DBMS Project Report

PES University

Database Management Systems

UE18CS252

Submitted By

PES2201800006 AISHWARYA RAMANATH SHANBHAG

EVENT MANAGEMENT SYSTEM

INTRODUCTION

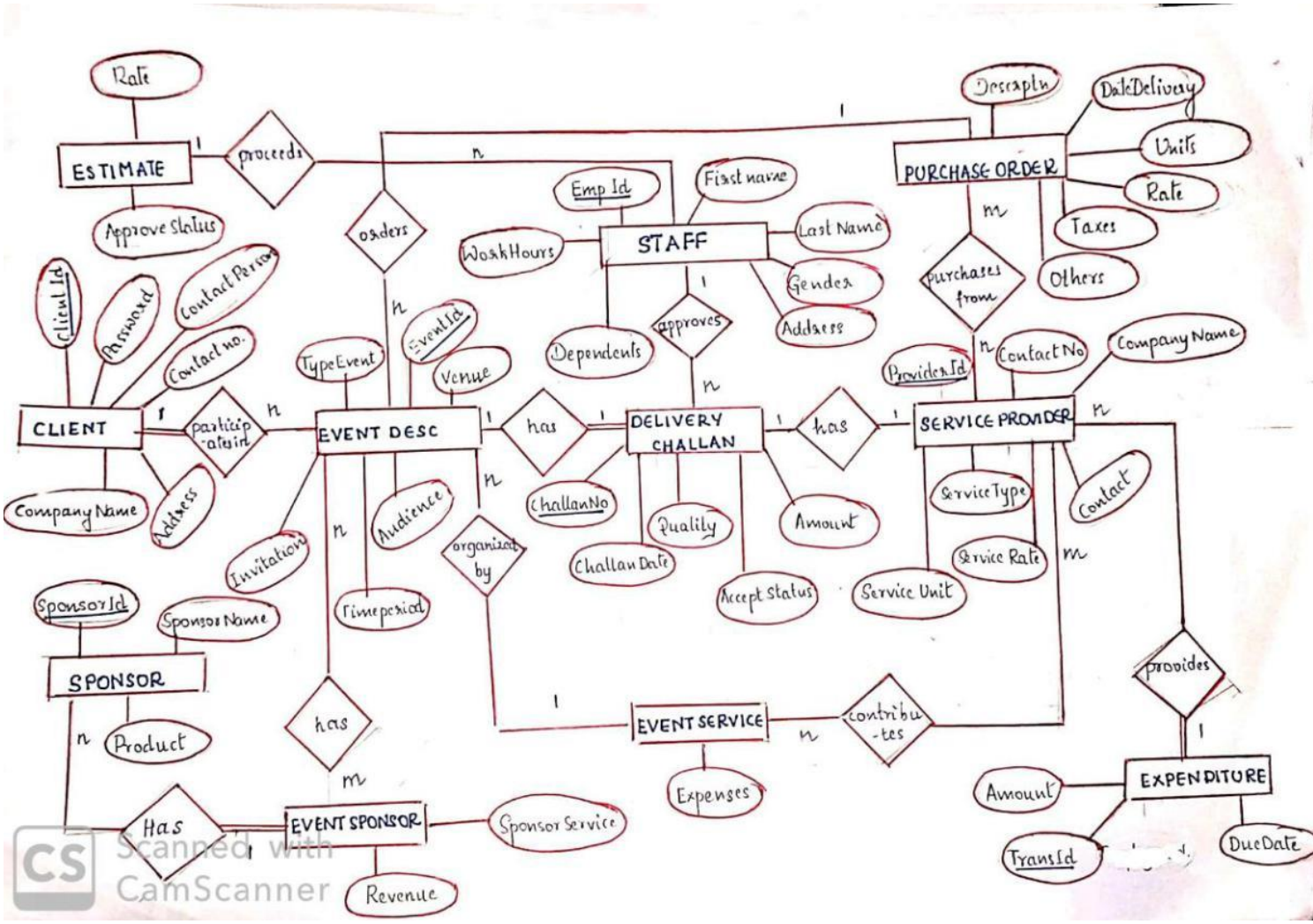
Event management system is used to manage all the activity related to event. In any event many service providers work simultaneously and it is very hard to manage these providers. It is also important for event organizer that he has all the contacts details of these service providers so that he can contact them any time to plan an event at given time. To manage all these activities we have developed this software. To get success in the event management

