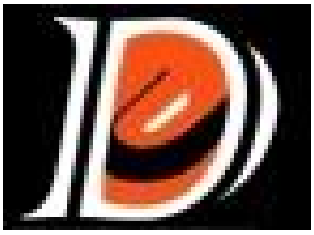


Dharmsinh Desai University
Faculty of Technology
Department of Computer Engineering

Lab Manual

Subject: Database Systems
B.Tech. (CE)
Semester V

Hardware/Software Requirement:

The Database System in use is based on client-server architecture. Learn more from appendix. 1. The Oracle Client/Server Architecture by Oracle.

Oracle database server computer accessible in local area network. Students are required to use any database client to connect to database server from their system. To access the database, GUI based sql client named SqlDeveloper is advised to use.

Here, because accessing remote database is through client software tool like SqlDeveloper, students may use Windows or Linux Operating System.

SqlDeveloper is supported by both operating systems.

Please, collect the server details and credentials from the faculty in charge of conducting lab sessions.

Refer to appendix 2. How to use SQLDeveloper – For Beginners.

Theory and Concept

Practical #1

Objective: Create tables, insert data, develop and test SQL for provided questionnaires.

Theory & Concepts:

Introduction about SQL-

SQL (Structured Query Language) is a nonprocedural language, you specify what you want, not how to get it. A block structured format of English key words is used in this Questionry language. It has the following components.

DDL (Data Definition Language)-

The SQL DDL provides command for defining relation schemas, deleting relations and modifying relation schema.

DML (DATA Manipulation Language)-

It includes commands to insert tuples into, delete tuples from and modify tuples in the database.

View definition-

The SQL DDL includes commands for defining views.
Transaction Control- SQL includes for specifying the beginning and ending of transactions.

Embedded SQL and Dynamic SQL-

Embedded and Dynamic SQL define how SQL statements can be embedded with in general purpose programming languages, such as C, C++, JAVA, COBOL, Pascal and Fortran.

Integrity-

The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must specify. Updates that violate integrity constraints are allowed.

Authorization-

The SQL DDL includes commands for specifying access rights to relations and views.

Data Definition Language-

The SQL DDL allows specification of not only a set of relations but also information about each relation, including-

- Schema for each relation
- The domain of values associated with each attribute.
- The integrity constraints.
- The set of indices to be maintained for each relation.

- The security and authorization information for each relation.
- The physical storage structure of each relation on disk.

Domain types in SQL-

The SQL standard supports a variety of built in domain types, including-

- Char (n)- A fixed length character length string with user specified length .
- Varchar (n)- A variable character length string with user specified maximum length n.
- Int- An integer.
- Small integer- A small integer.
- Numeric (p, d)-A Fixed point number with user defined precision.
- Real, double precision- Floating point and double precision floating point numbers with machine dependent precision.
- Float (n)- A floating point number, with precision of at least n digits.
- Date- A calendar date containing a (four digit) year, month and day of the month.
- Time- The time of day, in hours, minutes and seconds Eg. Time '09:30:00' .
- Number- Number is used to store numbers (fixed or floating point).
- Binary large objects

DDL statement for creating a table-

Syntax-

Create table tablename
(columnname datatype(size), columnname datatype(size));

Creating a table from a table-

Syntax-

CREATE TABLE TABLENAME
[(columnname, columnname, … … …)]
AS SELECT columnname, columnname… … ..FROM tablename;

Insertion of data into tables-

Syntax-

INSERT INTO tablename
[(columnname, columnname, … …
…)] Values(expression, expression);

Inserting data into a table from another table:

Syntax-

INSERT INTO tablename
SELECT columnname, columnname, … … .

FROM tablename;

Insertion of selected data into a table from another table:

Syntax-
INSERT INTO <tablename>
SELECT <columnname, columnname… … ..>
FROM <tablename>
[WHERE columnname = expression];

Retrieving of data from the tables-

Syntax-
SELECT * FROM <tablename>;

The retrieving of specific columns from a table-

Syntax-
SELECT <columnname, columnname, … .>
FROM <tablename>;

Elimination of duplicates from the select statement-

Syntax-
SELECT DISTINCT <columnname, columnname>
FROM <tablename>;

Selecting a data set from table data-

Syntax-
SELECT <columnname, columnname,...>
FROM <tablename>
[WHERE searchcondition];

Assignment No.1

Q1. Create the following tables:

i) client_master

Columnname	datatype	size
client_no	varchar2	6
name	varchar2	20
address1	varchar2	30

address2	varchar2	30
city	varchar2	15
state	varchar2	15
pincode	number	6
bal_due	number	10,2

ii) Product_master

Columnname datatype size Product_no varchar2

Description varchar2
Profit_percent number
Unit_measure varchar2
Qty_on_hand number
Reoder_lvl number
Sell_price number Cost_price number

Q2- Insert the following data into their respective tables:

Clientno	Name	city	pincode	state	bal.due
0001	Ivan	Bombay	400054	Maharashtra	15000
0002	Vandana	Madras	780001	Tamilnadu	0
0003	Pramada	Bombay	400057	Maharashtra	5000
0004	Basu	Bombay	400056	Maharashtra	0
0005	Ravi	Delhi	100001	UtterPradesh	2000
0006	Rukmani	Bombay	400050	Maharashtra	0

Data for Product Master:

Product No.	Description	Profit % Percent	Unit measured	Qty on hand	Reorder lvl	Sell price	Cost price
P00001	1.44floppies	5	piece	100	20	525	500
P03453	Monitors	6	piece	10	3	12000	11200
P06734	Mouse	5	piece	20	5	1050	500
P07865	1.22 floppies	5	piece	100	20	525	500
P07868	Keyboards	2	piece	10	3	3150	3050
P07885	CD Drive	2.5	piece	10	3	5250	5100
P07965	540 HDD	4	piece	10	3	8400	8000
P07975	1.44 Drive	5	piece	10	3	1050	1000
P08865	1.22 Drive	5	piece	2	3	1050	1000

Q3:- On the basis of above two tables answer the following Questionnaires:

i) Find out the names of all the clients.
ii) Retrieve the list of names and cities of all the clients.
iii) List the various products available from the product_master table.

iv) List all the clients who are located in Bombay.
v) Display the information for client no 0001 and 0002.
vi) Find the products with description as ‘1.44 drive’ and ‘1.22 Drive’ .
vii) Find all the products whose sell price is greater then 5000.
viii) Find the list of all clients who stay in in city ‘Bombay’ or city ‘Delhi’ or ‘Madras’ .
ix) Find the product whose selling price is greater than 2000 and less than or equal to 5000.
x) List the name, city and state of clients not in the state of ‘Maharashtra’ .

Theory and Concept
Practical #2

Objective:- To Manupulate the Operations on the table.
DML (Data Manipulation Language) Data manipulation is

- The retrieval of information stored in the database.
- The insertion of new information into the database.
- The deletion of information from the database.
- The modification of information stored by the appropriate data model. There are basically two types.

(i) **Procedural DML**:- require a user to specify what data are needed and how to get those data.

(ii) **Non Procedural DML** : require a user to specify what data are needed without specifying how to get those data.

Updating the content of a table:
In creation situation we may wish to change a value in table without changing all values in the tuple . For this purpose the update statement can be used.

