UNIVERSITY INSTITUTE OF TECHNOLOGY
BARKATULLAH , BHOPAL (M.P)

ASSIGNMENT

SUBJECT :- DBMS

SUBMITTED BY :- RUCHIR DONGRE

BATCH :- COMPUTER SCIENCE

SUBMITTED TO :- NEHA LIDORIYA

Certificate

Name: Ruchir Dongre. Class:

Roll No: Exam No:

Institution _____

This is certified to be the bonafide work of the student in the _____ *Laboratory during the academic year 20 / 20 .*

No. of practicals certified _____ *out of* _____ *in the subject of* _____

.....
Teacher In-charge

.....
Examiner's Signature

.....
Principal

Date: .....

Institution Rubber Stamp

(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)

Index

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
8.	To implement the Concept of indexes and ~~Views~~ View Views				
9.	To implement the basics of PL/SQL.				
10.	To implement the Concept of Cursor and Trigger				

Expt. No. _____ Date _____

Page No. _____

Experiment - 1

Objective :- Create tables and specify the Queries in SQL

Theory and Concepts

Introduction about SQL

SQL (Structured Questionary language) is a nonprocedual language, you specify what you want, not how to get it. A block structured formate of English key words is used in this Questionary language. It has the following Componets

DDL (Data Definition language)-

The SQL DDL provides Command for defining relation schemas, deleting relations and modefying relation ~~ster~~ 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 ~~Comm~~ 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 ~~sto~~ Statements Can be ~~embedd~~ embedded with in general ~~Purpose~~ Purpose

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Programming languages, such as C, C++, Java, COBOL, Pascal and ~~Forte~~ Fortran.

Integrity-

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

Authorization- .

The ~~SQL~~ SQL DDL ~~includes incut incu incla~~ includes commands for specifying access rights to relations and views.

Data Definition Language-

The SQL DDL allows specification of not only a sets of relations but also information about each relation, ~~including~~ 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~~ The physical storage structure of each ~~rela~~ relation on disk.

~~Do~~ Domain types in SQL-

Domain types in SQL

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

Teacher's Signature _____

Date

Expt. No. Page No.

- 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.
- 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 -

i) Creating a table :-

Syntax :-

```
CREATE TABLE TABLENAME
(Column1, Column2, Co..... Columnn);
```

Eg:-

```
CREATE TABLE Employee (
    Emp Id int, Emp Name Varchar2(50),
    Addres Varchar2(100), City Varchar2(20),
    Contact_No int);
```

Teacher's Signature

i) Creating a table:-

Table Created.

Name	NULL?	Type
EMP_ID		~~NUM~~ NUMBER(38)
EMP_NAME		VARCHAR2(50)
ADDRESS		VARCHAR2(100)
CITY		VARCHAR2(100)
CONTACT_NO		NUMBER(70)

ii) Insertion of data:-

EMP_ID	EMP_NAME	ADDRESS	CITY	CONTACT NO
1	Daniel	Standford Street	Bhopal	941 xxxx
2	David	Gandhi Nagar	~~Dettri~~ Delhi ~~Dettri~~	854xxxx

iii) Insertion ~~of~~ of Selected data into a table from another ~~tab~~ table:-

1 row inserted.

iv) Retrieving data from the table:-

EMP_ID	EMP_NAME	ADDRESS	CITY	CONTACT NO
1	Daniel	Standford street	Bhopal	941xxxx
2	David	~~@Gandhi Nagar~~ Gandhi Nagar	Delhi	~~854xxxx~~ 854xxxx

Expt. No. _____ Date _____

Page No. _____

DESC Employee:

ii) Insertion of data:-

Syntax:-

INSERT INTO tablename (Col1, Col2 .. Coln)
VALUES (value1, value 2, ... Valuen);

Or

INSERT INTO tablename
VALUES (value1, value 2, ... Valuen);

eg:- INSERT INTO Employee VALUES (1, 'Sam Daniel', 'Stanford Street', 'Bhopal', 941xxxx);
INSERT INTO Employee VALUES (2, 'David', 'Gandhi Nagar', 'Delhi', 8854xxxx');
Select * Employee;

iii) Insertion of selection Selected data into a table from another table;

