

DBMS - MINI PROJECT

NAME: CHAITRA BHAT
ROLL NO: 5

SRN: PES1UG20CS635
SECTION: K



<https://github.com/Chaitra-Bhat383/College-Fest-Management-System>

ABSTRACT

Fest Management is a Web Application that will be used to handle participant information, event information, organiser details, and create different conclusions. The system will include numerous modules such as a registration system and participation in various activities. The goal of this project is to automate the registration procedure during the fest and manage participation in various activities while preserving a full database and records.

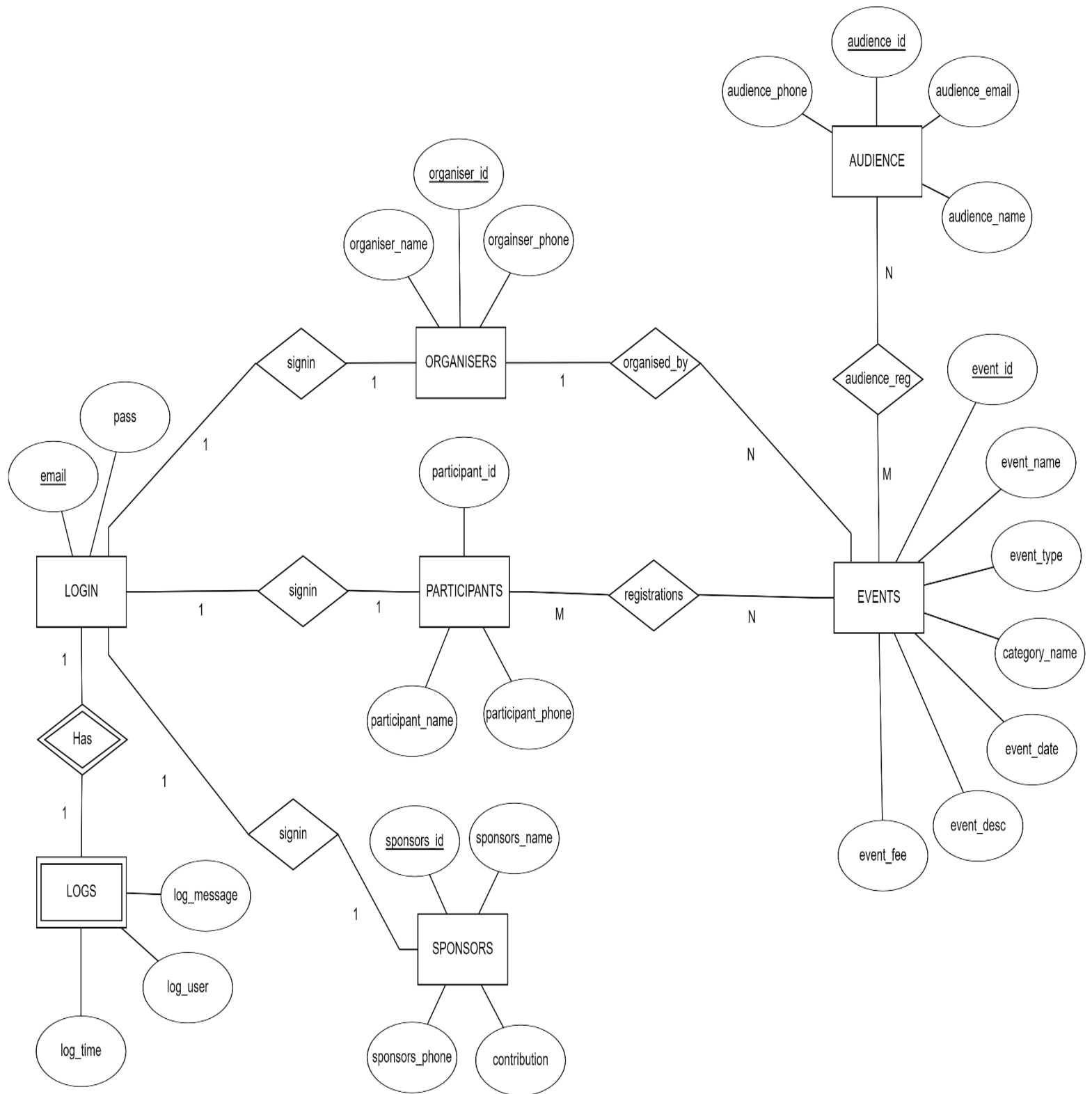
TOOLS USED

FRONT END : HTML, CSS, JAVASCRIPT

