ONLINE EVENT MANAGEMENT:

GROUP -1

AARDRA.R. NAIR: CB.SC.I5DAS19001
AISHWARYA.A: CB.SC.I5DAS19005
NIVEDHA.A: CB.SC.I5DAS19023
DHANUSHKA SHREE: CB.SC.I5DAS19036

TABLE OF CONTENTS

1. INTRODUCTION:

The online event management database we created is for the ease of handling for the customers to register their account and book multiple events with a variety of discounts and special preferences. And from an admin point of view, it is easy for them to manage all the events and its details. For customers' ease, each event type has their own special events, where each event has their own special preferences and discounts.

2. PROBLEM STATEMENT:

Every organization, whether big or small, faces issues in dealing with and controlling events, this database is created for an easy and accessible platform.

3. MODULES AND DESCRIPTION:

Entities:

- User:
Customers will have to register themselves with their email id so, their name, email id
password will be inserted in the table and they will be given a unique user id to book
events. Customers won't be able to book events without registering themselves.

- Admin:
Admin has the privilege of managing the event table, discount table, user table, special preference table and booking table and is allowed to update, delete and insert values. Admin looking over

- Event_details:
Details like event name, event type, date/time, duration, e amount are columns in this table and also event name and event type are their own ids. Each event has their own event type.

- Even_discount:
In this table the discounts are available for the users, and it gives a unique id called DID and the amount reduced for the discounts are also given under the column name as rate_reduced.

- Special_preference:
If the users need any kind of special preferences for the specific event they will be charged extra fees. Each special preference has their own id called SID.

- User booking:
Each event booked will have their own booking id called BID and also with their respective user id, with the number of seats booked. Also includes their Event ID, Discount ID, Special preference ID.

- Amount:
When a user books an event and pays for it, they will be given a paid id with mode of payment mentioned beside it.

HOME PAGE:
- Displays event categories, list of events under each category, cost of event

ADMIN:

Admin login:
- Admin allowed to login using name , email , password

Delete/update/add:
- Admin is allowed to delete, update and add events and its categories
- Changing discount rates and price (including price for special preferences for users)

USER:

User login:
- Users can login using their name , password and email

Booking:
- User can book an event by paying

Delete / Update:
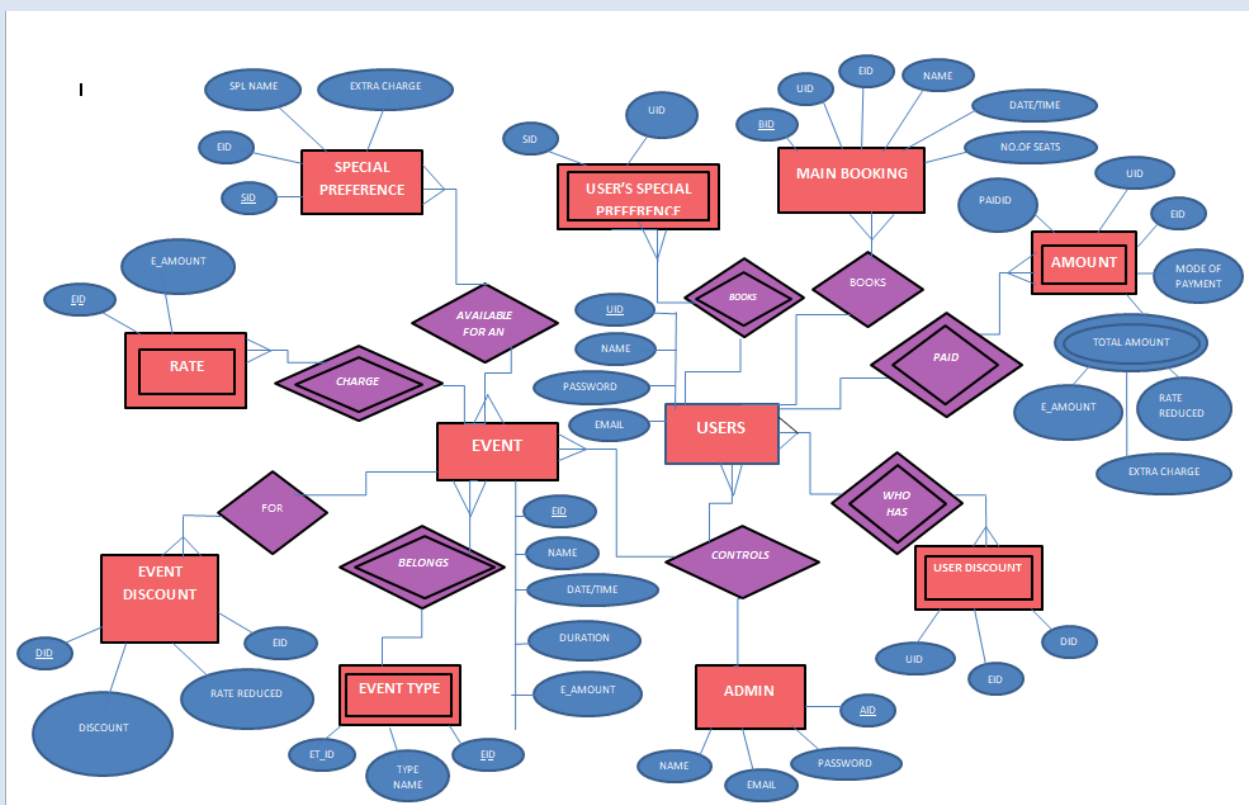- Users are allowed to delete or change their booking.