Syntax:-

INSERT INTO tablename SELECT [Col1, Col2 ....]
FROM tablename
WHERE Condition;

eg:- Insert into Employee Select * from Employee
Where Emp Id = 1;

iv) Retrieving data from the table:-

Syntax:- SELECT * FROM tablename;

eg:- SELECT * FROM Employee;

Teacher's Signature _____

v) EMP_ID EMP_NAME

1 Daniel

2 David

vi) CITY CITY

~~Bhopal~~ / Bhopal Bhopal

Delhi Delhi

vii) EMP_ID CITY

1 Bhopal

viii) Table altered;

Table altered;

Table altered;

v) Retrieving of specific Columns from a table :-
Syntax :- SELECT Col1, Col2 .... FROM tablename
eg:- SELECT Emp-Id, Emp Name FROM Employee;

vi) Elimination of duplicates from the Select Statement :-
Syntax :- SELECT DISTINCT Col1, Col2 .... FROM tablename;
eg:- SELECT DISTINCT City FROM Employee.;

vii) Selecting a ~~data~~ dataset from table data :-
Syntax :- SELECT Col1, Col2 .... FROM Tablename
WHERE Condition ;
eg:- SELECT Emp-Id, City FROM Employee
WHERE Emp Name = 'Daniel';

viii) Alter :- Syntax :-
ALTER TABLE tablename DROP ~~CO~~COLUMN ~~Colu~~ Colname;
eg:- ALTER TABLE Employee DROP COLUMN ~~Contact-No~~;
Contact-No;

ix) ALTER TABLE tablename ADD Col_name Col_type;
eg:- ALTER TABLE Employee ADD Email Varchar (50);

x) ALTER TABLE tablename MODIFY COLUMN Colname type;
eg:- ALTER TABLE Employee
MODIFY COLUMN Emp Name Varchar (50);

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Experiment-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 table without Changing all Values in the tuple. For this purpose the update statement can be used.

Update table name
Set Columnname = expression expression, Columnname = expression....
Where Columnname = expression;

Deletion Operation :-

A delet requestionst is expressed in much the same way as Questionary. We can delete whole tuple (rows) We can delete values on only Particulors attributes.

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Deletion of all rows. & rows.

Syntax:

Delete. from tablenamc:

Deletron. of specified. number. of rows.

Syntax:

Delete from table name.

Where. Seo Search Condition;

Com Computation in. expression. lists. Used. to. Select data.

+. Addition -Subtraction

* multiplication ** exponentration.

~~÷ Division~~ ~~() Enclosed operation.~~

/ Division () Enclosed operation.

Renaming. Columns. Used. With Expression Lists :- The default. output Column. names. Can be. renamed. by the User if required.

Syntax:

Select. ~~Colum~~ Column. name

Columnname

From table name;

Logical ~~Operation~~ Operators:

The logical operators. that Can be. Used in SQL. Sentenced are

AND all of must. be included

OR ony of may be included

NOT none of. Could Could be included

Teacher's Signature _____

Products

Product-ID	ProductName	Price
1	P1	18
2	P2	19
3	P3	10
4	P4	22
5	P5	20

Demo

Product ID	Product Name	Supplier ID	~~Price~~ Category ID	Price
1	P1	1	1	18
2	P2	1	1	19
3	P3	1	2	10
4	P4	2	2	22
5	P5	2	2	21.35

Expt. No. _____ Date _____

Page No. _____

Range Srara 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 ~~Sat Substr~~ ~~Substring~~ / ~~Substring~~ Substring. We Consider the following examples.
- 'Perry %' matches any String ~~beginning~~ beginning with perry
- '%.rdge'.% matches any String Containing 'rdeg as Substring.
- '___' matches any String exactly three Characters.
- '___%.% matches any String of at least of three Characters.

Or Oracle ~~functions~~

Oracle ~~Function~~ Oracle Function:

Functions are Used to ~~manu~~ manipulate data items and return result. function follow the format of function_name (argument 1, argument 2,...). An arrangement is User defined Variable or Constant. The Structure of function is Such that it ~~acc~~ ~~acc~~ accepts Zero or more arguments.

