**SQL**

Structured Query Language.

Database language designed for retrieval and management of data in relational database.

**Sql Commands :**

**DDL (Data Definition Language)**

Create ----🡪 ( Table,view,trigger,object)

Alter -----🡪 ( Modify existing database objects )

Drop ------🡪 (delete table, delete view , delete objects)

**DML (Data Manipulation Language)**

Select ----🡪 Record (Data Query Language)

Insert ----🡪 New Record

Update ---🡪 Modify Record

Delete ----🡪 Delete Record

**DCL (Data Control Language)**

Grant ---🡪 Give privilege (permission)

Revoke --🡪 Take back privilege (permission)

**Software :**

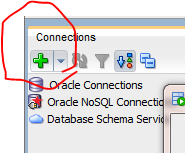
Oracle 11g Express Edition (used)  
SQL developer from Oracle (used)

**Tutorials :**

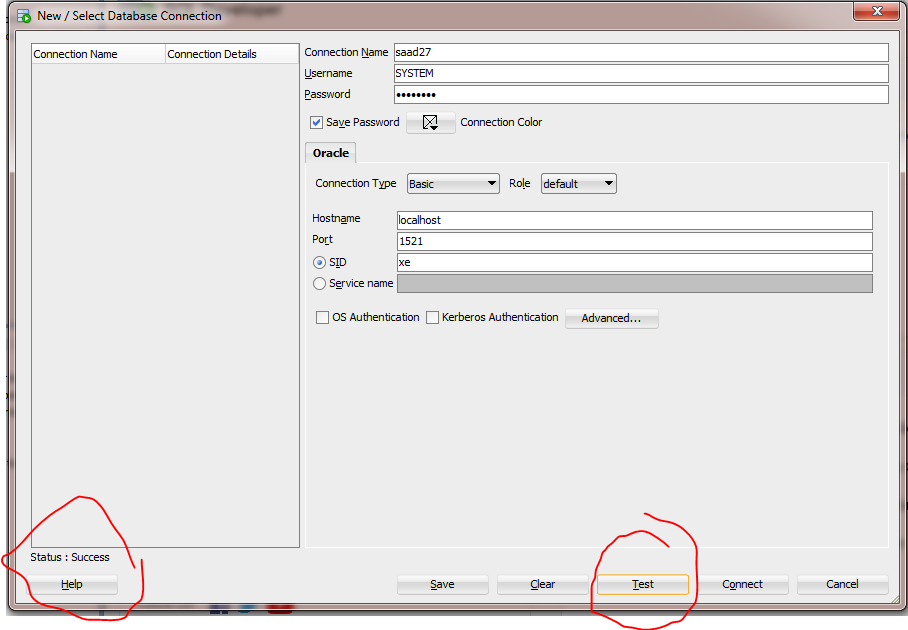
https://www.youtube.com/playlist?list=PLV8vIYTIdSnb7av5opUF2p3Xv9CLwOfbq

**Configure Sql developer to establish a connection :**

If sql developer is not installed then download from oracle site with ( jdk 1.8 required).



Add a new connection



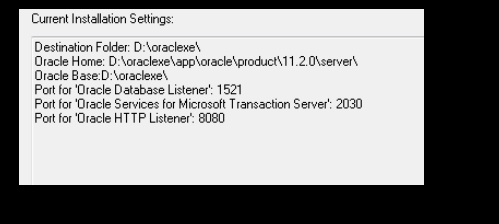
**Connection Name :** can be any   
**Username :** Specified when installing oracle 11g if not then it is SYSTEM

**Password :** Specified on installing oracle 11g  
**Hostname :** specified on installing oracle 11g

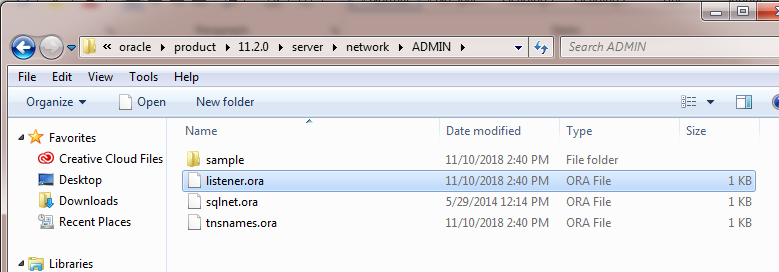
**Port :** 1521 (default)

Click on test if status:success then connection is working.

**Default settings :**

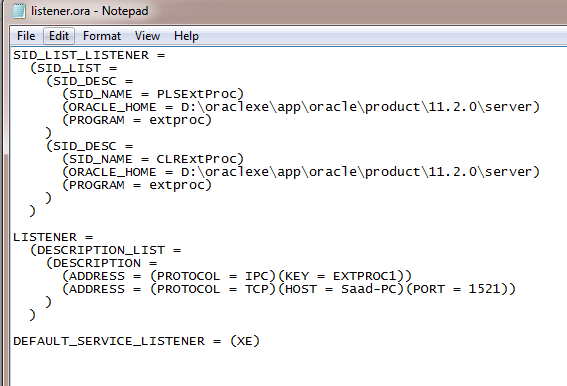


**For connection details :**

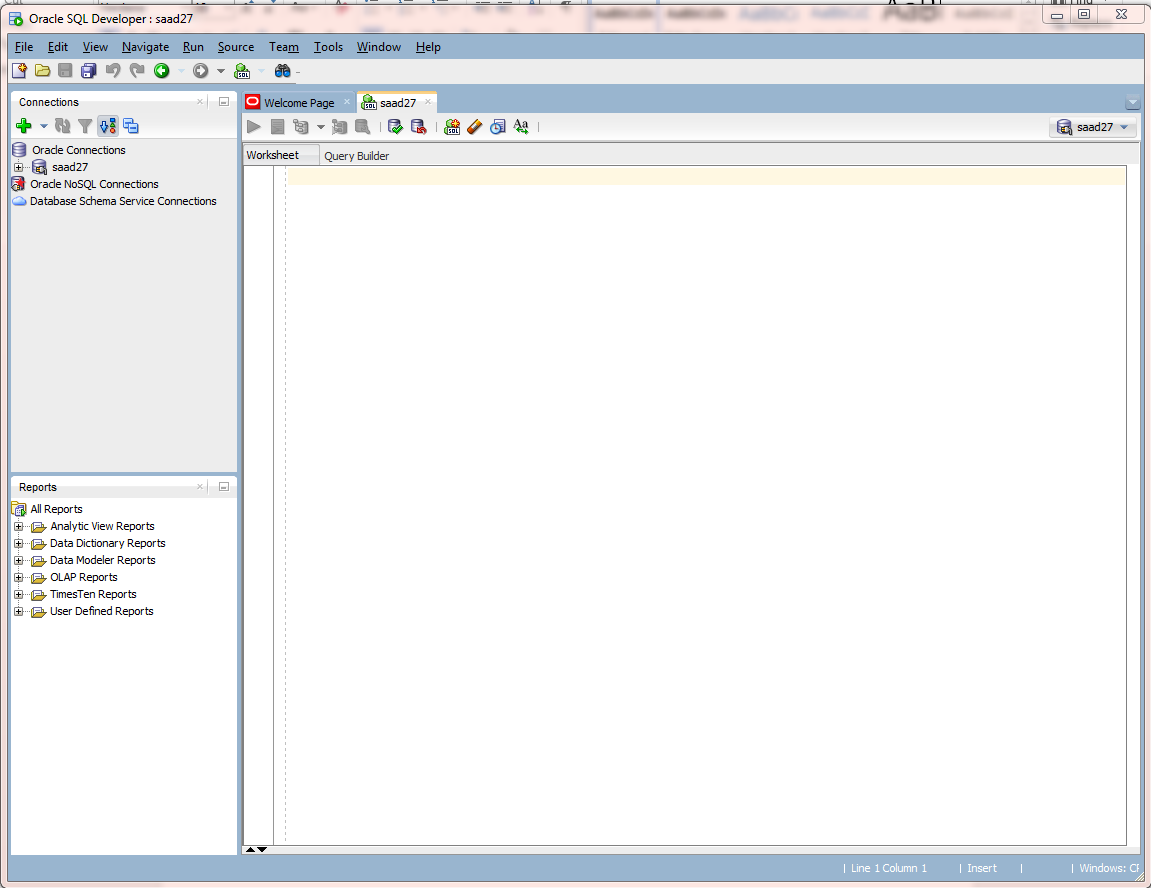


Open listener.ora with notepad it has connection settings.

