SUNBEAM

SUNBEAM

Institute of Information Technology

Database Technologies Notes

Course ..... Batch .....

Student Name ..... Reg ID .....

Index

Software Development Tools of MySQL

- MySQL database
- SQL
- MySQL-PL
- MySQL Command Line Client
- MySQL Workbench
- MySQL Connectors
- MySQL for Excel
- MySQL Notifier
- MySQL Enterprise Backup
- MySQL Enterprise High Availability
- MySQL Enterprise Encryption
- MySQL Enterprise Monitor
- MySQL Query Analyzer

MySQL Datatypes

- Char
- Varchar
- Tinytext
- Text
- Mediumtext
- Longtext
- Binary
- Varbinary
- Tinyblob
- Blob
- Mediumblob
- Longblob
- Enum
- Set

(Int Datatypes could be Signed or Unsigned):-

- Tinyint
- Smallint
- Mediumint
- Int
- Bigint
- Float
- Double
- Decimal
- Bit
- Boolean
- Date
- Time
- Datetime
- Year

Create table, Insert and Select statements

```
create table emp
(
Empno char(4),
Ename varchar(25),
Sal float,
City varchar(15),
Dob date
);

Insert into emp
Values('1', 'Amit', 5000, 'Mumbai', '1995-05-
04');

Select * from emp;

Select empno, ename from emp;

Select * from emp
where sal > 2000;

Select * from emp
Where sal > 2000 and sal < 3000;

Select * from emp
where job = 'MANAGER';

Select * from emp
where job = 'MANAGER' or job = 'CLERK';

select ename, sal*12 from emp;
```

```
select ename, sal*12 “ANNUAL” from emp;

select distinct job from emp;

select deptno, job, ename, sal, hiredate from emp
order by ename;

select deptno, job, ename, sal, hiredate from emp
order by ename desc;

select deptno, job, ename, sal, hiredate from emp
order by deptno;

select deptno, job, ename, sal, hiredate from emp
order by deptno, job;

select deptno, job, ename, sal, hiredate from emp
order by deptno desc, job;

select * from emp
where ename like ‘A%’;

select * from emp
where ename like ‘%A’;

select * from emp
where ename like ‘%A%’;

select * from emp
where ename like ‘_ _A%’;

select * from emp
where sal between 2000 and 3000;

select * from emp
where city in (‘Mumbai’ ‘Delhi’);
```

Some basic commands post logon

```
Show databases;

Use mysql;

Select user from user;

Use [db_name];

Show tables;

Desc emp;
```

Functions

Char Functions

```
Select Concat(fname,lname) from emp;

Select Upper(fname) from emp;

Select Lower(fname) from emp;

Select Lpad(ename,25,' ')  from emp;

Select Lpad(ename,25,'*') from emp;

Select Rpad(ename,25,' ') from emp;

Select Rpad(ename,25,'*') from emp;

Select Ltrim(ename) from emp;

Select Rtrim(ename) from emp;

Select Trim(ename) from emp;

Select Substr(ename,3) from emp;

Select Substr(ename,3,2) from emp;

Select Replace(ename,'un','xy') from emp;

Select Instr(ename,'un') from emp;

Select Length(ename) from emp;
```

```
Select Ascii(ename) from emp;

Select char(65 using utf8) from dual;

Select * from emp where soundex(ename) =
soundex('Aroon');
```

Number Functions

```
Select Round(sal) from emp;

Select Truncate(sal,0) from emp;

Select Ceil(sal) from emp;

Select Floor(sal) from emp;

Select Sign(-15) from dual;

Select Mod(9,5) from dual;

Select Sqrt(81) from dual;

Select Power(10,3) from dual;

Select Abs(-10) from dual;
```

MySQL Date Functions

```
Select Sysdate() from dual;

select adddate(sysdate(),2) from  dual;

select datediff(sysdate(),hiredate) from emp;

SELECT DATE_ADD(hiredate, INTERVAL 2 MONTH) from
emp;

SELECT DATE_ADD(hiredate, INTERVAL 1 year) from
emp;

Select last_day(hiredate) from emp;

select dayname(sysdate()) from dual;

Select  addtime('1997-01-15  10:00:00.00000','1')
from dual;
```

