

**NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI-15**



Department of Computer Applications

Event Management System

PROJECT WORK

Submitted By

Vijay Lakra – 205112063

Under the guidance of

Ms. R Eswari

Dr. Michael Arock

Submitted in fulfillment of the project in PL/SQL

**NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI-15**



CERTIFICATE

*This is to certify that **Mr.Vijay Lakra**, student of 2nd semester MCA (batch 2012-2015) of National Institute of Technology, Tiruchirappalli has successfully completed the project **Event Management System** in **PL/SQL** under the guidance of **Dr. Micheal Arock and Ms.R Eswari**.*

Date: 22-Apr-2013

Signature

(Ms.R Eswari)

CONTENTS

S_NO	Particulars	Page
1.	Introduction	4
2.	Problem Description and Analysis	5
3.	Schema and Tables Description	6-10
4.	TRIGGERS	11-12
5.	Cursors ,Procedures, Functions and Exceptions	13-38
6.	Enhancement and conclusion	39
7.	References & bibliography	40

INTRODUCTION

Event management is the application of project_management to the creation and development of festivals, events and conferences.

The recent growth of festivals and events as an industry around the world means that the management can no longer be *ad hoc*. Events and festivals, such as the Asian Games, have a large impact on their communities and, in some cases, the whole country.

The industry now includes events of all sizes from the Olympics down to a breakfast meeting for ten business people. Many industries, charitable organizations, and interest groups will hold events of some size in order to market themselves, build business relationships, raise money or celebrate.

Event managers and their teams are often behind-the-scenes running the event. Event managers may also be involved in more than just the planning and execution of the event, but also brand building, marketing and communication strategy. The event manager is an expert at the creative, technical and logistical elements that help an event succeed. This includes event design, audio-visual production, scriptwriting, logistics, budgeting, negotiation and, of course, client service. It is a multi-dimensional profession.

Events can be classified into four broad categories based on their purpose and objective:

1. Leisure events e.g. leisure sport, music, recreation.
2. Cultural events e.g. ceremonial, religious, art, heritage, and folklore.
3. Personal events e.g. weddings, birthdays, anniversaries.
4. Organizational events e.g. commercial, political, charitable, sales, product launch, expo.

Problem Description and Analysis

In today's world everyone is busy with lots of works and 24 hours a day is not enough for them. They used to missed out lots of important works as they spent more time of some other works. Some people used to forget about their schedule also. Some organization find it difficult to organize any events like lions club meeting as that are not sure about who are not coming. My project is a solution to all this problems.

The project entitled "**Event Management System**" is developed using Oracle 10g for storing the details of each users ,their events, organization which want to organize events, the event which have to be scheduled and it helps both the user and the organizer to maintain their schedule properly.

It consists of 8 modules client registration module, to store personal information and password, client order to store the details of order, schedule module to store schedule of client's events, venue module to store the location of events, sponsor module to store cost of sponsorship ,payment to sponsors, employee module to store the information of corresponding employees, bill module to calculate bill, feedback module to store the feedback of user for relative events.

It helps the users to know about their events ,location and it schedule. It even suggest you which events you have to concentrate more and which events we can avoid. All you have to do is just to give information about your events.

It helps the organization to organize the event and organize the sponsorship for that events. All they have to do is to give complete details about their event.

It even give suggestions to organizer regarding the events.

Schema and Tables Description

1. Client_reg

Create table client_reg(

Client_id varchar2(5) primary key,client_pwd varchar2(10),

Client_name varchar2(20),DOB date,city varchar2(20),

Nationality varchar2(20),gender char(2),Phone_no
number(15,0),Email varchar2(25)

);

alter table client_reg ADD constraint gend

check(gender IN('M','m','F','f'))ENABLE;

Name	Null?	Type

CLIENT_ID	NOT NULL	VARCHAR2(5)
CLIENT_PWD		VARCHAR2(10)
CLIENT_NAME		VARCHAR2(20)
DOB		DATE
CITY		VARCHAR2(20)
NATIONALITY		VARCHAR2(20)
GENDER		CHAR(2)
PHONE_NO		NUMBER(15)
EMAIL		VARCHAR2(25)

2. Client_order

Create table client_order(

Client_id varchar(5) not null, order_id varchar(5) primary key,

Typ_event varchar(25),date_evnt date,location

Varchar2(55),budget number(10,2),time timestamp,

Foreign key(client_id) references client_reg

);

