DRAMA THEATER MANAGEMENT

By GROUP 10

DBMS PROJECT

CONTENTS

SCOPE
ATTRIBUTES
CONSTRAINTS
RELATIONSHIPS
ER DIAGRAM
DATA INSERTION
TABLES
10 QUIRIES
PROCEDURE
FUNCTION
TRIGGER

SCOPE

- In the Drama Theatre, there are several administrations to schedule performances, manage actors and crew, and monitor financial transactions, including ticket sales and reservations.
- A Theatre has a name, location, and capacity, it can have many Performances, but each Performance takes place in one Theatre.
- Each Performance has a unique PerformanceID, PlayID, TheatreID, Date, Time and Ticket Price.
- Each Play has a unique PlayID, Title, Director, Genre and Duration.
  A Play can have many Performances, but each Performance is associated with one Play.
- To maintain Actor record we have a unique ActorID, Name, Gender, Date of Birth and Contact Information. An Actor can participate in many Performances, and each Performance can have multiple Actors.
- Customers will have the ability to browse play schedules, book tickets, and leave reviews.
- The Tickets bought by customers will have a unique TicketID, PerformanceID Seat Number, Price, Status of Ticket, whether sold or reserved.
- Tickets are sold by an Employee. Employee has a unique EmployeeID, Name, Position, Salary, Contact Information. Employee can work in one or more Theatres, and each Theatre can have many Employees.
- These tickets can be reserved. Reservation has a unique ReservationID, CustomerID, PerformanceID, Reserved On, Reserved For, Status of Reservation, whether pending or confirmed. Each Reservation is made by one Customer for one Performance.
- Payment for the above is done by Customer and has a unique PaymentID, CustomerID, Amount and Payment Date. Each Payment is made by one Customer for one or more Reservations.
- Customers are identified by unique CustomerID, Name, Email and Phone Number. Reviews are given by Customers.
- Each Review is written by one Customer for one Play. Each review has a unique ReviewID, PlayID, CustomerID, Rating and Comments.
- Performances are carried out smoothly by the Crew, each crew member have a unique CrewID, Name, Role. Each Performance may involve multiple Crew members.
- Parking facility is available for Customers. Parking have a unique LotID, Vehicle Type, Status if space is Available or Occupied.

ATTRIBUTES

Theatre: TheatreID, Name Location, Capacity

Play: PlayID, Title, Director, Genre, Duration

Actor: ActorID, Name, Gender, Date of Birth, Contact Information

Performance: PerformanceID, **PlayID**, **TheatreID**, Date, Time, Ticket_Price

Ticket: TicketID, **PerformanceID**, Seat Number, Price, Status

Employee: EmployeeID, Name, Position, Contact Information

Customer: CustomerID, Name, Email, Phone

Reservation: ReservationID, **CustomerID**, **PerformanceID**, Reserved_On, Reserved_For, Status

Payment: PaymentID, **CustomerID,** Amount, Payment_Date

Review: ReviewID, **PlayID**, **CustomerID**, Rating, Comments

Crew: CrewID**,** Name, Role

Parking: LotID, Vehicle_Type, Status

CONTRAINTS

Theatre:
TheatreID (Primary Key) (Check – Starts from T), Name (Not Null), Location (Not Null), Capacity (Not Null) (Check – Cannot be Null)

Play:
PlayID (Primary Key) (Check – Starts from P), Title (Not Null), Director (Not Null), Genre (Not Null), Duration (Not Null)

Actor:
ActorID (Unique Key) (Check – Starts from A), Name (Not Null), Gender, Date of Birth (Not Null), Contact Information (Not Null)

Performance:
PerformanceID (Primary Key) (Check – Starts from P), PlayID (Foreign Key), TheatreID (Foreign Key), Date (Not Null), Time (Not Null), Ticket_Price (Not Null)

Ticket:
TicketID (Unique Key), PerformanceID (Foreign Key), Seat_Number (Not Null), Price, Status (Check - Status in ('Sold','Reserved'))

Employee:
EmployeeID (Unique Key), Name (Not Null), Position (Not Null), Salary (Not Null), Contact_Information (Not Null)

Customer:
CustomerID (Primary Key) (Check – Starts from C), Name (Not Null), Email (Not Null), Phone (Not Null)

Reservation:
ReservationID (Unique Key) (Check – Starts from R), CustomerID (Foreign Key), PerformanceID (Foreign Key), Reserved_On (Not Null), Reserved_For (Not Null), Status (Check - Status in ('Sold','Reserved'))

Payment:
PaymentID (Unique Key), CustomerID (Foreign Key), Amount (Not Null), Payment Date (Not Null)

Review:
ReviewID (Unique Key), PlayID (Foreign Key), CustomerID (Foreign Key), Rating (Check - Rating in ('1','2','3','4','5')), Comments (Not Null)

