**Database Management System ASSIGNMENT-1**

1. Write a PL/SQL program to check whether a given character is letter or digit.

CREATE PROCEDURE `q1`(x char)

BEGIN

declare y int;

set y := ascii(x);

IF y between 97 and 122 or y between 65 and 90 then

select "Entered value is a letter!" as "Message Box";

ELSEIF y between 48 and 57 then

select "Entered value is a digit!" as "Message Box";

ELSE

select "Entered value is a invalid!" as "Message Box";

end if;

END

1. Write a program in PL/SQL to update the salary of a specific employee by 5% if the salary exceeds the mid-range of the salary against this job and update up to mid-range if the salary is less than the mid-range of the salary, and display a suitable message.

Note: Create a table named DEMOEMP just like EMP table with the same records.

CREATE TABLE DEMOEMP AS SELECT \* FROM EMP;

CREATE PROCEDURE `q7`(jb varchar(20))

BEGIN

declare a float;

SELECT AVG(SAL) INTO A FROM DEMOEMP WHERE JOB = JB;

UPDATE DEMOEMP SET SAL = IF(SAL<A, A, SAL + SAL\*.5) WHERE JOB = JB;

END

1. Write a program in PL/SQL to insert records from one table to another.

CREATE PROCEDURE `q3`()

BEGIN

declare vEMPNO, vMGR, vSAL, vCOMM, vDEPTNO, vBONUSID int;

declare vENAME, vJOB, vUSERNAME, vPWD varchar(50);

declare vhiredate date;

declare visActive bool;

declare c1 cursor for SELECT \* from EMP;

declare exit handler for NOT FOUND select "data transfered!" as "Message Box";

OPEN c1;

loopLabel: loop

FETCH c1 into vEMPNO, vENAME, vJOB, vMGR, vhiredate, vSAL, vCOMM, vDEPTNO, vBONUSID, vUSERNAME,

vPWD, visActive;

INSERT INTO demoemp values (vEMPNO, vENAME, vJOB, vMGR, vhiredate, vSAL, vCOMM, vDEPTNO, vBONUSID, vUSERNAME, vPWD, visActive);

end loop loopLabel;

CLOSE c1;

END

1. Write a program in PL/SQL to print the prime numbers between 1 to 50.

CREATE DEFINER=`root`@`localhost` PROCEDURE `q4`()

BEGIN

DECLARE i, j int;

select 'The prime numbers are:';

set i := 2;

lbl1:LOOP

set j := 2;

lbl2:LOOP

IF MOD (i, j) = 0 OR j = i then

leave lbl2;

END IF;

set j := j + 1;

END LOOP lbl2;

IF(j = i)THEN

select concat(i,' ');

END IF;

set i := i + 1;

IF i = 50 then

leave lbl1;

END IF;

END LOOP lbl1;

END

1. From the given table find

ord\_no purch\_amt ord\_date customer\_id salesman\_id

---------- --------------- ---------------- ---------------- -----------------

70001 160.5 2012-10-05 3005 5002

70009 370.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001

70004 110.5 2012-08-17 3009 5003

70007 948.5 2012-09-10 3005 5002

70005 2400.6 2012-07-27 3007 5001

70008 5760 2012-09-10 3002 5001

70010 1983.43 2012-10-10 3004 5006

70003 2480.4 2012-10-10 3009 5003

70012 250.45 2012-06-27 3008 5002

70011 75.29 2012-08-17 3003 5007

70013 3045.6 2012-04-25 3002 5001

CREATE TABLE neworders (ord\_no int, purch\_amt float, ord\_date date, customer\_id int, salesman\_id int, PRIMARY KEY (ord\_no));

INSERT INTO neworders VALUES ('70001', '160.5', '2012-10-05', '3005', '5002');

INSERT INTO neworders VALUES ('70009', '370.65', '2012-09-10', '3001', '5005');

INSERT INTO neworders VALUES ('70002', '65.26', '2012-10-05', '3002', '5001');

INSERT INTO neworders VALUES ('70004', '110.5', '2012-08-17', '3009', '5003');

INSERT INTO neworders VALUES ('70007', '948.5', '2012-09-10', '3005', '5002');

INSERT INTO neworders VALUES ('70005', '2400.6', '2012-07-27', '3007', '5001');

INSERT INTO neworders VALUES ('70008', '5760', '2012-09-10', '3002', '5001');

INSERT INTO neworders VALUES ('70010', '1983.43', '2012-10-10', '3004', '5006');

INSERT INTO neworders VALUES ('70003', '2480.4', '2012-10-10', '3009', '5003');

INSERT INTO neworders VALUES ('70012', '250.45', '2012-06-27', '3008', '5002');

INSERT INTO neworders VALUES ('70011', '75.29', '2012-08-17', '3003', '5007');

