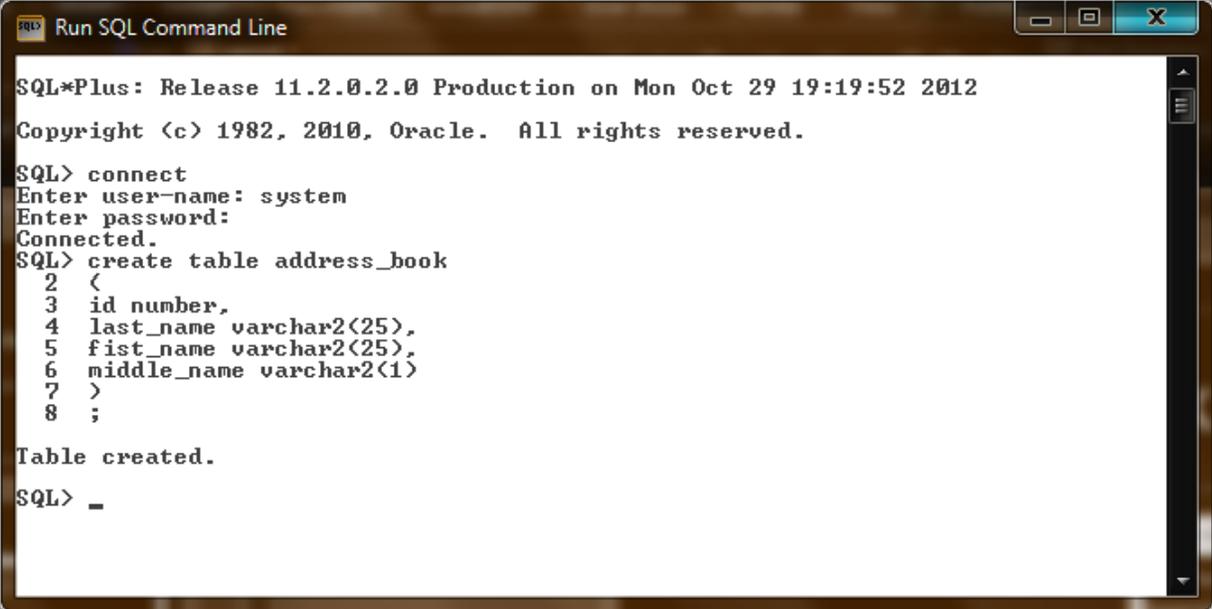
A screenshot of a Windows-style window titled "Run SQL Command Line". The window contains a text area with the following text: "SQL\*Plus: Release 11.2.0.2.0 Production on Mon Oct 29 15:33:34 2012", "Copyright (c) 1982, 2010, Oracle. All rights reserved.", "SQL> connect", "Enter user-name: system", "Enter password:", "Connected.", and "SQL> \_". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Mon Oct 29 15:33:34 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> _
```

# Create a Simple Table

To create a table, we use the expression *create table tablename* format.

Our tablename is address book. On line 2, we start with an open parenthesis. On line 3, we type the name of the first column and the data type.

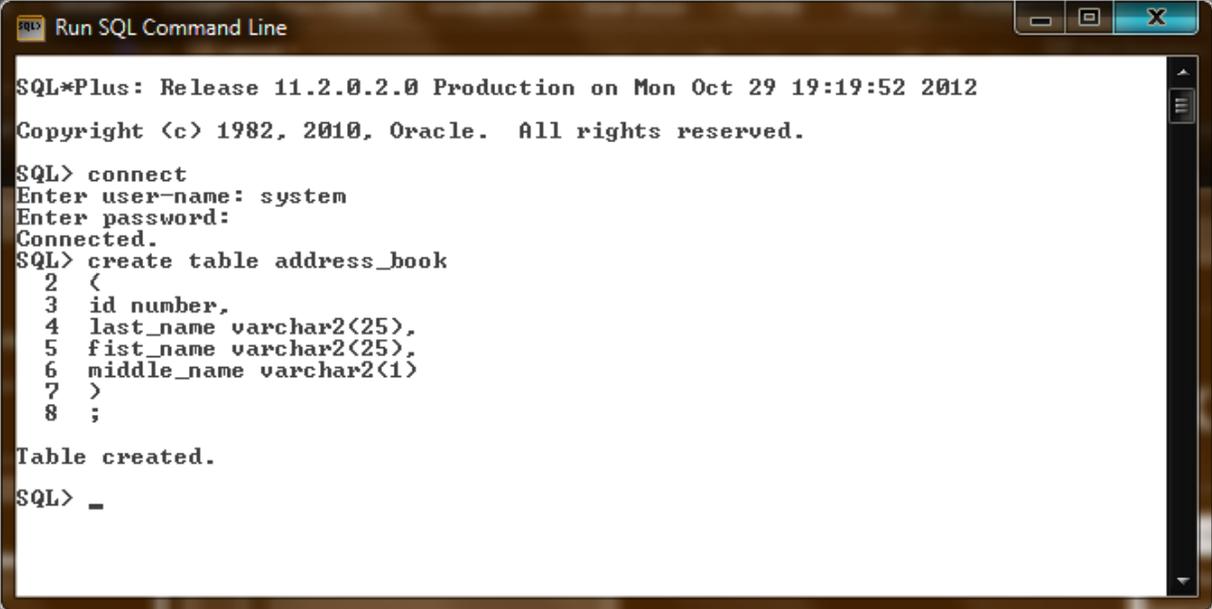