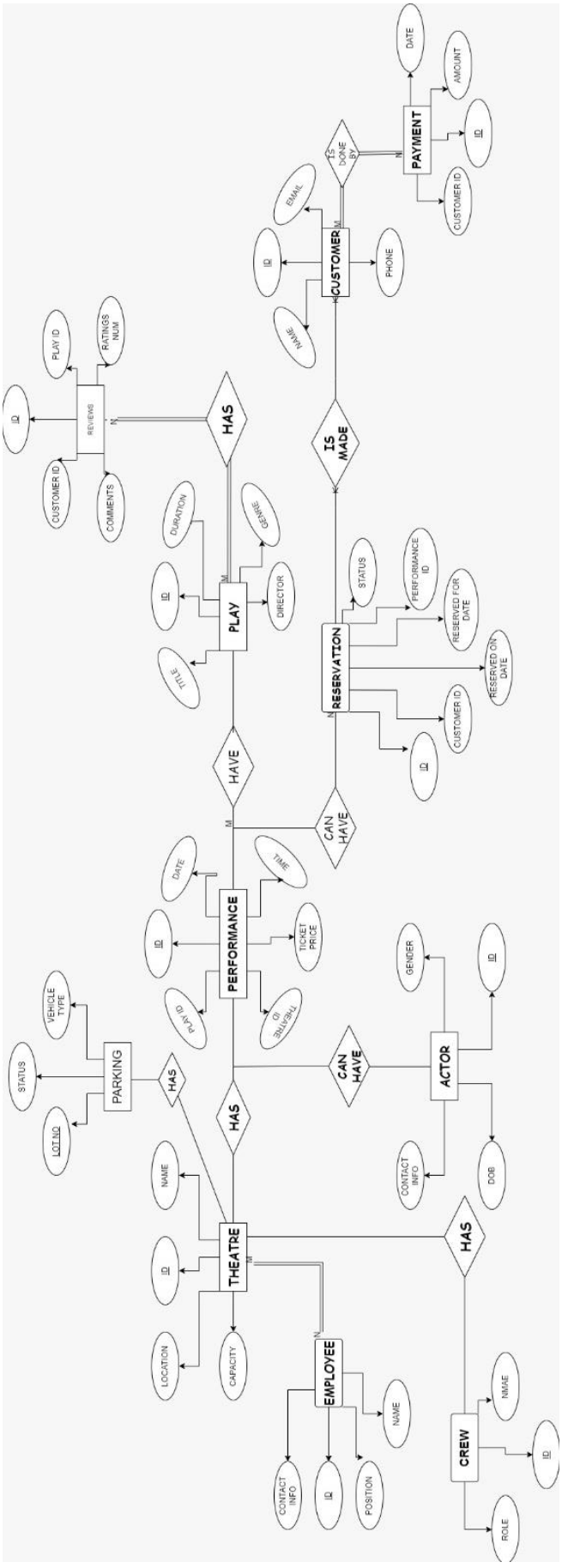
Crew:
CrewID (Unique Key) (Check – Starts from C)**,** Name (Not Null), Role (Not Null)

Parking:
LotID (Unique Key), Vehicle Type (Not Null), Status (Check – Status in ('Available','Occupied'))

RELATIONSHIP

- A Theatre can have many Performances, but each Performance takes place in one Theatre. (One-to-Many relationship between Theatre and Performance)
- A Play can have many Performances, but each Performance is associated with one Play. (One-to-Many relationship between Play and Performance)
- An Actor can participate in many Performances, and each Performance can have multiple Actors. (Many-to-Many relationship between Actor and Performance)
- An Employee can work in one or more Theatres, and each Theatre can have many Employees. (Many-to-Many relationship between Employee and Theatre).
- Each Reservation is made by one Customer for one Performance. (Many-to-One relationship between Customer and Reservation)
- Each Payment is made by one Customer for one or more Reservations. (Many-to-One relationship between Customer and Payment)
- Each Review is written by one Customer for one Play. (Many-to-One relationship between Customer and Review)
- Each Performance may involve multiple Crew members. (Many-to-Many relationship between Crew and Performance)

ER DIAGRAM


PAYMENT
DATE
AMOUNT
ID
CUSTOMER ID
IS DONE BY
CUSTOMER
EMAIL
ID
NAME
PHONE
IS MADE
HAS
REVIEWS
PLAY ID
RATINGS NUM
ID
CUSTOMER ID
COMMENTS
PLAY
DURATION
GENRE
ID
DIRECTOR
TITLE
RESERVATION
STATUS
PERFORMANCE ID
RESERVED FOR DATE
RESERVED ON DATE
CUSTOMER ID
ID
HAVE
CAN HAVE
PERFORMANCE
DATE
TIME
ID
TICKET PRICE
PLAY ID
THEATRE ID
PARKING
VEHICLE TYPE
STATUS
LOT NO
HAS
HAS
CAN HAVE
ACTOR
GENDER
ID
CONTACT INFO
DOB
THEATRE
NAME
ID
LOCATION
CAPACITY
EMPLOYEE
NAME
CONTACT INFO
ID
POSITION
HAS
CREW
NMAE
ID
ROLE

