**19CSE202**

**Database Management System**

## Project Final Phase (ER-Model/Relational Schema/DDL Commands)

## Abstract Number: 11

## Event Management System

# 

# 

# 

# 

# 

# 

# **Group-16**

T. Sai Jayanth - U4CSE19355

M. Ravikanth - U4CSE19334

P. Teja Venkata Subbareddy - U4CSE19341

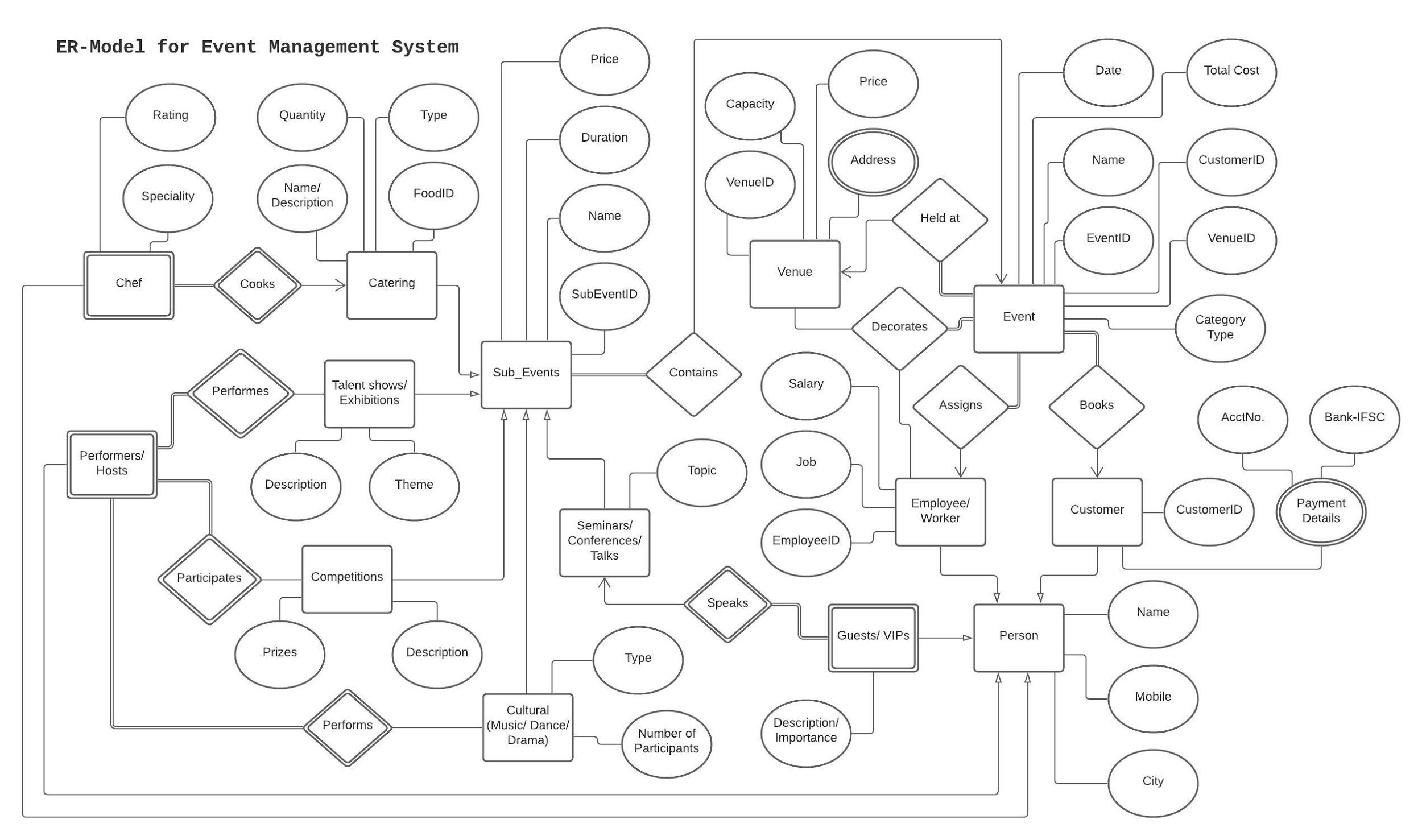
N. Sai Pavan Krishna - U4CSE19347

# **Event Management System**

## Project Abstract:

An event management group needs a database to help them in managing the different events that they help organize. Each event managed has a description which includes the name, category-level (like high-class, middle-class). The amount charged, the options they provide including food, venue, sub-events, specialized employees, etc. For example, a marriage may have a musical sub-event. For each of the options, separate information needs to be maintained like different types of food (including price), different venues (including address). The customers need to book for an event based on these available options. So, the customer data is also stored. The management group has employees who are allotted separate duties for each event.

