WNOSQL(*)- DATABASE

WNOSQL (*) DB

By

WILMIX JEMIN J,

JeminInformationTechnology

About the Author and Preface

This WNOSQL is Designed by Analzing many database documents…

Using WNOSQL one can Design the Database Projects as Fast

As could. I Thank God for this wisdom given to me…

----------Wilmix Jemin J,Jemin Information Technology

This EBOOK is Printed in Asia.

To Make Software Fast like Rabbit movement

and a global redistribution of prosperity

@2016 JeminInformationTechnology , All Rights Reserved

Acknowledgments

We'd like to acknowledge all of the people who played important roles in the creation

of this book. We'd also like to thank all of the developers who've spent time reading this manuscript

and pointing out all of the problems.

Finally, we'd like to extend a sincere thank you to the people who participated in the

WNOSQL Program. In particular, those who've left feedback in the Author

Online forum have had a strong impact on the quality of the final printed product.

And for providing English translations of the text resources, we'd like to thank

Github and our supporters.

Thanks to all!

---------WILMIX JEMIN J

About this Book

Welcome to WNOSQL! If you've picked up this book, we suspect you're a Database developer

working with database who's somehow or other heard about database like sqlserver or oracle.

Perhaps you've worked with the Other databases in the past, perhaps you've worked with

another Databases , or perhaps this is your first step into Database security.

Whichever path has led you here, you're probably looking for a good introduction

to the new WNOSQL securable database. This book intends to give you that introduction

and much more. If you've never heard of WNOSQL, we cover the basics in enough

depth to keep you in tow. If you know what WNOSQL does, but want a deeper understanding

of how it does it, we'll provide that too.

Roadmap

Book is focused on WNOSQL database security , if you have knowledge or experience about CDOLLAR and C# you can easily focus it.

But Minimum CDollar and C# Technical Knowledge is required to focus on

Studying, Designing WNOSQL database Modules .

WNOSQL is an Advanced Database.

The Brief Contents

UNIT 1	Introduction	10-13
UNIT 2	WNOSQL (WSQL*) Basics	14-28
UNIT 3	WNOSQL (WSQL*) PLSQL Basics	29-44
UNIT 4	WNOSQL (WSQL*) Forms And Reports	45-48
UNIT 5	WNOSQL(WSQL*) interaction With native databases	49
UNIT 6	WNOSQL(WSQL*) Test Exercises	50-52
UNIT 7	WNOSQL(WSQL*) PLSQL Programming	53-86
UNIT 8	WNOSQL (WSQL*) using CDollar,JAS,JDollar, etc. And WNOSQL Programming Exercises	87-137

UNIT 9	*WNOSQL(WSQL*) Test Exercises*	138 – 184

UNIT 10	WNOSQL (WSQL*) Mock Test Exercises And Practice Test For Professionals.	185- 186

Code conventions

The following typographical conventions are used throughout the book:

■ Courier typeface is used in all code listings.

■ Courier typeface is used within text for certain code words.

■ Italics are used for emphasis and to introduce new terms.

■ Code annotations are used in place of inline comments in the code. These highlight important concepts or areas of the code.

Code downloads

This will get you the WNOSQL.zip file by purchasing it.

a couple of WNOSQL archive files —as well as some documentation of the source. Instructions on how to install the application are contained in a README file in that download.

ABOUT WNOSQL

==========

WNOSQL means Wilmix NOSQL.(W*SQL)
is a Securable database invented by wilmix jemin j in GDollar,JDollar(JWEB) ,and JAVA .

No need to write any Queries but to pass parameters in WDBASQL database for PLSQL f(x)s.

We can also write WDBASql queries like SQL/Oracle Queries.

WNOSQL is focused only on Windows Platform.

ADVANTAGES of Using WNOSQL in Windows Platform

A) To secure the data from hackers.
data cannot be taken by hackers in windows.

b) your .wdba data can be accessed from any location.

no need of datasource ; so we can say wnosql has remote database connection.

c) You can store 1 Thrillions of data using WNOSQL.

d) All .wdba data will be stored in a encrypted form that cannot be viewed by hackers.

UNIT:1: Introduction

Welcome to WNOSQL security programming designed by Jemin Information Technology!

What is meant by WNOSQL?

WNOSQL means Wilmix NOSQL.(W*SQL) . No need to write SQL Queries but to pass parameters in WNOSQL database functions.. WNOSQL is also transport data from Oracle db/SQLSERVER to WNOSQL and viceversa.

When it is invented?

It is invented by wilmix jemin j in year 2016.

Why WNOSQL is most important for software development?

Since it provides security , transports data from wnosql to Oracle,Sqlserver, and viceversa. And it will connect with major databases like Oracle,Sqlserver,MYSQL, etc. And it protect the hackers and unwanted users stealing the data.

How WNOSQL database Works?

WNOSQL database uses WNOSQL Editor. WNOSQL(.dlls) are responsible for transporting , executing Query . WNOSQL did not contain server; Instead of that it contain only WebConsole to see the HTML outputs.

WNOSQL(W*SQL) takes the data from Sqlserver and store it. We can also perform retrieval ,Encrypt,Decrypt data , Select particular row, Join Operations, Aggegerate tools,etc. using WNOSQL.

Which you can learn more detail in Unit -II.

let us see how it works? Why it is more advanced?

Explanation:

========

When Users type WNOSQL(W*SQL) Program at first you compile the db program using compile at right corner of the editor.

And Next you run the program using Run at right corner of the editor.

When you click "RUN" it produces .obj and .exe file for futhure use with CDollar, JAVA, Dotnet, and JDollar, JSAUCER, etc. So You can selected only encrypted (.obj) file and run the database program.

SO choose the .dll file to run wnosql program

.When you insert a data in a program it stores it in WNOSQL encrypted file and finally it retrieves the data from encrypted file for futhure use. WNOSQL is responsible for data security with data storage and retrieval management.You can also do programmming simillar to PLSQLand we can change the theme of HTML Output. You can see the output in webconsole by pressing RUNWNOSQL(*) button at the right corner of WNOSQL(*) Editor.

State the Advantages of WNOSQL (*) Database:

a) WNOSQL(*) is a NOSQL databases

b) WNOSQL(*) need no datasource since it remotely connect with CDollar or JDollar Program

c) WNOSQL also has SQL Concepts and SQL Advanced concepts

d) You can see the output the WEB console there is no need for server

e) WNOSQL uses cluster memmory management to protect your data from hackers ,etc.

f) WNOSQL will Store Huge amount of data ie) > Thrillion Thrillion .....

g) WNOSQL(*) has Userfriendly WNOSQL cmd console

h) WNOSQL Prevents SQL INJECTION

i) WNOSQL(*) has attractive syntax.

j) WNOSQL(*) also has PLSQL to execute db statements as a batch.

k) WNOSQL(*) also has Advanced oops like JAVA and C#

l) WNOSQL is a Advanced Database.

m) WNOSQL also transfer to and from Oracle/SQLSERVER/MYSQL to WNOSQL Database; so it is called **PIPELINE DATABASE.**

n) We can also use CDollar dlls with WNOSQL

o) WNOSQL also store the output in .EXE format

p) WNOSQL also be used with other Programming languages through through Oracle/SQL Server.

q)WNOSQL PLSQL uses API format which hacker or unwanted user can' use it in website.

r) We can use WNOSQL (.wdba) data for futhure use with remote database WDBAJ$ at any linux type OS.

s) It is easy to use and Learnable

t) We can also do programming in WNOSQL database

u) We can also construct forms and reports

v) Here CDollar-JAVA.util packages are used

w) No Server for WNOSQL(*) db

x) Occupies only less amount of safe

y) Prevents from data loss by CLUSTERRESTORE

z) Performs Manipulations (WNOSQL(*)) in
huge amount of data say 1 thrillion.
never makes db very slow since datas are divided into
batches.

==========

UNIT:2: WNOSQL(WSQL*) Basics

The WNOSQL SQL(*) BASICS

1) CREATETABLE

CREATETABLE from Tablename index1 to lastindex1 , row to cols ?= 0 By 0 f(x) : {FIELDSNAMES}:{Fieldvaluesset1 , Fieldvaluesset2 .....} :{0};

Explanation

CREATETABLE from Tablename which is used to create a table by given rows and cols with Tablefields and Table field value....

2) SelectRowVAL

SELECTRVAL from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}

Explanation

SELECTRVAL is used to list all values from Tablename by given rows and cols.

3) DELETE

DELETE from Tablename index1 to lastindex1 , row to col ?= Character By X X : {value} : {0} :{0}

Explanation

DELETE is used to delete a particular value from a table
by given rows and cols.

4) SYSDATE

SYSDATE from Tablename index1 to lastindex1 , row to col ?= DATEFORMAT By X X : {0} :
{0} :{0}

Explanation

It is used to return SYSTEM DATE [according to DATEFORMAT- Optional]

5) INSERTINTO STATEMENT

INSERTINTO from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} :
{rowsetvalues1-- rowsetvaluesn} : {0}";

Explanation

It is used to insert rowset values from Tablename
by given rows and cols.

6) MATCH STATEMENT

MATCH from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}
Explanation:
IT is used to list row pairs where the given Character is matched
from Tablename by given rows and cols.

7) SelectOrderbyAsc

SelectOrderByASC from index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}

Explanation:

IT is used to list the values by Ascending order
from Tablename by given rows and cols.

8) SelectOrderbyDesc

SelectOrderByDESC from index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}

Explanation:

IT is used to list the values by Descending order
from Tablename by given rows and cols.

9) SelectIntOrderByAsc

SelectOrderByASC from index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}

Explanation:

IT is used to list the values by Ascending order
from Tablename by given rows and cols.

10) SelectIntOrderByDesc

SelectOrderByDESC from index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}

Explanation:

IT is used to list the values by Descending order
from Tablename by given rows and cols.

11) SelectALL Statement

SelectAll from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} :{0}

Explanation

SelectAll is used to list all values from Tablename
by given rows and cols.
but SelectRval is used to compute the size of ArrayList.

12) Select Statement

Select from Tablename index1 to searchselectindex , row to col ?= Character By X X : {0} : {0} :{0}

Explanation

Select is used to list a particular value using searchselectindex from Tablename
by given rows and cols.

13) Search Statement

Search from Tablename index1 to indexn , row to col ?= Character By X X : {0} : {0} :{0}

Explanation

=======

Search is used to list a arraylist value when given Character or a number is found f rom Tablename by given rows and cols.

14) SearchGT Statement

SearchGT from Tablename index1 to indexn , row to col ?= Searchednumber By X X : {0} : {0} :{0}

Explanation

SearchGT is used to list the values which is greater than a Searchednumber from the Tablename by given rows and cols.

15) SearchLS Statement

SearchLS from Tablename index1 to indexn , row to col ?= Searchednumber By X X : {0} : {0} :{0}

Explanation

SearchLS is used to list the values which is less than a Searchednumber from the Tablename by given rows and cols.

16) SelectRange Statement

SelectRange from Tablename indexrange1 to indexrangen , row to col ?= Char By X X : {0} : {0} :{0}

Explanation

SelectRange is used to list the values according to given indexranges(indexrange1 to indexrangen) from the Tablename by given rows and cols.

17) SelectAssign Statement

SelectAssign from Tablename indexrange1 to indexrangen , row to col ?= ASSIGNEDVALUE By X X : {0} : {0} :{0}

Explanation

SelectAssign is used to Assign the values

18) SelectRows Statement

SelectRows from Tablename x1 to x2 , row to col ?= X By X X : {0} : {0} :{0}

Explanation

SelectRows is used to list the Assigned Row value

19) RIGHTJOIN Statement

RIGHTJOIN from Tablename1 x1 to x2 , row to col ?= Tablename2 By 1 1 : {rowindexes} :{rowindexes} : {0} ;

Explanation:

It is used to print Right Table of the SQL TABLE PLUS those rows of Lefttable did not match with rows of Right table.

20) LEFTJOIN Statement

LEFTJOIN from Tablename1 x1 to x2 , row to col ?= Tablename2 By 1 1 : {rowindexes} :{rowindexes} : {0} ;

Explanation:

It is used to print LEFT Table of the SQL TABLE PLUS those rows of Lefttable did not match with rows of RIGHT table.

21) INNERJOIN Statement

INNERJOIN from Tablename1 x1 to x2 , row to col ?= Tablename2 By 1 1 : {rowindexes} :{rowindexes} : {0} ;

Explanation:

InnerJoin means the intersection between two tables.

ie) the common rows.

22) SelectIN Statement

SelectIN from Tablename1 x1 to x2 , row to col ?= MEMBERDATA By 1 1 : {0} :{0} : {0} ;

Explanation:

========

SelectIn data is used to test whether the given member data is found or not then print that data.

23) SelectNOTIN Statement

SelectNOTIN from Tablename1 x1 to x2 , row to col ?= MEMBERDATA By 1 1 : {0} :{0} : {0} ;

Explanation:

SelectIn data is used to test whether the given member data is found or not then print that data.

24) Count(*)

Count(*) from Tablename1 x1 to x2 , row to col ?= X By 1 1 : {0} :{0} : {0} ;

Explanation:

It is used to list no of rows in table tablename1.

not then print that data.

24) ENCRYPT

Encrypt from Tablename1 x1 to x2 , row to col ?= X By 1 1 : {0} :{0} : {0} ;

Explanation:

It is used to Encrypt a table tablename1.

25) DECRYPT

Decrypt from Tablename1 x1 to x2 , row to col ?= X By 1 1 : {0} :{0} : {0} ;

Explanation:

It is used to Decrypt a table tablename1.

26) SelectCols

SelectCols from Tablename1 x1 to x2 , row to col ?= X By 1 1 : {column indexes} :{0} : {0} ;

Explanation

It is used to list the column values according to column indexes.

27) Distinct

DISTINCT from Tablename1 x1 to x2 , row to col ?= X By 1 1 : {column indexes} :{0} : {0} ;

Explanation

It is used to remove duplicate column values according to column indexes.

28) INSERT STATEMENT

Insert from Tablename index1 to lastindex1 , row to col ?= Character By X X : {rowsetvalues1--rowsetvaluesn} : {0} : {0}";

Explanation

It is used to insert rowset values from Tablename

by given rows and cols.

29) SelectUPPER Statement

SelectUPPER from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0}";

Explanation

It is used to list the values from TableName in Uppercase

by given rows and cols.

30) SelectLOWER Statement

SelectLOWER from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0}";

Explanation

It is used to list the values from TableName in Lowercase

by given rows and cols.

31) PRIMARYKEY

PrimaryKey from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0};

Explanation

It is used to remove the Duplicate values or print unique values from TableName

by given rows and cols.

32) InsertDESC

InsertDESC from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0};

Explanation:

It is used to insert table describtion

32) SelectDESC

SelectDESC from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0};

Explanation:

It is used to list about table description.

33) SelectC*

SelectC* from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0};

Explanation

It is used to Count no of columns present in the table

33) SelectR*

SelectR* from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} : {0} : {0};

Explanation

To compute howmany rows in field table use SELECTR* and display it in table format

34) <HAVING> Clause

<HAVING> from Tablename index1 to lastindex1 , row to col ?=[Rowindexes1],[Rowindexes] 1 1 : {0} : {Rowsetvalues} :{0}

Explanation

It is used to perform aggregate f(x) to column and in condition (eg) (SUM)(fields) > 0 and combine the join results ..

35) UPDATE STATEMENT

UPDATE from Tablename index1 to lastindex1 , row to col ?= X By 1 1 : {value1} :{value2}:{0}

Explanation

It is used to Update value1 to value2 from Tablename

by given rows and cols.

36) SelectLike Statement

SelectLike from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} :{0}:{0}

Explanation:

Display all names with middle name , last, firstname

37) LOC Statement

Loc is used to find the given data stored in a location in a table.

SYNTAX:

LOC from Tablename index1 to lastindex1 , row to col ?= Character By X X : {0} :{0}:{0}

38) AVG()

AVG() from Tablename index1 to lastindex1 , row to col ?= X By X X : {numbervalues} :{0}:{0}

Explanation:

It is used to compute Avg of given values...

39) MAX()

MAX() from Tablename index1 to lastindex1 , row to col ?= X By X X : {numbervalues} :{0}:{0}

Explanation:

It is used to compute MAX of given values...

40) MIN()

MIN() from Tablename index1 to lastindex1 , row to col ?= X By X X : {numbervalues} :{0}:{0}

Explanation:

It is used to compute MIN of given values...

41) SUM()

SUM() from Tablename index1 to lastindex1 , row to col ?= X By X X : {numbervalues} :{0}:{0}

Explanation:

It is used to compute SUM of given values...

42) DATACOMPARE (ASC/DESC)

i) DateCompareDESC from DATES index1 to index2 , rows to cols ?= X By XX : {Indexvalues} :{0}:{0};

ii) DateCompareASC from DATES index1 to index2 , rows to cols ?= X By XX : {Indexvalues} :{0}:{0};

This statemets i) and ii) is used to List Dates in Ascending or Descending order according to IndexValues.

43) CLUSTER DEMOS

a) **CLUSTER:**

CLUSTER from index1 to index2 , rows to cols ?= x By x f(x) : {DATAVALUES}: {0} :{0}

Explanation:

To Store CLUSTER of Data form a
given range in a encrypted and retrieve from encrypted file...

b) **CLUSTERPROPERTY**

CLUSTERPROPERTY from index1 to index2 , rows to cols ?= x By x f(x) : {DATAVALUES}: {0} :{0};

Explanation

To compute clustertable size, display data, display system date, Display remaning space available to store values in a cluster table.

c) **BACKUPCLUSTER :**

BACKUPCLUSTER from index1 to index2 , rows to cols ?= x By x f(x) :{DATAVALUES}: {0} :{0};

Explanation

TO RESTORE the Lost CLUSTER DATA and automatically store the contents in a table.

What are the Things Needed to execute the Query?

Step-1:

```
String g = WDBASQL.WDBASQLS("databasename", "USEDATABASE", "dbpasswordtablename",
"pathwhereserverdataisstored");
```

Step-2:

```
String t = WDBASQL.WDBASQLS("dbusertable", "dbpwdtable", 1, "username", "password",
1, 5, g);
```

Step-3:

```
char c=' ';

String query ="WNOSQL DB STATEMENTS";
```

Step-4:

```
WDBA.writeln(""+WDBALIB.WDBAQUERY( query, t));
```

==========

UNIT:3- WNOSQL(PLSQL*) Basics

==========

The WNOSQL PLSQL statements

==========

The WnoSql statements which is listed which are given below.

Insert => Insert values into the table and create a new table

eg-1)

```
WDBASQL.Query("Insert","table" ,"",values,0,"","", null,"",0," ","",c,null,t,rows,cols);
```

```
WDBA.writeln((manipulate.Signal("MANIPULATE","Select query"
,"tablename","column1,column2,...","?,?...",rows+1,"Drivers","datasource","username","password
",
"newencrytedtable")));
```

Select All => Select all rows from the table from a given range

eg)

```
WDBASQL.Query("SelectAll","tablename" ,range1,null,range2,"","", null,"",0,"
","",c,null,t,rows,cols);
```

Select => Select a particular column (key) values from a table

```
WDBASQL.Query("Select","tablename" ,range1,null,key,"","", null,"",0," ","",c,null,t,rows,cols);
```

DateCompare => It is used to compare dates

WDBA.writeln("value="+WDBASQL.Query("DateCompare","datetable"
,"datetable2",null,noofcolumns,"datetable2",pwd, null,"",0," ","",c,null,t,rows,cols));

DateCompareAsc => it is used to compare dates and sort the date in ASC

WDBA.writeln("value="+WDBASQL.Query("DateCompareASC","datetable"
,"datetable2",null,noofcolumns,"datetable2",pwd, null,"",0," ","",c,null,t,rows,cols));

SelectRange => It is used to select column values with in a range

WDBA.writeln("value="+WDBASQL.Query("SelectRange","tablename"
,"datarange1",null,noofcolumns,"datarange2",pwd, null,"",0," ","",c,null,t,rows,cols));

Search => Search a data with in a table

WDBA.writeln("value="+WDBASQL.Query("Search","tablename" ,"0",null,totalrows,"data",pwd,
null,"",0," ","",c,null,t,rows,cols));

SearchGT => Search data which is greater than a given data

WDBA.writeln("value1="+WDBASQL.Query("SearchGT","tablename" ,"0",null,totalrows,"data",pwd,
null,"",0," ","",c,null,t,rows,cols));

SearchLS =>Search data which is lesser than a given data

WDBA.writeln("value2="+WDBASQL.Query("SearchLS","tablename" ,"0",null,totalrows,"data",pwd,
null,"",0," ","",c,null,t,rows,cols));

AVG => it is used to compute the avg of the given datas

WDBASQL.Query("AVG","tablename" ,"",null,noofcols,"","", null,"",0," ","",c,null,t,rows,cols);

MAX => it is used to find the max of the given datas

WDBASQL.Query("MAX","tablename" ,"",null,noofcols,"","", null,"",0," ","",c,null,t,rows,cols);

MIN=> it is used to find the max of the given datas

WDBASQL.Query("MIN","tablename" ,"",null,noofcols,"","", null,"",0," ","",c,null,t,rows,cols);

DSerialize=> it is used to deserialize and retrieve the data

WDBASQL.Query("DSerialize","wilmix" ,"","",0,"",pwd, null,"",0," ","",c,null,t,rows,cols);

Update => Insert a value into table but the last columns values be deleted

WDBASQL.Query("Update","tablename" ,"datatobeupdated",null,nooforows,"newdata",pwd, null,"",0," ","",c,null,t,rows,cols);

Delete => Delete a particular data or delete all the datas

WDBASQL.Query("Delete","tablename" ,"datatobeupdated",null,nooforows,"newdata",pwd, null,"",0," ","",c,null,t,rows,cols);

InsertDB => Store the data in WDBA file

WDBASQL.Query("INSERTDB","tablename" ,"",String,0,"",pwd, null,"",0," ","",c,null,t,rows,cols);

SelectIntOrderByAsc => Sort all Int data in Table by Ascending order

WDBASQL.Query("SelectIntOrderByAsc",tablename,startindex ,null,endingindex,"123","",null,"",0,"","",char,null,password,row,cols);

SelectIntOrderByDesc => Sort all Int data in Table by Descending order

WDBASQL.Query("SelectIntOrderByDesc",tablename,startindex
,null,endingindex,"123","",null,"",0,"","",char,null,password,row,cols);

SelectOrderByAsc => Sort all data in Table by Ascending order

WDBASQL.Query("SelectOrderByASC",tablename,startindex
,null,endingindex,"123","",null,"",0,"","",char,null,password,row,cols);

SelectIntOrderByDESC => Sort all data in Table by Ascending order

WDBASQL.Query("SelectOrderByDESC",tablename,startindex
,null,endingindex,"123","",null,"",0,"","",char,null,password,row,cols);

Insert => Insert the values from arraylist to the tablename

WDBASQL.Query("Insert",tablename ,"",arraylist,0,"","", null,"",0,"
","",char,null,password,row,cols);

SelectIN => Display or check whether the member is present in the table.

WDBASQL.Query("SelectIN",tablename
,startindex,null,endingindex,data,"",null,"",0,"","",char,null,password,row,cols);

SelectNOTIN => Display all member is present in the table.

WDBASQL.Query("SelectNOTIN",tablename
,startindex,null,endingindex,data,"",null,"",0,"","",char,null,password,row,cols);

SelectLike => Display all names with middle name , last, firstname

WDBASQL.Query("SelectLike",tablename
,startindex,null,endingindex,"","",null,"",0,"","",char,null,password,row,cols);

Count(*) => to count no of rows in the table

WDBASQL.Query("Count(*)",tablename ,"0",null,0,"","",null,"",0,"","",c,null,password,row,cols);

MATH => Apply mathematical functions in a table

WDBASQL.Query("MATH",tablename
,"0",null,0,"0","",null,"",0,"",function,c,null,password,row,cols);

Encrypt => Encrypt the table

WDBASQL.Query("Encrypt",tablename ,"0",null,0,"5","",null,"",0,"","",c,null,password,row,cols);

Dencrypt => Dencrypt the table

WDBASQL.Query("Dencrypt",tablename ,"0",null,12-
3,"5","",null,"",0,"","",c,null,password,row,cols);

Droptable => Drop the table

WDBASQL.Query("DropTable",tablename ,"0",null,0,"0","",null,"",0,"","",c,null,password,row,cols);

INSERTINTO => Insert Arraylist into tablename

WDBASQL.Query("INSERTINTO",tablename
,startindex,null,endingindex,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

DeleteAll=> Delete All the contents from table

WDBASQL.Query("DeleteAll",tablename ,"0",null,0,"","",null,"",0,"","",c,null,password,row,cols);

AVG=> AVG of nos from table

WDBASQL.Query("AVG()",tablename
,startindex,null,endingindex,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

MAX=> MAX of nos from table

WDBASQL.Query("MAX()",tablename
,startindex,null,endingindex,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

MIN=> MIN of nos from table

WDBASQL.Query("MIN()",tablename
,startindex,null,endingindex,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

MAX=>SelectColumns between range startingindex and endingindex

depends upon the the given arryalist values from table

WDBASQL.Query("SelectCols",tablename,startindex,null,endingindex,"0","",ARRAYLISTINSERTION
,"",0,"","",c,null,password,row,cols);

Count() => to count the occurance of data with in a given range

startingindex and endingindex from the table.

WDBASQL.Query("Count()",tablename
,startindex,null,endingindex,data,"",null,"",0,"","",c,null,password,row,cols);

DISTINCT => is used to remove duplicates from the table

WDBASQL.Query("DISTINCT",tablename ,startindex,null,endingindex,"","",
ARRAYLISTINSERTION,"",0,"11","",c,null,password,row,cols);

SUM() => To find the sum of nos in a given

Arraylist column indexes.

WDBASQL.Query("SUM()",tablename
,null,null,null,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

LOC() => is used to find the given data stored in a location in a table.

WDBASQL.Query("LOC()",tablename
,startindex,null,endingindex,data,"",null,"",0,"","",c,null,password,row,cols);

MATCH() => to get match columns locations in a arraylsit of
values with in a range matching the given data

WDBASQL.Query("MATCH",tablename
,startindex,null,endingindex,data,"",null,"",0,"","",c,null,password,row,cols);

DateCompareDESC => To sort the given dates present in a table by ascending order.

WDBASQL.Query("DateCompareDESC",tablename
,startindex,null,endingindex,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

DateCompareDESC => To sort the given dates present in a table by Decending order.

WDBASQL.Query("DateCompareASC",tablename
,startindex,null,endingindex,"","",ARRAYLISTINSERTION,"",0,"","",c,null,password,row,cols);

INNERJOIN => Join two table based on the matching column values

WDBASQL.Query("INNERJOIN",tablename1,"0",null,0,tablename2,"",ARRAYLISTINSERTION1,"",0,"","",c,ARRAYLISTINSERTION2,password,row,cols);

InsertDesc = > Create table fields

WDBASQL.Query("InsertDESC", tablename, "0", ARRAYLISTINSERTION, 0, "", "", null, "", 0, "", "", c, null,password,row,cols);

INSETINTO = > Used to insert a arraylist collection values in the table

this is mostly used for insertion.

WDBASQL.Query("INSERTINTO", tablename, "0",null,

arraylistsize(),"0","",null,"",0,"","",c,ARRAYLISTINSERTION2,password,row,cols);

SELECTRVAL =>Selectall values from rows in a table and compute the size.

WDBASQL.Query("SELECTRVAL",tablename, "0", null, 0, "0", "", null, "", 0, "", "", c,null,password,row,cols);

LEFTJOIN => The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

WDBASQL.Query("LEFTJOIN",tablename1,"0",null,0,tablename1,"",
ARRAYLISTINSERTION1,"",0,"","",c,ARRAYLISTINSERTION2,password,row,cols);

RIGHTJOIN =>The RIGHT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

WDBASQL.Query("RIGHTJOIN",tablename1,"0",null,0,tablename1,"",
ARRAYLISTINSERTION1,"",0,"","",c,ARRAYLISTINSERTION2,password,row,cols);

AND => is used to combine two results of Query and put

them into arraylist for futhure use.