Update table name
Set columnname = experision, columnname =expression…
Where columnname = expression;

Deletion Operation:-
A delete reQuestionst is expressed in much the same way as Questionry. We can delete whole tuple (rows) we can delete values on only particulars attributes.

Deletion of all rows

Syntax:
Delete from tablename :

Deletion of specified number of rows
Syntax:

Delete from table name
Where search condition ;

Computation in expression lists used to select data

+	Addition	- Subtraction	
*	multiplication	** exponentiation	
/ Division		() Enclosed operation	

Renaming columns used with Expression Lists: - The default output column names can be renamed by the user if required

Syntax:

Select column name result_columnname,
Columnname result_columnname,
From table name;

Logical Operators:
The logical operators that can be used in SQL sentenced are

AND	all of must be included
OR	any of may be included
NOT	none of could be included

Range Searching: Between operation is used for range searching.

Pattern Searching:
The most commonly used operation on string is pattern matching using the operation 'like' we describe patterns by using two special characters.

- Percent (%) ; the % character matches any substring we consider the following examples.
- 'Perry %' matches any string beginning with perry
- '% idge % matches any string containing' idge as substring.
- ' - - - ' matches any string exactly three characters.
- ' - - - % matches any string of at least of three characters.

Oracle functions:
Functions are used to manipulate data items and return result. function follow the format of function _name (argument1, argument2 ..) .An arrangement is user defined variable or constant. The structure of function is such that it accepts zero or more arguments.

Example s: Avg return average value of n

Syntax:
Avg ([distinct/all]n)
Min return minimum value of expr.

Syntax:

MIN((distict/all)expr)
Count Returns the no of rows where expr is not null
Syntax:
Count ([distinct/all)expr]
Count (*) Returns the no rows in the table, including duplicates and those with nulls.
Max Return max value of expr

Syntax:

Max ([distinct/all]expr)
Sum Returns sum of values of n

Syntax:
Sum ([distinct/all]n)

Sorting of data in table

Syntax:
Select columnname, columnname
From table
Order by columnname;

Assignment No. # 2

Question.1 Using the table client master and product master answer the following Questionries.

i. Change the selling price of '1.44 floppy drive to Rs.1150.00
ii. Delete the record with client 0001 from the client master table.
iii. Change the city of client_no'0005' to Bombay.
iv. Change the bal_due of client_no '0001, to 1000.
v. Find the products whose selling price is more than 1500 and also find the new selling price as original selling price *15.
vi. Find out the clients who stay in a city whose second letter is a.
vii. Find out the name of all clients having 'a' as the second letter in their names.
viii. List the products in sorted order of their description.
ix. Count the total number of orders
x. Calculate the average price of all the products.
xi. Calculate the minimum price of products.
xii. Determine the maximum and minimum prices . Rename the tittle as 'max_price' and min_price respectively.
xiii. Count the number of products having price greater than or equal to 1500.

Theory and Concept
Practical #3

Objective:- To Implement the restrictions on the table.

Data constraints: Besides the cell name, cell length and cell data type there are other parameters i.e. other data constrains that can be passed to the DBA at check creation time. The constraints can either be placed at column level or at the table level.

i. **Column Level Constraints:** If the constraints are defined along with the column definition, it is called a column level constraint.
ii. **Table Level Constraints:** If the data constraint attached to a specify cell in a table reference the contents of another cell in the table then the user will have to use table level constraints.

Null Value Concepts:- while creating tables if a row locks a data value for particular column that value is said to be null . Column of any data types may contain null values unless the column was defined as not null when the table was created

Syntax:

Create table tablename
(columnname data type (size) not null … …)

Primary Key: primary key is one or more columns is a table used to uniquickly identity each row in the table. Primary key values must not be null and must be uniQuestion across the column. A multicolumn primary key is called composite primary key.

Syntax: <u>primary key as a column constraint</u>
Create table tablename
(columnname datatype (size) primary key,….)
<u>Primary key as a table constraint</u>
Create table tablename
(columnname datatype (size), columnname datatype(size)
… Primary key (columnname,columnname));

UniQuestion key concept:-A uniQuestion is similar to a primary key except that the purpose of a uniQuestion key is to ensure that information in the column for each record is uniQuestion as with telephone or devices license numbers. A table may have many uniQuestion keys.

Syntax: <u>UniQuestion as a column constraint</u>.
Create table table name
(columnname datatype (size) uniQuestion);
<u>UniQuestion as table constraint:</u>
Create table tablename

(columnname datatype (size),columnname datatype (size)… uniQuestion (columnname,columnname));

Default value concept: At the line of cell creation a default value can be assigned to it. When the user is loading a record with values and leaves this cell empty, the DBA wil automatically load this cell with the default value specified. The data type of the default value should match the data type of the column

Syntax:

Create table tablename
(columnname datatype (size) default value,… .);

Foreign Key Concept : Foreign key represents relationship between tables. A foreign key is column whose values are derived from the primary key of the same of some other table . the existence of foreign key implies that the table with foreign key is related to the primary key table from which the foreign key is derived .A foreign key must have corresponding primary key value in the primary key table to have meaning.
Foreign key as a column constraint

Syntax :
Create table table name
(columnname datatype (size) references another table name);

Foreign key as a table constraint:

Syntax :
Create table name
(columnname datatype (size)
… . primary key
(columnname);
foreign key (columnname)references table name);

Check Integrity Constraints: Use the check constraints when you need to enforce intergrity rules that can be evaluated based on a logical expression following are a few examples of appropriate check constraints.

- A check constraints name column of the coient_master so that the name is entered in upper case.
- A check constraint on the client_no column of the client _master so that no client_no value starts with ‘c’

Syntax:
Create table tablename
(columnname datatype (size) CONSTRAINT
constraintname) Check (expression));

Question.2 Create the following tables:

i. **Salesman_master**

Columnname	**Datatype**	**Size**	**Attributes**
Salesman_no	varchar2	6	Primary key/first letter must start with ‘s’
Sal_name	varchar2	20	Not null
Address	varchar2		Not null
City	varchar2	20	
State	varchar2	20	
Pincode	Number	6	
Sal_amt	Number	8,2	Not null, cannot be 0
Tgt_to_get	Number	6,2	Not null, cannot be 0
Ytd_sales	Number	6,2	Not null, cannot be 0
Remarks	Varchar2	30	

ii. Sales_order

Columnname	**Datatype**	**Size**	**Attributes**
S_order_no	varchar2	6	Primary/first letter must be 0
S_order_date	Date		
Client_no	Varchar2	25	Primary key reference clientno of client_master table
Dely_add	Varchar2	6	
Salesman_no	Varchar2	6	Foreign key references salesman_no of salesman_master table
Dely_type	Char	1	Delivery part(p)/full(f),default f
Billed_yn	Char	1	
Dely_date	Date		Can not be lessthan s_order_date
Order_status	Varchar2	10	Values (‘in process’;’fulfilled’;back order’;’canceled

I. Sales_order_details

Column	Datatype	Size	Attributes
S_order_no	Varchar2	6	Primary key/foreign key references s_order_no of sales_order
Product_no	Varchar2	6	Primary key/foreign key references product_no of product_master
Qty_order	Number	8	
Qty_disp	Number	8	
Product_rate	Number	10,2	