Discount / Special preference:
- User can apply for special preference and can get discount accordingly

Payment:
- User are allowed to pay their money through various online platforms and then a message pops up saying "you have successfully registered"

4. **ER- DIAGRAM:**

1 NF:

Main_table(AID,A_PASSWORD,A_NAME,A_EMAIL ,UID , USER NAME , PSWD , EMAIL , BID,ET_ID,EID,EVENT TYPE NAME , EVENT_NAME , DATE/TIME , NO.OF SEATS , DURATION,MODE OF PAYMENT , E_AMOUNT,EXTRA CHARGE,RATE REDUCED,DID,DISCOUNT NAME , SID , SPL PREF NAME , TOTAL AMOUNT , PAID ID)

2 NF:

Userbooking(<u>BID</u>,UID,EID,DID,SID,SEATS)

Paymentuser(<u>PAIDID</u>,BID,MODE OF PAYEMENT)

Event_and_EventType(<u>EID</u>,EVENTNAME,DATE/TIME,DURATION,ET_ID, EVENT TYPE NAME,E_AMOUNT)

Event_discount(<u>DID</u>,DISCOUNTNAME, RATE REDUCED)

User(<u>UID</u>,USERNAME,PSWD,EMAIL)

Special_Preference(<u>SID</u>,EXTRA CHARGE,SPL PREF NAME)

ADMIN(AID,A_NAME,A_PASSWORD,A_EMAIL)

3 NF: <u>Schema formed from ER diagram</u>

User(<u>UID</u>,EMAIL,PSWD,USERNAME)

Event_and_EventType(<u>EID</u>,E_AMOUNT,EVENT TYPE NAME,ET_ID,DURATION,DATE/TIME,EVENT NAME)

Event_discount(<u>DID</u>,RATE REDUCED,DISCOUNT NAME)

Special_Preference(<u>SID</u>,EXTRA CHARGE , SPL PREF NAME)

Userbooking(<u>BID</u>,UID,EID,DID,SID,SEATS)

Amount(<u>PAIDID</u> ,BID , MODE OF PAYMENT)

ADMIN(AID,A_NAME,A_PASSWORD,A_EMAIL)

1. CREATING TABLES

- ➤ CREATE TABLE `sys`.`user` (
    `EMAIL` VARCHAR(45) NOT NULL,
    `UID` VARCHAR(45) NOT NULL,
    `PASSWORD` VARCHAR(45) NOT NULL,
    `USERNAME` VARCHAR(45) NULL,
    PRIMARY KEY (`UID`));

- ➤ INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('Kim2002@gmail.com', 'U01', 'Albat123', 'Kim');
    INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('Aliabatt234@gmail.com', 'U02', 'tannie624', 'Alia');
    INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('ishutink278@gmail.com', 'U03', 'TinkuB72', 'Ishu');
    INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('riyaemma@gmail.com', 'U04', '75riya', 'Emma');
    INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('parkprince@gmail.com', 'U05', '07princy', 'Park');
    INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('bangriyam@gmail.com', 'U06', 'Bang672', 'Ryan');
    INSERT INTO `sys`.`user` (`EMAIL`, `UID`, `PASSWORD`, `USERNAME`) VALUES ('Krish98@gmail.com', 'U07', '957krishy', 'Krishna');