WDBASQL.Query("AND", "", startindex,null,endingindex, "", "", ARRAYLISTINSERTION1, "", 0, "", "", c,ARRAYLISTINSERTION2,password,row,cols);

Foreignkey => It is used to set Foreign key between startindex and endingindex

of the table ; but it will also accept null and duplicate values.

WDBASQL.Query("ForeignKey",tablename1, startindex,null,endingindex, "", "", null, "", 0, "", "", c, null,password,row,cols);

HAVING => it is used to perform aggregate f(x) to column and in condition (eg) (SUM)(fields) > 0 and combine the join results ..

WDBASQL.Query("<HAVING>",tablename1,
startindex,null,endingindex,"[col1,col3..],[col1,col3..]","",ARRAYLISTINSERTION1,"",0,"","",
c,ARRAYLISTINSERTION2,password,row,cols);

To compute howmany rows in field table use SELECTR* and display it in table format

WDBASQL.Query("SELECTR*",tablename ,"0",null,0,"0","",null,"",0,"","",c, null,password,row,cols);

Select all Row values for given Row list value use SELECTROWS for the given range

WDBASQL.Query("SELECTROWS",tablename ,"0",null,0,"0","",null,"",0,"","",c,
null,password,row,cols);

Select all Column values for given column list Indexed value use SELECTCOLUMNS for the given range

WDBASQL.Query("SELECTCOLUMNS",tablename ,"0",null,0,data,"",INSERTIONARRAYLIST,"",0,"","",
c, null,password,row,cols);

Select all Row values for given row list value based on indexes for given range and increment the counter based on counter value

WDBASQL.Query("SELECTINDEXES",tablename ,"0",null,0,data,"",INSERTIONARRAYLIST,"",0,"","",c , null,password,row,cols);

Encode String to numbers

WDBASQL.Query("SelectAssign",tablename1, assignindex,null,endingindex,"1","",null,INSERTIONDATALIST,0,"","",c, null,password,row,cols);

InsertValues into the table

WDBASQL.Query("InsertValues",tablename1, startindex,null,endingindex,"","",null,INSERTIONDATAARRAYSTRINGLIST,0,"","INSERT1",c, null,password,row,cols);

To Count no of columns present in the table

WDBASQL.Query("SELECTC*",tablename ,"0",null,0,"0","",null,"",0,"","",c, null,password,0,1);

Select particular column value from the table

WDBASQL.Query("Select",tablename ,"0",null,endingindex,"0","",null,"",0,"","",c, null,password,0,1);

PrimaryKey => it is used to remove duplicates and null values

WDBASQL.Query("PrimaryKey",tablename1, startindex,null,endingindex, tablename1, "", null, "", 0, "", "", c, null,password,row,cols);

SelectUPPER => Select all the values in a given range from

table and convert to upper case.

WDBASQL.Query("SelectUPPER",tablename , startindex,null,endingindex,"","",INSERTIONARRAYLIST,"",0,"","",c, null,password,row,cols);

SelectLOWER => Select all the values in a given range from

table and convert to upper case.

WDBASQL.Query("SelectLOWER",tablename ,
startindex,null,endingindex,"","",INSERTIONARRAYLIST,"",0,"","",c, null,password,row,cols);

To display system date and time

WDBASQL.Query("SYSDATE","" ,"",null,0,"","",null,"",0,"","",c, null,password,row,cols);

To display dateandtime in a format given

WDBASQL.Query("ManipulateDate()","" ,
startindex,null,endingindex,"","",INSERTIONARRAYLIST,"",0,dateformatstring,"",c,
null,password,row,cols);

To Store CLUSTER of Data form a
given range in a encrypted and retrieve from encrypter file...

WDBASQL.Query("CLUSTER",tablename , startindex,null,endingindex,"","",
INSERTIONARRAYLIST,"",0," ","",c, null,password,row,cols);

TO RESTORE the Lost CLUSTER DATA and automatically store the
contents in a table.

WDBASQL.Query("BACKUPCLUSTER",tablename ,startindex,null,endingindex,"","", null,"",0," ","",c,
null,password,row,cols);

To compute clustertable size, display data, display system date, Display remaning
space available to store values in a cluster table.

WDBASQL.Query("CLUSTERPROPERTY",tablename ,startindex,null,endingindex,"","", null,"",0,"
","",c, null,password,row,cols).get(row,cols);

To convert date to calendar

DatetoCalendar() => Inputstring(fx,data)

To convert date to String

DatetoString() =>Inputstring(f(x) ,data)

To convert calendar to date

CalendartoDate()=>

To return dateand time in a given format for the columns (0-year, 1-month,2-year)=>
Specify this in arraylist.

getCalender() => Inputstring(f(x),columns)

To convert String to date

StringtoDate()=> Inputstring(f(x) ,data)

To search a data with in a given range
Search=> Inputstring(data,i111,key)

To set triggers and allias for the table.
InsertAllias,InsertTriggers =>Inputstring(obj)

To select all the Allias from the table
SelectAllias =>Inputstring(cmd,key)
To check whether given data is greater than or less than from
the range of values from the table.
SearchGT,SearchLS => Inputstring(i111,key,cmd)

To select all the Triggers from the table

SelectTriggers => Inputstring(cmd,key)

To select all the columns or field names from the table.

SelectDESC => Inputstring(0 to key)

To Serialize or Deserialize a table.

DSerialize,Serialize()=> Inputstring(cmd)

To select all the range values from the table

SelectRange()=>Inputstring(i111 to key)

To Delete all the contents form the table.

DeleteAll => Inputstring(cmd)

To Delete a range from a table

WDBASQL.Query("DELETE", "tablename", "0", null, 0, "0", "", columns, "", 0, "", "", c, null, password, rows, cols);

To Recall the values stored in the file and convert to table format.

RECALL => Inputstring(cmd)

To Insert values in .wdba file and retrieve it...

INSERTDB => Inputstring(cmd)

To create an empty table with fields

WDBASQL.Query("CREATETABLE", "tablename", "0", null, 0, "0", "", arraylisttablename, "", 0, "", "", c, arraylistintialvalues, t, rows,cols);

TO insert values into batch table.

WDBASQL.Query("INSERTBATCHTABLE", "tablename", "0", null, 0, "0", "", arraylisttablenames, "", 0, "", "", c, arraylisttablevalues, t, rows,cols);

To insert password into password table

WDBASQL.Query("Password", "tablename", "0",passwordarraylist, 0, "0", "", null, "", 0, "", "", c, null, t, rows,cols);

To insert username into username table

WDBASQL.Query("Username", "tablename", "0",usernamearraylist, 0, "0", "", null, "", 0, "", "", c, null, t, rows,cols);

a) CREDENTIALS methods and how you create a database ?

==========

WDBASQL.Query("CreateDatabase","datastorehgh" ,"0",dbpwd,0,"","", null,"",0," ","",c,null,t,0,0);

String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE", "wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);

ArrayList ar788=WDBASQL.Query("Password", "dbpwds", "0",ar9s, 0, "0", "", null, "", 0, "", "", c, null, t, 1,5);

ArrayList ar788=WDBASQL.Query("Username", "dbuser", "0",ar9s, 0, "0", "", null, "", 0, "", "", c, null, t, 1,5);

default will be "Username" for username and pwd for password

i) COMMIT for dropping the table

WDBASQL.Query("commit", tablename, "0", null, 0, "", "", null, "", 0, "", "", char, null, pasword, rows, cols);

j) CLUSTER to store group of data in a encrypted form for futhure use.

WDBASQL.Query("CLUSTER",tablename ,rangestart,null,rangeend,"","", arraylistdata,"",0," ","",char, null, pasword, rows, cols);

Most Important Method API for WNOSQL PLSQL ODBCDRIVER

public static ArrayList deserialize(String line, String cmd,String i111,Object obj,int key, String data,ArrayList columns,String values,int sheetno,String fx,String sname,char c2,ArrayList columns1,int rc , int cc)

line =>command

cmd => table

i111=> startingrange

obj => To pass any datatype

key => endingrange

data=> data to be searched or manipulated

Columns => table1 columns indexed values to be manipulated

values => Insert values

int sheetno=> sheetno (default must be 0)

String fx=> mainly used for math functions

String sname => Sheetname

char c2=> Mostly used in character occurance testing

ArrayList columns1=>table1 columns indexed values to be manipulated

int rc => rows

int cc=> cols

Note: You can use this api to pass the Inputstring parameters

WDBASQL.QUERY(????)

FAQS?

How WNOSQL DB store and retrieve the data? how it restore the data ?

WNOSQL stores the data in the form of cluster.

Every database file .wdba has .cluster file and

WNOSQL database .wdba
contents is first passed to cluster datastructure and such data is
stored in a cluster form and cluster encrypted form by using CLUSTER QUERY.

CLUSTERSTORE will retrieve the contents and put the contents
again in .wdba file for futhure use.

If you lost all the data no need to worry about it.

just pass BACKUPCLUSTER with clustername in WNOSQLQUERY..

.wdba and cluster files with datas are restored

from cluster encrypted file.

so don't delete cluster encrypted files in c:/Program/CDollar...

==========

UNIT:4-> WNOSQL(WSQL*) Forms and Reports

==========

WDBA.writeln("html tags");

HTML.displayhtml("html file");

<TRY> => try block

<CATCH> => Catch block in wnosql

<EXE> => Exception

SYNTAX:

=======

<WNOSQL>

<PACK>

<DATALIB> namespacename

<DATA>

public <CLASS> classname

{

public void main()

{

```
<!WNOSQL Statements !>

}
}

</DATA>
```

EXAMPLE:

========

```
<WNOSQL>
<PACK>
<DATALIB> namespacename
<DATA>

public class WDBALogin --> <Serialize>
{
public void main( ) throws <EXE> // main program with throw //Exception
{
<TRY>
{


WDBA.writeln("<html><head><title>WDBA LOGIN</title> </head>");


WDBA.writeln("<body class=fancy>");
```

```
WDBA.writeln("<form action=http://localhost:5000/view1.WNOSQL method=post >");


WDBA.writeln("<div id=pageContainer>");
    WDBA.writeln("<img src=images/banner1.png alt=WDBA Login @ Jemin Information
Technology (C) ALL RIGHTS RESERVED>");
    WDBA.writeln("<div id=pageContent>");


     WDBA.writeln("<div id=chaptersAccordion>");


        WDBA.writeln("<h2><a href=#chapter1>Enter your System Details</a></h2>");
        WDBA.writeln("<div>");



WDBA.writeln("<p>Enter your Username : <input type=text name=uname size=15/></p>");
WDBA.writeln("<p>Enter the password : <input type=password name=password size=25
/></p>");
WDBA.writeln("<p>Enter the TABLE NAME : <input type=password name=table11 size=25
/></p>");
WDBA.writeln("<p>Enter the system password : <input type=password name=spwd size=25
/></p></div>");
WDBA.writeln("<div><input type=submit name=Click><input type=reset
name=Clear></div>");
WDBA.writeln("</form></body></html>");


}
<CATCH>(<EXE> e){}
}
}
</DATA>
</WNOSQL>
```

Explanation:

========

All WNOSQL program should begin with <WNOSQL>
and <PACK> is used to import utilities packages like
arraylist , linked lsit etc.

And <DATALIB> is namespace for WNOSQL.

and <DATA> represents the Logic of WNOSQL
oops plsql logic.

WDBA.Writeln => to print the string in webconsole

HTML.displayhtml("htmlfile")=> it is used
to display html forms and reports.

==========

UNIT:5-> WNOSQL(WSQL*) interaction with native databases

==========

For Oracle Server

WDBA.writeln(""+manipulate.Signal("MANIPULATE","Select * from student","student","sno,tmark,rank","?,?,?",4,"oracle.jdbc.driver.OracleDriver","jdbc:oracle:thin:@localhost:1521:xe","system","jemin","wilmix2"));

FOR SQL SERVER

==========

WDBA.writeln(""+manipulate.Signal("MANIPULATE","Select * from student","student","sno,tmark,rank","?,?,?",4,"sun.jdbc.Odbc.JdbcOdbcDriver","jdbc:odbc:sa","system","jemin","wilmix2"));

FOR MYSQL

========

Use MYSQL database driver instead of sun.jdbc.Odbc.JdbcOdbcDriver.

note: select the following dll which is given below

manipulate.dll , ikvm.open.jdbc.dll to compile this WNOSQL plsql program.

UNIT:6-> WNOSQL(WSQL*) TEST Exercises

1) Write a WNOSQL Program to select all students from a student table?

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA; //load wnosql packages

<DATALIB> ps
<DATA>
public <CLASS> SQL3
{



public void main()



{


String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
WDBASQL.Query("SelectAll","student" ,"0",null,15,"","",, null,"",0," ","",c,null,t,1,1);
```

```
}
}

</DATA>
```

2) **Apply SelectOrderByASC to the PLSQL to table Orders for 0 to 19 records what happens?**

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA; //load CDollar.WDBA libraries
<DATALIB> ps // namespace ps
<DATA>
public <CLASS> DATA

{



public void main()

{
```

```
String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char  c=' ';

WDBASQL.Query("SelectOrderByASC","Orders","0" ,null,19,"123","",null,"",0,"","",c,null,t,1,1);

}




}

}
```

==========

UNIT:7 : WNOSQL(WSQL*) PLSQL Programming

WNOSQL(*) PLSQL follows basic WNOSQL(*) api syntax WNOSQL(*) put all the results in arraylist

for future use.

Program-1:

Write a program to display the matching data rows and

perform innerjoins between two table.

<WNOSQL> // Beginning of wnosql plsql program
<PACK> //load all wnosql packages
<USE> CDollar.WDBA; //use CDollar.WDBA packages
<DATALIB> ps // create name space ps
<DATA> //write wnosql logic
public <CLASS> DATA

{

```
public void main() //like  C main

{

//kindly  refer  wnosql  fundemantals


String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);

char  c=' ';


ArrayList arhd1gy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0001","
",null,"",0,"1","",c,null,t,1,1);


ArrayList arhd1gy1 =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0002","
",null,"",0,"1","",c,null,t,1,1);


ArrayList arhd1gyy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0003","
",null,"",0,"1","",c,null,t,1,1);


ArrayList arhd1gy1y =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0005","
",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList  artr11= new  ArrayList();

for(int  i=0;i<arhd1gy.size();i++)
artr11.add(arhd1gy.get(i));

for(int  i=0;i<arhd1gy1.size();i++)
artr11.add(arhd1gy1.get(i));

for(int  i=0;i<arhd1gyy.size();i++)
artr11.add(arhd1gyy.get(i));

for(int  i=0;i<arhd1gy1y.size();i++)
artr11.add(arhd1gy1y.get(i));
```

```
ArrayList arhd1gy17 =WDBASQL.Query("MATCH","employess" ,"0",null,11,"0001","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy117 =WDBASQL.Query("MATCH","employess" ,"0",null,11,"0002","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy178 =WDBASQL.Query("MATCH","employess" ,"0",null,13,"0003","
",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList arhd1gy1178 =WDBASQL.Query("MATCH","employess" ,"0",null,13,"0005","
",null,"",0,"1","",c,null,t,1,1);



ArrayList  artr117= new  ArrayList();

for(int  i=0;i<arhd1gy17.size();i++)
artr117.add(arhd1gy17.get(i));

for(int  i=0;i<arhd1gy117.size();i++)
artr117.add(arhd1gy117.get(i));
for(int  i=0;i<arhd1gy178.size();i++)
artr117.add(arhd1gy178.get(i));

for(int  i=0;i<arhd1gy1178.size();i++)
artr117.add(arhd1gy1178.get(i));

ArrayList  datas1=WDBASQL.Query("INNERJOIN","Orders","0",null,19,"employess","",
artr11,"",0,"","",c,artr117,t,1,1);



}

}



}
```

Kindly use WNOSQL EDITOR to see the output.

Program-2: Write a Program to finding matching data rows
and perform right join, use having clause , use innerjoin in this case:

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{




String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char c=' ';


ArrayList arhd1gy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0001","
",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList arhd1gy1 =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0002","
",null,"",0,"1","",c,null,t,1,1);



ArrayList arhd1gyy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0003","
",null,"",0,"1","",c,null,t,1,1);



ArrayList arhd1gy1y =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0005","
",null,"",0,"1","",c,null,t,1,1);




ArrayList  artr11= new  ArrayList();

for(int  i=0;i<arhd1gy.size();i++)
artr11.add(arhd1gy.get(i));

for(int  i=0;i<arhd1gy1.size();i++)
artr11.add(arhd1gy1.get(i));

for(int  i=0;i<arhd1gyy.size();i++)
artr11.add(arhd1gyy.get(i));

for(int  i=0;i<arhd1gy1y.size();i++)
artr11.add(arhd1gy1y.get(i));
```

```
ArrayList arhd1gy17 =WDBASQL.Query("MATCH","employess" ,"0",null,11,"0001","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy117 =WDBASQL.Query("MATCH","employess" ,"0",null,11,"0002","
",null,"",0,"1","",c,null,t,1,1);



ArrayList arhd1gy178 =WDBASQL.Query("MATCH","employess" ,"0",null,13,"0003","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy1178 =WDBASQL.Query("MATCH","employess" ,"0",null,13,"0005","
",null,"",0,"1","",c,null,t,1,1);



ArrayList  artr117= new  ArrayList();

for(int  i=0;i<arhd1gy17.size();i++)
artr117.add(arhd1gy17.get(i));

for(int  i=0;i<arhd1gy117.size();i++)
artr117.add(arhd1gy117.get(i));
for(int  i=0;i<arhd1gy178.size();i++)
artr117.add(arhd1gy178.get(i));

for(int  i=0;i<arhd1gy1178.size();i++)
artr117.add(arhd1gy1178.get(i));
```

```
ArrayList  cols = new  ArrayList();

cols.add(0);
cols.add(1);
cols.add(2);
cols.add(3);
cols.add(4);
cols.add(5);
//cols.add(6);
//cols.add(7);
//cols.add(8);
//cols.add(9);
//cols.add(10);
//cols.add(11);

cols.add(0);
cols.add(1);
cols.add(2);


ArrayList  cols111 = new  ArrayList();
cols111.add(0);

cols111.add(1);


cols111.add(2);
cols111.add(3);
cols111.add(4);
cols111.add(5);
//cols111.add(6);
```

```
//cols111.add(7);
//cols111.add(8);
cols111.add(9);
cols111.add(10);
cols111.add(11);



ArrayList datas44=WDBASQL.Query("RIGHTJOIN","Orders","0",null,0,"employess","",
cols,"",0,"","",c,cols111,t,1,1);



ArrayList colss7 = new ArrayList();



colss7.add(2);

ArrayList datas16=WDBASQL.Query("<HAVING>","Orders","0",null,1,"[3,6,2,2],[2,5,2,2]","",colss
7,"",0,"","",c,datas44,t,1,1);



ArrayList datas1=WDBASQL.Query("INNERJOIN","Orders","0",null,19,"employess","",
artr11,"",0,"","",c,artr117,t,1,1);




}

}


}
```

Program -3: Use Intorderby Ascending and descending order
and use Orderby ascending and descending order for the String datatype table.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char c=' ';
```

```
WDBASQL.Query("SelectOrderByASC","Orders","0" ,null,19,"123","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("SelectOrderByDESC","Orders","0" ,null,19,"123","",null,"",0,"","",c,null,t,1,1);


WDBASQL.Query("SelectIntOrderByAsc","nos","0" ,null,4,"123","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("SelectIntOrderByDesc","nos","0" ,null,4,"123","",null,"",0,"","",c,null,t,1,1);


}

}


}
```

Program 4:

==========

Write a WNOSQL Program to store the student query values in WDBA table from SQLSERVER for the given fields sno,tmark,rank and store it in encrypted form and again store the data in sqlserver for future use with C# program.

```
<WNOSQL>
<PACK>

<DATALIB> ps
```

```
<DATA>
public <CLASS> SQL3
{



public void main()



{

WDBA.writeln((manipulate.Signal("MANIPULATE","Select * from student","student","sno,tmark,rank","?,?,?",4,"oracle.jdbc.driver.OracleDriver","jdbc:oracle:thin:@localhost:1521:xe","system","jemin","wilmix2"))
}
}

</DATA>
```

Note : use manipulte .dll in this case..

Program5:

Write a program and use the following WNOSQL commands

and perform manipulation using

a) SELECT IN

b) SELECTLIKE

c) COUNT(*)

d) Encrypt

e) Decrypt

f) SelectAll

g) AVG(), MAX() , MIN() ,LOC(),SUM()

h) SelectCOLS, Count(),Distinct,MATCH

i) Insert

j) DatecompareAsc/DESC

k)InsertDesc ,AND ,Foreign Key

for WNOSQL TABLE

<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{

public void main()

{

```
String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char  c=' ';

ArrayList arhd111 =WDBASQL.Query("SelectIN","employess"
,"0",null,11,"0002","",null,"",0,"","",c,null,t,1,1);

ArrayList arhd112 =WDBASQL.Query("SelectNOTIN","employess"
,"0",null,11,"0002","",null,"",0,"","",c,null,t,1,1);



c='D';
WDBASQL.Query("SelectLike","Orders" ,"0",null,11,"","",null,"",0,"","",c,null,t,1,1);



WDBASQL.Query("Count(*)","Orders" ,"0",null,0,"","",null,"",0,"","",c,null,t,1,1);
WDBASQL.Query("MATH","nos" ,"0",null,0,"0","",null,"",0,"","acos",c,null,t,1,1);

WDBASQL.Query("Encrypt","nos" ,"0",null,0,"0","",null,"",0,"","",c,null,t,1,1);


WDBASQL.Query("Decrypt","nos" ,"0",null,0,"0","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("SelectAll","nos" ,"0",null,4,"0","",null,"",0,"","",c,null,t,1,1);
```

```
WDBASQL.Query("SelectAll","nos" ,"0",null,4,"4","",null,"",0,"","",c,null,t,1,1);



ArrayList  myList= new ArrayList();

myList.add("2005/01/12");

myList.add("2012/03/12");
myList.add("2006/03/12");
myList.add("2006/01/12");
myList.add("2005/11/12");

ArrayList  arms1d  = new  ArrayList();

arms1d.add(3);
arms1d.add(6);
arms1d.add(9);
arms1d.add(12);
arms1d.add(15);
arms1d.add(18);



ArrayList  sum55=WDBASQL.Query("AVG()","Orders" ,"0",null,6,"","",arms1d,"",0,"","",c,null,t,1,1);

ArrayList  sum55r=WDBASQL.Query("MAX()","Orders"
,"0",null,19,"","",arms1d,"",0,"","",c,null,t,1,1);
ArrayList  sum55gr=WDBASQL.Query("MIN()","Orders"
,"0",null,19,"","",arms1d,"",0,"","",c,null,t,1,1);

ArrayList arhd1g1 =WDBASQL.Query("LOC()","Orders"
,"0",null,19,"0002","",null,"",0,"","",c,null,t,1,1);
```

```
ArrayList  sum557=WDBASQL.Query("SUM()","Orders"
,"0",null,0,"","",arms1d,"",0,"","",c,null,t,1,1);



ArrayList  arts1= new  ArrayList();
arts1.add(3);
arts1.add(4);
arts1.add(5);
arts1.add(6);
arts1.add(7);
arts1.add(8);

ArrayList arh =WDBASQL.Query("SelectCols","Orders" ,"0",null,12-
3,"5","",arts1,"",0,"","",c,null,t,1,1);


ArrayList arhd =WDBASQL.Query("Count()","Orders" ,"0",null,13,"u","",null,"",0,"","",c,null,t,1,1);



ArrayList  art= new  ArrayList();



art.add(0);
art.add(1);
art.add(2);
```

```
art.add(3);
art.add(4);
art.add(5);
art.add(6);
art.add(7);
art.add(8);
art.add(9);
art.add(10);
art.add(11);
WDBASQL.Query("DISTINCT","abc1","0",null,11,"","", art,"",0,"11","",c,null,t,1,1);


ArrayList arhd1gy =WDBASQL.Query("MATCH","Orders"
,"0",null,19,"0001","",null,"",0,"1","",c,null,t,1,1);



ArrayList  ardds= new ArrayList();

for (int i=0;i<myList.size();i++)
ardds.add(i);

WDBASQL.Query("Insert","emp6" ,"",myList,0,"","", null,"",0," ","",c,null,t,1,1);

ArrayList  sum55grh=WDBASQL.Query("DateCompareDESC","emp6"
,"0",null,10,"","",ardds,"",0,"","",c,null,t,1,1);

ArrayList  sum55grhr=WDBASQL.Query("DateCompareASC","emp6"
,"0",null,10,"","",ardds,"",0,"","",c,null,t,1,1);


ArrayList st  = new  ArrayList();
```

```
st.add(1);
st.add("wilmix");



st.add("100");

st.add(2);
st.add("jem");

st.add("200");
st.add(4);
st.add("Peter");

st.add("200");

//st.add(3);
//st.add("Diana");

//st.add("100");
st.add(1);
st.add("");



st.add("500");



WDBASQL.Query("InsertDESC", "emp", "0", st, 0, "", "", null, "", 0, "", "", c, null, t, 0, 1);
ArrayList st111  = new  ArrayList();

st111.add(1);
st111.add("wilmix");
```

```
st111.add("100");

st111.add(2);
st111.add("jem");

st111.add("200");
st111.add(4);
st111.add("Peter");

st111.add("200");

//st111.add(3);
//st111.add("Diana");

//st111.add("100");
st111.add(1);
st111.add("");


st111.add("500");



ArrayList   tsf1p11= WDBASQL.Query("AND", "", "0", null, 11, "", "", sum55grh, "", 0, "", "",
c,sum55grhr, t, 1, 4);
```

```
ArrayList   tsf1p1 = WDBASQL.Query("ForeignKey", "Orders", "0", null, 17, "employess", "", null,
"", 0, "", "", c, null, t, 1, 1);

}

}


}
```

Program6:

Write a program and use the following WNOSQL commands

and perform manipulation using

a)DropTable ,InsertDesc,Insert,Insertinto,SelectRval

operations in WNOSQL Table.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB>  ps
<DATA>
public  <CLASS>  DATA
```

```
{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char  c=' ';

ArrayList st  = new  ArrayList();

st.add("indno");
st.add("name");

st.add("scoreno");
//WDBASQL.Query("DropTable","nos" ,"0",null,12-3,"5","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("InsertDESC", "emp", "0", st, 0, "", "", null, "", 0, "", "", c, null, t, 0, 1);

WDBASQL.Query("Insert", "emp", "0", st, 0, "", "", null, "", 0, "", "", c, null, t, 1, 4);

ArrayList st111  = new  ArrayList();

st111.add(1);
st111.add("wilmix");
```

```
st111.add("100");

st111.add(2);
st111.add("jem");

st111.add("200");
st111.add(4);
st111.add("Peter");

st111.add("200");

//st111.add(3);
//st111.add("Diana");

//st111.add("100");
st111.add(1);
st111.add("");


st111.add("500");


WDBASQL.Query("INSERTINTO","emp" ,"0",null,0,"0","",null,"",0,"","",c,st111,t,1,4);
ArrayList ts3j = WDBASQL.Query("SELECTRVAL", "emp", "0", null, 0, "0", "", null, "", 0, "", "", c,
null, t, 1, 4);



}
```

```
}

}
```

Program -7:

Write a program and use the following WNOSQL commands

and perform manipulation using

a)Insert,SelectAll,CLUSTER,BACKUPCLUSTER operations in WNOSQL Table.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{
```

```
String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char  c=' ';




ArrayList  cols = new  ArrayList();

for (int  i=0;i<=1990;i+=5)

cols.add(i);



ArrayList  cols1 = new  ArrayList();

for (int  i=0;i<=1990;i+=1)

cols1.add(i);



ArrayList  colsd = new  ArrayList();
WDBASQL.Query("Insert","emp6" ,"0",cols,1999,"","  ", null,"",0," ","",c,null,t,1,1);
```

```
WDBASQL.Query("CLUSTER","emp6" ,"0",null,1990,"","", cols1,"",0," ","",c,null,t,1,1);

ArrayList  colsdg =WDBASQL.Query("SelectAll","emp6" ,"0",null,1990,"","", null,"",0,"
","",c,null,t,1,1);

WDBASQL.Query("CLUSTERPROPERTY","emp6" ,"0",null,1990,"","", null,"",0," ","",c,null,t,1,1);

WDBASQL.Query("BACKUPCLUSTER","emp6" ,"0",null,1990,"","", null,"",0," ","",c,null,t,1,1);

WDBASQL.Query("SelectAll","emp6" ,"0",null,1990,"","", null,"",0," ","",c,null,t,1,1);




}

}




}
```

Program:8

Write a program and use the following WNOSQL commands

and perform manipulation using

a) Insertdesc , INSERTINTO,Insert

b) Selectdesc , SelectC*,Select R*,MATH

c) SELECTROWS,SELECTRVAL,SELECTINDEXES

operations in WNOSQL Table.