Environment Functions

```
Select User() from dual;

Show character set;
```

Sub-queries

```
select ename from emp
   where sal =
   (select min(sal) from emp);

select * from emp
   where deptno =
   (select deptno from emp
   where ename = 'Thomas');

select * from emp
   where job =
   (select job from emp
   where ename = 'Kirun');
```

Indexes

- To see which all indexes are created for specific table:-

```
show indexes from emp;
```

- To see all indexes on all tables in the database:-

```
use information_schema;
Select * from statistics;
```

- To drop the index:-

```
Drop index indexname on emp;
```

Alter table

```
Rename table emp to employees;

alter table emp add panno char(10);

alter table emp drop column panno;

alter table emp modify ename varchar(30);

Alter table emp modify ename varchar(20);

Alter table emp modify empno char(4);

Insert into emp select * from emp2;

Create table emp_copy
As
Select * from emp;
```

Constraints

Primary key constraint

```
select * from
information_schema.table_constraints;

Select * from
information_schema.key_column_usage where
table_name = 'EMP';
```

Not null constraint:-

- To see not null columns:-

```
Desc emp;

Alter table emp modify ename varchar(25) null;
```

Unique constraint

```
Show indexes from emp;
```

- Unique constraint is also an index so to drop it use:-

```
drop index deptno on emp;

drop index mob_no on emp;
```

Foreign key constraint

```
Alter table emp drop foreign key fk_emp_deptno;
```

- Can specify on delete cascade and on update cascade

- To disable the foreign key constraint:-

```
   For current connection:-
   set foreign_key_checks = 0;
   set foreign_key_checks = 1;
For all connections:-
   set global foreign_key_checks = 0;
   set global foreign_key_checks = 1;
```

Stored Procedures

```
delimiter //
create procedure abc()
Begin
insert into tempp values(1, 'Hello');
end //
delimiter ;
```

- To call the stored procedure:-

```
call abc();
```

```
delimiter //
create procedure abc()
Begin
declare x int;
set x = 10;
insert into tempp values(x, 'Hello');
end //
delimiter ;
delimiter //
create procedure abc()
Begin
declare x int(4) default 10;
insert into tempp values(x, 'Hello');
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x char(15) default 'CDAC';
insert into tempp values(1, x);
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x char(15) default 'KING';
declare y float(7,2) default 3000;
declare z float(2,1) default 0.4;
declare hra float(7,2);
set hra = y*z;
insert into tempp values(y,x);
insert into tempp values(hra,'HRA');
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x int(4);
select sal into x from emp where ename = 'KING';
insert into tempp values(x, 'KING');
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x int(4);
declare y char(15);
select sal, job into x, y from emp where ename =
'KING';
insert into tempp values(x, y);
end //
delimiter ;
```

- To see which all procedures are created:-

```
show procedure status; <- shows all procedures in
all schemas

show procedure status where name like 'A%';
```

- To view the source code of stored procedure:-

```
show create procedure abc;
```

Remarks

- -- for Single Line comment
- /* … */ for Multi-Line comment

If statement

```
delimiter //
create procedure abc()
Begin
declare x int(4);
select sal into x from emp where ename = 'KING';
if x > 4000 then
insert into tempp values(x, 'High sal');
end if;
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x int(4);
select sal into x from emp where ename = 'KING';
if x > 5000 then
insert into tempp values(x, 'High sal');
Else
insert into tempp values(x, 'Low sal');
end if;
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x int(4);
select sal into x from emp where ename = 'KING';
if x > 4000 then
insert into tempp values(x, 'High sal');
else
if x < 5000 then
insert into tempp values(x, 'Low sal');
else
insert into tempp values(x, 'Medium sal');
end if;
end  if;
end //
delimiter ;
```

```
delimiter //
create procedure abc()
Begin
declare x int(4);
select sal into x from emp where ename = 'KING';
if x > 5000 then
insert into tempp values(x, 'High sal');
elseif x < 5000 then
insert into tempp values(x, 'Low sal');
else
insert into tempp values(x, 'Medium sal');
end if;
end //
```

```
delimiter ;
```

Case statement

```
delimiter //
create procedure abc()
Begin
declare x int(4);
select sal into x from emp where ename = 'KING';
case
when x > 5000 then
   insert into tempp values(x, 'High sal');
when x < 5000 then
   insert into tempp values(x, 'Low sal');
else
   insert into tempp values(x, 'Medium sal');
end case;
end //
delimiter ;
```