## ER-Model



## 

## Relational Schema

* Table: Person

Entity type: Strong

Specialized from: None

Generalized from: Customer/Employee/Chef/Performer/Guest

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| P\_ID | varchar | 5 | Primary Key/Check first letter=’P’ |
| Pname | varchar | 20 | NOT NULL |
| City | varchar | 15 | NOT NULL |
| Phone | bigint | default | NOT NULL |

* Table: Customer

Entity type: Strong

Specialized from: Person

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| C\_ID | varchar | 5 | Primary Key/Check first letter=’C’ |
| P\_ID | varchar | 5 | References from Person(P\_ID) |
| CType | char | default | Values in {‘G’, ‘R’, ’P’} |
| Account\_No | integer | default | NOT NULL |
| Bank\_IFSC | varchar | 7 | NOT NULL |

* Table: Employee

Entity type: Strong

Specialized from: Person

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| EM\_ID | varchar | 5 | Primary Key/Check starts with ’EM’ |
| P\_ID | varchar | 5 | References from Person(P\_ID) |
| Job | varchar | 10 | NOT NULL |
| Salary | integer | default | NOT NULL |

* Table: Events

Entity type: Strong

Specialized from: Person

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| EV\_ID | varchar | 5 | Primary Key/Check starts with ’EV’ |
| EvName | varchar | 20 | NOT NULL |
| Category | char | default | Values in {‘H’, ‘M’, ‘L’} |
| Date | date | default | NOT NULL |
| C\_ID | varchar | 5 | References from Customer(C\_ID) |
| V\_ID | varchar | 5 | References from Venue(V\_ID) |
| EM\_ID | varchar | 5 | References from Employee(EM\_ID) |
| TotalCost | integer | default | NOT NULL |

* Table: Venue

Entity type: Strong

Specialized from: None

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| V\_ID | varchar | 5 | Primary Key/Check first letter=’V’ |
| Capacity | integer | default | NOT NULL |
| Price | integer | default | NOT NULL |
| Address | varchar | 30 | NOT NULL |
| EM\_ID | varchar | 5 | References from Employee(EM\_ID) |

* Table: SubEvents

Entity type: Strong

Specialized from: None

Generalized from: Catering/Shows/Conferences/Cultural/Competitions

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| SE\_ID | varchar | 5 | Primary Key/Check starts with ‘SE’ |
| SEname | varchar | 15 | NOT NULL |
| Duration | integers | default | NOT NULL |
| Price | integer | default | NOT NULL |
| EV\_ID | varchar | 5 | References from Events(EV\_ID) |

* Table: Catering

Entity type: Strong

Specialized from: SubEvents

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Food\_ID | varchar | 5 | Primary Key/Check first letter=’F’ |
| Fname | varchar | 15 | NOT NULL |
| FType | varchar | 10 | NOT NULL |
| ChefID | varchar | 5 | References from Chef(ChefID) |
| Quantity | integer | default | NOT NULL |
| SE\_ID | varchar | 5 | References from SubEvents(EV\_ID) |

* Table: Shows

Entity type: Strong

Specialized from: SubEvents

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Show\_ID | varchar | 5 | Primary Key |
| Theme | varchar | 15 | NOT NULL |
| Description | varchar | 50 | NOT NULL |
| Per\_ID | varchar | 5 | References from Performers(Per\_ID) |
| SE\_ID | varchar | 5 | References from SubEvents(EV\_ID) |

* Table: Competitions

Entity type: Stong

Specialized from: SubEvents

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Comp\_ID | varchar | 5 | Primary Key |
| Description | varchar | 50 | NOT NULL |
| Prizes | varchar | 15 | NOT NULL |
| Per\_ID | varchar | 5 | References from Performers(Per\_ID) |
| SE\_ID | varchar | 5 | References from SubEvents(EV\_ID) |