Insert the following data into their respective tables using insert statement:

Data for sales_man master table

Salesman_no	Salesman name	Address	City	Pin code	State	Salamt	Tgt_to_get	Ytd Sales	Remark
500001	Kiran	A/14 worli	Bombay	400002	Mah	3000	100	50	Good
500002	Manish	65,nariman	Bombay	400001	Mah	3000	200	100	Good
500003	Ravi	P-7 Bandra	Bombay	400032	Mah	3000	200	100	Good
500004	Ashish	A/5 Juhu	Bombay	400044	Mah	3500	200	150	Good

(ii)

Data for salesorder table:

S_orderno	S_orderdate	Client no	Dely type	Bill yn	Salesman no	Delay date	Orders tatus
019001	12-jan-	0001	F	N	50001	20-jan-	Ip

	96					96	
019002	25-jan-96	0002	P	N	50002	27-jan-96	C
016865	18-feb-96	0003	F	Y	500003	20-feb-96	F
019003	03-apr-96	0001	F	Y	500001	07-apr-96	F
046866	20-may-96	0004	P	N	500002	22-may-96	C
010008	24-may-96	0005	F	N	500004	26-may-96	Ip

(iii)

Data for sales_order_details table:

S_order no	Product no	Qty ordered	Qty disp	Product_rate
019001	P00001	4	4	525
019001	P07965	2	1	8400
019001	P07885	2	1	5250
019002	P00001	10	0	525
046865	P07868	3	3	3150
046865	P07885	10	10	5250
019003	P00001	4	4	1050
019003	P03453	2	2	1050
046866	P06734	1	1	12000
046866	P07965	1	0	8400
010008	P07975	1	0	1050
010008	P00001	10	5	525

Theory and Concept
Practical No.4

Objective:- To Implement the structure of the table

Modifying the Structure of Tables- Alter table command is used to changing the structure of a table. Using the alter table clause you cannot perform the following tasks:

(i) change the name of table
(ii) change the name of column
(iii) drop a column
(iv) decrease the size of a table if table data exists.

The following tasks you can perform through alter table command.

(i) **Adding new columns**: Syntax
ALTER TABLE tablename
ADD (newcolumnname newdatatype (size));

(ii) Modifying existing table
Syntax:
ALTER TABLE tablename
MODIFY (newcolumnname newdatatype (size));

***NOTE*:** Oracle not allow constraints defined using the alter table, if the data in the table, violates such constraints.

Removing/Deleting Tables- Following command is used for removing or deleting a table.

Syntax:
DROP TABLE tabename:

Defining Integrity constraints in the ALTER TABLE command-

You can also define integrity constraints using the constraint clause in the ALTER TABLE command. The following examples show the definitions of several integrity constraints.

(1) **Add PRIMARY KEY**-Syntax:
ALTER TABLE tablename
ADD PRIMARY KEY(columnname);

(2) **Add FOREIGN KEY**-Syntax:
ALTER TABLE tablename

ADD CONSTRAINT constraintname

FOREIGN KEY(columnname) REFERENCES tablename; Droping integrity constraints in the ALTER TABLE command:

You can drop an integrity constraint if the rule that if enforces is no longer true or if the constraint is no longer needed. Drop the constraint using the ALTER TABLE command with the DROP clause. The following examples illustrate the droping of integrity constraints.

(1) **DROP the PRIMARY KEY**-Syntax:
ALTER TABLE tablename
DROP PRIMARY KEY

(2) **DROP FOREIGN KEY**-Syntax:
ALTER TABLE tablename
DROP CONSTRAINT constraintname;

Table Name : Challan_Header

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key
s_order_no	Varchar2	6	Foreign key references s_order_no of sales_order table
challan_date	Date		Not null
billed_yn	Char	1	Values ('Y','N') default 'N'

Table Name : Challan_Details

Column name	data type	size	Attributes
Challan_no	varchar2	6	Primary key/Foreign key references Product_no of product_master
Qty_disp	number	4,2	not null

(i)

Challan No	S_order No	Challan Date Billed
CH9001	019001	12-DEC-95 Y
CH865	046865	12-NOV-95Y
CH3965	010008	12-OCT-95 Y

Data for challan_details table

Challan No	Product No	Qty Disp
CH9001	P00001	4
CH9001	P07965	1
CH9001	P07885	1
CH6865	P07868	3
CH6865	P03453	4
CH6865	P00001	10
CH3965	P00001	5
CH3965	P07975	2

Assignment#4

Objective –Answer the following Questionries

Q1. Make the primary key to client_no in client_master.

Q2. Add a new column phone_no in the client_master table.

Q3. Add the not null constraint in the product_master table with the columns description, profit percent , sell price and cost price.

Q4. Change the size of client_no field in the client_master table.

Q5. Select product_no, description where profit percent is between 20 and 30 both inclusive.

Theory & Concept
Practical #5

Objective:- to implement the concept of Joins

Joint Multiple Table (Equi Join): Some times we require to treat more than one table as though manipulate data from all the tables as though the tables were not separate object but one single entity. To achieve this we have to join tables.Tables are joined on column that have dame data type and data with in tables.

The tables that have to be joined are specified in the FROM clause and the joining attributes in the WHERE clause.

Algorithm for JOIN in SQL:

1. Cartesian product of tables (specified in the FROM clause)
2. Selection of rows that match (predicate in the WHERE clause)
3. Project column specified in the SELECT clause.

1. Cartesian product:-

Consider two table student and course

```
Select B.*,P.*
FROM student B, course P;
```

2. INNER JOIN:

Cartesian product followed by selection

```
Select B.*,P.*
FROM student B, Course P
WHERE B.course # P.course # ;
```

3. LEFT OUTER JOIN:

LEFT OUTER JOIN = Cartesian product + selection but include rows from the left table which are unmatched pat nulls in the values of attributes belonging to th e second table

Exam:

```
Select  B.*,P*
FROM student B left join course
p ON B.course # P.course #;
```

4. RIGHT OUTER JOIN:

RIGHT OUTER JOIN = Cartesian product + selection but include rows from right table which are unmatched

Exam:

```
Select B.*,P.*
```

From student B RIGHT JOIN course P

B.course# = P course # ;

5. FULL OUTER JOIN

Exam
Select B.*,P.*
From student B FULL JOIN course P
On B.course # = P course # ;

ASSIGNMENT NO. 5

OBJECTIVE: Answer the following Questionries:
Questionstion.

1. Find out the product which has been sold to ‘ Ivan Sayross.’
2. Find out the product and their quantities that will have do delivered.
3. Find the product_no and description of moving products.
4. Find out the names of clients who have purchased ‘CD DRIVE’
5. List the product_no and s_order_no of customers haaving qty ordered less than 5 from the order details table for the product “ 1.44 floppies” .
6. Find the products and their quantities for the orders placed by ‘Vandan Saitwal ’ and “ Ivan Bayross” .
7. Find the products and their quantities for the orders placed by client_no “ C00001” and “ C00002”
8. Find the order No,, Client No and salesman No. where a client has been received by more than one salesman.
9. Display the s_order_date in the format “ dd-mm-yy” e.g. “ 12- feb-96”
10. Find the date , 15 days after date.