```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Mon Oct 29 19:19:52 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> create table address_book
2  (
3  id number,
4  last_name varchar2(25),
5  first_name varchar2(25),
6  middle_name varchar2(1)
7  )
8  ;
Table created.
SQL> _
```

The data type for the ID is number, which means that a numeric value can identify each person in the address book. We end the line with a comma.

# Adding Columns

The next column is last name (last\_name) which has the varchar2 data type. Variable characters can be numbers, letters and special characters. The 25 in parenthesis is the number of characters allowed in the last name. We end the line with a comma.



```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Mon Oct 29 19:19:52 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> create table address_book
 2 <
 3 id number,
 4 last_name varchar2(25),
 5 first_name varchar2(25),
 6 middle_name varchar2(1)
 7 >
 8 ;

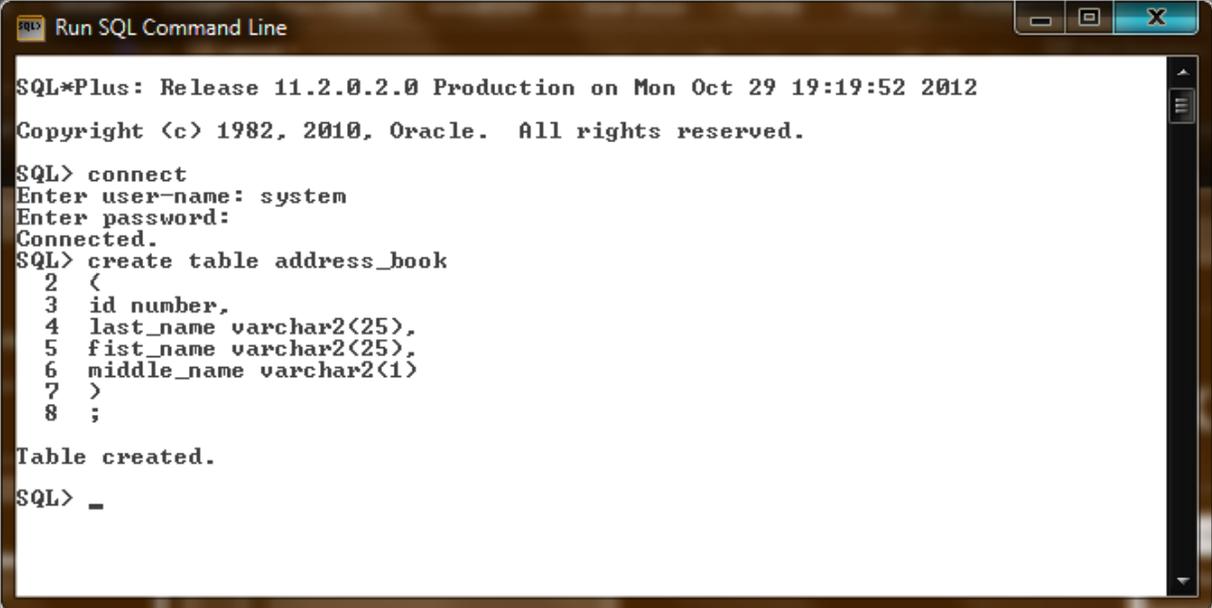
Table created.