<WNOSQL>

<PACK>

<USE> CDollar.WDBA;

<DATALIB> ps

<DATA>

public <CLASS> DATA

{

public void main()

{

String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);

char c=' ';

```
ArrayList ar= new ArrayList();

for (int i=1;i<=99;i++)

ar.add(i);

ArrayList ar1= new ArrayList();
for (int i=0;i<=99;i+=3)

ar1.add(i);




ArrayList ar7= new ArrayList();
ar7.add("INO");
ar7.add("NOS");

ar7.add("NAME");
ar7.add("SALARY");



WDBASQL.Query("InsertDESC","nosd" ,"0",ar7,0,"","",null,"",0,"","",c,null,t,0,1);
WDBASQL.Query("Insert","nosd" ,"0",ar7,0,"","",null,"",0,"","",c,null,t,1,3);
WDBASQL.Query("INSERTINTO","nosd" ,"0",null,0,"0","",null,"",0,"","",c,ar1,t,1,3);

WDBASQL.Query("SelectDESC","nosd" ,"0",null,1,"0","",null,"",0,"","",c,null,t,0,1);
```

```
WDBASQL.Query("SELECTC*","nosd" ,"0",null,0,"0","",null,"",0,"","",c,null,t,0,1);



WDBASQL.Query("SELECTR*","nosd" ,"0",null,0,"0","",null,"",0,"","",c,null,t,1,3);




ArrayList  art= new  ArrayList();



art.add(0);
art.add(1);
art.add(2);

WDBASQL.Query("MATH","nosd" ,"0",null,0,"0","",null,"",0,"acos","",c,null,t,1,3);

WDBASQL.Query("SELECTROWS","nosd" ,"0",null,0,"0","",art,"",0,"","",c,null,t,1,3);

WDBASQL.Query("SELECTRVAL","nosd" ,"0",null,0,"0","",null,"",0,"","",c,null,t,1,3);

ArrayList  ar71= new  ArrayList();

ar71.add(4);
```

```
ArrayList arhg8ey =WDBASQL.Query("SELECTINDEXES","nosd"
,"0",null,0,"4","",ar71,"",0,"","",c,null,t,1,3);


}

}


}
```

Program 9:

=======

Write a program and use the following WNOSQL commands

and perform manipulation using

a)SelectAssign, Insertvalues, Primary key ,AND

b) SeLectupper,Selectlower

c) SYSDATE, MANIPULATE

d) ENCRYPT,DENCRYPT

operations in WNOSQL Table.

<WNOSQL>

<PACK>

```
<USE> CDollar.WDBA;
<DATALIB>  ps
<DATA>
public  <CLASS>  DATA

{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);char  c=' ';



ArrayList  art= new  ArrayList();




art.add(0);
art.add(1);
art.add(2);
```

```
ArrayList   tsf1=WDBASQL.Query("SelectAssign","columns"
,"1",null,1,"1","",null,"123,345",0,"","",c,null,t,1,1);



WDBASQL.Query("InsertValues","columns" ,"1",null,12-
1,"","",null,art.toString(),0,"","INSERT1",c,null,t,1,1);




ArrayList   tsf1p=new ArrayList();



ArrayList   tsf1p1= WDBASQL.Query("PrimaryKey", "abc", "0", null, 11, "abc1", "", null, "", 0, "", "",
c, null, t, 1, 1);

ArrayList   tsf1p11= WDBASQL.Query("AND", "", "0", null, 11, "", "", tsf1p, "", 0, "", "", c, tsf1p1, t,
1, 1);

ArrayList  art1= new  ArrayList();



for  (int  i=1;i<=6;i++)

art1.add(i);

ArrayList ass1=WDBASQL.Query("SelectUPPER","employess"
,"0",null,6,"","",art1,"",0,"","",c,null,t,1,1);
ArrayList ass11=WDBASQL.Query("SelectLOWER","employess"
,"0",null,6,"","",art1,"",0,"","",c,null,t,1,1);
ArrayList ass12=WDBASQL.Query("SYSDATE","" ,"0",null,6,"","",art1,"",0,"","",c,null,t,1,1);
```

```
ArrayList  art11= new  ArrayList();

art11.add(2016);

art11.add(10);
art11.add(15);

art11.add(5);
art11.add(-5);


WDBASQL.Query("ManipulateDate()","" ,"0",null,6,"","",art1,"",0,"yyyy MMM dd","",c,null,t,1,1);



WDBASQL.Query("Encrypt", "employess", "0", null, 12 - 3, "5", "", null, "", 0, "", "", c, null, t, 1, 1);


 WDBASQL.Query("Dencrypt", "employess", "0", null, 12 - 3, "5", "", null, "", 0, "", "", c, null, t, 1,
1);

}

}

}
```

Program 10:

=========

Write a program and use the following WNOSQL commands

and perform manipulation using

A)Search a DATA

B) SearchLS ,SearchGT

c) SelectRange Operations in WNOSQL TABLE.

<WNOSQL>

<PACK>

<USE> CDollar.WDBA;

<DATALIB> ps

<DATA>

public <CLASS> DATA

{

public void main()

{

String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

```
String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix", "Wilmixjemin12345",1,5,g);
char  c=' ';




WDBASQL.Query("Search","Orders" ,"0",null,15,"100","", null,"",0," ","",c,null,t,1,1);

WDBASQL.Query("SearchGT","emp6" ,"0",null,150,"100","", null,"",0," ","",c,null,t,1,1);
WDBASQL.Query("SearchLS","emp6" ,"0",null,150,"100","", null,"",0," ","",c,null,t,1,1);


WDBASQL.Query("SelectRange","Orders" ,"0",null,15,"","", null,"",0," ","",c,null,t,1,1);



}

}
```

==========

UNIT:8 WNOSQL (WSQL*) using CDollar,JAS,JDollar, etc. AND WNOSQL Program Exercises

==========

A) How to use WNOSQL db with CDollar,JDollar, and JAS?

Step-1: Convert WNOSQL PLSQL to WNOSQL .dll files.
to be used with CDollar, JDollar, and JAS
Since this programming accept .dll files.
or
Step-2:
You can add the WNOSQL.dll to JDollar CWE editor
Directly write WNOSQL Queries with JDollar CDollar ,JAS ,etc.
and by pressing button browse button at bottom of J$ or C$ CWE Editor
and after that press compile button in CWE Editor and
Run the Program using Run at top right.

==========

😎 WNOSQL PROGRAM EXERCISES

==========

Program-1:WNOSQL

<WNOSQL> //starting of wnosql or WDBA program
<PACK> // import all wdba packages
<USE> CDollar.WDBA; // load Cdollar.wdba packages
<USE> WDBA; //load wdba packages

```
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\Wcod");



// let database name me datastores , database pwd be dbpwds and the path be C:\\Pr
ograms\\WNOSQL\\WNOSQLProgramfiles\\Wcod

        String  t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);

//pass  dbuser and dbpwds as  wilmix78 ,wilmix78

String s1 = "CREATETABLE from Telecom 0 to 0 , 1 to 7 ?= 6639 By 6639 f(x) :
{SNO,CLASS,CHILDS}:
{1,A,a1,2,A,a2,3,A,a3,4,B,b1,5,B,b2,6,B,b3,7,C,c1,8,C,c2,9,C,c3} :{2,4}";

//create a  table  Telecom  with fields  SNO,CLASS,CHILDS... and set  rows =  1  and cols
=  7

//and intialize  the value {1,A,a1,2,A,a2,3,A,a3,4,B,b1,5,B,b2,6,B,b3,7,C,c1,8,C,c2,9,C,c3} =
> total  values =30

//here  1  indicates  SNO, A indicates  CLASS  , and  ,CHILD  indicates  a1  and  so -on .
```

```
ArrayList ar= WDBALIB.WDBAQUERY(s1, t); //  now  pass the query  in  the  WDBAQUERY  api

// so  what  happens  this  statement  will  execute  the  query  and  create  a  table

//and  it  will  insert  the  values  in  the  table Telecom.



}

}

}
```

Program-2:WNOSQL

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB>  ps
<DATA>
public  <CLASS>  DATA

{
```

```
public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNO");



        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);

String  s1 ="SELECTRVAL from Telecom 3 to 30 , 1 to 7 ?= C By 1 1 : {0} : {0} :{1}";
// At first  omit 3  fields and  start  from  3.
  //select  all row  values  from table Telecom  from   3  to   30
ArrayList ar= WDBALIB.WDBAQUERY(s1, t);

//now   when  you  execute   the  query  it  displays  all  the row  values..



}

}


}
```

Program-2 :WNOSQL

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB>  ps
<DATA>
public  <CLASS>  DATA

{



public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");




        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);

String  s11 ="SELECTRVAL from Telecom 3 to 33 , 1 to 7 ?= C By 1 1 : {0} : {0} :{1}";

String  s1 ="DELETE from Telecom 3 to "+WDBALIB.WDBAQUERY(s11, t).size() +" , 1 to 7 ?= A By
0 0 : {a1} : {xx}: {XX}";
ArrayList arf= WDBALIB.WDBAQUERY(s1, t);

// delete  the  value  from  rows  with  string  a1
```

String s16 ="SYSDATE from Telecom 3 to "+WDBALIB.WDBAQUERY(s11, t).size() +" , 1 to 7 ?= A By 0 0 : {a1} : {xx}: {XX}";

ArrayList arfh= WDBALIB.WDBAQUERY(s16, t);

//compute sysdate for the Telecom

}

}

}

Program3 : WNOSQL

<WNOSQL>

<PACK>

<USE> CDollar.WDBA;

<USE> WDBA;

<DATALIB> ps

<DATA>

public <CLASS> DATA

{

```
public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");

        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);

    String  s11 ="SELECTRVAL from Telecom 3 to 33 , 1 to 7 ?= A By 1 1 : {0} : {Telecom}
:{0}";

ArrayList ar1 = WDBALIB.WDBAQUERY(s11, t);

//compute  the  size  of  Telecom  table  inorder  to  insert  the  values (1,A,a1) after  that

//so  we  will   choose   the  second   section   for  values  insertion.

String  s1 ="INSERTINTO from Telecom 3 to "+ar1.size() +" , 1 to 7 ?= A By 1 1 : {0} : {1,A,a1} :
{0}";
```

```
ArrayList ar= WDBALIB.WDBAQUERY(s1, t);

//execute  the  query

}

}


}
```

Program4: WNOSQL

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB>  ps
<DATA>
public  <CLASS>  DATA

{



public void main()

{
```

```
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");



        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);



String  s1 ="MATCH from Telecom 3 to 29 , 1 to 7 ?= C 0001 1 1 : {0} : {0} :{0}";

// we are choosing values  starting  from 3 to  29

// and  perform match operations  and test  what rows are matched by char C


 WDBA.writeln(""+WDBALIB.WDBAQUERY( s1, t));



}

}


}
```

Telecom table contents :

[SNO CLASS CHILDS 1 A XXX5 2 A a2 3 A a3 4 B b1 5 B b2 6 B b3 7 C c1 8 C c2 9 C c3]

MATCHED ROWS

[22, 23, 25, 26, 28, 29]

for eg)

22 indicates 7 after that C character succeed and ends

so 22 23 will be taken into account.

simillarly so-on.

Program-5:WNOSQL

<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB> ps
<DATA>
public <CLASS> DATA

{

```
public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");



        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);



String  s1 ="SelectOrderByASC from Telecom 3 to 29 , 1 to 7 ?= 123 By 1 1 : {0} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s1,  t));

// we know by definition  this  statement order  the table contents in Ascending order.


String  s11 ="SelectOrderByDESC from Telecom 3 to 29 , 1 to 7 ?= 123 By 1 1 : {0} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s11,  t));

// we know by definition  this  statement order  the table
contents in  Descending order.

String  s118 ="SelectRange from Telecom 3 to 13 , 1 to 7 ?= C By 1 1 : {0} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s118,  t));

// list  a range of values  from rows 3 to 13
```

String s12 ="SelectIntOrderByAsc from datastoreh 50 to 2000 , 1 to 1 ?= 123 By 1 1 : {0} : {0} :{0}";

WDBA.writeln(""+WDBALIB.WDBAQUERY(s12, t));

// we know by definition this statement order the table contents containing integer nos in Ascending order.

String s121 ="SelectIntOrderByDesc from datastoreh 50 to 2000 , 1 to 1 ?= 123 By 1 1 : {0} : {0} :{0}";

WDBA.writeln(""+WDBALIB.WDBAQUERY(s121, t));

// we know by definition this statement order the table contents containing integer nos in Descending order.

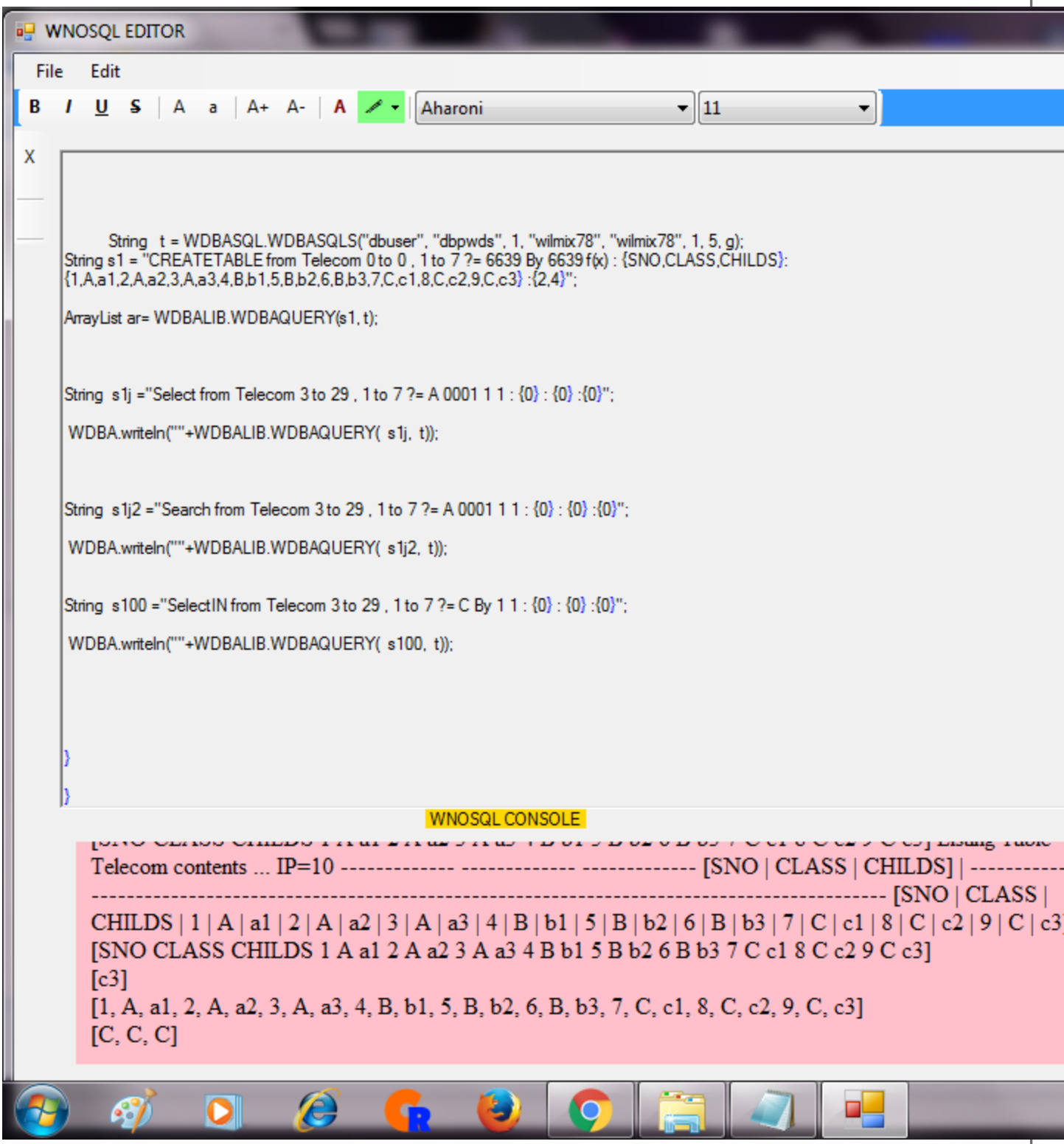
String s1217 ="SelectAll from datastoreh 50 to 2000 , 1 to 1 ?= 123 By 1 1 : {0} : {0} :{0}";

WDBA.writeln("datas"+WDBALIB.WDBAQUERY(s1217, t));

// we know by
definition this statement will list all the table values from 50 to 2000

```
String  s121377 ="SearchGT from datastoreh 50 to 200 , 1 to 1 ?= 100 By 1 1 : {0} : {0} :{0}";

 WDBA.writeln("datas51"+WDBALIB.WDBAQUERY( s121377,  t));

 //  we know by definition this statement will list  all the  table values from 50  to 100
//which is greater than the value  100 from  table values..




}

}


}
```

Program-6:WNOSQL



Look at the figure of wnosql using CWE editor

a) if you use Select statement from 3 to 29
; here last given range value 29 is taken into account

and it displays c3 as value.

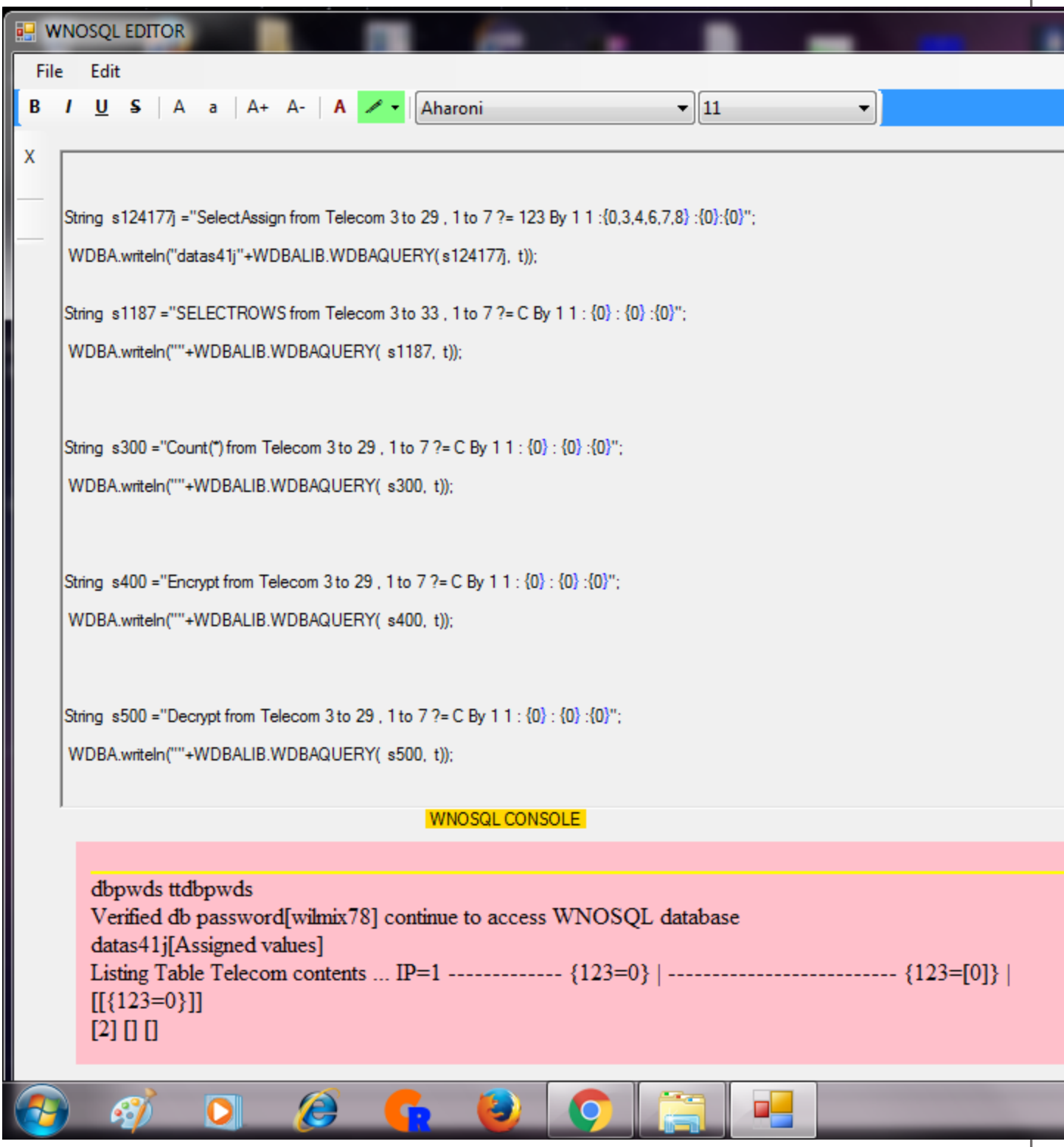
b) If you use Search statement from
3 to 29 ; this statement will list the values that match the character A

and displays the result.

c) SelectIN statement will check the member value C from telecom

and displays C as 3 times since C is found 3 times.

Program7:WNOSQL



SelectAssign statement will Assign 123 value to Telecom table

Count(*) will displays no of rows in Telecom table so ans is [2].

Program8:WNOSQL

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");



        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);
```

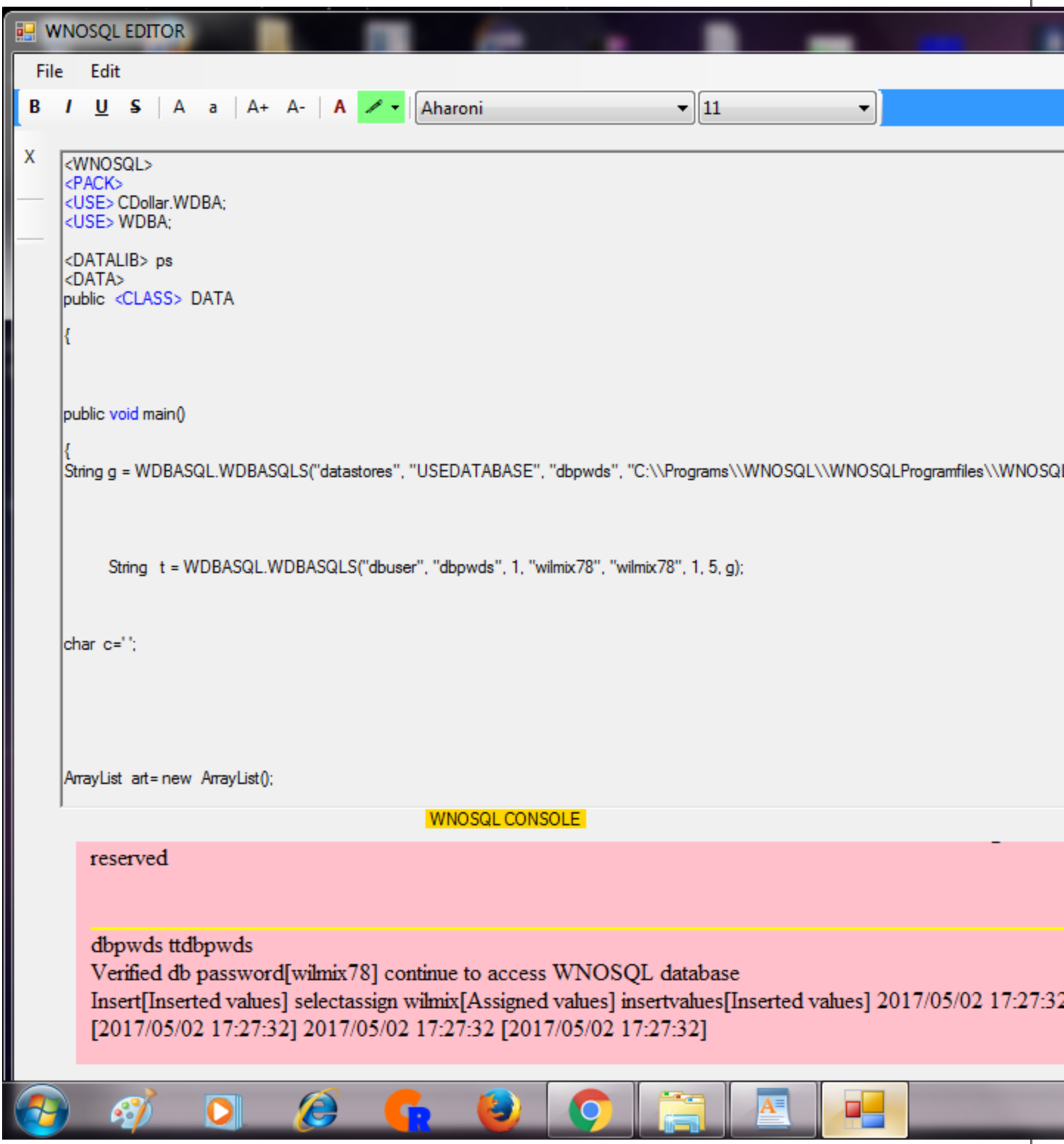
```
char  c=' ';




ArrayList  art= new  ArrayList();



art.add(0);
art.add(1);
art.add(2);



String  s2 ="Insert from columns 0 to 9 , 1 to 1 ?= X By 1 1 : {3,4,5,6,7,8} : {0} :{0}";

 WDBA.writeln("Insert"+WDBALIB.WDBAQUERY(  s2,  t));



String  s23 ="SelectAssign from columns 0 to 9 , 1 to 1 ?= X By 123,345 1 : {3,4,5,6,7,8} : {0}
:{0}";

 WDBA.writeln("selectassign wilmix"+WDBALIB.WDBAQUERY(  s23,  t));
```

```
String  s231 ="InsertValues from columns 1 to 11 , 1 to 1 ?= X By 123,345 1 : {3,4,5,6,7,8} : {0}
:{0}";

 WDBA.writeln("insertvalues"+WDBALIB.WDBAQUERY(  s231,  t));


String  s23211 ="SYSDATE from Telecom 0 to 6 , 1 to 7 ?= X By 1 1 : {0} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s23211,  t));




ArrayList  art11= new  ArrayList();

art11.add(2016);

art11.add(10);
art11.add(15);

art11.add(5);
art11.add(-5);



String  s232111 ="SYSDATE from employess 0 to 6 , 1 to 1 ?= yyyy MMM dd By 1 1 :
{3,4,5,6,7,8} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s232111,  t));
```

```
}

}

}
```

Output

```
====
```



Program9:WNOSQL

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB>  ps
<DATA>
public  <CLASS>  DATA

{



public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:||Programs||WNOSQL||WNOSQLProgramfiles||WNOSQL");



        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);



char  c=' ';
```

```
String  s23l02 ="SELECTR* from datastorehg 0 to 0 , 1 to 1 ?= X By 1 1 : {0} : {0} : {0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s23l02,  t));