UID	USERNAME	EMAIL	▼	PASSWORD
U01	Kim	Kim2002@gmail.com		Albat123
U02	Alia	Aliabatt234@gmail.com		tannie624
U03	Ishu	ishutink278@gmail.com		TinkuB72
U04	Emma	riyaemma@gmail.com		75riya
U05	Park	parkprince@gmail.com		07princy
U06	Ryan	bangriyam@gmail.com		Bang672
U07	Krishna	Krish98@gmail.com		957krishy

- ➤ SELECT * FROM sys.admin;CREATE TABLE `event_details` (
    `EID` varchar(45) NOT NULL,
    `EVENT_NAME` varchar(45) NOT NULL,
    `ET_ID` varchar(45) NOT NULL,
    `EVENT_TYPE` varchar(45) DEFAULT NULL,
    `DATE_TIME` datetime NOT NULL,

```
        `DURATION` varchar(45) NOT NULL,
        `E_AMOUNT` int NOT NULL,
        PRIMARY KEY (`EID`));
```

- INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0101', 'CLASSICAL', 'ET01', 'DANCE', '2021-08-29 02:30:00', '2:00:00', '2000');
  INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0102', 'HIP HOP', 'ET01', 'DANCE', '2021-09-01 11:30:00', '2:00:00', '1000');
  INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0103', 'FOLK', 'ET01', 'DANCE', '2021-09-30 10:00:00', '2:00:00', '3000');
  INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0301', 'CARNATIC', 'ET03', 'MUSIC', '2021-10-01 09:30:00', '2:30:00', '1000');
  INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0302', 'HINDUSTANI', 'ET03', 'MUSIC', '2021-11-12 17:00:00', '1:30:00', '900');
  INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0201', 'PAINTINGS', 'ET02', 'ARTS', '2021-11-21 16:00:00', '1:00:00', '1500');
  INSERT INTO `sys`.`event_details` (`EID`, `EVENT_NAME`, `ET_ID`, `EVENT_TYPE`, `DATE_TIME`, `DURATION`, `E_AMOUNT`) VALUES ('E0401', 'TENNIS', 'ET04', 'SPORTS', '2021-10-22 10:30:00', '3:00:00', '2000');

EID	EVENT_NAME	ET_ID	EVENT_TYPE	DATE_TIME	DURATION	E_AMOUNT
E0101	CLASSICAL	ET01	DANCE	2021-08-29 02:30:00	2:00:00	2000
E0102	HIP HOP	ET01	DANCE	2021-09-01 11:30:00	2:00:00	1000
E0103	FOLK	ET01	DANCE	2021-09-30 10:00:00	2:00:00	3000
E0201	PAINTINGS	ET02	ARTS	2021-11-21 16:00:00	1:00:00	1500
E0301	CARNATIC	ET03	MUSIC	2021-10-01 09:30:00	2:30:00	1000
E0302	HINDUSTANI	ET03	MUSIC	2021-11-12 17:00:00	1:30:00	900
E0401	TENNIS	ET04	SPORTS	2021-10-22 10:30:00	3:00:00	2000

- CREATE TABLE `sys`.`event_discount` (
  `DID` VARCHAR(45) NOT NULL,
  `DISCOUNT_NAME` VARCHAR(45) NOT NULL,
  `RATE_REDUCED` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`DID`));

- INSERT INTO `sys`.`event_discount` (`DID`, `DISCOUNT_NAME`, `RATE_REDUCED`)
  VALUES ('D00', 'NO DISCOUNT APPLIED', '0');
  INSERT INTO `sys`.`event_discount` (`DID`, `DISCOUNT_NAME`, `RATE_REDUCED`)
  VALUES ('D01', 'PREMIUM ACCOUNT', '1000');
  INSERT INTO `sys`.`event_discount` (`DID`, `DISCOUNT_NAME`, `RATE_REDUCED`)
  VALUES ('D02', '10 ADULTS', '500');
  INSERT INTO `sys`.`event_discount` (`DID`, `DISCOUNT_NAME`, `RATE_REDUCED`)
  VALUES ('D03', 'COUPON', '700');
  INSERT INTO `sys`.`event_discount` (`DID`, `DISCOUNT_NAME`, `RATE_REDUCED`)
  VALUES ('D04', 'TOTAL AMOUNT > 15000', '2000');