DATA INSERTION:CREATINGTABLES

THEATRE TABLE

```
SQL> create table THEATRE
  2  (TheatreID varchar2(6) primary key check (TheatreID like 'T%'),
  3  Name varchar2(35) not null,
  4  Location varchar2(25) not null,
  5  Capacity number(10) not null check (Capacity!=0)
  6  );

Table created.

SQL> desc THEATRE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 THEATREID                                 NOT NULL VARCHAR2(6)
 NAME                                      NOT NULL VARCHAR2(35)
 LOCATION                                  NOT NULL VARCHAR2(25)
 CAPACITY                                  NOT NULL NUMBER(10)
```

PLAY TABLE

```
 SQL> create table PLAY
   2  (PlayID varchar2(25) primary key check (PlayID like 'P%'),
   3  Title varchar2(25) not null,
   4  Director varchar2(25) not null,
   5  Genre varchar2(25) not null,
   6  Duration varchar2(10) not null
   7  );

SQL> desc PLAY;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PLAYID                                    NOT NULL VARCHAR2(25)
 TITLE                                     NOT NULL VARCHAR2(25)
 DIRECTOR                                  NOT NULL VARCHAR2(25)
 GENRE                                     NOT NULL VARCHAR2(25)
 DURATION                                  NOT NULL VARCHAR2(10)
```

ACTOR TABLE

```
SQL> create table ACTOR
  2  (ActorID varchar2(25) unique check (ActorID like 'A%'), Name varchar2(25) not null,
  3  Gender char(10),
  4  Date_of_Birth date not null,
  5  Contact_Information number(10) not null
  6  );

Table created.

SQL> desc ACTOR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACTORID                                            VARCHAR2(25)
 NAME                                      NOT NULL VARCHAR2(25)
 GENDER                                             CHAR(10)
 DATE_OF_BIRTH                             NOT NULL DATE
 CONTACT_INFORMATION                       NOT NULL NUMBER(10)
```

PERFORMANCE TABLE

```
SQL> create table PERFORMANCE
  2  (PerformanceID varchar2(25) primary key check (PerformanceID like 'P%'),
  3  PlayID varchar2(25) constraint c1 references PLAY(PlayID),
  4  TheatreID varchar2(6) constraint c2 references THEATRE(TheatreID),
  5  P_Date date not null,
  6  Time varchar2(10) not null,
  7  Ticket_Price number(5,2) not null
  8  );

Table created.

SQL> desc PERFORMANCE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERFORMANCEID                             NOT NULL VARCHAR2(25)
 PLAYID                                             VARCHAR2(25)
 THEATREID                                          VARCHAR2(6)
 P_DATE                                    NOT NULL DATE
 TIME                                      NOT NULL VARCHAR2(10)
 TICKET_PRICE                              NOT NULL NUMBER(5,2)
```

TICKET TABLE

```
SQL> create table TICKET
  2  (TicketID varchar2(25) unique check (TicketID like 'T%'),
  3  PerformanceID varchar2(25) constraint c3 references PERFORMANCE(PerformanceID),
  4  Seat_Number number(5,2) not null,
  5  Price number(5),
  6  Status char(10) check(Status in ('Sold','Reserved'))
  7  );

Table created.

SQL> desc TICKET;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TICKETID                                           VARCHAR2(25)
 PERFORMANCEID                                      VARCHAR2(25)
 SEAT_NUMBER                               NOT NULL NUMBER(5,2)
 PRICE                                              NUMBER(5)
 STATUS                                             CHAR(10)
```

EMPLOYEE TABLE

```
SQL> create table EMPLOYEE
  2  (EmployeeID varchar2(25) unique check(EmployeeID like 'E%'),
  3  Name char(25) not null,
  4  Position varchar2(25) not null,
  5  Salary number(7) not null,
  6  Contact_Information number(10) not null
  7  );

Table created.

SQL> desc EMPLOYEE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEEID                                         VARCHAR2(25)
 NAME                                      NOT NULL CHAR(25)
 POSITION                                  NOT NULL VARCHAR2(25)
 SALARY                                    NOT NULL NUMBER(7)
 CONTACT_INFORMATION                       NOT NULL NUMBER(10)
```

CUSTOMER TABLE

```
SQL> create table CUSTOMER
  2  (CustomerID varchar2(25) primary key check (CustomerID like 'C%'),
  3  Name char(25) not null,
  4  Email varchar2(25) not null,
  5  Phone number(10) not null
  6  );

Table created.

SQL> desc CUSTOMER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMERID                                NOT NULL VARCHAR2(25)
 NAME                                      NOT NULL CHAR(25)
 EMAIL                                     NOT NULL VARCHAR2(25)
 PHONE                                     NOT NULL NUMBER(10)
```

