

Jawaharlal Nehru Engineering College

Laboratory Manual

Relational Database Management System

For

Second year Students

11, July 2016 –MCA- ISO 9000 Tech Document

© Author JNEC, Aurangabad

FORWARD

It is my great pleasure to present this laboratory manual for second year MCA students for the subject Relational Database Management System keeping in view the vast coverage required for process involved in database development.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9000 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9000 Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relieved them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Prof.Dr H.H. Shinde.
Principal

LABORATORY MANUAL CONTENTS

This manual is intended for the second year students of Master of Computer Application in the subject of Relational Database Management Systems. This manual typically contains practical/Lab Sessions related Database management Systems covering various aspects related the subject to enhanced understanding.

Although, as per the syllabus, study of creating database is prescribed, we have made the efforts to cover various aspects of Database management Systems covering different aspects of creating database which contains writing basic SQL SELECT statements, Restricting and sorting data. Displaying data from multiple tables, Aggregating data using group functions, Manipulating data, Creating and managing tables, constraints, Creating views, and Controlling user access.

Students are advised to go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Dr. S.S.deshmukh

MGM's



Jawaharlal Nehru Engineering College, Aurangabad

Department of Master of Computer Application.

Vision of MCA Department:

Build a strong technical teaching and learning environment that respond swiftly to the challenges and need of the current industry trends.

Mission of the MCA Department:

- Provide excellent post graduate education in a state of art environment, preparing students for careers as computer technologist in self-employment, industry, government and of IT sector.
- Support society by participating in and encouraging technology transfer.

Programme Educational Objectives:

- **Students will inculcate ability to apply knowledge of mathematics, science , computing and engineering.**
- **Engineering will be employed as software engineer in the field of Information technology or pursue further advanced courses of education**
- **Enrich student with the ability to identify, formulate and solve computer science and engineering problems and to define the computing requirements appropriate to their solution**
- **Post graduate will communicate effectively ,work collaboratively and exhibit high level of professionalism and ethical responsibility**

Programme Outcomes

(POs): Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

SUBJECT INDEX

- 1) Study of E-R Diagram
- 2) Study of Data Definition Language commands
- 3) Study of Data Manipulation Language
- 4) Study of data communication using SQL functions.
- 5) Study Join command
- 6) Study sub queries
- 7) Study Creating Views
- 8) Study Triggers
- 9) Study Procedures
- 10) Study DBA commands
- 11) Normalization in oracle

DOs and DON'Ts in Laboratory:

1. Do not handle any equipment before reading the instructions/Instruction manuals
2. Read carefully the power ratings of the equipment before it is switched on whether ratings 230 V/50 Hz or 115V/60 Hz. For Indian equipments, the power ratings are normally 230V/50Hz. If you have equipment with 115/60 Hz ratings, do not insert power plug, as our normal supply is 230V/50 Hz, which will damage the equipment.
3. Observe type of sockets of equipment power to avoid mechanical damage
4. Do not forcefully place connectors to avoid the damage
5. Strictly observe the instructions given by the teacher/Lab Instructor

Instruction for Laboratory Teachers::

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.
2. Students should be taught for taking the printouts under the observation of lab teacher.
3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

WARMUP EXERCISES:

Define Database.

Define tables and how to create it.

How can you create a user in database environment?

How database is restricted.

What manipulations you can do on tables?

What is SQL?

What is relational model?

What are group functions?

How can you create views?