}

}


}
```

OUTPUT

=====

dbpwds ttdbpwds

Verified db password[wilmix78] continue to access WNOSQL database

[EMPLOYEENAME SALARY EMPLOYEENAME SALARY 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102 3103

3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119
3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130 3131 3132 3133 3134 3135
3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146 3147 3148 3149 3150 3151
3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162 3163 3164 3165 3166 3167
3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182 3183
3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194 3195 3196 3197 3198 3199
3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215
3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226 3227 3228 3229 3230 3231
3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247
3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263
3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274 3275 3276 3277 3278 3279
3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 3294 3295
3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311
3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327
3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338 3339 3340 3341 3342 3343
3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359
3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370 3371 3372 3373 3374 3375
3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386 3387 3388 3389 3390 3391
3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402 3403 3404 3405 3406 3407
3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422 3423
3424 3425 3426 3427 3428 3429 3430 3431 3432 3433 3434 3435 3436 3437 3438 3439
3440 3441 3442 3443 3444 3445 3446 3447 3448 3449 3450 3451 3452 3453 3454 3455
3456 3457 3458 3459 3460 3461 3462 3463 3464 3465 3466 3467 3468 3469 3470 3471
3472 3473 3474 3475 3476 3477 3478 3479 3480 3481 3482 3483 3484 3485 3486 3487
3488 3489 3490 3491 3492 3493 3494 3495 3496 3497 3498 3499 3500 3501 3502 3503
3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514 3515 3516 3517 3518 3519
3520 3521 3522 3523 3524 3525 3526 3527 3528 3529 3530 3531 3532 3533 3534 3535
3536 3537 3538 3539 3540 3541 3542 3543 3544 3545 3546 3547 3548 3549 3550 3551
3552 3553 3554 3555 3556 3557 3558 3559 3560 3561 3562 3563 3564 3565 3566 3567
3568 3569 3570 3571 3572 3573 3574 3575 3576 3577 3578 3579 3580 3581 3582 3583
3584 3585 3586 3587 3588 3589 3590 3591 3592 3593 3594 3595 3596 3597 3598 3599
3600 3601 3602 3603 3604 3605 3606 3607 3608 3609 3610 3611 3612 3613 3614 3615
3616 3617 3618 3619 3620 3621 3622 3623 3624 3625 3626 3627 3628 3629 3630 3631
3632 3633 3634 3635 3636 3637 3638 3639 3640 3641 3642 3643 3644 3645 3646 3647
3648 3649 3650 3651 3652 3653 3654 3655 3656 3657 3658 3659 3660 3661 3662 3663
3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674 3675 3676 3677 3678 3679
3680 3681 3682 3683 3684 3685 3686 3687 3688 3689 3690 3691 3692 3693 3694 3695
3696 3697 3698 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 3709 3710 3711
3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725 3726 3727
3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 3738 3739 3740 3741 3742 3743
3744 3745 3746 3747 3748 3749 3750 3751 3752 3753 3754 3755 3756 3757 3758 3759
3760 3761 3762 3763 3764 3765 3766 3767 3768 3769 3770 3771 3772 3773 3774 3775

3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 3787 3788 3789 3790 3791
3792 3793 3794 3795 3796 3797 3798 3799 3800 3801 3802 3803 3804 3805 3806 3807
3808 3809 3810 3811 3812 3813 3814 3815 3816 3817 3818 3819 3820 3821 3822 3823
3824 3825 3826 3827 3828 3829 3830 3831 3832 3833 3834 3835 3836 3837 3838 3839
3840 3841 3842 3843 3844 3845 3846 3847 3848 3849 3850 3851 3852 3853 3854 3855
3856 3857 3858 3859 3860 3861 3862 3863 3864 3865 3866 3867 3868 3869 3870 3871
3872 3873 3874 3875 3876 3877 3878 3879 3880 3881 3882 3883 3884 3885 3886 3887
3888 3889 3890 3891 3892 3893 3894 3895 3896 3897 3898 3899 3900 3901 3902 3903
3904 3905 3906 3907 3908 3909 3910 3911 3912 3913 3914 3915 3916 3917 3918 3919
3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932 3933 3934 3935
3936 3937 3938 3939 3940 3941 3942 3943 3944 3945 3946 3947 3948 3949 3950 3951
3952 3953 3954 3955 3956 3957 3958 3959 3960 3961 3962 3963 3964 3965 3966 3967
3968 3969 3970 3971 3972 3973 3974 3975 3976 3977 3978 3979 3980 3981 3982 3983
3984 3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999
4000 4001 4002 4003 4004 4005 4006 4007 4008 4009 4010 4011 4012 4013 4014 4015
4016 4017 4018 4019 4020 4021 4022 4023 4024 4025 4026 4027 4028 4029 4030 4031
4032 4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 4043 4044 4045 4046 4047
4048 4049 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 4060 4061 4062 4063
4064 4065 4066 4067 4068 4069 4070 4071 4072 4073 4074 4075 4076 4077 4078 4079
4080 4081 4082 4083 4084 4085 4086 4087 4088 4089 4090 4091 4092 4093 4094 4095
4096 4097 4098 4099 4100 4101 4102 4103 4104 4105 4106 4107 4108 4109 4110 4111
4112 4113 4114 4115 4116 4117 4118 4119 4120 4121 4122 4123 4124 4125 4126 4127
4128 4129 4130 4131 4132 4133 4134 4135 4136 4137 4138 4139 4140 4141 4142 4143
4144 4145 4146 4147 4148 4149 4150 4151 4152 4153 4154 4155 4156 4157 4158 4159
4160 4161 4162 4163 4164 4165 4166 4167 4168 4169 4170 4171 4172 4173 4174 4175
4176 4177 4178 4179 4180 4181 4182 4183 4184 4185 4186 4187 4188 4189 4190 4191
4192 4193 4194 4195 4196 4197 4198 4199 4200 4201 4202 4203 4204 4205 4206 4207
4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223
4224 4225 4226 4227 4228 4229 4230 4231 4232 4233 4234 4235 4236 4237 4238 4239
4240 4241 4242 4243 4244 4245 4246 4247 4248 4249 4250 4251 4252 4253 4254 4255
4256 4257 4258 4259 4260 4261 4262 4263 4264 4265 4266 4267 4268 4269 4270 4271
4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282 4283 4284 4285 4286 4287
4288 4289 4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303
4304 4305 4306 4307 4308 4309 4310 4311 4312 4313 4314 4315 4316 4317 4318 4319
4320 4321 4322 4323 4324 4325 4326 4327 4328 4329 4330 4331 4332 4333 4334 4335
4336 4337 4338 4339 4340 4341 4342 4343 4344 4345 4346 4347 4348 4349 4350 4351
4352 4353 4354 4355 4356 4357 4358 4359 4360 4361 4362 4363 4364 4365 4366 4367
4368 4369 4370 4371 4372 4373 4374 4375 4376 4377 4378 4379 4380 4381 4382 4383
4384 4385 4386 4387 4388 4389 4390 4391 4392 4393 4394 4395 4396 4397 4398 4399
4400 4401 4402 4403 4404 4405 4406 4407 4408 4409 4410 4411 4412 4413 4414 4415
4416 4417 4418 4419 4420 4421 4422 4423 4424 4425 4426 4427 4428 4429 4430 4431
4432 4433 4434 4435 4436 4437 4438 4439 4440 4441 4442 4443 4444 4445 4446 4447

4448 4449 4450 4451 4452 4453 4454 4455 4456 4457 4458 4459 4460 4461 4462 4463
4464 4465 4466 4467 4468 4469 4470 4471 4472 4473 4474 4475 4476 4477 4478 4479
4480 4481 4482 4483 4484 4485 4486 4487 4488 4489 4490 4491 4492 4493 4494 4495
4496 4497 4498 4499 4500 4501 4502 4503 4504 4505 4506 4507 4508 4509 4510 4511
4512 4513 4514 4515 4516 4517 4518 4519 4520 4521 4522 4523 4524 4525 4526 4527
4528 4529 4530 4531 4532 4533 4534 4535 4536 4537 4538 4539 4540 4541 4542 4543
4544 4545 4546 4547 4548 4549 4550 4551 4552 4553 4554 4555 4556 4557 4558 4559
4560 4561 4562 4563 4564 4565 4566 4567 4568 4569 4570 4571 4572 4573 4574 4575
4576 4577 4578 4579 4580 4581 4582 4583 4584 4585 4586 4587 4588 4589 4590 4591
4592 4593 4594 4595 4596 4597 4598 4599 4600 4601 4602 4603 4604 4605 4606 4607
4608 4609 4610 4611 4612 4613 4614 4615 4616 4617 4618 4619 4620 4621 4622 4623
4624 4625 4626 4627 4628 4629 4630 4631 4632 4633 4634 4635 4636 4637 4638 4639
4640 4641 4642 4643 4644 4645 4646 4647 4648 4649 4650 4651 4652 4653 4654 4655
4656 4657 4658 4659 4660 4661 4662 4663 4664 4665 4666 4667 4668 4669 4670 4671
4672 4673 4674 4675 4676 4677 4678 4679 4680 4681 4682 4683 4684 4685 4686 4687
4688 4689 4690 4691 4692 4693 4694 4695 4696 4697 4698 4699 4700 4701 4702 4703
4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715 4716 4717 4718 4719
4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731 4732 4733 4734 4735
4736 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747 4748 4749 4750 4751
4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763 4764 4765 4766 4767
4768 4769 4770 4771 4772 4773 4774 4775 4776 4777 4778 4779 4780 4781 4782 4783
4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795 4796 4797 4798 4799
4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810 4811 4812 4813 4814 4815
4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831
4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 4847
4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858 4859 4860 4861 4862 4863
4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879
4880 4881 4882 4883 4884 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895
4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911
4912 4913 4914 4915 4916 4917 4918 4919 4920 4921 4922 4923 4924 4925 4926 4927
4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943
4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 4958 4959
4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975
4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991
4992 4993 4994 4995 4996 4997 4998 4999 5000 5001 5002 5003 5004 5005 5006 5007
5008 5009 5010 5011 5012 5013 5014 5015 5016 5017 5018 5019 5020 5021 5022 5023
5024 5025 5026 5027 5028 5029 5030 5031 5032 5033 5034 5035 5036 5037 5038 5039
5040 5041 5042 5043 5044 5045 5046 5047 5048 5049 5050 5051 5052 5053 5054 5055
5056 5057 5058 5059 5060 5061 5062 5063 5064 5065 5066 5067 5068 5069 5070 5071
5072 5073 5074 5075 5076 5077 5078 5079 5080 5081 5082 5083 5084 5085 5086 5087
5088 5089 5090 5091 5092 5093 5094 5095 5096 5097 5098 5099 5100 5101 5102 5103
5104 5105 5106 5107 5108 5109 5110 5111 5112 5113 5114 5115 5116 5117 5118 5119

5120 5121 5122 5123 5124 5125 5126 5127 5128 5129 5130 5131 5132 5133 5134 5135
5136 5137 5138 5139 5140 5141 5142 5143 5144 5145 5146 5147 5148 5149 5150 5151
5152 5153 5154 5155 5156 5157 5158 5159 5160 5161 5162 5163 5164 5165 5166 5167
5168 5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183
5184 5185 5186 5187 5188 5189 5190 5191 5192 5193 5194 5195 5196 5197 5198 5199
5200 5201 5202 5203 5204 5205 5206 5207 5208 5209 5210 5211 5212 5213 5214 5215
5216 5217 5218 5219 5220 5221 5222 5223 5224 5225 5226 5227 5228 5229 5230 5231
5232 5233 5234 5235 5236 5237 5238 5239 5240 5241 5242 5243 5244 5245 5246 5247
5248 5249 5250 5251 5252 5253 5254 5255 5256 5257 5258 5259 5260 5261 5262 5263
5264 5265 5266 5267 5268 5269 5270 5271 5272 5273 5274 5275 5276 5277 5278 5279
5280 5281 5282 5283 5284 5285 5286 5287 5288 5289 5290 5291 5292 5293 5294 5295
5296 5297 5298 5299 5300 5301 5302 5303 5304 5305 5306 5307 5308 5309 5310 5311
5312 5313 5314 5315 5316 5317 5318 5319 5320 5321 5322 5323 5324 5325 5326 5327
5328 5329 5330 5331 5332 5333 5334 5335 5336 5337 5338 5339 5340 5341 5342 5343
5344 5345 5346 5347 5348 5349 5350 5351 5352 5353 5354 5355 5356 5357 5358 5359
5360 5361 5362 5363 5364 5365 5366 5367 5368 5369 5370 5371 5372 5373 5374 5375
5376 5377 5378 5379 5380 5381 5382 5383 5384 5385 5386 5387 5388 5389 5390 5391
5392 5393 5394 5395 5396 5397 5398 5399 5400 5401 5402 5403 5404 5405 5406 5407
5408 5409 5410 5411 5412 5413 5414 5415 5416 5417 5418 5419 5420 5421 5422 5423
5424 5425 5426 5427 5428 5429 5430 5431 5432 5433 5434 5435 5436 5437 5438 5439
5440 5441 5442 5443 5444 5445 5446 5447 5448 5449 5450 5451 5452 5453 5454 5455
5456 5457 5458 5459 5460 5461 5462 5463 5464 5465 5466 5467 5468 5469 5470 5471
5472 5473 5474 5475 5476 5477 5478 5479 5480 5481 5482 5483 5484 5485 5486 5487
5488 5489 5490 5491 5492 5493 5494 5495 5496 5497 5498 5499 5500 5501 5502 5503
5504 5505 5506 5507 5508 5509 5510 5511 5512 5513 5514 5515 5516 5517 5518 5519
5520 5521 5522 5523 5524 5525 5526 5527 5528 5529 5530 5531 5532 5533 5534 5535
5536 5537 5538 5539 5540 5541 5542 5543 5544 5545 5546 5547 5548 5549 5550 5551
5552 5553 5554 5555 5556 5557 5558 5559 5560 5561 5562 5563 5564 5565 5566 5567
5568 5569 5570 5571 5572 5573 5574 5575 5576 5577 5578 5579 5580 5581 5582 5583
5584 5585 5586 5587 5588 5589 5590 5591 5592 5593 5594 5595 5596 5597 5598 5599
5600 5601 5602 5603 5604 5605 5606 5607 5608 5609 5610 5611 5612 5613 5614 5615
5616 5617 5618 5619 5620 5621 5622 5623 5624 5625 5626 5627 5628 5629 5630 5631
5632 5633 5634 5635 5636 5637 5638 5639 5640 5641 5642 5643 5644 5645 5646 5647
5648 5649 5650 5651 5652 5653 5654 5655 5656 5657 5658 5659 5660 5661 5662 5663
5664 5665 5666 5667 5668 5669 5670 5671 5672 5673 5674 5675 5676 5677 5678 5679
5680 5681 5682 5683 5684 5685 5686 5687 5688 5689 5690 5691 5692 5693 5694 5695
5696 5697 5698 5699 5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710 5711
5712 5713 5714 5715 5716 5717 5718 5719 5720 5721 5722 5723 5724 5725 5726 5727
5728 5729 5730 5731 5732 5733 5734 5735 5736 5737 5738 5739 5740 5741 5742 5743
5744 5745 5746 5747 5748 5749 5750 5751 5752 5753 5754 5755 5756 5757 5758 5759
5760 5761 5762 5763 5764 5765 5766 5767 5768 5769 5770 5771 5772 5773 5774 5775
5776 5777 5778 5779 5780 5781 5782 5783 5784 5785 5786 5787 5788 5789 5790 5791

5792 5793 5794 5795 5796 5797 5798 5799 5800 5801 5802 5803 5804 5805 5806 5807
5808 5809 5810 5811 5812 5813 5814 5815 5816 5817 5818 5819 5820 5821 5822 5823
5824 5825 5826 5827 5828 5829 5830 5831 5832 5833 5834 5835 5836 5837 5838 5839
5840 5841 5842 5843 5844 5845 5846 5847 5848 5849 5850 5851 5852 5853 5854 5855
5856 5857 5858 5859 5860 5861 5862 5863 5864 5865 5866 5867 5868 5869 5870 5871
5872 5873 5874 5875 5876 5877 5878 5879 5880 5881 5882 5883 5884 5885 5886 5887
5888 5889 5890 5891 5892 5893 5894 5895 5896 5897 5898 5899 5900 5901 5902 5903
5904 5905 5906 5907 5908 5909 5910 5911 5912 5913 5914 5915 5916 5917 5918 5919
5920 5921 5922 5923 5924 5925 5926 5927 5928 5929 5930 5931 5932 5933 5934 5935
5936 5937 5938 5939 5940 5941 5942 5943 5944 5945 5946 5947 5948 5949 5950 5951
5952 5953 5954 5955 5956 5957 5958 5959 5960 5961 5962 5963 5964 5965 5966 5967
5968 5969 5970 5971 5972 5973 5974 5975 5976 5977 5978 5979 5980 5981 5982 5983
5984 5985 5986 5987 5988 5989 5990 5991 5992 5993 5994 5995 5996 5997 5998 5999
6000] Listing Table datastorehg contents ... IP=1502 ---------- ----------
[EMPLOYEENAME | SALARY] | ----------
[EMPLOYEENAME | SALARY | EMPLOYEENAME | SALARY | 3001 | 3002 | 3003 | 3004 | 3005 |
3006 | 3007 | 3008 | 3009 | 3010 | 3011 | 3012 | 3013 | 3014 | 3015 | 3016 | 3017 | 3018 |
3019 | 3020 | 3021 | 3022 | 3023 | 3024 | 3025 | 3026 | 3027 | 3028 | 3029 | 3030 | 3031 |
3032 | 3033 | 3034 | 3035 | 3036 | 3037 | 3038 | 3039 | 3040 | 3041 | 3042 | 3043 | 3044 |
3045 | 3046 | 3047 | 3048 | 3049 | 3050 | 3051 | 3052 | 3053 | 3054 | 3055 | 3056 | 3057 |
3058 | 3059 | 3060 | 3061 | 3062 | 3063 | 3064 | 3065 | 3066 | 3067 | 3068 | 3069 | 3070 |
3071 | 3072 | 3073 | 3074 | 3075 | 3076 | 3077 | 3078 | 3079 | 3080 | 3081 | 3082 | 3083 |
3084 | 3085 | 3086 | 3087 | 3088 | 3089 | 3090 | 3091 | 3092 | 3093 | 3094 | 3095 | 3096 |
3097 | 3098 | 3099 | 3100 | 3101 | 3102 | 3103 | 3104 | 3105 | 3106 | 3107 | 3108 | 3109 |
3110 | 3111 | 3112 | 3113 | 3114 | 3115 | 3116 | 3117 | 3118 | 3119 | 3120 | 3121 | 3122 |
3123 | 3124 | 3125 | 3126 | 3127 | 3128 | 3129 | 3130 | 3131 | 3132 | 3133 | 3134 | 3135 |
3136 | 3137 | 3138 | 3139 | 3140 | 3141 | 3142 | 3143 | 3144 | 3145 | 3146 | 3147 | 3148 |
3149 | 3150 | 3151 | 3152 | 3153 | 3154 | 3155 | 3156 | 3157 | 3158 | 3159 | 3160 | 3161 |
3162 | 3163 | 3164 | 3165 | 3166 | 3167 | 3168 | 3169 | 3170 | 3171 | 3172 | 3173 | 3174 |
3175 | 3176 | 3177 | 3178 | 3179 | 3180 | 3181 | 3182 | 3183 | 3184 | 3185 | 3186 | 3187 |
3188 | 3189 | 3190 | 3191 | 3192 | 3193 | 3194 | 3195 | 3196 | 3197 | 3198 | 3199 | 3200 |
3201 | 3202 | 3203 | 3204 | 3205 | 3206 | 3207 | 3208 | 3209 | 3210 | 3211 | 3212 | 3213 |
3214 | 3215 | 3216 | 3217 | 3218 | 3219 | 3220 | 3221 | 3222 | 3223 | 3224 | 3225 | 3226 |
3227 | 3228 | 3229 | 3230 | 3231 | 3232 | 3233 | 3234 | 3235 | 3236 | 3237 | 3238 | 3239 |
3240 | 3241 | 3242 | 3243 | 3244 | 3245 | 3246 | 3247 | 3248 | 3249 | 3250 | 3251 | 3252 |
3253 | 3254 | 3255 | 3256 | 3257 | 3258 | 3259 | 3260 | 3261 | 3262 | 3263 | 3264 | 3265 |
3266 | 3267 | 3268 | 3269 | 3270 | 3271 | 3272 | 3273 | 3274 | 3275 | 3276 | 3277 | 3278 |
3279 | 3280 | 3281 | 3282 | 3283 | 3284 | 3285 | 3286 | 3287 | 3288 | 3289 | 3290 | 3291 |
3292 | 3293 | 3294 | 3295 | 3296 | 3297 | 3298 | 3299 | 3300 | 3301 | 3302 | 3303 | 3304 |
3305 | 3306 | 3307 | 3308 | 3309 | 3310 | 3311 | 3312 | 3313 | 3314 | 3315 | 3316 | 3317 |
3318 | 3319 | 3320 | 3321 | 3322 | 3323 | 3324 | 3325 | 3326 | 3327 | 3328 | 3329 | 3330 |
3331 | 3332 | 3333 | 3334 | 3335 | 3336 | 3337 | 3338 | 3339 | 3340 | 3341 | 3342 | 3343 |

3344 | 3345 | 3346 | 3347 | 3348 | 3349 | 3350 | 3351 | 3352 | 3353 | 3354 | 3355 | 3356 |
3357 | 3358 | 3359 | 3360 | 3361 | 3362 | 3363 | 3364 | 3365 | 3366 | 3367 | 3368 | 3369 |
3370 | 3371 | 3372 | 3373 | 3374 | 3375 | 3376 | 3377 | 3378 | 3379 | 3380 | 3381 | 3382 |
3383 | 3384 | 3385 | 3386 | 3387 | 3388 | 3389 | 3390 | 3391 | 3392 | 3393 | 3394 | 3395 |
3396 | 3397 | 3398 | 3399 | 3400 | 3401 | 3402 | 3403 | 3404 | 3405 | 3406 | 3407 | 3408 |
3409 | 3410 | 3411 | 3412 | 3413 | 3414 | 3415 | 3416 | 3417 | 3418 | 3419 | 3420 | 3421 |
3422 | 3423 | 3424 | 3425 | 3426 | 3427 | 3428 | 3429 | 3430 | 3431 | 3432 | 3433 | 3434 |
3435 | 3436 | 3437 | 3438 | 3439 | 3440 | 3441 | 3442 | 3443 | 3444 | 3445 | 3446 | 3447 |
3448 | 3449 | 3450 | 3451 | 3452 | 3453 | 3454 | 3455 | 3456 | 3457 | 3458 | 3459 | 3460 |
3461 | 3462 | 3463 | 3464 | 3465 | 3466 | 3467 | 3468 | 3469 | 3470 | 3471 | 3472 | 3473 |
3474 | 3475 | 3476 | 3477 | 3478 | 3479 | 3480 | 3481 | 3482 | 3483 | 3484 | 3485 | 3486 |
3487 | 3488 | 3489 | 3490 | 3491 | 3492 | 3493 | 3494 | 3495 | 3496 | 3497 | 3498 | 3499 |
3500 | 3501 | 3502 | 3503 | 3504 | 3505 | 3506 | 3507 | 3508 | 3509 | 3510 | 3511 | 3512 |
3513 | 3514 | 3515 | 3516 | 3517 | 3518 | 3519 | 3520 | 3521 | 3522 | 3523 | 3524 | 3525 |
3526 | 3527 | 3528 | 3529 | 3530 | 3531 | 3532 | 3533 | 3534 | 3535 | 3536 | 3537 | 3538 |
3539 | 3540 | 3541 | 3542 | 3543 | 3544 | 3545 | 3546 | 3547 | 3548 | 3549 | 3550 | 3551 |
3552 | 3553 | 3554 | 3555 | 3556 | 3557 | 3558 | 3559 | 3560 | 3561 | 3562 | 3563 | 3564 |
3565 | 3566 | 3567 | 3568 | 3569 | 3570 | 3571 | 3572 | 3573 | 3574 | 3575 | 3576 | 3577 |
3578 | 3579 | 3580 | 3581 | 3582 | 3583 | 3584 | 3585 | 3586 | 3587 | 3588 | 3589 | 3590 |
3591 | 3592 | 3593 | 3594 | 3595 | 3596 | 3597 | 3598 | 3599 | 3600 | 3601 | 3602 | 3603 |
3604 | 3605 | 3606 | 3607 | 3608 | 3609 | 3610 | 3611 | 3612 | 3613 | 3614 | 3615 | 3616 |
3617 | 3618 | 3619 | 3620 | 3621 | 3622 | 3623 | 3624 | 3625 | 3626 | 3627 | 3628 | 3629 |
3630 | 3631 | 3632 | 3633 | 3634 | 3635 | 3636 | 3637 | 3638 | 3639 | 3640 | 3641 | 3642 |
3643 | 3644 | 3645 | 3646 | 3647 | 3648 | 3649 | 3650 | 3651 | 3652 | 3653 | 3654 | 3655 |
3656 | 3657 | 3658 | 3659 | 3660 | 3661 | 3662 | 3663 | 3664 | 3665 | 3666 | 3667 | 3668 |
3669 | 3670 | 3671 | 3672 | 3673 | 3674 | 3675 | 3676 | 3677 | 3678 | 3679 | 3680 | 3681 |
3682 | 3683 | 3684 | 3685 | 3686 | 3687 | 3688 | 3689 | 3690 | 3691 | 3692 | 3693 | 3694 |
3695 | 3696 | 3697 | 3698 | 3699 | 3700 | 3701 | 3702 | 3703 | 3704 | 3705 | 3706 | 3707 |
3708 | 3709 | 3710 | 3711 | 3712 | 3713 | 3714 | 3715 | 3716 | 3717 | 3718 | 3719 | 3720 |
3721 | 3722 | 3723 | 3724 | 3725 | 3726 | 3727 | 3728 | 3729 | 3730 | 3731 | 3732 | 3733 |
3734 | 3735 | 3736 | 3737 | 3738 | 3739 | 3740 | 3741 | 3742 | 3743 | 3744 | 3745 | 3746 |
3747 | 3748 | 3749 | 3750 | 3751 | 3752 | 3753 | 3754 | 3755 | 3756 | 3757 | 3758 | 3759 |
3760 | 3761 | 3762 | 3763 | 3764 | 3765 | 3766 | 3767 | 3768 | 3769 | 3770 | 3771 | 3772 |
3773 | 3774 | 3775 | 3776 | 3777 | 3778 | 3779 | 3780 | 3781 | 3782 | 3783 | 3784 | 3785 |
3786 | 3787 | 3788 | 3789 | 3790 | 3791 | 3792 | 3793 | 3794 | 3795 | 3796 | 3797 | 3798 |
3799 | 3800 | 3801 | 3802 | 3803 | 3804 | 3805 | 3806 | 3807 | 3808 | 3809 | 3810 | 3811 |
3812 | 3813 | 3814 | 3815 | 3816 | 3817 | 3818 | 3819 | 3820 | 3821 | 3822 | 3823 | 3824 |
3825 | 3826 | 3827 | 3828 | 3829 | 3830 | 3831 | 3832 | 3833 | 3834 | 3835 | 3836 | 3837 |
3838 | 3839 | 3840 | 3841 | 3842 | 3843 | 3844 | 3845 | 3846 | 3847 | 3848 | 3849 | 3850 |
3851 | 3852 | 3853 | 3854 | 3855 | 3856 | 3857 | 3858 | 3859 | 3860 | 3861 | 3862 | 3863 |
3864 | 3865 | 3866 | 3867 | 3868 | 3869 | 3870 | 3871 | 3872 | 3873 | 3874 | 3875 | 3876 |
3877 | 3878 | 3879 | 3880 | 3881 | 3882 | 3883 | 3884 | 3885 | 3886 | 3887 | 3888 | 3889 |

3890 | 3891 | 3892 | 3893 | 3894 | 3895 | 3896 | 3897 | 3898 | 3899 | 3900 | 3901 | 3902 |
3903 | 3904 | 3905 | 3906 | 3907 | 3908 | 3909 | 3910 | 3911 | 3912 | 3913 | 3914 | 3915 |
3916 | 3917 | 3918 | 3919 | 3920 | 3921 | 3922 | 3923 | 3924 | 3925 | 3926 | 3927 | 3928 |
3929 | 3930 | 3931 | 3932 | 3933 | 3934 | 3935 | 3936 | 3937 | 3938 | 3939 | 3940 | 3941 |
3942 | 3943 | 3944 | 3945 | 3946 | 3947 | 3948 | 3949 | 3950 | 3951 | 3952 | 3953 | 3954 |
3955 | 3956 | 3957 | 3958 | 3959 | 3960 | 3961 | 3962 | 3963 | 3964 | 3965 | 3966 | 3967 |
3968 | 3969 | 3970 | 3971 | 3972 | 3973 | 3974 | 3975 | 3976 | 3977 | 3978 | 3979 | 3980 |
3981 | 3982 | 3983 | 3984 | 3985 | 3986 | 3987 | 3988 | 3989 | 3990 | 3991 | 3992 | 3993 |
3994 | 3995 | 3996 | 3997 | 3998 | 3999 | 4000 | 4001 | 4002 | 4003 | 4004 | 4005 | 4006 |
4007 | 4008 | 4009 | 4010 | 4011 | 4012 | 4013 | 4014 | 4015 | 4016 | 4017 | 4018 | 4019 |
4020 | 4021 | 4022 | 4023 | 4024 | 4025 | 4026 | 4027 | 4028 | 4029 | 4030 | 4031 | 4032 |
4033 | 4034 | 4035 | 4036 | 4037 | 4038 | 4039 | 4040 | 4041 | 4042 | 4043 | 4044 | 4045 |
4046 | 4047 | 4048 | 4049 | 4050 | 4051 | 4052 | 4053 | 4054 | 4055 | 4056 | 4057 | 4058 |
4059 | 4060 | 4061 | 4062 | 4063 | 4064 | 4065 | 4066 | 4067 | 4068 | 4069 | 4070 | 4071 |
4072 | 4073 | 4074 | 4075 | 4076 | 4077 | 4078 | 4079 | 4080 | 4081 | 4082 | 4083 | 4084 |
4085 | 4086 | 4087 | 4088 | 4089 | 4090 | 4091 | 4092 | 4093 | 4094 | 4095 | 4096 | 4097 |
4098 | 4099 | 4100 | 4101 | 4102 | 4103 | 4104 | 4105 | 4106 | 4107 | 4108 | 4109 | 4110 |
4111 | 4112 | 4113 | 4114 | 4115 | 4116 | 4117 | 4118 | 4119 | 4120 | 4121 | 4122 | 4123 |
4124 | 4125 | 4126 | 4127 | 4128 | 4129 | 4130 | 4131 | 4132 | 4133 | 4134 | 4135 | 4136 |
4137 | 4138 | 4139 | 4140 | 4141 | 4142 | 4143 | 4144 | 4145 | 4146 | 4147 | 4148 | 4149 |
4150 | 4151 | 4152 | 4153 | 4154 | 4155 | 4156 | 4157 | 4158 | 4159 | 4160 | 4161 | 4162 |
4163 | 4164 | 4165 | 4166 | 4167 | 4168 | 4169 | 4170 | 4171 | 4172 | 4173 | 4174 | 4175 |
4176 | 4177 | 4178 | 4179 | 4180 | 4181 | 4182 | 4183 | 4184 | 4185 | 4186 | 4187 | 4188 |
4189 | 4190 | 4191 | 4192 | 4193 | 4194 | 4195 | 4196 | 4197 | 4198 | 4199 | 4200 | 4201 |
4202 | 4203 | 4204 | 4205 | 4206 | 4207 | 4208 | 4209 | 4210 | 4211 | 4212 | 4213 | 4214 |
4215 | 4216 | 4217 | 4218 | 4219 | 4220 | 4221 | 4222 | 4223 | 4224 | 4225 | 4226 | 4227 |
4228 | 4229 | 4230 | 4231 | 4232 | 4233 | 4234 | 4235 | 4236 | 4237 | 4238 | 4239 | 4240 |
4241 | 4242 | 4243 | 4244 | 4245 | 4246 | 4247 | 4248 | 4249 | 4250 | 4251 | 4252 | 4253 |
4254 | 4255 | 4256 | 4257 | 4258 | 4259 | 4260 | 4261 | 4262 | 4263 | 4264 | 4265 | 4266 |
4267 | 4268 | 4269 | 4270 | 4271 | 4272 | 4273 | 4274 | 4275 | 4276 | 4277 | 4278 | 4279 |
4280 | 4281 | 4282 | 4283 | 4284 | 4285 | 4286 | 4287 | 4288 | 4289 | 4290 | 4291 | 4292 |
4293 | 4294 | 4295 | 4296 | 4297 | 4298 | 4299 | 4300 | 4301 | 4302 | 4303 | 4304 | 4305 |
4306 | 4307 | 4308 | 4309 | 4310 | 4311 | 4312 | 4313 | 4314 | 4315 | 4316 | 4317 | 4318 |
4319 | 4320 | 4321 | 4322 | 4323 | 4324 | 4325 | 4326 | 4327 | 4328 | 4329 | 4330 | 4331 |
4332 | 4333 | 4334 | 4335 | 4336 | 4337 | 4338 | 4339 | 4340 | 4341 | 4342 | 4343 | 4344 |
4345 | 4346 | 4347 | 4348 | 4349 | 4350 | 4351 | 4352 | 4353 | 4354 | 4355 | 4356 | 4357 |
4358 | 4359 | 4360 | 4361 | 4362 | 4363 | 4364 | 4365 | 4366 | 4367 | 4368 | 4369 | 4370 |
4371 | 4372 | 4373 | 4374 | 4375 | 4376 | 4377 | 4378 | 4379 | 4380 | 4381 | 4382 | 4383 |
4384 | 4385 | 4386 | 4387 | 4388 | 4389 | 4390 | 4391 | 4392 | 4393 | 4394 | 4395 | 4396 |
4397 | 4398 | 4399 | 4400 | 4401 | 4402 | 4403 | 4404 | 4405 | 4406 | 4407 | 4408 | 4409 |
4410 | 4411 | 4412 | 4413 | 4414 | 4415 | 4416 | 4417 | 4418 | 4419 | 4420 | 4421 | 4422 |
4423 | 4424 | 4425 | 4426 | 4427 | 4428 | 4429 | 4430 | 4431 | 4432 | 4433 | 4434 | 4435 |

4436 | 4437 | 4438 | 4439 | 4440 | 4441 | 4442 | 4443 | 4444 | 4445 | 4446 | 4447 | 4448 |
4449 | 4450 | 4451 | 4452 | 4453 | 4454 | 4455 | 4456 | 4457 | 4458 | 4459 | 4460 | 4461 |
4462 | 4463 | 4464 | 4465 | 4466 | 4467 | 4468 | 4469 | 4470 | 4471 | 4472 | 4473 | 4474 |
4475 | 4476 | 4477 | 4478 | 4479 | 4480 | 4481 | 4482 | 4483 | 4484 | 4485 | 4486 | 4487 |
4488 | 4489 | 4490 | 4491 | 4492 | 4493 | 4494 | 4495 | 4496 | 4497 | 4498 | 4499 | 4500 |
4501 | 4502 | 4503 | 4504 | 4505 | 4506 | 4507 | 4508 | 4509 | 4510 | 4511 | 4512 | 4513 |
4514 | 4515 | 4516 | 4517 | 4518 | 4519 | 4520 | 4521 | 4522 | 4523 | 4524 | 4525 | 4526 |
4527 | 4528 | 4529 | 4530 | 4531 | 4532 | 4533 | 4534 | 4535 | 4536 | 4537 | 4538 | 4539 |
4540 | 4541 | 4542 | 4543 | 4544 | 4545 | 4546 | 4547 | 4548 | 4549 | 4550 | 4551 | 4552 |
4553 | 4554 | 4555 | 4556 | 4557 | 4558 | 4559 | 4560 | 4561 | 4562 | 4563 | 4564 | 4565 |
4566 | 4567 | 4568 | 4569 | 4570 | 4571 | 4572 | 4573 | 4574 | 4575 | 4576 | 4577 | 4578 |
4579 | 4580 | 4581 | 4582 | 4583 | 4584 | 4585 | 4586 | 4587 | 4588 | 4589 | 4590 | 4591 |
4592 | 4593 | 4594 | 4595 | 4596 | 4597 | 4598 | 4599 | 4600 | 4601 | 4602 | 4603 | 4604 |
4605 | 4606 | 4607 | 4608 | 4609 | 4610 | 4611 | 4612 | 4613 | 4614 | 4615 | 4616 | 4617 |
4618 | 4619 | 4620 | 4621 | 4622 | 4623 | 4624 | 4625 | 4626 | 4627 | 4628 | 4629 | 4630 |
4631 | 4632 | 4633 | 4634 | 4635 | 4636 | 4637 | 4638 | 4639 | 4640 | 4641 | 4642 | 4643 |
4644 | 4645 | 4646 | 4647 | 4648 | 4649 | 4650 | 4651 | 4652 | 4653 | 4654 | 4655 | 4656 |
4657 | 4658 | 4659 | 4660 | 4661 | 4662 | 4663 | 4664 | 4665 | 4666 | 4667 | 4668 | 4669 |
4670 | 4671 | 4672 | 4673 | 4674 | 4675 | 4676 | 4677 | 4678 | 4679 | 4680 | 4681 | 4682 |
4683 | 4684 | 4685 | 4686 | 4687 | 4688 | 4689 | 4690 | 4691 | 4692 | 4693 | 4694 | 4695 |
4696 | 4697 | 4698 | 4699 | 4700 | 4701 | 4702 | 4703 | 4704 | 4705 | 4706 | 4707 | 4708 |
4709 | 4710 | 4711 | 4712 | 4713 | 4714 | 4715 | 4716 | 4717 | 4718 | 4719 | 4720 | 4721 |
4722 | 4723 | 4724 | 4725 | 4726 | 4727 | 4728 | 4729 | 4730 | 4731 | 4732 | 4733 | 4734 |
4735 | 4736 | 4737 | 4738 | 4739 | 4740 | 4741 | 4742 | 4743 | 4744 | 4745 | 4746 | 4747 |
4748 | 4749 | 4750 | 4751 | 4752 | 4753 | 4754 | 4755 | 4756 | 4757 | 4758 | 4759 | 4760 |
4761 | 4762 | 4763 | 4764 | 4765 | 4766 | 4767 | 4768 | 4769 | 4770 | 4771 | 4772 | 4773 |
4774 | 4775 | 4776 | 4777 | 4778 | 4779 | 4780 | 4781 | 4782 | 4783 | 4784 | 4785 | 4786 |
4787 | 4788 | 4789 | 4790 | 4791 | 4792 | 4793 | 4794 | 4795 | 4796 | 4797 | 4798 | 4799 |
4800 | 4801 | 4802 | 4803 | 4804 | 4805 | 4806 | 4807 | 4808 | 4809 | 4810 | 4811 | 4812 |
4813 | 4814 | 4815 | 4816 | 4817 | 4818 | 4819 | 4820 | 4821 | 4822 | 4823 | 4824 | 4825 |
4826 | 4827 | 4828 | 4829 | 4830 | 4831 | 4832 | 4833 | 4834 | 4835 | 4836 | 4837 | 4838 |
4839 | 4840 | 4841 | 4842 | 4843 | 4844 | 4845 | 4846 | 4847 | 4848 | 4849 | 4850 | 4851 |
4852 | 4853 | 4854 | 4855 | 4856 | 4857 | 4858 | 4859 | 4860 | 4861 | 4862 | 4863 | 4864 |
4865 | 4866 | 4867 | 4868 | 4869 | 4870 | 4871 | 4872 | 4873 | 4874 | 4875 | 4876 | 4877 |
4878 | 4879 | 4880 | 4881 | 4882 | 4883 | 4884 | 4885 | 4886 | 4887 | 4888 | 4889 | 4890 |
4891 | 4892 | 4893 | 4894 | 4895 | 4896 | 4897 | 4898 | 4899 | 4900 | 4901 | 4902 | 4903 |
4904 | 4905 | 4906 | 4907 | 4908 | 4909 | 4910 | 4911 | 4912 | 4913 | 4914 | 4915 | 4916 |
4917 | 4918 | 4919 | 4920 | 4921 | 4922 | 4923 | 4924 | 4925 | 4926 | 4927 | 4928 | 4929 |
4930 | 4931 | 4932 | 4933 | 4934 | 4935 | 4936 | 4937 | 4938 | 4939 | 4940 | 4941 | 4942 |
4943 | 4944 | 4945 | 4946 | 4947 | 4948 | 4949 | 4950 | 4951 | 4952 | 4953 | 4954 | 4955 |
4956 | 4957 | 4958 | 4959 | 4960 | 4961 | 4962 | 4963 | 4964 | 4965 | 4966 | 4967 | 4968 |
4969 | 4970 | 4971 | 4972 | 4973 | 4974 | 4975 | 4976 | 4977 | 4978 | 4979 | 4980 | 4981 |

4982 | 4983 | 4984 | 4985 | 4986 | 4987 | 4988 | 4989 | 4990 | 4991 | 4992 | 4993 | 4994 |
4995 | 4996 | 4997 | 4998 | 4999 | 5000 | 5001 | 5002 | 5003 | 5004 | 5005 | 5006 | 5007 |
5008 | 5009 | 5010 | 5011 | 5012 | 5013 | 5014 | 5015 | 5016 | 5017 | 5018 | 5019 | 5020 |
5021 | 5022 | 5023 | 5024 | 5025 | 5026 | 5027 | 5028 | 5029 | 5030 | 5031 | 5032 | 5033 |
5034 | 5035 | 5036 | 5037 | 5038 | 5039 | 5040 | 5041 | 5042 | 5043 | 5044 | 5045 | 5046 |
5047 | 5048 | 5049 | 5050 | 5051 | 5052 | 5053 | 5054 | 5055 | 5056 | 5057 | 5058 | 5059 |
5060 | 5061 | 5062 | 5063 | 5064 | 5065 | 5066 | 5067 | 5068 | 5069 | 5070 | 5071 | 5072 |
5073 | 5074 | 5075 | 5076 | 5077 | 5078 | 5079 | 5080 | 5081 | 5082 | 5083 | 5084 | 5085 |
5086 | 5087 | 5088 | 5089 | 5090 | 5091 | 5092 | 5093 | 5094 | 5095 | 5096 | 5097 | 5098 |
5099 | 5100 | 5101 | 5102 | 5103 | 5104 | 5105 | 5106 | 5107 | 5108 | 5109 | 5110 | 5111 |
5112 | 5113 | 5114 | 5115 | 5116 | 5117 | 5118 | 5119 | 5120 | 5121 | 5122 | 5123 | 5124 |
5125 | 5126 | 5127 | 5128 | 5129 | 5130 | 5131 | 5132 | 5133 | 5134 | 5135 | 5136 | 5137 |
5138 | 5139 | 5140 | 5141 | 5142 | 5143 | 5144 | 5145 | 5146 | 5147 | 5148 | 5149 | 5150 |
5151 | 5152 | 5153 | 5154 | 5155 | 5156 | 5157 | 5158 | 5159 | 5160 | 5161 | 5162 | 5163 |
5164 | 5165 | 5166 | 5167 | 5168 | 5169 | 5170 | 5171 | 5172 | 5173 | 5174 | 5175 | 5176 |
5177 | 5178 | 5179 | 5180 | 5181 | 5182 | 5183 | 5184 | 5185 | 5186 | 5187 | 5188 | 5189 |
5190 | 5191 | 5192 | 5193 | 5194 | 5195 | 5196 | 5197 | 5198 | 5199 | 5200 | 5201 | 5202 |
5203 | 5204 | 5205 | 5206 | 5207 | 5208 | 5209 | 5210 | 5211 | 5212 | 5213 | 5214 | 5215 |
5216 | 5217 | 5218 | 5219 | 5220 | 5221 | 5222 | 5223 | 5224 | 5225 | 5226 | 5227 | 5228 |
5229 | 5230 | 5231 | 5232 | 5233 | 5234 | 5235 | 5236 | 5237 | 5238 | 5239 | 5240 | 5241 |
5242 | 5243 | 5244 | 5245 | 5246 | 5247 | 5248 | 5249 | 5250 | 5251 | 5252 | 5253 | 5254 |
5255 | 5256 | 5257 | 5258 | 5259 | 5260 | 5261 | 5262 | 5263 | 5264 | 5265 | 5266 | 5267 |
5268 | 5269 | 5270 | 5271 | 5272 | 5273 | 5274 | 5275 | 5276 | 5277 | 5278 | 5279 | 5280 |
5281 | 5282 | 5283 | 5284 | 5285 | 5286 | 5287 | 5288 | 5289 | 5290 | 5291 | 5292 | 5293 |
5294 | 5295 | 5296 | 5297 | 5298 | 5299 | 5300 | 5301 | 5302 | 5303 | 5304 | 5305 | 5306 |
5307 | 5308 | 5309 | 5310 | 5311 | 5312 | 5313 | 5314 | 5315 | 5316 | 5317 | 5318 | 5319 |
5320 | 5321 | 5322 | 5323 | 5324 | 5325 | 5326 | 5327 | 5328 | 5329 | 5330 | 5331 | 5332 |
5333 | 5334 | 5335 | 5336 | 5337 | 5338 | 5339 | 5340 | 5341 | 5342 | 5343 | 5344 | 5345 |
5346 | 5347 | 5348 | 5349 | 5350 | 5351 | 5352 | 5353 | 5354 | 5355 | 5356 | 5357 | 5358 |
5359 | 5360 | 5361 | 5362 | 5363 | 5364 | 5365 | 5366 | 5367 | 5368 | 5369 | 5370 | 5371 |
5372 | 5373 | 5374 | 5375 | 5376 | 5377 | 5378 | 5379 | 5380 | 5381 | 5382 | 5383 | 5384 |
5385 | 5386 | 5387 | 5388 | 5389 | 5390 | 5391 | 5392 | 5393 | 5394 | 5395 | 5396 | 5397 |
5398 | 5399 | 5400 | 5401 | 5402 | 5403 | 5404 | 5405 | 5406 | 5407 | 5408 | 5409 | 5410 |
5411 | 5412 | 5413 | 5414 | 5415 | 5416 | 5417 | 5418 | 5419 | 5420 | 5421 | 5422 | 5423 |
5424 | 5425 | 5426 | 5427 | 5428 | 5429 | 5430 | 5431 | 5432 | 5433 | 5434 | 5435 | 5436 |
5437 | 5438 | 5439 | 5440 | 5441 | 5442 | 5443 | 5444 | 5445 | 5446 | 5447 | 5448 | 5449 |
5450 | 5451 | 5452 | 5453 | 5454 | 5455 | 5456 | 5457 | 5458 | 5459 | 5460 | 5461 | 5462 |
5463 | 5464 | 5465 | 5466 | 5467 | 5468 | 5469 | 5470 | 5471 | 5472 | 5473 | 5474 | 5475 |
5476 | 5477 | 5478 | 5479 | 5480 | 5481 | 5482 | 5483 | 5484 | 5485 | 5486 | 5487 | 5488 |
5489 | 5490 | 5491 | 5492 | 5493 | 5494 | 5495 | 5496 | 5497 | 5498 | 5499 | 5500 | 5501 |
5502 | 5503 | 5504 | 5505 | 5506 | 5507 | 5508 | 5509 | 5510 | 5511 | 5512 | 5513 | 5514 |
5515 | 5516 | 5517 | 5518 | 5519 | 5520 | 5521 | 5522 | 5523 | 5524 | 5525 | 5526 | 5527 |

5528 | 5529 | 5530 | 5531 | 5532 | 5533 | 5534 | 5535 | 5536 | 5537 | 5538 | 5539 | 5540 |
5541 | 5542 | 5543 | 5544 | 5545 | 5546 | 5547 | 5548 | 5549 | 5550 | 5551 | 5552 | 5553 |
5554 | 5555 | 5556 | 5557 | 5558 | 5559 | 5560 | 5561 | 5562 | 5563 | 5564 | 5565 | 5566 |
5567 | 5568 | 5569 | 5570 | 5571 | 5572 | 5573 | 5574 | 5575 | 5576 | 5577 | 5578 | 5579 |
5580 | 5581 | 5582 | 5583 | 5584 | 5585 | 5586 | 5587 | 5588 | 5589 | 5590 | 5591 | 5592 |
5593 | 5594 | 5595 | 5596 | 5597 | 5598 | 5599 | 5600 | 5601 | 5602 | 5603 | 5604 | 5605 |
5606 | 5607 | 5608 | 5609 | 5610 | 5611 | 5612 | 5613 | 5614 | 5615 | 5616 | 5617 | 5618 |
5619 | 5620 | 5621 | 5622 | 5623 | 5624 | 5625 | 5626 | 5627 | 5628 | 5629 | 5630 | 5631 |
5632 | 5633 | 5634 | 5635 | 5636 | 5637 | 5638 | 5639 | 5640 | 5641 | 5642 | 5643 | 5644 |
5645 | 5646 | 5647 | 5648 | 5649 | 5650 | 5651 | 5652 | 5653 | 5654 | 5655 | 5656 | 5657 |
5658 | 5659 | 5660 | 5661 | 5662 | 5663 | 5664 | 5665 | 5666 | 5667 | 5668 | 5669 | 5670 |
5671 | 5672 | 5673 | 5674 | 5675 | 5676 | 5677 | 5678 | 5679 | 5680 | 5681 | 5682 | 5683 |
5684 | 5685 | 5686 | 5687 | 5688 | 5689 | 5690 | 5691 | 5692 | 5693 | 5694 | 5695 | 5696 |
5697 | 5698 | 5699 | 5700 | 5701 | 5702 | 5703 | 5704 | 5705 | 5706 | 5707 | 5708 | 5709 |
5710 | 5711 | 5712 | 5713 | 5714 | 5715 | 5716 | 5717 | 5718 | 5719 | 5720 | 5721 | 5722 |
5723 | 5724 | 5725 | 5726 | 5727 | 5728 | 5729 | 5730 | 5731 | 5732 | 5733 | 5734 | 5735 |
5736 | 5737 | 5738 | 5739 | 5740 | 5741 | 5742 | 5743 | 5744 | 5745 | 5746 | 5747 | 5748 |
5749 | 5750 | 5751 | 5752 | 5753 | 5754 | 5755 | 5756 | 5757 | 5758 | 5759 | 5760 | 5761 |
5762 | 5763 | 5764 | 5765 | 5766 | 5767 | 5768 | 5769 | 5770 | 5771 | 5772 | 5773 | 5774 |
5775 | 5776 | 5777 | 5778 | 5779 | 5780 | 5781 | 5782 | 5783 | 5784 | 5785 | 5786 | 5787 |
5788 | 5789 | 5790 | 5791 | 5792 | 5793 | 5794 | 5795 | 5796 | 5797 | 5798 | 5799 | 5800 |
5801 | 5802 | 5803 | 5804 | 5805 | 5806 | 5807 | 5808 | 5809 | 5810 | 5811 | 5812 | 5813 |
5814 | 5815 | 5816 | 5817 | 5818 | 5819 | 5820 | 5821 | 5822 | 5823 | 5824 | 5825 | 5826 |
5827 | 5828 | 5829 | 5830 | 5831 | 5832 | 5833 | 5834 | 5835 | 5836 | 5837 | 5838 | 5839 |
5840 | 5841 | 5842 | 5843 | 5844 | 5845 | 5846 | 5847 | 5848 | 5849 | 5850 | 5851 | 5852 |
5853 | 5854 | 5855 | 5856 | 5857 | 5858 | 5859 | 5860 | 5861 | 5862 | 5863 | 5864 | 5865 |
5866 | 5867 | 5868 | 5869 | 5870 | 5871 | 5872 | 5873 | 5874 | 5875 | 5876 | 5877 | 5878 |
5879 | 5880 | 5881 | 5882 | 5883 | 5884 | 5885 | 5886 | 5887 | 5888 | 5889 | 5890 | 5891 |
5892 | 5893 | 5894 | 5895 | 5896 | 5897 | 5898 | 5899 | 5900 | 5901 | 5902 | 5903 | 5904 |
5905 | 5906 | 5907 | 5908 | 5909 | 5910 | 5911 | 5912 | 5913 | 5914 | 5915 | 5916 | 5917 |
5918 | 5919 | 5920 | 5921 | 5922 | 5923 | 5924 | 5925 | 5926 | 5927 | 5928 | 5929 | 5930 |
5931 | 5932 | 5933 | 5934 | 5935 | 5936 | 5937 | 5938 | 5939 | 5940 | 5941 | 5942 | 5943 |
5944 | 5945 | 5946 | 5947 | 5948 | 5949 | 5950 | 5951 | 5952 | 5953 | 5954 | 5955 | 5956 |
5957 | 5958 | 5959 | 5960 | 5961 | 5962 | 5963 | 5964 | 5965 | 5966 | 5967 | 5968 | 5969 |
5970 | 5971 | 5972 | 5973 | 5974 | 5975 | 5976 | 5977 | 5978 | 5979 | 5980 | 5981 | 5982 |
5983 | 5984 | 5985 | 5986 | 5987 | 5988 | 5989 | 5990 | 5991 | 5992 | 5993 | 5994 | 5995 |
5996 | 5997 | 5998 | 5999 | 6000] | [1502]

==========

Program10:WNOSQL

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<USE> WDBA;

<DATALIB>  ps
<DATA>
public  <CLASS>  DATA

{



public void main()

{
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");



        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);



char  c=' ';
```

```
String  s2 ="SelectCols from datastorehg 0 to 1000 , 1 to 1 ?= C By 1 1 : {3,4,5,6,7,8} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s2,  t));

//select  all  the  column {3,4,5,6,7,8} values   stored  in  the  table  datastorehg




String  s3 ="Count() from datastorehg 0 to 1000 , 1 to 1 ?= 3001 By 1 1 : {0} : {0} :{0}";

 WDBA.writeln("datacount="+WDBALIB.WDBAQUERY(  s3,  t));

// to  count  the  occurance  of  data 3001  from  table  datastorehg with  in a given  range
// startingindex  and  endingindex  from  the  table.




String  s4 ="DISTINCT from datastorehg 0 to 1000 , 1 to 1 ?= C By 1 1 : {0,0,1,1,5,5} : {0} :{0}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s4,  t));
```