RESERVATION TABLE

```
SQL> create table RESERVATION
  2  (ReservationID varchar2(25) unique check (ReservationID like 'R%'),
  3  CustomerID varchar2(25) constraint c4 references CUSTOMER(CustomerID),
  4  PerformanceID varchar2(25) constraint c5 references PERFORMANCE(PerformanceID),
  5  Reserved_On date not null,
  6  Reserved_For date not null,
  7  Status char(10) check(Status in ('Pending','Confirmed'))
  8  );

Table created.

SQL> desc RESERVATION;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RESERVATIONID                                      VARCHAR2(25)
 CUSTOMERID                                         VARCHAR2(25)
 PERFORMANCEID                                      VARCHAR2(25)
 RESERVED_ON                               NOT NULL DATE
 RESERVED_FOR                              NOT NULL DATE
 STATUS                                             CHAR(10)
```

PAYMENT TABLE

```
SQL> create table PAYMENT
  2  (PaymentID varchar2(25) unique,
  3  CustomerID varchar2(25) constraint c6 references CUSTOMER(CustomerID),
  4  Amount number(7,2) not null,
  5  Payment_Date date not null
  6  );

Table created.

SQL> desc PAYMENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PAYMENTID                                          VARCHAR2(25)
 CUSTOMERID                                         VARCHAR2(25)
 AMOUNT                                    NOT NULL NUMBER(7,2)
 PAYMENT_DATE                              NOT NULL DATE
```

REVIEW TABLE

```
SQL> create table REVIEW
  2  (ReviewID varchar2(25) unique,
  3  PlayID varchar2(25) constraint c7 references PLAY(PlayID),
  4  CustomerID varchar2(25) constraint c8 references CUSTOMER(CustomerID),
  5  Rating number(1) check(Rating in ('1','2','3','4','5')),
  6  Comments varchar2(30) not null
  7  );

Table created.

SQL> desc REVIEW;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 REVIEWID                                           VARCHAR2(25)
 PLAYID                                             VARCHAR2(25)
 CUSTOMERID                                         VARCHAR2(25)
 RATING                                             NUMBER(1)
 COMMENTS                                  NOT NULL VARCHAR2(30)
```

CREW TABLE

```
SQL> create table CREW
  2  (CrewID varchar2(25) unique check (CrewID like 'C%'),
  3  Name char(20) not null,
  4  Role varchar2(25) not null
  5  );

Table created.

SQL> desc CREW;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CREWID                                             VARCHAR2(25)
 NAME                                      NOT NULL CHAR(20)
 ROLE                                      NOT NULL VARCHAR2(25)
```

PARKING TABLE

```
SQL> create table PARKING
  2  (LotID varchar2(10) unique,
  3  Vehicle_Type varchar2(25) not null,
  4  Status varchar2(20) check(Status in ('Available','Occupied'))
  5  );

Table created.

SQL> desc PARKING;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOTID                                              VARCHAR2(10)
 VEHICLE_TYPE                              NOT NULL VARCHAR2(25)
 STATUS                                             VARCHAR2(20)
```

INSERTING DATA INTO TABLES

THEATRE TABLE

```
SQL> insert into THEATRE values ('T10001','Theatre 1','Mumbai',850);

1 row created.

SQL> insert into THEATRE values ('T10002','Theatre 2','Mumbai',1000);

1 row created.

SQL> insert into THEATRE values ('T10003','Theatre 3','Mumbai',500);

1 row created.

SQL> insert into THEATRE values ('T10004','Theatre 4','Mumbai',1500);

1 row created.

SQL> insert into THEATRE values ('T10005','Theatre 5','Mumbai',1000);

1 row created.
```

PLAY TABLE

```
SQL> insert into PLAY values ('P10001','Natasamrat','Kusumagraj','Family','3h 46m');

1 row created.

SQL> insert into PLAY values ('P10002','Dear Lier','Satyadev Dubey','Drama','3h 5m');

1 row created.

SQL> insert into PLAY values ('P10003','Yayati','Girish Karnad','Mythical','4h');

1 row created.

SQL> insert into PLAY values ('P10004','Taj Mahal ka Tender','Salim Arif','Comedy','2h 39m');

1 row created.

SQL> insert into PLAY values ('P10005','Hey Ram Nathuram','Sharad Ponkshe','Crime','3h 8m');

1 row created.
```

ACTOR TABLE

```
SQL> insert into ACTOR values ('A10001','Rajesh Kumar','Male','14-Feb-1995',7648925367);

1 row created.

SQL> insert into ACTOR values ('A10002','Nitin Parab','Male','30-Dec-1990',9648277209);

1 row created.

SQL> insert into ACTOR values ('A10003','Sulbha Thakur','Female','03-Jan-1987',8953575321);

1 row created.

SQL> insert into ACTOR values ('A10004','Sanjana Hole','Female','24-Oct-2000',9468122393);

1 row created.

SQL> insert into ACTOR values ('A10005','Manish Shah','Male','21-Jun-1995',9654425662);

1 row created.
```