Theory & Concept
Practical # 6

Objective:- To implement the concept of grouping of Data. Grouping Data From Tables:

There are circumstances where we would like to apply the aggregate function not only to a single set of tuples, but also to a group of sets of tuples, we specify this wish in SQL using the group by clause. The attribute or attributes given in the group by clause are used to form group. Tuples with the same value on all attributes in the group by clause are placed in one group.

Syntax:

```
SELECT columnname, columnname FROM tablename

GROUP BY columnname;
```

At times it is useful to state a condition that applies to groups rather than to tuples. For example we might be interested in only those branches where the average account balance is more than 1200. This condition does not apply to a single tuple, rather it applies to each group constructed by the GROUP BY clause. To express such Questionry, we use the having clause of SQL. SQL applies predicates in the having may be used.

Syntax:

```
SELECT columnname, columnname
FROM tablename
GROUP BY columnname;
HAVING searchcondition;
```

Assignment No.6

Objective- Answer the following Questionries:

Q1.- Print the description and total quantity sold for each product. Q2.- Find the value of each product sold.

Q3.- Calculate the average quantity sold for each client that has a maximum order value of 15000.

Q4.- Find out the products which has been sold to Ivan. Q5.- Find the names of clients who have 'CD Drive'.

Q6.- Find the products and their quantities for the orders placed by 'Vandana' and 'Ivan'.

Q7.- Select product_no, total qty_ordered for each product.

Q8.- Select product_no, product description and qty ordered for each product.

Q9.- Display the order number and day on which clients placed their order.

Q10.- Display the month and Date when the order must be delivered.

Theory & Concept
Practical #7

Objective:- To implement the concept of SubQuestionries.

SubQuestionries:- A subQuestionry is a form of an SQL statement that appears inside another SQL statement. It also termed as nested Questionry. The statement containing a subQuestionry called a parent statement. The rows returned bu the subQuestionry are use by the following statement.

It can be used by the following commands:

1. To insert records in the target table.
2. To create tables and insert records in this table.
3. To update records in the target table.
4. To create view.
5. To provide values for the condition in the WHERE , HAVING IN , SELECT,UPDATE, and DELETE statements.

Exam:-

Creating clientmaster table from oldclient_master, table

```
Create table client_master
AS SELECT * FROM oldclient_master;
```

Using the Union, Intersect and Minus Clause:

Union Clause:

The user can put together multiple Questionries and combine their output using the union clause . The union clause merges the output of two or more Questionries into a single set of rows and column. The final output of union clause will be

Output: = Records only in Questionry one + records only in Questionry two + A single set of records with is common in the both Questionries.

Syntax:

```
SELECT columnname, columname
FROM tablename 1
UNION
SELECT columnname,  columnname
From tablename2;
```

Intersect Clause: The use can put together multiple Questionries and their output using the interest clause. The final output of the interest clause will be :

Output =A single set of records which are common in both Questionries

Syntax:

```
SELECT  columnname, columnname
FROM tablename 1
INTERSECT
SELECT columnname, columnname
FROM tablename 2;
```

MINUS CLAUSE:- The user can put together multiple Questionries and combine their output = records only in Questionry one

Syntax:

```
SELECT columnname,
columnname FROM tablename ;
MINUS
SELECT columnname,
columnname FROM tablename ;
```

Assignment NO.7

Objective: Answer the following Questionaries:

Question.

1. Find the product_no and description of non- moving products.
2. Find the customer name, address, city and pincode for the client who has placed order no " 019001"
3. Find the client names who have placed order before the month of may 96.
4. Find out if product " 1.44 Drive" is ordered by only client and print the client_no name to whom it was sold.
5. find the names of client who have placed orders worth Rs.10000 or more.
6. Select the orders placed by 'Rahul Desai"
7. Select the names of persons who are in Mr. Pradeep's department and who have also worked on an inventory control system.
8. Select all the clients and the salesman in the city of Bombay.
9. Select salesman name in " Bombay" who has atleast one client located at " Bombay"
10. Select the product_no, description, qty_on-hand,cost_price of non_moving items in the product_master table.

Theory and Concept
Practical # 8

Objective:- To implement the concept of Indexes and views.

Indexes- An index is an ordered list of content of a column or group of columns in a table. An index created on the single column of the table is called simple index. When multiple table columns are included in the index it is called composite index.

Creating an Index for a table:-

Syntax (Simple)

```
CREATE INDEX index_name
ON tablename(column name);
```

Composite Index:-

```
CREATE INDEX index_name
ON tablename(columnname,columnname);
```

Creating an UniQuestion Index:-

```
CREATE UNIQUESTION INDEX indexfilename
ON tablename(columnname);
```

Dropping Indexes:-

An index can be dropped by using DROP INDEX

SYNTAX:-

```
DROP INDEX indexfilename;
```

Views:-

Logical data is how we want to see the current data in our database. Physical data is how this data is actually placed in our database.

Views are masks placed upon tables. This allows the programmer to develop a method via which we can display predetermined data to users according to our desire.

Views may be created fore the following reasons:

1. The DBA stores the views as a definition only. Hence there is no duplication of data.
2. Simplifies Questionries.
3. Can be Questionried as a base table itself.
4. Provides data security.
5. Avoids data redundancy.

Creation of Views:-

Syntax:-

```
CREATE VIEW viewname AS
SELECT columnname,columnname
FROM tablename
WHERE columnname=expression_list;
```

Renaming the columns of a view:-

Syntax:-

```
CREATE VIEW viewname AS
SELECT newcolumnname…      .
FROM tablename
WHERE columnname=expression_list;
```

Selecting a data set from a view-

Syntax:-

```
SELECT columnname, columnname
FROM viewname
WHERE search condition;
```

Destroying a view-

Syntax:-

```
DROP VIEW viewname;
```

Assignment # 8

Objective : Answer the following Questionstions

Q1. Create an index on the table client_master, field client_no.

Q2. Create an index on the sales_order, field s_order_no.

Q3. Create an composite index on the sales_order_details table for the columns s_order_no and product_no.

Q4. Create an composite index ch_index on challan_header table for the columns challan no and s_order_no.

Q5. Create an uniQuestion index on the table salesman_master, field salesman_no.

Q6. Drop index ch_index on table challan_header.

Q7. Create view on salesman_master whose sal_amt is less than 3500.

Q8. Create a view client_view on client_master and rename the columns as name, add1, add2, city, pcode, state respectively.

Q9. Select the client names from client_view who lives in city ‘Bombay’ .

Q10. Drop the view client_view.

Theory and Concept
Practical No. 9

Objective:- To Implement the concept of Triggers and TCL statements.

The objective of the lab is to create a trigger and test how does it fire. Also to learn the purpose when shall database designers utilize the concept of trigger.

Trigger is a block of code which runs automatically when the event as defined occurs. Know that event in terms of data change and the automatic action will be run by dbms as per the instruction written in the definition of trigger. Hence, it is told that dbms trigger triggers when defined requirement happens.

Each trigger shall have unique name to identify.

