

# EVENT MANAGEMENT DATABASE

## Assignment for Database Systems (CS F212)

### Team Members:

Aafia Iqbal	2018A7PS0061U
El Hanoouf Alberia Soufain	2018A7PS0315U
Mardiyah Khadijah	2018A7PS0257U
Safura Ateeq	2018A7PS0080U
Syeda Suha Amber	2018A7PS0057U

### Table of Contents:

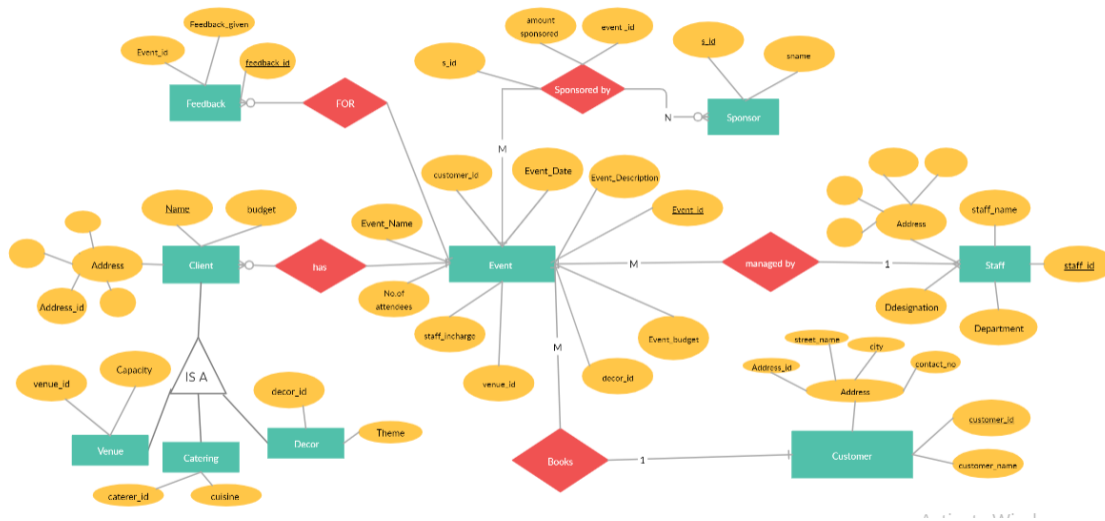
Introduction	2
Entity-Relation Diagram	3
Converting to Relational Model	4
Screenshots	8
Answers to Queries	10
Triggers, Procedures and Functions	13

## Introduction

Create an event management database designed for easier event bookings. The database holds information of event managers and creative specialists who organize weddings, exhibitions, conferences. Design a comprehensive plan to ensure that events are delivered on time and on budget. The event management database has the following requirements:

- Details for every event needs to be stored; each event has an Event ID (unique identifier), name, budget, date of event, number of attendees and a description of the event.
- Each event is managed by one staff member of the event management company. Each staff member has an ID (unique identifier), name, contact number, address, designation and a department they belong to.
- The company only has 3 departments: Finance, Event Management and Support Staff
- Every event may have only one of each: venue, catering service and decor.
  - The capacity and the address of the venue should be recorded.
  - The type of cuisine should be recorded by caterers.
  - The decor should record the themes of events they provide for.
- Each event can be booked by only one customer. Each customer has an ID (unique identifier), name and address
- Detailed budget for each event is a bonus.
- Each event is sponsored by a number of sponsors. Each sponsor's Sponsor ID (unique identifier), name and amount to be sponsored must be recorded.
- At the end of the event, feedback is given. Each feedback has an ID (unique identifier) and must be identified by the event it is for.

# Entity-Relation Diagram



## Converting to Relational Model

Keeping the ER diagram that we have designed in mind, we have decided to convert the entities into the form of tables or relations. All relations except “sponsored\_by” are one-many relations and are represented by foreign keys.

All relations have been normalized to 3NF. The following are the required functional dependencies. Accordingly, we have formed the tables.

(Staff\_id) -> (staff\_name, department, designation, address\_id)  
(Address\_id) -> (Street\_name, City, Contact\_number)  
(Venue\_id) -> (Name, Budget, Capacity, Address\_id)  
(caterer\_id) -> (Name, Cuisine, Budget, address\_id)  
(Decor\_id) -> (Name, Theme, Budget, address\_id)  
(customer\_id) -> (customer\_name, address\_id)  
(S\_id) -> (S\_name)  
(Feedback\_id) -> (feedback\_given, Event\_id)  
(Event\_id, Staff\_incharge, Customer\_id) -> (Event\_Name, Event\_Date, Event\_Description, Event\_Budget, No.\_of\_attendees, Venue\_id, Caterer\_id, Decor\_id)  
(s\_id, event\_id) -> (Amount\_sponsored)