PERFORMANCE TABLE

```
SQL> insert into PERFORMANCE values ('PF10001','P10001','T10001','12-Oct-2023','17:00 PM',600);

1 row created.

SQL> insert into PERFORMANCE values ('PF10002','P10002','T10002','13-Oct-2023','15:00 PM',800);

1 row created.

SQL> insert into PERFORMANCE values ('PF10003','P10003','T10003','14-Oct-2023','14:00 PM',800);

1 row created.

SQL> insert into PERFORMANCE values ('PF10004','P10004','T10004','15-Oct-2023','17:00 PM',500);

1 row created.

SQL> insert into PERFORMANCE values ('PF10005','P10005','T10005','16-Oct-2023','16:00 PM',700);

1 row created.
```

TICKET TABLE

```
SQL> insert into TICKET values ('T10001','PF10001',100,600,'Sold');

1 row created.

SQL> insert into TICKET values ('T10002','PF10002',100,799,'Reserved');

1 row created.

SQL> insert into TICKET values ('T10003','PF10003',100,699,'Sold');

1 row created.

SQL> insert into TICKET values ('T10004','PF10004',100,900,'Reserved');

1 row created.

SQL> insert into TICKET values ('T10005','PF10005',100,899,'Sold');

1 row created.
```

EMPLOYEE TABLE

```
SQL> insert into EMPLOYEE values ('E10001','Kevin Patel','Manager',150000,8956345378);

1 row created.

SQL> insert into EMPLOYEE values ('E10002','Anish Kaul','Technical',70000,9586238110);

1 row created.

SQL> insert into EMPLOYEE values ('E10003','Kushboo Raul','Hospitality',50000,7845104279);

1 row created.

SQL> insert into EMPLOYEE values ('E10004','Monisha Shinde','Security',25000,9967183995);

1 row created.

SQL> insert into EMPLOYEE values ('E10005','Vivek Patil','PR',30000,7956382401);

1 row created.
```

CUSTOMER TABLE

```
SQL> insert into CUSTOMER values ('C10001','Shagun Mishra','shagunmishra.gmail.com',9887856278);

1 row created.

SQL> insert into CUSTOMER values ('C10002','Bhavya Gupta','bhavyagupta.gmail.com',8976352967);

1 row created.

SQL> insert into CUSTOMER values ('C10003','Disha Tilak','dishatilak.gmail.com',8789674329);

1 row created.

SQL> insert into CUSTOMER values ('C10004','Tanmay Gurav','tanmaygurav.gmail.com',9978664389);

1 row created.

SQL> insert into CUSTOMER values ('C10005','Lokesh Sawant','lokeshsawant.gmail.com',8879677458);

1 row created.
```

RESERVATION TABLE

```
SQL> insert into RESERVATION values ('R10001','C10001','PF10001','05-Sep-2023','16-Oct-2023','Pendin
g');

1 row created.

SQL> insert into RESERVATION values ('R10002','C10002','PF10002','04-Sep-2023','15-Oct-2023','Confir
med');

1 row created.

SQL> insert into RESERVATION values ('R10003','C10003','PF10003','03-Sep-2023','14-Oct-2023','Pendin
g');

1 row created.

SQL> insert into RESERVATION values ('R10004','C10004','PF10004','02-Sep-2023','13-Oct-2023','Confir
med');

1 row created.

SQL> insert into RESERVATION values ('R10005','C10005','PF10005','01-Sep-2023','12-Oct-2023','Confir
med');

1 row created.
```

PAYMENT TABLE

```
SQL> insert into PAYMENT values ('PY10001','C10001',1000,'20-Sep-2023');

1 row created.

SQL> insert into PAYMENT values ('PY10002','C10002',1500,'04-Sep-2023');

1 row created.

SQL> insert into PAYMENT values ('PY10003','C10003',1698,'15-Sep-2023');

1 row created.

SQL> insert into PAYMENT values ('PY10004','C10004',800,'02-Sep-2023');

1 row created.

SQL> insert into PAYMENT values ('PY10005','C10005',699,'01-Sep-2023');

1 row created.
```

REVIEW TABLE

```
SQL> insert into REVIEW values ('R10001','P10001','C10001','5','Very Good');

1 row created.

SQL> insert into REVIEW values ('R10002','P10002','C10002','4','Excellent');

1 row created.

SQL> insert into REVIEW values ('R10003','P10003','C10003','5','Outstanding');

1 row created.

SQL> insert into REVIEW values ('R10004','P10004','C10004','4','Very Good');

1 row created.

SQL> insert into REVIEW values ('R10005','P10005','C10005','5','Must Watch');

1 row created.
```

CREW TABLE

```
SQL> insert into CREW values ('C10001','Babu Sahab','Electric');

1 row created.

SQL> insert into CREW values ('C10002','Manu Shah','Sound');

1 row created.

SQL> insert into CREW values ('C10003','Soham Kumar','Lights');

1 row created.

SQL> insert into CREW values ('C10004','Shakhib Shik','Stage');

1 row created.

SQL> insert into CREW values ('C10005','Dilip Thakkar','Paints');

1 row created.
```