business, user should have strong network contacts of service provider. These contacts are essentially providers of specific services who can be mobilized quickly to participate in any given event. In present system event company have to do all management work manually. They keep all payment information on papers. There is no system to check the past expenses on any event. To do this they have to check payment register and this task is very time consuming and tiresome. Keeping all these problems in mind we have developed this system. This system helps the event management company to manage their paper work online and they can also retrieve report of last event they have completed.

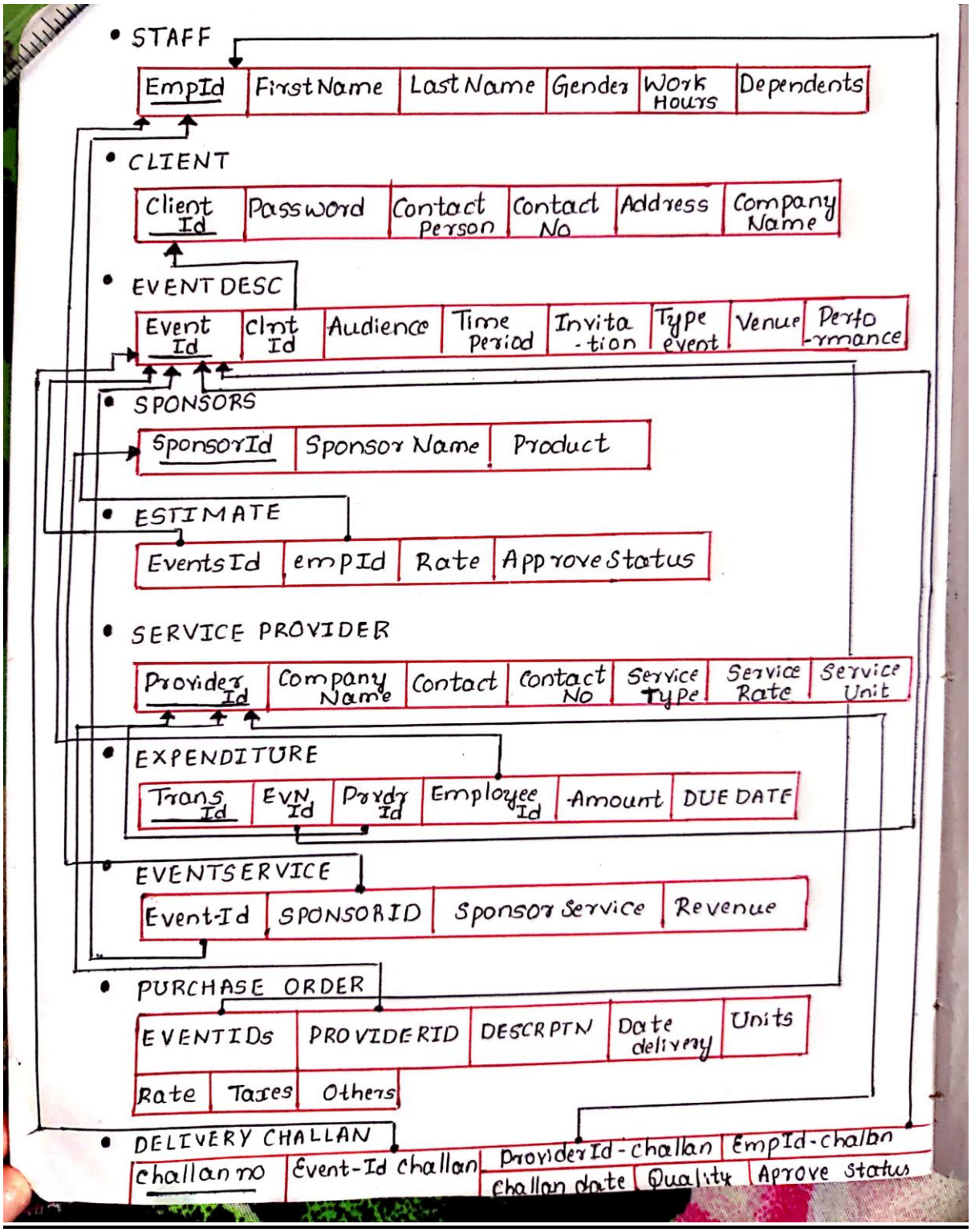
- Event management database keeps track of clients, providers, sponsors and staffs.
- This database contains details of the employees in the table with each having unique id's. Here the employees we mentioned will be heading each event individually.
- There are going to be clients asking for their service and they are given with unique client id's. They are supposed to give some details such as contact numbers, contact person, their company's address to the database management system.
- We have described each event with unique event id and additional details such as the type of events, number of audience expected, invitation, venue of the event etc.
- Here we even accept sponsors for different events. The sponsors are given with unique sponsor id. The database contains information about the products they sponsor.
- Even the database has a separate table for estimation of the cost required for the event and it needs to be approved by the employee leading the team.
- To make an event successful event manager needs different service provider like Sound systems services, Lighting providers, Canteen services, Stage construction and so on. In present system we have a table for the service providers and each provider is given with a unique provider id. The table also contains information about the service they provide, number of units supplied, cost of the units etc.
- We have a separate table for expenditure which keeps track of the money actually spent on each event with unique transaction id. It also tells about the duedate before which the payment has to be done to the providers.
- It also keeps track of the revenue generated for event sponsors in a respective table.
- The database also keeps track of the purchases done in the table purchase order. It holds a description about the same and it tells about the date of delivery, rate of each unit, taxes on each item, provider id regarding a particular event.
- All the payments and transactions are done through challans. The database keeps track of it in a table called deliverychallan which contains all the information regarding the transactions such as the event id for which it was issued, the challan number ,amount submitted, in favour of the particular provider id and the approval status regarding the challan.