* Table: Cultural

Entity type: Strong

Specialized from: SubEvents

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Cul\_ID | varchar | 5 | Primary Key |
| CuType | varchar | 15 | NOT NULL |
| ParticipantCount | integer | default | NOT NULL |
| Description | varchar | 50 | NOT NULL |
| Per\_ID | varchar | 5 | References from Performers(Per\_ID) |
| SE\_ID | varchar | 5 | References from SubEvents(EV\_ID) |

* Table: Conferences

Entity type: Strong

Specialized from: SubEvents

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Conf\_ID | varchar | 5 | Primary Key |
| CoType | varchar | 15 | NOT NULL |
| Description | varchar | 50 | NOT NULL |
| Guest\_ID | varchar | 5 | References from Guest(Guest\_ID) |
| SE\_ID | varchar | 5 | References from SubEvents(EV\_ID) |

* Table: Chef

Entity type: Weak

Specialized from: Person

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Chef\_ID | varchar | 5 | Primary Key/References from Person(P\_ID) |
| Specialty | varchar | 15 |  |
| Rating | numeric | 5,1 | Check value between 0 and 5 |

* Table: Performer

Entity type: Weak

Specialized from: Person

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Per\_ID | varchar | 5 | Primary Key/References from Person(P\_ID) |
| Type | varchar | 15 | NOT NULL |
| Rating | numeric | 5,1 | Check value between 0 and 5 |

* Table: Guest

Entity type: Weak

Specialized from: Person

Generalized from: None

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Datatype** | **Size** | **Constraints** |
| Guest\_ID | varchar | 5 | Primary Key/References from Person(P\_ID) |
| Description | varchar | 50 | NOT NULL |

## DDL Commands

### Creation

create table person(

p\_id varchar(5) primary key,

pname varchar(20) not null,

city varchar(15) not null,

phone bigint not null

check (p\_id like 'P%')

);

create table customer(

c\_id varchar(5) primary key,

p\_id varchar(5) references person(p\_id),

ctype char check(ctype in ('G','R','P')),

account\_no integer not null,

bank\_ifsc varchar(7) not null

check (c\_id like 'C%')

);

create table employee(

em\_id varchar(5) primary key,

p\_id varchar(5) references person(p\_id),

job varchar(10) not null,

salary integer not null,

check (em\_id like 'EM%')

);

create table venue(

v\_id varchar(5) primary key,

capacity integer not null,

price integer not null,

address varchar(30) not null,

em\_id varchar(5) references employee(em\_id),

check (v\_id like 'V%')

);

create table events(

ev\_id varchar(5) primary key,

evname varchar(20) not null,

category char check(category in ('H','M','L')),

date date not null,

c\_id varchar(5) references customer(c\_id),

v\_id varchar(5) references venue(v\_id),

em\_id varchar(5) references employee(em\_id),

totalcost integer not null

check (ev\_id like 'EV%')

);

create table subevents(

se\_id varchar(5) primary key,

sename varchar(15) not null,

duration integer not null,

price integer not null,

ev\_id varchar(5) references events(ev\_id)

check (se\_id like 'SE%')

);

create table chef(

chef\_id varchar(5) primary key references person(p\_id),

specialty varchar(15),

rating numeric(5,1) check(rating between 0 and 5)

);

create table catering(

food\_id varchar(5) primary key,

fname varchar(15) not null,

ftype varchar(10) not null,

chef\_id varchar(5) references chef(chef\_id),

se\_id varchar(5) references subevents(se\_id)

check (food\_id like 'F%')

);

create table performer(

per\_id varchar(5) primary key,

ptype varchar(15) not null,

rating numeric(5,1) check(rating between 0 and 5)

);

create table guest(

guest\_id varchar(5) primary key,

description varchar(15) not null

);

create table shows(

show\_id varchar(5) primary key,

theme varchar(15) not null,

description varchar(50) not null,

per\_id varchar(5) references performer(per\_id),

se\_id varchar(5) references subevents(se\_id)

);

create table competitions(

comp\_id varchar(5) primary key,

description varchar(50) not null,

prizes varchar(15) not null,

per\_id varchar(5) references performer(per\_id),

se\_id varchar(5) references subevents(se\_id)

);

create table conferences(

conf\_id varchar(5) primary key,

cotype varchar(15) not null,

description varchar(50) not null,

guest\_id varchar(5) references guest(guest\_id),

se\_id varchar(5) references subevents(se\_id)