DID	DISCOUNT_NAME	RATE_REDUCED
D00	NO DISCOUNT APPLIED	0
D01	PREMIUM ACCOUNT	1000
D02	10 ADULTS	500
D03	COUPON	700
D04	TOTAL AMOUNT > 15000	2000

SPECIAL_PREFERENCE TABLE:

- CREATE TABLE `sys`.`special_preference` (
   `SID` VARCHAR(45) NOT NULL,
   `SPL_PREFERENCE_NAME` VARCHAR(45) NOT NULL,
   `EXTRA CHARGE` VARCHAR(45) NOT NULL,
   PRIMARY KEY (`SID`));

SPECIAL PREFERENCE VALUES TABLE:

- INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`,
  `EXTRA_CHARGE`) VALUES ('S00', 'NO_SPL_APPLIED', '0');
  INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`, `EXTRA
  CHARGE`) VALUES ('S01', 'DANCE_VIP SEATS', '1000');
  INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`, `EXTRA
  CHARGE`) VALUES ('S02', 'DANCE_BACKSTAGE VISIT', '5000');
  INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`, `EXTRA
  CHARGE`) VALUES ('S03', 'MUSIC_VIP SEATS', '1500');
  INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`, `EXTRA
  CHARGE`) VALUES ('S04', 'MUSIC_BACKSTAGE VIST', '2500');
  INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`, `EXTRA
  CHARGE`) VALUES ('S05', 'ARTS_PAINTINGS ALONE TIME PER HOUR', '2500');

INSERT INTO `sys`.`special_preference` (`SID`, `SPL_PREFERENCE_NAME`, `EXTRA CHARGE`) VALUES ('S06', 'SPORTS_VIP SEATS', '3000');

SID	SPL_PREFERENCE_NAME	EXTRA_CHARGE
S00	NO_SPL_APPLIED	0
S01	DANCE_VIP SEATS	1000
S02	DANCE_BACKSTAGE VISIT	5000
S03	MUSIC_VIP SEATS	1500
S04	MUSIC_BACKSTAGE VIST	2500
S05	ARTS_PAINTINGS ALONE TIME PER HOUR	2500
S06	SPORTS_VIP SEATS	3000

USER_BOOKING TABLE

> CREATE TABLE `sys`.`user_booking` (
 `BID` VARCHAR(45) NOT NULL,
 `UID` VARCHAR(45) NOT NULL,
 `EID` VARCHAR(45) NOT NULL,
 `DID` VARCHAR(45) NULL,
 `SID` VARCHAR(45) NULL,
 `SEATS` INT NOT NULL,
 PRIMARY KEY (`BID`),
 INDEX `fk_uid_idx` (`UID` ASC) VISIBLE,
 INDEX `fk_eid_idx` (`EID` ASC) VISIBLE,
 INDEX `fk_did_idx` (`DID` ASC) VISIBLE,
 INDEX `fk_sid_idx` (`SID` ASC) VISIBLE,
 CONSTRAINT `fk_uid`
  FOREIGN KEY (`UID`)
  REFERENCES `sys`.`user` (`UID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
 CONSTRAINT `fk_eid`
  FOREIGN KEY (`EID`)
  REFERENCES `sys`.`event_details` (`EID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
 CONSTRAINT `fk_did`
  FOREIGN KEY (`DID`)
  REFERENCES `sys`.`event_discount` (`DID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
 CONSTRAINT `fk_sid`
  FOREIGN KEY (`SID`)
  REFERENCES `sys`.`special_preference` (`SID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);

- INSERT INTO `sys`.`user_booking` (`BID`, `UID`, `EID`, `DID`, `SID`, `SEATS`) VALUES ('B01', 'U01', 'E0101', 'D01', 'S00', '3');
  INSERT INTO `sys`.`user_booking` (`BID`, `UID`, `EID`, `DID`, `SID`, `SEATS`) VALUES ('B02', 'U07', 'E0103', 'D00', 'S01', '4');
  INSERT INTO `sys`.`user_booking` (`BID`, `UID`, `EID`, `DID`, `SID`, `SEATS`) VALUES ('B03', 'U03', 'E0302', 'D01', 'S03', '1');
  INSERT INTO `sys`.`user_booking` (`BID`, `UID`, `EID`, `DID`, `SID`, `SEATS`) VALUES ('B04', 'U01', 'E0301', 'D02', 'S00', '5');
  INSERT INTO `sys`.`user_booking` (`BID`, `UID`, `EID`, `DID`, `SID`, `SEATS`) VALUES ('B06', 'U02', 'E0201', 'D03', 'S00', '6');
  INSERT INTO `sys`.`user_booking` (`BID`, `UID`, `EID`, `DID`, `SID`, `SEATS`) VALUES ('B07', 'U04', 'E0101', 'D02', 'S02', '2');