Name	Null?	Type
-----	-----	-----
CLIENT_ID	NOT NULL	VARCHAR2(5)
ORDER_ID	NOT NULL	VARCHAR2(5)
TYP_EVENT		VARCHAR2(25)
DATE_EVNT		DATE
LOCATION		VARCHAR2(55)
BUDGET		NUMBER(10,2)
TIME		TIMESTAMP(6)

3. Schedule

Create table schedule(

Order_id varchar2(5) not null,evnt_name char(25),

Date_evnt date,venue varchar2(15),time timestamp,

Foreign key(order_id) references client_order

);

Name	Null?	Type
-----	-----	-----
ORDER_ID	NOT NULL	VARCHAR2(5)
EVNT_NAME		CHAR(25)
DATE_EVNT		DATE
VENUE		VARCHAR2(15)
TIME		TIMESTAMP(6)

4. Venue

Create table venue(evnt_typ char(30) primary key,
capacity number(10),
location varchar2(55)
);

Name	Null?	Type

EVNT_TYP	NOT NULL	CHAR(30)
CAPACITY		NUMBER(10)
LOCATION		VARCHAR2(55)

5. Sponser

Create table sponsor(

Order_id varchar2(5) not null, evnt_name char(25),
Typ_event char(25), spon_cost number(10,2),
spon_pay number(10,2), Date_pay date, date_evnt date,
Foreign key(order_id) references client_order
);

Field Name	Datatype	length	constraints	description	keys
order_id	varchar2	5	NOT NULL	order identification	foreignkey
evnt_name	Char	25		Event name	
Typ_event	Char	25		Type of event	
spon_cost	Number	10		Sponsorship cost Of event	
Spon_pay		10		Payment of money for sponser	
Date_pay	Date			Date of payment	
Date_evnt	Date			Date of event	

6. Bill

Create table bill(

Order_id varchar2(5) not null,

Bill_id varchar2(5) primary key,bill_date date,

Advance number(10,2),rem_amt number(10,2),

Total number(10,2),duedate date,

Foreign key(order_id) references client_order

);

Name	Null?	Type

ORDER_ID	NOT NULL	VARCHAR2(5)
BILL_ID	NOT NULL	VARCHAR2(5)
BILL_DATE		DATE
ADVANCE		NUMBER(10,2)
REM_AMT		NUMBER(10,2)
TOTAL		NUMBER(10,2)
DUEDATE		DATE

7. emp_det

Create table emp_det(

Emp_id varchar2(5) primary key,name varchar2(25),

Design char(25),sal number(8,2),

Address varchar2(20),mobno number(15),email varchar2(30)

);

Name	Null?	Type

EMP_ID	NOT NULL	VARCHAR2(5)
NAME		VARCHAR2(25)
DESIGN		CHAR(25)
SAL		NUMBER(8,2)
ADDRESS		VARCHAR2(20)

MOBNO	NUMBER(15)
EMAIL	VARCHAR2(30)

8. feedback

Create table feedback(

Client_id varchar2 (5) not null,order_id varchar2(5),

Dofb date,evnt_name varchar2(30),credit_rating varchar2(9)

);

alter table feedback ADD constraint credit_check

check(credit_rating IN('EXCELLENT','GOOD','POOR'))ENABLE;

Name	Null?	Type
-----	-----	-----
CLIENT_ID	NOT NULL	VARCHAR2(5)
ORDER_ID		VARCHAR2(5)
DOFB		DATE
EVNT_NAME		VARCHAR2(20)
CREDIT_RATING		VARCHAR2(9)

TRIGGERS

1. date and time validation For table client_order

```
Create or replace trigger odt
    Before insert on client_order for each row

Declare
tm timestamp;
Begin
Select time into tm from client_order;
    If :new.date_evnt<sysdate then
        Raise_application_error(-20002,'Date Expired');
    Elself :new.time=tm then
        Raise_application_error(-20002, 'THERE IS AN EVENT IN SAME DATE');

        end if;
    end;
```

/

2. date and time validation For table schedule

```
Create or replace trigger schdt
    Before insert
    on schedule for each row

Declare
    tm timestamp;

Begin
Select time into tm from schedule;
    If :new.date_evnt<sysdate then
        Raise_application_error(-20002,'Date Expired');
    Elself :new.date=date then
        Raise_application_error(-20002, 'THERE IS AN EVENT IN SAME DATE');

        end if;
    end;
```

