

Database Technologies Lab-2

Name – Arvind Krishna
SRN – PES1UG19CS090
Sem – 6, Sec – B

Movie Ticket Booking System

CRUD Statements

Create.sql

```
CREATE TABLE cashier
(
  cashier_ID VARCHAR(6) NOT NULL ,
  cashier_name VARCHAR(30) NOT NULL,
  cashier_address VARCHAR(50) NOT NULL,
  PRIMARY KEY (cashier_ID)
);

CREATE TABLE theatre
(
  theatre_ID VARCHAR(6) NOT NULL ,
  theatre_name VARCHAR(25) NOT NULL,
  theatre_address VARCHAR(50) NOT NULL,
  seats_available INT CHECK(seats_available >= 0),
  PRIMARY KEY (theatre_ID)
);

CREATE TABLE customer
(
  cust_name VARCHAR(15) NOT NULL ,
  cust_ID VARCHAR(6) NOT NULL,
  email_id VARCHAR(50) NOT NULL,
  phone_no VARCHAR(10) NOT NULL,
  PRIMARY KEY (cust_ID)
);

CREATE TABLE movie
(
  movie_ID VARCHAR(6) NOT NULL ,
  movie_name VARCHAR(50) NOT NULL,
```

```

    director VARCHAR(25) NOT NULL,
    release_date VARCHAR(10) NOT NULL,
    PRIMARY KEY (movie_ID)
);

CREATE TABLE actors
(
    Actor_name VARCHAR(30) NOT NULL,
    Age INT NOT NULL,
    Sex CHAR(1) NOT NULL,
    movie_ID VARCHAR(6) NOT NULL,
    PRIMARY KEY (Actor_name, movie_ID),
    FOREIGN KEY (movie_ID) REFERENCES movie(movie_ID) on update cascade on delete
cascade
);

CREATE TABLE ticket
(
    ticket_no VARCHAR(6) NOT NULL ,
    seat_no INT NOT NULL,
    price INT NOT NULL,
    cust_ID VARCHAR(6) NOT NULL,
    theatre_ID VARCHAR(6) NOT NULL,
    PRIMARY KEY (ticket_no),
    FOREIGN KEY (show_ID) REFERENCES shows(show_ID) on update cascade on delete
cascade,
    FOREIGN KEY (cust_ID) REFERENCES customer(cust_ID),
    FOREIGN KEY (offer_ID) REFERENCES offer(offer_ID) on update cascade on delete
cascade,
    FOREIGN KEY (theatre_ID) REFERENCES theatre(theatre_ID) on update cascade on
delete cascade
);

```

Insert.sql

```

INSERT into cashier values ('8HNFGB','Steve', 'No 169, Majestic');
INSERT into cashier values ('6HDIA','Harvey', '450 Stone, Satelite town');
INSERT into cashier values ('ABJD82','Richard', '390 Block, Yelahanka');
INSERT into cashier values ('AJNF5Y','Nelson', '7th main, JP nagar');

--Inserting theatres

```

```

INSERT into theatre values ('A7VGBD','Banglore cinemas', 'Banglore Cinemas,
Banglore', 60);
INSERT into theatre values ('AHS8A','Central Movies', 'City Cetral Mall,
Banglore', 100);
INSERT into theatre values ('8UHD7A','Fun Zone', 'Jayanagar, 4th cross', 150);

--inserting customer values
INSERT into customer values ('Alex','HNFOSA', 'alex@mymail.com',7267875614);
INSERT into customer values ('Lucius','DUJD87', 'lkurten0@studiopress.com',
9598072503);
INSERT into customer values ('Grigs','KIFH76', 'rgrigs1@statcounter.com',
1789448756);
INSERT into customer values ('Mennear','DUNS87', 'gmennear2@lulu.com',
9009039174);

--MOVIES
INSERT into movie values('SUDG7J','How to Meet Girls from a
Distance','Gentner','10/14/2021');
INSERT into movie values('H7S9KJ','Horrible Dr. Hichcock,
The','OBrogane','12/19/2020');
INSERT into movie values('HSA8H2','Operation Mad Ball','Bellelli','11/17/2020');
INSERT into movie values('HSAUH7','Nô','Gaukroger','2/26/2021');
INSERT into movie values('HSA8H5','Silentium','Hadlington','8/8/2021');

--actors
Insert into actors values( 'McClymont',55,'F', 'HSA8H5');
Insert into actors values('Kitney',99,'F', 'HSAUH7');
Insert into actors values('Bradford',60,'F', 'HSAUH7');
Insert into actors values('Richardson',100,'M','H7S9KJ');

```

Select queries:

```

-- select all customer whose price was less than $100
select * from customer where cust_id=((select cust_id from customer) intersect
(select cust_id from ticket where final_price<100));

-- Cashier queries:
-- select the cashier who has sold the ticket to a particular person
select cashier_name from cashier as C,sale as S where C.cashier_id=S.cashier_id
and ticket_no in (select ticket_no from ticket as T where T.cust_id in (select
cust_id from customer where cust_name='Alex'));

-- select the cashier who has sold the ticket for a particular movie

```

```

select C.cashier_name from cashier as C where cashier_id in(select S.cashier_id
from sale as S where S.ticket_no in (select ticket_no from ticket as T where
T.show_id in ((select movie_id from movie as M , movie_name='Fame' and
M.movie_id=S.movie_id) union (select movie_id from movie where
movie_name='Flicker'))));

-- Actor queries:
-- select the actors acting in a given movie
select actors.Actor_name, movie.movie_name from actors, movie where
actors.movie_id=movie.movie_id and movie.movie_name = 'Chocolate factory';

--Theatre queries:
--return the theatre details that runs a movie directed by some specific director
select * from theatre where theatre_id in (select theatre_id from shows where
movie_id=((select movie_id from movie where director='Gentner') intersect (select
movie_id from shows)));

```