Like this :

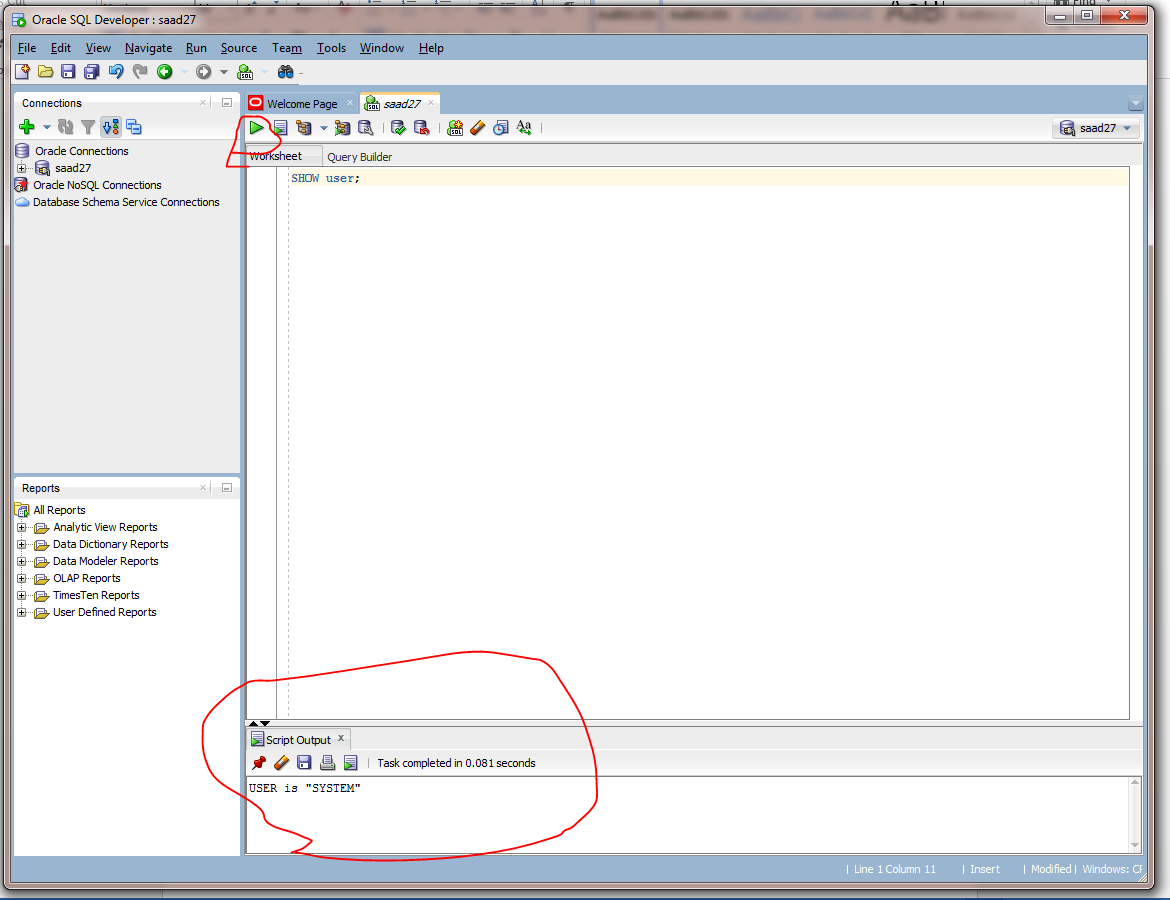


**After successful connection :**



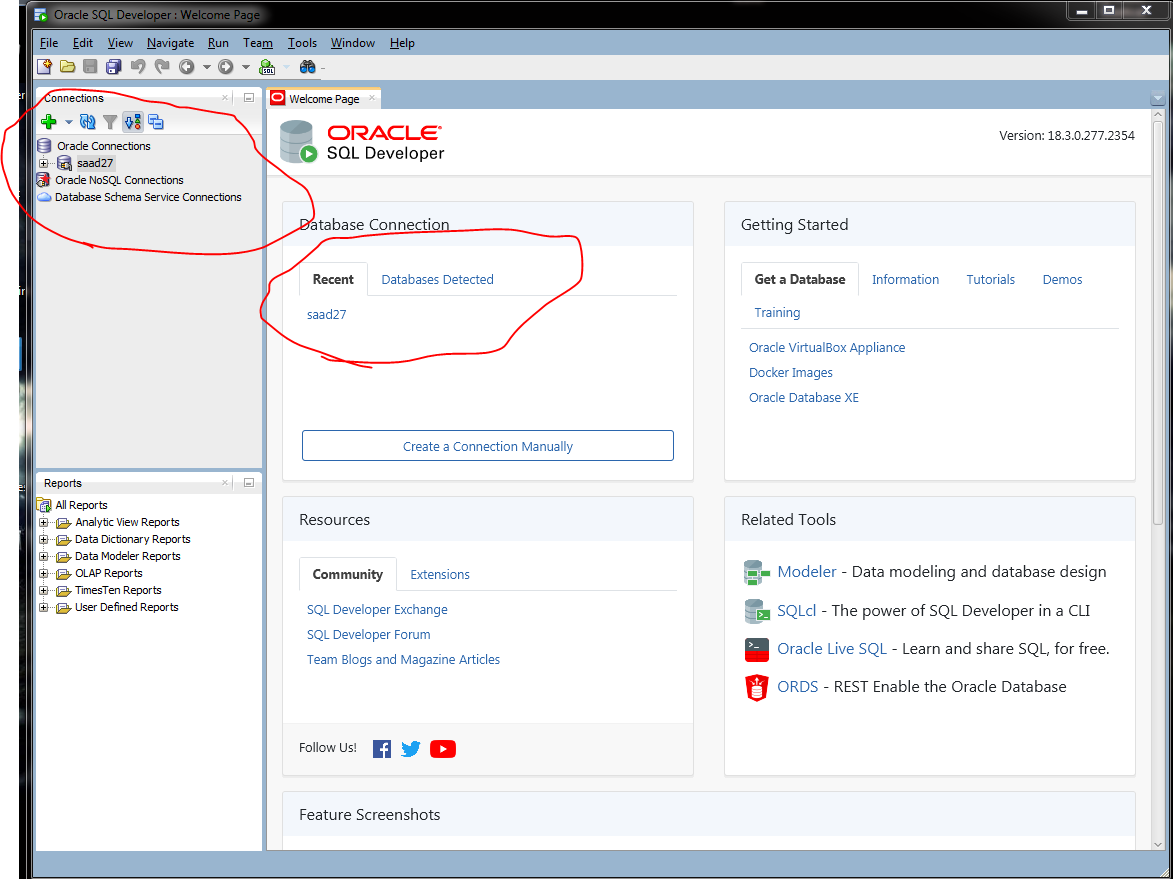
**This where queries are typed.**

**Example :**



Ctrl + enter for executing query or that green play button.