3. date validation for table sponsor

```
Create or replace trigger spdt

Before insert on sponser for each row
```

```
                Begin
                    If :new.date_evnt<sysdate then
                        Raise_application_error(-20002,'Date Expired');
                    end if;
                end;
            /
```

4 .MINIMUM SALARY OF EMPLOYEE

```
create or replace trigger min_sal
before INSERT or UPDATE on emp_det
for each row
begin
    if INSERTING then
        if(:new.sal<1000) then
            raise_application_error(-210000,'MINIMUM BALANCE CAN NOT PROCEED');
        end if;
    elsif UPDATING then
        if(:new.sal<1000) then
            raise_application_error(-210000,'MINIMUM BALANCE CAN NOT PROCEED');
        end if;
    end if;
end;
/
```

5. Minimum age

```
Create or replace trigger dob
before INSERT or UPDATE on client_reg
For each row
declare
age number;
Begin
age:=sysdate-:new.dob;
If(:new.dob>sysdate or age<20 or age>85)then
    raise_application_error(-210000,'MINIMUM AGE CAN NOT BE PROCEED');
end if;
end;
/
```

Cursors ,Procedures,Functions and Exceptions

1. Insertion inClient registration

Declare

```
cid client_reg.client_id%type;  
cpwd client_reg.client_pwd%type;  
cname client_reg.client_name%type;  
dob client_reg.dob%type;  
city client_reg.city%type;  
nationality client_reg.nationality%type;  
gender client_reg.gender%type;  
pno client_reg.phone_no%type;  
email client_reg.email%type;  
cursor c_reg is select * from client_reg;  
reg_det c_reg%rowtype;
```

begin

```
cid := '&cid';  
cpwd:= ' &cpwd';  
cname:='&cname';  
dob:= ' &dob';  
city:= ' &city';  
nationality:=' &nationality';  
gender:=' &gender';  
pno:= ' &pno';
```

```
email:=' &email';

open c_reg;

loop

fetch c_reg into reg_det;

exit when c_reg%notfound;

insert into client_reg values(cid,cpwd,cname,dob,city,nationality,
gender,pno,email);

commit;

dbms_output.put_line('client_id:' || reg_det.client_id);

dbms_output.put_line('client_pwd:' || reg_det.client_pwd);

dbms_output.put_line('name:' || reg_det.client_name);

dbms_output.put_line('dob:' || reg_det.dob);

dbms_output.put_line('City:' || reg_det.city);

dbms_output.put_line('nationality:' || reg_det.nationality);

dbms_output.put_line('Gender:' || reg_det.gender);

dbms_output.put_line('Phone_no:' || reg_det.phone_no);

dbms_output.put_line('Email:' || reg_det.email);

dbms_output.put_line('-----');

dbms_output.put_line(' you have successfully registered');

dbms_output.put_line('-----');

end loop;

close c_reg;

end;

/
```

OUTPUT:

```
Enter value for cid: 103
old 14:  cid := '&cid';
new 14:  cid := '103';
Enter value for cpwd: 8529
old 15:  cpwd:=' &cpwd';
new 15:  cpwd:=' 8529';
Enter value for cname: sohel
old 16:  cname:='&cname';
new 16:  cname:='sohel';
Enter value for dob: 24-feb-1986
old 17:  dob:= ' &dob';
new 17:  dob:= ' 24-feb-1986';
Enter value for city: mumbai
old 18:  city:= ' &city';
new 18:  city:= ' mumbai';
Enter value for nationality: indian
old 19:  nationality:=' &nationality'
new 19:  nationality:=' indian';
Enter value for gender: m
old 20:  gender:=' &gender';
new 20:  gender:=' m';
Enter value for pno: 225684
old 21:  pno:=' &pno';
new 21:  pno:=' 225684';
Enter value for email: sohel.ink@gmail.
old 22:  email:=' &email';
new 22:  email:=' sohel.ink@gmail.com
client_id:101
client_pwd: 1234
name:vijay
dob:24-JUN-90
City: rnc
nationality: indian
Gender: m
Phone_no:963134926
Email: vijay.lakra@gmail.com
-----
you have successfully registered
-----
```

```
SQL> select * from client_reg;
```