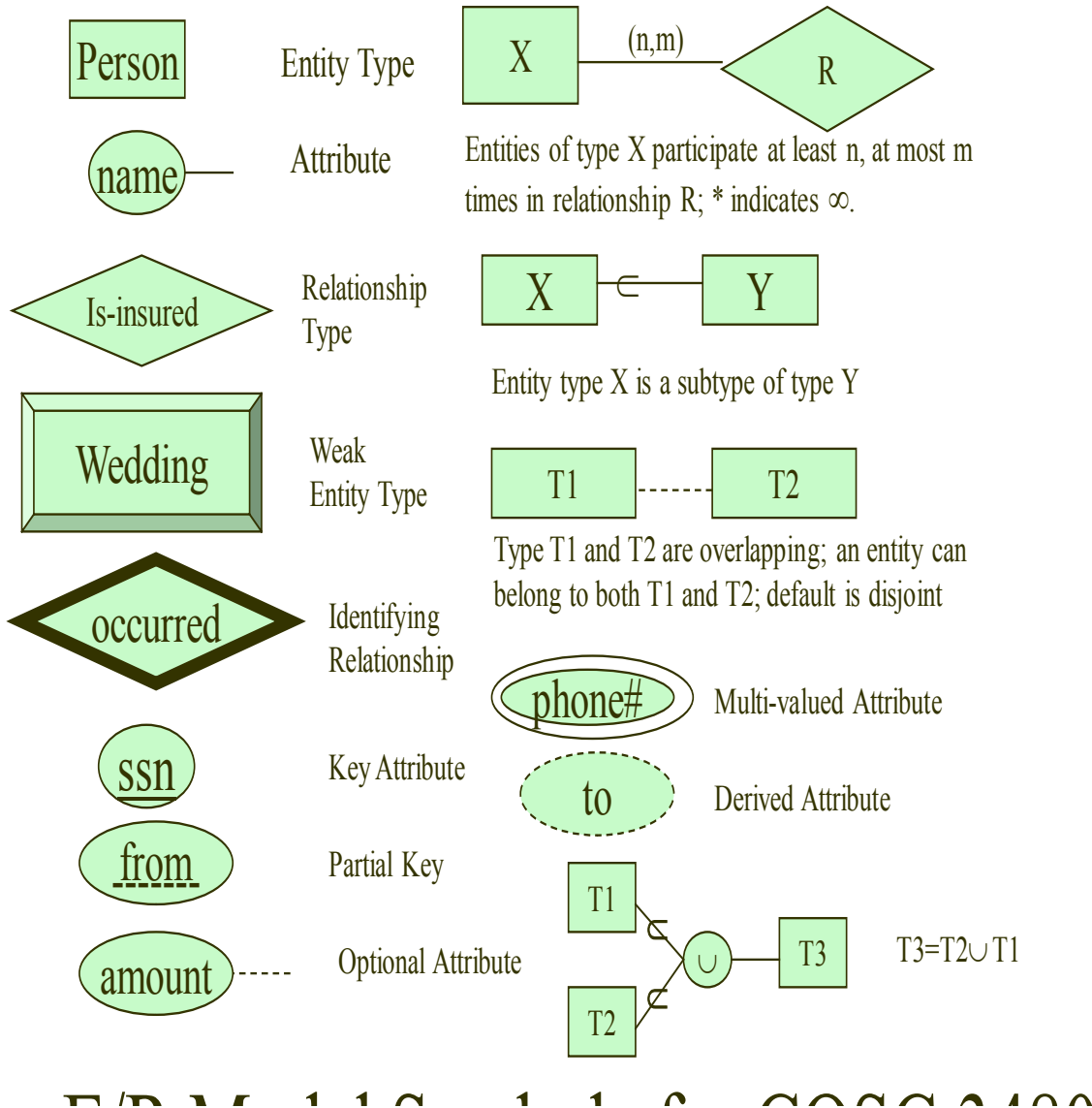
How can you retrieve data?

1. Lab Exercises:

Exercise No 1: (2 Hours)

Study of E-R Diagram

E-R diagram is a popular, high-level conceptual data model. This model and its variations are frequently used for conceptual design of database applications, and many database design tools employ its concepts. E-R model diagrammatic notations are as follows:



Assignment:

Design a “good” entity-relationship diagram for Bank Database. Indicate the cardinalities for each relationship type; assign roles (role names) to each relationship if there are ambiguities! Use sub-types, if helpful to express constraints!

Bank Database

The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. The bank monitors the assets of each branch.

Bank customers are identified by their customer_id values. The bank stores each customer’s name and the street and city where the customer lives. Customers may have accounts and can take out loans. A customer may be associated with a particular banker, who may act as loan officer or personal banker for the customer.

Bank employees are identified with their employee_id values. The bank administrator stores the name and telephone number of each employee, the names of the employee’s dependents, and their employee_id number of the employee’s manager. The bank also keeps track of the employees start date and thus the length of employment.

The bank offers two types of accounts – the savings and the checking accounts. Accounts can be held by more than one customer, and a customer can have more than one account. Each account is assigned a unique account number. The bank maintains a record of each account’s balance and the most recent date on which account was accessed by each customer holding the account. In addition, each savings account has an interest rate and overdrafts are recorded for each checking account.

A loan originates at a particular branch and can be held by one or more customers. A loan is identified by a unique loan number. For each loan, the bank keeps track of the loan amount and the loan payments. Although a loan payment number does not uniquely identify a particular payment among all those for all the bank loans, a payment number does identify a particular payment for a specific loan. The date and amount are recorded for each payment.

2. Lab Exercises:

Exercise No2: (2 Hours)

Study of Data Definition Language (DDL)

DDL is a language that allows the DBA or user to describe and name the entities, attributes and relationships required for the application, together with any associated integrity and security constraints. The database schema is specified by a set of definition expressed by means of a special language called a Data Definition Language. The DDL is used to define a schema or to modify an existing one; it cannot be used to manipulate data.