~~Examples~~ Examples:

Examples:

Avg return average Value of n

Teacher's Signature _____

Customers

Customerid	Customer Name	Contact Name	Address	City	Country
1	A1	B1	obere Str. 57	Berlin	Germany
2	A2	B2	Avda-dela Constitucion 2222	~~Mexico~~ Mexico	Mexico
3	A3	B3	Mataderos 2312	Mexico	Mexico
4	A4	B4	120 Hanover Sq.	London	UK
5	A5	B5	~~Bergovsvagen~~ A50, Street, Berlin	Berlin	Germany.

i) Customers

Customer id	Customer	Contact Name	Address	City	Country
1	A1	B6	~~Frankfurt~~ obere Str. 57	Frankfurt	Germany
2	A2	B2	Avda-dela Constitucion 2222	Mexico	Mexico
3	A3	B3	Mataderos 2312	Mexico	Mexico
4	A4	B4	120 Hanover Sq.	London	UK
5	A5	B5	A50, Street Berlin	Berlin	Germany

ii) Customers

iii) Customers

Customer id	Customer	Contact Name	Address	City	Country
1	A1	B6	obere Str. 57	Frankfurt	Germany

Expt. No. _____ Date _____

Page No. _____

Syntax:

Avg. ([distinct/all] expr)

Count Count. Returns minimum Value of expr.

Syntax:

MIN. ((distinct/all) expr))

Count Returns 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 Returns. max. Value. of expr

Syntax:

Max ([distinct/all] Max ([distinct/all] n)

Max ([distinct/all] expr)

Sum Returns. Sum of. Values. of n

Syntax:

Sum ([distinct/all] n)

Sorting of data in table table

Syntax:

Select. Columnname, Columnname

From table

Order by Columnname;

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

i) Update Comand

Update ~~Customers~~ Coustomers Customers

Set . ContactName = 'B~~a~~', City = '~~Mexico~~' 'Frankfurt'

Where Customer ID = 1;

ii) Delete From ~~Co~~.Customers

iii) Delete From Customers Where Customer Id = 2, Coustomer Id = 3,

Customer Id = 4, . Customer Id = 5;

iv) Select . Min (Price) As. Smallest Price

From products;

v) Select Max(Price) As Largest Price

From products;

Price.

vi) Select. Count (~~Product ID~~)

From products;

vii) Select Avg. (Price)

From products;

viii) Select. Sum (Price)

From products;

Teacher's Signature _____

vi) ~~10~~

vii) 28.86

viii) 89

Expt. No. _____ Date _____

Page No. _____

Experiment-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 Constraints

ii) Table level Constraints: If the data Constraints 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~~ said to be null. Column of any data type may contain null values Unless the Column was defined as not null when the table was ~~for~~ Created

Syntax:

Create table tablename
(Columnname data type (size) not null....)

eg:- Create table Person
(~~Firsts~~ ~~Firste~~ FirstName VarChar (~~255~~ (255) not null);

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Primary Key :- Primary Key is one more Columns. 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: primary key as a Column Constraint

```
Create table tablename,
(Columnname datatype (Size) primary key ....)
```

eg:-

```
Create table Persons(
    ID int NOT NULL PRIMARY KEY KEY,
    LastName Varchar (255) NOT NULL,
    First Name Varchar (255),
    Age int
);
```

Primary key as a table Constraint.

```
Create table tablename
(Columnname datatype (Size), Columnname datatype (Size)....
primary key (Columnaname, Columnname));
```

eg:-

```
Create Table Persons(
    ID int NOT NULL,
    LastName Varchar (255) NOT NULL,
    First Name Varchar (255),
    Age int,
    Primary key (ID)
);
```

Teacher's Signature _____

Persons

ID	~~First Name~~	~~First Name~~	First Name	Last Name	City
1			F1	L1	C1
2			F2	L2	C2
3			F3	L3	C3
4			F4	L4	C4

i) Here ID is assigned as primary key

ii) Here ID and Last Name. ~~Columns~~. Attributes do not contain any ~~reo~~ repeating values.

iii) The Default City for all ~~Peo~~ ~~Pers~~ People is ~~Mexico~~ 'Sandnes'

Expt. No. _____ Date _____

Page No. _____

Uni Question key Concept:- A Uniquestion is similar to a primary key except that the purpose of a Uni Uni Question 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 Uni Question keys.

Syntax: Uni Question as a Column Constraint.