CLIE	CLIENT_PWD	CLIENT_NAME	DOB	CITY
NATIONALITY	GE	PHONE_NO	EMAIL	
101	1234	vijay	24-JUN-90	rnc
indian	m	963134926	vijay.lakra@gmail.com	
102	4567	vicky	07-JUL-87	pune
indian	m	2235478	viky.info@gmail.com	
103	8529	sohel	24-FEB-86	mumbai
indian	m	225684	sohel.ink@gmail.com	

2. Insertion in client_order:

Create or replace procedure clnord

is

```
cid client_order.client_id%type;
```

```
oid client_order.order_id%type;
```

```
typ client_order.typ_event%type;
```

```
date_evnt client_order.date_evnt%type;
```

```
loc client_order.location%type;
```

```
budget client_order.budget%type;
```

```
tm client_order.time%type;
```

```
cursor reg is select * from client_reg;
```

```
creg reg%rowtype;
```

```
begin
```

```
cid:='&cid';
```

```
oid:='&oid';
```



```
typ:='&typ';  
date_evnt:='&date_evnt';  
loc:='&loc';  
budget:='&budget';  
tm:='&tm';  
  
open reg;  
loop  
fetch reg into creg;  
exit when reg%notfound;  
  
if(creg.client_id=cid)then  
insert into client_order values(cid,oid,typ,  
date_evnt,loc,budget,tm);  
commit;  
end if;  
end loop;  
  
dbms_output.put_line('=====');  
dbms_output.put_line('client_id:| | ' || cid);  
dbms_output.put_line('order_id:| | ' || oid);  
dbms_output.put_line('event type:| | ' || typ);  
dbms_output.put_line('event date:| | ' || date_evnt);  
dbms_output.put_line('location:| | ' || loc);  
dbms_output.put_line('budget:| | ' || budget);
```

```
dbms_output.put_line('time:|'|'|tm);  
  
dbms_output.put_line('=====');  
  
dbms_output.put_line('=====');  
dbms_output.put_line('EVENTS SCHEDULED SUCCESSFULLY');  
dbms_output.put_line('=====');  
  
close reg;  
  
end;  
  
/
```

OUTPUT:

```
Enter value for cid: 101  
old 15: cid:='&cid';  
new 15: cid:='101';  
Enter value for oid: 002  
old 16: oid:='&oid';  
new 16: oid:='002';  
Enter value for typ: computer training  
old 17: typ:='&typ';  
new 17: typ:='computer training';  
Enter value for date_evnt: 20-apr-2013  
old 18: date_evnt:='&date_evnt';  
new 18: date_evnt:='20-apr-2013';  
Enter value for loc: ranchi  
old 19: loc:='&loc';  
new 19: loc:='ranchi';  
Enter value for budget: 50000  
old 20: budget:='&budget';  
new 20: budget:='50000';  
Enter value for tm: 20-apr-2013 10.00.00.00 AM  
old 21: tm:='&tm';  
new 21: tm:='20-apr-2013 10.00.00.00 AM';
```

```
=====
client_id: 101
order_id: 002
event type: computer training
event date: 20-APR-13
location: ranchi
budget: 50000
time: 20-APR-13 10.00.00.000000 AM
=====
```

EVENTS SCHEDULED SUCCESSFULLY

```
=====
```

SQL> select * from client_order;

CLIE	ORDER	TYP_EVENT	DATE_EVNT	LOCATION	BUDGET

TIME					

102	003	communication skill	29-APR-13	delhi	65000
29-APR-13 10.00.00.000000 AM					
103	004	book fair	05-MAY-13	ranchi	64000
05-MAY-13 10.00.00.000000 AM					
101	001	leadership seminar	25-APR-13	trichy	98600
25-APR-13 11.30.00.000000 AM					
CLIE	ORDER	TYP_EVENT	DATE_EVNT	LOCATION	BUDGET

TIME					

101	002	computer training	20-APR-13	ranchi	50000
20-APR-13 10.00.00.000000 AM					

3. Insertion in Schedule

Declare

flag number;

oid schedule.order_id%type;

ename schedule.evnt_name%type;

cursor schd is select * from schedule;

sd schd%rowtype;

cursor or_det is select * from client_order;

ord or_det%rowtype;

begin

oid:='&oid';

ename:='&ename';

open schd;

open or_det;

loop

fetch or_det into ord;

exit when or_det%notfound;

if(ord.order_id=oid)then

insert into schedule(order_id,evnt_name)

values(oid,ename);

update schedule set date_evnt=(select date_evnt from client_order where
order_id=oid);

```
update schedule set time=(select time from client_order where order_id=oid);

update schedule set venue=(select location from client_order where order_id=oid);

commit;

flag:=1;

end if;

end loop;

if(flag=1)then

    dbms_output.put_line('=====');

    dbms_output.put_line('EVENTS SCHEDULED SUCCESSFULLY');

    dbms_output.put_line('=====');

else

    dbms_output.put_line('invalid order id.');
```

end if;

close schd;

close or_det;

end;