### **Staff**

Staff_id	staff_name	department	Designation	Address_id
Int	varchar(50)	varchar(50)	varchar(50)	int
Primary key				Foreign key Refers to Address_id in table Address

### **Address**

Address_id	Street_name	City	Contact_number
int	varchar(50)	varchar(50)	int
Primary key			

**Venue**

Venue_id	Name	Capacity	Budget	Address_id
int	varchar(50)	int	int	int
Primary key				Foreign key Refers to Address_id in table Address

**Catering**

Caterer_id	Name	Cuisine	Budget	Address_id
int	varchar(50)	varchar(50)	int	int
Primary key				Foreign key Refers to Address_id in table Address

**Decor**

Decor_id	Name	Theme	Budget	Address_id
int	varchar(50)	varchar(50)	int	int
Primary Key				Foreign key Refers to Address_id in table Address

**Customer**

customer_id	customer_name	Address_id
int	varchar(50)	int
Primary Key		Foreign key Refers to Address_id in table Address

**Sponsor**

S_id	S_name
int	varchar(50)
Primary Key	

**Feedback**

feedback_id	feedback_given	Event_id
int	varchar(150)	int
Primary key		Foreign key Refers to Event_id in table event Primary Key

**Event**

Event_id	Event_Name	Event_Date	Event_Descripti on	No._of_attendes
int	varchar(50)	date	varchar(150)	int
Primary key				

Table event continued...

Staff_incharge	Customer_id	Venue_id	Caterer_id	Decor_id
int	int	int	int	int
Foreign key Refers to Staff_id in table Staff	Foreign key Refers to Customer_id in table Customer	Foreign key Refers to Venue_id in table Venue	Foreign key refers to caterer_id in table caterer	Foreign key Refers to Decor_id in table Decor

**Sponsor\_by**

S_id	Event_id	Amount_sponsored
Int	Int	Int
Foreign key Refers to S_id in table sponsor  Primary key	Foreign key Refers to Event_id in table event  Primary key	

**Budget**

Event_id	Catering Per Person	No of Attendees	Total Catering	Venue	Decor	Total
int	int	Int	int	int	int	int
Refers to Event_id In table Event Primary Key						

The Budget table is created on inserting a record in the Event table. The set\_Event\_Budget trigger is responsible for this.

There are 3 procedures designed for this database:

1. A procedure that displays the list of details for given caterer including Caterer name, Caterer Id and the cuisine
2. A procedure to change the theme of decorator with ID and theme given as parameters
3. A procedure to the Customer of any given event

There is 1 function designed for this database:

1. A function which when given the name of a department, returns the count of the number of staff in that department

# Screenshots

## 1. Table Event: all rows

```
157
158 • select * from Evnt;
159 • select * from Budget;
```

Event_Id	Event_Name	Event_Date	Event_Description	No_of_attendees	Staff_Incharge	Customer_Id	Venue_Id	Caterer_Id	Decor_Id
601	Wedding	2020-02-20	Brahmin Style Indian Wedding	500	201	561	310	402	503
602	Baby Shower	2020-04-18	Baby Shower for Baby Boy	100	202	531	305	404	504
603	Proposal	2020-02-14	Proposal in front of Burj Khalifa	40	203	551	303	406	503
604	Engagement	2020-03-28	Ring Ceremony	200	204	521	307	408	506
605	Conference	2020-05-20	Conference for Global Leaders	300	205	541	309	410	509
606	HouseWarming	2020-02-02	Inauguration of new house and housewarming ...	50	206	521	302	412	510
607	Wedding	2020-06-17	Minimalistic Wedding Ceremony	300	201	581	311	414	508
608	Wedding	2020-06-05	Muslim Wedding , separate areas for men and ...	400	201	591	306	403	508
609	Birthday	2020-02-20	Birthday Party for a 2 year old Boy	100	207	571	305	401	501
610	Anniversary	2020-05-25	Silver Wedding Jubilee	200	207	521	311	407	502
611	Inauguration	2020-05-13	Inauguration of new office	60	208	541	308	405	509
612	Birthday	2020-07-05	Birthday of a 75 year old man	100	207	571	311	409	505
613	Proposal	2020-04-17	Proposal on a yacht	20	203	581	304	411	508
614	Exhibition	2020-07-05	Art Exhibition for local artists	100	208	541	309	413	509
615	Wedding	2020-06-05	Christian Wedding Ceremony	400	201	551	311	402	503
616	Birthday	2020-06-18	Birthday of a 5 year old girl	100	207	531	305	401	501
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 2. Table Budget: all rows inserted by trigger

```
157
158 • select * from Evnt;
159 • select * from Budget;
```