// distinct which is used to remove duplicates of columns

}

}

}

OUTPUT

======

W N O S Q L –wnosql D A T A B A S E Non sql(*) JeminInformationTechnology copy right 2014 @ all rights reserved

dbpwds ttdbpwds Verified db password[wilmix78] continue to access WNOSQL database

Listing Table datastorehg contents ... IP=1 ---------- ---------- ---------- ---------- ---------- ---------- 3 | 4 | 5 | 6 | 7 | 8 |

SALARY | 3001 | 3002 | 3003 | 3004 | 3005 |

[SALARY, 3001, 3002, 3003, 3004, 3005]

datacount=[1]

[SALARY, EMPLOYEENAME, 3002]

Program11:WNOSQL

AGGREGATE FUNCTIONS :

```
String  s390 ="AVG() from datastorehg 5 to 1000 , 1 to 1 ?= 3001 By 1 1 : {5,6,8,9} : {0} :{0}";

 WDBA.writeln("datacount="+WDBALIB.WDBAQUERY(  s390,  t));

// AVG  for  the  columns  5,6,8,9   from  the   table  datastorehg



String  s3909 ="MIN() from datastorehg 5 to 1000 , 1 to 1 ?= 3001 By 1 1 : {5,6,8,9} : {0} :{0}";

 WDBA.writeln("datacount="+WDBALIB.WDBAQUERY(  s3909,  t));

// MIN  for  the  values  in  the   columns  5,6,8,9   from the  table  datastorehg


String  s39068 ="SUM() from datastorehg 0 to 1000 , 1 to 1 ?= 3001 By 1 1 : {5,6,8,9} : {0} :{0}";

 WDBA.writeln("datacount="+WDBALIB.WDBAQUERY(  s39068,  t));



// SUM  for  the  values  in  the   columns  5,6,8,9   from the  table  datastorehg
```

OUTPUT:

======

```
datacount=[12.016]

datacount=[3002.0]

datacount=[12016.0]
```

==========

Program12:WNOSQL

```
String  s114 ="INNERJOIN from datastorehg 3 to 29 , 1 to 1 ?= datastorehg By 1 1 : {1,2,3} : {1,2,3} :{1,2,3}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s114,  t));

// Innerjoin exists when there exits between two tables



String  s12 ="RIGHTJOIN from datastorehg 10 to 1000 , 1 to 1 ?= datastoreh By 1 1 : {1,2,3,5,10,100} : {1,2,3,11,12,13} :{1,2,3}";

 WDBA.writeln(""+WDBALIB.WDBAQUERY(  s12,  t));

//   This wnosql is to perform rightjoin  between two tables



String s1tabh = "Insert from datastorehg 0 to 0 , 1 to 1 ?= 6639 By 1 f(x) : {267,4344,4333,4333,5544,5455,54544,66565,6565}: {0} :{0}";

ArrayList artabh= WDBALIB.WDBAQUERY(s1tabh, t);
```

```
WDBA.writeln("ans14="+artabh);

// insert the values into the table



String s1tabhj = "CLUSTER from datastorehg 0 to 9 , 1 to 1 ?= x By 1 f(x) :
{267,4344,4333,4333,5544,5455,54544,66565,6565}: {0} :{0}";

ArrayList artabhj= WDBALIB.WDBAQUERY(s1tabhj, t);

WDBA.writeln("ans15="+artabhj);

// create a cluster for the table

ie) CLUSTER to store group of data in a encrypted form for futhure use.



String s1tabhj1 = "CLUSTERPROPERTY from datastorehg 0 to 9 , 1 to 1 ?= x By 1 f(x) :
{267,4344,4333,4333,5544,5455,54544,66565,6565}: {0} :{0}";

ArrayList artabhj1= WDBALIB.WDBAQUERY(s1tabhj1, t);

WDBA.writeln("ans15="+artabhj1);

// To compute clustertable size, display data, display system date, Display remaning
//space available to store values in a cluster table.
```

String s1tabhj1f = "BACKUPCLUSTER from datastorehg 0 to 9 , 1 to 1 ?= x By 1 f(x) : {267,4344,4333,4333,5544,5455,54544,66565,6565}: {0} :{0}";

ArrayList artabhj11= WDBALIB.WDBAQUERY(s1tabhj1f, t);

WDBA.writeln("ans151="+artabhj11);

//TO RESTORE the Lost CLUSTER DATA and automatically store the contents in a t able.

