## **ADVANCE DATABASE MANAGEMENT SYSTEM**

**Group: Neon**

**Movie Ticket Management System**

## Table Creation:

CREATE TABLE LOGIN( LOGINID VARCHAR(10) PRIMARY KEY, PASSWORD VARCHAR(20) ,FLAG NUMBER(1));

CREATE TABLE EMPLOYEE( EMPID VARCHAR(10) PRIMARY KEY, EMPNAME VARCHAR(40) ,LOGINID VARCHAR(10) ,ROLE VARCHAR(20) ,SALARY NUMBER(15));

CREATE TABLE CUSTOMER( CUSTOMERID VARCHAR(10) PRIMARY KEY, CUSTOMERNAME VARCHAR(40) ,LOGINID VARCHAR(10) ,EMAIL VARCHAR(40) );

CREATE TABLE MOVIE ( MID VARCHAR(10) PRIMARY KEY, MOVIENAME VARCHAR(50), RELEASEYEAR NUMBER(4), EMPID VARCHAR(10) );

CREATE TABLE MTIME (MCODE VARCHAR(10) PRIMARY KEY, DAY VARCHAR(10), MSLOT NUMBER(1),MID VARCHAR(10), THEATERID VARCHAR(10) , AVAILSEAT NUMBER(3));

CREATE TABLE SLOT( MSLOT NUMBER(1) PRIMARY KEY, STARTTIME NUMBER(4) ,ENDTIME NUMBER(4));

CREATE TABLE THEATER (THEATERID VARCHAR(10) PRIMARY KEY, THEATERNAME VARCHAR(20) , CAPACITY NUMBER(3));

CREATE TABLE TICKET (TRANSID VARCHAR(10) PRIMARY KEY , MCODE VARCHAR(10) ,CUSTOMERID VARCHAR(10) , EMPID VARCHAR(10), SEAT NUMBER(3));

CREATE TABLE CHANGEDPASS (LOGINID VARCHAR(10), OLDPASSWORD VARCHAR(40), NEWPASSWORD VARCHAR(10), CHANGEDDATE DATE);

CREATE TABLE DELETED\_CUSTOMER(OLDCUSTOMERID VARCHAR(10), OLDCUSTOMERNAME VARCHAR(40), OLDLOGINID VARCHAR(10), OLDEMAIL VARCHAR(10));

CREATE TABLE DELETEDMOVIE(NEWMID VARCHAR(10), NEWMOVIENAME VARCHAR(40), DELETEDDATE DATE);

ALTER TABLE MOVIE DROP COLUMN EMPID ;

## Value Insertion & Views:

INSERT INTO LOGIN VALUES ('E1001', '1111' ,0) ;

INSERT INTO LOGIN VALUES ('E1002', '1111' ,1) ;

INSERT INTO LOGIN VALUES ('E1003', '1111' ,1) ;

INSERT INTO LOGIN VALUES ('E1004', '1111' ,1) ;

INSERT INTO LOGIN VALUES ('E1005', '1111' ,1) ;

INSERT INTO LOGIN VALUES ('C1001', '1111' ,2) ;

INSERT INTO LOGIN VALUES ('C1002', '1111' ,2) ;

INSERT INTO LOGIN VALUES ('C1003', '1111' ,2) ;

INSERT INTO LOGIN VALUES ('C1004', '1111' ,2) ;

INSERT INTO LOGIN VALUES ('C1005', '1111' ,2) ;

SELECT \* FROM LOGIN;

INSERT INTO EMPLOYEE VALUES ('E10001','MUNTASIR','E10001','MANAGER', 50000);

INSERT INTO EMPLOYEE VALUES ('E10002',RAHIN,'E10002','SELLER', 20000);

INSERT INTO EMPLOYEE VALUES ('E10003',ZINNET,'E10003','SELLER', 25000);

INSERT INTO EMPLOYEE VALUES ('E10004',MORAD,'E10004','SELLER', 15000);

INSERT INTO EMPLOYEE VALUES ('E10005','RAJU','E10005','SELLER', 27000);