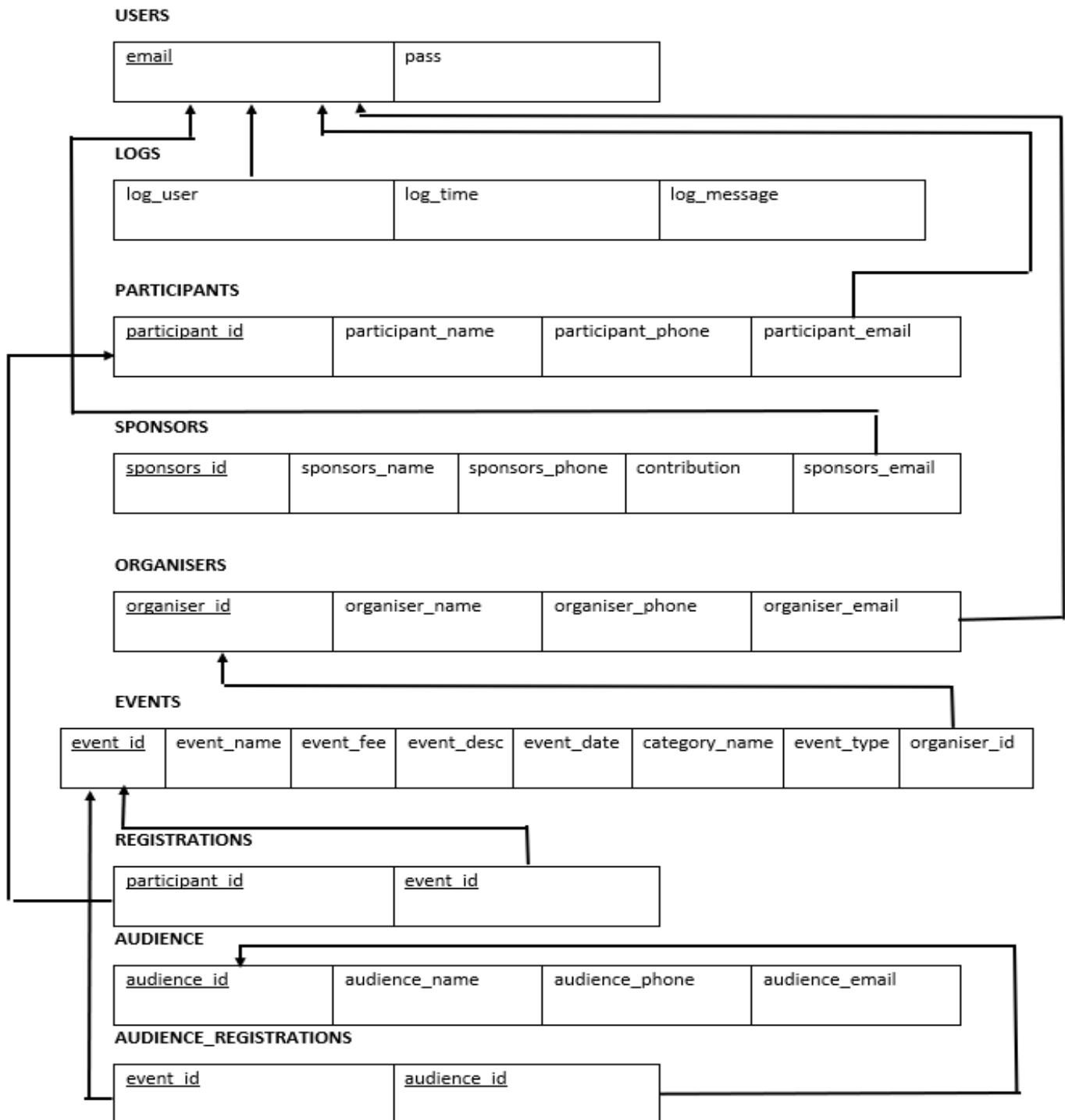
BACK END : PHP

DATABASE : MYSQL

ER DIAGRAM



RELATIONAL SCHEMA



DDL - BUILDING THE DATABASE

LOGIN

```
CREATE TABLE `login` (  
  `email` varchar(255) NOT NULL,  
  `pass` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
ALTER TABLE login  
  ADD PRIMARY KEY (`email`);
```

LOGS

```
CREATE TABLE `logs` (  
  `log_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `log_user` varchar(255) NOT NULL,  
  `log_message` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
ALTER TABLE `logs`  
  ADD KEY `log_user` (`log_user`);
```

```
ALTER TABLE `logs`  
  ADD CONSTRAINT `logs_ibfk_1` FOREIGN KEY (`log_user`) REFERENCES `login`  
  (`email`) ON DELETE RESTRICT;
```