The result of the compilation of The DDL statements is a set of tables stored in special files collectively called the system catalog. The system catalog integrates the metadata, is data that describes objects to be accessed or manipulated.

The SQL DDL allows database objects such as schemas, domains, tables, views and indexes to be created and destroyed.

The main SQL DDL statements are:

- **CREATE TABLE** TableName
 {(ColumnName dataType [**NOT NULL**] [**UNIQUE**]
 [**DEFAULT** defaultOption] [**CHECK** (searchCondition)] [,...]}
 [**PRIMARY KEY** (listOfColumns),]
 {[**UNIQUE** (listOfColumns),] [,...]}
 {[**FOREIGN KEY**(listOfForeignKeyColumns)
 REFERENCES ParentTableName [(listOfCandidateKeyColumns)],
 [**MATCH** {**PARTIAL** | **FULL**}
 [**ON UPDATE** referentialAction]
 [**ON DELETE** referentialAction]] [,...]}
 {[**CHECK** (searchCondition)] [,...]}
 }
- **ALTER TABLE** TableName
 [**ADD [COLUMN]** columnName dataType [**NOT NULL**] [**UNIQUE**]
 [**DEFAULT** defaultOption] [**CHECK** (searchCondition)]
 [**DROP [COLUMN]** columnName [**RESTRICT** | **CASCADE**]]
 [**ADD [CONSTRAINT** [constraintName]] tableConstraintDefinition]
 [**DROP [CONSTRAINT** constraintName [**RESTRICT** | **CASCADE**]]
 [**ALTER [COLUMN] SET DEFAULT** defaultOption]
 [**ALTER [COLUMN] DROP DEFAULT**]
 }
- **DROP TABLE** TableName [**RESTRICT** | **CASCADE**]

Assignment

Following Tables are used for all assignments given in the lab manual.

PropertyForRent (PropertyNo, street, city, postcode, Type, rooms, rent, ownerNo, StaffNo, BranchNo)

Staff (StaffNo, FName, LName, position,sex , dob, salary, branchNo)

Branch (BranchNo, Street,City, PostCode)

Client (ClientNo, FName, LName, TelNo, PrefType, MaxRent)

PrivateOwner (OwnerNo, FName, LName, Address, TelNo)

Viewing (ClientNo, PropNo, ViewDate, Comment)

Registration (ClientNo, BranchNo, StaffNo, DateJoined)

- Create the PropertyForRent table using the available features of the CREATE TABLE statement.
- Change the Staff table by removing the default of 'Assistant' for the position column and setting the default for the sex column to Female.
- Change the PropertyForRent table by removing the constraint that staff is not allowed to handle more than 100 properties at a time. Change the client table by adding a new column representing the preferred number of rooms.
- Remove the propertyforrent table
- on deletion of staffno from staff table the concern record from PropertyForRent must getdeleted.

3. Lab Exercises:

Exercise No 3: (2 Hours)

Study of Data Manipulation Language (DML) COMMANDS:

DML is a language that provides a set of operations to support the basic data manipulation operations on the data held in the database. Data manipulation operations usually include the following.

- Insertion of new data into the database

- Modification of data stored in the database
- Retrieval of data contained in the database.
- Deletion of data from the database.

The SQL DML statements are:

- SELECT –To query data in the database

```
SELECT      [DISTINCT | ALL] {*} | [columnExpression [AS newName]] [,....]]
FROM       TableName [alias] [,...]
[WHERE]     condition]
[GROUP BY  columnList] [HAVING condition]
[ORDER BY  columnList]
```

- INSERT – To insert data into the table

```
INSERT INTO TableName [(ColumnList)]
VALUES (data ValueList)
```

- UPDATE – To update data in a table

```
UPDATE TableName
SET columnName1=dataValue1 [, columnName2=dataValue2....]
[WHERE searchCondition]
```

- DELETE – To delete data from a table

```
DELETE FROM TableName
[WHERE searchCondition]
```

Assignment

- List full details of all Staff.
- Produce a list of salaries for all staff showing only the staff number, the first and last names, and the salary details.
- List the property numbers of all properties that have been viewed.
- Produce a list of monthly salaries for all staff, showing a staff number, the first and last names and the salary details.
- List all staff with a salary greater than \$10000.
- List the addresses of all branch offices in London or Glasgow.