SELECT \* FROM EMPLOYEE ;

INSERT INTO CUSTOMER VALUES ('C10001','ALEX','C10001','ALEX@GMAIL.COM');

INSERT INTO CUSTOMER VALUES ('C10002','JOHN','C10002','JOHN@GMAIL.COM');

INSERT INTO CUSTOMER VALUES ('C10003','THOMAS','C10003','THOMAS@GMAIL.COM');

INSERT INTO CUSTOMER VALUES ('C10004','PITTER','C10004','PITTER@GMAIL.COM');

INSERT INTO CUSTOMER VALUES ('C10005','RAJU','C10005','RAJU@GMAIL.COM');

SELECT \* FROM CUSTOMER ;

INSERT INTO MOVIE VALUES ('M2001','CAPTAIN MARVEL',2019,'E10004');

INSERT INTO MOVIE VALUES ('M2002','BLACK PANTHER',2018,'E10001');

INSERT INTO MOVIE VALUES ('M2003','ROMA',2018,'E10005');

INSERT INTO MOVIE VALUES ('M2004','DEBI',2018,'E10004');

INSERT INTO MOVIE VALUES ('M2005','THE GODFATHER',1972,'E10004');

INSERT INTO MOVIE VALUES ('M2006','LA LA LAND',2016,'E10004');

SELECT \* FROM MOVIE ;

INSERT INTO MTIME VALUES ('MC5001','SUNDAY',1 ,'M2003','TH4002' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5002','MONDAY',3 ,'M2005','TH4003' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5003','TUESDAY',2 ,'M2005','TH4001' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5004','THURSDAY',1 ,'M2001','TH4004' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5005','FRIDAY',3 ,'M2006','TH4001' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5006','FRIDAY',1 ,'M2002','TH4001' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5007','SUNDAY',3 ,'M2004','TH4004' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5008','WEDNESDAY',3,'M2004','TH4005' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5009','SUNDAY',2 ,'M2006','TH4006' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5010','FRIDAY',2 ,'M2006','TH4003' ,100 ) ;

INSERT INTO MTIME VALUES ('MC5011','TUESDAY',1 ,'M2001','TH4001' ,100 ) ;

SELECT \* FROM MTIME ;

INSERT INTO SLOT VALUES(1,0930,1200);

INSERT INTO SLOT VALUES(2,1230,1500);

INSERT INTO SLOT VALUES(3,1530,1800);

SELECT \* FROM SLOT;

CREATE TABLE THEATER (THEATERID VARCHAR(10) PRIMARY KEY, THEATERNAME VARCHAR(20) , CAPACITY NUMBER(3));

INSERT INTO THEATER VALUES('TH4001','PADMA',100);

INSERT INTO THEATER VALUES('TH4002','JAMUNA',100);

INSERT INTO THEATER VALUES('TH4003','BONOLOTA',100);

INSERT INTO THEATER VALUES('TH4004','NOBORATRI',100);

INSERT INTO THEATER VALUES('TH4005','GULBAHAR',100);

INSERT INTO THEATER VALUES('TH4006','CHARULOTA',100);

SELECT \* FROM THEATER ;

## PROCEDURES:

create or replace PROCEDURE CREATE\_MCODE( HID IN THEATER.THEATERID%TYPE ,ID OUT MTIME.MCODE%TYPE, CAP OUT THEATER.CAPACITY%TYPE)

AS

BEGIN

SELECT MAX(MCODE) INTO ID FROM MTIME;

SELECT CAPACITY INTO CAP FROM THEATER WHERE THEATERID=HID ;

END;​

create or replace PROCEDURE CREATE\_MID( ID OUT MOVIE.MID%TYPE)

AS

BEGIN SELECT MAX(MID) INTO ID FROM MOVIE ;

END;​

create or replace PROCEDURE DELETE\_MOVIE( ID MOVIE.MID%TYPE )

AS

BEGIN

DELETE FROM MOVIE WHERE MID=ID ;