/

OUTPUT:

```
Enter value for oid: 003
old 10: oid:='&oid';
new 10: oid:='003';
Enter value for ename: Public Speaking Skills
old 11: ename:='&ename';
new 11: ename:='Public Speaking Skills';
invalid order id.
invalid order id.
invalid order id.
```

```
=====
EVENTS SCHEDULED SUCCESSFULLY
=====
```

PL/SQL procedure successfully completed.

SQL> select * from schedule;

ORDER	EVNT_NAME	DATE_EVNT	VENUE	TIME
003	Public Speaking Skills	29-APR-13	delhi	29-APR-13 10.00.00.000000 AM
004	book fair	05-MAY-13	ranchi	05-MAY-13 10.00.00.000000 AM
002	computer basic	20-APR-13	ranchi	20-APR-13 10.00.00.000000 AM

4. Insertion in sponser table

```
declare
    flag number;
    oid sponsor.order_id%type;
    scost sponsor.spon_cost%type;
    spay sponsor.spon_pay%type;
    dpay sponsor.date_pay%type;
    cursor spns is select * from sponsor;
    sp spns%rowtype;
    cursor or_det is select * from client_order;
    ord or_det%rowtype;

begin
    oid:='&oid';
    scost:='&scost';
    spay:='&spay';
    dpay:='&dpay';

    open spns;
    open or_det;

    loop
        fetch or_det into ord;
        exit when or_det%notfound;
```

```
if(ord.order_id=oid)then
insert into sponsor(order_id,spon_cost,spon_pay,date_pay)
values(oid,scost,spay,dpay);
update sponsor set evnt_name=(select evnt_name from schedule where
order_id=oid)
where order_id=oid;
update sponsor set typ_event=(select typ_event from client_order where
order_id=oid)
where order_id=oid;
update sponsor set date_evnt=(select date_evnt from client_order where
order_id=oid)
where order_id=oid;
commit;
flag:=1;
end if;
end loop;
if(flag=1)then
dbms_output.put_line('=====');
dbms_output.put_line('EVENTS HAS BEEN SUCCESSFULLY SPONSERED');
dbms_output.put_line('=====');
else
dbms_output.put_line('invalid order id. ');
end if;
close spns;
close or_det;
end;
/
```

OUTPUT:

```
Enter value for oid: 001
old 13: oid:='&oid';
new 13: oid:='001';
Enter value for scost: 40000
old 14: scost:='&scost';
new 14: scost:='40000';
Enter value for spay: 35000
old 15: spay:='&spay';
new 15: spay:='35000';
```

```
Enter value for dpay: 28-mar-2013
old 16: dpay:='&dpay';
new 16: dpay:='28-mar-2013';
=====
EVENTS HAS BEEN SUCCESSFULLY SPONSERED
=====
```

SQL> select * from sponsor;

ORDER	EVNT_NAME	TYP_EVENT	SPON_COST	SPON_PAY	DATE_PAY	DATE_EVNT
003	Public Speaking Skills	communication skill	35000	30000	28-MAR-13	29-APR-13
004	book fair	book fair	35000	30000	05-APR-13	05-MAY-13
002	computer basic	computer training	25000	22000	11-MAR-13	20-APR-13