PARKING TABLE

```
SQL> insert into PARKING values ('L100','2-Wheeler','Available');

1 row created.

SQL> insert into PARKING values ('L101','2-Wheeler','Available');

1 row created.

SQL> insert into PARKING values ('L201','4-Wheeler','Occupied');

1 row created.

SQL> insert into PARKING values ('L202','4-Wheeler','Available');

1 row created.

SQL> insert into PARKING values ('L203','4-Wheeler','Occupied');

1 row created.
```

TABLES

THEATRE TABLE

TheatreID	Name	Location	Capacity
T10001	Theatre 1	Mumbai	850
T10002	Theatre 2	Mumbai	1000
T10003	Theatre 3	Mumbai	500
T10004	Theatre 4	Mumbai	1500
T10005	Theatre 5	Mumbai	1000

PLAY TABLE

PlayID	Title	Director	Genre	Duration
P10001	Natasamrat	Kusumagraj	Family	3h 46m
P10002	Dear Lier	Satyadev Dubey	Drama	3h 5m
P10003	Yayati	Girish Karnad	Mythical	4h
P10004	Taj Mahal ka Tender	Salim Arif	Comedy	2h 39m
P10005	Hey Ram Nathuram	Sharad Ponkshe	Crime	3h 8m

ACTOR TABLE

ActorID	Name	Gender	Date_of_Birth	Contact_Information
A10001	Rajesh Kumar	Male	14-Feb-1995	7648925367
A10002	Nitin Parab	Male	30-Dec-1990	9648277209
A10003	Sulbha Thakur	Female	03-Jan-1987	8953575321
A10004	Sanjana Hole	Female	24-Oct-2000	9468122393
A10005	Manish Shah	Male	21-Jun-1995	9654425662

PERFORMANCE TABLE

PerformanceID	PlayID	TheatreID	P_Date	Time	Ticket_Price
PF10001	P10001	T10001	12-Oct-2023	17:00 PM	600
PF10002	P10002	T10002	13-Oct-2023	15:00 PM	800
PF10003	P10003	T10003	14-Oct-2023	14:00 PM	800
PF10004	P10004	T10004	15-Oct-2023	17:00 PM	500
PF10005	P10005	T10005	16-Oct-2023	16:00 PM	700

TICKET TABLE

TicketID	PerformanceID	Seat_Number	Price	Status
T10001	PF10001	100	600	Sold
T10002	PF10002	100	799	Reserved
T10003	PF10003	100	699	Sold
T10004	PF10004	100	900	Reserved
T10005	PF10005	100	899	Sold

EMPLOYEE TABLE

EmployeeID	Name	Position	Salary	Contact_Information
E10001	Kevin Patel	Manager	150000	8956345378
E10002	Anish Kaul	Technical	70000	9586238110
E10003	Kushboo Raul	Hospitality	50000	7845104279
E10004	Monisha Shinde	Security	25000	9967183995
E10005	Vivek Patil	PR	30000	7956382401

CUSTOMER TABLE

CustomerID	Name	Email	Phone
C10001	Shagun Mishra	shagunmishra.gmail.com	9887856278
C10002	Bhavya Gupta	bhavyagupta.gmail.com	8976352967
C10003	Disha Tilak	dishatilak.gmail.com	8789674329
C10004	Tanmay Gurav	tanmaygurav.gmail.com	9978664389
C10005	Lokesh Sawant	lokeshsawant.gmail.com	8879677458

RESERVATION TABLE

ReservationID	Customer ID	PerformanceID	Reserved_On	Reserved_For	Status
R10001	C10001	PF10001	05-Sep-2023	16-Oct-2023	Pending
R10002	C10002	PF10002	04-Sep-2023	15-Oct-2023	Confirmed
R10003	C10003	PF10003	03-Sep-2023	14-Oct-2023	Pending
R10004	C10004	PF10004	02-Sep-2023	13-Oct-2023	Confirmed
R10005	C10005	PF10005	01-Sep-2023	12-Oct-2023	Confirmed

PAYMENT TABLE

PaymentID	CustomerID	Amount	Payment_Date
PY10001	C10001	1000	20-Sep-2023
PY10002	C10002	1500	04-Sep-2023
PY10003	C10003	1698	15-Sep-2023
PY10004	C10004	800	02-Sep-2023
PY10005	C10005	699	01-Sep-2023

REVIEW TABLE

ReviewID	PlayID	CustomerID	Rating	Comments
R10001	P10001	C10001	5	Very Good
R10002	P10002	C10002	4	Excellent
R10003	P10003	C10003	5	Outstanding
R10004	P10004	C10004	4	Very Good
R10005	P10005	C10005	5	Must Watch

CREW TABLE

CrewID	Name	Role
C10001	Babu Sahab	Electric
C10002	Manu Shah	Sound
C10003	Soham Kumar	Lights
C10004	Shakhib Shik	Stage
C10005	Dilip Thakkar	Paints

PARKING TABLE

LotID	Vehicle_Type	Status
L100	2-Wheeler	Available
L101	2-Wheeler	Available
L201	4-Wheeler	Occupied
L202	4-Wheeler	Available
L203	4-Wheeler	Occupied