SQL> _
```

The next column is first name (first\_name) which has the varchar2 data type. Again, the 25 in parenthesis is the number of characters allowed in the first name. We end that line with a comma.

# Adding More Columns

The next column is middle name (middle\_name) which has the varchar2 data type. The 1 in parenthesis is the number of characters allowed in the last name. We do not end this last line with a comma.

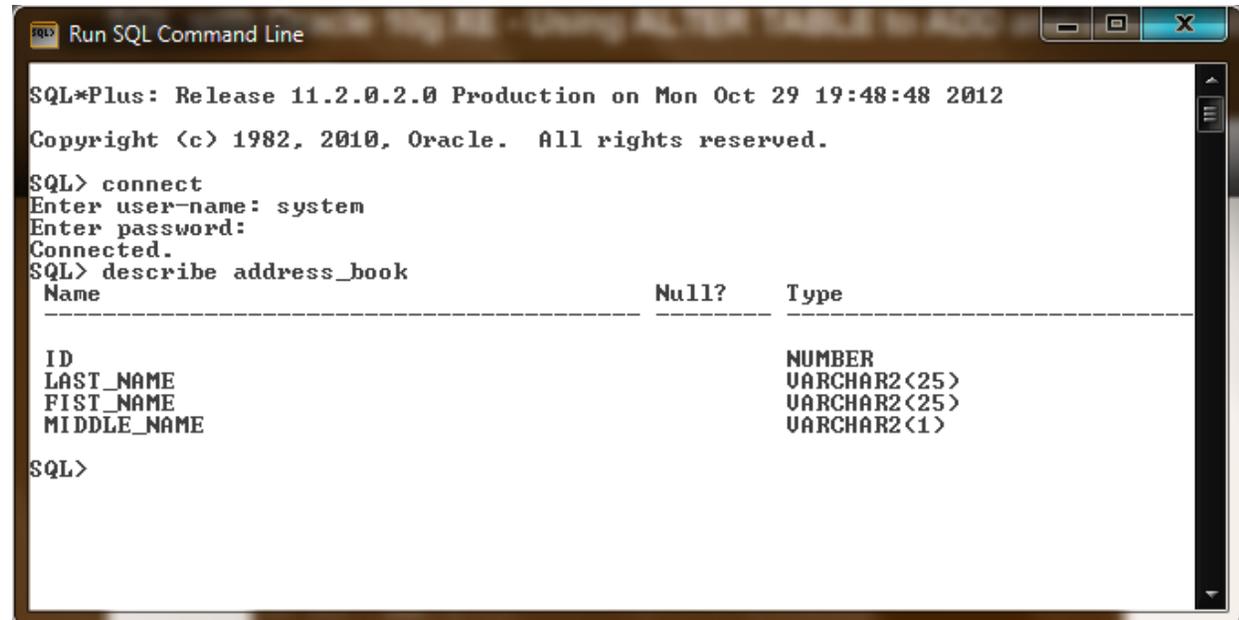