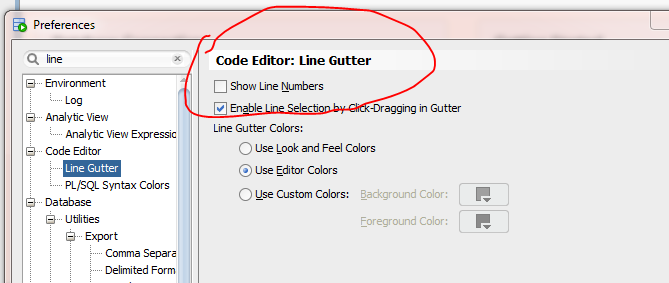
**New workspace :**



Connection is necessary for new work space .   
alt + f10 for new work space

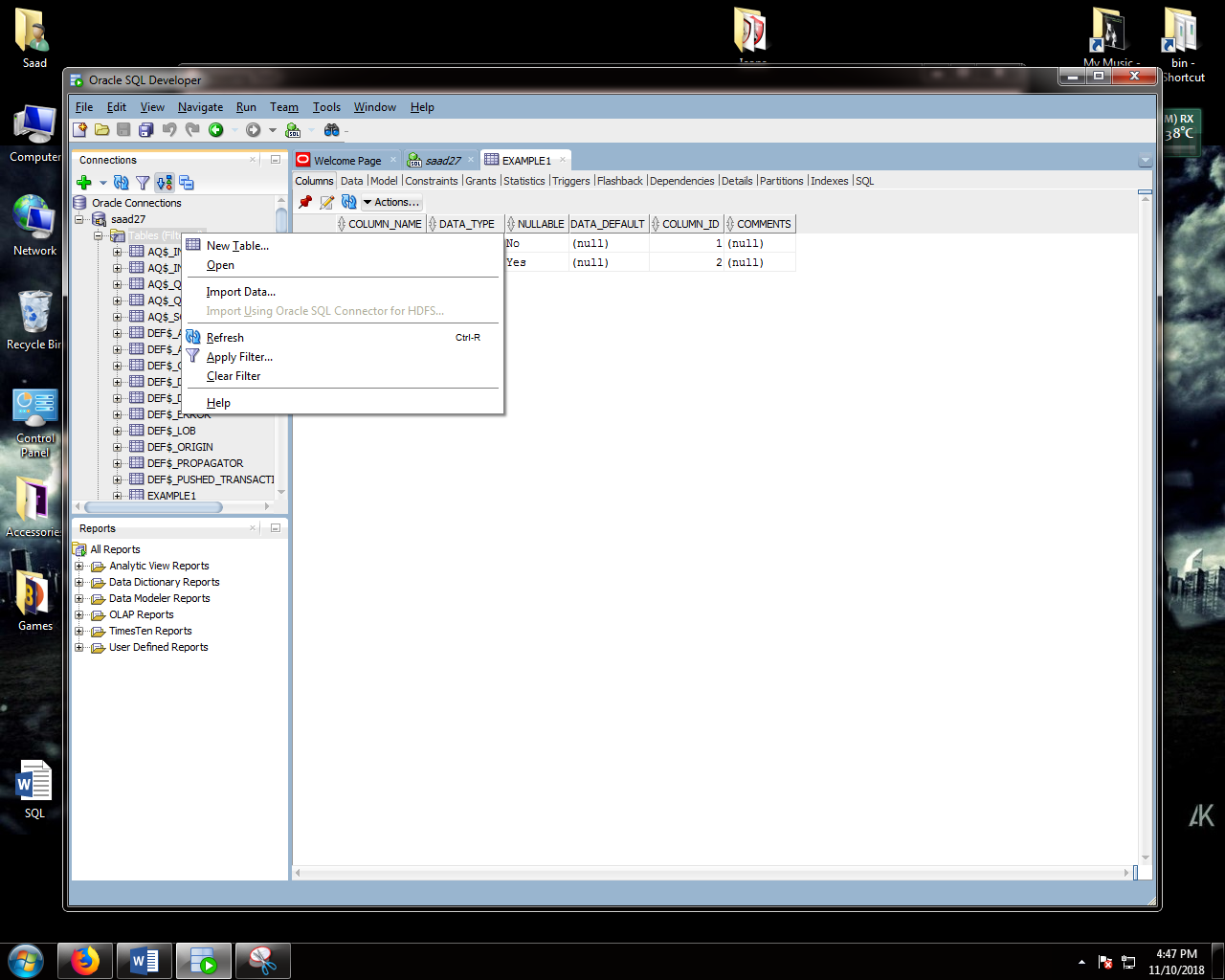
Select connection and your done.

Enable line numbers for work space :



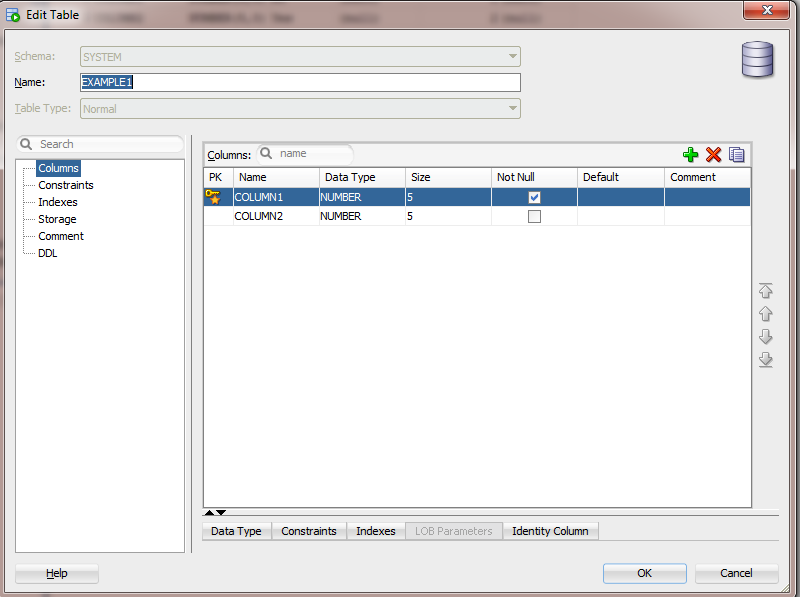
Tools🡪 preferences 🡪 line gutter 🡪 show line numbers.

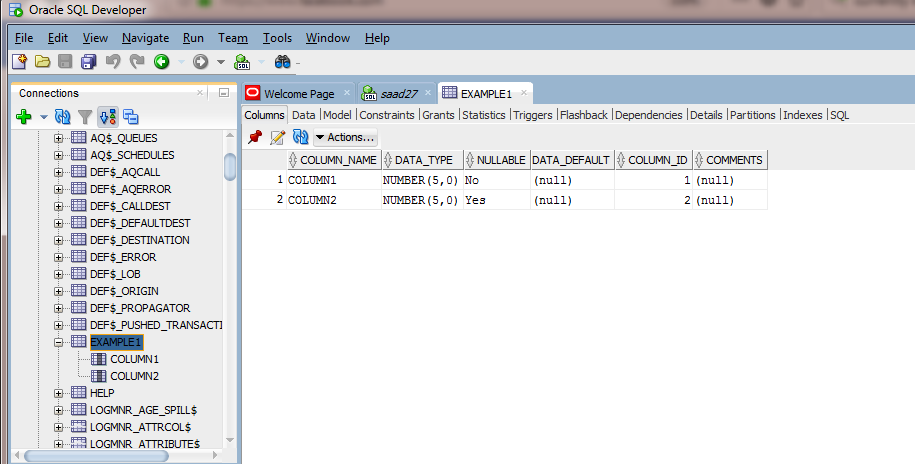
**Creating a table :**



Click on connection

On new table



Set desired values. 

**By SQL Code :**

**Syntax :**

CREATE TABLE <name> (

column1 datatype(size),

column2 datatype(size),……… ) ;

**SQL Constraints :**

Used to specify rules for data in a table .

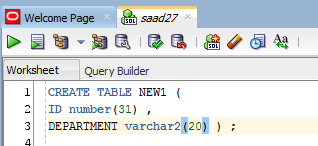
Can be specified during creation of a table or modification (Alter).

**Syntax :**

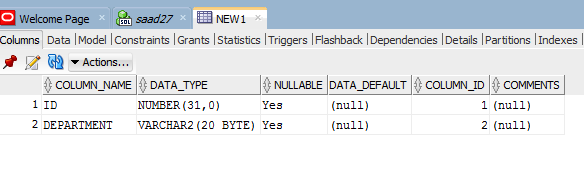
CREATE TABLE <name> (

column1 datatype(size) constraint\_name,

column2 datatype(size) constraint\_name,……… ) ;







**Some Constraints in SQL :**

Not null  
Primary key   
Foreign key   
Check  
Default

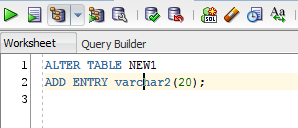
**SQL ALTER STATEMENT :**

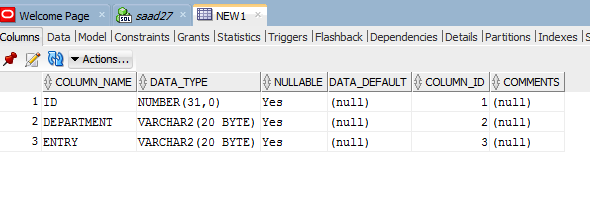
Does Modifications in a table .