Syntax with example of a trigger on securityman table. The logic is whenever insert or update DML on this table is run for each row, if it is inserting we want to have joining_date column value as of system date automatically. Know that value for this column is not provided by the insert statement.Also, same story with age when joined should be found out based on dob provided in the insertion values. NEW refers to data values provided in insert/update statement. OLD refers to existing in the table record. :

If later somehow when dob is required to be updated, ageWhenJoined shall be recalculated and updated as well.

```
create or replace trigger tr_smanagewhenjoined
BEFORE
insert or update on "SECURITYMAN"
for each row
begin
  if inserting then
    :NEW.joining_date := sysdate;
    :NEW.agewhenjoined := floor(months_between(sysdate, :NEW.dob)/12);
 end if;

 if updating then
  :NEW.agewhenjoined := floor(months_between(:OLD.joining_date, :NEW.dob)/12);
 end if;

end;

drop trigger tr_smanagewhenjoined;
```

Exercise:

select floor(months_between(sysdate, to_date('01-MAR-1988','DD-MON-YYYY'))/12) from dual;

```
create table securityman
(id number, name varchar2(20), dob date, agewhenjoined number, joining_date date);

select * from securityman; --No records as of now.

@create_trigger.sql; --Run the above trigger definition.
```

After inserting below record, check the agewhenjoined column and learn that eventhough it is not provided in insert statement, because of internal trigger it is counted and recorded automatically. (If the trigger is not present then you will not see this effect)

```
insert into securityman
(id, name, dob)
values
(1,'James','01-JAN-1997');

insert into securityman
(id, name, dob)
values
(2,'Karen','01-MAR-1990');

insert into securityman
(id, name, dob)
values
(3,'Naic','25-APR-1995');
```

Also, try updating existing record with a different birth date, it shall automatically run the update block of trigger and have the agewhenjoined recounted and recorded.

```
update securityman
set dob=to_date('15-AUG-1993','DD-MON-YYYY')
where id=2;


select * from securityman;
```

TCL Statements:

Transaction Control Language includes:
-commit
-rollback
-savepoint

Commit:

Make the changes permanent to database;

Explicit commit using the "commit" keyword.

Implicit commit happens (1) before and after any DDL (create, alter, drop, grant, revoke) automatically. (2) successful exit of client.

Any DML statements prior to commit event (implicity or explicity) are in a sort of staging area (temporary), where the user can work in what is almost a "draft" mode. However, any commit event, explicit or implicit - will make changes permanent and expose the new data to the user population at large.

Rollback:
Somewhat equivalent to "undo".
It undoes any changes to the draft within a given session till prior commit point. It does not remove any changes that have been already been committed.

i.e.
commit;
insert into ...;
delete from ...;
rollaback;

In this example, one insert and one delete statement (DML) are issued. When we issue rollback the both changes are eliminated from the current draft.

Savepoint:
Demarcation points within a transaction in order to empower any following commit and or rollback statments to subdivide the points at which data can be saved or undone.

If periodic savepoints (i.e. labels/checkpoints/names of SCN-System change number) have been issued along the way, then the following commit or rollback can be designed to save or restore data to those points in time.

i.e.
commit;
update ...;
savepoint sp_one;
update ...;
rollback work to sp_one;
commit;

In this example, because of savepoint we have a choice to rollback bottom update and commit only upto sp_one, which includes any and all prior to that savepoint.

Theory and Concept
Practical No. 10
Objective:- To Implement the concept of Forms and reports.

The objective of the lab is to create a form using a parent table and a child table to take advantage of the schema's relationships.

A data block in Oracle Forms = A table in the database.
Each block contains items which equal table columns or fields.
These are arranged into records.

1. Start Schema Builder. Open S_Customer and S_Order or S_Order1.
2. Start Form Builder. Use the data block wizard to create a form for S_Customer, including the Name, ID, Phone, and Address columns.
3. After the form is created, click on Window on the Object Navigator to expand it. Right click on Window1. Click on Property Pallet. Go to Physical on property pallet. Make sure Show Horizontal Scroll Bar and Show Vertical Scroll Bar both are YES.
4. Run the form. Execute the Questionry. Notice that data already exists in this table.
5. Highlight Data Blocks in the Object Navigator. Go up to Tools – Data Block Wizard.
6. Create a form for S_Order or S_Order1.
7. Include the columns ID – Customer_ID – Date_Ordered – Date_Shipped – Total.
8. Click Create Relationship. Click OK. Make sure Autojoin Datablocks is checked.
9. Check Detail Item to Customer_ID and Master Item to ID. This says that the parent table, the table on the one side of the relationship has the primary key of ID in the S_Customer table, and the foreign key on the many side is Customer_ID in the S_Order table. This relationship can be seen if you open schema builder and look at the tables and the relationship between them.
10. Make the layout tabular.
11. Records displayed will be 5 and Display Scrollbar will be checked off.
12. Run the form and execute the Questionry. Scroll through the data and notice that the orders are linked with the customers.
13. If you input a detail, the foreign key is automatically filled with the value of the current primary key displayed by the customer.

To delete a record, you must first delete the details, save the change, then delete the master and save. After you make any change, save the data. If there is any problem with integrity, the record won't save.

If you have two table joined by a relationship table, the relationship table MUST contain valid data or you will not be able to add data to the detail table.
IE – If you have two tables and one relation table such as – Students – Have – Marks
The have table would include at least one column, a StudentID. To input a mark, the StudentID must be filled in the Students table, a matching StudentID would have to exist in the Have table, and a valid MarkID must exist in the marks table that = a StudentID in the Have table which = a StudentID in the Students table.

To create a grouped report:

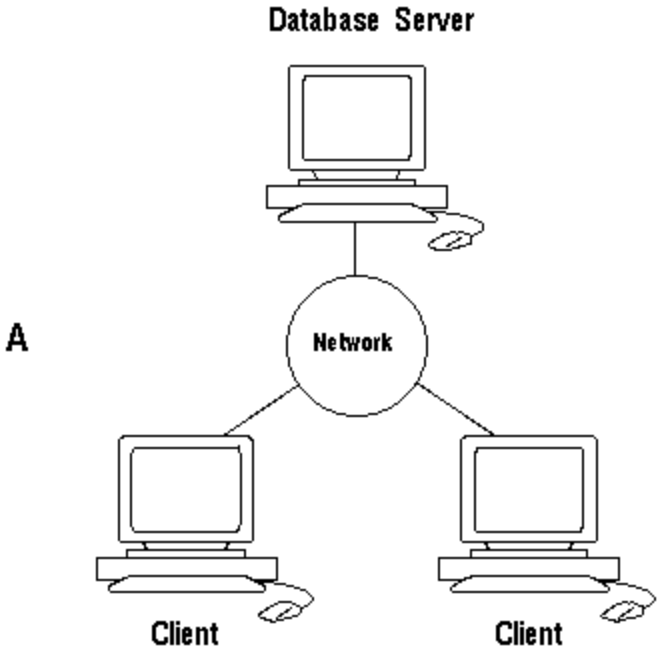
1. Start report builder.
2. Select " use the report wizard"
3. Title the report " Mainenance Bill" and select Group Above and click next.
4. Click on Questionry Builder.
5. Select the tables Jdrivers, JmaintenanceBill, JMBCodeandcosts, and Jvendors.
6. Select the fields by double clicking them. A check mark should appear to the left of the field when it is selected. The fields you should select are:
   - Jdrivers – Drsupervisor
   - JmaintenanceBill – All fields except VandorNumber
   - JMBCodeandcosts – All except Mbillnumber
   - Jvendors – Vencompanyname
7. Click OK. Click Next.
8. The grouping window will now open. Select Mbillno and include it. Above it on the right hand side, the words Group 1 should appear. Next select Vencompanyname. The words Group 2 should appear above it. Click on the Vencompanyname you have just selected on the right and drag it up to group 1. It is now included in group 1 if the Group 2 is no longer displayed. Do this for the remaining fields. Include all fiends EXCEPT SRVCODE, SCPARTCOST, and SCLABCOST. Click next.
9. Now you can include the remaining fields you left out before. Ie- SRVCODE, SCPARTCOST, and SCLABCOST. Click next.
10. We will do two sums on some fields. Select SCPARTCOSE and click the sum button. Do the same for SCLABCOST. Click next.
11. When choosing the layout template, select Cyan Grid. Click finished.
12. If the layout is not as you would like, you can edit the layout by clicking on View on the menu bar and selecting Layout Model.