- List all staff with a salary between \$20000 and \$30000.
- List all managers and supervisors.
- List the details of all viewings on Property PG4 where a comment has not been supplied.
- Produce a list of salaries for all staff range in descending order of salary.
- List the details of PrivateOwners where address field contains a string 'Glasgow'.
- Produce an abbreviated list of properties range in order of property type.
- How many properties cost more than \$350 per month to rent?
- How many different properties were viewed in May 2001?
- Find the total no of managers and the sum of their salaries.
- Find the Min and Max staff salaries.
- Find the no of staff working in each branch and the sum of their salaries.
- For each branch office with more than one member of staff find the no of staff working in each branch and the sum of their salaries.
- List the staffs who work in the branch at '163 Main Street'.
- List all staff whose salary is greater than the average salary and show by how much their salary is greater than the average.
- List the properties that are handled by staff who work in the branch at '163 Main Street'.
- Find all staff s whose salary is larger than the salary of at least one member of staff at branch B003.
- Find all staff whose salary is larger than the salary of every member of staff at branch B003.
- Insert a new row into the staff table supplying data for all columns.
- Populate the StaffPropCount table using details from the staff and PropertyForRent table.
- Give all staff a 3% pay increase
- Give all managers a 5% pay increase
- Promote David Ford (StaffNo-SG14) to manager and change his salary to \$18000.
- Delete all viewings that relate to property PG4.
- Delete all rows from the viewing table.

4. Lab Exercises:

Exercise No 4: (2 Hours)

Study of data communication using SQL functions:

Functions serve the purpose of maintaining data items and returning a result .Functions are also capable of accepting user supplied values and constants and manipulating them. Such user defined values or constants are called arguments. Any number of arguments can be passed to function.

Functions that act on only one value at a time are called scalar functions. Functions can be classified corresponding to different data types as:

Aggregate Functions

- AVG (DISTINCT | ALL | n)
- MIN (DISTINCT | ALL | expr)
- COUNT (DISTINCT | ALL | expr)
- COUNT (*)
- MAX (DISTINCT | ALL | expr)
- SUM (DISTINCT | ALL | n)

Numeric Functions

- ABS(n)
- POWER(m,n)
- ROUND(n,m)
- SQRT(n)

String Functions

- LOWER(char)
- INITCAP(char)
- UPPER(char)
- SUBSTR(char,m[,n])
- LENGTH(word)
- LTRIM(char[,set])
- RTRIM(char[,set])
- LPAD(char1,n [,char2])
- RPAD(char1,n [,char2])

Conversion Functions

- TO_NUMBER(char)
- TO_CHAR(n[,fmt])

Date Conversion Functions

- TO_DATE (char [, fmt])

DATE Functions

- ADD_MONTHS(d,n)
- LAST_DAY(d)
- MONTHS_BETWEEN(d1,d2)
- NEXT_DAY(date, char)

Set Operations

- UNION
- UNION ALL
- INTERSECTION
- MINUS

Assignment

- List the no of jobs available in the EMP table

- List the no of employees working with the company.
- List the dept no and number of employees in each dept.
- List the total salary, max and min salary and the average salary of employees' job wise for deptno 20 only.
- List the average salary for all department employing more than five people.
- List the max salary of employee working as salesman.
- Display the name in the lowercase, job in the initial capitalization for all managers.
- Display the value 45.923 rounded to the hundredth number and ten decimal places.
- For employees in dept 30, display the ename and number of weeks/months employed.
- Display the head of the company, who has no manager, display that there is no manager for that name.
- Display the job and total salary for each job category with a total salary for each job category 3000 and sort list by total salary.
- Display the number of managers without listing them.
- Write a query to display the minimum and maximum salary for each job type ordered alphabetically.

5. Lab Exercises:

Exercise No 5: (2 Hours)

Study Join command

The SQL join operation combines information from two tables by forming pairs of related rows from the two tables. The row pairs that make up the joined table are those where the matching columns in each of the two tables have the same value.

If we need to obtain information from more than one table the choice is between using a sub query and using a join.

In a relational database system, the total required data might not be available in a single table. Generally the data is scattered. Because if the total data is stored in a single table it will lead to a lot of redundancy. So, often we have to bring the data of two or more tables together to get the required information.

The act of combining two or more tables in such a way that you retrieve values from the columns of all the tables, to get the required data is called as joining. In order to join two tables, there must be a common column between those two tables. For example to join COURSES and BATCHES table, we use CCODE column as that is a common column between these two tables.

The following SELECT command is used to get information about batches from both COURSES and BATCHES table. We take a row from BATCHES table and get the name of course from COURSES table using CCODE column of BATCHES. CCODE column of BATCHES is used to get the corresponding row from COURSES table. As we used equal to operator to join these two tables, the join is also called as equi-join.

```
select bcode, batches.ccode, name, fcode, stdate from batches, courses where batches.ccode = courses.ccode
```

BCODE	CCODE	NAME	FCODE	STDATE
b1	ora	Oracle database	gk	12-JAN-01
b2	asp	ASP.NET	da	15-JAN-01
b3	c	C programming	hs	20-JAN-01
b4	xml	XML Programming	dh	02-MAR-01
b5	java	Java Language	hs	05-APR-01
b6	vbnet	VB.NET	da	12-JUL-01
b7	ora	Oracle database	jc	15-AUG-01

In the above query the following are the important points.