Loops

While loop

```
WHILE expression DO
   ……..;
   ……..;
END WHILE;
```

```
delimiter //
```

```
create procedure abc()
Begin
declare x int default 1;
while x < 10 do
    insert into tempp values(x, 'in while loop');
    set x = x + 1;
end while;
end //
delimiter ;


delimiter //
create procedure abc()
Begin
declare x int default 1;
declare y int default 1;
while x < 10 do
    while y < 10 do
    insert into tempp values(y, 'in y loop');
    set y = y + 1;
    end while;
    insert into tempp values(x, 'in x loop');
    set x = x + 1;
end while;
end //
delimiter ;
```

Repeat loop (similar to Do While loop)

```
REPEAT
    …….;
    …….;
UNTIL expression
END REPEAT;
```

Repeat loop

```
delimiter //
create procedure abc()
Begin
declare x int default 1;
repeat
    insert into tempp values(x, 'in loop');
    set x = x + 1;
    until x > 5
end  repeat;
end //
delimiter ;
```

Stored Functions

```
delimiter //
create function abc()
returns int
Deterministic
Begin
return 10;
end;
//
```

- To see which all functions are created:-

show function status; <- shows all functions in all schemas

show function status where name like 'A%';

- To view the source code of stored function:-

show create function abc;

SQL Exercise 1

1. Create the table SEMP with the following structure:-

EMPNO	CHAR(4)
EMPNAME	CHAR(20)
BASIC	FLOAT
DEPTNO	CHAR(2)
DEPTHEAD	CHAR(4)

2. Create the table SDEPT with the following structure:-

DEPTNO	CHAR(2)
DEPTNAME	CHAR(15)

3. Insert into the SDEPT table the following values:-

10, Development
20, Training

4. Insert into the SEMP table the following values:-

0001, SUNIL, 6000, 10
0002, HIREN, 8000, 20
0003, ALI, 4000, 10, 0001
0004, GEORGE, 6000, 0002

Create S, P, J, SPJ tables as specified below and insert a few rows in each table:-

SUPPLIER
(S#, Sname, Status, City) - S
PARTS
(P#, Pname, Color, Weight, City) - P
PROJECTS
(J#, Jname, City) - J
SUPPLIER-PARTS- PROJECT
(S#, P#, J#, Qty) -
SPJ

Sample data for S# column:- 'S1', 'S2', 'S3', etc.
Sample data for P# column:- 'P1', 'P2', 'P3', etc.
Sample data for J# column:- 'J1', 'J2', 'J3', etc.
Sample data for Status column:- 10, 20, 30, etc.