5. Bill calculation

```
declare
    flag number;
    oid bill.order_id%type;
    bid bill.bill_id%type;
    sdate date;
    bud number(10,2);
    spay number(8,0);
    duedate date;
    cursor or_det is select * from client_order;
    ord or_det%rowtype;
    cursor spns is select * from sponsor;
    sp spns%rowtype;

begin
    oid:='&oid';
    bid:='&bid';
    open spns;
```



```
open or_det;

select date_pay into sdate from sponsor where order_id=oid;
select budget into bud from client_order where order_id=oid;
select spon_pay into spay from sponsor where order_id=oid;

    duedate:=sdate+7;
if(duedate>=sysdate)then
loop
    fetch or_det into ord;
    exit when or_det%notfound;

    if(ord.order_id=oid)then

        insert into bill(order_id,bill_id,bill_date)
        values(oid,bid,sysdate);
        update bill set advance=(select spon_pay from sponsor where order_id=oid)
            where order_id=oid;

        update bill set rem_amt=bud-spay
        where order_id=oid;

        update bill set total= (select budget from client_order where order_id=oid)
            where order_id=oid;

        update bill set duedate=duedate
            where order_id=oid;
        commit;

    flag:=1;
    end if;
    end loop;
    close or_det;
    close spns;
    if(flag=1)then
        dbms_output.put_line('=====');
        dbms_output.put_line('BILL HAS BEEN CALCULATE');
        dbms_output.put_line('=====');
```

```

else
dbms_output.put_line('invalid order id.');
```

end if;

```

else
    dbms_output.put_line('=====');
    dbms_output.put_line('EVENT IS CANCELED');
    dbms_output.put_line('=====');
    delete from sponsor where order_id=oid;
    delete from schedule where order_id=oid;
    delete from client_order where order_id=oid;
    commit;

end if;

end;
/
```

OUTPUT:

```

Enter value for oid: 001
old 15:  oid:='&oid';
new 15:  oid:='001';
Enter value for bid: 001
old 16:  bid:='&bid';
new 16:  bid:='001';
=====
EVENT IS CANCELED
=====
```

SQL> select * from sponsor;

ORDER	EVNT_NAME	TYP_EVENT	SPON_COST	SPON_PAY
DATE_PAY	DATE_EVNT			
004	computer basic 25-APR-13 20-APR-13	computer training	25000	15000
003	computer basic	computer training	30000	25000

05-APR-13 20-APR-13

002 computer basic	computer training	25000	25000
28-MAR-13 20-APR-13			

6. ORDER SCHEDULE OF CLIENT

create or replace procedure sch(cid number)

is

flag number;

cursor c1 is select * from client_order;

cc c1%rowtype;

cursor c2 is select * from schedule;

cs c2%rowtype;

begin

open c1;

open c2;

loop

fetch c1 into cc;

exit when c1%notfound;

fetch c2 into cs;

exit when c2%notfound;

if(cc.client_id=cid)then

flag:=1;

end if;

end loop;

close c1;

close c2;

if(flag=1)then

dbms_output.put_line('order_id:|'|'|cs.order_id);

dbms_output.put_line('Event Name:|'|'|cs.evnt_name);

dbms_output.put_line('date:|'|'|cs.date_evnt);

dbms_output.put_line('venue:|'|'|cs.venue);

```
        dbms_output.put_line('time:' || ' ' || cs.time);  
        else  
        dbms_output.put_line('Invalid client id');  
end if;  
end;  
/
```

OUTPUT:

SQL> exec sch(103);

```
order_id: 002  
Event Name: computer basic  
date: 20-APR-13  
venue: ranchi  
time: 20-APR-13 10.00.00.000000 AM
```

7. Get the budget of an event

create or replace procedure budg(evname char)
is
flag number;

```
cursor c1 is select * from client_order;  
cc c1%rowtype;  
cursor c2 is select * from schedule;  
cs c2%rowtype;
```

```
begin  
open c1;  
open c2;
```

```
loop  
    fetch c1 into cc;  
    exit when c1%notfound;  
    fetch c2 into cs;  
    exit when c2%notfound;
```

```
        if(cs.evnt_name=evname)then
            flag:=1;
        end if;
    end loop;
close c1;
close c2;
    if(flag=1)then
        dbms_output.put_line('Budget:' || ' ' || cc.budget);
    else
        dbms_output.put_line('Event not exist');
    end if;
end;
/
```

OUTPUT:

```
SQL> exec budg('book fair');
Budget: 50000
```

7. Insertion into emp_det table

```
declare
eid number:=&emp_id;
begin
    insert into emp_det
    values( eid,&name,&design,&sal,
        &address,&mobno,&email');
dbms_output.put_line('Data entered successfully.');
```

```
end;
/
```