ORGANISERS

```
CREATE TABLE `organisers` (  
  `organiser_id` int(11) NOT NULL,  
  `organiser_name` varchar(255) NOT NULL,  
  `organiser_email` varchar(255) NOT NULL,  
  `organiser_phone` varchar(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
ALTER TABLE `organisers`  
  ADD PRIMARY KEY (`organiser_id`),  
  ADD KEY `organiser_email` (`organiser_email`);
```

```
ALTER TABLE `organisers`  
  MODIFY `organiser_id` int(11) NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=67551;
```

```
ALTER TABLE `organisers`  
  ADD CONSTRAINT `on_organiser1` FOREIGN KEY (`organiser_email`)  
REFERENCES `login` (`email`);
```

EVENTS

```
CREATE TABLE `events` (  
  `event_id` int(11) NOT NULL,  
  `event_name` varchar(255) NOT NULL,  
  `event_type` varchar(255) NOT NULL,  
  `category_name` varchar(255) NOT NULL,  
  `event_date` date NOT NULL,  
  `event_fee` int(11) NOT NULL,  
  `event_desc` text NOT NULL,  
  `organiser_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
ALTER TABLE `events`  
  ADD PRIMARY KEY (`event_id`),  
  ADD KEY `organiser_id` (`organiser_id`);
```

```
ALTER TABLE `events`  
  MODIFY `event_id` int(11) NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=313817;
```

```
ALTER TABLE `events`  
  ADD CONSTRAINT `events_ibfk_2` FOREIGN KEY (`organiser_id`) REFERENCES  
`organisers` (`organiser_id`) ON DELETE RESTRICT;
```

PARTICIPANTS

```
CREATE TABLE `participants` (  
  `participant_id` int(11) NOT NULL,  
  `participant_name` varchar(255) NOT NULL,  
  `participant_email` varchar(255) NOT NULL,  
  `participant_phone` varchar(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
ALTER TABLE `participants`  
  ADD PRIMARY KEY (`participant_id`),  
  ADD KEY `participant_email` (`participant_email`);
```

```
ALTER TABLE `participants`
```

```

    MODIFY `participant_id` int(11) NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=213651;

ALTER TABLE `participants`
    ADD CONSTRAINT `on_participant1` FOREIGN KEY (`participant_email`)
    REFERENCES `login` (`email`);

```

AUDIENCE

```

CREATE TABLE `audience` (
    `audience_id` int(11) NOT NULL,
    `audience_name` varchar(255) NOT NULL,
    `audience_phone` varchar(10) NOT NULL,
    `audience_email` varchar(255) NOT NULL
);

ALTER TABLE `audience`
    ADD PRIMARY KEY (`audience_id`);

ALTER TABLE `audience`
    MODIFY `audience_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=99;

```

REGISTRATIONS

```

CREATE TABLE `registrations` (
    `participant_id` int(11) NOT NULL,
    `event_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

ALTER TABLE `registrations`
    ADD PRIMARY KEY (`participant_id`,`event_id`),
    ADD KEY `participant_id` (`participant_id`),
    ADD KEY `event_id` (`event_id`);

ALTER TABLE `registrations`
    ADD CONSTRAINT `registrations_ibfk_1` FOREIGN KEY (`participant_id`)
    REFERENCES `participants` (`participant_id`) ON DELETE CASCADE,
    ADD CONSTRAINT `registrations_ibfk_2` FOREIGN KEY (`event_id`)
    REFERENCES `events` (`event_id`) ON DELETE RESTRICT;
COMMIT;

```

AUDIENCE_REGISTRATIONS

```

CREATE TABLE `audience_registrations` (

```

```

    `audience_id` int(11) NOT NULL,
    `event_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

ALTER TABLE `audience_registrations`
  ADD PRIMARY KEY (`audience_id`,`event_id`),
  ADD KEY `audience_id` (`audience_id`),
  ADD KEY `event_id` (`event_id`);

ALTER TABLE `audience_registrations`
  ADD CONSTRAINT `audience_registrations_ibfk_1` FOREIGN KEY
(`audience_id`) REFERENCES `audience` (`audience_id`) ON DELETE CASCADE,
  ADD CONSTRAINT `audience_registrations_ibfk_2` FOREIGN KEY (`event_id`)
REFERENCES `events` (`event_id`) ON DELETE RESTRICT;
COMMIT;

```

SPONSORS

```

CREATE TABLE `sponsors` (
  `sponsors_id` int(11) NOT NULL,
  `sponsors_name` varchar(255) NOT NULL,
  `sponsors_phone` varchar(10) NOT NULL,
  `sponsors_email` varchar(255) NOT NULL,
  `contribution` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

ALTER TABLE `sponsors`
  ADD PRIMARY KEY (`sponsors_id`),
  ADD KEY `sponsors_email` (`sponsors_email`);

ALTER TABLE `sponsors`
  MODIFY `sponsors_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

ALTER TABLE `sponsors`
  ADD CONSTRAINT `on_sponsors1` FOREIGN KEY (`sponsors_email`) REFERENCES
`login` (`email`);