Create table tablename.
(Columname datatype (Size) Uni Question);

Uniquestion as table Constraint:

Create table tablename.
(Columnname datatype (size) Columnname datatype (size)
.... Uni. Question.
(Columnname, Columnname));

Data eg:- Create Table Persons (
Id ID int NOT NULL,
LastName Varchar (255) NOT NULL,
FirstName Varchar (255),
Age int,
Constraint UC_Person Unique (ID, LastName)
);

Default Value Concept:- At the time 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

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Value specified. The data type of the default value. Value should match the data type of the column.

Syntax:

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

iii) Eg:- Create table Persons (
ID int not null
Last Name Varchar (255) NOTNULL,
First Name Varchar (255),
Age int,
City Varchar (255) DEFAULT 'Sandnes' ('Mexico' 'Sandnes'
);

iv) Foreign Key Concept:- Foreign Key represents relationship between tables.. A foreign key is column whose value are derived from the primary key of the same of some other table. the existence of foregin key implies that the table with forign key is related to the primary key table from which the foreign key is derived

Syntax:

Create table tablename
(Columnname datatype (Size) references another tablename);

Teacher's Signature _____

Expt. No. Date

Page No.

Syntax:
Create tablename
(Columnname datatype (Size) ...
Primary key (Columnname);
foreign key (Columnnname) references tablename);

eg:- Create Persons. Student.
(Columnname Salary Int (50)
Primary Key (ID)
foreign key City references Persons);

v) Check Integrity Constraints: Use the Check Constraints when you need to enforce integrity.
- A Check Constraints name Column of the Corent master d. So that the name is entered. In upper case.
- A Check Constraint on the Client no. Column of the Client. Client_master so that no. Client. Client no value starts with 'C'

Syntax:
Create table tablename
(Columnname datatype (size) CONSTRAINT Constraintname)
Check (expression));

eg:- Create table Student.
(ID Int (50) CONSTRAINT (NOT NULL)
Check (City = C1));

Teacher's Signature

vi) Student

ID	Salary	City	~~Count~~ Grade
1	50,000	C1	A1
2	50,000	C2	A2
3	40;000	C3	A3
4	40,000	C4	A4

~~Stude~~

v)

ID	~~Salary~~
1	

Expt. No. _____ Date _____

Page No. _____

Experiment -4

objective :- To Implement the Structure of the table.

Modifying the Structure of Tables :- After table Command is Used to Changing the Structure of a table. Using the after table clause you Cannot perform the ~~following~~ 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));

eg :- Alter table persons
Add (new Columnname Id new Int.);

ii) Modifying existing table

Syntax

Alter table tablename
Modify (newColumnname newdatatype(size));

Teacher's Signature _____

Persons.

S.NO	First name	Lastname
1	Sham	das
2	Ram	das

i)

S.No	Firtname	Lastname	Id
1	Sham	das	1
2	Ram	das	2

ii

Expt. No. _____ Date _____

Page No. _____

Removing / Deleating Tables - following Command is used for for removing or deleting a table.

Syntax:

DR Drop Table tablename;

eg:- Drop Persons

The following examples show the definition of Several integ integrity Constraint Constraints.

i) Add Primary key:

Syntax:

Alter table tablename

Add Primary key (Columnname);

eg:- Alter table Persons

Add Primary key (Id);

ii) Add Foreign key:-

Syntax

Alter table tablename

Add Constraint Constraintname

Foreign Key (Columnname) References tablename;

eg:- Alter table Persons

Add Constraint Not Null

Foreign Key (Id) References Details;

Teacher's Signature _____

Details

Id	Grade	Salery
1	A1	50,000.
2	A2	46,000

ii) Persons

S.No	Firstname	Lastname	Id
1	Sham	Das	1
2	Ram	Das	2.

Here. Primary key is Id

iii) Persons.

S.No	Firstname	Lastname	Id
1	Shom	Das	1
2	Rom	Das	2

Here Id ~~Can~~ r Connot have null value.

Expt. No. _____ Date _____

Page No. _____

You can drop an integrity constraints if the rule that it enforces is no longer true or if the constraint is no longer needed.

i) Drop the primary key:-

Syntax:

Alter table tablename
Drop primary key

eg:- Alter table Persons
Drop primary key

ii) Drop foreign key:-

Syntax:

Alter table tablename
Drop constraint Constraintname

eg:- Alter table Persons
Drop constraint not null

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Example -5

objective - To implement the Concept of Joins

~~Join~~ ~~Join~~ ~~To~~ 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 seperate object but one single entity. To archieve this we ~~ha~~ have to join ~~to~~ tables ~~Tables~~ Tables are jointed on Column that have dome data type and data with in table. The tables that have to be joined are specified in the ~~for~~ 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~~ Project Column Specified in the SELECT Clause.