```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Mon Oct 29 19:19:52 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> create table address_book
2 <
3 id number,
4 last_name varchar2(25),
5 first_name varchar2(25),
6 middle_name varchar2(1)
7 >
8 ;
Table created.
SQL> _
```

We close the table with a close parenthesis and a semicolon. A “Table created” comment is returned.

# Describe the Table

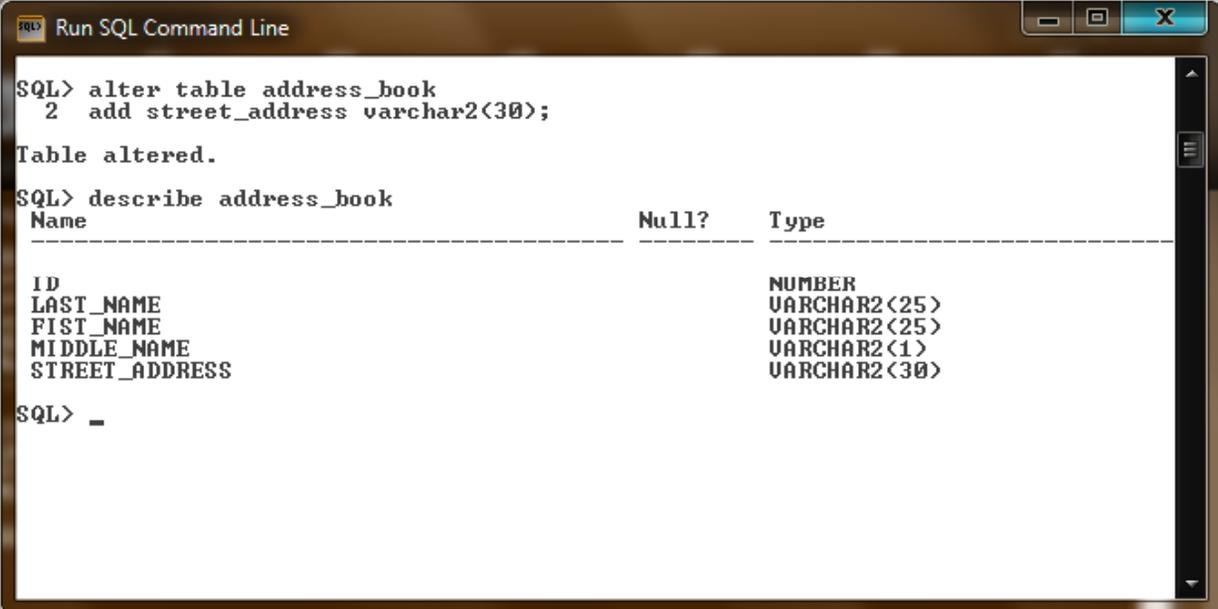
We use the Describe function to see our column names, null field and data types.



```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Mon Oct 29 19:48:48 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> describe address_book
Name                                Null?    Type
-----
ID                                    NUMBER
LAST_NAME                            VARCHAR2(25)
FIST_NAME                             VARCHAR2(25)
MIDDLE_NAME                           VARCHAR2(1)
SQL>
```

# Add Another Column to a Table

To add another column to a database table, we use the *alter table tablename* expression, so we input *alter table address\_book*. In the next, we will add a new column using the Add function.



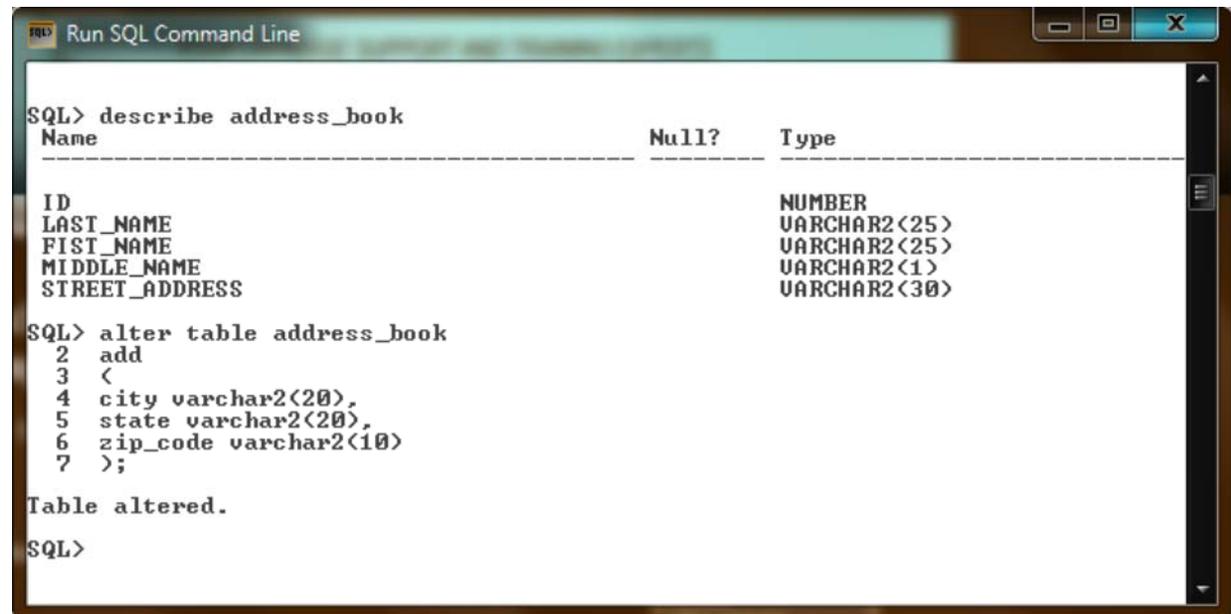
```
SQL> alter table address_book
2 add street_address varchar2(30);
Table altered.
SQL> describe address_book
Name                               Null?    Type
-----
ID                                  NUMBER
LAST_NAME                          VARCHAR2(25)
FIST_NAME                           VARCHAR2(25)
MIDDLE_NAME                          VARCHAR2(1)
STREET_ADDRESS                       VARCHAR2(30)
SQL> -
```

On line 2, we type `add street_address varchar2(30)`, and we end the expression with a semicolon.

To remove a column, we use the Alter function in the first line and the Drop function in the next line.

# Add Multiple Columns to a Table

When we want to add multiple columns to a database table, again we use the *alter table tablename* expression, so we input *alter table address\_book*. In the next, we will add a new column using the Add function.