DELETE FROM MTIME WHERE MID=ID;

END;​

create or replace PROCEDURE GETMSLOT( DAYTXT IN MTIME.DAY%TYPE , TID IN MTIME.THEATERID%TYPE , SCUR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN SCUR FOR SELECT MSLOT FROM SLOT WHERE MSLOT NOT IN (SELECT MSLOT FROM MTIME WHERE DAY= DAYTXT AND THEATERID = TID);

END;​

create or replace PROCEDURE SELL\_TICKET( CID CUSTOMER.CUSTOMERID%TYPE,EID EMPLOYEE.EMPID%TYPE,ID MOVIE.MID%TYPE,SDAY MTIME.DAY%TYPE , TID THEATER.THEATERID%TYPE ,SLOT SLOT.MSLOT%TYPE ,SEAT MTIME.AVAILSEAT%TYPE, AVSEAT MTIME.AVAILSEAT%TYPE)

AS

TIMECODE MTIME.MCODE%TYPE ;

BEGIN

SELECT MCODE INTO TIMECODE FROM MTIME WHERE MID =ID AND DAY=SDAY AND THEATERID=TID AND MSLOT=SLOT ;

INSERT INTO TICKET VALUES (SELL\_SEQ.NEXTVAL, TIMECODE,CID ,EID, SEAT);

UPDATE MTIME SET AVAILSEAT=AVSEAT WHERE MCODE=TIMECODE;

END;​

## SEQUENCES:

CREATE SEQUENCE SELL\_SEQ

MINVALUE 1000001

MAXVALUE 999999999999999999999999999

START WITH 1000001

INCREMENT BY 1

NOCACHE ;

## FUNCTION:

create or replace FUNCTION CREATE\_EMP\_ID

RETURN VARCHAR

IS

ID EMPLOYEE.EMPID%TYPE;

BEGIN

SELECT MAX(EMPID) INTO ID FROM EMPLOYEE;

RETURN ID;

END;​

## 

## VIEWS:

CREATE VIEW MOVIE\_VIEW

AS

SELECT MID,MOVIENAME FROM MOVIE

CREATE VIEW EMPLOYEE\_VIEW

AS

SELECT \* FROM EMPLOYEE

CREATE VIEW TICKET\_VIEW

AS

SELECT \* FROM TICKET ORDER BY CUSTOMERID DESC

## TRIGGERS:

CREATE OR REPLACE TRIGGER "MOVIE\_LIMIT"

BEFORE INSERT ON MTIME

FOR EACH ROW

DECLARE

LIMIT NUMBER;

BEGIN

SELECT COUNT(\*) INTO LIMIT FROM MTIME WHERE MID =:NEW.MID ;

IF LIMIT=3 THEN

RAISE\_APPLICATION\_ERROR(20001,'LIMIT CROSSED');

END IF;

END;

CREATE OR REPLACE TRIGGER "MOVIE\_DELETE"

AFTER

DELETE ON MOVIE

FOR EACH ROW

BEGIN

INSERT INTO DELETEDMOVIE VALUES (:NEW.MID,:NEW.MOVIENAME,SYSDATE) ;

END;

CREATE OR REPLACE TRIGGER "DELETED\_ID"

BEFORE DELETE ON CUSTOMER

FOR EACH ROW

BEGIN

INSERT INTO DELETED\_CUSTOMER VALUES (:OLD.CUSTOMERID,:OLD.CUSTOMERNAME,:OLD.LOGINID,:OLD.EMAIL);

DELETE FROM LOGIN WHERE LOGINID= :OLD.CUSTOMERID;

END;

CREATE OR REPLACE TRIGGER "CHANGE\_PASS"

AFTER

UPDATE OF PASSWORD ON LOGIN

FOR EACH ROW

BEGIN

INSERT INTO CHANGEDPASS VALUES (:NEW.LOGINID,:OLD.PASSWORD ,:NEW.PASSWORD ,SYSDATE) ;

END;

Table Procedure Function View Trigger Sequence