1. Cartesian Product:-

Consider two table student and Course

```
Select B.*,P.*
From Student B, Course P
~~WHERE~~ Where B.Course#=P.Course#;
From Student B, Course P;
```

Teacher's Signature _____

Persons

S.No	First Name	LastName	Id
1	Sham	Das	1
2	Ram	Das	2

~~Details~~

Details

Id	Grade	Salery
1	A1	50,000
2	A2	46,000

1

Expt. No. Date

Page No.

eg:- Select b.*, person.*, b.*, P.*
from Student Person.B, Detail.P;

2. Inner Join:
Cartesian product followed by Selection.
eg: Select B.*, P.*.
From Student Person.B., Detail P
Where B.Person = P.Person;

3. LEFT Outer Join:
Left. Left. Outer join - Cartesian product + Selection.
but Include rows from the left. left. table which are
Unmatched pat nulls in the values of attributes.
belonging to the Second table.

Sele. Select B.*, P.*.
From Student.B left. join Course P
ON B Course # P Course #;

S.No Id Grade

eg: Select B.*, P.*
Id Grade
From Person.B left join Detail.P
ON B.Id = P.Id
ON Person.ID = Person.ID.

4. Right Outer Join:
Right Outer Join Join = Cartesian product + Selection
but include rows from right table which are
Unmatched.

Teacher's Signature

1.)

S.NO	FirstName	LastName	Id	~~Grade~~ Grade	Salery
1	Sham	Das	1	A1	50,000
2	Ram	Das	2	A2	46,000

2.)

S.NO	First Name	Last Name	Id	Grade	Salery
1	Sham	Das	1	A1	50,000
2	Ram	Das	2	A2	46,000

3)

S.NO	First Name	Last Name	Id	Grade	Salery
1	Sham	Das	1	A1	50,000
2	Ram	Das	2	A2	46,000

3)

Id	~~Grade~~ Grade
1	A1
2	A2

Date

Expt. No. _____ Page No. _____

~~Syntax~~

Syntax ~~Id Grade Persons.Id,~~ Details.Grade.

Select B.*, P.* Id

From. ~~Student~~ persons.B Right Join Detail.P Grade.

B.persons.Id = Detail.~~Gr~~Id

5. ~~Full. Outer. Jo.~~

5. Full Outer Join.

Syntax

. Select. Persons.Id, . Details. ~~Grade~~ Grade

From ~~Student~~ Person.Id .Full Outer. ~~Jo~~ ~~Join~~ Join. J Join.

Orders. ON.

Persons.Id = Detail.Id.

Teacher's Signature _____

4.

Id	Grade
1	A1
2	A2

5.

Id	Grade
1	A1
2	A2

Expt. No. Date

Page No.

Experiment - 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.

1. Syntax:

```
Select Columnname, Columnname
from tablename
group by Columnname;
```

eg:-

```
Select Count (CustomerId), Country
from Customers
group by Country
```

2. At times it is useful to state a condition that applies to groups rather than to tuples. For example we might be intrested 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.

Teacher's Signature

Expt. No. _____ Page No. _____

Experiment - 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.

1. Syntax:

```
Select Columnname, Columnname
from tablename
group by Columnname;
```

eg:-

```
Select Count (CustomerId), Country
from Customers
group by Country
```

2. At times it is useful to state a condition that applies to groups rather than to tuples. For example we might be intrested 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.

Teacher's Signature _____

Customers

Customer ID	CustomersName	Country	City
1	Alfreds Futterkiste	Germany	Berlin
2	Ana Trujillo Emparendatosy helados	Mexico	Mexico D.P
3	Antonio. Moreno Taqueria	.Mexico	Mexico D.P
4	Around the Horn.	Uk	London
5	.Berglbunds .Snabbköp	Sweden	~~Luteå~~ Luleå

1.