```

POPULATING THE DATABASE

```

INSERT INTO `login` (`email`,`pass`) VALUES
('chaitrabhat084@gmail.com',123);
INSERT INTO `login` (`email`,`pass`) VALUES
('chaitrabhat234@gmail.com',234);

```



```

INSERT INTO `login`(`email`,`pass`) VALUES ('david@gmail.com',123);
INSERT INTO `login`(`email`,`pass`) VALUES ('stokes@gmail.com',123);
INSERT INTO `login`(`email`,`pass`) VALUES ('ben4@gmail.com',123);
INSERT INTO `login`(`email`,`pass`) VALUES ('user1@gmail.com',123);
INSERT INTO `login`(`email`,`pass`) VALUES ('user2@gmail.com',123);
INSERT INTO `login`(`email`,`pass`) VALUES ('user3@gmail.com',123);
INSERT INTO `login`(`email`,`pass`) VALUES ('user4@gmail.com',123);

```

```

INSERT INTO `organisers` (`organiser_id`, `organiser_name`,
`organiser_email`,`organiser_phone`) VALUES
(67547, 'User1','user1@gmail.com', '8147497087');
INSERT INTO `organisers` (`organiser_id`, `organiser_name`,
`organiser_email`,`organiser_phone`) VALUES
(67548, 'User2','user2@gmail.com', '9845939755');
INSERT INTO `organisers` (`organiser_id`, `organiser_name`,
`organiser_email`,`organiser_phone`) VALUES
(67549, 'User3','user3@gmail.com', '9845939756');
INSERT INTO `organisers` (`organiser_id`, `organiser_name`,
`organiser_email`,`organiser_phone`) VALUES
(67550, 'User4','user4@gmail.com', '9845939757');

```

```

INSERT INTO `events` (`event_id`, `event_name`, `event_type`,
`category_name`, `event_date`, `event_fee`, `event_desc`, `organiser_id`)
VALUES
(313813, 'Coding & Debugging', 'Individual', 'Technical', '2022-12-18',
100, 'Can you beat the time? Most importantly, the errors?', 67549);
INSERT INTO `events` (`event_id`, `event_name`, `event_type`,
`category_name`, `event_date`, `event_fee`, `event_desc`, `organiser_id`)
VALUES
(313814, 'Dance', 'Individual', 'Cultural', '2022-12-20', 200, 'Lace Your
Dancing Shoes!', 67548);
INSERT INTO `events` (`event_id`, `event_name`, `event_type`,
`category_name`, `event_date`, `event_fee`, `event_desc`, `organiser_id`)
VALUES
(313815, 'Cricket', 'Group', 'Cultural', '2022-12-20', 1000, 'Either hit a
wicket or give a free hit. The excitement is equal for both. The hearts
will beat for both.', 67550);
INSERT INTO `events` (`event_id`, `event_name`, `event_type`,
`category_name`, `event_date`, `event_fee`, `event_desc`, `organiser_id`)
VALUES
(313816, 'Singing', 'Individual', 'Cultural', '2022-12-20', 200, 'Good
Music! Good Vibes!', 67549);

```

```

INSERT INTO `participants` (`participant_id`, `participant_name`,
`participant_email`, `participant_phone`) VALUES
(213648, 'David', 'david@gmail.com', '9787747958');
INSERT INTO `participants` (`participant_id`, `participant_name`,
`participant_email`, `participant_phone`) VALUES
(213649, 'Chaitra Bhat', 'chaitrabhat084@gmail.com', '8147497087');
INSERT INTO `participants` (`participant_id`, `participant_name`,
`participant_email`, `participant_phone`) VALUES
(213650, 'Stokes', 'stokes@gmail.com', '7624845977');

INSERT INTO
`audience`(`audience_id`,`audience_name`,`audience_phone`,`audience_email`
) VALUES
(96,'Ben','9867778767','ben4@gmail.com');
INSERT INTO
`audience`(`audience_id`,`audience_name`,`audience_phone`,`audience_email`
) VALUES
(97,'Chaitra','9845939755','chaitrabhat234@gmail.com');
INSERT INTO
`audience`(`audience_id`,`audience_name`,`audience_phone`,`audience_email`
) VALUES
(98,'Aura','9845939759','aura@gmail.com');

INSERT INTO `registrations` (`participant_id`, `event_id`) VALUES
(213649, 313814);
INSERT INTO `registrations` (`participant_id`, `event_id`) VALUES
(213648, 313813);
INSERT INTO `registrations` (`participant_id`, `event_id`) VALUES
(213648, 313814);
INSERT INTO `registrations` (`participant_id`, `event_id`) VALUES
(213650, 313815);
INSERT INTO `registrations` (`participant_id`, `event_id`) VALUES
(213650, 313816);

INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(96,313813);
INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(98,313813);
INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(97,313814);
INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(98,313815);
INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(97,313815);