DATA MODEL

E.R diagram of event management system



RELATIONAL SCHEMA FOR EVENT DATABASE SYSTEM



FD and Normalization

A functional dependency is a relationship between 2 attributes typically between the foreign key and other non key attributes within a table.

Our database contains the following functional

dependencies

1)TABLE CLIENT

{ ClientId → ContactNo}

2)TABLE EVENTDESC

{ EventId → ClntId, ClntId → EventId}

3)TABLE SPONSORS

{ SponsorId → SponsorName}

4)TABLE ESTIMATE

{ EventsId → EMPiD}

5)TABLE SERVICEPROVIDER

{ ProviderId → CompanyName, CompanyName → ContactNo, ProviderId → ContactNo }

6)TABLE EXPENDITURE

{ TransId → EVNId, EVNId → PrvdrId, TransId → PrvdrId}

7)TABLE EventService

{ EventID → ProvdriD}

8)TABLE EVENTSPONSOR

{ EVENT_Id → SponsorID}

9)TABLE PURCHASEORDER

{ EVENTIDs → PROVIDERID}

10)TABLE DELIVERYCHALLAN

{ ChallanNo → EventId_challan, EventId_challan → ProviderId_challan, ChallanNo → ProviderId }

On converting the ER model to relational schema following all the rules of ER model to relational schema mapping , the relations obtained are not in the normal form.

Our database is not in First Normal Form because of the presence of multi valued attributes in the Table names CLIENT. This table contains an attribute contact no , though it is unique there are chances of presence of multiple contact numbers that the client might submit.