Expr1000	Country.
1	~~Berl~~ Germany
2	Mexico
1	Uk
1	Sweden

Customers

Customer ID	CustomersName	Country	City
1	Alfreds Futterkiste	Germany	Berlin
2	Ana Trujillo Emparendatosy helados	Mexico	Mexico D.P
3	Antonio. Moreno Taqueria	.Mexico	Mexico D.P
4	Around the Horn.	Uk	London
5	.Berglblunds .Snabbköp	Sweden	~~Luteå~~ Luleå

1.

Expr1000	Country.
1	~~Berl~~ Germany
2	Mexico
1	Uk
1	Sweden

Expt. No. _____ Date _____

Page No. _____

Group. Constructed by the GROUP BY clause. To express Such Questionary, 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.

ii) eg:- Select. CustomerID, Country.
From Customers
Group. by. Country
b Having Count(Customer. Country) = Mexico

Teacher's Signature _____

	CustomerId	~~Coun~~ ~~Country~~ Country
	2	Mexico
	3	Mexico

Expt. No. _____ Date _____

Page No. _____

Experiment - 7

Objective : To implement the Concept of Sub Questionries.

Sub Questionries :- A Sub Questionries is a form of an SQL Statement that appears inside another SQL Statement. It also termed as nested Questionary. The Statement Containing a SubQuestionary called a parent Statement. The row returned by the SubQuestionary 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.

Creating Clientmaster table from oldclient_master, table

```
Create table Client_master
As Select * from oldclient_master;
```

Using the Union, Inte. Insert. Intersect and Minus clause:

Union clause:

The User Can put together multiple Questionries and Combine their output of two or more Questionary.

Teacher's Signature _____

Person 1

Name	State / ~~City~~	Grade	Id
Ramdas	~~Mexico~~ ~~India~~ U.P	A1	1
Sham das	~~India~~ Bihar. Bhi	A2.	2

Person 2

Name	State.	Grade	Id.
Aman.	Delhi	A3	1
Raj	M.P	A4	2.

Expt. No. _____ Date _____

Page No. _____

Questionries. 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 both Questionries.

Syntax:

i) Select Columnname, Columnname.
from tablename 1,
UNION
Select Columnname, Columnname
from from tablename2;

eg: Select State City, Grade.
from person 1
UNION State
Select City, Grade
from person2;

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

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

iv) ~~Name~~ . State Grade

U.P	1
Bihar	2
Delhi	1
M.P.	2

Expt. No. Date

Page No.

ii) Syntax:

Select Columnname, Columnname,
From tablename1
Intersect
Select Columnname, Columnname
from tablename2;

eg: Insert Into Person1 (Name, State, Grade, Id)
Select Name, State, Grade, Id
from Person2

iii) Minus Clause: The User can put together multiple Questionrs Questionries and Combine their Output= records only in Questionry one.

Syntax:
Select Columnname, Columnname
from tablename;
Minus
Select Columnname, Columnname
from tablename;

Or eg: Select Name, Class
from Student;
Minus
Select Name
from Subject;

Teacher's Signature

ii)

Name	State	Grade	Id
Ram das	U.P	A1	1
Sham das	~~Bihar~~ Bihar	A2	2
Aman	Delhi	A3	1
Raj	M.P	A4	2

Student

Name	Class
Ram das	XII
Shamdas	XII

~~Teacher~~ Subject

Subject Name	Name
English	Ramdas
Maths ~~maths~~	Shamdas

iii)

Name	Class	Subject Name
Ramdas	XII	English
Shamdas	XII	~~Maths~~ ~~Maths~~ = Maths

Expt. No. _____ Date _____

Page No. _____

Experiment-8

objective :- To implement the Concept of Index 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 Singled 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 Uni-Qerestion Index :-

```
Create UniQuestion Index indexfilename
On tablename (Columnname);
```

Droping Indexes :-

An index can be dropped by using DROP INDEX

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Syntax:

Drop index. iden indexfilname.

eg:- Create Index. Student Student
ON Persons. (Name)

Create Index. Student
on Persons (Name, Class Class Class)
UniQuestion

Create Create Index. Student Create UniQuestion
on Persons (Name). Cr

Drop Index (Name).

Create UniQuestion. Index. Student
On Persons (Name)

Drop Drop. Index. (Name)
Drop Index Student.

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

View are masks placed upon tables. This allows the programm programmer to develop a method Via which we Can display predetermined data to Users according to our desire.