```

```
INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(97,313816);
INSERT INTO `audience_registrations` (`audience_id`, `event_id`) VALUES
(98,313816);
```

```
INSERT INTO `sponsors` (`sponsors_id`,`sponsors_name`,
`sponsors_phone`,`sponsors_email`, `contribution`) VALUES
(1,'Chaitra Bhat', '8147497087', 'chaitrabhat084@gmail.com',100);
INSERT INTO `sponsors` (`sponsors_id`,`sponsors_name`,
`sponsors_phone`,`sponsors_email`, `contribution`) VALUES
(2,'Chaitra', '9845939755', 'chaitrabhat234@gmail.com',1000);
INSERT INTO `sponsors` (`sponsors_id`,`sponsors_name`,
`sponsors_phone`,`sponsors_email`, `contribution`) VALUES
(3,'Ben', '8988878997', 'ben4@gmail.com',50);
```

JOIN QUERIES

-- 1. JOIN QUERIES

-- a) EQUI JOIN - Display number of the audience for each event in the fest.

```
SELECT event_name, COUNT(audience_id) FROM audience_registrations as a JOIN events as e where a.event_id = e.event_id GROUP BY event_name;
```

```
MariaDB [festmanagement]> SELECT event_name, COUNT(audience_id) FROM audience_registrations as a JOIN events as e where
a.event_id = e.event_id GROUP BY event_name;
```

event_name	COUNT(audience_id)
Coding & Debugging	2
Cricket	2
Dance	1
Singing	2

```
4 rows in set (0.003 sec)
```

-- b) NATURAL JOIN - Display organiser name, event fee and event name for all the events.

```
select organiser_name, event_name, event_fee FROM organisers NATURAL JOIN events;
```

```
MariaDB [festmanagement]> select organiser_name, event_name, event_fee FROM organisers NATURAL JOIN events;
```

organiser_name	event_name	event_fee
User3	Coding & Debugging	100
User2	Dance	200
User4	Cricket	1000
User3	Singing	200

```
4 rows in set (0.001 sec)
```

-- c) JOIN - Display the name of the audience and the events they would attend.

select audience_name, event_name from audience_registrations as ar JOIN audience as a JOIN events as e where ar.audience_id = a.audience_id and ar.event_id = e.event_id;

```
MariaDB [festmanagement]> select audience_name, event_name from audience_registrations as ar JOIN audience as a JOIN events as e where ar.audience_id = a.audience_id and ar.event_id = e.event_id;
```

audience_name	event_name
Ben	Coding & Debugging
Aura	Coding & Debugging
Chaitra	Dance
Chaitra	Cricket
Aura	Cricket
Chaitra	Singing
Aura	Singing

```
7 rows in set (0.001 sec)
```

CO-RELATED QUERIES

-- 8. CO-RELATED QUERIES

-- a) NOT EXISTS - Display the participants who have not sponsored any event.

select * from participants where not exists(select * from sponsors where participants.participant_email = sponsors.sponsors_email);

```
MariaDB [festmanagement]> select * from participants where not exists(select * from sponsors where participants.participant_email = sponsors.sponsors_email);
```

participant_id	participant_name	participant_email	participant_phone
213648	David	david@gmail.com	9787747958
213650	Stokes	stokes@gmail.com	7624845977

```
2 rows in set (0.002 sec)
```

-- b) EXISTS - Display the participants who has sponsored any one event.

select * from participants where exists(select * from sponsors where participants.participant_email = sponsors.sponsors_email);

```
MariaDB [festmanagement]> select * from participants where exists(select * from sponsors where participants.participant_email = sponsors.sponsors_email);
```

participant_id	participant_name	participant_email	participant_phone
213649	Chaitra Bhat	chaitrabhat084@gmail.com	8147497087

1 row in set (0.001 sec)

-- c) NOT EXISTS - Display the sponsors who have not participated in any event.

select * from sponsors where not exists(select * from participants where participants.participant_email = sponsors.sponsors_email);

```
MariaDB [festmanagement]> select * from sponsors where not exists(select * from participants where participants.participant_email = sponsors.sponsors_email);
```

sponsors_id	sponsors_name	sponsors_phone	sponsors_email	contribution
2	Chaitra	9845939755	chaitrabhat234@gmail.com	1000
3	Ben	8988878997	ben4@gmail.com	50