Though ,Values stored in the columns are of same domain and All the columns in the table has unique names. But still due to the above reason it can't be in first normal form.

The tables such as EVENTDESC and SPONSORS have a partially dependent functional dependency, which therefore leads to the violation of second normal form.

If the database is not in second normal form, then we can obviously say that the database is not in 3rd normal form, fourth and so on.

Also in the tables such as DELIVEYCHALLAN, EXPENDITURE and SERVICEPROVIDER the fd's are of the form X → Y ,Y → Z and X → Z, which represents clear transitive dependency ,hence violating the condition of the third normal form.

DDL

CREATING THE DATABASE AND THE TABLES

Create database event_management

Use event_management

CREATE TABLE STAFF

(EmpId Varchar(50) NOT NULL,

FirstName Varchar(30) ,

LastName Varchar((30),

Gender Varchar((10),

Address Varchar(10),

WorkHours int,

dependents int,
PRIMARY KEY(EmpId));

CREATE TABLE CLIENT(

ClientId Varchar(50) not null,

Password Varchar(30) not null,

ContactPerson Varchar(50),

ContactNo Varchar(11) Unique ,

Address Varchar(100),

CompanyName Varchar(80),

PRIMARY KEY(ClientId));

CREATE TABLE EVENTDESC(

EventId int NOT NULL,

ClntId Varchar(50),

Audience Varchar(50),

TimePeriod Varchar(10),

Invitation Varchar(50),

TypeEvent Varchar(30),

Venue Varchar(100),

Performance Varchar(100),PRIMARY KEY(EventId));

CREATE TABLE SPONSORS(

SponsorId int NOT NULL,

SponsorName Varchar(100),

Product Varchar(100),PRIMARY KEY(SponsorId));

CREATE TABLE ESTIMATE(

EventsId int,

emPiD Varchar(50),

Rate int,

ApproveStatus BIT);

CREATE TABLE SERVICEPROVIDER

(ProviderId int NOT NULL,

CompanyName Varchar(100),

Contact Varchar(50),

ContactNo Varchar(11),

ServiceType Varchar(100),

ServiceRate int,

ServiceUnit Varchar(20),Primary key (ProviderId));

CREATE TABLE EXPENDITURE

(TransId int NOT NULL,

EVNId int,

PrvdrId int,

EmployeeId Varchar(50),

Amount int, DUEDATE varchar(50), primary key(TransId));

CREATE TABLE EVENTSERVICE
(

EventID int,

ProvdriD int,

Expenses int);

CREATE TABLE EventSponsor

(

EVENT_Id int,

SponsorID int,

SponsorService Varchar(50),

Revenue int);

CREATE TABLE PURCHASEORDER(

EVENTIDs int,

PROVIDERID int,

Descrptn Varchar(300),

DateDelivery VARCHAR(30),

Units int,

Rate int,

Taxes int,

Others Varchar(20));

CREATE TABLE DELIVERYCHALLAN(

ChallanNo int NOT NULL,

EventId_challan int,

ProviderId_challan int,

EmpId_challan Varchar(50),

ChallanDate VARCHAR(20),

QualityBit,

Amount int,

AcceptStatus Bit,PRIMARY KEY(ChallanNo));

ADDING CHECK AND REFERENTIAL INTEGRITY CONSTRAINTS:

ALTER TABLE PURCHASEORDER ADD CONSTRAINT taxes check(Taxes<=30)

ALTER TABLE DELIVERYCHALLAN ADD CONSTRAINT Amount check(Amount>1000 and Amount<1000000)

ALTER TABLE EVENTDESC ADD CONSTRAINT ClntId FOREIGN KEY(ClntId)references CLIENT(ClientId)on delete cascade on update cascade

ALTER TABLE ESTIMATE ADD CONSTRAINT EventsId FOREIGN KEY(EventsId)references EVENTDESC(EventId)on delete cascade on update cascade