**ADD COLUMN :**

SYNTAX :

ALTER TABLE <table\_name>   
ADD <column\_name> data\_type;

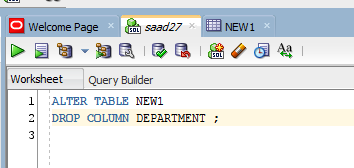


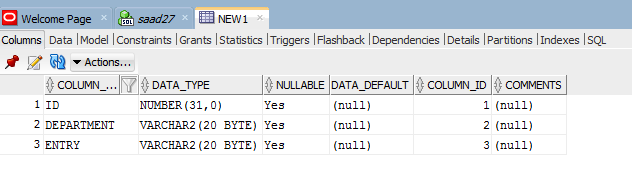


**DELETE COLUMN :**

SYNTAX :

ALTER TABLE <table\_name>   
DROP COLUMN <column\_name> ;





**CHANGE DATA TYPE :**

ALTER TABLE <table\_name>

ALTER COLUMN <column> <specify \_data type >(value);

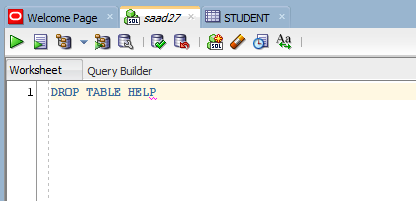
**Sql drop statement :**

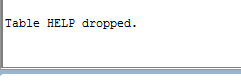
**Delete Table :**

Deletes a table :

**Syntax :**

DROP TABLE <table\_name>





**Delete Database :**

Deletes a database :

**Syntax :**

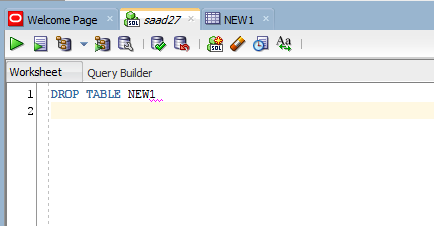
DROP DATABASE <table\_name>

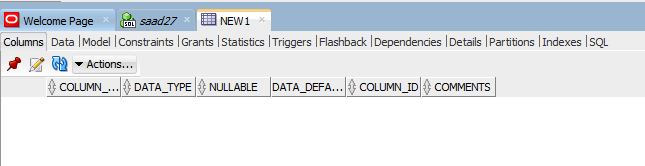
**TRUNCATE TABLE**

Delete contents but settings remain.

**Syntax :**

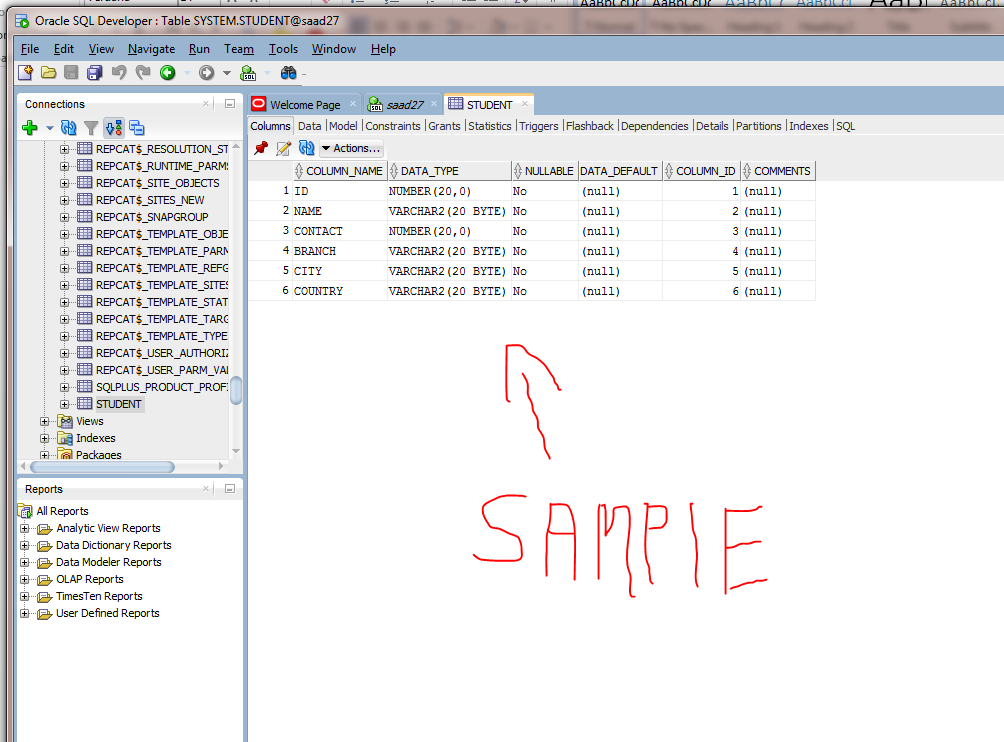
TRUNCATE TABLE <table\_name >



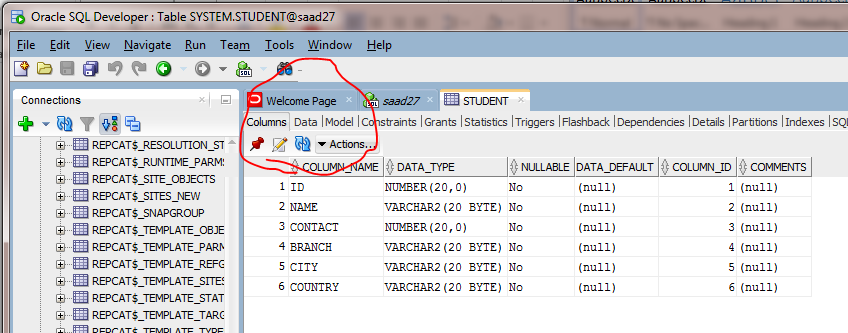


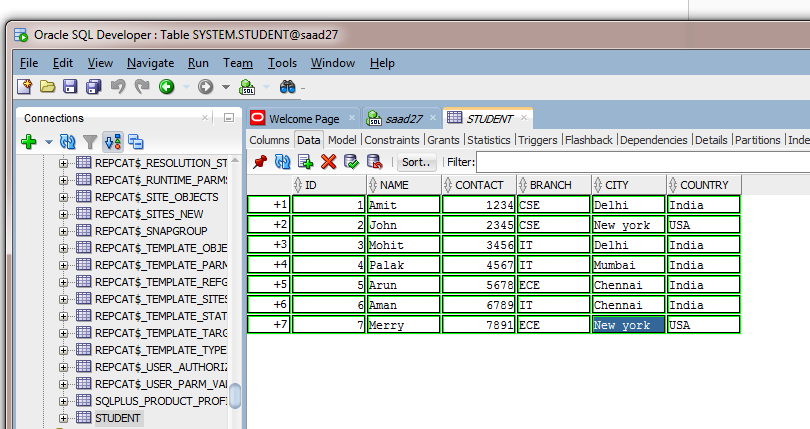
**Inserting Data in a table :**

Open the table first :

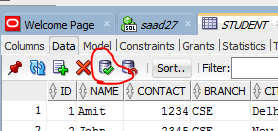


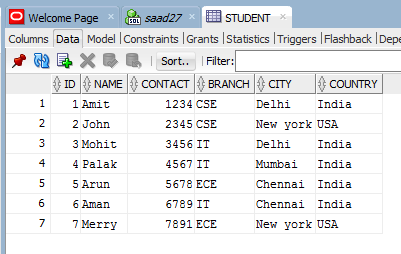
Click on data option :





Add data by adding rows with respect to fields.

 Tick to save changes or F11 key.



**Sql Select Statement :**

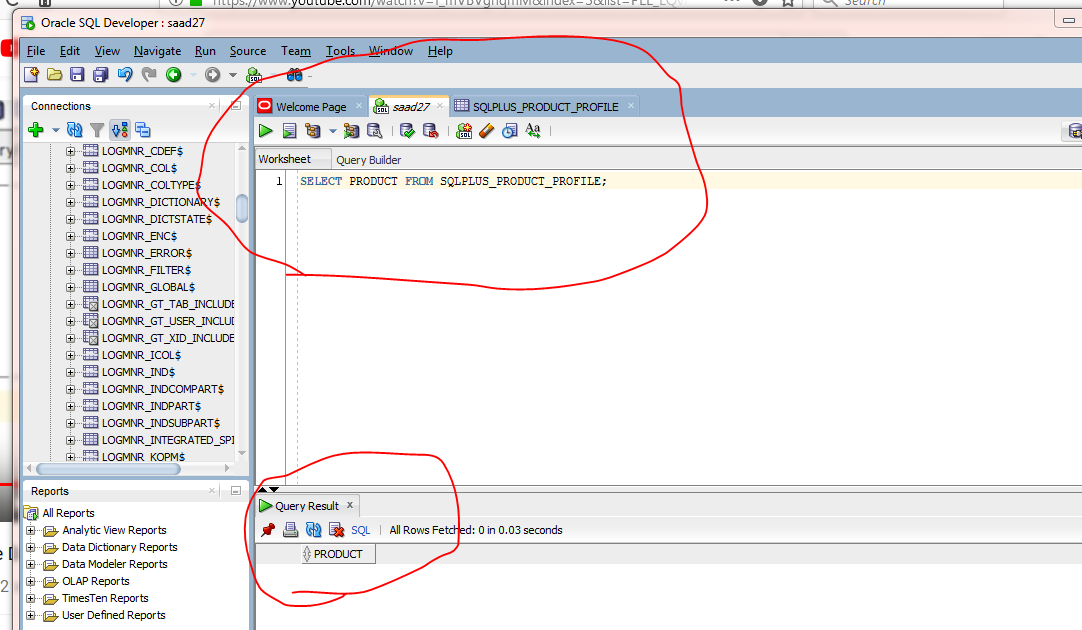
The select statement is used to select data from a database.