1. Two tables are used in FROM clause. Tables are separated by comma.
2. WHERE clause contains the condition using which both the tables are to be joined.
3. Column CCODE is to be qualified using the table name as it exists in both the table that are used in FROM clause.

Table Alias

While joining the table we have to qualify the columns that are in more than one table using the table name. If table name is lengthy this process could be very tedious. Table alias is a short name that can be used to refer to the table name in the query. For example, the above query can be rewritten using table alias as follows:

```
select bcode, b.ccode, name, fcode, stdate
from batches b, courses c
where b.ccode = c.ccode
```

B is the alias to table BATCHES and C is the alias for COURSES. Throughout the query table BATCHES can be referred using the alias B and COURSES using alias C. As the purpose of using an alias is to shorten the reference to table, alias is generally very short. Also remember alias is available only in the query in which it is created.

Product of two tables

While tables are joined, if WHERE clause is not given then it results in PRODUCT of the table that are being joined. Product is the result in which each row of first table is joined with each row of the second table. This is also called as Cartesian product. The following example will join

```
select bcode, b.ccode, name, fcode, stdate, enddate from batches b, courses c;
```

The following example will join

```
select bcode, b.ccode, name, fcode, stdate, enddate from batches b, courses c;
```

The above command will result in 42 (6* 7) rows as we have 6 rows in COURSES and 7 rows in BATCHES table. Product is generally the result of an error than the desired result.

Join condition and normal condition

While joining two tables the condition used to join the table is called as join condition. As we have above, without join condition, the result of joining will be product of the table. Apart from the join condition, which is required for joining, normal conditions can also be given in WHERE clause. Then Oracle uses normal condition to filter rows and then joins the filtered rows based on the join condition.

The following query will get the details of batches that are completed along with name of the course. To get the name of the course it uses COURSES table and to select only those batches that are completed, it uses a normal condition as follows.

```
select bcode, b.ccode, name, fcode, stdate from batches b, courses c where b.ccode = c.ccode and enddate is not null
```

BCODE	CCODE	NAME	CODE	STDATE
b1	ora	Oracle database	gk	12-JAN-01
b2	asp	ASP.NET	da	15-JAN-01
b3	c	C programming	hs	20-JAN-01
b4	xml	XML Programming	dh	02-MAR-01
b5	java	Java Language	hs	05-APR-01

Joining more than two tables

Just like how two tables are joined, more than two tables can also be joined to get the required information. For example, in the above query we retrieved information about batches and course name. What if we want to get name of the faculty and not just the code? Then we have to use FACULTY table along with BATCHES and COURSES table. To get the details of batches along with name of the course and name of the faculty, give the following:

```
select bcode, c.name course, f.name faculty, stdate from batches b, courses c, faculty f where b.ccode = c.ccode and b.fcode = f.fcode
```

The query takes data from three tables – BATCHES, COURSES and FACULTY. It uses CCODE to join BATCHES table with COURSES table. It uses FCODE to join BATCHES table with FACULTY table.

When two or more tables are joined then the minimum number of conditions to be given is derived from the following formula:

Number of join conditions = Number of tables being joined - 1

If more than one column is used to join tables then the number of join conditions may be even more.

There are various types of joins

- Cross joins
- Equijoin
- Natural joins
- USING clause
- Self join
- Non-equijoin
- Outer join- left outer, right outer and full outer

Assignment

1. List the total information of EMP table along with Dname and Loc of all the emps working under 'ACCOUNTING' or 'RESEARCH' in the asc Deptno.
2. List the Empno, Ename, Sal, Dname of all the 'MANAGER' and 'ANALYST' working in New York or Dallas with an exp more than 7 years without receiving the Comma sc order of Loc.
3. Display the total information of the emps along with Grades I the asc order of grade.
4. List all the grade 2 and Grade 3 emps.
5. Display all grade 4.5 Analyst and Manager.
6. List Empno, Sal, Dname, grade, Experience and Annual sal of emps working for Dept 10 or 20.
7. 8List the details of the Depts along with Empno, Ename or without the emps.
8. List the emps whose are senior to their own Manager.
9. List the emps of Deptno 20 whose Jobs are same as Deptno 10.
10. List the emps whose Salary is same as FORD or SMITH in desc order of Salary.
11. List the emps whose Jobs are sale as MILLER or Sal is more than ALLEN.
12. List the emps whose Sal is more than the total remuneration of the SALESMAN.
13. List the emps who are senior to BLAKE working at CHICAGO & BOSTON.
14. List the emps of Grade 3,4 belongs to the dept ACCOUNTING and RESEARCH whose Sal is more than ALLEN and exp more than SMITH in the asc order of Exp.
15. List the emps whose jobs same as SMITH or ALLEN.
16. List the most recently hired employee of Grade 3 belongs to the Loc CHICAGO.
17. List the emps with their Dept names.
18. List the emps who are not working in 'Sales' dept.
19. List the emps Name, Dept, Sal & Comm for those whose Salary is between 2000 and 5000 and Loc is Chicago.
20. List the emps whose Salary is greater than his Manager's salary.