ALTER TABLE ESTIMATE ADD CONSTRAINT emPiD FOREIGN KEY(emPiD)references STAFF(EmpId)on delete cascade on update cascade

ALTER TABLE EXPENDITURE ADD CONSTRAINT EmployeeId FOREIGN KEY(EmployeeId)references STAFF(EmpId)on delete cascade on update cascade

ALTER TABLE EXPENDITURE ADD CONSTRAINT EVNId FOREIGN KEY(EVNId)references EVENTDESC(EventId)on delete cascade on update cascade

ALTER TABLE EXPENDITURE ADD CONSTRAINT PrvdrId FOREIGN KEY(PrvdrId)references SERVICEPROVIDER(ProviderId)on delete cascade on update cascade

ALTER TABLE EVENTSERVICE ADD CONSTRAINT ProvdriD FOREIGN KEY(ProvdriD)references SERVICEPROVIDER(ProviderId)on delete cascade on update cascade

ALTER TABLE EVENTSERVICE ADD CONSTRAINT EventID FOREIGN KEY(EventID)references EVENTDESC(EventId)on delete cascade on update cascade

ALTER TABLE EventSponsor ADD CONSTRAINT SponsorID FOREIGN KEY(SponsorID)references SPONSORS(SponsorId)on delete cascade on update cascade

ALTER TABLE EventSponsor ADD CONSTRAINT EVENT_Id FOREIGN KEY(EVENT_Id)references EVENTDESC(EventId)on delete cascade on update cascade

ALTER TABLE PURCHASEORDER ADD CONSTRAINT EVENTIDs FOREIGN KEY(EVENTIDs)references EVENTDESC(EventId)on delete cascade on update cascade

ALTER TABLE PURCHASEORDER ADD CONSTRAINT PROVIDERID FOREIGN KEY(PROVIDERID)references SERVICEPROVIDER(ProviderId)on delete cascade on update cascade

ALTER TABLE DELIVERYCHALLAN ADD CONSTRAINT ProviderId_challan FOREIGN KEY(ProviderId_challan)references SERVICEPROVIDER(ProviderId)on delete cascade on update cascade

ALTER TABLE DELIVERYCHALLAN ADD CONSTRAINT EventId_challan FOREIGN KEY(EventId_challan)references EVENTDESC(EventId)on delete cascade on update cascade

ALTER TABLE DELIVERYCHALLAN ADD CONSTRAINT EmpId_challan FOREIGN KEY(EmpId_challan)references STAFF(EmpId)on delete cascade on update cascade

STATEMENTS FOR INSERTING VALUES INTO THE TABLES

INSERT INTO STAFF

VALUES('BZ113','Natasha','Dsilva','Female','Bangalore','6','3'),

('BZ215','Niyathi','Shanbag','Female','Mangalore','7','2'),

('BZ116','Viksith','Hegde','male','Bangalore','9','3'),

('BZ998','Mokshith','Kapoor','male','mumbai','6','3'),

('BZ118','Arnav','Gupta','male','Bangalore','6','3');

INSERT INTO CLIENT

VALUES('CL001','974A','MR.SHARMA','1134','MUMBAI','FUTURE PHARMA'),

('CL002','127B','MR.KHAN','6859','UDUPI','EASY FIT'),

('CL003','792C','MRS.PHILOMINA','9135','CHENNAI','STAR CAFE'),

('CL004','099D','MR.HEGDE','2934','AGRA','HEGDE AND COMPANY'),

('CL005','452E','MR.RAO','6312','UDUPI','LIVE LOVE LAUGH');