Write the SELECT queries to do the following:-

5. Display all the data from the S table.

6. Display only the S# and SNAME fields from the S table.

7. Display the PNAME and COLOR from the P table for the CITY=”London”.

8. Display all the Suppliers from London.

9. Display all the Suppliers from Paris or Athens.

10. Display all the Projects in Athens.

11. Display all the Partnames with the weight between 12 and 14 (inclusive of both).

12. Display all the Suppliers with a Status greater than or equal to 20.

13. Display all the Suppliers except the Suppliers from London.

14. Display only the Cities from where the Suppliers come from.

15. Assuming that the Part Weight is in GRAMS, display the same in MILLIGRAMS and KILOGRAMS.

SQL Exercise 2

1. Display the Supplier table in the descending order of CITY.

2. Display the Part Table in the ascending order of CITY and within the city in the ascending order of Part names.

3. Display all the Suppliers with a status between 10 and 20.

4. Display all the Parts and their Weight, which are not in the range of 10 and 15.

5. Display all the Part names starting with the letter ‘S’.

6. Display all the Suppliers, belonging to cities starting with the letter ‘L’.

7. Display all the Projects, with the third letter in JNAME as ‘n’.

SQL Exercise 3

1. Display all the Supplier names with the initial letter capital.

2. Display all the Supplier names in upper case.

3. Display all the Supplier names in lower case.

4. Display all the Supplier names padded to 25 characters, with spaces on the left.

5. Display all the Supplier names (with 'la' replaced by 'ro').
   HINT: REPLACE.

6. Implement the above command such that 'l' is replaced with 'r' and 'a' is replaced with 'o'.

7. Display the Supplier names and the lengths of the names.

8. Use the soundex function to search for a supplier by the name of 'BLOKE'.

9. Display the Supplier name and the status (as Ten, Twenty, Thirty, etc.).

10. Display the current day (e.g. Thursday).

SQL Exercise 4

1. Display the minimum Status in the Supplier table.

2. Display the maximum Weight in the Parts table.

3. Display the average Weight of the Parts.

4. Display the total Quantity sold for part 'P1'.

5. Display the total Quantity sold for each part.

6. Display the average Quantity sold for each part.

7. Display the maximum Quantity sold for each part, provided the maximum Quantity is greater than 800.

8. Display the Status and the count of Suppliers with that Status.

9. Display the count of Projects going on in different cities.

10. What is the difference between COUNT(Status) and COUNT(*) ?

11. Display the Status and the Count of Suppliers with that Status in the following format as shown below:-

Status	Count
Ten	1
Twenty	2
Thirty	3

SQL Exercise 5

1. Display the Supplier name and the Quantity sold.

2. Display the Part name and Quantity sold.

3. Display the Project name and Quantity sold.

4. Display the Supplier name, Part name, Project name and Quantity sold.

5. Display the Supplier name, Supplying Parts to a Project in the same City.

6. Display the Part name that is 'Red' is color, and the Quantity sold.

7. Display all the Quantity sold by Suppliers with the Status = 20.

8. Display all the Parts and Quantity with a Weight > 14.

9. Display all the Project names and City, which has bought more than 500 Parts.

10. Display all the Part names and Quantity sold that have a Weight less than 15.

11. Display all the Employee names and the name of their Managers.

SQL Exercise 6

1. Display all the Suppliers with the same Status as the supplier, 'CLARK'.

2. Display all the Employees in the same department as the employee 'MILLER'.

3. Display all the Parts which have more Weight than all the Red parts.

4. Display all the Projects going on in the same city as the project 'TAPE'.

5. Display all the Parts with Weight less than all the Green parts.

6. Display the name of the Supplier who has sold the maximum Quantity (in one sale).

7. Display the name of the Employee with the minimum Salary.

8. Display the name of the Supplier who has sold the maximum overall Quantity (sum of Sales).

9. Display the name of the Department with the maximum number of Employees.

MySQL-PL Exercise 1

1. Write a program that computes the perimeter and the area of a rectangle. Define your own values for the length and width. (Assuming that L and W are the length and width of the rectangle, Perimeter = 2*(L+W) and Area = L*W.

2. Convert a temperature in Fahrenheit (F) to its equivalent in Celsius (C) and vice versa. Therequired formulae are:- C= (F-32)*5/9
F= 9/5*C + 32

3. Write a program that enables a user to input an integer. The program should then state whether the integer is evenly divisible by 5.

MySQL-PL Exercise 2

1. Select from any table a number and determine whether it is within a given range (for example, between 1 and 10).

2. Select from any table three positive integers representing the sides of a triangle, and determine whether they form a valid triangle. Hint: In a triangle, the sum of any two sides must always be greater than the third side.

3. Check if a given a year is a leap year. The condition is:- year should be (divisible by 4 and not divisible by 100) or (divisible by 4 and divisible by 400.). The year should be Selected from some table.

MySQL-PL Exercise 3

1. Write a program containing a loop that iterates from 1 to 1000 using a variable I,which is incremented each time around the loop. The program should output the value of I every hundred iterations (i.e., the output should be 100, 200, etc.).

2. Write a program that Selects from any table a minimum and maximum value for a radius, along with an increment factor, and generates a series of radii by repeatedly adding the increment to the minimum until the maximum is reached. For each value of the radius, compute and display the circumference, area, and volume of the sphere. (Be sure to include both the maximum and the minimum values.).

3. A palindrome is a word that is spelled the same forward and backward, such as level, radar, etc. Write a program to Selects from any table a five letter word and determine whether it is a palindrome.

MySQL-PL Exercise 4

1. Write a stored function to take three parameters, the sides of a triangle. The sides of the triangle should be accepted from the user. The function should return a Boolean value:- trueif the triangle is valid, false otherwise. A triangle is valid if the length of each side is

less than the sum of the lengths of the other two sides. Check if the dimensions entered can forma valid triangle.

2. Write a function that generates a random number between 1 and 10. Use any logic of your choice to achieve this.

3. Create a function that accepts a string of n characters and exchanges the first character with the last, the second with the next – to – last, and so forth until n exchanges have been made. What will the final string look like? Write the function to verify your conclusion.