Views may be Created for fore f f for the following reasons:

Teacher's Signature _____

Persons

Name	Class	Subject	Id
Ram das	~~XII~~ 12	PCM	1
Shamdas	~~XII~~ 12	~~PCM~~ PCM ~~PCM~~	2

i) Name

Ram das

Sham das

Expt. No. Date

Page No.

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

Creation of Views :

ii) Syntax :-

Create View viewname as
Select Columnname, Columnname
from tablename
Where Columnname = expression list;

eg:- Create View Student1 as
Select Nam Name, Clas Name Name, Clas
from persons
Where Subject = P Clas = 12 ;

iii) Renaming the Columns of a Views :-

Syntax :-

Create View View Viewname As
Select newColumnname ....
From tablename
Where Columnname = expression list; expression
Where Columnname = Expression list;

Teacher's Signature

ii)

Name	~~Class~~	Class
Ram das		~~XII~~ 12
Sham das.		~~XII~~ 12

iii)

Id
1
2

iv)

Name	Class
Ram das	12
Sham das	12

eg:- Create View Student2 As
Select Id,
From Persons
Where Class = 12;

Selection

Selecting a data set from a view :-

Syntax:-

Select Columnname, Columnname
From viewname
Where Search Condition;

eg:- Select name, class
From Student1
Where Class = 12;

Destroying a View :-

Syntax:-

Drop View Viewname;

eg:- Drop View Student1;

Expt. No. Date

Page No.

Experiment - 9

objective :- To implement the basic of PL/SQL.

Introduction :- PL/SQL bridges the gap between database technology and procedural Proco procedural Program programming languages. It can be thought of as a development tool that extends the facilities of Oracles SQL database language. Via PL/SQL you can insert, delete, update and retrieve table data as well as use Proceedual procedural techniQuestions. Such as writing loops or branching to another block of code.

PL/SQL Block Structure :-

Declare DECLARE

Declarations of memory variables. Used later later

BEGIN

SQL executable Statements for manipulating to table data.

EXCEPTIONS

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

END;

Displaying User Messages on on the Screen - Any programming tool requires a method through through which message messages can be displayed to the User.

dbms

dbms Output is a package that includes a number of Procedure and function that accumulate information

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

dbms output is a package that includes a number of procedural 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.O

DBMS OUTPUT.PUT LINE ('Display User message');

ii) Conditional Control in PL/SQL.-

Syntax:-

If <Condition> Then
<Action>
Elseif <Condition>
<Action>
Else
<Action>
Endif;

eg: If <Id = 1> Then
print 'Ram das'
Elseif <Id = 2> Then

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Print 'Sham das'
Else
Print 'Not a Student'
EndIf;

ii) The ~~While~~ While LOOP :

Syntax:

~~WHILE~~ While <Condition>
~~Do~~ Loop
<Action>
~~End loop;~~ End Loop ~~End loop~~ Endloop;

eg:- While <Class = 12>
loop
Print '~~Stu~~ Student of Class 12'
~~End loop;~~ End loop;

iii) The for loop Statement :

Syntax :

for Variable in [Reverse] Start - end
Loop
<Action>
~~End loop;~~ End loop;

eg:- for U.P in [State] Start - end
Loop
print 'State is U.P'
End loop;

Teacher's Signature _____

Persons

Name	Id	~~Class~~	Class	State
Ram das	1		~~12~~ 12	U.P
Sham das	2		12	Bihar

~~ii)~~ ~~Ram das~~

iii) State is U.P

Expt. No. _____ Date _____

Page No. _____

The Goto Statement: ~~The~~ The goto Statement allows you to ~~Changes~~ ~~Change~~ Change the flow of Control within a PL/SQL Block.

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Experiment - 10

Objective :- To implement the Concept of Cursor and Trigger.

Cursor :- We have seen how Oracle execute 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 explicit declare a Cursor to Process the rows individually. A Cursor declare by the User is Called Explicit Cursor. For Questionaries that return more then 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 Want 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 sets are :-

1. Declare a Cursor that specifies the SQL Select Statement that You Want to Process.

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

- Open the Cursor.
- Fetch the data from the Cursor one rows at a time.
- Close the Cursor.