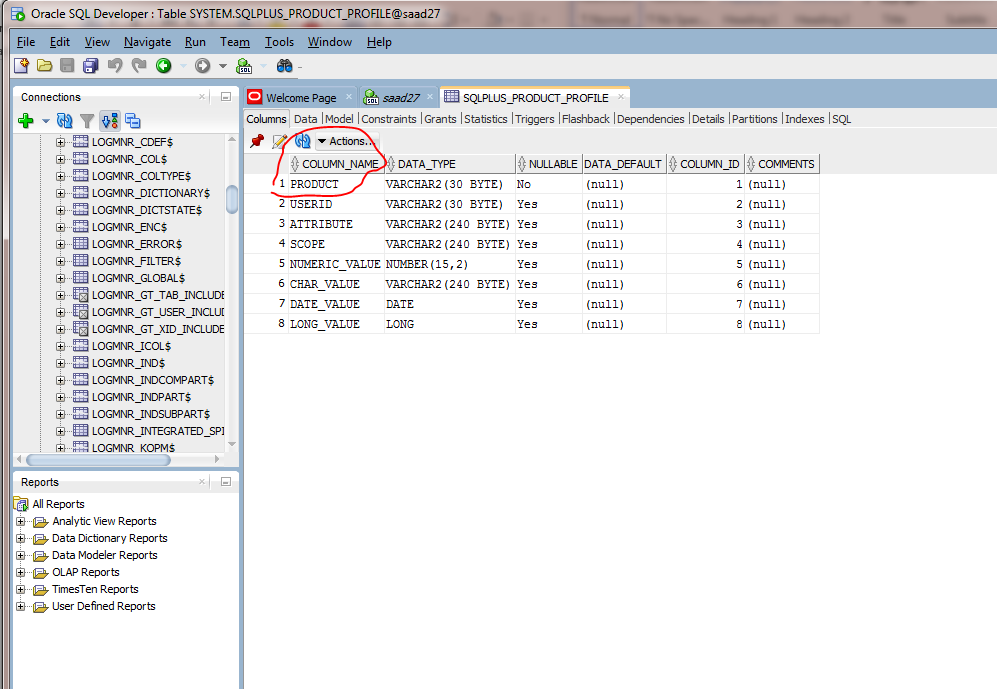
The result is stored in a result table called result set

**Syntax :**

SELECT [expression] FROM <table\_name>;