MySQL-PL Exercise 5

1. Write a stored procedure by the name of Comp_intr to calculate the amount of interest ona bank account that compounds interest yearly. The formula is:- I
= p (1+ r) y – p where:-
I is the total interest earned.
p is the principal.
r is the rate of interest as a decimal less than 1, and y is the number of years themoney is earning interest.

Your stored procedure should accept the values of p, r and y as parameters and insert the Interest and Total amount into tempp table.

2. Create a stored function by the name of Age_calc. Your stored function should accept the date of birth of a person as a parameter. The stored function should calculate the age of the person in years. The stored function should return the age in years.

SQL Assignment – 1

Creating sample tables and inserting values.

Create the following tables with the given structures and insert sample data as specified: -

A) SALESPEOPLE

Snum	int(4)
Sname	varchar(10)
City	varchar(10)
Comm	float(3,2)

B) CUSTOMERS

Cnum	int(4)
Cname	varchar(10)
City	varchar(10)
Rating	int(4)
Snum	int(4)

C) ORDERS

Onum	int(4)
Amt	float(7,2)
Odate	date
Cnum	int(4)
Snum	int(4)

Data

SALES PEOPLE

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	10

CUSTOMERS

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007
2007	Pereira	Rome	100	1004

ORDERS

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	03-OCT-1990	2008	1007
3003	767.19	03- OCT -1990	2001	1001
3002	1900.10	03- OCT -1990	2007	1004
3005	5160.45	03- OCT -1990	2003	1002
3006	1098.16	03- OCT -1990	2008	1007
3009	1713.23	04- OCT -1990	2002	1003
3007	75.75	04- OCT -1990	2004	1002
3008	4723.00	05- OCT -1990	2006	1001
3010	1309.95	06- OCT -1990	2004	1002
3011	9891.88	06- OCT -1990	2006	1001

SQL Assignment – 2

Introducing Relational Databases.

1) Which field of the Customers table is the primary key?

2) What is the 4th column of the Customers table?

3) What is another word for row? For column?

4) Why isn't it possible to see the first five rows of a table?

SQL Assignment – 3

Overview of SQL.

1) Does ANSI recognize the data type DATE?

2) Which subdivision of SQL is used to insert values in tables?

SQL Assignment – 4

Retrieving Information from Tables.

1) Write a select command that produces the order number, amount, and date for all rows in the Orders table.

2) Write a query that produces all rows from the Customers table for which the salesperson's number is 1001.

3) Write a query that displays the Salespeople table with the columns in the following order: city, sname, snum, comm.

4) Write a select command that produces the rating followed by the name of each customer in San Jose.

SQL Assignment – 5

Relational and Logical Operators.

1) Write a query that will give you all orders for more than Rs. 1,000.

2) Write a query that will give you the names and cities of all salespeople in London with a commission above .10.

3) Write a query on the Customers table whose output will exclude all customers with a rating <= 100, unless they are located in Rome.

4) What will be the output from the following query?

```
Select * from Orders
 where (amt < 1000 OR
  NOT (odate = '1990-10-03'
                    AND cnum > 2003));
```

5) What will be the output of the following query?

```
Select * from Orders
 where NOT ((odate = '1990-10-03' OR snum
                    >1006) AND amt >= 1500);
```

6) What is a simpler way to write this query?

```
Select snum, sname, city, comm From Salespeople   where
(comm > .12 OR comm <.14);
```

SQL Assignment – 6

Using Special Operators in Conditions.

1) Write two different queries that would produce all orders taken on October 3rd or 4th, 1990.

2) Write a query that selects all of the customers serviced by Peel or Motika. (Hint: the snum field relates the two tables to one another).

3) Write a query that will produce all the customers whose names begin with a letter from 'A' to 'G'.

4) Write a query that selects all customers whose names begin with the letter 'C'.

5) Write a query that selects all orders except those with zeroes or NULLs in the amt field.

SQL Assignment – 7

Summarizing Data with Aggregate Functions.

1) Write a query that counts all orders for October 3.

2) Write a query that counts the number of different non-NULL city values in the Customers table.

3) Write a query that selects each customer's smallest order.

4) Write a query that selects the first customer, in alphabetical order, whose name begins with G.