BID	UID	EID	DID	SID	SEATS
B01	U01	E0101	D01	S00	3
B02	U07	E0103	D00	S01	4
B03	U03	E0302	D01	S03	1
B04	U01	E0301	D02	S00	5
B06	U02	E0201	D03	S00	6
B07	U04	E0101	D02	S02	2

- CREATE TABLE `sys`.`amount` (
`PAIDID` VARCHAR(45) NOT NULL,
`BID` VARCHAR(45) NOT NULL,
`MODE OF PAYMENT` VARCHAR(45) NOT NULL,
PRIMARY KEY (`PAIDID`),
INDEX `fk_bid_idx` (`BID` ASC) VISIBLE,
CONSTRAINT `fk_bid`
FOREIGN KEY (`BID`)
REFERENCES `sys`.`user_booking` (`BID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

- INSERT INTO `sys`.`amount` (`PAIDID`, `BID`, `MODE OF PAYMENT`) VALUES ('PID01', 'B01', 'CREDIT');
  INSERT INTO `sys`.`amount` (`PAIDID`, `BID`, `MODE OF PAYMENT`) VALUES ('PID02', 'B02', 'DEBIT');
  INSERT INTO `sys`.`amount` (`PAIDID`, `BID`, `MODE OF PAYMENT`) VALUES ('PID03', 'B03', 'CASH');
  INSERT INTO `sys`.`amount` (`PAIDID`, `BID`, `MODE OF PAYMENT`) VALUES ('PID04', 'B04', 'BANK TRANSFER');

INSERT INTO `sys`.`amount` (`PAIDID`, `BID`, `MODE OF PAYMENT`) VALUES ('PID05', 'B06', 'E WALLETS');
INSERT INTO `sys`.`amount` (`PAIDID`, `BID`, `MODE OF PAYMENT`) VALUES ('PID06', 'B07', 'CREDIT');

PAIDID	BID	MODE OF PAYMENT
PID01	B01	CREDIT
PID02	B02	DEBIT
PID03	B03	CASH
PID04	B04	BANK TRANSFER
PID05	B06	E WALLETS
PID06	B07	CREDIT

ADMIN TABLE

- CREATE TABLE `sys`.`admin` (
  `AID` VARCHAR(45) NOT NULL,
  `A_NAME` VARCHAR(45) NOT NULL,
  `A_PASSWORD` VARCHAR(45) NOT NULL,
  `A_EMAIL` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`AID`));

ADMIN VALUE TABLE

- INSERT INTO `sys`.`admin` (`AID`, `A_NAME`, `A_PASSWORD`, `A_EMAIL`) VALUES ('A01', 'AISHU', 'aishujk12', 'aishu123@gmail.com');
  INSERT INTO `sys`.`admin` (`AID`, `A_NAME`, `A_PASSWORD`, `A_EMAIL`) VALUES ('A02', 'AARDRA', 'aardra123', 'aardraramesh@gmail.com');
  INSERT INTO `sys`.`admin` (`AID`, `A_NAME`, `A_PASSWORD`, `A_EMAIL`) VALUES ('A03', 'DHANU', 'dhanubrabu', 'dhanushka23@gmail.com');
  INSERT INTO `sys`.`admin` (`AID`, `A_NAME`, `A_PASSWORD`, `A_EMAIL`) VALUES ('A04', 'NIVI', 'nivi2002', 'nivianbu@gmail.com');

AID	A_NAME	A_PASSWORD	A_EMAIL
A01	AISHU	aishujk12	aishu123@gmail.com
A02	AARDRA	aardra123	aardraramesh@gmail.com
A03	DHANU	dhanubrabu	dhanushka23@gmail.com
A04	NIVI	nivi2002	nivianbu@gmail.com

2. 5 SIMPLE QUERIES

➤ select * from sys.special_preference where SPL_PREFERENCE_NAME like 'DANCE%' ;

SID	SPL_PREFERENCE_NAME	EXTRA CHARGE
S01	DANCE_VIP SEATS	1000
S02	DANCE_BACKSTAGE VISIT	5000

➤ select * from sys.event_details order by E_AMOUNT;