);

create table cultural(

cul\_id varchar(5) primary key,

cutype varchar(15) not null,

partipant\_count integer not null,

description varchar(50) not null,

per\_id varchar(5) references performer(per\_id),

se\_id varchar(5) references subevents(se\_id)

);

## DML Commands

### Insertion

insert into person values

('P2301','Ravi','Vizag',9182736455),

('P2302','Rahul','Goa',9182345624),

('P2303','Rishi','Kochi',9344521784),

('P2304','Rajesh','Vizag',9184859612),

('P2305','Ron','Delhi',8562795361),

('P2306','Ramesh','Nagpur',8756231486),

('P2307','Rithwik','Mumbai',5478963210),

('P2308','Remi','Hyderabad',1452369870),

('P2309','Ram','Kolkata',4253617980),

('P2310','Rukia','Sikkim',7896540123),

('P2311','Riyaz','Bhopal',3201478569),

('P2312','Rellangi','Indore',6938215470),

('P2313','Riya','Chennai',8521346097),

('P2314','Regina','Gwalior',1452789236),

('P2315','Rakul','Vizag',9463012785),

('P2316','Sahit','Vizag',9182736456),

('P2317','Sanketh','Goa',9182345625),

('P2318','Sai','Kochi',9344521785),

('P2319','Santhosh','Vizag',9184859613),

('P2320','Sam','Delhi',8562795362),

('P2321','Suresh','Nagpur',8756231487),

('P2322','Shiva','Mumbai',5478963218),

('P2323','Sunil','Hyderabad',1452369877),

('P2324','Saran','Kolkata',4253617986),

('P2325','Shilpa','Sikkim',7896540124),

('P2326','Sunny','Bhopal',3201478564),

('P2327','Supriya','Indore',6938215475),

('P2328','Selena','Chennai',8521346096),

('P2329','Sampath','Gwalior',1452789237),

('P2330','Sherlock','Vizag',9463012788);

insert into customer values

('C6341','P2301','G','45623178','SBI2C10'),

('C6342','P2303','G','45623175','SBI2C10'),

('C6343','P2305','R','45623785','SBI2C11'),

('C6344','P2307','P','45785178','SBI2C11'),

('C6345','P2309','P','45623736','SBI2C10'),

('C6346','P2311','P','45623254','SBI2C10'),

('C6347','P2313','R','45623171','SBI2C12'),

('C6348','P2315','R','45624697','SBI2C12'),

('C6349','P2317','P','45623643','SBI2C10'),

('C6350','P2319','P','45623436','SBI2C12');

insert into employee values

('EM311','P2302','Waiter',10000),

('EM312','P2304','Designer',20000),

('EM313','P2306','Manager',40000),

('EM314','P2308','Supervisor',32000),

('EM315','P2310','Waiter',12000),

('EM316','P2312','Designer',25000),

('EM317','P2314','Supervisor',38000),

('EM318','P2316','Supervisor',21000),

('EM319','P2318','Waiter',9000),

('EM320','P2320','Supervisor',18000);

insert into venue values

('V7491',150,80000,'Kochi','EM314'),

('V7492',250,100000,'Vizag','EM317'),

('V7493',100,65000,'Delhi','EM318'),

('V7494',150,90000,'Goa','EM314'),

('V7495',300,175000,'Vizag','EM320');

insert into events values

('EV021','Marriage','M','12-08-2020','C6344','V7493','EM318',128000),

('EV022','Farewell','H','04-09-2020','C6347','V7495','EM314',412000),

('EV023','Birthday','L','25-08-2020','C6348','V7491','EM320',88000),

('EV024','Marriage','H','15-09-2020','C6342','V7492','EM318',206000),

('EV025','Trade Expo','M','01-08-2020','C6345','V7491','EM317',156000);

insert into subevents values

('SE811','Lunch',2,65000,'EV021'),

('SE812','Dance',1,20000,'EV021'),

('SE813','Sangeeth',2,32000,'EV021'),

('SE821','Dinner',2,25000,'EV023'),

('SE822','Magic Show',1,10000,'EV023'),

('SE823','Competition',1,12000,'EV023'),

('SE831','Dinner',2,63000,'EV022'),

('SE832','Conference',1,6000,'EV022'),

('SE833','Music',2,25000,'EV022'),

('SE841','Dinner',2,65000,'EV024'),

('SE842','Talent Show',2,20000,'EV024'),

('SE851','Auto Show',4,95000,'EV025'),

('SE852','Tech exhibition',3,32500,'EV025'),

('SE853','Talk',2,75000,'EV025');