2 rows in set (0.001 sec)

-- d) EXISTS - Display the sponsors who have participated in any one event.

select * from sponsors where exists(select * from participants where participants.participant_email = sponsors.sponsors_email);

```
MariaDB [festmanagement]> select * from sponsors where exists(select * from participants where participants.participant_email = sponsors.sponsors_email);
```

sponsors_id	sponsors_name	sponsors_phone	sponsors_email	contribution
1	Chaitra Bhat	8147497087	chaitrabhat084@gmail.com	100

1 row in set (0.001 sec)

NESTED QUERIES

-- 4. NESTED QUERIES

-- a) IN - Display ids and names of all the events which are organised by either user3 or user2.

SELECT event_id,event_name from events where organiser_id IN (SELECT organiser_id

from organisers where organiser_name = 'User3' or organiser_name = 'User2');

```
MariaDB [festmanagement]> SELECT event_id,event_name from events where organiser_id IN (SELECT organiser_id
-> from organisers where organiser_name = 'User3' or organiser_name = 'User2');
+-----+-----+
| event_id | event_name |
+-----+-----+
| 313814 | Dance |
| 313813 | Coding & Debugging |
| 313816 | Singing |
+-----+-----+
3 rows in set (0.001 sec)
```

-- b) NOT IN - Display ids and names of all the events which are not organised by either user3 or user2.

SELECT event_id,event_name from events where organiser_id NOT IN (SELECT
organiser_id from organisers where organiser_name = 'User3' or organiser_name =
'User2');

```
MariaDB [festmanagement]> SELECT event_id,event_name from events where organiser_id NOT IN (SELECT organiser_id from org
anisers where organiser_name = 'User3' or organiser_name = 'User2');
+-----+-----+
| event_id | event_name |
+-----+-----+
| 313815 | Cricket |
+-----+-----+
1 row in set (0.001 sec)
```

AGGREGATE FUNCTIONS

-- 2.AGGREGATE FUNCTION

-- a) SUM - Display the total contributions from all the sponsors.

SELECT SUM(contribution) as contribution FROM sponsors;

```
MariaDB [festmanagement]> SELECT SUM(contribution) as contribution FROM sponsors;
+-----+
| contribution |
+-----+
| 1150 |
+-----+
1 row in set (0.001 sec)
```

-- b) COUNT - Display the number of audiences who registered for various events along with event id.

SELECT event_id, COUNT(audience_id) FROM audience_registrations GROUP BY event_id;

```
MariaDB [festmanagement]> SELECT event_id, COUNT(audience_id) FROM audience_registrations GROUP BY event_id;
+-----+-----+
| event_id | COUNT(audience_id) |
+-----+-----+
| 313813   | 2                   |
| 313814   | 1                   |
| 313815   | 2                   |
| 313816   | 2                   |
+-----+-----+
4 rows in set (0.001 sec)
```

-- c) **COUNT** - Display the number of participants who registered for various events along with event id.

SELECT event_id,COUNT(*) AS count FROM registrations GROUP BY event_id ORDER BY COUNT(participant_id);

```
MariaDB [festmanagement]> SELECT event_id,COUNT(*) AS count FROM registrations GROUP BY event_id ORDER BY COUNT(participant_id);
+-----+-----+
| event_id | count |
+-----+-----+
| 313813   | 1     |
| 313815   | 1     |
| 313816   | 1     |
| 313814   | 2     |
+-----+-----+
4 rows in set (0.006 sec)
```

-- d) **MAX** - Display name, email and the contribution of the sponsor who contributed the highest.

SELECT sponsors_name, sponsors_email, MAX(contribution) from sponsors group by sponsors_email DESC LIMIT 1;

```
MariaDB [festmanagement]> SELECT sponsors_name, sponsors_email, MAX(contribution) from sponsors group by sponsors_email DESC LIMIT 1;
+-----+-----+-----+
| sponsors_name | sponsors_email      | MAX(contribution) |
+-----+-----+-----+
| Chaitra      | chaitrabhat234@gmail.com | 1000              |
+-----+-----+-----+
1 row in set (0.001 sec)
```

SET OPERATIONS

-- 3.SET OPERATION

-- c) UNION - Display name, phone number and email id all the users who are participant or audience.