INSERT INTO neworders VALUES ('70013', '3045.6', '2012-04-25', '3002', '5001');

INSERT INTO neworders VALUES ('80009', '2370.65', '2012-09-10', '3001', '5005');

INSERT INTO neworders VALUES ('80008', '3760', '2012-09-10', '3001', '5001');

INSERT INTO neworders VALUES ('80007', '9348.5', '2012-09-10', '3002', '5002');

INSERT INTO neworders VALUES ('80002', '615.26', '2012-10-05', '3002', '5001');

INSERT INTO neworders VALUES ('80001', '1360.5', '2012-10-05', '3003', '5002');

INSERT INTO neworders VALUES ('80010', '1980.43', '2012-10-10', '3001', '5006');

INSERT INTO neworders VALUES ('80003', '2490.4', '2012-10-10', '3002', '5003');

1. Write a SQL statement to find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

mysql> select salesman\_id, max(purch\_amt) from neworders where ord\_date ='2012-08-17' group by salesman\_id;

1. Write a SQL statement to find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount

mysql> select customer\_id, ord\_date, max(purch\_amt) from neworders group by customer\_id, ord\_date order by customer\_id, ord\_date;

1. Write a SQL statement to find the highest purchase amount ordered by each customer with their ID and highest purchase amount

mysql> select customer\_id, max(purch\_amt) from neworders group by customer\_id;

1. Write a SQL statement to find the highest purchase amount with their ID and order date, for only those customers who have highest purchase amount in a day is more than 2000.

mysql> select customer\_id, ord\_date, max(purch\_amt) from neworders group by ord\_date, customer\_id having max(purch\_amt) > 2000;

1. Write a SQL statement to find the highest purchase amount with their ID, for only those customers whose ID is within the range 3002 and 3007

mysql> select customer\_id, ord\_date, max(purch\_amt) from neworders where customer\_id between 3002 and 3007 group by ord\_date, customer\_id;

1. a) Write a program in PL/SQL to create an explicit cursor with for loop.

CREATE PROCEDURE ` q6a`()

BEGIN

declare vDEPTNO int;

declare vDNAME, vLOC, vPWD varchar(50);

declare c1 cursor for SELECT \* from dept;

declare exit handler for NOT FOUND select "Done!" as "Message Box";

OPEN c1;

loopLabel: loop

FETCH c1 into vDEPTNO, vDNAME, vLOC, vPWD;

select vDEPTNO, vDNAME, vLOC, vPWD;

end loop loopLabel;

CLOSE c1;

END

b) Write a program in PL/SQL to create an implicit cursor with for loop.

CREATE PROCEDURE `q6b`(dnm varchar(20))

BEGIN

declare vEMPNO, vMGR, vSAL, vCOMM, vDEPTNO, vBONUSID, a int;

declare vENAME, vJOB, vUSERNAME, vPWD varchar(50);

declare vhiredate date;

declare visActive bool;

declare c1 cursor for SELECT \* from EMP where deptno = a;

declare exit handler for NOT FOUND select "Done!" as "Message Box";

select deptno into a from DEPT where dname = dnm;

OPEN c1;

loopLabel: loop

FETCH c1 into vEMPNO, vENAME, vJOB, vMGR, vhiredate, vSAL, vCOMM, vDEPTNO, vBONUSID, vUSERNAME, vPWD, visActive;

select vEMPNO, vENAME, vJOB, vMGR, vhiredate, vSAL, vCOMM, vDEPTNO, vBONUSID, vUSERNAME, vPWD, visActive;

end loop loopLabel;

CLOSE c1;

END

1. Write a block in PL/SQL to print the specific number of rows from a table.

CREATE PROCEDURE `q8`(x int)

BEGIN

declare vEMPNO, vMGR, vSAL, vCOMM, vDEPTNO, vBONUSID int;

declare vENAME, vJOB, vUSERNAME, vPWD varchar(50);

declare vhiredate date;

declare visActive bool;

declare c1 cursor for SELECT \* from EMP limit x;

declare exit handler for NOT FOUND select "Done!" as "Message Box";

OPEN c1;

loopLabel: loop

FETCH c1 into vEMPNO, vENAME, vJOB, vMGR, vhiredate, vSAL, vCOMM, vDEPTNO, vBONUSID,

vUSERNAME, vPWD, visActive;

select vEMPNO, vENAME, vJOB, vMGR, vhiredate, vSAL, vCOMM, vDEPTNO, vBONUSID, vUSERNAME, vPWD, visActive;

end loop loopLabel;

CLOSE c1;

END

1. Write a block in PL/SQL to print a report which shows that, the employee id, name, hire date, and the incentive amount they achieved according to their working experiences, who joined in the month of current date.