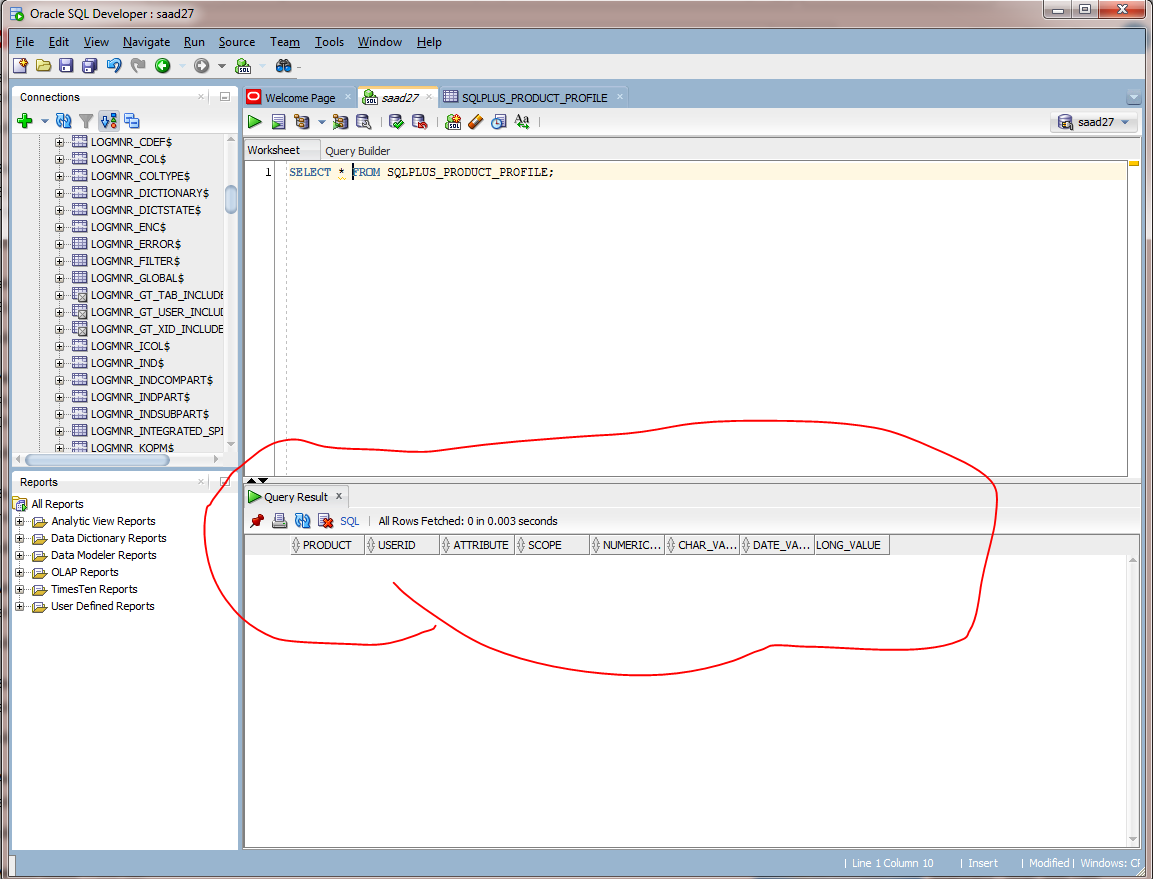
**Example :**





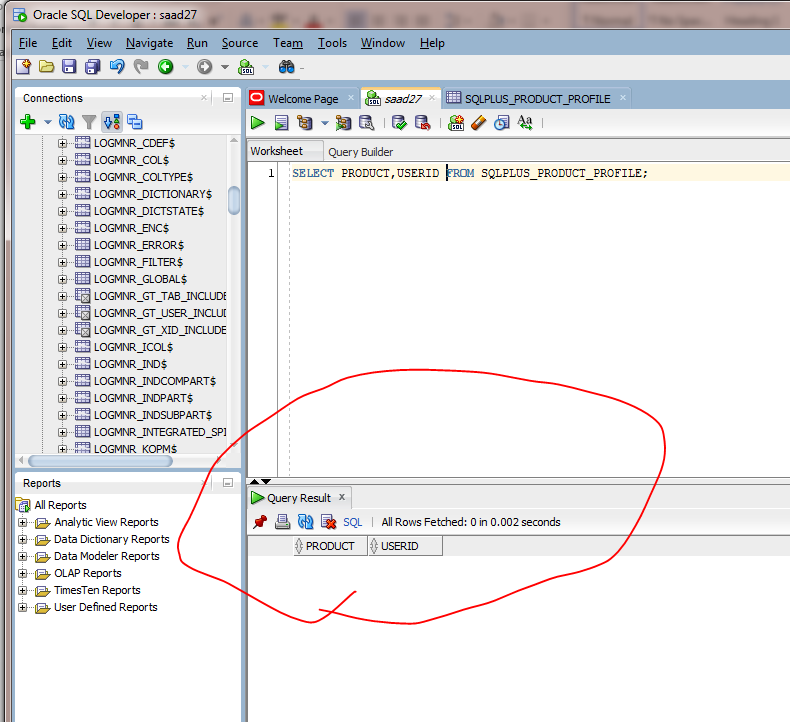
**For selecting whole :**

SELECT \* FROM <table\_name>



**For selecting two specified :**

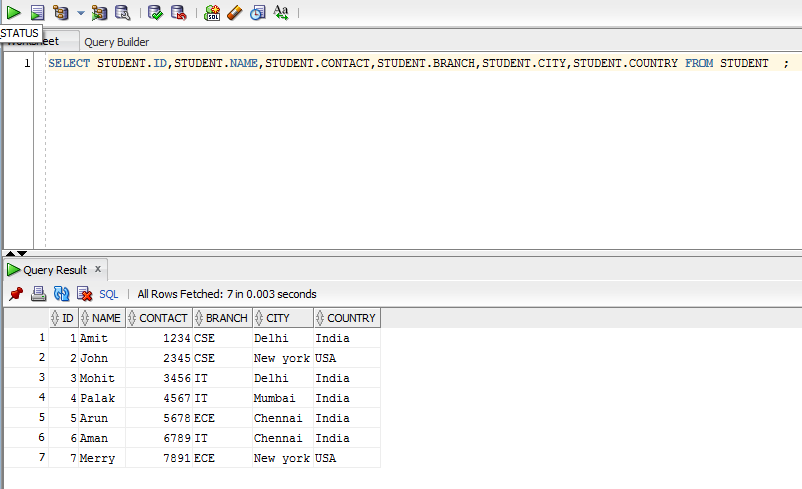
SELECT [COL1,COL2] FROM <table\_name>



If a columns contains similar name of data the write like this :

SELECT DISTINCT [COLUMN] FROM <table\_name>

**Selecting specified :**



**SQL Where Clause :**

The where clause is used to find data for a specified condition .

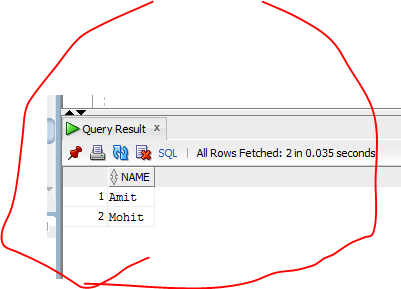
**Syntax :**

SELECT [COLUMN] FROM <table\_name> where [COLUMN] operator value ;

**Example :**Select student name from delhi city from table.

**Query :**

SELECT NAME FROM STUDENT WHERE CITY=’Delhi’;



**Sql Operators :**

= Equal

<> Not equal to also written as “ != “ in some cases

> Greater than

< Less than

>= Greater than equal to

<= Less than or equal to

BETWEEN Between a certain range

LIKE Search for similar stuff

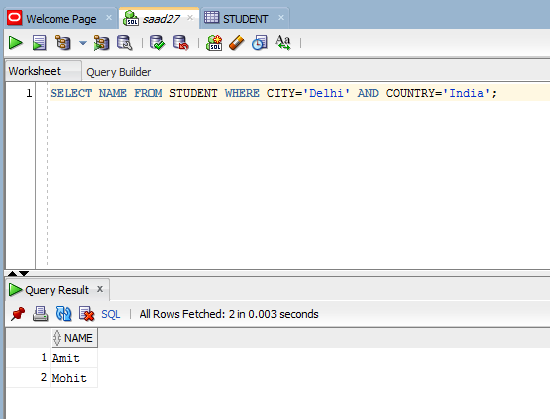
IN Multiple possible values in a column.

**Sql AND Operator :**

Executes if both the first and second condition are true.

**Syntax : AND**

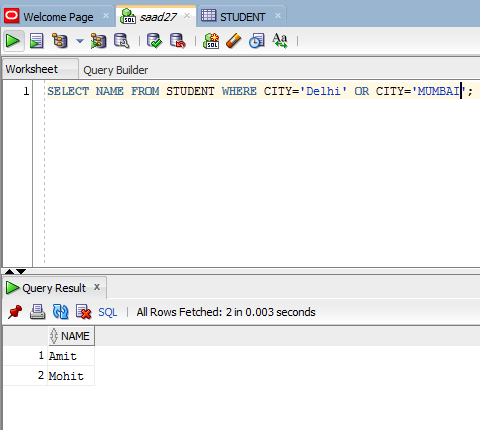
**Example :**



**Sql OR Operator :**

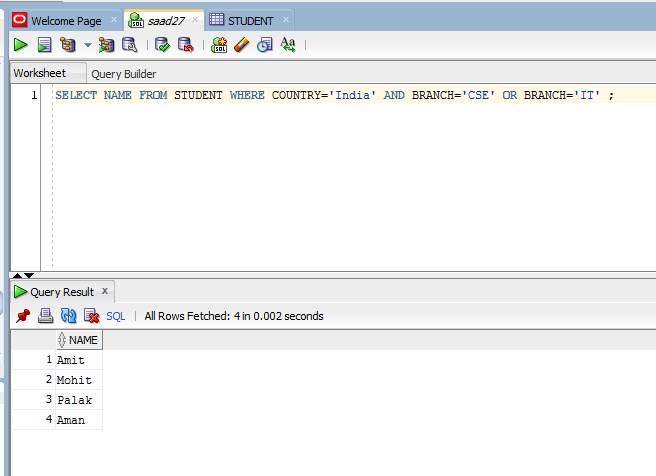
Executes if either the first or 2nd condition is true.

**Syntax : OR**



**EXAMPLE :**

Select a student studying CSE OR IT living in India.



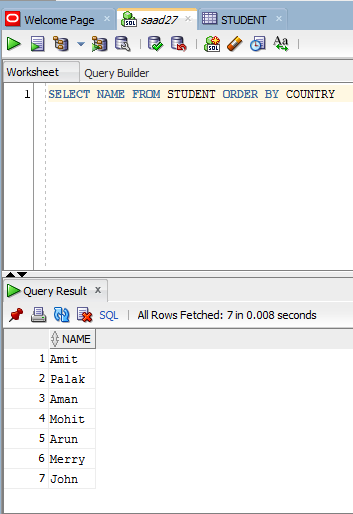
**SQL Order by clause :**

Used to sort the selected result in ascending or descending order.   
Default sorting order is ascending.

**Syntax :**

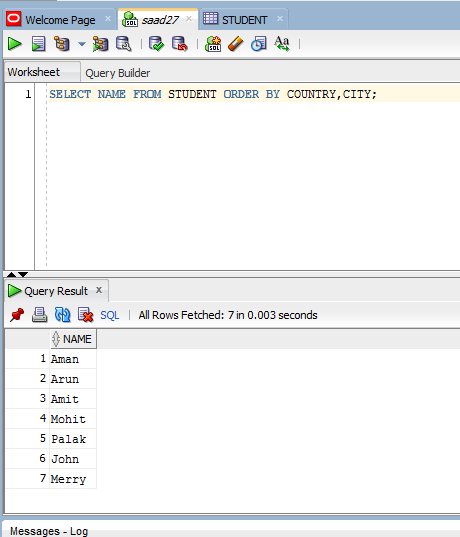
SELECT [COLUMN] FROM <table\_name> order by [column] ASC\DESC.

**EXAMPLE :**



**Example :**

Display name of students in ascending order by country and city.



**SQL Insert into :**

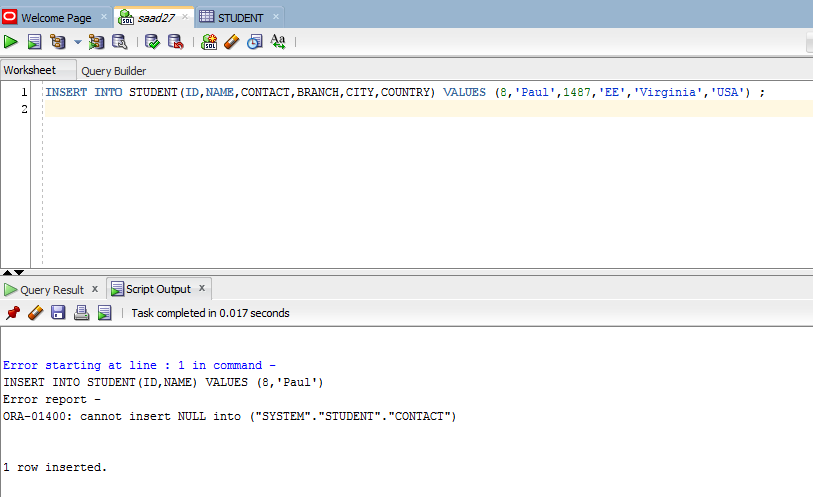
Used to insert new data into table.