```
SQL> describe address_book
Name                                     Null?      Type
-----
ID                                         NUMBER
LAST_NAME                                VARCHAR2(25)
FIRST_NAME                               VARCHAR2(25)
MIDDLE_NAME                              VARCHAR2(1)
STREET_ADDRESS                           VARCHAR2(30)

SQL> alter table address_book
2  add
3  (
4  city varchar2(20),
5  state varchar2(20),
6  zip_code varchar2(10)
7  );

Table altered.

SQL>
```

On line 2, we type add and on line 3, we input an open parenthesis. On line 4, we type city varchar2(20), and we end the expression with a comma.

On line 5, we type state varchar2(20), and we end the expression with a comma. On line 6, we type zip\_code varchar2(10), and we do not end the expression with a comma. We close the expression with a close parenthesis and a semicolon.

# Add More Columns

We need to add more columns to the address book table.

We need to add the following columns:

home\_phone varchar2(12)

business\_phone varchar2(12)

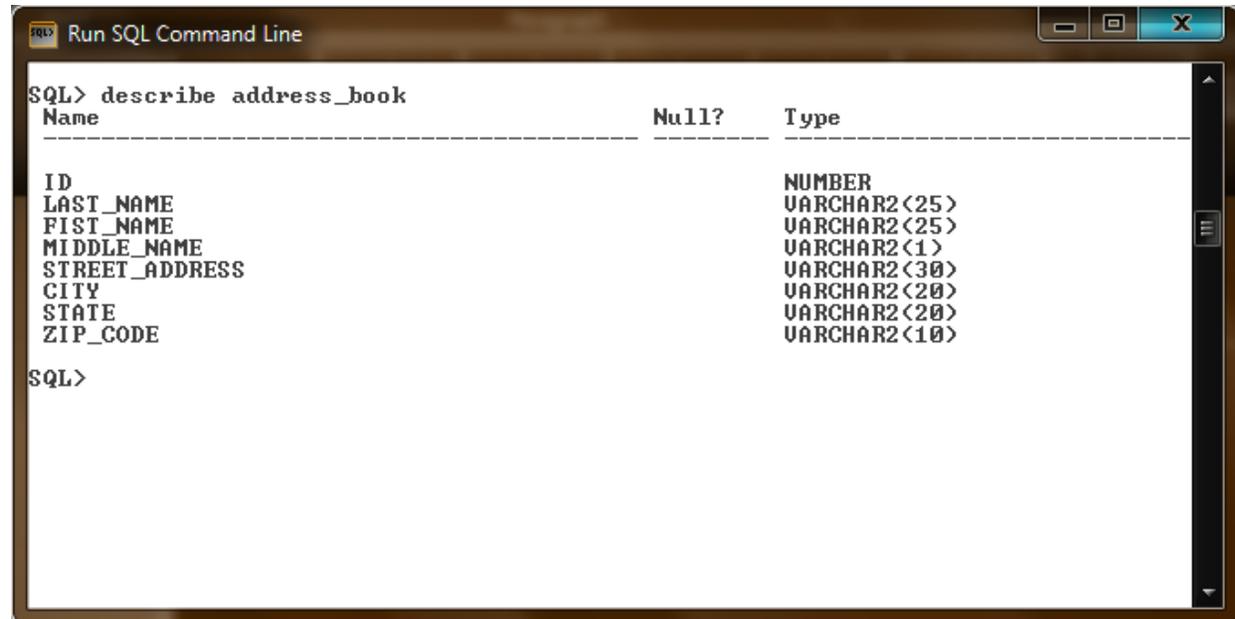
cell\_phone varchar2(12)

email1 varchar2(50)

email2 varchar2(50)

title varchar2(20)

organization varchar2(30)

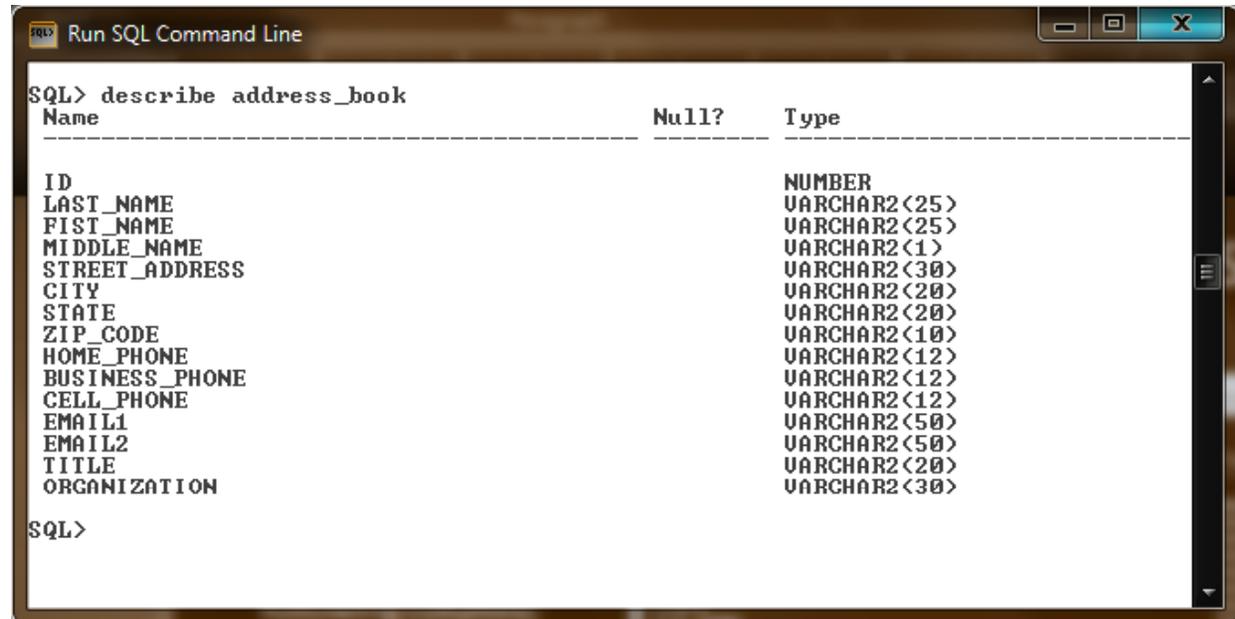


