SET 3

-----

1. Write a stored procedure that accepts the month and year as inputs and prints the ordernumber, orderdate and status of the orders placed in that month.

***Example***: call order\_status(2005, 11);

ANS: CREATE PROCEDURE Order\_status (IN in\_month DATETIME , IN in\_year YEAR , OUT in\_status VARCHAR(50) )

Begin

SELECT \*, year(orderdate), month (orderdate) FROM orders

WHERE month = in\_month AND year = in\_year ;

End //

Call order\_status (2003,01);

2. Write a stored procedure to insert a record into the cancellations table for all cancelled orders.

STEPS:

1. Create a table called cancellations with the following fields

id (primary key),

customernumber (foreign key - Table customers),

ordernumber (foreign key - Table Orders),

comments

All values except id should be taken from the order table.

ANS : create table cancellations(id int primary key,

foreign key customernumber references customers),

foreign key ordernumber references Orders));

b. Read through the orders table . If an order is cancelled, then put an entry in the cancellations table.

ANS: Select order\_id from orders o

Where Not Exists

(Select \* From order\_products

Where order\_id = o.order\_id

And cancelled = 1)

3. a. Write function that takes the customernumber as input and returns the purchase\_status based on the following criteria . [table:Payments]

if the total purchase amount for the customer is < 25000 status = Silver, amount between 25000 and 50000, status = Gold

if amount > 50000 Platinum

ANS : CREATE FUNCTION purchase\_status (

cid int

)

RETURNS VARCHAR(20)

DETERMINISTIC

BEGIN

DECLARE status VARCHAR(20);

DECLARE credit numeric;

SET credit = (select sum(Amount) from Payment where customerNumber = cid);

IF credit > 50000 THEN

SET status = 'platinum';

ELSEIF (credit >= 25000 AND

credit <= 50000) THEN

SET status = 'gold';

ELSEIF credit < 25000 THEN

SET status = 'silver';

END IF;

RETURN (status);

END

b. Write a query that displays customerNumber, customername and purchase\_status from customers table.

ANS: Select customerNumber, customername , purchase\_status FROM Customer;

4. Replicate the functionality of 'on delete cascade' and 'on update cascade' using triggers on movies and rentals tables. Note: Both tables - movies and rentals - don't have primary or foreign keys. Use only triggers to implement the above.

ANS : Delete Cascade

   Where memid IN(select id FROM rental)ON DELETE CASCADE );

DELETE FROM movies WHERE id = 102;

Update Cascade

ALTER TABLE Rental ADD CONSTRAINT `rentalTable`

UPDATE movies SET  id = 102 WHERE  id = 103;

5. Select the first name of the employee who gets the third highest salary. [table: employee]

ANS : Select f\_name, MAX(salary) AS maxSal

From employee

Group by f\_name

Order by maxSal

Limit 2,1;

6. Assign a rank to each employee based on their salary. The person having the highest salary has rank 1. [table: employee]

ANS : SELECT

firstName,

lastName,

salary,

RANK(), OVER (ORDER BY salary) salaryRank

FROm employee;