EID	EVENT_NAME	ET_ID	EVENT_TYPE	DATE_TIME	DURATION	E_AMOUNT
E0302	HINDUSTANI	ET03	MUSIC	2021-11-12 17:00:00	1:30:00	900
E0102	HIP HOP	ET01	DANCE	2021-09-01 11:30:00	2:00:00	1000
E0301	CARNATIC	ET03	MUSIC	2021-10-01 09:30:00	2:30:00	1000
E0201	PAINTINGS	ET02	ARTS	2021-11-21 16:00:00	1:00:00	1500
E0101	CLASSICAL	ET01	DANCE	2021-08-29 02:30:00	2:00:00	2000
E0401	TENNIS	ET04	SPORTS	2021-10-22 10:30:00	3:00:00	2000
E0103	FOLK	ET01	DANCE	2021-09-30 10:00:00	2:00:00	3000

➤ SELECT event_name FROM sys.event_details WHERE e_amount between 900 and 2000 LIMIT 2;

event_name
CLASSICAL
HIP HOP

➤ select event_type from sys.event_details where ET_ID = 'ET03' LIMIT 1 ;

event_type
MUSIC

➤ select uid from sys.user_booking where seats > 2 ;

uid
U01
U07
U01
U02

12

3. 5 AGGREGATE QUERIES

- ➢ select avg(e_amount) from sys.event_details ;

avg(e_amount)
1628.5714

- ➢ select count(*) from sys.user ;

count(*)
7

- ➢ select max(EXTRA_CHARGE) from sys.special_preference ;

max(EXTRA_CHARGE)
5000

- ➢ SELECT COUNT(EID), ET_ID FROM sys.event_details GROUP BY ET_ID;

COUNT(EID)	ET_ID
3	ET01
1	ET02
2	ET03
1	ET04

- ➢ select sum(seats) from sys.user_booking where EID='E0101' OR 'S02';

sum(seats)
5

4. 5 JOIN QUERIES

➤ SELECT sys.user.UID, sys.user_booking.seats, sys.user_booking.BID
FROM sys.user
INNER JOIN sys.user_booking ON sys.user.UID=sys.user_booking.UID;

UID	seats	BID
U01	3	B01
U07	4	B02
U03	1	B03
U01	5	B04
U02	6	B06
U04	2	B07

➤ SELECT sys.user_booking.UID, sys.amount.PAIDID
FROM sys.user_booking
LEFT JOIN sys.amount ON sys.user_booking.BID = sys.amount.BID
ORDER BY sys.user_booking.UID ;

UID	PAIDID
U01	PID01
U01	PID04
U02	PID05
U03	PID03
U04	PID06
U07	PID02

➤ select sys.event_details.date_time , sys.user_booking.uid
from sys.event_details
inner join sys.user_booking
on sys.event_details.eid = sys.user_booking.eid

date_time	uid
2021-08-29 02:30:00	U01
2021-09-30 10:00:00	U07
2021-11-12 17:00:00	U03
2021-10-01 09:30:00	U01
2021-11-21 16:00:00	U02
2021-08-29 02:30:00	U04

➤ select sys.user_booking.uid , sys.amount.mode_of_payment

from sys.user_booking
right join sys.amount
on sys.user_booking.bid = sys.amount.bid

uid	mode_of_payment
U01	CREDIT
U07	DEBIT
U03	CASH
U01	BANK TRANSFER
U02	E WALLETS
U04	CREDIT

- select sys.user.username ,sys.user_booking.uid, sys.event_details.e_amount
  from sys.user_booking
  inner join sys.event_details
  on sys.user_booking.eid = sys.event_details.eid
  inner join sys.user
  on sys.user_booking.uid=sys.user.uid

username	uid	e_amount
Kim	U01	1000
Kim	U01	2000
Alia	U02	1500
Ishu	U03	900
Emma	U04	2000
Krishna	U07	3000

5. CREATE 5 VIEWS

VIEW 1 : To Display The Total Amount The User Has To Pay Including Their Extra Charge(From Special Preference) And Rate Reduced(From Discounts) And The Actual Event Amount – Multiplied With No.Of Seats Booked