6. Lab Exercises:

Exercise No 6: (2 Hours)

Study Sub queries

A Subquery is a query that appears within another SQL command. Three of VFP's SQL DML commands (SELECT, DELETE, and UPDATE) support subqueries, though the rules and reasons for using them vary. VFP 9 increased the capabilities of subqueries and the ways they can be used. Some subqueries stand alone; you can run the subquery independent of the command that contains it. Other subqueries rely on fields from the containing command—these subqueries are said to be correlated

A subquery is a complete query, but cannot contain the TO or INTO clause. Subqueries are enclosed in parentheses in the containing query. Subqueries can appear in the WHERE clause of SELECT, UPDATE, and DELETE. Beginning in VFP9, subqueries can also be used in the field list of SELECT, in the SET clause of UPDATE, and in the FROM clause of SELECT, UPDATE, and DELETE.

```
SELECT select list FROM table
WHERE expr operator
      (SELECT
        Select List FROM table);
```

The sub query (inner query) executes once before the main query.
The result of the sub query is used by the main query (outer query)

You can place the subquery in a number of SQL clauses

- WHERE clauses
- HAVING clause
- FROM clause

In the syntax; operator includes a comparison operator such as >, =, or IN. Comparison operators fall into two classes: single-row operators (> , = , >= , < , < > , <=) and multiple-row operators (IN , ANY , ALL).

```
SELECT ename FROM EMP WHERE sal >( SELECT sal FROM emp WHERE empno=7566);
```

```
SELECT ename, sal, deptno FROM EMP WHERE sal IN ( SELECT MIN(sal) FROM emp GROUP BY deptno );
```

Assignments:

- List the names of all clients who have viewed a property along with any comment supplied.
- For each branch office list the numbers and names of staff who manage properties and the properties that they manage.
- For each branch list the numbers and names of staff who manage properties including the city in which the branch is located and the properties that the staff manage.
- Find the no of properties handled by each staff member.
- List the branch offices and properties that are in the same city along with any unmatched branches.
- List the branch offices and properties that are in the same city along with any unmatched properties.

- List the branch offices and properties that are in the same city along with any unmatched branches or properties.
- Find all staff who work in London branch office.

7. Lab Exercises:
Exercise No 7: (2 Hours)

Creating views:

After a table is created and populated with data, it may become necessary to prevent all users from accessing all columns of a table for data security reasons. This would mean creating several tables having the appropriate number of columns and assigning specific user to each table as required. This will answer data security requirements very well but will give rise to a great deal of redundant data being resident in tables in the database.

To reduce redundant data to the minimum possible, oracle allows the creation of an object called a view. A view is mapped to a select sentence. The table on which the view is based is described in the FROM clause of the SELECT statement.

Assignment:

1. Use SQL'S CREATE TABLE statement to create the appropriate tables having the table definitions mentioned below.

Table Name: Material_Mstr
 Column Definition:

Column Name	Data Type	Width	Allow Null	Default
RmId	Number	5	No	
RmName	VarChar2	20	No	
ReOrdLv1	Number	5	No	

Table Name: Material_Dtls
 Column Definition:

Column Name	Data Type	Width	Allow Null	Default
RmId	Number	5	No	
SupIrId	Number	5	No	
PurDt	Date	--	No	

2. Using the SQL's ALTER TABLE statement, modify the table structure such as to add the relationship constraints as described below.

Table Name : Material_Mstr
 Primary Key : Material_Mstr. RmId
 Table Name : Material_Dtls
 Foreign Key : Material_Mstr. RmId, Supplier_Mstr. SuplrId

3. Using ANSI SQL's INSERT statement, insert the following set of information into the tables created earlier (Material_Mstr and Material_Dtls).

