DBMS Queries :

Mysql -u root -p

Ass1: ddl

Create database property\_dealer;

Use property\_dealer;

Create table table\_name() values ();

Show tables;

Select \* from table\_name;

Select client.iname from client join privateowner on client.fname = privateowner.fname and client.iname = privateowner.iname

Select \* from staff where fname like’ %a%a%’;

Select fname ,max(salary) as max\_salary from staff group by fname order by max\_salary desc;

Select privateowner.fname , privateowner.iname from privateowner left join client on privateowner.fname = client.fname and privateowner.iname = client.iname where client.cuid is null;

Select \* from privateowner where fname = ‘\_c%’ or iname=’\_c%’;

Select fname,iname from staff where position =’saleswomen’ or position = ‘salesmens’;

Select \*from client where rent>1000;

Ass 2:DML

For setting the foreign key we need to set up the primary in the other table

Salerepresentative -> customers-> orders for the foreign key

Select c.company ,SR.name ,SR.RE-OFFICE FROM CUSTOMER C JOIN SALESREPRENSTIVE SR WHERE C.CUST\_REP = SR.REP\_NO;

SELECT O.ORDER\_NO,C.COMAPANY,SR.NAME ,O.AMOUNT FROM ORDER O JOIN CUSTOMER C ON O.CUST = C.CUST\_NO

JOIN SELESREPRESENTIVE SR ON C.CUST\_NO= SR.REP\_NO

SELECT DISTINCT O.PRODUCT

FROM ORDER O

JOIN CUSTOMER C ON O.CUST = C.CUST\_NO

WHERE C.COMPANY = 'TCS';

SELECT C.COMAPANY ,MAX(O.DISC) AS MAX\_DISC

FROM ORDER O

JOIN CUSTOMER C ON C.CUST\_NO = O.CUST

GROUP BY C.COMPANY

ORDER BY MAX\_DISC DESC

LIMIT 1;

SELECT SR1.NAME,SR2.NAME

FROM SALESREPRESENTATIVE SR1

JOIN SALESREPRESENTATION SR2 ON SR1.RE\_OFFICE = SR2.RE\_OFFICE

WHERE SR1.REP\_NO != SR2.REP\_NO

SELECT C.COMAPANY ,C.CREDIT\_LIMIT,O.DISC

FROM ORDER O

JOIN CUSTOMER C ON O.CUST = C.CUST\_NO;

SELECT C.COMPANY ,C.CREDIT\_LIMIT,SR.NAME,SR.SALES FROM CUSTOMER C JOIN SALESREPRESENTATIVE SR ON C.CUST\_REP = SR.REP\_NO;

SELECT O.ORDER\_NO,O.AMOUNT,C.COMPANY,C.CREADIT\_LIMIT FROM ORDER O JOIN CUSTOMER C ON O.CUST = C.CUST\_NO;

SELECT O.PRODUCT ,SUM(O.AMOUNT) AS SALES\_AMOUNT

FROM ORDERS O

GROUP BY O.PRODUCT

HAVING SUM(O.AMOUNT)>12000;

SELECT O.AMOUNT ,C.COMPANY ,SR.NAME

FROM ORRDERS O

JOIN CUSTOMER C ON O.CUST = C.CUST\_NO

JOIN SALES\_REPRESENTATION SR ON C.CUST\_REP = SR.REP\_NO

WHERE O.PRODUCT =’SOFTWARE’;

SELECT C.COMPANY , C.CREDIT\_LIMIT,O.DISC

FROM ORDERS O

JOIN CUSTOMERS C ON O.CUST = C.CUST\_NO;

SELECT SR1.NAME , SR2.NAME

FROM SALES\_REPRESENTATION SR1

JOIN SALES\_REPRESENTATION SR2 ON SR1.RE\_OFFICE = SR2.RE\_OFFICE

WHERE SR1.REP\_NO != SR2.REP\_NO;

**ASS3:**

CREATE PROCEDURE FINECAL(IN N\_ROLLNO INT ,IN N\_NAMEOFBOOK VARCHAR(50)) BEGIN

DECLARE N\_DATEOFISSUE DATE,

DECLARE N\_STATUS VARCHAR(40),

DECLARE N\_CURRENTDATE DATE ,

DECLARE N\_FINEAMT INT ,

DECLARE N\_DAYS\_DIFF INT ;

DECLARE CONTINUE HANDLER FOR NOT FOUND BEGIN

SELECT ‘NO RECORD FOUND FOR THE PROVIDED ROLLNO AND NAMEOFBOOK.’ AS MESSAGE

END;

SET N\_CURRENTDATE = CURDATE();

SELECT DATAOFISSUES,STATUS INTO

N\_DATEOFISSUE,N\_STUTAS

FROM BORROWER

WHERE ROLLNO = N\_ROLLNO AND NAMEOFBOOK = N\_NAMEOFBOOK;

SET N\_DAYS\_DIFF = DATEDIFF(N\_CURRENTDATE,N\_DATEOFISSUE);

IF N\_DAYS\_DIFF> 30 THEN SET N\_FINEAMT := 5\*30+50\*(N\_DAY\_DIFF - 30);

ELSEIF N\_DAYS\_DIFF BETWEEN 15 AND 30 THEN

SET N\_FINEAMT := 5\*N\_DAYS\_DIFF;

ELSE SET N\_FINEAMT :=0;

END IF;

IF N\_FINEAMT>0 THEN INSERT INTO FINE(ROLLNO,DATE,AMOUNT) VALUES (N\_ROLLNO,N\_CURRENTDATE,N\_FINEAMT);

END IF;

END//

DELIMITER;

**CALLING STORE PROCEDURE :**

CALL FINECALL(01,’CNS’);

**ASS4 :**

DELIMITER \\

CREATER PROCEDURE PROC\_GRADE()

BEGIN

DECLARE DONE INT DEFAULT SIZE ;

DECLARE V\_NAME VARCHAR(100);

DECLARE V\_ROLL\_NO INT ;

DECLARE V\_TOTAL\_MARKS INT ;

DECLARE V\_CLASS VARCHAR(50);

DECLARE CUR CURSOR FOR SELECT NAME ,ROLL\_NO , TOTAL\_MARKS FROM STUD\_MARKS;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;

OPEN CUR;

READ\_LOOP:LOOP

FETCH CUR INTO V\_NAME ,V\_ROLL\_NO,V\_TOTAL\_MARKS;

IF DONE THEN

LEAVE READ\_LOOP;

END IF;

IF V\_TOTAL\_MARKS<=1500 AND V\_TOTAL\_MARKS>=900 THEN

SET V\_CLASS = ‘DISTINCTION’;

ELSEIF V\_TOTAL\_MARKS BETWEEN 900 AND 989

SET v\_CLASS =’FIRST CLASS’;

ELSEIF V\_TOTAL\_MARKS BETWEEN 825 AND 899

SET v\_CLASS =’HIGHER SECOND CLASS’;

ELSEIF V\_TOTAL\_MARKS BETWEEN 750 AND 825

SET v\_CLASS =’PASS’;

ELSE

SET V\_CLASS =’FAIL’;

End if

INSERT INTO RESULT (NAME,ROLL\_NO,CLASS)VALUES(V\_NAME,V\_ROLL\_NO,V\_CLASS);

END LOOP ;

CLOSE CUR;

END//

DELIMITER ;

CALL PROC\_GRADE();

DROP PROCEDURE IF EXISTS PROC\_GRADE;