OUTPUT:

```
Enter value for emp_id: 101
old 2: eid number:=&emp_id;
new 2: eid number:=101;
```

```
Enter value for name: vijay
Enter value for design: manager
Enter value for sal: 70000
old 5: values( eid,'&name','&design','&sal',
new 5: values( eid,'vijay','manager','70000',
Enter value for address: trichy
Enter value for mobno: 9631349236
Enter value for email: vijay.lakra96@gmail.com
old 6: '&address','&mobno','&email');
new 6: 'trichy','9631349236','vijay.lakra96@gmail.com');
Data entered successfully.
```

```
SQL> select * from emp_det;
```

EMP_ID	NAME	DESIGN	SAL	ADDRESS	MOBNO	EMAIL
101	vijay	manager	70000	trichy	963134923	vijay.lakra@gmail.com
102	vicky	creative director	56000	trichy	965874265	vicky.creative@yahoo.com
103	victor	coordinator	75000	pune	984563227	victor.info@gmail.com

8. Get salary of employee

```
create or replace function get_sal(eid number) return real
is
```

```
    s real;
```

```
begin
```

```
    select sal into s
```

```
    from emp_det
```

```
    where emp_id=eid;
```

```
    return (s);
```

```
end;
```

```
/
```

```
-----
declare
```

```
eid varchar2(5):=&eid;  
s real;  
begin  
  s:=get_sal(eid);  
  dbms_output.put_line('Salary :'|s);  
end;  
/  
-----
```

OUTPUT:

```
Enter value for eid: 103  
old 2: eid varchar2(5):=&eid;  
new 2: eid varchar2(5):=103;  
Salary :75000
```

8. REMOVAL OF FINISHED EVENTS

```
declare  
flag number;  
cursor or_det is select * from client_order;  
ord or_det%rowtype;  
  
begin  
open or_det;  
loop  
  fetch or_det into ord;  
  exit when or_det%notfound;  
  
  if(ord.date_evnt<sysdate)then  
  
    delete from client_order where date_evnt<sysdate;  
    delete from schedule where date_evnt<sysdate;  
    delete from sponsor where date_evnt<sysdate;  
    flag:=1;  
  end if;  
end loop;  
close or_det;  
  
if(flag=1)then
```

```
dbms_output.put_line('Old event had been removed');  
else  
dbms_output.put_line('There is no old event');  
end if;  
end;  
/
```

OUTPUT:

There is no old event

PL/SQL procedure successfully completed.

9. CANCELATION OF EVENT:

```
create or replace procedure can(oid number)  
  
is  
  
flag number;  
  
cursor or_det is select * from client_order;  
  
ord or_det%rowtype;  
  
begin  
  
open or_det;  
  
loop  
  
fetch or_det into ord;  
  
exit when or_det%notfound;  
  
if(ord.order_id=oid)then  
  
dbms_output.put_line('CANCELATION OF EVENT '||' '||ord.typ_event);  
  
  
  
delete from sponsor where order_id=oid;  
  
delete from schedule where order_id=oid;
```



```
delete from client_order where order_id=oid;

flag:=1;

end if;

end loop;

close or_det;

if(flag=1)then

dbms_output.put_line('CANCEL OF EVENT'||' '||ord.typ_event);

else

dbms_output.put_line('INVALID ORDER ID');

end if;

end;

/
```

OUTPUT:

```
SQL> exec can(002);
```

CANCELATION OF EVENT computer training

CANCEL OF EVENT computer training

10. Update events

```
declare
flag number;
oid varchar2(5);
dt date;
loc varchar2(20);
tm timestamp;

cursor ord_det is select * from client_order;
ord ord_det%rowtype;
begin

oid:='&oid';
```

```
dt:='&dt';
loc:='&loc';
tm:='&tm';
open ord_det;
loop
fetch ord_det into ord;
exit when ord_det%notfound;

if(ord.order_id=oid)then

update client_order set date_evnt=dt
where order_id=oid;
update client_order set location=loc
where order_id=oid;
update schedule set date_evnt=dt
where order_id=oid;
update schedule set venue=loc
where order_id=oid;
update schedule set time=tm
where order_id=oid;

update sponsor set date_evnt=dt
where order_id=oid;
end if;

flag:=1;
end loop;
close ord_det;
if (flag=1)then
dbms_output.put_line('UPDATE SUCCESSFULL');
else
dbms_output.put_line('INVALID ORDER ID');
end if;
end;
/
```

OUTPUT:

```
Enter value for oid: 003
old 11: oid:='&oid';
new 11: oid:='003';
Enter value for dt: 30-apr-2013
old 12: dt:='&dt';
new 12: dt:='30-apr-2013';
```

```
Enter value for loc: delhi
old 13: loc:='&loc';
new 13: loc:='delhi';
Enter value for tm: 30-apr-2013 11.30.00.00 AM
old 14: tm:='&tm';
new 14: tm:='30-apr-2013 11.30.00.00 AM';
```

PL/SQL procedure successfully completed.