```
SQL> describe address_book
Name                               Null?    Type
-----
ID                                  NUMBER
LAST_NAME                          VARCHAR2(25)
FIRST_NAME                          VARCHAR2(25)
MIDDLE_NAME                         VARCHAR2(1)
STREET_ADDRESS                      VARCHAR2(30)
CITY                                 VARCHAR2(20)
STATE                                VARCHAR2(20)
ZIP_CODE                             VARCHAR2(10)
SQL>
```

Use the Alter and Add functions to modify the address book table.

# The Address Book Table

We run the describe address\_book expression to inspect the database table.



```
SQL> describe address_book
```

Name	Null?	Type
ID		NUMBER
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
MIDDLE_NAME		VARCHAR2(1)
STREET_ADDRESS		VARCHAR2(30)
CITY		VARCHAR2(20)
STATE		VARCHAR2(20)
ZIP_CODE		VARCHAR2(10)
HOME_PHONE		VARCHAR2(12)
BUSINESS_PHONE		VARCHAR2(12)
CELL_PHONE		VARCHAR2(12)
EMAIL1		VARCHAR2(50)
EMAIL2		VARCHAR2(50)
TITLE		VARCHAR2(20)
ORGANIZATION		VARCHAR2(30)

```
SQL>
```

# Additional Assignments

1. Make a table to record the information for a customer. The fields would enable the business to bill the customer and ship products to them.
2. Make a table to record the inventory of a computer lab. The fields would record specifics such processor, hard drive size, and the amount of RAM.
3. Make a table to record the information for a computer repair company. The columns will describe the customer's address, their computer, problem and fix.