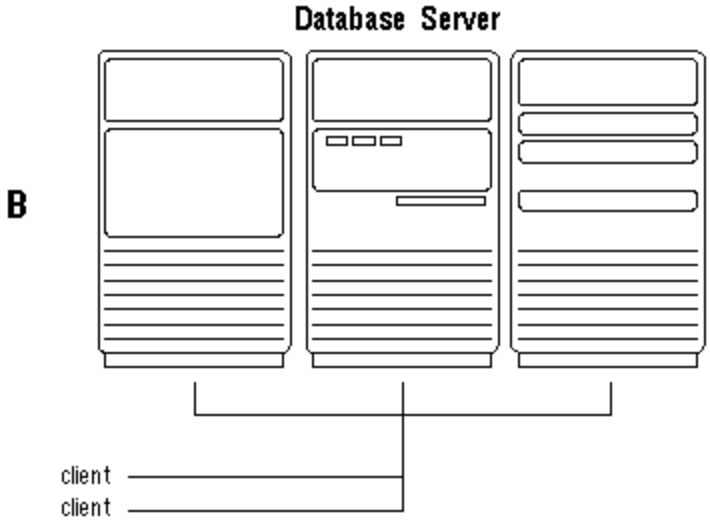
Appendix

1. The Oracle Client/Server Architecture

Courtesy: http://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch20.htm#o%20client/server

In the Oracle client/server architecture, the database application and the database are separated into two parts: a front-end or client portion, and a back-end or server portion. The client executes the database application that accesses database information and interacts with a user through the keyboard, screen, and pointing device such as a mouse. The server executes the Oracle software and handles the functions required for concurrent, shared data access to an Oracle database.





Although the client application and Oracle can be executed on the same computer, it may be more efficient and effective when the client portion(s) and server portion are executed by different computers connected via a network. The following sections discuss possible variants in the Oracle client/server architecture.

Note: In a distributed database, one server (Oracle) may need to access a database on another server. In this case, the server requesting the information is a client.

Distributed Processing
Distributed processing is the use of more than one processor to divide the processing for an individual task. The following are examples of distributed processing in Oracle database systems:

The client and server are located on different computers; these computers are connected via a network (see Figure appendix. 1, Part A).

A single computer has more than one processor, and different processors separate the execution of the client application from Oracle (see appendix. 1, Part B).

Benefits of the Oracle client/server architecture in a distributed processing environment include the following:

Client applications are not responsible for performing any data processing. Client applications can concentrate on requesting input from users, requesting desired data from the server, and then analyzing and presenting this data using the display capabilities of the client workstation or the terminal (for example, using graphics or spreadsheets).

Client applications can be designed with no dependence on the physical location of the data. If the data is moved or distributed to other database servers, the application continues to function with little or no modification.

Oracle exploits the multitasking and shared-memory facilities of its underlying operating system. As a result, it delivers the highest possible degree of concurrency, data integrity, and performance to its client applications.

Figure appendix 1. . The Client/Server Architecture and Distributed Processing

Client workstations or terminals can be optimized for the presentation of data (for example, by providing graphics and mouse support) and the server can be optimized for the processing and storage of data (for example, by having large amounts of memory and disk space).

If necessary, Oracle can be scaled. As your system grows, you can add multiple servers to distribute the database processing load throughout the network (horizontally scaled). Alternatively, you can replace Oracle on a less powerful computer, such as a microcomputer, with Oracle running on a minicomputer or mainframe, to take advantage of a larger system's performance (vertically scaled). In either case, all data and applications are maintained with little or no modification, since Oracle is portable between systems.

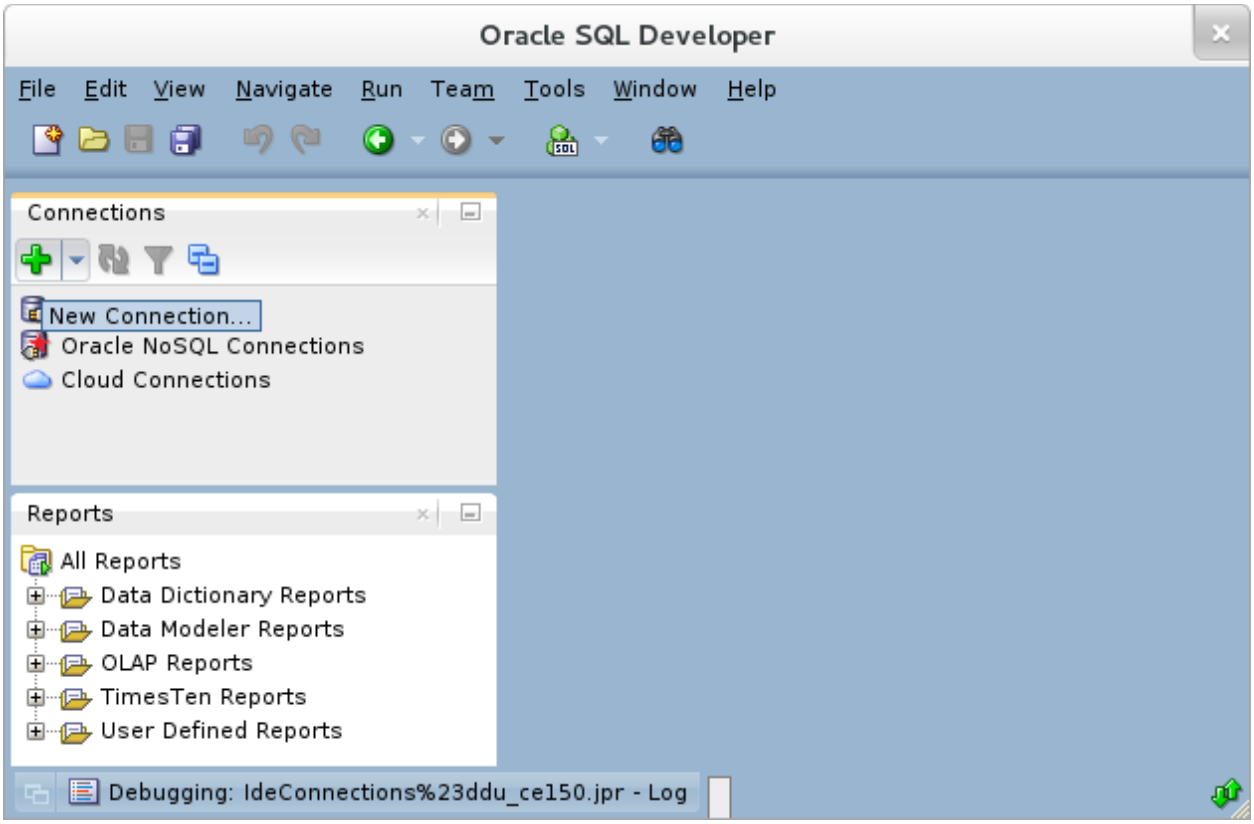
In networked environments, shared data is stored on the servers, rather than on all computers in the system. This makes it easier and more efficient to manage concurrent access.