5) Write a query that selects the highest rating in each city.

6) Write a query that counts the number of salespeople registering orders for each day. (Ifa salesperson has more than one order on a givenday, he or she should be counted only once.).

SQL Assignment – 8

Formatting Query output.

1) Assume each salesperson has a 12% commission. Write a query on the orders table that will produce the order number, the salesperson number, and the amount of the salesperson's commission for that order.

2) Write a query on the Customers table that will find the highest rating in each city. Put the output in this form:

For the city (city), the highest rating is : (rating).

3) Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer's name and number.

4) Write a query that totals the orders for each day and places the results in descending order.

SQL Assignment – 9

Querying Multiple Tables at Once.

1) Write a query that lists each order number followed by the name of the customer who made the order.

2) Write a query that gives the names of both the salesperson and the customer for each order along with the order number.

3) Write a query that produces all customers serviced by salespeople with a commission above 12%. Output the customer's name, the salesperson's name, and the salesperson's rate of commission.

4) Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

SQL Assignment – 10

Joining a Table to Itself.

1) Write a query that produces all pairs of salespeople who are living in the same city. Exclude combinations of salespeople with themselves as well as duplicate rows with the order reversed.

2) Write a query that produces the names and cities of all customers with the same rating as Hoffman.

SQL Assignment – 11

Subqueries

1) Write a query that uses a subquery to obtain all orders for the customer named Cisneros. Assume you do not know his customer number (cnum).

2) Write a query that produces the names and ratings of all customers who have above-average orders.

3) Write a query that selects the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.

SQL Assignment – 12

Using the operators IN, ANY, and ALL.

1) Write a query that selects all customers whose ratings are equal to or greater than ANY of Serres'.

2) Write a query using ANY or ALL that will find all salespeople who have no customers located in their city.

3) Write a query that selects all orders for amounts greater than any for the customers in London.

4) Write the above query using MIN or MAX.

SQL Assignment – 13

Using the UNION clause.

1) Create a union of two queries that shows the names, cities, and ratings of all customers. Those with rating of 200 or greater will also have the words "High Rating", while the others will have the words "Low Rating".

2) Write a command that produces the name and number of each salesperson and each customer with more than one current order. Put the results in alphabetical order.

3) Form a union of three queries. Have the first select the snums of all salespeople in San Jose; the second, the cnums of all customers in San Jose; and the third the onums of all orders on October 3. Retain duplicates between the last two queries but eliminate any redundancies between either of them and the first.
(Note: in the sample tables as given, there would be no such redundancy. This is besides the point.)

SQL Assignment – 14

Entering, Deleting, and Changing Field Values.

1) Write a command that puts the following values, in their given order, into the salespeople table: city – San Jose, name – Blanco, comm – NULL, cnum – 1100.

2) Write a command that removes all orders from customer Clemens from the Orders table.

3) Write a command that increases the rating of all customers in Rome by 100.

4) Salesperson Serres has left the company.Assign her customers to Motika.

SQL Assignment – 15

Using Subqueries with DML Commands.

1) Assume there is a table called Multicust, with all of the same column definitions as Salespeople. Write a command that inserts all salespeople with more than one customer into this table.

2) Write a command that deletes all customers with no current orders.

3) Write a command that increases by twenty percent the commissions of all salespeople with total orders above Rs. 3,000.

MongoDB Assignment

1. Install MongoDB database server
2. Create a directory/folder to store MongoDB datafiles
3. Set necessary path environment variable
4. Startup MongoDB database
5. Connect to MongoDB database through the MongoDB shell command line interface
6. View list of available databases
7. Create a new database named CDAC and connect to it
8. View list of available collections in CDAC database
9. Create a new collection by the name of LIBRARY
10. Insert the following document in the LIBRARY collection:-
    title:'MongoDB programming',
    author:'Sameer', likes:100
11. View the recently inserted document and note the _id field
12. Insert another document in the LIBRARY collection as follows:-
    title:'MySQL programming',
    authors:['Jack','Jill'], likes:200
13. View the inserted documents
14. View only the first inserted document
15. View the documents using the pretty() method
16. Update the document where author name Sameer and change it to Sameer Dehadrai
17. Delete all documents that have 100 likes
18. Drop the LIBRARY collection
19. Drop the CDAC database
20. Exit from MongoDB shell
21. Stop MongoDB server