String s1g ="SELECTRVAL from datastorehg 0 to 9 , 1 to 1 ?= C By 1 1 : {0} : {0} :{1}";

ArrayList arjkk= WDBALIB.WDBAQUERY(s1g, t);

WDBA.writeln("ans151j="+arjkk);

Output:

========

[SALARY, EMPLOYEENAME, SALARY] [SALARY, EMPLOYEENAME, SALARY, 3002, 8, 3007, 9, 3097, 10] ans14=[Inserted values] converting to class file is completed successfully. ans15=[[0=267, 4344, 4333, 4333, 5544, 5455, 54544, 66565, 6565]] ans15=[CLUSTER SIZE=9, CLUSTER DATA=[{0=267, 4344, 4333, 4333, 5544, 5455, 54544, 66565, 6565}], SYSTEM DATE=Fri May 05 21:09:30 IST 2017, WNOSQL(*) CLUSTER SPACE AVAILABLE=2991] converting to class file is completed successfully. ans151=[[0=267, 4344, 4333, 4333, 5544, 5455, 54544, 66565, 6565]] [EMPLOYEENAME SALARY EMPLOYEENAME SALARY 3001 3002

3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018
3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034
3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050
3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066
3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082
3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098
3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114
3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130
3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146
3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162
3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178
3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194
3195 3196 3197 3198 3199 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210
3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226
3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242
3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258
3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274
3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290
3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306
3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322
3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338
3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354
3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370
3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386
3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402
3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418
3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430 3431 3432 3433 3434
3435 3436 3437 3438 3439 3440 3441 3442 3443 3444 3445 3446 3447 3448 3449 3450
3451 3452 3453 3454 3455 3456 3457 3458 3459 3460 3461 3462 3463 3464 3465 3466
3467 3468 3469 3470 3471 3472 3473 3474 3475 3476 3477 3478 3479 3480 3481 3482
3483 3484 3485 3486 3487 3488 3489 3490 3491 3492 3493 3494 3495 3496 3497 3498
3499 3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514
3515 3516 3517 3518 3519 3520 3521 3522 3523 3524 3525 3526 3527 3528 3529 3530
3531 3532 3533 3534 3535 3536 3537 3538 3539 3540 3541 3542 3543 3544 3545 3546
3547 3548 3549 3550 3551 3552 3553 3554 3555 3556 3557 3558 3559 3560 3561 3562
3563 3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 3574 3575 3576 3577 3578
3579 3580 3581 3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 3592 3593 3594
3595 3596 3597 3598 3599 3600 3601 3602 3603 3604 3605 3606 3607 3608 3609 3610
3611 3612 3613 3614 3615 3616 3617 3618 3619 3620 3621 3622 3623 3624 3625 3626
3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 3637 3638 3639 3640 3641 3642
3643 3644 3645 3646 3647 3648 3649 3650 3651 3652 3653 3654 3655 3656 3657 3658
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674

3675 3676 3677 3678 3679 3680 3681 3682 3683 3684 3685 3686 3687 3688 3689 3690
3691 3692 3693 3694 3695 3696 3697 3698 3699 3700 3701 3702 3703 3704 3705 3706
3707 3708 3709 3710 3711 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722
3723 3724 3725 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 3738
3739 3740 3741 3742 3743 3744 3745 3746 3747 3748 3749 3750 3751 3752 3753 3754
3755 3756 3757 3758 3759 3760 3761 3762 3763 3764 3765 3766 3767 3768 3769 3770
3771 3772 3773 3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786
3787 3788 3789 3790 3791 3792 3793 3794 3795 3796 3797 3798 3799 3800 3801 3802
3803 3804 3805 3806 3807 3808 3809 3810 3811 3812 3813 3814 3815 3816 3817 3818
3819 3820 3821 3822 3823 3824 3825 3826 3827 3828 3829 3830 3831 3832 3833 3834
3835 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 3846 3847 3848 3849 3850
3851 3852 3853 3854 3855 3856 3857 3858 3859 3860 3861 3862 3863 3864 3865 3866
3867 3868 3869 3870 3871 3872 3873 3874 3875 3876 3877 3878 3879 3880 3881 3882
3883 3884 3885 3886 3887 3888 3889 3890 3891 3892 3893 3894 3895 3896 3897 3898
3899 3900 3901 3902 3903 3904 3905 3906 3907 3908 3909 3910 3911 3912 3913 3914
3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930
3931 3932 3933 3934 3935 3936 3937 3938 3939 3940 3941 3942 3943 3944 3945 3946
3947 3948 3949 3950 3951 3952 3953 3954 3955 3956 3957 3958 3959 3960 3961 3962
3963 3964 3965 3966 3967 3968 3969 3970 3971 3972 3973 3974 3975 3976 3977 3978
3979 3980 3981 3982 3983 3984 3985 3986 3987 3988 3989 3990 3991 3992 3993 3994
3995 3996 3997 3998 3999 4000 4001 4002 4003 4004 4005 4006 4007 4008 4009 4010
4011 4012 4013 4014 4015 4016 4017 4018 4019 4020 4021 4022 4023 4024 4025 4026
4027 4028 4029 4030 4031 4032 4033 4034 4035 4036 4037 4038 4039 4040 4041 4042
4043 4044 4045 4046 4047 4048 4049 4050 4051 4052 4053 4054 4055 4056 4057 4058
4059 4060 4061 4062 4063 4064 4065 4066 4067 4068 4069 4070 4071 4072 4073 4074
4075 4076 4077 4078 4079 4080 4081 4082 4083 4084 4085 4086 4087 4088 4089 4090
4091 4092 4093 4094 4095 4096 4097 4098 4099 4100 4101 4102 4103 4104 4105 4106
4107 4108 4109 4110 4111 4112 4113 4114 4115 4116 4117 4118 4119 4120 4121 4122
4123 4124 4125 4126 4127 4128 4129 4130 4131 4132 4133 4134 4135 4136 4137 4138
4139 4140 4141 4142 4143 4144 4145 4146 4147 4148 4149 4150 4151 4152 4153 4154
4155 4156 4157 4158 4159 4160 4161 4162 4163 4164 4165 4166 4167 4168 4169 4170
4171 4172 4173 4174 4175 4176 4177 4178 4179 4180 4181 4182 4183 4184 4185 4186
4187 4188 4189 4190 4191 4192 4193 4194 4195 4196 4197 4198 4199 4200 4201 4202
4203 4204 4205 4206 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218
4219 4220 4221 4222 4223 4224 4225 4226 4227 4228 4229 4230 4231 4232 4233 4234
4235 4236 4237 4238 4239 4240 4241 4242 4243 4244 4245 4246 4247 4248 4249 4250
4251 4252 4253 4254 4255 4256 4257 4258 4259 4260 4261 4262 4263 4264 4265 4266
4267 4268 4269 4270 4271 4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282
4283 4284 4285 4286 4287 4288 4289 4290 4291 4292 4293 4294 4295 4296 4297 4298
4299 4300 4301 4302 4303 4304 4305 4306 4307 4308 4309 4310 4311 4312 4313 4314
4315 4316 4317 4318 4319 4320 4321 4322 4323 4324 4325 4326 4327 4328 4329 4330
4331 4332 4333 4334 4335 4336 4337 4338 4339 4340 4341 4342 4343 4344 4345 4346

4347 4348 4349 4350 4351 4352 4353 4354 4355 4356 4357 4358 4359 4360 4361 4362
4363 4364 4365 4366 4367 4368 4369 4370 4371 4372 4373 4374 4375 4376 4377 4378
4379 4380 4381 4382 4383 4384 4385 4386 4387 4388 4389 4390 4391 4392 4393 4394
4395 4396 4397 4398 4399 4400 4401 4402 4403 4404 4405 4406 4407 4408 4409 4410
4411 4412 4413 4414 4415 4416 4417 4418 4419 4420 4421 4422 4423 4424 4425 4426
4427 4428 4429 4430 4431 4432 4433 4434 4435 4436 4437 4438 4439 4440 4441 4442
4443 4444 4445 4446 4447 4448 4449 4450 4451 4452 4453 4454 4455 4456 4457 4458
4459 4460 4461 4462 4463 4464 4465 4466 4467 4468 4469 4470 4471 4472 4473 4474
4475 4476 4477 4478 4479 4480 4481 4482 4483 4484 4485 4486 4487 4488 4489 4490
4491 4492 4493 4494 4495 4496 4497 4498 4499 4500 4501 4502 4503 4504 4505 4506
4507 4508 4509 4510 4511 4512 4513 4514 4515 4516 4517 4518 4519 4520 4521 4522
4523 4524 4525 4526 4527 4528 4529 4530 4531 4532 4533 4534 4535 4536 4537 4538
4539 4540 4541 4542 4543 4544 4545 4546 4547 4548 4549 4550 4551 4552 4553 4554
4555 4556 4557 4558 4559 4560 4561 4562 4563 4564 4565 4566 4567 4568 4569 4570
4571 4572 4573 4574 4575 4576 4577 4578 4579 4580 4581 4582 4583 4584 4585 4586
4587 4588 4589 4590 4591 4592 4593 4594 4595 4596 4597 4598 4599 4600 4601 4602
4603 4604 4605 4606 4607 4608 4609 4610 4611 4612 4613 4614 4615 4616 4617 4618
4619 4620 4621 4622 4623 4624 4625 4626 4627 4628 4629 4630 4631 4632 4633 4634
4635 4636 4637 4638 4639 4640 4641 4642 4643 4644 4645 4646 4647 4648 4649 4650
4651 4652 4653 4654 4655 4656 4657 4658 4659 4660 4661 4662 4663 4664 4665 4666
4667 4668 4669 4670 4671 4672 4673 4674 4675 4676 4677 4678 4679 4680 4681 4682
4683 4684 4685 4686 4687 4688 4689 4690 4691 4692 4693 4694 4695 4696 4697 4698
4699 4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714
4715 4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730
4731 4732 4733 4734 4735 4736 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746
4747 4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762
4763 4764 4765 4766 4767 4768 4769 4770 4771 4772 4773 4774 4775 4776 4777 4778
4779 4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794
4795 4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810
4811 4812 4813 4814 4815 4816 4817 4818 4819 4820 4821 4822 4823 4824 4825 4826
4827 4828 4829 4830 4831 4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842
4843 4844 4845 4846 4847 4848 4849 4850 4851 4852 4853 4854 4855 4856 4857 4858
4859 4860 4861 4862 4863 4864 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874
4875 4876 4877 4878 4879 4880 4881 4882 4883 4884 4885 4886 4887 4888 4889 4890
4891 4892 4893 4894 4895 4896 4897 4898 4899 4900 4901 4902 4903 4904 4905 4906
4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919 4920 4921 4922
4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938
4939 4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954
4955 4956 4957 4958 4959 4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970
4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982 4983 4984 4985 4986
4987 4988 4989 4990 4991 4992 4993 4994 4995 4996 4997 4998 4999 5000 5001 5002
5003 5004 5005 5006 5007 5008 5009 5010 5011 5012 5013 5014 5015 5016 5017 5018

5019 5020 5021 5022 5023 5024 5025 5026 5027 5028 5029 5030 5031 5032 5033 5034
5035 5036 5037 5038 5039 5040 5041 5042 5043 5044 5045 5046 5047 5048 5049 5050
5051 5052 5053 5054 5055 5056 5057 5058 5059 5060 5061 5062 5063 5064 5065 5066
5067 5068 5069 5070 5071 5072 5073 5074 5075 5076 5077 5078 5079 5080 5081 5082
5083 5084 5085 5086 5087 5088 5089 5090 5091 5092 5093 5094 5095 5096 5097 5098
5099 5100 5101 5102 5103 5104 5105 5106 5107 5108 5109 5110 5111 5112 5113 5114
5115 5116 5117 5118 5119 5120 5121 5122 5123 5124 5125 5126 5127 5128 5129 5130
5131 5132 5133 5134 5135 5136 5137 5138 5139 5140 5141 5142 5143 5144 5145 5146
5147 5148 5149 5150 5151 5152 5153 5154 5155 5156 5157 5158 5159 5160 5161 5162
5163 5164 5165 5166 5167 5168 5169 5170 5171 5172 5173 5174 5175 5176 5177 5178
5179 5180 5181 5182 5183 5184 5185 5186 5187 5188 5189 5190 5191 5192 5193 5194
5195 5196 5197 5198 5199 5200 5201 5202 5203 5204 5205 5206 5207 5208 5209 5210
5211 5212 5213 5214 5215 5216 5217 5218 5219 5220 5221 5222 5223 5224 5225 5226
5227 5228 5229 5230 5231 5232 5233 5234 5235 5236 5237 5238 5239 5240 5241 5242
5243 5244 5245 5246 5247 5248 5249 5250 5251 5252 5253 5254 5255 5256 5257 5258
5259 5260 5261 5262 5263 5264 5265 5266 5267 5268 5269 5270 5271 5272 5273 5274
5275 5276 5277 5278 5279 5280 5281 5282 5283 5284 5285 5286 5287 5288 5289 5290
5291 5292 5293 5294 5295 5296 5297 5298 5299 5300 5301 5302 5303 5304 5305 5306
5307 5308 5309 5310 5311 5312 5313 5314 5315 5316 5317 5318 5319 5320 5321 5322
5323 5324 5325 5326 5327 5328 5329 5330 5331 5332 5333 5334 5335 5336 5337 5338
5339 5340 5341 5342 5343 5344 5345 5346 5347 5348 5349 5350 5351 5352 5353 5354
5355 5356 5357 5358 5359 5360 5361 5362 5363 5364 5365 5366 5367 5368 5369 5370
5371 5372 5373 5374 5375 5376 5377 5378 5379 5380 5381 5382 5383 5384 5385 5386
5387 5388 5389 5390 5391 5392 5393 5394 5395 5396 5397 5398 5399 5400 5401 5402
5403 5404 5405 5406 5407 5408 5409 5410 5411 5412 5413 5414 5415 5416 5417 5418
5419 5420 5421 5422 5423 5424 5425 5426 5427 5428 5429 5430 5431 5432 5433 5434
5435 5436 5437 5438 5439 5440 5441 5442 5443 5444 5445 5446 5447 5448 5449 5450
5451 5452 5453 5454 5455 5456 5457 5458 5459 5460 5461 5462 5463 5464 5465 5466
5467 5468 5469 5470 5471 5472 5473 5474 5475 5476 5477 5478 5479 5480 5481 5482
5483 5484 5485 5486 5487 5488 5489 5490 5491 5492 5493 5494 5495 5496 5497 5498
5499 5500 5501 5502 5503 5504 5505 5506 5507 5508 5509 5510 5511 5512 5513 5514
5515 5516 5517 5518 5519 5520 5521 5522 5523 5524 5525 5526 5527 5528 5529 5530
5531 5532 5533 5534 5535 5536 5537 5538 5539 5540 5541 5542 5543 5544 5545 5546
5547 5548 5549 5550 5551 5552 5553 5554 5555 5556 5557 5558 5559 5560 5561 5562
5563 5564 5565 5566 5567 5568 5569 5570 5571 5572 5573 5574 5575 5576 5577 5578
5579 5580 5581 5582 5583 5584 5585 5586 5587 5588 5589 5590 5591 5592 5593 5594
5595 5596 5597 5598 5599 5600 5601 5602 5603 5604 5605 5606 5607 5608 5609 5610
5611 5612 5613 5614 5615 5616 5617 5618 5619 5620 5621 5622 5623 5624 5625 5626
5627 5628 5629 5630 5631 5632 5633 5634 5635 5636 5637 5638 5639 5640 5641 5642
5643 5644 5645 5646 5647 5648 5649 5650 5651 5652 5653 5654 5655 5656 5657 5658
5659 5660 5661 5662 5663 5664 5665 5666 5667 5668 5669 5670 5671 5672 5673 5674
5675 5676 5677 5678 5679 5680 5681 5682 5683 5684 5685 5686 5687 5688 5689 5690

5691 5692 5693 5694 5695 5696 5697 5698 5699 5700 5701 5702 5703 5704 5705 5706
5707 5708 5709 5710 5711 5712 5713 5714 5715 5716 5717 5718 5719 5720 5721 5722
5723 5724 5725 5726 5727 5728 5729 5730 5731 5732 5733 5734 5735 5736 5737 5738
5739 5740 5741 5742 5743 5744 5745 5746 5747 5748 5749 5750 5751 5752 5753 5754
5755 5756 5757 5758 5759 5760 5761 5762 5763 5764 5765 5766 5767 5768 5769 5770
5771 5772 5773 5774 5775 5776 5777 5778 5779 5780 5781 5782 5783 5784 5785 5786
5787 5788 5789 5790 5791 5792 5793 5794 5795 5796 5797 5798 5799 5800 5801 5802
5803 5804 5805 5806 5807 5808 5809 5810 5811 5812 5813 5814 5815 5816 5817 5818
5819 5820 5821 5822 5823 5824 5825 5826 5827 5828 5829 5830 5831 5832 5833 5834
5835 5836 5837 5838 5839 5840 5841 5842 5843 5844 5845 5846 5847 5848 5849 5850
5851 5852 5853 5854 5855 5856 5857 5858 5859 5860 5861 5862 5863 5864 5865 5866
5867 5868 5869 5870 5871 5872 5873 5874 5875 5876 5877 5878 5879 5880 5881 5882
5883 5884 5885 5886 5887 5888 5889 5890 5891 5892 5893 5894 5895 5896 5897 5898
5899 5900 5901 5902 5903 5904 5905 5906 5907 5908 5909 5910 5911 5912 5913 5914
5915 5916 5917 5918 5919 5920 5921 5922 5923 5924 5925 5926 5927 5928 5929 5930
5931 5932 5933 5934 5935 5936 5937 5938 5939 5940 5941 5942 5943 5944 5945 5946
5947 5948 5949 5950 5951 5952 5953 5954 5955 5956 5957 5958 5959 5960 5961 5962
5963 5964 5965 5966 5967 5968 5969 5970 5971 5972 5973 5974 5975 5976 5977 5978
5979 5980 5981 5982 5983 5984 5985 5986 5987 5988 5989 5990 5991 5992 5993 5994
5995 5996 5997 5998 5999 6000]ans151j=[[EMPLOYEENAME, SALARY, EMPLOYEENAME,
SALARY, 3001, 3002, 3003, 3004, 3005, 3006, 3007, 3008, 3009, 3010, 3011, 3012, 3013,
3014, 3015, 3016, 3017, 3018, 3019, 3020, 3021, 3022, 3023, 3024, 3025, 3026, 3027,
3028, 3029, 3030, 3031, 3032, 3033, 3034, 3035, 3036, 3037, 3038, 3039, 3040, 3041,
3042, 3043, 3044, 3045, 3046, 3047, 3048, 3049, 3050, 3051, 3052, 3053, 3054, 3055,
3056, 3057, 3058, 3059, 3060, 3061, 3062, 3063, 3064, 3065, 3066, 3067, 3068, 3069,
3070, 3071, 3072, 3073, 3074, 3075, 3076, 3077, 3078, 3079, 3080, 3081, 3082, 3083,
3084, 3085, 3086, 3087, 3088, 3089, 3090, 3091, 3092, 3093, 3094, 3095, 3096, 3097,
3098, 3099, 3100, 3101, 3102, 3103, 3104, 3105, 3106, 3107, 3108, 3109, 3110, 3111,
3112, 3113, 3114, 3115, 3116, 3117, 3118, 3119, 3120, 3121, 3122, 3123, 3124, 3125,
3126, 3127, 3128, 3129, 3130, 3131, 3132, 3133, 3134, 3135, 3136, 3137, 3138, 3139,
3140, 3141, 3142, 3143, 3144, 3145, 3146, 3147, 3148, 3149, 3150, 3151, 3152, 3153,
3154, 3155, 3156, 3157, 3158, 3159, 3160, 3161, 3162, 3163, 3164, 3165, 3166, 3167,
3168, 3169, 3170, 3171, 3172, 3173, 3174, 3175, 3176, 3177, 3178, 3179, 3180, 3181,
3182, 3183, 3184, 3185, 3186, 3187, 3188, 3189, 3190, 3191, 3192, 3193, 3194, 3195,
3196, 3197, 3198, 3199, 3200, 3201, 3202, 3203, 3204, 3205, 3206, 3207, 3208, 3209,
3210, 3211, 3212, 3213, 3214, 3215, 3216, 3217, 3218, 3219, 3220, 3221, 3222, 3223,
3224, 3225, 3226, 3227, 3228, 3229, 3230, 3231, 3232, 3233, 3234, 3235, 3236, 3237,
3238, 3239, 3240, 3241, 3242, 3243, 3244, 3245, 3246, 3247, 3248, 3249, 3250, 3251,
3252, 3253, 3254, 3255, 3256, 3257, 3258, 3259, 3260, 3261, 3262, 3263, 3264, 3265,
3266, 3267, 3268, 3269, 3270, 3271, 3272, 3273, 3274, 3275, 3276, 3277, 3278, 3279,
3280, 3281, 3282, 3283, 3284, 3285, 3286, 3287, 3288, 3289, 3290, 3291, 3292, 3293,
3294, 3295, 3296, 3297, 3298, 3299, 3300, 3301, 3302, 3303, 3304, 3305, 3306, 3307,

3308, 3309, 3310, 3311, 3312, 3313, 3314, 3315, 3316, 3317, 3318, 3319, 3320, 3321,
3322, 3323, 3324, 3325, 3326, 3327, 3328, 3329, 3330, 3331, 3332, 3333, 3334, 3335,
3336, 3337, 3338, 3339, 3340, 3341, 3342, 3343, 3344, 3345, 3346, 3347, 3348, 3349,
3350, 3351, 3352, 3353, 3354, 3355, 3356, 3357, 3358, 3359, 3360, 3361, 3362, 3363,
3364, 3365, 3366, 3367, 3368, 3369, 3370, 3371, 3372, 3373, 3374, 3375, 3376, 3377,
3378, 3379, 3380, 3381, 3382, 3383, 3384, 3385, 3386, 3387, 3388, 3389, 3390, 3391,
3392, 3393, 3394, 3395, 3396, 3397, 3398, 3399, 3400, 3401, 3402, 3403, 3404, 3405,
3406, 3407, 3408, 3409, 3410, 3411, 3412, 3413, 3414, 3415, 3416, 3417, 3418, 3419,
3420, 3421, 3422, 3423, 3424, 3425, 3426, 3427, 3428, 3429, 3430, 3431, 3432, 3433,
3434, 3435, 3436, 3437, 3438, 3439, 3440, 3441, 3442, 3443, 3444, 3445, 3446, 3447,
3448, 3449, 3450, 3451, 3452, 3453, 3454, 3455, 3456, 3457, 3458, 3459, 3460, 3461,
3462, 3463, 3464, 3465, 3466, 3467, 3468, 3469, 3470, 3471, 3472, 3473, 3474, 3475,
3476, 3477, 3478, 3479, 3480, 3481, 3482, 3483, 3484, 3485, 3486, 3487, 3488, 3489,
3490, 3491, 3492, 3493, 3494, 3495, 3496, 3497, 3498, 3499, 3500, 3501, 3502, 3503,
3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511, 3512, 3513, 3514, 3515, 3516, 3517,
3518, 3519, 3520, 3521, 3522, 3523, 3524, 3525, 3526, 3527, 3528, 3529, 3530, 3531,
3532, 3533, 3534, 3535, 3536, 3537, 3538, 3539, 3540, 3541, 3542, 3543, 3544, 3545,
3546, 3547, 3548, 3549, 3550, 3551, 3552, 3553, 3554, 3555, 3556, 3557, 3558, 3559,
3560, 3561, 3562, 3563, 3564, 3565, 3566, 3567, 3568, 3569, 3570, 3571, 3572, 3573,
3574, 3575, 3576, 3577, 3578, 3579, 3580, 3581, 3582, 3583, 3584, 3585, 3586, 3587,
3588, 3589, 3590, 3591, 3592, 3593, 3594, 3595, 3596, 3597, 3598, 3599, 3600, 3601,
3602, 3603, 3604, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613, 3614, 3615,
3616, 3617, 3618, 3619, 3620, 3621, 3622, 3623, 3624, 3625, 3626, 3627, 3628, 3629,
3630, 3631, 3632, 3633, 3634, 3635, 3636, 3637, 3638, 3639, 3640, 3641, 3642, 3643,
3644, 3645, 3646, 3647, 3648, 3649, 3650, 3651, 3652, 3653, 3654, 3655, 3656, 3657,
3658, 3659, 3660, 3661, 3662, 3663, 3664, 3665, 3666, 3667, 3668, 3669, 3670, 3671,
3672, 3673, 3674, 3675, 3676, 3677, 3678, 3679, 3680, 3681, 3682, 3683, 3684, 3685,
3686, 3687, 3688, 3689, 3690, 3691, 3692, 3693, 3694, 3695, 3696, 3697, 3698, 3699,
3700, 3701, 3702, 3703, 3704, 3705, 3706, 3707, 3708, 3709, 3710, 3711, 3712, 3713,
3714, 3715, 3716, 3717, 3718, 3719, 3720, 3721, 3722, 3723, 3724, 3725, 3726, 3727,
3728, 3729, 3730, 3731, 3732, 3733, 3734, 3735, 3736, 3737, 3738, 3739, 3740, 3741,
3742, 3743, 3744, 3745, 3746, 3747, 3748, 3749, 3750, 3751, 3752, 3753, 3754, 3755,
3756, 3757, 3758, 3759, 3760, 3761, 3762, 3763, 3764, 3765, 3766, 3767, 3768, 3769,
3770, 3771, 3772, 3773, 3774, 3775, 3776, 3777, 3778, 3779, 3780, 3781, 3782, 3783,
3784, 3785, 3786, 3787, 3788, 3789, 3790, 3791, 3792, 3793, 3794, 3795, 3796, 3797,
3798, 3799, 3800, 3801, 3802, 3803, 3804, 3805, 3806, 3807, 3808, 3809, 3810, 3811,
3812, 3813, 3814, 3815, 3816, 3817, 3818, 3819, 3820, 3821, 3822, 3823, 3824, 3825,
3826, 3827, 3828, 3829, 3830, 3831, 3832, 3833, 3834, 3835, 3836, 3837, 3838, 3839,
3840, 3841, 3842, 3843, 3844, 3845, 3846, 3847, 3848, 3849, 3850, 3851, 3852, 3853,
3854, 3855, 3856, 3857, 3858, 3859, 3860, 3861, 3862, 3863, 3864, 3865, 3866, 3867,
3868, 3869, 3870, 3871, 3872, 3873, 3874, 3875, 3876, 3877, 3878, 3879, 3880, 3881,
3882, 3883, 3884, 3885, 3886, 3887, 3888, 3889, 3890, 3891, 3892, 3893, 3894, 3895,

