**eliminating duplicate rows & 2nd letter is Q & order it:**

*select distinct colName from table where colName like ‘\_Q%’ order by colName;*

**some functions:**

*select upper(colName),lower(colName), length(colName), substr(colName,m,n) from table;*

*select round (colName, n), trunc (colName, n) from table; # the column should be Number type*

*select months\_between(colName, colName), add\_months(colName, n) from table; # the column should be Date type*

*select to\_char(pType,’fmt’),* *to\_number(pType,[’fmt’]), to\_date(pType, 'fmt')) from table; # pType is the colName of the column which should be convert it*

*select nvl(colName,n) from table; # the n could be any value 0,1, or’ABC’, etc.*

*select avg(colName), max(colName), min(colName), sum(colName), count(colName) from table group by colName; # you can’t use where just having*

**data manipulation language (DML):**

*insert into table (colName1, colName2) values (value1, 'VALUE2'');*

*update table set colName = value [where condition];*

*delete from table [where condition]; commit**rollback*

**table / view functions:**

*create table tableNew (colName1 colType1(n), colName2 colType2(m, k)); # n, m, k are numbers*

*create table tableNew as (select \* from tableOld where 1 = 2); # creating new table & copy another table’s column*

*insert into tableNew (colName1, colName2) (select colName3, colName4 from tableOld [where condition]); # copy another table’s rows*

*rename tableOld to tableNew; # to rename a table;*

*alter table table add (colName colType(n)); # to add a column in a table, n is a number*

*alter table table modify (colName, colType(n)); # to modify a column in a table, n is a number*

*alter table table drop column colName ; # to delete a column in a table*

*drop table table; # to delete a table truncate table table; # to delete all data in a table*

*create view v as select colName from table [where condition]; to create a view drop view v; # to delete a view*

*create table tableNew(colName, colType(n) not null, colName2 colType2(n), constraint table\_ colName2 \_uk unique (colName2), constraint table\_ colName2\_pk primary key(colName2), constraint table\_ colName2\_fk foreign key (colName2) references tableOld (colName2), constraint table\_ colName2\_ck check (colName2='value' or colName2=' value ')); # assign constraints*

*alter table table add constraint table\_ colName\_pk primary key (colName); # assign constraints after creating table*

*alter table table drop constraint table\_colName\_fk; # drop constraints*

*alter table table [dis/en]able constraint table\_ colName\_pk; # (dis/en)able constraints*

|  |
| --- |
| *declare*  *colName1 colType(n);*  *begin*  *select colName2*  *into colName1*  *from table*  *DBMS\_OUTPUT.PUT\_LINE ('this is colName1 ' || colName1);*  *exception*  *when excepName then*  *...*  *end;* |

*In BEGIN*

*IF condition THEN*

*statements;*

*[ELSIF condition THEN*

*statements;]*

*[ELSE*

*statements;]*

*END IF;*

*In DECLARE*

*colName1 table. colName2%TYPE;*

*colName3 colType(n);*

*colName4 colName3 %TYPE := value;*

*# to copy column type to another column*

*h*

*In EXCEPTION*

*excepName like no\_data\_found*

*create or replace procedure proName( colName colType) is*

*begin*

*……*

*end;*

*create or replace procedure proName is*

*colName colType (n) := vlue;*

*begin*

*DBMS\_OUTPUT.PUT\_LINE('This is colName '||colName);*

*end proName;*

*exec proName(value); # to execute procedure SET SERVEROUTPUT ON; # before everything*

*create or replace function funName return returnType is*

*var varType := value;*

*begin*

*…….*

*return var;*

*end;*

*# returnType & varType* ***should*** *be* ***equal*** *in type*

*create or replace trigger t*

*after insert on table*

*for each row*

*begin*

*DBMS\_OUTPUT.PUT\_LINE('newRow');*

*end;*

*alter trigger t [dis/en]able; # to (dis/en)able trigger*

*drop trigger t; # to drop trigger*

*­­­­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**In JAVA:**

**In mian:%**

*Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@coestudb.qu.edu.qa:1521/STUD.qu.edu.*

*qa","ha1401812", " ha1401812");*

*Statement stmt = conn.createStatement();*

*ResultSet rs = stmt.executeQuery("select colName1, colName2 from table");*

**while** (rs.next()) {

String *colName1*= rs.getString(1);

**int** *colName2* = rs.getInt(2);

System.***out***.println(*colName1*+ " " + *colName2*); }**%**

rs.close();

conn.close();

{String sql="update table set colName=value"+" where condition";//or insert or delete

**int** affectedRows =stmt.executeUpdate(sql); this is satisfied for insert

System.***out***.println("Number of updated records "+affectedRows);}

{PreparedStatement stmt=conn.prepareStatement("insert into table(colName1, colName2) values (?,?)");// *or select or update or delete*

stmt.setDouble(1, *value*);// *colName1 value*

stmt.setString(2,"value");// *colName2 value*

stmt.executeUpdate();}

{CallableStatement stCall=conn.prepareCall("{call proName(?,?)}");

Same as the previous statement inner}