insert into chef values

('P2327','Biryani',4.0),

('P2324','Kheer',3),

('P2318','Paneer Masala',4),

('P2325','Butter Naan',3),

('P2306','Chicken 65',4),

('P2310','Rice',5);

insert into catering values

('F7623','Food set-1','NorthIndia','P2324','SE821'),

('F7624','Food set-2','SouthIndia','P2318','SE841'),

('F7625','Food set-3','Chinese','P2325','SE811'),

('F7626','Food set-4','Mexican','P2310','SE831');

insert into performer values

('P2321','Anchor','4.2'),

('P2309','Singer','4.8'),

('P2322','Singer','4.6'),

('P2304','Dancer','3.5'),

('P2327','Mime','3.2'),

('P2328','Dancer','2.9'),

('P2313','Comedian','3.8'),

('P2330','Artist','4.0'),

('P2315','Singer','2.2'),

('P2311','Artist','4.9');

insert into guest values

('P2301','famous writer'),

('P2329','forbes #42'),

('P2314','oscar winner'),

('P2307','Multi-Talented'),

('P2320','tech genius');

insert into conferences values

('CO503','Seminar','Climate Changes','P2301','SE832'),

('CO504','Talk','Dev Hacks','P2320','SE853');

insert into cultural values

('CU884','Music',1,'Carnatic','P2309','SE813'),

('CU885','Music',2,'Tollywood','P2328','SE833'),

('CU886','Dance',1,'Hip-Hop','P2304','SE812'),

('CU887','Music',2,'Tollywood','P2315','SE833');

insert into competitions values

('COM14','Eating','Eat till win','P2321','SE823');

insert into shows values

('SH254','Magic','The greatest showman','P2330','SE822'),

('SH255','Talent','Show me what you got','P2311','SE842'),

('SH256','Auto Expo','Best of the best','P2321','SE851'),

('SH257','Techi1a','The Tech expo','P2330','SE852');

## DQL Commands

### Selection

1. Group by…. having
   1. **Query:** select ctype,count(c\_id) from customer group by ctype,bank\_ifsc having bank\_ifsc='SBI2C10';
   2. **Use-Case:** Find the count of each category type of customers who got accounts in the bank having ifsc=’SBI2C10’.
2. Order by
   1. **Query:** select ev\_id,category,totalcost from events order by totalcost desc;
   2. **Use-Case:** List EventID, Category type, total cost of the event in descending order of total cost.
3. Join
   1. **Query:** select pname from customer natural join person where ctype='P';
   2. **Use-Case:** Display names of all Premium customers.
4. Aggregate function
   1. **Query:** select max(salary) from employee group by job having job='Supervisor';
   2. **Use-Case:** Find the highest salary of Supervisors.
5. A query having Boolean operators
   1. **Query:** select pname,city from person natural join customer where ctype='P' and (city='Vizag' or city='Goa');
   2. **Use-Case:** Display all customers living in vizag or goa and a premium user.
6. A query having arithmetic operators
   1. **Query:** select em\_id,salary\*0.25 as incremented\_Salary from employee where job='Waiter';
   2. **Use-Case:** Calculate salary increment by 25% for the waiters.
7. A search query using string operators
   1. **Query:** select upper(pname),account\_no||'-'||bank\_ifsc as bank\_details from person natural join customer where pname like 'S%';
   2. **Use-Case:** Select customer name in uppercase and bank details in the format of account number-ifsc whose names start with S.
8. Usage of to\_char, extract
   1. **Query:** select ev\_id,to\_char(date,'YYYY-DD-MM') as given\_format,extract(month from date) as month from events order by date;
   2. **Use-Case:** Display eventID,date in format of(year/date/month) and month of all events.
9. Between, in, not between, not in
   1. **Query:** select v\_id,address from venue where (price between 70000 and 100000) and (capacity not between 100 and 200) ;
   2. **Use-Case:**  Display venueID,address where the price is in [70000,100000] and capacity less than 100 or greater than 200.
10. Set operations
    1. **Query:** select p\_id from employee intersect select per\_id from performer;
    2. **Use-Case:** Display the personID who is an employee and also performs.

### 