3896, 3897, 3898, 3899, 3900, 3901, 3902, 3903, 3904, 3905, 3906, 3907, 3908, 3909,
3910, 3911, 3912, 3913, 3914, 3915, 3916, 3917, 3918, 3919, 3920, 3921, 3922, 3923,
3924, 3925, 3926, 3927, 3928, 3929, 3930, 3931, 3932, 3933, 3934, 3935, 3936, 3937,
3938, 3939, 3940, 3941, 3942, 3943, 3944, 3945, 3946, 3947, 3948, 3949, 3950, 3951,
3952, 3953, 3954, 3955, 3956, 3957, 3958, 3959, 3960, 3961, 3962, 3963, 3964, 3965,
3966, 3967, 3968, 3969, 3970, 3971, 3972, 3973, 3974, 3975, 3976, 3977, 3978, 3979,
3980, 3981, 3982, 3983, 3984, 3985, 3986, 3987, 3988, 3989, 3990, 3991, 3992, 3993,
3994, 3995, 3996, 3997, 3998, 3999, 4000, 4001, 4002, 4003, 4004, 4005, 4006, 4007,
4008, 4009, 4010, 4011, 4012, 4013, 4014, 4015, 4016, 4017, 4018, 4019, 4020, 4021,
4022, 4023, 4024, 4025, 4026, 4027, 4028, 4029, 4030, 4031, 4032, 4033, 4034, 4035,
4036, 4037, 4038, 4039, 4040, 4041, 4042, 4043, 4044, 4045, 4046, 4047, 4048, 4049,
4050, 4051, 4052, 4053, 4054, 4055, 4056, 4057, 4058, 4059, 4060, 4061, 4062, 4063,
4064, 4065, 4066, 4067, 4068, 4069, 4070, 4071, 4072, 4073, 4074, 4075, 4076, 4077,
4078, 4079, 4080, 4081, 4082, 4083, 4084, 4085, 4086, 4087, 4088, 4089, 4090, 4091,
4092, 4093, 4094, 4095, 4096, 4097, 4098, 4099, 4100, 4101, 4102, 4103, 4104, 4105,
4106, 4107, 4108, 4109, 4110, 4111, 4112, 4113, 4114, 4115, 4116, 4117, 4118, 4119,
4120, 4121, 4122, 4123, 4124, 4125, 4126, 4127, 4128, 4129, 4130, 4131, 4132, 4133,
4134, 4135, 4136, 4137, 4138, 4139, 4140, 4141, 4142, 4143, 4144, 4145, 4146, 4147,
4148, 4149, 4150, 4151, 4152, 4153, 4154, 4155, 4156, 4157, 4158, 4159, 4160, 4161,
4162, 4163, 4164, 4165, 4166, 4167, 4168, 4169, 4170, 4171, 4172, 4173, 4174, 4175,
4176, 4177, 4178, 4179, 4180, 4181, 4182, 4183, 4184, 4185, 4186, 4187, 4188, 4189,
4190, 4191, 4192, 4193, 4194, 4195, 4196, 4197, 4198, 4199, 4200, 4201, 4202, 4203,
4204, 4205, 4206, 4207, 4208, 4209, 4210, 4211, 4212, 4213, 4214, 4215, 4216, 4217,
4218, 4219, 4220, 4221, 4222, 4223, 4224, 4225, 4226, 4227, 4228, 4229, 4230, 4231,
4232, 4233, 4234, 4235, 4236, 4237, 4238, 4239, 4240, 4241, 4242, 4243, 4244, 4245,
4246, 4247, 4248, 4249, 4250, 4251, 4252, 4253, 4254, 4255, 4256, 4257, 4258, 4259,
4260, 4261, 4262, 4263, 4264, 4265, 4266, 4267, 4268, 4269, 4270, 4271, 4272, 4273,
4274, 4275, 4276, 4277, 4278, 4279, 4280, 4281, 4282, 4283, 4284, 4285, 4286, 4287,
4288, 4289, 4290, 4291, 4292, 4293, 4294, 4295, 4296, 4297, 4298, 4299, 4300, 4301,
4302, 4303, 4304, 4305, 4306, 4307, 4308, 4309, 4310, 4311, 4312, 4313, 4314, 4315,
4316, 4317, 4318, 4319, 4320, 4321, 4322, 4323, 4324, 4325, 4326, 4327, 4328, 4329,
4330, 4331, 4332, 4333, 4334, 4335, 4336, 4337, 4338, 4339, 4340, 4341, 4342, 4343,
4344, 4345, 4346, 4347, 4348, 4349, 4350, 4351, 4352, 4353, 4354, 4355, 4356, 4357,
4358, 4359, 4360, 4361, 4362, 4363, 4364, 4365, 4366, 4367, 4368, 4369, 4370, 4371,
4372, 4373, 4374, 4375, 4376, 4377, 4378, 4379, 4380, 4381, 4382, 4383, 4384, 4385,
4386, 4387, 4388, 4389, 4390, 4391, 4392, 4393, 4394, 4395, 4396, 4397, 4398, 4399,
4400, 4401, 4402, 4403, 4404, 4405, 4406, 4407, 4408, 4409, 4410, 4411, 4412, 4413,
4414, 4415, 4416, 4417, 4418, 4419, 4420, 4421, 4422, 4423, 4424, 4425, 4426, 4427,
4428, 4429, 4430, 4431, 4432, 4433, 4434, 4435, 4436, 4437, 4438, 4439, 4440, 4441,
4442, 4443, 4444, 4445, 4446, 4447, 4448, 4449, 4450, 4451, 4452, 4453, 4454, 4455,
4456, 4457, 4458, 4459, 4460, 4461, 4462, 4463, 4464, 4465, 4466, 4467, 4468, 4469,
4470, 4471, 4472, 4473, 4474, 4475, 4476, 4477, 4478, 4479, 4480, 4481, 4482, 4483,

4484, 4485, 4486, 4487, 4488, 4489, 4490, 4491, 4492, 4493, 4494, 4495, 4496, 4497,
4498, 4499, 4500, 4501, 4502, 4503, 4504, 4505, 4506, 4507, 4508, 4509, 4510, 4511,
4512, 4513, 4514, 4515, 4516, 4517, 4518, 4519, 4520, 4521, 4522, 4523, 4524, 4525,
4526, 4527, 4528, 4529, 4530, 4531, 4532, 4533, 4534, 4535, 4536, 4537, 4538, 4539,
4540, 4541, 4542, 4543, 4544, 4545, 4546, 4547, 4548, 4549, 4550, 4551, 4552, 4553,
4554, 4555, 4556, 4557, 4558, 4559, 4560, 4561, 4562, 4563, 4564, 4565, 4566, 4567,
4568, 4569, 4570, 4571, 4572, 4573, 4574, 4575, 4576, 4577, 4578, 4579, 4580, 4581,
4582, 4583, 4584, 4585, 4586, 4587, 4588, 4589, 4590, 4591, 4592, 4593, 4594, 4595,
4596, 4597, 4598, 4599, 4600, 4601, 4602, 4603, 4604, 4605, 4606, 4607, 4608, 4609,
4610, 4611, 4612, 4613, 4614, 4615, 4616, 4617, 4618, 4619, 4620, 4621, 4622, 4623,
4624, 4625, 4626, 4627, 4628, 4629, 4630, 4631, 4632, 4633, 4634, 4635, 4636, 4637,
4638, 4639, 4640, 4641, 4642, 4643, 4644, 4645, 4646, 4647, 4648, 4649, 4650, 4651,
4652, 4653, 4654, 4655, 4656, 4657, 4658, 4659, 4660, 4661, 4662, 4663, 4664, 4665,
4666, 4667, 4668, 4669, 4670, 4671, 4672, 4673, 4674, 4675, 4676, 4677, 4678, 4679,
4680, 4681, 4682, 4683, 4684, 4685, 4686, 4687, 4688, 4689, 4690, 4691, 4692, 4693,
4694, 4695, 4696, 4697, 4698, 4699, 4700, 4701, 4702, 4703, 4704, 4705, 4706, 4707,
4708, 4709, 4710, 4711, 4712, 4713, 4714, 4715, 4716, 4717, 4718, 4719, 4720, 4721,
4722, 4723, 4724, 4725, 4726, 4727, 4728, 4729, 4730, 4731, 4732, 4733, 4734, 4735,
4736, 4737, 4738, 4739, 4740, 4741, 4742, 4743, 4744, 4745, 4746, 4747, 4748, 4749,
4750, 4751, 4752, 4753, 4754, 4755, 4756, 4757, 4758, 4759, 4760, 4761, 4762, 4763,
4764, 4765, 4766, 4767, 4768, 4769, 4770, 4771, 4772, 4773, 4774, 4775, 4776, 4777,
4778, 4779, 4780, 4781, 4782, 4783, 4784, 4785, 4786, 4787, 4788, 4789, 4790, 4791,
4792, 4793, 4794, 4795, 4796, 4797, 4798, 4799, 4800, 4801, 4802, 4803, 4804, 4805,
4806, 4807, 4808, 4809, 4810, 4811, 4812, 4813, 4814, 4815, 4816, 4817, 4818, 4819,
4820, 4821, 4822, 4823, 4824, 4825, 4826, 4827, 4828, 4829, 4830, 4831, 4832, 4833,
4834, 4835, 4836, 4837, 4838, 4839, 4840, 4841, 4842, 4843, 4844, 4845, 4846, 4847,
4848, 4849, 4850, 4851, 4852, 4853, 4854, 4855, 4856, 4857, 4858, 4859, 4860, 4861,
4862, 4863, 4864, 4865, 4866, 4867, 4868, 4869, 4870, 4871, 4872, 4873, 4874, 4875,
4876, 4877, 4878, 4879, 4880, 4881, 4882, 4883, 4884, 4885, 4886, 4887, 4888, 4889,
4890, 4891, 4892, 4893, 4894, 4895, 4896, 4897, 4898, 4899, 4900, 4901, 4902, 4903,
4904, 4905, 4906, 4907, 4908, 4909, 4910, 4911, 4912, 4913, 4914, 4915, 4916, 4917,
4918, 4919, 4920, 4921, 4922, 4923, 4924, 4925, 4926, 4927, 4928, 4929, 4930, 4931,
4932, 4933, 4934, 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943, 4944, 4945,
4946, 4947, 4948, 4949, 4950, 4951, 4952, 4953, 4954, 4955, 4956, 4957, 4958, 4959,
4960, 4961, 4962, 4963, 4964, 4965, 4966, 4967, 4968, 4969, 4970, 4971, 4972, 4973,
4974, 4975, 4976, 4977, 4978, 4979, 4980, 4981, 4982, 4983, 4984, 4985, 4986, 4987,
4988, 4989, 4990, 4991, 4992, 4993, 4994, 4995, 4996, 4997, 4998, 4999, 5000, 5001,
5002, 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011, 5012, 5013, 5014, 5015,
5016, 5017, 5018, 5019, 5020, 5021, 5022, 5023, 5024, 5025, 5026, 5027, 5028, 5029,
5030, 5031, 5032, 5033, 5034, 5035, 5036, 5037, 5038, 5039, 5040, 5041, 5042, 5043,
5044, 5045, 5046, 5047, 5048, 5049, 5050, 5051, 5052, 5053, 5054, 5055, 5056, 5057,
5058, 5059, 5060, 5061, 5062, 5063, 5064, 5065, 5066, 5067, 5068, 5069, 5070, 5071,

5072, 5073, 5074, 5075, 5076, 5077, 5078, 5079, 5080, 5081, 5082, 5083, 5084, 5085,
5086, 5087, 5088, 5089, 5090, 5091, 5092, 5093, 5094, 5095, 5096, 5097, 5098, 5099,
5100, 5101, 5102, 5103, 5104, 5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113,
5114, 5115, 5116, 5117, 5118, 5119, 5120, 5121, 5122, 5123, 5124, 5125, 5126, 5127,
5128, 5129, 5130, 5131, 5132, 5133, 5134, 5135, 5136, 5137, 5138, 5139, 5140, 5141,
5142, 5143, 5144, 5145, 5146, 5147, 5148, 5149, 5150, 5151, 5152, 5153, 5154, 5155,
5156, 5157, 5158, 5159, 5160, 5161, 5162, 5163, 5164, 5165, 5166, 5167, 5168, 5169,
5170, 5171, 5172, 5173, 5174, 5175, 5176, 5177, 5178, 5179, 5180, 5181, 5182, 5183,
5184, 5185, 5186, 5187, 5188, 5189, 5190, 5191, 5192, 5193, 5194, 5195, 5196, 5197,
5198, 5199, 5200, 5201, 5202, 5203, 5204, 5205, 5206, 5207, 5208, 5209, 5210, 5211,
5212, 5213, 5214, 5215, 5216, 5217, 5218, 5219, 5220, 5221, 5222, 5223, 5224, 5225,
5226, 5227, 5228, 5229, 5230, 5231, 5232, 5233, 5234, 5235, 5236, 5237, 5238, 5239,
5240, 5241, 5242, 5243, 5244, 5245, 5246, 5247, 5248, 5249, 5250, 5251, 5252, 5253,
5254, 5255, 5256, 5257, 5258, 5259, 5260, 5261, 5262, 5263, 5264, 5265, 5266, 5267,
5268, 5269, 5270, 5271, 5272, 5273, 5274, 5275, 5276, 5277, 5278, 5279, 5280, 5281,
5282, 5283, 5284, 5285, 5286, 5287, 5288, 5289, 5290, 5291, 5292, 5293, 5294, 5295,
5296, 5297, 5298, 5299, 5300, 5301, 5302, 5303, 5304, 5305, 5306, 5307, 5308, 5309,
5310, 5311, 5312, 5313, 5314, 5315, 5316, 5317, 5318, 5319, 5320, 5321, 5322, 5323,
5324, 5325, 5326, 5327, 5328, 5329, 5330, 5331, 5332, 5333, 5334, 5335, 5336, 5337,
5338, 5339, 5340, 5341, 5342, 5343, 5344, 5345, 5346, 5347, 5348, 5349, 5350, 5351,
5352, 5353, 5354, 5355, 5356, 5357, 5358, 5359, 5360, 5361, 5362, 5363, 5364, 5365,
5366, 5367, 5368, 5369, 5370, 5371, 5372, 5373, 5374, 5375, 5376, 5377, 5378, 5379,
5380, 5381, 5382, 5383, 5384, 5385, 5386, 5387, 5388, 5389, 5390, 5391, 5392, 5393,
5394, 5395, 5396, 5397, 5398, 5399, 5400, 5401, 5402, 5403, 5404, 5405, 5406, 5407,
5408, 5409, 5410, 5411, 5412, 5413, 5414, 5415, 5416, 5417, 5418, 5419, 5420, 5421,
5422, 5423, 5424, 5425, 5426, 5427, 5428, 5429, 5430, 5431, 5432, 5433, 5434, 5435,
5436, 5437, 5438, 5439, 5440, 5441, 5442, 5443, 5444, 5445, 5446, 5447, 5448, 5449,
5450, 5451, 5452, 5453, 5454, 5455, 5456, 5457, 5458, 5459, 5460, 5461, 5462, 5463,
5464, 5465, 5466, 5467, 5468, 5469, 5470, 5471, 5472, 5473, 5474, 5475, 5476, 5477,
5478, 5479, 5480, 5481, 5482, 5483, 5484, 5485, 5486, 5487, 5488, 5489, 5490, 5491,
5492, 5493, 5494, 5495, 5496, 5497, 5498, 5499, 5500, 5501, 5502, 5503, 5504, 5505,
5506, 5507, 5508, 5509, 5510, 5511, 5512, 5513, 5514, 5515, 5516, 5517, 5518, 5519,
5520, 5521, 5522, 5523, 5524, 5525, 5526, 5527, 5528, 5529, 5530, 5531, 5532, 5533,
5534, 5535, 5536, 5537, 5538, 5539, 5540, 5541, 5542, 5543, 5544, 5545, 5546, 5547,
5548, 5549, 5550, 5551, 5552, 5553, 5554, 5555, 5556, 5557, 5558, 5559, 5560, 5561,
5562, 5563, 5564, 5565, 5566, 5567, 5568, 5569, 5570, 5571, 5572, 5573, 5574, 5575,
5576, 5577, 5578, 5579, 5580, 5581, 5582, 5583, 5584, 5585, 5586, 5587, 5588, 5589,
5590, 5591, 5592, 5593, 5594, 5595, 5596, 5597, 5598, 5599, 5600, 5601, 5602, 5603,
5604, 5605, 5606, 5607, 5608, 5609, 5610, 5611, 5612, 5613, 5614, 5615, 5616, 5617,
5618, 5619, 5620, 5621, 5622, 5623, 5624, 5625, 5626, 5627, 5628, 5629, 5630, 5631,
5632, 5633, 5634, 5635, 5636, 5637, 5638, 5639, 5640, 5641, 5642, 5643, 5644, 5645,
5646, 5647, 5648, 5649, 5650, 5651, 5652, 5653, 5654, 5655, 5656, 5657, 5658, 5659,

5660, 5661, 5662, 5663, 5664, 5665, 5666, 5667, 5668, 5669, 5670, 5671, 5672, 5673, 5674, 5675, 5676, 5677, 5678, 5679, 5680, 5681, 5682, 5683, 5684, 5685, 5686, 5687, 5688, 5689, 5690, 5691, 5692, 5693, 5694, 5695, 5696, 5697, 5698, 5699, 5700, 5701, 5702, 5703, 5704, 5705, 5706, 5707, 5708, 5709, 5710, 5711, 5712, 5713, 5714, 5715, 5716, 5717, 5718, 5719, 5720, 5721, 5722, 5723, 5724, 5725, 5726, 5727, 5728, 5729, 5730, 5731, 5732, 5733, 5734, 5735, 5736, 5737, 5738, 5739, 5740, 5741, 5742, 5743, 5744, 5745, 5746, 5747, 5748, 5749, 5750, 5751, 5752, 5753, 5754, 5755, 5756, 5757, 5758, 5759, 5760, 5761, 5762, 5763, 5764, 5765, 5766, 5767, 5768, 5769, 5770, 5771, 5772, 5773, 5774, 5775, 5776, 5777, 5778, 5779, 5780, 5781, 5782, 5783, 5784, 5785, 5786, 5787, 5788, 5789, 5790, 5791, 5792, 5793, 5794, 5795, 5796, 5797, 5798, 5799, 5800, 5801, 5802, 5803, 5804, 5805, 5806, 5807, 5808, 5809, 5810, 5811, 5812, 5813, 5814, 5815, 5816, 5817, 5818, 5819, 5820, 5821, 5822, 5823, 5824, 5825, 5826, 5827, 5828, 5829, 5830, 5831, 5832, 5833, 5834, 5835, 5836, 5837, 5838, 5839, 5840, 5841, 5842, 5843, 5844, 5845, 5846, 5847, 5848, 5849, 5850, 5851, 5852, 5853, 5854, 5855, 5856, 5857, 5858, 5859, 5860, 5861, 5862, 5863, 5864, 5865, 5866, 5867, 5868, 5869, 5870, 5871, 5872, 5873, 5874, 5875, 5876, 5877, 5878, 5879, 5880, 5881, 5882, 5883, 5884, 5885, 5886, 5887, 5888, 5889, 5890, 5891, 5892, 5893, 5894, 5895, 5896, 5897, 5898, 5899, 5900, 5901, 5902, 5903, 5904, 5905, 5906, 5907, 5908, 5909, 5910, 5911, 5912, 5913, 5914, 5915, 5916, 5917, 5918, 5919, 5920, 5921, 5922, 5923, 5924, 5925, 5926, 5927, 5928, 5929, 5930, 5931, 5932, 5933, 5934, 5935, 5936, 5937, 5938, 5939, 5940, 5941, 5942, 5943, 5944, 5945, 5946, 5947, 5948, 5949, 5950, 5951, 5952, 5953, 5954, 5955, 5956, 5957, 5958, 5959, 5960, 5961, 5962, 5963, 5964, 5965, 5966, 5967, 5968, 5969, 5970, 5971, 5972, 5973, 5974, 5975, 5976, 5977, 5978, 5979, 5980, 5981, 5982, 5983, 5984, 5985, 5986, 5987, 5988, 5989, 5990, 5991, 5992, 5993, 5994, 5995, 5996, 5997, 5998, 5999, 6000]]

What is the Major Advantage of WNoSQL(DB*)?

It will store and retrieve infinitive no of data in WNOSQL DB.

It is used for storing higher volume of data when compared to other databases.

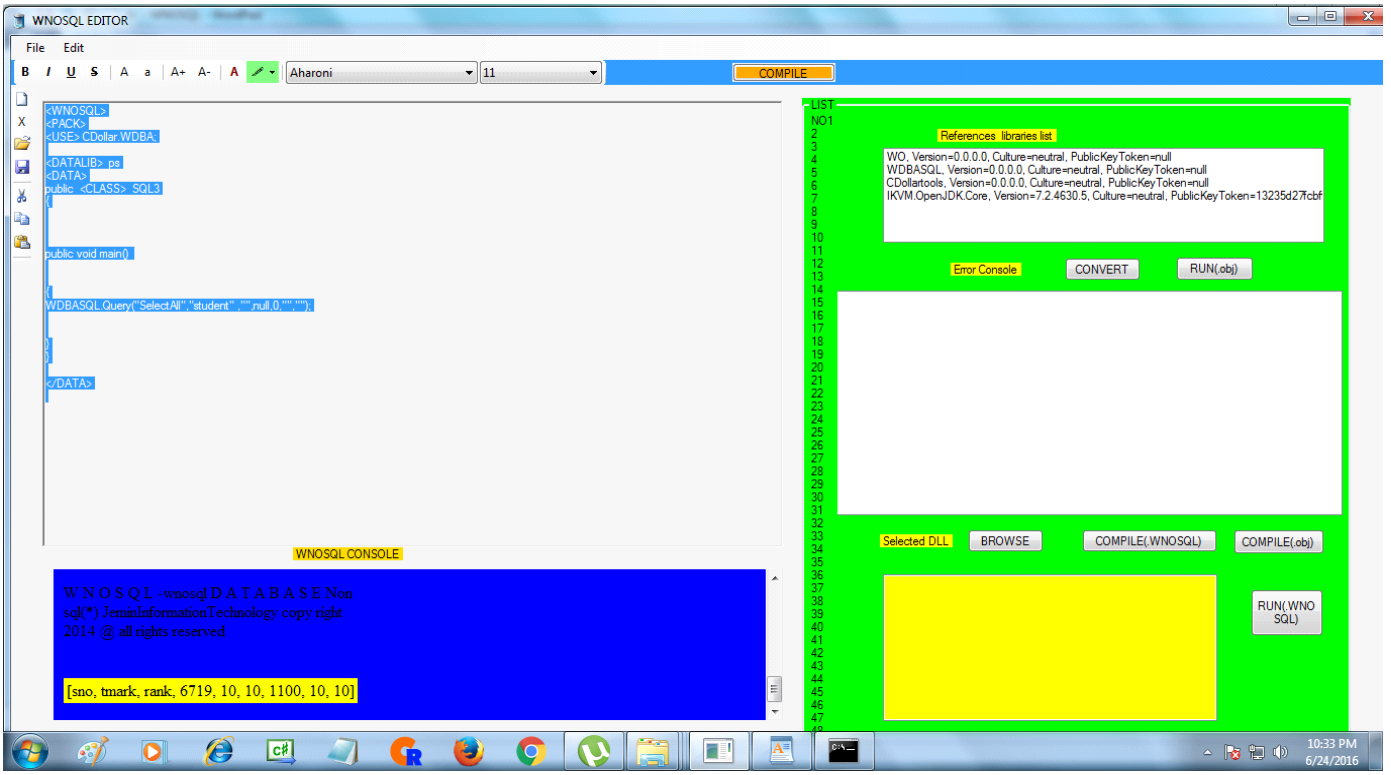
UNIT:9: WNOSQL(WSQL*) TEST Exercises

1) Write a WNOSQL Program to select all students

from a student table?

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA; //load wnosql contents.


<DATALIB> ps
<DATA>
public <CLASS> SQL3
{



public void main()


{


String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");
```

```
String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
WDBASQL.Query("SelectAll","student" ,"0",null,15,"","",, null,"",0,"
","",c,null,t,1,1);




}
}

</DATA>
```



2) Write a WNOSQL program to Insert

values from sql database to wnosql database and transport to sqlserver database?

A) Do it your own by using WNOSQL fundemantal syntax.

3) Write a WNOSQL program to

>select a particular column value from a table

>select a particular column value from a table

by a given range.

>to compare dates and sort in ascending order.

> to Search a data from a given table range.

> Compute avg,max,minum marks from classtable

> Search data which is greater than a given data

> Search data which is lesser than a given data

> to count total no of rows in atable

> To store a serialized data in WDBA file

and retrieve and print it.

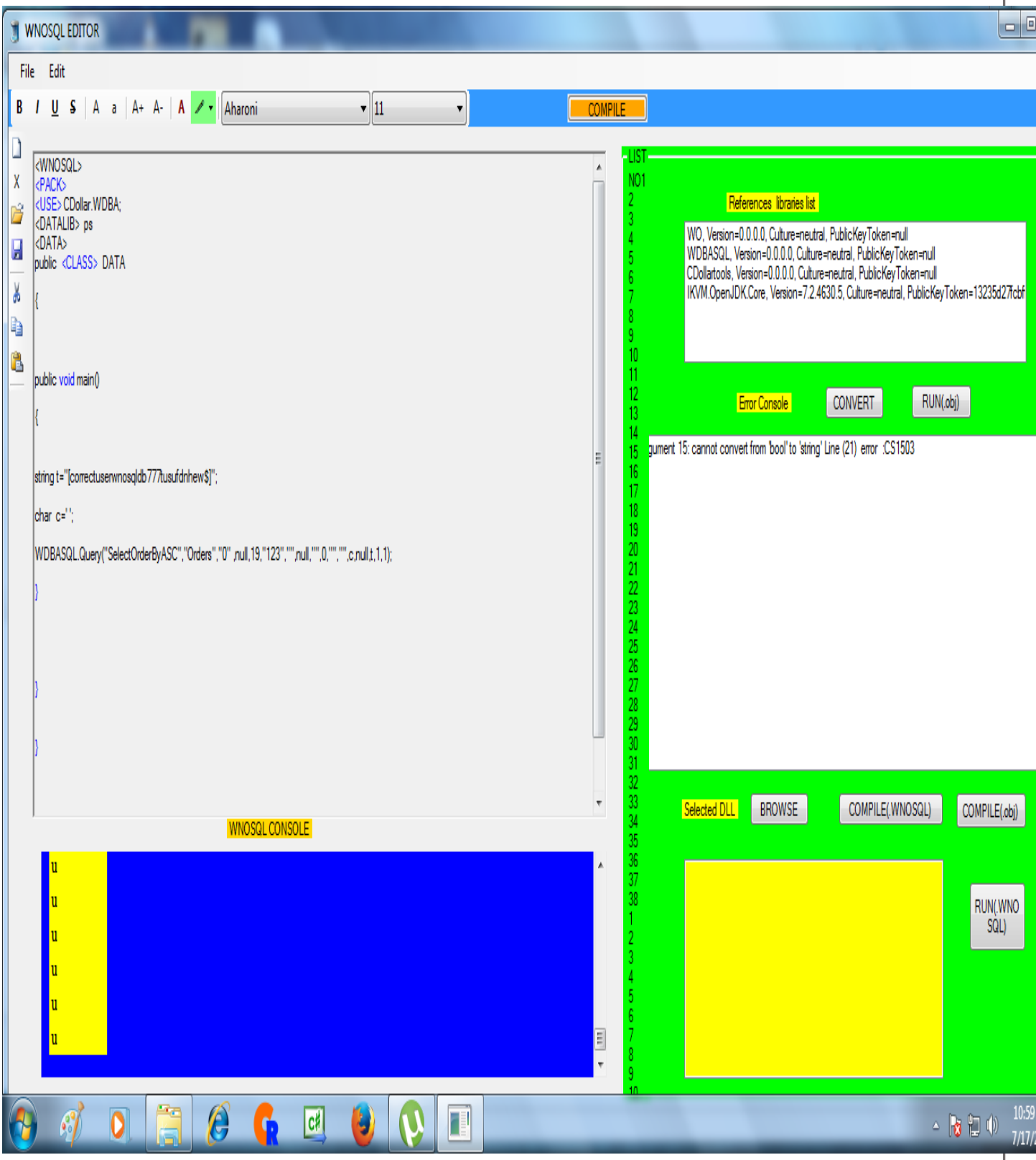
> Delete all the contents of table and drop the table.

4) Test using Your Dotnet or JAVA Program with WNOSQL database

5) Apply SelectOrderByASC to the PLSQL to table Orders for 0 to 19 records what happens?

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA; //load CDollar.WDBA libraries
<DATALIB> ps // namespace ps
<DATA>
public <CLASS> DATA

{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char c=' ';

WDBASQL.Query("SelectOrderByASC","Orders","0"
,null,19,"123","",null,"",0,"","",c,null,t,1,1);

}
```

```
}

}
```


WNOSQL EDITOR
File Edit
Aharoni
11
COMPILE
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA
{
public void main()
{
string t="[correctuserwnosqldb777tusufdnhew$]";
char c=' ';
WDBASQL.Query("SelectOrderByASC","Orders","0",null,19,"123","",null,"",0,"","",c,null,t,1,1);
}
}
}
WNOSQL CONSOLE
u
u
u
u
u
u
LIST
References libraries list
WO, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
WDBASQL, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
CDollartools, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
IKVM.OpenJDK.Core, Version=7.2.4630.5, Culture=neutral, PublicKeyToken=13235d27fcbf
Error Console
CONVERT
RUN(.obj)
gument 15: cannot convert from 'bool' to 'string' Line (21) error :CS1503
Selected DLL
BROWSE
COMPILE(.WNOSQL)
COMPILE(.obj)
RUN(.WNOSQL)

WNOSQL(*) PLSQL follows basic WNOSQL(*) api syntax WNOSQL(*) put all the results in arraylist for future use.