Event_id	CateringPerPerson	No_of_Attendees	TotalCatering	Venue	Decor	Total
601	60	500	30000	300	2500	32800
602	65	100	6500	1200	300	8000
603	55	40	2200	750	2500	5450
604	65	200	13000	550	750	14300
605	80	300	24000	3500	1500	29000
606	60	50	3000	1000	250	4250
607	70	300	21000	5000	2000	28000
608	55	400	22000	4500	2000	28500
609	50	100	5000	1200	500	6700
610	70	200	14000	5000	700	19700
611	50	60	3000	1000	1500	5500
612	60	100	6000	5000	500	11500
613	70	20	1400	1500	2000	4900
614	70	100	7000	3500	1500	12000
615	60	400	24000	5000	2500	31500
616	50	100	5000	1200	500	6700
NULL	NULL	NULL	NULL	NULL	NULL	NULL



### 3. Staff details with address:

```
160 • select *
161 from Staff s, Address a
162 where s.address_id = a.address_id;
```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:									
Staff_Id	Staff_Name	Department	Designation	Address_id	Address_id	Street_name	City	Contact_number	
201	Jack Miller	Finance	Accountant	101	101	Sheikh Zayed Road	Dubai	555551874	
202	Tom Hank	Finance	Financial Manager	102	102	Tudor Road	Abu Dhabi	525558685	
203	Fin Ray	Finance	Internal Auditor	103	103	Al Najdah Road	Ajman	505551316	
204	Mary Ross	Event Management	Event Manager	104	104	New Corniche Rd	Fujairah	585550518	
205	Nick Mark	Event Management	Event coordinator	105	105	Nahda Street	Dubai	585556247	
206	Juliet Daniel	Event Management	PR Manager	106	106	Al Barsha Road	Dubai	505556571	
207	Emily Smith	Support Staff	Administrator	107	107	Al Quoz Street	Dubai	505556727	
208	Danny Wilson	Support Staff	Technician	108	108	Arabian Gulf Road	Ajman	525554878	

### 4. Table Venue: all rows

```
163
164 • select *
165 from Venue;
```

Result Grid   Filter Rows:   Edit:   Export/Import:   Wrap Cell Content:					
Venue_Id	Venue_Name	Capacity	Budget	Address_id	
301	Mohammed Bin Rashid Hall	500	2500	117	
302	Hilton Event Hall	150	1000	118	
303	The Venue	100	750	119	
304	The Venue	400	1500	119	
305	Aurora Events	300	1200	120	
306	Mushrif Wedding Hall	700	4500	121	
307	The Domes	100	550	122	
308	The Domes	200	1000	122	
309	The Domes	500	3500	122	
310	Holiday Inn Events	50	300	123	
311	Grand Ballroom	800	5000	124	
* NULL	NULL	NULL	NULL	NULL	

## Answers to Queries

The following queries are solved for the given database:

### 1. Find the event with the maximum budget.

```
1 • Select e.Event_Id, e.Event_Name, e.Event_Description
2 From Evnt e, budget b
3 Where e.Event_Id= b.Event_id and b.total = (select max(b1.total)
4 from budget b1);
5
6
7
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Event_Id	Event_Name	Event_Description	
601	Wedding	Brahmin Style Indian Wedding	

### 2. Find all events where the total event budget is greater than the average of the total event budget at all the events.

```
7
8 • with Event_total(Event_id, value) as (Select Event_id, total from Budget)
9 Select e.Event_Id, e.Event_Name, e.Event_Description
10 From Evnt e, Event_total t
11 Where e.Event_Id= t.Event_id and t.value > (select avg(total) from Budget);
12
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Event_Id	Event_Name	Event_Description	
601	Wedding	Brahmin Style Indian Wedding	
605	Conference	Conference for Global Leaders	
607	Wedding	Minimalistic Wedding Ceremony	
608	Wedding	Muslim Wedding , separate areas for men and ...	
610	Anniversary	Silver Wedding Jubilee	
615	Wedding	Christian Wedding Ceremony	

### 3. Query to show MAX and MIN Budget as well as average capacity for each Venue.

```
13
14 • SELECT MAX(Budget), MIN(Budget), AVG(Capacity), Venue_Name
15 FROM Venue
16 GROUP BY Venue_Name;
17
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MAX(Budget)	MIN(Budget)	AVG(Capacity)	Venue_Name
2500	2500	500.0000	Mohammed Bin Rashid Hall
1000	1000	150.0000	Hilton Event Hall
1500	750	250.0000	The Venue
1200	1200	300.0000	Aurora Events
4500	4500	700.0000	Mushrif Wedding Hall
3500	550	266.6667	The Domes
300	300	50.0000	Holiday Inn Events
5000	5000	800.0000	Grand Ballroom

### 4. Find all events whose staff incharge stays in Dubai.