Table Name:Material_Mstr		
RmId	RmName	ReodrLvl

1	Monitor	4
2	32 MB RAM	10
3	PC Cabinet	10
4	Mother Board	3
5	Floppy Drive	15
6	CD Writer	3
7	CD Drive	5
8	Key Board	10
9	Power Supply	8
10	Mouse	10

Table Name:Material_Dtls				
RmId	SuplId	PurDt	QtyPur	UtCost
1	2	01/02/2003	10	5550
2	2	01/02/2003	20	1250
3	2	01/02/2003	20	500
4	5	24/02/2003	5	10000
5	3	25/02/2003	25	300
6	3	25/02/2003	5	4250
7	3	25/02/2003	10	3000
8	1	28/02/2003	20	400
10	1	28/02/2003	20	500
9	4	01/03/2003	20	500
3	4	01/03/2003	15	450

4. Use the ANSI SQL's SELECT statement to perform the following operations.
 - a) View all records from the table Material_Mstr.
 - b) View all records from the table Material_Dtls.
 - c) View the quantities of each material purchased listed alphabetically.
 - d) View the names of material whose total quantities is greater than 10.
 - e) View the names of material which has the highest total quantity.
 - f) View the total quantities of various material purchased.
 - g) View just the names of material purchased without repeating any item.

5. Using ANSI SQL's UPDATE statements, to perform the following operations:
 - a) Replace the material named Power Supply with UPS,
 - b) Add 10% to the quantities purchased for material named Mouse,
 - c) Increase the quantities of material purchased by 5 for all purchases done after Feb 2003.
6. Using ANSI SQL's DELETE statement, to perform the following operation.
Delete records from Material_Dtls for material having RmId as 5.
7. Create an English sentence, by joining predetermined string values with column data retrieved from the Material_Mstr table.

The string literals are:

The Material	Has the identity number as
--------------	----------------------------

The columns are:

RmName	RmId
--------	------

8. Create VIEW OBJECT containing the following columns:

Column Definition:

Column Name	Data Type	Width	Allow Null	Default
RmName	Varchar2	20	No	
PurDt	Date	-	No	
QtyPur	Number	5	No	0
UtCost	Number	7,2	No	0.00

The data required in this VIEW should be taken for Material_Mstr and Material_Dtls tables.

8. Lab Exercises:

Exercise No 8: (2 Hours)

Writing and Executing PL/SQL block:

- Cursor
- Triggers

Cursor

The RDBMS engine uses a work area that is a specific place in computer's memory for all its internal processing in order to execute SQL statement. This work area is private to SQL operations and is called a cursor.

The data that is present in the cursor is called the active data state. Conceptually the size of the cursor in memory must be the size required to hold all the rows of the active data state. However RDBMS engines built-in memory management capabilities and the amount of free memory available in the computers determines actual cursor size.

Assignment:

- An HRD has decided to raise the salary of employees working in department 30 by 0.15%. Write a PL/SQL block to update the same using implicit cursors.
- An HRD manager has decided to raise the salary of all employees working whose salary is less than Rs.5000/- by Rs.1000.
- Write a PL/SQL code block that does this using an Explicit Cursor and treats the updating process as a single transaction.
- Write a PL/SQL code block that performs the task mentioned in Exercise 2 while using the FOR... IN statement.
- The Emp table mentioned below will be used as a base table for creating the solutions for the above exercises.

Table Name: EMP

EmpNo	EmpName	DeptNo	Salary
E0001	Mamta	30	6000
E0003	Hansel	10	6500
E0005	Ashwini	20	6000

9. Lab Exercises:

Exercise No 9:(2 Hours)

To study Triggers

Information about triggers can be retrieved from two views, the all_triggers view and the all_trigger_cols view. Most of the time you will find all the information you need in all_triggers. The all_trigger_cols view contains a list of all database columns referenced in the trigger. This view is sometimes useful when you are troubleshooting because it can show you which trigger references or modify any given database column.

To find out whether any triggers have been defined on table, query all_triggers as shown in ex .1 for 'Listing the names of triggers on a table' given below.

Ex.1

```
SET VERIFY OFF
COLUMN description FORMAT A40 WORD_WRAPPED
COLUMN status FORMAT A10
SELECT description, status
FROM all_triggers
WHERE table_owner =UPPER ('&owner')
AND table_name=UPPER ('&table_name');
```

Ex.2 shows a script that will describe a single trigger in detail.

The script's putput is a CREATE TRIGGER statement that may be used to re-create the trigger. The FOLD_AFTER option is used in the column commands to force each column to begin a new line of output. SET PAGESIZE 0 GETS RID OF any page titles and column headings that would otherwise clutter the output .The trigger _body column is of type LONG, so SET LONG 5000 ensures that you will see at least the first 5000 bytes of a trigger body. Use a higher value if your triggers are longer than that.