INSERT INTO EVENTDESC

VALUES('001','CL001','BELOVE THOUSAND','5HRS','E-MAIL','AWARD-FUNCTION','BANGALORE','GOOD'),

('013','CL004','BELOVE THREE HUNDRED','3HRS','HANDBILLS','AWARD-FUNCTION','MANGALORE','BEST'),

('095','CL005','BELOVE FIVE THOUSAND','2HRS','PERSONALLY SENT','INNAGURATION','AGRA','POOR'),

('032','CL002','BELOVE SEVEN HUNDRED','3HRS','E-MAIL','FELICITATION','PUNE','AVG');

INSERT INTO SPONSORS

VALUES('114','DAIRY DAY','DAIRY PRODUCTS'),

('232','EKAM LEARNING','EDUCATIONAL'),

('512','HUDA','COSMETICS'),

('119','LENOVO','ELECTRONICS'),

('239','YONEX','SPORTS');

INSERT INTO ESTIMATE

VALUES('001','BZ113','300000','1'),

('013','BZ215','100000','1'),

('095','BZ998','200000','1'),

('032','BZ118','350000','0');

```
INSERT INTO SERVICEPROVIDER

VALUES('5113','SERVICE SUPER FAST','SALES
DEPARTMENT','11356','HOSPITALITY','100000','ABOVE 100'),

('5115','CHEF INDIA','SERVICE
DEPARTMENT','11356','CATERING','50000','ABOVE 150'),

('5117','MANAPURAM GOLD LOAN','MD','11356','FINANCIAL','30000','ABOVE 200'),

('5118','EASY GO','SALES DEPARTMENT','11356','LOGISTICS','45000','ABOVE 190');



INSERT INTO EXPENDITURE

VALUES('131','013','5115','BZ215','350000','22 JULY 2020'),

('146','095','5117','BZ116','230000','22 SEPT 2020'),

('235','032','5113','BZ998','450000','25 NOV 2020');

INSERT INTO EVENTSERVICE

VALUES('013','5115','350000'),

('095','5117','230000'),

('032','5113','450000');

INSERT INTO EventSponsor

values('013','114','GIFT HAMPER','13000'),

('095','232','FOOD','20000'),

('032','512','SECURITY','50000');

INSERT INTO PURCHASEORDER

VALUES('013','5115','FOOD PLATES','22 MAY 2020','52','150','20','PACKING'),

('095','5117','LUCKY COUPONS','29 JUNE 2020','35','2000','10',NULL),

('032','5113','CHAIRS','21 MAY 2020','32','30','18',NULL);
```

```
INSERT INTO DELIVERYCHALLAN
VALUES('3219','013','5115','BZ215','15 may 2020','1','93600','1'),
('1536','095','5117','BZ116','16 may 2020','1','77150','1'),
('3339','032','5113','BZ998','16 may 2020','1','96000','1');
```

TRIGGERS

1.CREATING A TRIGGER AFTER INSERTING DATA ON TABLE DELIVERYCHALLAN.

```
GO
 CREATE TRIGGER violationafterinsert on DELIVERYCHALLAN AFTER INSERT AS
 DECLARE @Amount int; DECLARE @EventId_challan int; DECLARE @log_action
varchar(20); select @Amount=i.Amount,@EventId_challan=i.EventId_challan from inserted
i;
 set @log_action='inserted record'; if(@Amount<1000 or @Amount>1000000) insert into
violationafterinsert(message,EventId_challan,log_action,log_timestamp)values('rule
violated',@EventId_challan,@log_action,getdate()); else insert into
 violationafterinsert(message,EventId_challan,log_action,log_timestamp)values(",@EventId_
challan,@log_action,getdate()); print'after inserted triggered fired'
```

2.CREATING A TRIGGER IF ANY UPDATE MADE ON DELIVERYCHALLAN TABLE.

```
GO
CREATE TRIGGER violationafterupdate on DELIVERYCHALLAN AFTER update AS
DECLARE @Amount int; DECLARE @EventId_challan int; DECLARE @log_action
varchar(20); select @Amount=i.Amount,@EventId_challan=i.EventId_challan from inserted
i; set @log_action='updated record'; if(@Amount<1000 or @Amount>1000000) insert into
 violationafterupdate(message,EventId_challan,log_action,log_timestamp)values('rule
violated',@EventId_challan,@log_action,getdate()); else insert into
violationafterupdate(message,EventId_challan,log_action,log_timestamp)values(",@EventId
_challan,@log_action,getdate()); print'after updated triggered fired'
```