10 QUIRIES

1] Display theatre table by capacity in descending order

```
SQL> select * from THEATRE
  2  order by Capacity DESC;

THEATR NAME                               LOCATION                  CAPACITY
------ ---------------------------------- ------------------------ ----------
T10004 Theatre 4                          Mumbai                         1500
T10002 Theatre 2                          Mumbai                         1000
T10005 Theatre 5                          Mumbai                         1000
T10001 Theatre 1                          Mumbai                          850
T10003 Theatre 3                          Mumbai                          500
```

2] Fetch name of the crew member starting by D

```
SQL> select Name from CREW where Name LIKE 'D%';

NAME
--------------------
Dilip Thakkar
```

3] Show all the tickets in the range of 100 to 899

```
SQL> select * from TICKET where Price between 100 and 899;

TICKETID                PERFORMANCEID             SEAT_NUMBER      PRICE
----------------------- ------------------------- ----------- ----------
STATUS
----------
T10001                  PF10001                           100        600
Sold

T10002                  PF10002                           100        799
Reserved

T10003                  PF10003                           100        699
Sold


TICKETID                PERFORMANCEID             SEAT_NUMBER      PRICE
----------------------- ------------------------- ----------- ----------
STATUS
----------
T10005                  PF10005                           100        899
Sold
```

4] Increase the salary of all the employees by 5%

```
SQL> UPDATE EMPLOYEE SET Salary = Salary + (Salary * 5/100);

5 rows updated.

SQL> select * from EMPLOYEE;

EMPLOYEEID                NAME                      POSITION
------------------------- ------------------------- -------------------------
    SALARY CONTACT_INFORMATION
---------- ------------------
E10001                    Kevin Patel               Manager
    157500          8956345378

E10002                    Anish Kaul                Technical
     73500          9586238110

E10003                    Kushboo Raul              Hospitality
     52500          7845104279


EMPLOYEEID                NAME                      POSITION
------------------------- ------------------------- -------------------------
    SALARY CONTACT_INFORMATION
---------- ------------------
E10004                    Monisha Shinde            Security
     26250          9967183995

E10005                    Vivek Patil               PR
     31500          7956382401
```

5] Display comments which holds the value Very Good

```
SQL> select * from REVIEW where Comments='Very Good';

REVIEWID                  PLAYID                    CUSTOMERID
------------------------- ------------------------- -------------------------
    RATING COMMENTS
---------- ------------------------------
R10001                    P10001                    C10001
         5 Very Good

R10004                    P10004                    C10004
         4 Very Good
```

6] Display the number of tickets available and for which show it is available?

```
SQL> select PerformanceID, Status
  2  FROM TICKET
  3  where Status='available';

no rows selected
```

7] Display the show which have least ticket price?

```
SQL> select Price, PerformanceID
  2  from TICKET
  3  order by Price;

     PRICE PERFORMANCEID
---------- -------------------------
       600 PF10001
       699 PF10003
       799 PF10002
       899 PF10005
       900 PF10004
```

8] To find number of employees with salary greater than 30000 and having count greater than equal to 1.

```
SQL>     SELECT Position, COUNT(*) AS "Number of Employees"
  2      FROM EMPLOYEE
  3      WHERE Salary > 30000
  4      GROUP BY Position
  5      HAVING COUNT(*) >= 1;

POSITION                  Number of Employees
------------------------- -------------------
Manager                                     1
Technical                                   1
Hospitality                                 1
```

9] Find the total payment according to each day

```
SQL> select Payment_Date, count(CustomerID)
  2  from Payment
  3  group by Payment_Date;

PAYMENT_D COUNT(CUSTOMERID)
--------- -----------------
20-SEP-23                 1
04-SEP-23                 1
02-SEP-23                 1
15-SEP-23                 1
01-SEP-23                 1
```

10] PL/SQL Block to generate the report of position wise no of employee & total salary

```
SQL> set serveroutput on;
SQL> declare
  2      cursor c1 is select Position, count(*) as no_of_employee, sum(Salary) as total_salary
  3                   from Employee
  4                   group by Position
  5                   order by Position;
  6      v_total int:= 0;
  7      v_totsal int := 0;
  8
  9  begin
 10      dbms_output.put_line(lpad('■',60,'■'));
 11      dbms_output.put_line(rpad('Position name',20)||rpad('no_of_employees',20)||lpad('total sala
ry',15));
 12      dbms_output.put_line(lpad('■',60,'■'));
 13      for i in c1
 14      LOOP
 15         dbms_output.put_line(chr(9)||rpad(i.Position,20)||rpad(i.no_of_employee,18)||lpad(i.tota
l_salary,7));
 16         v_total:= v_total+i.no_of_employee;
 17         v_totsal:= v_totsal+i.total_salary;
 18      END LOOP;
 19      dbms_output.put_line(lpad('■',60,'■'));
 20      dbms_output.put_line(chr(9)||lpad('TOTAL:'||v_total,21)||lpad('grand total:'||v_totsal,24))
;
 21      dbms_output.put_line(lpad('■',60,'■'));
 22  end;
 23  /
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
Position name       no_of_employees        total salary
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
        Hospitality         1                   50000
        Manager             1                  150000
        PR                  1                   30000
        Security            1                   25000
        Technical           1                   70000
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
                    TOTAL:5      grand total:325000
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

PL/SQL procedure successfully completed.

SQL> |
```