Program-1:

Write a program to display the matching data rows and perform innerjoins between two table.

```
<WNOSQL> // Beginning of wnosql plsql program
<PACK> //load all wnosql packages
<USE> CDollar.WDBA; //use CDollar.WDBA packages
<DATALIB> ps // create name space ps
<DATA> //write wnosql logic
public <CLASS> DATA

{


public void main() //like C main

{
```

```
//kindly  refer  wnosql  fundemantals


String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");


String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);


char  c=' ';


ArrayList arhd1gy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0001","
",null,"",0,"1","",c,null,t,1,1);


ArrayList arhd1gy1 =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0002","
",null,"",0,"1","",c,null,t,1,1);


ArrayList arhd1gyy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0003","
",null,"",0,"1","",c,null,t,1,1);


ArrayList arhd1gy1y =WDBASQL.Query("MATCH","Orders"
,"0",null,19,"0005"," ",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList  artr11= new  ArrayList();

for(int  i=0;i<arhd1gy.size();i++)
artr11.add(arhd1gy.get(i));

for(int  i=0;i<arhd1gy1.size();i++)
artr11.add(arhd1gy1.get(i));

for(int  i=0;i<arhd1gyy.size();i++)
artr11.add(arhd1gyy.get(i));

for(int  i=0;i<arhd1gy1y.size();i++)
artr11.add(arhd1gy1y.get(i));
```

```
ArrayList arhd1gy17 =WDBASQL.Query("MATCH","employess"
,"0",null,11,"0001"," ",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy117 =WDBASQL.Query("MATCH","employess"
,"0",null,11,"0002"," ",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy178 =WDBASQL.Query("MATCH","employess"
,"0",null,13,"0003"," ",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList arhd1gy1178 =WDBASQL.Query("MATCH","employess"
,"0",null,13,"0005"," ",null,"",0,"1","",c,null,t,1,1);



ArrayList  artr117= new  ArrayList();

for(int  i=0;i<arhd1gy17.size();i++)
artr117.add(arhd1gy17.get(i));

for(int  i=0;i<arhd1gy117.size();i++)
artr117.add(arhd1gy117.get(i));
for(int  i=0;i<arhd1gy178.size();i++)
artr117.add(arhd1gy178.get(i));

for(int  i=0;i<arhd1gy1178.size();i++)
artr117.add(arhd1gy1178.get(i));

ArrayList
datas1=WDBASQL.Query("INNERJOIN","Orders","0",null,19,"employess","",
artr11,"",0,"","",c,artr117,t,1,1);



}

}


}
```

Kindly use WNOSQL EDITOR to see the output.

Program-2: Write a Program to finding matching data rows and perform right join, use having clause , use innerjoin in this case:

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char c=' ';
```

```
ArrayList arhd1gy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0001","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy1 =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0002","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gyy =WDBASQL.Query("MATCH","Orders" ,"0",null,19,"0003","
",null,"",0,"1","",c,null,t,1,1);

ArrayList arhd1gy1y =WDBASQL.Query("MATCH","Orders"
,"0",null,19,"0005"," ",null,"",0,"1","",c,null,t,1,1);

ArrayList  artr11= new  ArrayList();

for(int  i=0;i<arhd1gy.size();i++)
artr11.add(arhd1gy.get(i));

for(int  i=0;i<arhd1gy1.size();i++)
artr11.add(arhd1gy1.get(i));

for(int  i=0;i<arhd1gyy.size();i++)
artr11.add(arhd1gyy.get(i));
```

```
for(int  i=0;i<arhd1gy1y.size();i++)
artr11.add(arhd1gy1y.get(i));
```

```
ArrayList arhd1gy17 =WDBASQL.Query("MATCH","employess"
,"0",null,11,"0001"," ",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList arhd1gy117 =WDBASQL.Query("MATCH","employess"
,"0",null,11,"0002"," ",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList arhd1gy178 =WDBASQL.Query("MATCH","employess"
,"0",null,13,"0003"," ",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList arhd1gy1178 =WDBASQL.Query("MATCH","employess"
,"0",null,13,"0005"," ",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList  artr117= new  ArrayList();
```

```
for(int  i=0;i<arhd1gy17.size();i++)
artr117.add(arhd1gy17.get(i));
```

```
for(int  i=0;i<arhd1gy117.size();i++)
artr117.add(arhd1gy117.get(i));
for(int  i=0;i<arhd1gy178.size();i++)
artr117.add(arhd1gy178.get(i));

for(int  i=0;i<arhd1gy1178.size();i++)
artr117.add(arhd1gy1178.get(i));


ArrayList  cols = new  ArrayList();

cols.add(0);
cols.add(1);
cols.add(2);
cols.add(3);
cols.add(4);
cols.add(5);
//cols.add(6);
//cols.add(7);
//cols.add(8);
//cols.add(9);
//cols.add(10);
//cols.add(11);

cols.add(0);
cols.add(1);
cols.add(2);


ArrayList  cols111 = new  ArrayList();
```

```
cols111.add(0);

cols111.add(1);



cols111.add(2);
cols111.add(3);
cols111.add(4);
cols111.add(5);
//cols111.add(6);
//cols111.add(7);
//cols111.add(8);
cols111.add(9);
cols111.add(10);
cols111.add(11);



ArrayList
datas44=WDBASQL.Query("RIGHTJOIN","Orders","0",null,0,"employess","",
cols,"",0,"","",c,cols111,t,1,1);


ArrayList  colss7 = new  ArrayList();


colss7.add(2);

ArrayList
datas16=WDBASQL.Query("<HAVING>","Orders","0",null,1,"[3,6,2,2],[2,5,2,2]
","",colss7,"",0,"","",c,datas44,t,1,1);
```

ArrayList
datas1=WDBASQL.Query("INNERJOIN","Orders","0",null,19,"employess","",
artr11,"",0,"","",c,artr117,t,1,1);

}

}

}

Program -3: Use Intorderby Ascending and descending order

and use Orderby ascending and descending order for the String datatype

table.

<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>

```
public  <CLASS>  DATA

{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char  c=' ';



WDBASQL.Query("SelectOrderByASC","Orders","0"
,null,19,"123","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("SelectOrderByDESC","Orders","0"
,null,19,"123","",null,"",0,"","",c,null,t,1,1);



WDBASQL.Query("SelectIntOrderByAsc","nos","0"
,null,4,"123","",null,"",0,"","",c,null,t,1,1);
```

WDBASQL.Query("SelectIntOrderByDesc","nos","0"
,null,4,"123","",null,"",0,"","",c,null,t,1,1);

}

}

}

Program 4:

==========

Write a WNOSQL Program to store the student query values in WDBA table

from SQLSERVER for the given fields sno,tmark,rank and store it in encrypted form

and again store the data in sqlserver for futhure use with C# program.

<WNOSQL>
<PACK>

<DATALIB> ps

```
<DATA>
public  <CLASS>  SQL3
{



public void main()



{

WDBA.writeln((manipulate.Signal("MANIPULATE","Select  *  from
student","student","sno,tmark,rank","?,?,?",4,"sun.jdbc.odbc.JdbcOdbcDriver","jd
bc:odbc:dsn","sa","jemin","wilmix21")));



}
}

</DATA>
```

Note : use manipulte .dll in this case..

Program5:

Write a program and use the following WNOSQL commands

and perform manipulation using

a) SELECT IN

b) SELECTLIKE

c) COUNT(*)
d) Encrypt

e) Decrypt

f) SelectAll

g) AVG(), MAX() , MIN() ,LOC(),SUM()

h) SelectCOLS, Count(),Distinct,MATCH

i) Insert
j) DatecompareAsc/DESC
k)InsertDesc ,AND ,Foreign Key

for WNOSQL TABLE

<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{

```
public void main()

{




String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char  c=' ';

ArrayList arhd111 =WDBASQL.Query("SelectIN","employess"
,"0",null,11,"0002","",null,"",0,"","",c,null,t,1,1);

ArrayList arhd112 =WDBASQL.Query("SelectNOTIN","employess"
,"0",null,11,"0002","",null,"",0,"","",c,null,t,1,1);




c='D';
WDBASQL.Query("SelectLike","Orders"
,"0",null,11,"","",null,"",0,"","",c,null,t,1,1);



WDBASQL.Query("Count(*)","Orders"
,"0",null,0,"","",null,"",0,"","",c,null,t,1,1);
```

```
WDBASQL.Query("MATH","nos"
,"0",null,0,"0","",null,"",0,"","acos",c,null,t,1,1);

WDBASQL.Query("Encrypt","nos" ,"0",null,0,"0","",null,"",0,"","",c,null,t,1,1);


WDBASQL.Query("Decrypt","nos" ,"0",null,0,"0","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("SelectAll","nos" ,"0",null,4,"0","",null,"",0,"","",c,null,t,1,1);


WDBASQL.Query("SelectAll","nos" ,"0",null,4,"4","",null,"",0,"","",c,null,t,1,1);


ArrayList  myList= new ArrayList();

myList.add("2005/01/12");

myList.add("2012/03/12");
myList.add("2006/03/12");
myList.add("2006/01/12");
myList.add("2005/11/12");

ArrayList  arms1d  = new  ArrayList();

arms1d.add(3);
arms1d.add(6);
arms1d.add(9);
arms1d.add(12);
arms1d.add(15);
arms1d.add(18);
```

```
ArrayList  sum55=WDBASQL.Query("AVG()","Orders"
,"0",null,6,"","",arms1d,"",0,"","",c,null,t,1,1);

ArrayList  sum55r=WDBASQL.Query("MAX()","Orders"
,"0",null,19,"","",arms1d,"",0,"","",c,null,t,1,1);
ArrayList  sum55gr=WDBASQL.Query("MIN()","Orders"
,"0",null,19,"","",arms1d,"",0,"","",c,null,t,1,1);

ArrayList arhd1g1 =WDBASQL.Query("LOC()","Orders"
,"0",null,19,"0002","",null,"",0,"","",c,null,t,1,1);

ArrayList  sum557=WDBASQL.Query("SUM()","Orders"
,"0",null,0,"","",arms1d,"",0,"","",c,null,t,1,1);

ArrayList  arts1= new  ArrayList();
arts1.add(3);
arts1.add(4);
arts1.add(5);
arts1.add(6);
arts1.add(7);
arts1.add(8);

ArrayList arh =WDBASQL.Query("SelectCols","Orders" ,"0",null,12-
3,"5","",arts1,"",0,"","",c,null,t,1,1);
```

```
ArrayList arhd =WDBASQL.Query("Count()","Orders"
,"0",null,13,"u","",null,"",0,"","",c,null,t,1,1);


ArrayList  art= new  ArrayList();


art.add(0);
art.add(1);
art.add(2);
art.add(3);
art.add(4);
art.add(5);
art.add(6);
art.add(7);
art.add(8);
art.add(9);
art.add(10);
art.add(11);
WDBASQL.Query("DISTINCT","abc1","0",null,11,"","",
art,"",0,"11","",c,null,t,1,1);


ArrayList arhd1gy =WDBASQL.Query("MATCH","Orders"
,"0",null,19,"0001","",null,"",0,"1","",c,null,t,1,1);
```

```
ArrayList  ardds= new ArrayList();

for (int i=0;i<myList.size();i++)
ardds.add(i);

WDBASQL.Query("Insert","emp6" ,"",myList,0,"","", null,"",0," ","",c,null,t,1,1);

ArrayList  sum55grh=WDBASQL.Query("DateCompareDESC","emp6"
,"0",null,10,"","",ardds,"",0,"","",c,null,t,1,1);

ArrayList  sum55grhr=WDBASQL.Query("DateCompareASC","emp6"
,"0",null,10,"","",ardds,"",0,"","",c,null,t,1,1);


ArrayList st  = new  ArrayList();

st.add(1);
st.add("wilmix");


st.add("100");

st.add(2);
st.add("jem");

st.add("200");
st.add(4);
st.add("Peter");

st.add("200");
```

```
//st.add(3);
//st.add("Diana");

//st.add("100");
st.add(1);
st.add("");


st.add("500");


WDBASQL.Query("InsertDESC", "emp", "0", st, 0, "", "", null, "", 0, "", "", c, null, t, 0, 1);
ArrayList st111  = new  ArrayList();

st111.add(1);
st111.add("wilmix");


st111.add("100");

st111.add(2);
st111.add("jem");

st111.add("200");
st111.add(4);
st111.add("Peter");

st111.add("200");
```

```
//st111.add(3);
//st111.add("Diana");

//st111.add("100");
st111.add(1);
st111.add("");


st111.add("500");



ArrayList   tsf1p11= WDBASQL.Query("AND", "", "0", null, 11, "", "", sum55grh, "", 0, "", "", c,sum55grhr, t, 1, 4);




ArrayList   tsf1p1= WDBASQL.Query("ForeignKey", "Orders", "0", null, 17, "employess", "", null, "", 0, "", "", c, null, t, 1, 1);

}

}


}
```

Program6:

Write a program and use the following WNOSQL commands

and perform manipulation using

a)DropTable ,InsertDesc,Insert,Insertinto,SelectRval

operations in WNOSQL Table.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{


public void main()

{
```

```
String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char  c=' ';

ArrayList st  = new  ArrayList();

st.add("indno");
st.add("name");

st.add("scoreno");
//WDBASQL.Query("DropTable","nos" ,"0",null,12-
3,"5","",null,"",0,"","",c,null,t,1,1);

WDBASQL.Query("InsertDESC", "emp", "0", st, 0, "", "", null, "", 0, "", "", c, null,
t, 0, 1);

WDBASQL.Query("Insert", "emp", "0", st, 0, "", "", null, "", 0, "", "", c, null, t, 1,
4);

ArrayList st111  = new  ArrayList();

st111.add(1);
st111.add("wilmix");



st111.add("100");

st111.add(2);
st111.add("jem");
```

```
st111.add("200");
st111.add(4);
st111.add("Peter");

st111.add("200");

//st111.add(3);
//st111.add("Diana");

//st111.add("100");
st111.add(1);
st111.add("");


st111.add("500");


WDBASQL.Query("INSERTINTO","emp"
,"0",null,0,"0","",null,"",0,"","",c,st111,t,1,4);
ArrayList ts3j = WDBASQL.Query("SELECTRVAL", "emp", "0", null, 0, "0", "",
null, "", 0, "", "", c, null, t, 1, 4);




}

}


}
```

Program -7:

Write a program and use the following WNOSQL commands

and perform manipulation using

a)Insert,SelectAll,CLUSTER,BACKUPCLUSTER operations in WNOSQL Table.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()

{
```

```
String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char  c=' ';




ArrayList  cols = new  ArrayList();

for (int  i=0;i<=1990;i+=5)

cols.add(i);



ArrayList  cols1 = new  ArrayList();

for (int  i=0;i<=1990;i+=1)

cols1.add(i);
```

```
ArrayList  colsd = new  ArrayList();
WDBASQL.Query("Insert","emp6" ,"0",cols,1999,"","  ", null,"",0,"
","",c,null,t,1,1);



WDBASQL.Query("CLUSTER","emp6" ,"0",null,1990,"","", cols1,"",0,"
","",c,null,t,1,1);

ArrayList  colsdg =WDBASQL.Query("SelectAll","emp6" ,"0",null,1990,"","",
null,"",0,"  ","",c,null,t,1,1);

WDBASQL.Query("CLUSTERPROPERTY","emp6" ,"0",null,1990,"","",
null,"",0,"  ","",c,null,t,1,1);

WDBASQL.Query("BACKUPCLUSTER","emp6" ,"0",null,1990,"","", null,"",0,"
","",c,null,t,1,1);

WDBASQL.Query("SelectAll","emp6" ,"0",null,1990,"","", null,"",0,"
","",c,null,t,1,1);




}

}




}
```

Program:8

Write a program and use the following WNOSQL commands

and perform manipulation using

a) Insertdesc , INSERTINTO,Insert

b) Selectdesc , SelectC,Select R*,MATH*

c) SELECTROWS,SELECTRVAL,SELECTINDEXES

operations in WNOSQL Table.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{


public void main()

{
```

```
String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char  c=' ';



ArrayList  ar= new  ArrayList();

for  (int  i=1;i<=99;i++)

ar.add(i);

ArrayList  ar1= new  ArrayList();
for  (int  i=0;i<=99;i+=3)

ar1.add(i);




ArrayList  ar7= new  ArrayList();
ar7.add("INO");
ar7.add("NOS");
```

```
ar7.add("NAME");
ar7.add("SALARY");



WDBASQL.Query("InsertDESC","nosd"
,"0",ar7,0,"","",null,"",0,"","",c,null,t,0,1);
WDBASQL.Query("Insert","nosd" ,"0",ar7,0,"","",null,"",0,"","",c,null,t,1,3);
WDBASQL.Query("INSERTINTO","nosd"
,"0",null,0,"0","",null,"",0,"","",c,ar1,t,1,3);

WDBASQL.Query("SelectDESC","nosd"
,"0",null,1,"0","",null,"",0,"","",c,null,t,0,1);

WDBASQL.Query("SELECTC*","nosd"
,"0",null,0,"0","",null,"",0,"","",c,null,t,0,1);



WDBASQL.Query("SELECTR*","nosd"
,"0",null,0,"0","",null,"",0,"","",c,null,t,1,3);



ArrayList  art= new  ArrayList();
```

```
art.add(0);
art.add(1);
art.add(2);


WDBASQL.Query("MATH","nosd"
,"0",null,0,"0","",null,"",0,"acos","",c,null,t,1,3);

WDBASQL.Query("SELECTROWS","nosd"
,"0",null,0,"0","",art,"",0,"","",c,null,t,1,3);

WDBASQL.Query("SELECTRVAL","nosd"
,"0",null,0,"0","",null,"",0,"","",c,null,t,1,3);

ArrayList  ar71= new  ArrayList();

ar71.add(4);




ArrayList arhg8ey =WDBASQL.Query("SELECTINDEXES","nosd"
,"0",null,0,"4","",ar71,"",0,"","",c,null,t,1,3);


}

}
```

}

Program 9:

Write a program and use the following WNOSQL commands

and perform manipulation using

a)SelectAssign, Insertvalues, Primary key ,AND

b) SeLectupper,Selectlower

c) SYSDATE, MANIPULATE

d) ENCRYPT,DENCRYPT

operations in WNOSQL Table.

<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

```
{



public void main()

{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");

String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);char  c=' ';



ArrayList  art= new  ArrayList();



art.add(0);
art.add(1);
art.add(2);

ArrayList   tsf1=WDBASQL.Query("SelectAssign","columns"
,"1",null,1,"1","",null,"123,345",0,"","",c,null,t,1,1);
```

```
WDBASQL.Query("InsertValues","columns" ,"1",null,12-
1,"","",null,art.toString(),0,"","INSERT1",c,null,t,1,1);



ArrayList   tsf1p=new ArrayList();


ArrayList   tsf1p1= WDBASQL.Query("PrimaryKey", "abc", "0", null, 11, "abc1",
"", null, "", 0, "", "", c, null, t, 1, 1);

ArrayList   tsf1p11= WDBASQL.Query("AND", "", "0", null, 11, "", "", tsf1p, "", 0,
"", "", c, tsf1p1, t, 1, 1);

ArrayList  art1= new  ArrayList();


for  (int  i=1;i<=6;i++)

art1.add(i);

ArrayList ass1=WDBASQL.Query("SelectUPPER","employess"
,"0",null,6,"","",art1,"",0,"","",c,null,t,1,1);
ArrayList ass11=WDBASQL.Query("SelectLOWER","employess"
,"0",null,6,"","",art1,"",0,"","",c,null,t,1,1);
ArrayList ass12=WDBASQL.Query("SYSDATE",""
,"0",null,6,"","",art1,"",0,"","",c,null,t,1,1);


ArrayList  art11= new  ArrayList();
```

```
art11.add(2016);

art11.add(10);
art11.add(15);

art11.add(5);
art11.add(-5);


WDBASQL.Query("ManipulateDate()","" ,"0",null,6,"","",art1,"",0,"yyyy MMM
dd","",c,null,t,1,1);



WDBASQL.Query("Encrypt", "employess", "0", null, 12 - 3, "5", "", null, "", 0, "",
"", c, null, t, 1, 1);


 WDBASQL.Query("Dencrypt", "employess", "0", null, 12 - 3, "5", "", null, "", 0,
"", "", c, null, t, 1, 1);

}

}

}
```

Program 10:

Write a program and use the following WNOSQL commands

and perform manipulation using

A)Search a DATA

B) SearchLS ,SearchGT

c) SelectRange

Operations in WNOSQL TABLE.

```
<WNOSQL>
<PACK>
<USE> CDollar.WDBA;
<DATALIB> ps
<DATA>
public <CLASS> DATA

{



public void main()
```

```
{



String g=WDBASQL.WDBASQLS("datastorehgh","USEDATABASE",
"wilmix","C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL");


String t= WDBASQL.WDBASQLS("loginuser","pwduser",1,"Wilmix",
"Wilmixjemin12345",1,5,g);
char  c=' ';




WDBASQL.Query("Search","Orders" ,"0",null,15,"100","", null,"",0,"
","",c,null,t,1,1);


WDBASQL.Query("SearchGT","emp6" ,"0",null,150,"100","", null,"",0,"
","",c,null,t,1,1);
WDBASQL.Query("SearchLS","emp6" ,"0",null,150,"100","", null,"",0,"
","",c,null,t,1,1);



WDBASQL.Query("SelectRange","Orders" ,"0",null,15,"","", null,"",0,"
","",c,null,t,1,1);



}


}
```

A) How to use WNOSQL db with CDollar,JDollar, and JAS?

Step-1: *Convert WNOSQL PLSQL to WNOSQL .dll files.*

to be used with CDollar, JDollar, and JAS

Since this programming accept .dll files.

or

Step-2:

You can add the WNOSQL.dll to JDollar CWE editor

Directly write WNOSQL Queries with JDollar CDollar ,JAS ,etc.

and by pressing button browse button at bottom of J$ or C$ CWE Editor

and after that press compile button in CWE Editor and

Run the Program using Run at top right.

How to connect WNOSQL with PHP , JAVA , and other programming languages?

WNOSQL will not directly connect with JAVA ,PHP, etc

So you had to follow the given steps

a) STORE it in SQLSERVER

b)

Write a WNOSQL Program to store the student query values in WDBA table

from SQLSERVER for the given fields sno,tmark,rank and store it in encrypted form

and again store the data in sqlserver for futhure use with C# program.

<WNOSQL>
<PACK>

<DATALIB> ps
<DATA>
public <CLASS> SQL3
{

public void main()

{

*WDBA.writeln((manipulate.Signal("MANIPULATE","Select * from student","student","sno,tmark,rank","?,?,?",4,"sun.jdbc.odbc.JdbcOdbcDriver","jdbc:odbc:dsn","sa","jemin","wilmix21")));*

}

}

</DATA>

Note : use manipulate .dll in this case..

How to connect WNOSQL with JAVA7?

Wnosql directly connects with JAVA7(OAKJAVA).

We can also connect with Core java(JDk version) using WNOSQLDBCONNECTOR.

inorder to hide the database details JAVA7(OAKJAVA) is mostly followed.

How to connect WNOSQL with C#?

it's a very easy step add all the wnosql .dll in

C# program. and you can use the wnosql query in C# program

for manipulation like Add,Find ,update,delete,etc.

How to connect WNOSQL with PHP , and other programming languages?

WNOSQL will not directly connect with PHP, etc

So you had to follow the given steps

Usually we store the data in mysql database
for PHP or any familar database in today market .

we can transport using manipulate.Signal(.....) API and give mysqldriver in this API;

and transport to WNOSQL Db. ALL DATA is safely stored and cannot be seen by hackers.

UNIT :10: WNOSQL(WSQL) Mock Test Exercises and Practice Test for Professionals*

Time Duration : 2 1/2 hours

SECTION -A

A) Create an Online Test Project using CDollar and C#

Use WNOSQL (*) Database in this case ***(1 * 20 = 20 marks)***

B) Describe Briefly about CLUSTER MEMMORY MANAGEMENT

and State the Advantages for WNOSQL (*) database over

Other databases.

(1* 20 = 20 marks)

C) Write any 30 WNOSQL(*) commands and Describe briefly about it.

(1 *25 = 25 marks)

SECTION -B

D) Write a WNOPLSQL(*) Program for storing 15,000 records

from SQLSERVER database. ***(1 *20 = 20 marks)***

and perform the operations

a) SELECTIN,DISTINCT,MATCH

b) Search the data

c) UPDATE the data

e) Delete the data and drop the table

f) Select a Particular data

g) SelectRange of values

h) use SearchGT and SearchLS

i) USE CLUSTER ,CLUSTERPROPERTY , and BACKUPCLUSTER

j) SelectAssign,SelectIndexes,SelectRval

E) Write a WNOPLSQL(*) Program for storing 10,000 records from SQLSERVER database.

and perform the operations ***(1 *15 = 15 Marks)***

A) Perform FULLJOIN ,RIGHT JOIN ,LEFTJOIN ,USE HAVING CLAUSE bettween two tables.

Note: Practice this Test and do any projects or assignments