Explicit Cursor Attributes :- Oracle provides certain attributes/ Cursor ~~Variable~~ Variable to Control the execution of the Cursor. Whenever any Cursor (explicit or implicit) is opened and used Oracle Creates a set of four System ~~Variable~~ ~~Variat~~ ~~Vario~~ 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.

i) How to Declare the Cursor :-

The General Syntax to Create any particular Cursor is a follows :-

Cursor < Cursorname > is Sql Statement ;

eg :-

Cursor < Student >

ii) How to open the Cursor :-

The General Syntax to open any particular Cursor is a follows :-

Open Cursorname ;

eg :-

open Student ;

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

- Open the Cursor.
- Fetch the data from the Cursor. one rows at a time.
- Close the Cursor.

Explicit Cursor Attributes :- Oracle provides certain attributes/ Cursor ~~Variable~~ Variable to Control the execution of the Cursor. Whenever any Cursor (explicit or Implicit) is opened and used Oracle Creates a set of four System ~~Variable~~ ~~Variat~~ ~~Vario~~ 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.

i) How to Declare the Cursor :-
The General Syntax to Create any perticular Cursor is a follows :-

Cursor < Cursorname > is Sql Statement ;

eg :-

Cursor < Student >

ii) How to open the Cursor :-
The General Syntax to open any particular Cursor is a follows :-

Open Cursorname ;

eg :-

Open Student ;

Teacher's Signature _____

Person.

Name	Class	Id	State
Ram das	12	1	U.P Bihar
Sham das	12	2	~~Bihar~~ Bihar ~~Bihar~~ ~~Bihar~~ Bihar

i) Student Cursor. Declare

ii) Student. Cursor Open

iii) Student Cursor Close

iv) Student Cursor ~~Deteated~~ ~~Drop~~ Deleted. Deleted.

Fetching a record from the Cursor:-
The fetch statement retrieves the rows from the active set to the variable one at a time. Each time a fetch is executed. The focus of the DBA. DBA Cursor advances to the next row in the Cursor 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 at a time.

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

Close <Cursorname>;

eg:-

Close <Student>;

Database Triggers:-
Database Triggers are procedure that are stored in the database and are implicitly executed (fired). When the Contents of a tables are Changed.

Use of Database Triggers:-
Database triggers Support Oracle to provide a highly Customised DBMS. database management System.

- A Trigger can permit DML Statement against a table only if they are issued during regular bussiness hours or on predermined Weekdays.
- A trigger can also be Used to keep an audit trail of table along with the operation Performed and the time on which the operation

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

Fetching a record from the Cursor:-

The fetch statement retrieves the rows from the active Sets to the variable one at a time time time time time time time. Each time a fetch is executed. The focus of the DBA. DBA. Cursor advances to the next row in the Cursor Set. One can make use of any loop Structure to fetch (loop - End loop along with While For) for) the records from the Cursor into variable one at a time time.

iii) The General Syntax to Close the Cursor is as follows:-

Close <Cursorname>;

eg:-

Close <Student>;

Database Triggers:-

Database Triggers are procedure that are stored in the database and are implicitly executed (fired). When the contents of a tables are changed.

Use of Database Triggers:-

Database triggers support Oracle to provide a highly. Customised database management System.

- A Trigger can permit DML statement against a table only if they are issued during regular bussiness hours or on predermined weekdays.
- A trigger can also be used to keep an audit trail of table along with the operation Performed and the time on which the operation

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Ido Was performed

... If Can It. Can be Used to prevent invalid transactions

.. Enforce Complex Security authorizations.

How to apply 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 Statement Trigger:- After executing the trigring Statement and applying any deferred integrity Constraints the trigger action is exicuted.
4. After row Trigger:- After modefying each row affected by the triggering Statement and possibly applying appropriate integrity Constraints, the

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

trigger action is executed for the Current row if the trigger restriction either evaluates to eval evaluates evaluates to TRUE or was not included.

Syntax for Creating Creating Creating Creating Trigg Trigger Trigger:- Trigger:-

The Syntax for Creating the Trigger is a follows:-

Create or replace Trigger <Triggername> {Before after} {Before after} {Delete, Insert, Update} On <Tablename> For Each row When Condition

Declare

<Variable declarations>;

<Constant Declarations>;

Begin

<PL/SQL> Subprogram Body;

Exception Exception

Exception PI/SQL block;

End;

How How to Delete a Trigger:-

The Syntax for Deleting the Trigger is as follows:-

Drop Trigger <Triggername>;

eg eg:-

Drop Trigger <Student>;

Teacher's Signature _____