

**LAB MANUAL**

**FOR**

**DATABASE MANAGEMENT**

**SYSTEM**

JNTU World

**CSE- 216 E****Database Management Systems Lab**

L     T     P  
 -     -     2

Class Work: 25  
 Exam: 25  
 Total: 50  
 Duration of Exam: 3

Hrs.

I. Create a database and write the programs to carry out the following operation :

1. Add a record in the database
2. Delete a record in the database
3. Modify the record in the database
4. Generate queries
5. Generate the report
6. List all the records of database in ascending order.

II Develop a menu driven project for management of database system:

1. Library information system
  - (a) Engineering
  - (b) MCA
2. Inventory control system
  - (c) Computer Lab
  - (d) College Store
3. Student information system
  - (e) Academic
  - (f) Finance
4. Time table development system
  - (g) CSE, IT & MCA Departments
  - (h) Electrical & Mechanical Departments

Usage of S/w:

1. VB, ORACLE and/or DB2
2. VB, MSACCESS
3. ORACLE, D2K
4. VB, MS SQL SERVER 2000

**Note: At least 5 to 10 more exercises to be given by the teacher concerned.**

## **Rational behind DBMS Lab**

Database management has evolved from a specialized computer application to a central component of a modern computing environment and as a result knowledge about database system has become an essential part of computer science. The course serves as a visual guide to the material presented during our lectures. The aim of this course is to provide an introduction to Database management system, with an emphasis on foundational material

The fundamental concepts and algorithms covered are based on those used in existing commercial or experimental database systems. Our aim is to present these concepts and algorithms in general setting.

### **Objectives**

At the end of the course students should

- 1 have a good understanding of how several fundamental algorithms work, particularly those concerned with creation and updating of tables.
- 2 have a good understanding of the fundamental DBMS used in computer science
- 3 be able to understand various queries and their execution.
- 4 be able to design new database and modify existing ones for new applications and reason about the efficiency of the result

## **Software and Hardware Requirements**

### **Software Required:**

1. VB, ORACLE and/or DB2
2. VB, MSACCESS
3. ORACLE, D2K
4. VB, MS SQL SERVER 2000

### **Hardware Required:**

Processor : Pentium III  
RAM : 128 MB  
Hard Disk : 40 GB

## **Theory and Concept** **Practical #1**

**Objective:** Create tables and specify the Questionnaires in SQL.

### **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 Query 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).

### **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	vchar2	6
name	vchar2	20
address1	vchar2	30
address2	vchar2	30
city	vchar2	15
state	vchar2	15
pincode	number	6
bal_due	number	10,2

ii) **Product\_master**

Columnname	datatype	size
Product_no	vchar2	
Description	vchar2	
Profit_percent	number	
Unit_measure	vchar2	
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		2000
0006	Rukmini	Bombay	400050	Maharashtra	0



**Data for Product Master:**

Product No.	Description	Profit %	Unit	Qty	Reorder	Sell	Cost
	price	Percent		measured	on hand	lvl	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 Questionries:**

- 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 Manipulate 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 = expression, columnname =expression.....

Where columnname = expression;

**Deletion Operation:-**

A delete query is expressed in much the same way as Query. 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

\* multiplication

/ Division

- Subtraction

\*\* exponentiation

() 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 sentences 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.

Examples:

Avg                    return average value of n

### Syntax:

Avg ([distinct/all]n)

Min                    return minimum value of expr.

### Syntax:

MIN((distinct/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 Questionnaires.**

- 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 title 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 constraints 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 lacks 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 datatype (size) not null .....)

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

**Syntax: primary key as a column constraint**

Create table tablename

(columnname datatype (size) primary key,....)

**Primary key as a table constraint**

Create table tablename

(columnname datatype (size), columnname datatype( size)...

Primary key (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 will 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 integrity 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 client\_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. Sales\_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	6	Primary key reference clientno of client_master table
Client_no	Varchar2	25	
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 less than 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	Rem
500001	Kiran	A/14 worli	Bom bay	400002	Mah	3000	100	50	Good
500002	Manish	65,nariman	Bom bay	400001	Mah	3000	200	100	Good
500003	Ravi	P-7 Bandra	Bom bay	400032	Mah	3000	200	100	Good
500004	Ashish	A/5 Juhu	Bom bay	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	Orderstatus
019001	12-jan-96	0001	F	N	50001	20-jan-96	Ip
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 .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 tablename;

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;
```

Dropping integrity constraints in the ALTER TABLE command:

You can drop an integrity constraint if the rule that it 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 dropping 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;
```

## Assignment No.4

Question 1. Create the following tables:

### 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

Q2. Insert the following values into the challan header and challan\_details tables:

(i)	Challan No	S_order No	Challan Date	Billed
	CH9001	019001	12-DEC-95	Y
	CH865	046865	12-NOV-95	Y
	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

**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 same 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 with nulls in the values of attributes belonging to the 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 Queries:

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 Query, 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 Queries:

**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 SubQueries.**

**SubQueries:-** A subQuery is a form of an SQL statement that appears inside another SQL statement. It also termed as nested Query. The statement containing a subQuery called a parent statement. The rows returned by the subQuery are used 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 Queries and combine their output using the union clause . The union clause merges the output of two or more Queries into a single set of rows and column. The final output of union clause will be

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

Syntax:

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

**Intersect Clause:** The use can put together multiple Queries 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 Queries

Syntax:

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

**MINUS CLAUSE:-** The user can put together multiple Queries and combine their output = records only in Query one

Syntax:

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

## Assignment NO.7

**Objective:** Answer the following Queries:

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 Unique Index:-

```
CREATE UNIQUE 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 for the following reasons:

1. The DBA stores the views as a definition only. Hence there is no duplication of data.
2. Simplifies Queries.
3. Can be Queried 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 No # 8**

### **Objective : Answer the following Questions**

- 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 # 9

#### 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, JMBCCodeandcosts, 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
  - JMBCCodeandcosts – 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.

## References

- Database System Concepts by A. Silberschatz, H.F. Korth and S. Sudarshan, 3<sup>rd</sup> edition, 1997, McGraw-Hill, International Edition.
- PL/SQL By Ivan Bayross
- Introduction to Database Management system by Bipin Desai, 1991, Galgotia Pub.
- Fundamentals of Database Systems by R. Elmasri and S.B. Navathe, 3<sup>rd</sup> edition, 2000, Addison-Wesley, Low Priced Edition.
- An Introduction to Database Systems by C.J. Date, 7<sup>th</sup> edition, Addison-Wesley, Low Priced Edition, 2000.
- Database Management and Design by G.W. Hansen and J.V. Hansen, 2<sup>nd</sup> edition, 1999, Prentice-Hall of India, Eastern Economy Edition.
- Database Management Systems by A.K. Majumdar and P. Bhattacharyya, 5<sup>th</sup> edition, 1999, Tata McGraw-Hill Publishing.
- A Guide to the SQL Standard, Date, C. and Darwen, H. 3<sup>rd</sup> edition, Reading, MA: 1994, Addison-Wesley.



# New Practicals besides University syllabus

## Theory and Concept Practical No. 1

### Objective:- To implement the basics of PL/SQL.

**Introduction** – PL/SQL bridges the gap between database technology and procedural programming languages. It can be thought of as a development tool that extends the facilities of Oracle's SQL database language. Via PL/SQL you can insert, delete, update and retrieve table data as well as writing loops or branching to another block of code.

### PL/SQL Block structure-

DECLARE

Declarations of memory variables used later

BEGIN

SQL executable statements for manipulating table data.

EXCEPTIONS

SQL and/or PL.SQL code to handle errors.

END;

**Displaying user Messages on the screen** – Any programming tool requires a method through which messages can be displayed to the user.

**dbms\_output** is a package that includes a number of procedure and functions that accumulate information in a buffer so that it can be retrieved later. These functions can also be used to display message to the user.

**put\_line**: put a piece of information in the buffer followed by a end of line marker. It can also be used to display message to the user.

Setting the server output on:

### SET SERVER OUTPUT ON:

Example: Write the following code in the PL/SQL block to display message to user  
DBMS\_OUTPUT.PUT\_LINE('Display user message');

### Conditional control in PL/SQL-

Syntax:

IF <condition> THEN

<Action>

ELSEIF<condition>

<Action>

```
ELSE  
<Action>  
ENDIF;
```

### **The WHILE LOOP:**

#### Syntax:

```
WHILE <condition>  
LOOP  
<Action>  
END LOOP;
```

### **The FOR LOOP statement:**

#### Syntax:

```
FOR variable IN [REVERSE] start—end  
LOOP  
<Action>  
END LOOP;
```

**The GOTO statement:** The goto statement allows you to change the flow of control within a PL/SQL Block.

## Assignment No.1

Q1. WAP in PL/SQL for addition of two numbers.

Q2. WAP in PL/SQL for addition of 1 to 100 numbers.

Q3. WAP in PL/SQL to check the given number is even or odd.

Q4. WAP in PL/SQL to inverse a number, eg. Number 5639 when inverted must be display  
output 9365.

Q5. WAP in PL/SQL for changing the price of product 'P00001' to 4000 if the price is less than  
4000 in product\_master table. The change is recorded in the old\_price\_table along  
with  
product\_no and the date on which the price was changed last.

## Theory and Concept Practical No. 2

**Objective:- To implement the concept of Cursor and Trigger.**

**Cursor**– We have seen how oracle executes an SQL statement. Oracle DBA uses a work area for its internal processing. This work area is private to SQL's operation and is called a **cursor**.

The data that is stored in the cursor is called the **Active Data set**. The size of the cursor in memory is the size required to hold the number of rows in the Active Data Set.

**Explicit Cursor**- You can explicitly declare a cursor to process the rows individually. A cursor declared by the user is called **Explicit Cursor**. For Queries that return more than one row, You must declare a cursor explicitly.

The data that is stored in the cursor is called the **Active Data set**. The size of the cursor in memory is the size required to hold the number of rows in the Active

**Why use an Explicit Cursor**- Cursor can be used when the user wants to process data one row at a time.

**Explicit Cursor Management**- The steps involved in declaring a cursor and manipulating data in the active data set are:-

- Declare a cursor that specifies the SQL select statement that you want to process.
- Open the Cursor.
- Fetch the data from the cursor one row at a time.
- Close the cursor.

**Explicit Cursor Attributes**- Oracle provides certain attributes/ cursor variables to control the execution of the cursor. Whenever any cursor(explicit or implicit) is opened and used Oracle creates a set of four system variables via which Oracle keeps track of the 'Current' status of the cursor. You

- Declare a cursor that specifies the SQL select statement that you want to process.
- Open the Cursor.
- Fetch the data from the cursor one row at a time.
- Close the cursor.

**How to Declare the Cursor:-**

The General Syntax to create any particular cursor is as follows:-

**Cursor <Cursorname> is Sql Statement;**

**How to Open the Cursor:-**

The General Syntax to Open any particular cursor is as follows:-

**Open Cursorname;**

### **Fetching a record From the Cursor:-**

The fetch statement retrieves the rows from the active set to the variables one at a time. Each time a fetch is executed. The focus of the DBA cursor advances to the next row in the Active set.

One can make use of any loop structure(Loop-End Loop along with While,For) to fetch the records from the cursor into variable one row at a time.

The General Syntax to Fetch the records from the cursor is as follows:-

**Fetch cursorname into variable1,variable2,\_\_\_\_\_**

### **Closing a Cursor:-**

The General Syntax to Close the cursor is as follows:-

**Close <cursorname>;**

### **Database Triggers:-**

Database triggers are procedures that are stored in the database and are implicitly executed(fired) when the contents of a table are changed.

### **Use of Database Triggers:-**

Database triggers support Oracle to provide a highly customized database management system. Some of the uses to which the database triggers can be put to customize management information in Oracle are as follows:-

- A Trigger can permit DML statements against a table any if they are issued, during regular business hours or on predetermined weekdays.
- A trigger can also be used to keep an audit trail of a table along with the operation performed and the time on which the operation was performed.
- It can be used to prevent invalid transactions.
- Enforce complex security authorizations.

### **How to apply DataBase Triggers:-**

A trigger has three basic parts:-

1. A triggering event or statement.
2. A trigger restriction
3. A trigger action.

### **Types of Triggers:-**

Using the various options , four types of triggers can be created:-

1. **Before Statement Trigger:-** Before executing the triggering statement, the trigger action is executed.
2. **Before Row Trigger:-** Before modifying the each row affected by the triggering statement and before appropriate integrity constraints, the trigger is executed if the trigger restriction either evaluated to TRUE or was not included.'

3. **After Statement Trigger:-** After executing the triggering statement and applying any deferred integrity constraints, the trigger action is executed.
4. **After row Trigger:-** After modifying each row affected by the triggering statement and possibly applying appropriate integrity constraints, the trigger action is executed for the current row if the trigger restriction either evaluates to TRUE or was not included.

### **Syntax For Creating Trigger:-**

The syntax for Creating the Trigger is as follows:-

```
Create or replace Trigger<Triggername> {Before,After} {Delete, Insert, Update } On
<Tablename> For Each row when Condition
Declare
<Variable declarations>;
<Constant Declarations>;
Begin
<PL/SQL> Subprogram Body;
Exception
Exception PL/SQL block;
End;
```

### **How to Delete a Trigger:-**

The syntax for Deleting the Trigger is as follows:-

```
Drop Trigger <Triggername>;
```

## Frequently Asked Questions

- What is database?
- What is DBMS?
- What is a Database system?
- What are the Advantages and disadvantages of DBMS?
- What is a view? How it is related to data independence?
- What is E-R model?
- What is an Entity?
- What is an Entity set?
- What is DDL (Data Definition Language)?
- What is DML (Data Manipulation Language)?
- What is a query?
- What do you mean by Correlated subquery?
- What are the unary operations in Relational Algebra?
- Are the resulting relations of PRODUCT and JOIN operation the same?
- Define SQL and state the differences between SQL and other conventional programming Languages
- What is database Trigger?
- What are cursors give different types of cursors?
- What operator performs pattern matching?
- What operator tests column for the absence of data?
- Which command executes the contents of a specified file?
- What are the wildcards used for pattern matching?
- What are the privileges that can be granted on a table by a user to others?
- What command is used to get back the privileges offered by the GRANT command?

- Which system table contains information on constraints on all the tables created?
- What is the difference between TRUNCATE and DELETE commands?
- What will be the output of the following query? `SELECT DECODE(TRANSLATE('A','1234567890','1111111111'), '1','YES', 'NO' );`
- What does the following query do? `SELECT SAL + NVL(COMM,0) FROM EMP;`
- Which date function is used to find the difference between two dates?
- Which function is used to find the largest integer less than or equal to a specific value?
- What is the use of CASCADE CONSTRAINTS?
- What is the use of DESC in SQL?

JNTU World