In networked environments, inexpensive, low-end client workstations can be used to access the remote data of the server effectively.

In networked environments, client applications submit database requests to the server using SQL statements. Once received, the SQL statement is processed by the server, and the results are returned to the client application. Network traffic is kept to a minimum because only the requests and the results are shipped over the network.

. How to use SQLDeveloper – For Beginners

-> Click on the green plus sign from left panel to create a new profile of database connection:

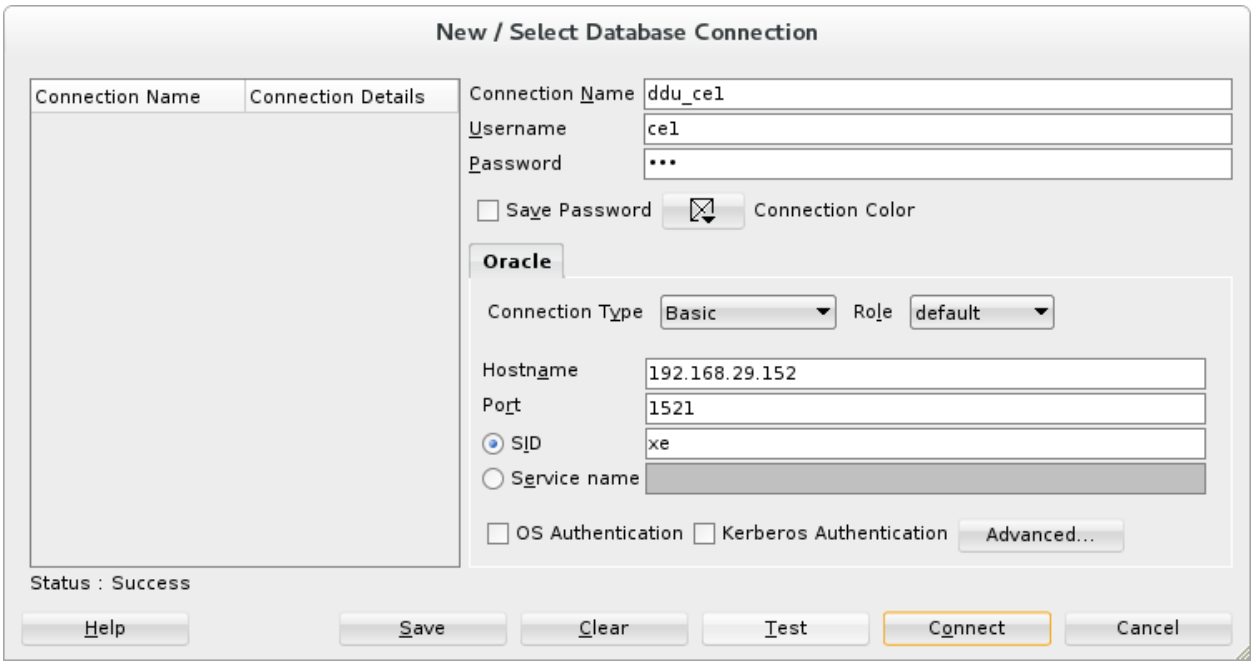


-> Fill the connection details as provided by the lab in-charge:

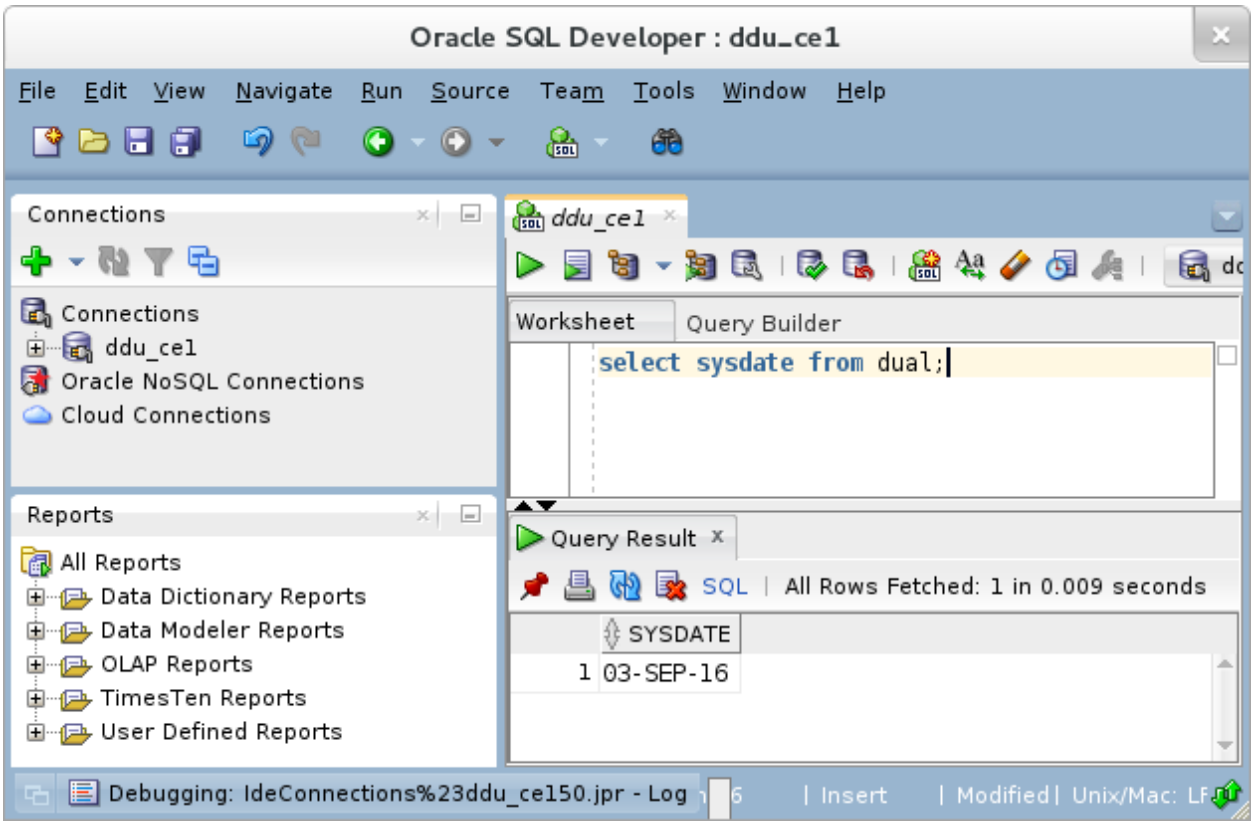
Connection Name – However you will want to identify this database connection for the future use.