Ex.2

```
SET VERIFY OFF
SET LONG 5000
SET PAGESIZE 0
COLUMN create_ stmt FOLD_AFTER
COLUMN description FOLD_AFTER
COLUMN when_clause FOLD_AFTER

SELECT 'CREATE OR REPLACE TRIGGER'create_stmt
Description,
CASE WHEN when_clause IS NOT NULL THEN
'WHEN (||when_clause||)'
ELSE
  v
END  when_clause,
Trigger_body
FROM all_triggers
WHERE owner =UPPER ('&trigger_name');
AND trigger_name=UPPER ('&trigger_name');

SET PAGESIZE 14
```

The following invocation of Ex.2 shows the definition for the trigger emp_delete_check.

```
SQL>@ex2-2
Enter value for owner:gennick
Enter value for trigger_name:emp_delete_check
CREATE OR REPLACE TRIGGER
Emp_delete_check
BEFORE DELETE on employee
```



```
FOR EACH ROW
```

```
BEGIN
```

```
IF(:OLD .employee_termination_date IS NULL)
```

```
    OR (OLD.employee_termination_date >=TRUNC (SYSDATE) +1) THEN
```

```
        RAISE_APPLICATION_ERROR (-20001,'you must terminate an employee before  
        deleting his record');
```

```
END IF;
```

```
END;
```

This output contains a blank line in front of the BEGIN keyword. That blank line is where the WHEN clause would go, if one had been defined when the trigger was created.

10. Lab Exercises:

Exercise No 10: (2 Hours)

To Study database Administration

GRANT and REVOKE DBA commands

SQL provides the GRANT and REVOKE statements to allow security to be set up on the tables in the database. The security mechanism is based on the concepts of authorization, identifiers, ownership and privileges.

Assignment:

- Give the user with authorization identifier Manager full privileges to staff table.
- Give users Personal and Director privileges SELECT and UPDATE on column salary of the staff table.
- Give all users the privilege SELECT on the Branch table.
- Revoke the privilege SELECT on the Branch table from all users
- Revoke all privileges you have given to Director on the Staff table.

11. Lab Exercises:

Exercise No 11: (2 Hours)

Normalization in oracle

It is a technique for producing a set of relations with desirable properties given the data requirements of an enterprise. Normalization is often performed as a series of tests on a relation to determine whether it satisfies or violates the requirements of a given normal form. Three normal forms are mostly used called First (1 NF), Second (2 NF) and Third (3 NF). Subsequently, R. Boyce introduced a stronger definition of third normal form called Boyce-Codd normal form(BCNF) .All these normal forms are based on functional dependencies among the attributes of a relation .Higher normal form Fourth(4 NF) and Fifth (5 NF) are rarely used.

Assignment

1. A client gives a set of printed Challans to an analyst or database designer. The printed invoices include information like:

Challan number	Challan date	Purchase Order number
Supplier number	Supplier name	Supplier address

As well as item details like:

Raw material number	Raw material name
Raw material unit Price	Quantity delivered

As the analyst designer, create a mother matrix (single table) based on the above information.

2. Decompose the mother matrix generated in Exercise 1 into tables and eliminate all repeating groups of data.

12. Quiz on the subject:

Quiz should be conducted on tips in the laboratory, recent trends and knowledge of the subject. The quiz questions should be formulated such that questions are normally from the scope outside of the books. However twisted questions and self formulated questions by the faculty can be asked but correctness of it is necessarily to be thoroughly checked before the conduction of the quiz.

13. Conduction of Viva-Voce Examinations:

Teacher should conduct oral exams of the students with full preparation. Normally, the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the students in the subject is tested. Oral examinations are to be conducted in co-cordial environment amongst the teachers taking the examination. Teachers taking such examinations should not have ill thoughts about each other and courtesies should be offered to each other in case of difference of opinion, which should be critically suppressed in front of the students.

	Style						
Forward matter	Bookman Style	old	12	-----	-----	-----	-----
Lab man Contents title	Bookman Style	old	12	-----	-----	Yes	Capital
Index title	Bookman Style	old	12	Yes	-----	Yes	Capital
Index contents	Bookman Style	old	12	-----	-----	-----	-----
Heading	Tahoma		14	Yes	Yes	Yes	-----
Running Matter	Comic Sans MS		10	-----	-----	-----	-----

15. Evaluation and marking system:

Basic honesty in the evaluation and marking system is absolutely essential and in the process impartial nature of the evaluator is required in the examination system to become popular amongst the students. It is a wrong approach or concept to award the students by way of easy marking to get cheap popularity among the students to which they do not deserve. It is a primary responsibility of the teacher that right students who are really putting up lot of hard work with right kind of intelligence are correctly awarded.

The marking patterns should be justifiable to the students without any ambiguity and teacher should see that students are faced with unjust circumstances.