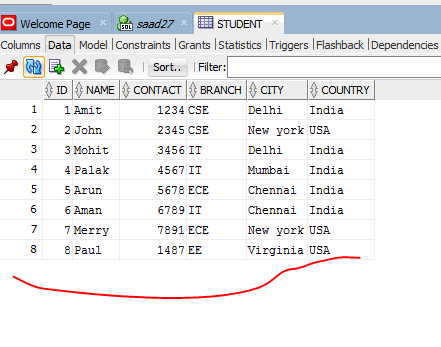
**SYNTAX:**

insert into <table\_name> (column1,column2….) values (value1,value2…) ;

#value1 will be inserted in column1   
#value2 will be inserted in column2

Example :





**SQL UPDATE STATEMENT :**

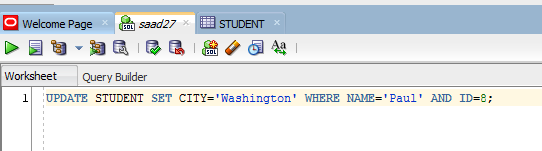
Changes the data in the table for updates or making adjustments.

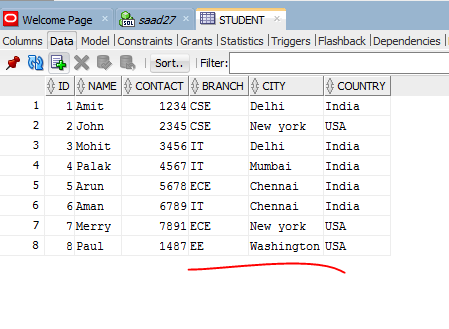
**Syntax :**

UPDATE <table\_name> set [column]=new.value ,[column1]=newvalue WHERE [SPECIFIED column]=row ;

**Example :**

Change city of paul to Washington





WHERE clause is necessary cause it specifies the row for updating record.

**SQL DELETE STATEMENT :**

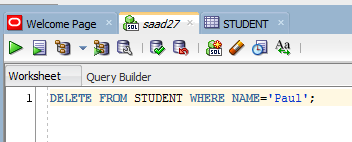
Deletes a specified record or data from table.

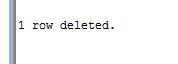
**Syntax :**

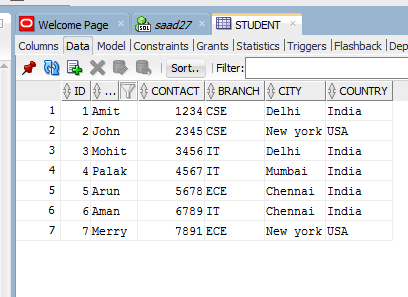
DELETE FROM <table\_name> WHERE ROW=SPECIFIED;

**Example :**

Delete entry of paul from student.







**SQL LIKE OPERATOR :**

Used to find specified pattern in a record by where clause.

**Syntax :**

SELECT [COLUMN] FROM <table\_name> WHERE COLUMN LIKE PATTERN ;

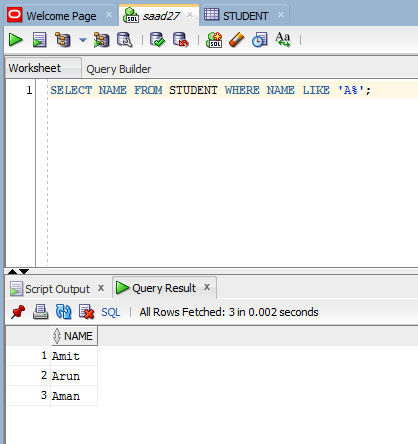
Pattern is of types :

% ( it means any number of characters before or after when specified )

\_ (it means only one character included )

**Example :**

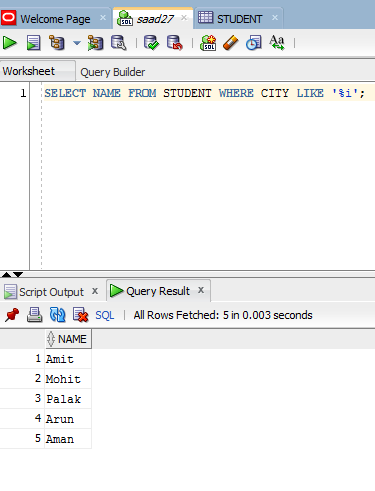
Select students whose name starts with A.



A% shows that name can start with A and anything can come after.

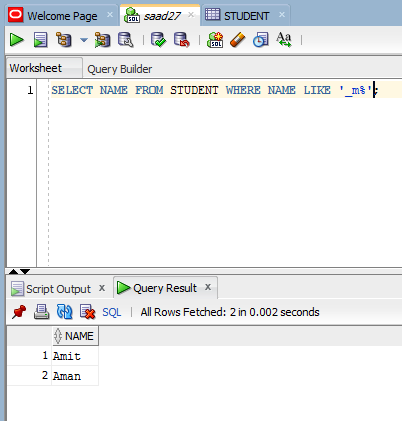
**Example :**

**Find students where city ends with i**



**Example :**

**Find student name whose 2nd letter is M.**



**Here \_ shows only 1st letter available m placed as 2nd letter and % means anything after.**

**SQL SET OPERATION (UNION OPERATOR)**

Combines the result of two or more select statements.

Each select statement must have same number of columns and data types.

Columns should also be in same order.

**SYNTAX :**

SELECT [COLUMN1,COLUMN2] FROM TABLE1

UNION

SELECT [COLUMN1,COLUMN2] FROM TABLE2

#This will remove duplicate(similar) values.

**Syntax 2 :**

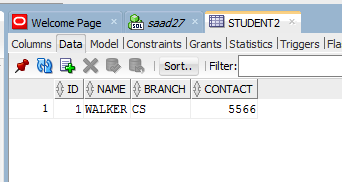
SELECT [COLUMN1,COLUMN2] FROM TABLE1

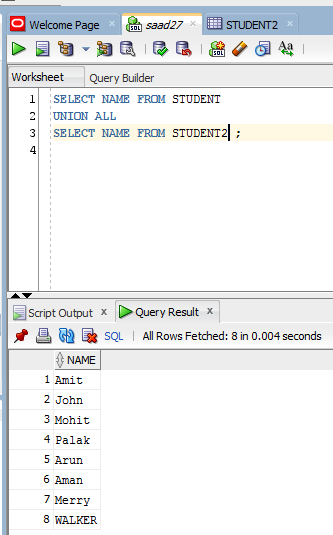
UNION ALL

SELECT [COLUMN1,COLUMN2] FROM TABLE2

#This will not remove duplicate values.

**Example**





**SQL SET OPERATION (INTERSECTION OPERATOR)**

Selects same values from two tables.

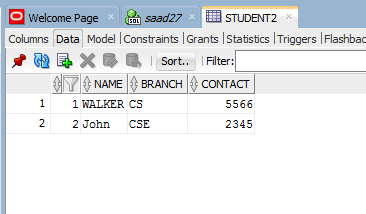
**Syntax :**

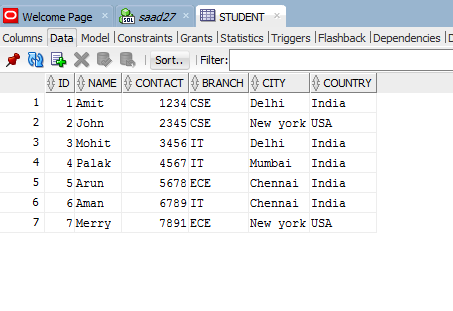
SELECT [COLUMN] FROM <table1>

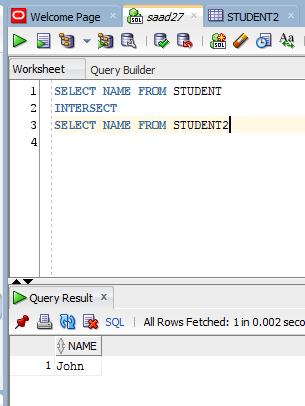
INTERSECT

SELECT [COLUMN] FROM <table2>

**Example :**







**Sql operators ( - minus operator ) :**

Selects values which are present in one table but not in the other.