UserName and Password of your database user
HostName and other details of the database server

Click on Save. Then click on Test and see the status shall read “Status:success”. Connect now.



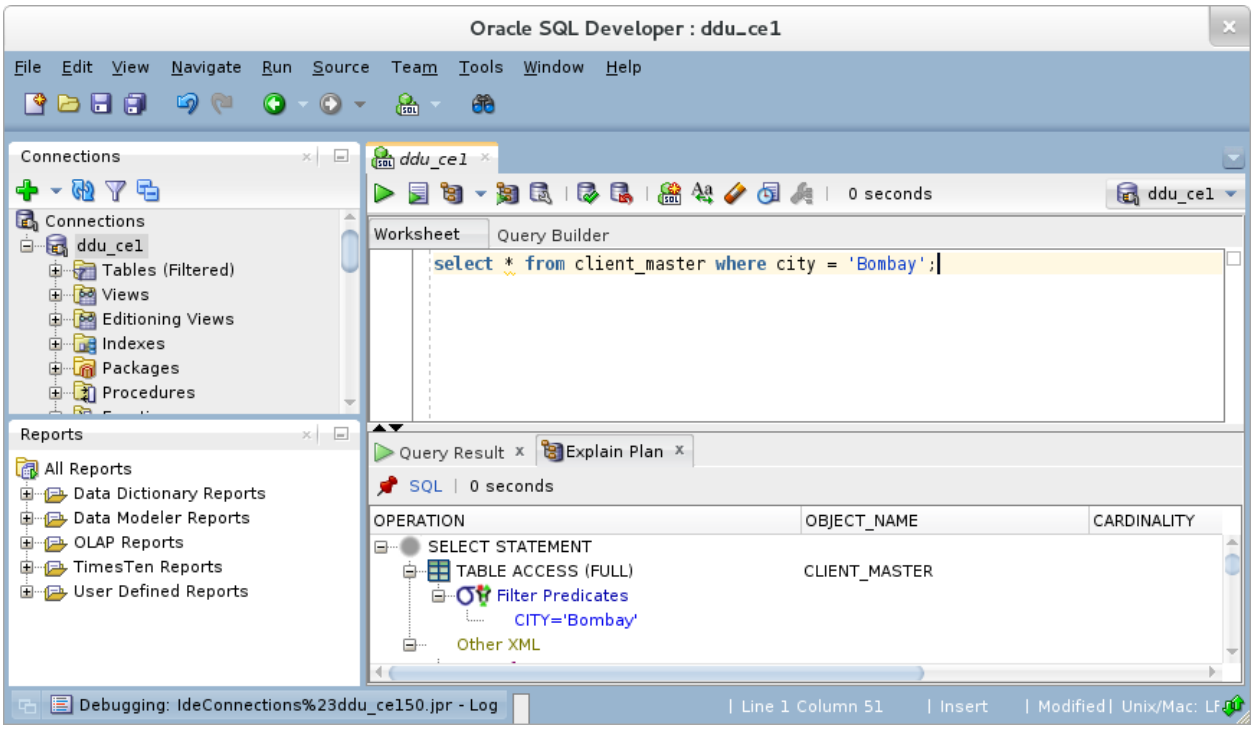
-> After connection establishes:
Left panel shows connection opened to your database with database objects to access.
Center panel provides you worksheet to type in and test SQL statements.
You may use Ctrl+Enter to run the individual SQL statement where currently cursor is. Or click on the Green play button.



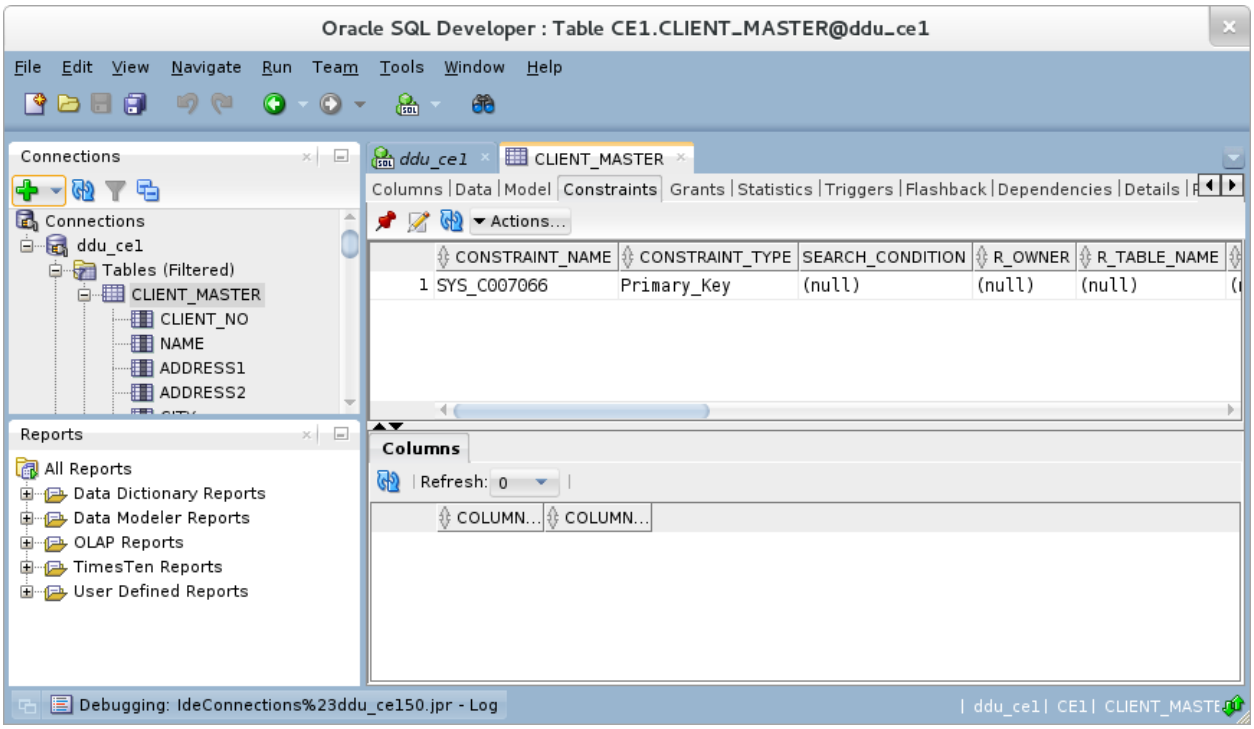
-> Know that for the same database, you may open multiple worksheet the way you open multiple tabs in firefox, etc.

Right Click on the connection plug and choose “Open SQL Worksheet”.

-> For any query to learn the “Explain Plan” which shows estimated cost operations and how database management system is planning to execute your query.



-> You may learn about any objects including tables, views, constraints, etc.



-> Also, explore more features like export data into csv or spreadsheet formats. How to import database dump or

csv data file. Also, if there are sql files containing create like ddl and insert like dml then open it from file->open and then hit F5 to run as a script to have table(s) created and data inserted.