SQL QUERIES

1)Retrieve the first name,last name , gender , and no of working hours of the staffs who live in Bangalore

```
SELECT FirstName as First_name,Lastname as Last_name ,Gender as gender, WorkHours
as workhour FROM STAFF WHERE Address = 'Bangalore'
```

2) Retrieve id first name and last name of staff who have expenditure Amount>400000

SELECT EmpId as Employee_id,FirstName as First_name,Lastname as Last_name

FROM STAFF AS s WHERE EXISTS (select * FROM EXPENDITURE AS e WHERE e.Amount> 400000 and s.EmpId=e.EmployeeId)

3) Retrieve id first name, last name ,Address and number of working hours whose estimate rate is between 200000 and 300000

SELECT s.EmpId as Employee_id,s.FirstName as First_name,s.Lastname as Last_name ,s.Address as my_address ,s.WorkHours as no_of_hours

FROM STAFF AS s WHERE EXISTS(SELECT* FROM ESTIMATE AS e WHERE (e.Rate BETWEEN 200000 and 300000) and s.EmpId=e.EmpId)

4)Retrieve the Client id ,contact number address and company name whose event desc venue isnt in bangalore

SELECT ClientId as clnt_id, ContactNo as contact_num, Address as address,CompanyName as company_name FROM CLIENT as c WHERE NOT EXISTS(SELECT* FROM EVENTDESC AS e WHERE e.Venue='BANGALORE' and c.ClientId=e.ClntId)

AGGREGATE FUNCTIONS

1)Count the number of staffs and total number of clients

SELECT COUNT(EmpId) from STAFF SELECT COUNT(ClientId) from CLIENT

2)USING ALL THE BASIC AGGREGATE FUNCTIONS

SELECT MAX(Rate) from ESTIMATE SELECT SUM(Rate) from ESTIMATE SELECT MIN(Rate) from ESTIMATE SELECT AVG(Rate) from ESTIMATE SELECT MAX(WorkHours) FROM STAFF SELECT SUM(WorkHours) FROM STAFF SELECT MIN(WorkHours) FROM STAFF SELECT AVG(WorkHours) FROM STAFF

3)GROUP BY FUNCTION -> Find the sum of the workhours by male and female

SELECT Gender ,SUM(WorkHours) from STAFF GROUP BY GENDER

OUTER JOIN FUNCTIONS

1)Retrieve names of EMPLOYEES worked for the event id'095'

select FirstName ,LastName
from (ESTIMATE as E left outer join STAFF as S on S.EmpId=E.emPiD) join EVENTDESC as ED on ED.EventId=E.EventsId where E.EventsId=095

2)Retrieve the sponsor names for the event id'032'

```
select S.SponsorName as SponsorName
 from (EventSponsor as ES right outer join SPONSORS as S on ES.SponsorID =
S.SponsorId)join EVENTDESC as DESCR on DESCR.EventId=ES.Event_ID where
ES.Event_ID=032
```

CONCLUSION

This system provides the following advantages over the traditional paper-based system 1.Robust event reporting 2.Advanced event budget management 3.Bill processing and making is easy. 4.Powerful integrations

Limitations

1.The system could crash or the database may be hacked. 2.The data could be lost due to some error. 3.Failing to update the data on time could lead to multiple payments etc.

Future Scope: The system could be used to run and manage ads on social media. Who needs mailing lists when there are social ads that shape the decisions of event registrants? The future of event management is definitely connected with social media advertising that allows event promoting, indicating who is going to attend, and sharing information.

THANK YOU