```
18 • select Event_id, Event_name
19 from evnt e
20 where e.Staff_Incharge In (select staff_id
21 from staff
22 where staff.address_id In(select address_id
23 from address
24 where address.city="Dubai"));
```

26 • select sponsor\_id, sponsor\_name

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Event_id	Event_name
▶	601	Wedding
	607	Wedding
	608	Wedding
	615	Wedding
	605	Conference
	606	HouseWarming
	609	Birthday
	610	Anniversary
	612	Birthday
	616	Birthday

5. Find sponsors who have sponsored 'Birthdays'.

```
26 • select Sponsor_id, Sponsor_name
27 from Sponsor s, sponsor_by s2
28 where s.Sponsor_id = s2.S_Id and s2.event_Id in (select event_id
29 from Evnt e
30 where e.event_name="Birthday");
31
32
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Sponsor_id	Sponsor_name			
810	Cleveland Clinic Abu Dhabi			

6. Find the event with the highest amount sponsored.

```
32
33 • SELECT MAX(Amount_Sponsored)
34 FROM sponsor_by s, Evnt e
35 WHERE s.Event_Id=e.Event_Id;
36
37
38
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
MAX(Amount_Sponsored)				
20000				

## Triggers, Procedures and Functions

### TRIGGER 1:

**To record information in the Budget table when a record in the Event table is inserted.**

delimiter \$\$

```
CREATE TRIGGER Set_Event_Budget
```

```
    after INSERT ON evnt
```

```
    FOR EACH ROW
```

```
BEGIN
```

```
Declare caterer_budget integer;
```

```
Declare decor_budget integer;
```

```
Declare venue_budget integer;
```

```
Declare new_budget integer;
```

```
Select Catering.Budget into caterer_budget from Catering where
```

```
NEW.Caterer_Id=Catering.Caterer_id;
```

```
Select Decor.Budget into decor_budget from Decor where NEW.Decor_Id=Decor.Decor_id;
```

```
Select Venue.Budget into venue_budget from Venue where NEW.Venue_Id=Venue.Venue_id;
```

```
Select ((caterer_budget*NEW.No_of_attendees) + decor_budget + venue_budget ) into  
new_budget;
```

```
insert into Budget values(NEW.Event_id, caterer_budget, NEW.No_of_attendees,  
caterer_budget*NEW.No_of_attendees, venue_budget, decor_budget, new_budget);
```

```
END$$
```

delimiter ;

### PROCEDURE 1:

**The following procedure displays a table consisting of a list of details for given caterer including Caterer name, Caterer Id and the cuisine.**

delimiter \$\$

drop procedure if exists giveCatererDetails \$\$

```
CREATE PROCEDURE giveCatererDetails(  
    IN Cname varchar(50)  
)  
BEGIN  
    SELECT  
        Caterer_name,  
        Caterer_id,  
        Cuisine  
    FROM EventManagement.Catering  
    WHERE  
        Caterer_name = Cname;  
END$$  
delimiter ;
```

//calling procedure

call giveCatererDetails("Keita Catering");

## PROCEDURE 2:

**Create a procedure to change the theme of decor services with ID and theme given as parameters.**

```
delimiter $$
CREATE PROCEDURE changeTheme(
    IN id_no int,
    IN theme_new varchar(50)
)
BEGIN
    Update Decor
    Set Theme=theme_new
    where Decor_Id=id_no;
END$$

delimiter ;

//calling procedure
call changeTheme(507, "Baby Shower");
select * from Decor where Decor_id = 507 ;
```

### PROCEDURE 3:

**Create a procedure to change the customer of any given event.**

DELIMITER \$\$

DROP procedure IF EXISTS ChangeCustomer \$\$

```
CREATE PROCEDURE ChangeCustomer(  
    IN new_cust int,  
    IN evnt_id int  
)  
BEGIN  
    declare old_cust int;  
    select Customer_id into old_cust from evnt where Event_id = evnt_id;  
    if old_cust<>new_cust then  
        UPDATE evnt  
        SET Customer_Id = new_cust  
        WHERE Event_id=evnt_id;  
    end if;  
END$$
```

DELIMITER ;

```
//calling procedure  
call ChangeCustomer(561,601);  
select * from evnt where Event_Id = 601;
```



### FUNCTION 1:

**Define a function which when given the name of a department, returns the count of the number of staff in that department.**

DELIMITER \$\$

```
create function dept_count (  
    dept_name varchar(20)  
)  
returns integer  
READS SQL DATA  
DETERMINISTIC  
begin  
    declare d_count integer;  
  
    select count(*) into d_count  
        from Staff  
        where Staff.department = dept_name;  
  
return d_count;  
End$$
```

DELIMITER ;

```
//calling function  
select dept_count("Finance");
```