SELECT participant_name as name,participant_phone as phone,participant_email as email from participants UNION SELECT audience_name, audience_phone,audience_email from audience;

```
MariaDB [festmanagement]> SELECT participant_name as name,participant_phone as phone,participant_email as email from participants UNION SELECT audience_name, audience_phone,audience_email from audience;
```

name	phone	email
David	9787747958	david@gmail.com
Chaitra Bhat	8147497087	chaitrabhat084@gmail.com
Stokes	7624845977	stokes@gmail.com
Ben	9867778767	ben4@gmail.com
Chaitra	9845939755	chaitrabhat234@gmail.com
Aura	9845939759	aura@gmail.com

6 rows in set (0.001 sec)

-- b) INTERSECT - Display the name and phone number of the user who is participant as well as the sponsor.

SELECT participant_name as name,participant_phone as phone from participants INTERSECT SELECT sponsors_name, sponsors_phone from sponsors;

```
MariaDB [festmanagement]> SELECT participant_name as name,participant_phone as phone from participants INTERSECT SELECT sponsors_name, sponsors_phone from sponsors;
```

name	phone
Chaitra Bhat	8147497087

1 row in set (0.001 sec)

-- c) EXCEPT - Display the name and phone number of the user who are audiences but not sponsors.

SELECT audience_name as name, audience_phone as phone,audience_email as email from audience EXCEPT SELECT sponsors_name, sponsors_phone,sponsors_email from sponsors;

```
MariaDB [festmanagement]> SELECT audience_name as name, audience_phone as phone,audience_email as email from audience EXCEPT SELECT sponsors_name, sponsors_phone,sponsors_email from sponsors;
```

name	phone	email
Ben	9867778767	ben4@gmail.com
Aura	9845939759	aura@gmail.com

2 rows in set (0.001 sec)

VIEW

-- 7. VIEWS

-- Create a view representing event ids and respective number of audience in each.

```
create view stat2 as select e.event_id, event_name, count(audience_id) as audience from events as e, audience_registrations as ar where ar.event_id = e.event_id group by e.event_id;
```

```
MariaDB [festmanagement]> create view stat2 as select e.event_id, event_name, count(audience_id) as audience from events as e, audience_registrations as ar where ar.event_id = e.event_id group by e.event_id;
Query OK, 0 rows affected (0.009 sec)
```

```
select * from stat2;
```

```
MariaDB [festmanagement]> select * from stat2;
+-----+-----+-----+
| event_id | event_name      | audience |
+-----+-----+-----+
| 313813   | Coding & Debugging | 2        |
| 313814   | Dance           | 1        |
| 313815   | Cricket         | 2        |
| 313816   | Singing         | 2        |
+-----+-----+-----+
4 rows in set (0.002 sec)
```

-- Create a view representing event ids and respective number of participants in each.

```
create view stat1 as select e.event_id, event_name, count(participant_id) as participant from events as e, registrations as r where r.event_id = e.event_id group by e.event_id;
```

```
MariaDB [festmanagement]> create view stat1 as select e.event_id, event_name, count(participant_id) as participant from events as e, registrations as r where r.event_id = e.event_id group by e.event_id;
Query OK, 0 rows affected (0.007 sec)
```

```
select * from stat1;
```

```
MariaDB [festmanagement]> select * from stat1;
```

event_id	event_name	participant
313813	Coding & Debugging	1
313814	Dance	2
313815	Cricket	1
313816	Singing	1

```
4 rows in set (0.002 sec)
```

-- Create a view representing event ids and respective number of audience and number of participants in each i.e summary of both the above views

create view stats as select stat1.event_id,stat1.event_name,participant,audience from stat1 INNER JOIN stat2 where stat1.event_id = stat2.event_id;

```
MariaDB [festmanagement]> create view stats as select stat1.event_id,stat1.event_name,participant,audience from stat1 INNER JOIN stat2 where stat1.event_id = stat2.event_id;
Query OK, 0 rows affected (0.009 sec)
```

select * from stats;

```
MariaDB [festmanagement]> select * from stats;
```

event_id	event_name	participant	audience
313813	Coding & Debugging	1	2
313814	Dance	2	1
313815	Cricket	1	2
313816	Singing	1	2

```
4 rows in set (0.003 sec)
```

FUNCTION

-- 5. FUNCTIONS

-- Function to display the remaining seats if available or display that seats are full for each event.

```

DELIMITER $$
CREATE FUNCTION Seats_Available(e_id varchar(50))
RETURNS varchar(100)
DETERMINISTIC
BEGIN
DECLARE N INT;
DECLARE E VARCHAR(50);
DECLARE MSG varchar(1000);
select count(*) from audience_registrations where audience_registrations.event_id = e_id
into N;
select event_name from events where events.event_id = e_id into E;
if N>=2 then
set MSG:=concat("SEATS ARE FULL FOR ", E," EVENT!!!");
else
set MSG:=concat("YOU HAVE ",2-N," SEATS LEFT FOR ",E," EVENT!!!");
end if;
RETURN MSG;
END $$
DELIMITER ;