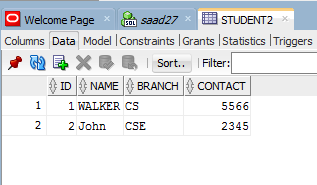
Syntax :

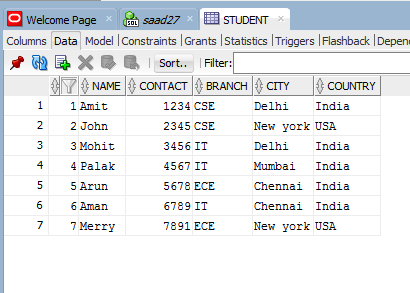
SELECT [column] FROM <table>

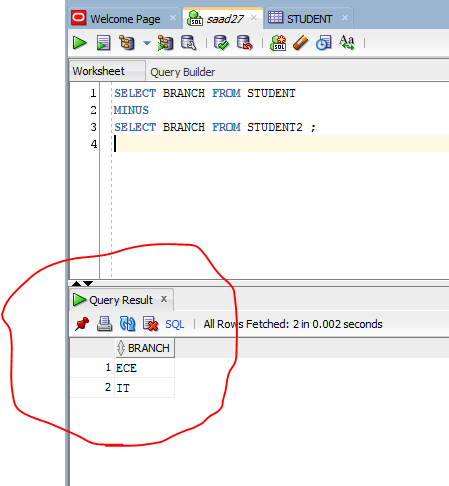
MINUS

SELECT [COLUMN] FROM <table>

Example :







**SQL Aggregate Functions :**

Usually performed on numeric values.

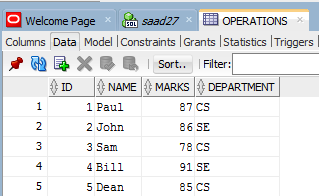
**Sum:**

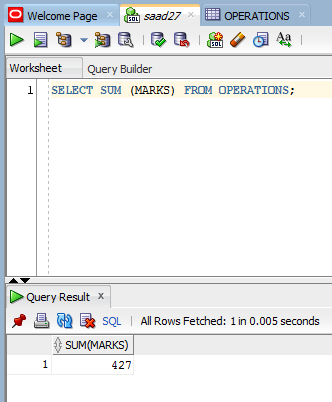
Provides the sum of a column specified.

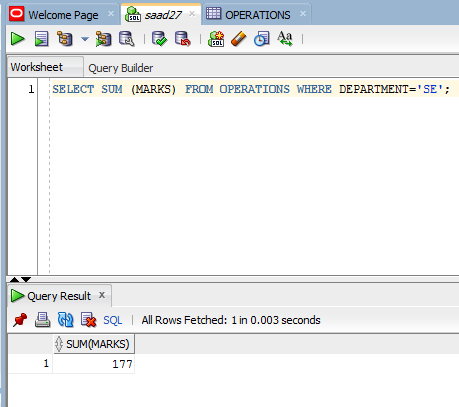
**Syntax :**

SELECT SUM ([COLUMN]) FROM TABLE WHERE ROW=SPECIFIED ;

**Example :**







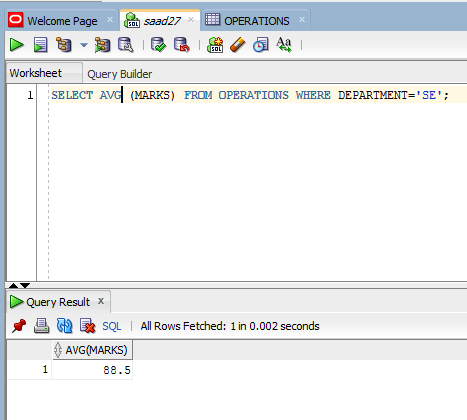
**Average:**

Provides the average of a column specified.

**Syntax :**

SELECT AVG ([COLUMN]) FROM TABLE WHERE ROW=SPECIFIED ;

**Example :**



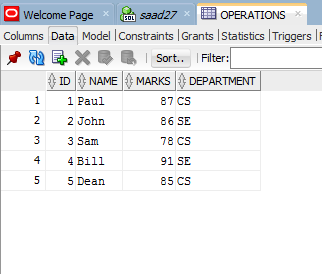
**Count :**

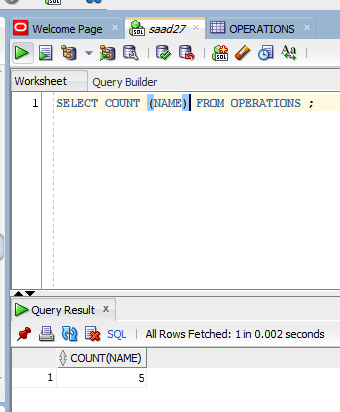
Provides the numbers of rows from a column.

**SYNTAX :**

SELECT COUNT ([COLUMN]) FROM <table>;

**Example :**





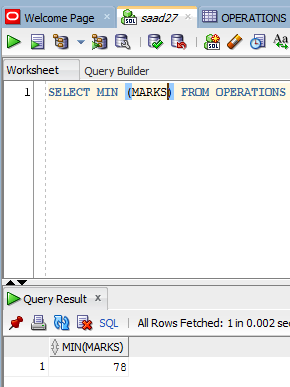
**MINIMUN :**

Provides the smallest value in a [column].

**Syntax :**

SELECT MIN ([COLUMN]) FROM TABLE;

**Example :**



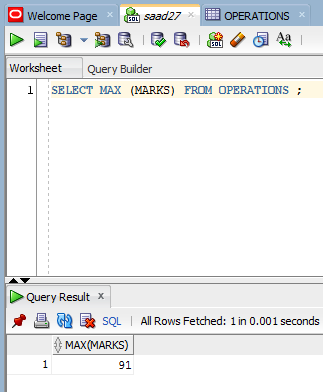
**MAXIMUM :**

Provides the HIGHEST value in a [column].

**Syntax :**

SELECT MAX ([COLUMN]) FROM TABLE;

**Example :**



# SQL Join :

# Used to join rows from two or more tables.

# 

# Sql Inner join :

# Will display the exact common rows of the tables specified.

# Syntax :

# SELECT [COLUMNS] FROM <table1>

# INNER JOIN <table2>

# ON table1.[matching column]=table2.[matching column] ;

# Example :

# 

# 

# 

# EXAMPLE 2 :

# 

# Here STUDENT.COLUMN specifies where the column belongs to.

# SQL LEFT OUTER JOIN :

# Will display left side along with common point.

# Syntax :

# SELECT [COLUMNS] FROM <table1>

# LEFT OUTER JOIN <table2>

# ON table1.[matching column]=table2.[matching column] ;

# Example :

# 

# SQL RIGHT OUTER JOIN :

# Will display right side along with common point.

# Syntax :

# SELECT [COLUMNS] FROM <table1>

# right OUTER JOIN <table2>

# ON table1.[matching column]=table2.[matching column] ;

# Example :

# 

# NULL SHOWS LEFT SIDE IS EMPTY.

# SQL FULL OUTER JOIN :

# Will display right side along with LEFT SIDE and common point.

# Syntax :

# SELECT [COLUMNS] FROM <table1>

# FULL OUTER JOIN <table2>

# ON table1.[matching column]=table2.[matching column] ;

# Example :

# 

# Null values to the left of walker shows that table 2 has no left values. Same goes for the values right of table 1.

# Create Database :

# Used to create a database for entries.

# Syntax :

# CREATE DATABASE <name> ;

**SQL NESTED QUERY :**

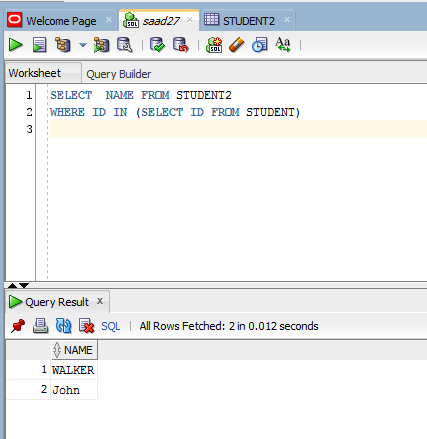
A query where another query can also be initialized within.

**SELECT :**

SYNTAX:

SELECT COLUMN1,COLUMN2 FROM TABLE1

WHERE COLUMN OPERATOR (SELECT COLUMN FROM TABLE 2)



Used when one part of data is in table 1 and another part in table 2.