SQL> select * from client_order;

CLIEN	ORDER	TYP_EVENT	DATE_EVNT	LOCATION	BUDGET	TIME
102	003	communication skill	30-APR-13	delhi	65000	29-APR-13 10.00.00.000000 AM
103	004	book fair	05-MAY-13	ranchi	640000	5-MAY-13 10.00.00.000000 AM
101	002	computer training	20-APR-13	ranchi	50000	20-APR-13 10.00.00.000000 AM

11.Insertion data into venue

```
declare

typ char(30);

cap number(10);

loc varchar2(55);

begin

insert into venue values('&typ','&cap','&loc');

commit;

end;

/
```

OUTPUT:

```
Enter value for typ: wedding
Enter value for cap: 500
Enter value for loc: Northwest Building,45 Bekery street pune
old 6:      insert into venue values('&typ','&cap','&loc');
new 6:      insert into venue values('wedding','500','Northwest Building,45 Bekery street
pune');
```

```
select * from venue;
```

EVNT_TYP	CAPACITY	LOCATION
meeting and conference	300	Memorial Hall,45 Quincy St, Cambridge MA
catered events	500	Adolphus Busch 29 Kirkland St, Cambridge MA
summer conference	300	The Plaza Tent,1 Oxford Street, Cambridge, MA
wedding	500	Northwest Building,45 Bekery street pune

12.Insert data into feedback:

```
declare
cid varchar2(5);
oid varchar2(5);
flag number;
cursor cor is select * from client_order;
ord cor%rowtype;

begin

cid:='&cid';

oid:='&oid';
```

```
open cor;

loop

fetch cor into ord;

exit when cor%notfound;

if(ord.client_id=cid AND ord.order_id=oid)then

insert into feedback (client_id,order_id,dofb,credit_rating)

values(cid,oid,'&dofb','&credit_rating');

update feedback set evnt_name=(select evnt_name from

sponsor where order_id=oid)

where order_id=oid;

flag:=1;

commit;

end if;

end loop;

close cor;

if(flag=1)then

dbms_output.put_line('Thank You For Your Feedback');

else

dbms_output.put_line('Invalid client id or invalid order id');
```

```
end if;
```

```
end;
```

```
/
```

OUTPUT:

```
Enter value for cid: 101
old 10:  cid:='&cid';
new 10:  cid:='101';
Enter value for oid: 002
old 11:  oid:='&oid';
new 11:  oid:='002';
Enter value for dofb: 14-apr-2013
Enter value for credit_rating: GOOD
old 21:  values(cid,oid,'&dofb','&credit_rating');
new 21:  values(cid,oid,'14-apr-2013','GOOD');
```

Thank You For Your Feedback

```
SQL> select * from feedback;
```

CLIN	ORDER	DOFB	EVNT_NAME	CREDIT_RATING
101	002	14-APR-13	computer basic	GOOD

Enhancement and conclusion

Enhancement:

This project helps you to handle the events properly and maintain your daily schedule .This is mainly to help in manage your timing for all your works. Enhancements like scheduling events for every day, every month can be performed.

It's like how a user can engage to more events in a same day.

1. We can provide the facility of time duration calculation for each event to schedule it for a day to both user and organization.
2. We can help organization to book halls and to choose some proper date for events so that it will help in successful organization.
3. We can event provide the users important key facts of each events so that they may came to know the important.

Conclusion:

This project helped in realizing the concepts of analyzing databases and programming in SQL, PL/SQL. As all the codes are in PL/SQL it helps in updating more than two tables in same calculation. The triggers, procedures and cursors concept are used in this project to make it effective. It provides knowledge about how to schedule events. It helps in utilizing the time in a proper and unconfused way. It helps in making the events successful. As the numbers of people who will come surely are calculated before event day it helps in making better arrangements for them.

References and Bibliography

1. Ivan Bayross," SQL, PL/SQL The programming language of Oracle"
2. Korth, Sudershan," Databse System Concepts"
3. <http://www.java2s.com>