PROCEDURE

```
SQL> CREATE OR REPLACE PROCEDURE GET_EMPLOYEE_OR_CUSTOMER(P_ID VARCHAR2) IS
  2  BEGIN
  3
  4    FOR E IN (SELECT * FROM EMPLOYEE WHERE EmployeeID = P_ID) LOOP
  5      DBMS_OUTPUT.PUT_LINE('EmployeeID: ' || E.EmployeeID);
  6      DBMS_OUTPUT.PUT_LINE('Name: ' || E.Name);
  7      DBMS_OUTPUT.PUT_LINE('Position: ' || E.Position);
  8      DBMS_OUTPUT.PUT_LINE('Contact Information: ' || E.Contact_Information);
  9      RETURN;
 10    END LOOP;
 11
 12    FOR C IN (SELECT * FROM CUSTOMER WHERE CustomerID = P_ID) LOOP
 13      DBMS_OUTPUT.PUT_LINE('CustomerID: ' || C.CustomerID);
 14      DBMS_OUTPUT.PUT_LINE('Name: ' || C.Name);
 15      DBMS_OUTPUT.PUT_LINE('Email: ' || C.Email);
 16      DBMS_OUTPUT.PUT_LINE('Phone: ' || C.Phone);
 17      RETURN;
 18    END LOOP;
 19    DBMS_OUTPUT.PUT_LINE('ID not found in either Employees or Customers.');
 20  END GET_EMPLOYEE_OR_CUSTOMER;
 21  /

Procedure created.
```

```
SQL> BEGIN
  2    GET_EMPLOYEE_OR_CUSTOMER('E10003');
  3  end;
  4  /
EmployeeID: E10003
Name: Kushboo Raul
Position: Hospitality
Contact Information: 7845104279

PL/SQL procedure successfully completed.

SQL>
SQL> BEGIN
  2    GET_EMPLOYEE_OR_CUSTOMER('C10002');
  3  END;
  4  /
CustomerID: C10002
Name: Bhavya Gupta
Email: bhavyagupta.gmail.com
Phone: 8976352967

PL/SQL procedure successfully completed.
```

FUNCTION

```
SQL> set serveroutput on;
SQL> set verify off;
SQL> CREATE OR REPLACE FUNCTION get_employee_contact(EmployeeID varchar2)
  2  RETURN number
  3  IS
  4     v_contact_info NUMBER;
  5  BEGIN
  6
  7     SELECT Contact_Information
  8     INTO v_contact_info
  9     FROM Employee
 10     WHERE EmployeeID = get_employee_contact.EmployeeID;
 11
 12     RETURN v_contact_info;
 13  EXCEPTION
 14     WHEN NO_DATA_FOUND THEN
 15
 16        RETURN 'Employee not found';
 17  END get_employee_contact;
 18  /

Function created.

SQL> DECLARE
  2      v_EmployeeID VARCHAR2(25) := 'E10001'; -- Provide the desired EmployeeID
  3      v_ContactInfo NUMBER;
  4  BEGIN
  5      v_ContactInfo := get_employee_contact(v_EmployeeID);
  6
  7      IF v_ContactInfo IS NULL THEN
  8          DBMS_OUTPUT.PUT_LINE('Employee not found.');
  9      ELSE
 10          DBMS_OUTPUT.PUT_LINE('Contact Information: ' || v_ContactInfo);
 11      END IF;
 12  END;
 13  /
Contact Information: 8956345378

PL/SQL procedure successfully completed.
```

TRIGGER

```
SQL> create or replace trigger checkname_5931
  2  after update on ACTOR
  3  for each row
  4  when (old.Name!=new.Name)
  5  begin
  6  dbms_output.put_line('Name'||:old.Name||'has changed to'||:new.Name);
  7  end;
  8  /

Trigger created.

SQL> update ACTOR set Name='Rohan Shah'
  2  where ActorID='A10005';
NameManish Shahhas changed toRohan Shah

1 row updated.

SQL> select * from ACTOR;

ACTORID                NAME                     GENDER     DATE_OF_B
---------------------- ------------------------ ---------- ---------
CONTACT_INFORMATION
--------------------
A10001                 Rajesh Kumar             Male       14-FEB-95
         7648925367

A10002                 Nitin Parab              Male       30-DEC-90
         9648277209

A10003                 Sulbha Thakur            Female     03-JAN-87
         8953575321


ACTORID                NAME                     GENDER     DATE_OF_B
---------------------- ------------------------ ---------- ---------
CONTACT_INFORMATION
--------------------
A10004                 Sanjana Hole             Female     24-OCT-00
         9468122393

A10005                 Rohan Shah               Male       21-JUN-95
         9654425662
```