```

```

MariaDB [festmanagement]> DELIMITER $$
MariaDB [festmanagement]> CREATE FUNCTION Seats_Available(e_id varchar(50))
  -> RETURNS varchar(100)
  -> DETERMINISTIC
  -> BEGIN
  -> DECLARE N INT;
  -> DECLARE E VARCHAR(50);
  -> DECLARE MSG varchar(1000);
  -> select count(*) from audience_registrations where audience_registrations.event_id = e_id into N;
  -> select event_name from events where events.event_id = e_id into E;
  -> if N>=2 then
  -> set MSG:=concat("SEATS ARE FULL FOR ", E," EVENT!!!");
  -> else
  -> set MSG:=concat("YOU HAVE ",2-N," SEATS LEFT FOR ",E," EVENT!!!");
  -> end if;
  -> RETURN MSG;
  -> END $$
Query OK, 0 rows affected, 2 warnings (0.010 sec)

MariaDB [festmanagement]> DELIMITER ;
MariaDB [festmanagement]>

```

```

select seats_available(event_id) from events;

```

```

MariaDB [festmanagement]> select seats_available(event_id) from events;
+-----+
| seats_available(event_id) |
+-----+
| YOU HAVE 1 SEATS LEFT FOR Dance EVENT!!! |
| SEATS ARE FULL FOR Coding & Debugging EVENT!!! |
| SEATS ARE FULL FOR Singing EVENT!!! |
| SEATS ARE FULL FOR Cricket EVENT!!! |
+-----+
4 rows in set (0.003 sec)

```

PROCEDURE

-- Create a Procedure which displays the sponsors who has contributed atleast 100

DELIMITER &&

CREATE PROCEDURE sponsors_who_contributed_atleast_100()

BEGIN

SELECT * FROM sponsors WHERE contribution >= 100;

END &&

DELIMITER ;

```

MariaDB [festmanagement]> DELIMITER &&
MariaDB [festmanagement]> CREATE PROCEDURE sponsors_who_contributed_atleast_100()
-> BEGIN
-> SELECT * FROM sponsors WHERE contribution >= 100;
-> END &&
Query OK, 0 rows affected (0.008 sec)

MariaDB [festmanagement]> DELIMITER ;
MariaDB [festmanagement]>

```

CALL sponsors_who_contributed_atleast_100();

```
MariaDB [festmanagement]> CALL sponsors_who_contributed_atleast_100();
```

sponsors_id	sponsors_name	sponsors_phone	sponsors_email	contribution
1	Chaitra Bhat	8147497087	chaitrabhat084@gmail.com	100
2	Chaitra	9845939755	chaitrabhat234@gmail.com	1000

```
2 rows in set (0.001 sec)

Query OK, 0 rows affected (0.024 sec)
```

TRIGGER

-- Trigger to check how many more participants can register for an event.

```
DELIMITER $$
CREATE TRIGGER Seats_Left
BEFORE INSERT
ON registrations
FOR EACH ROW
BEGIN
declare ERROR_MSG varchar(100);
declare val varchar(100);
set ERROR_MSG =("Maximum Limit for Number of Participants Reached! Sorry Next
Time");
if(select count(*) from registrations where registrations.event_id = new.event_id) >= 2
then signal sqlstate '45000'
set message_text = ERROR_MSG;
end if;
END $$
DELIMITER ;
```

```

MariaDB [festmanagement]> DELIMITER $$
MariaDB [festmanagement]> CREATE TRIGGER Seats_Left
-> BEFORE INSERT
-> ON registrations
-> FOR EACH ROW
-> BEGIN
-> declare ERROR_MSG varchar(100);
-> declare val varchar(100);
-> set ERROR_MSG =("Maximum Limit for Number of Participants Reached! Sorry Next Time");
-> if(select count(*) from registrations where registrations.event_id = new.event_id) >= 2
-> then signal sqlstate '45000'
-> set message_text = ERROR_MSG;
-> end if;
-> END $$
Query OK, 0 rows affected (0.010 sec)

MariaDB [festmanagement]> DELIMITER ;
MariaDB [festmanagement]>

```

Insert into registrations values(213648,313816);

```

MariaDB [festmanagement]> Insert into registrations values(213648,313816);
Query OK, 1 row affected (0.007 sec)

```

Insert into registrations values(213650,313814);

```

MariaDB [festmanagement]> Insert into registrations values(213650,313816);
ERROR 1644 (45000): Maximum Limit for Number of Participants Reached! Sorry Next Time

```