- CREATE VIEW sys.total_amount_users as
  select sys.user_booking.uid, ((sys.event_details.e_amount + sys.special_____ -
  sys.event_discount.rate_reduced)*sys.user_booking.seats) as total
  from sys.user_booking
  inner join sys.event_details
  on sys.user_booking.eid=sys.event_details.eid
  inner join sys.special_preference
  on sys.user_booking.sid=sys.special_preference.sid
  inner join sys.event_discount
  on sys.user_booking.did=sys.event_discount.did

uid	total
U07	16000
U01	3000
U03	1400
U01	2500
U04	13000
U02	4800

VIEW 2 : To Know How Many Times A User Did Their Booking In This Website

- create view sys.no_of_times_user_booked as
  select sys.user.username, count(sys.user_booking.uid) as no_of_times_user_booked
  from sys.user_booking
  inner join sys.user
  on sys.user.uid=sys.user_booking.uid
  group by sys.user_booking.uid
  select * from sys.no_of_times_user_booked

username	no_of_times_user_booked
Kim	2
Alia	1
Ishu	1
Emma	1
Krishna	1

VIEW 3 : Users Who Have Registered But Not Booked An

15

> create view sys.user_not_booked as
> select sys.user.uid as users_no_bookings , sys.user.username
> from sys.user
> left join sys.user_booking
> on sys.user.uid = sys.user_booking.uid
> where sys.user_booking.uid IS NULL

users_no_bookings	username
U05	Park
U06	Ryan

> create view sys.cross_join as
> select sys.event_details.event_name , sys.event_details.e_amount , sys.event_discount.discount_name
> , sys.event_discount.rate_reduced
> from sys.event_details
> cross join sys.event_discount

event_name	e_amount	discount_name	rate_reduced
CLASSICAL	2000	TOTAL AMOUNT > 15000	2000
CLASSICAL	2000	COUPON	700
CLASSICAL	2000	10 ADULTS	500
CLASSICAL	2000	PREMIUM ACCOUNT	1000
CLASSICAL	2000	NO DISCOUNT APPLIED	0
HIP HOP	1000	TOTAL AMOUNT > 15000	2000
HIP HOP	1000	COUPON	700
HIP HOP	1000	10 ADULTS	500
HIP HOP	1000	PREMIUM ACCOUNT	1000
HIP HOP	1000	NO DISCOUNT APPLIED	0
FOLK	3000	TOTAL AMOUNT > 15000	2000
FOLK	3000	COUPON	700
FOLK	3000	10 ADULTS	500
FOLK	3000	PREMIUM ACCOUNT	1000
FOLK	3000	NO DISCOUNT APPLIED	0
PAINTINGS	1500	TOTAL AMOUNT > 15000	2000
PAINTINGS	1500	COUPON	700
PAINTINGS	1500	10 ADULTS	500
PAINTINGS	1500	PREMIUM ACCOUNT	1000
PAINTINGS	1500	NO DISCOUNT APPLIED	0
CARNATIC	1000	TOTAL AMOUNT > 15000	2000
CARNATIC	1000	COUPON	700
CARNATIC	1000	10 ADULTS	500
CARNATIC	1000	PREMIUM ACCOUNT	1000
CARNATIC	1000	NO DISCOUNT APPLIED	0
HINDUSTANI	900	TOTAL AMOUNT > 15000	2000
HINDUSTANI	900	COUPON	700
HINDUSTANI	900	10 ADULTS	500
HINDUSTANI	900	PREMIUM ACCOUNT	1000
HINDUSTANI	900	NO DISCOUNT APPLIED	0
TENNIS	2000	TOTAL AMOUNT > 15000	2000
TENNIS	2000	COUPON	700
TENNIS	2000	10 ADULTS	500
TENNIS	2000	PREMIUM ACCOUNT	1000
TENNIS	2000	NO DISCOUNT APPLIED	0

> create view sys.e_amount_asc as
> select * from sys.event_details
> order by e_amount asc

EID	EVENT_NAME	ET_ID	EVENT_TYPE	DATE_TIME	DURATION	E_AMOUNT
E0302	HINDUSTANI	ET03	MUSIC	2021-11-12 17:00:00	1:30:00	900
E0102	HIP HOP	ET01	DANCE	2021-09-01 11:30:00	2:00:00	1000
E0301	CARNATIC	ET03	MUSIC	2021-10-01 09:30:00	2:30:00	1000
E0201	PAINTINGS	ET02	ARTS	2021-11-21 16:00:00	1:00:00	1500
E0101	CLASSICAL	ET01	DANCE	2021-08-29 02:30:00	2:00:00	2000
E0401	TENNIS	ET04	SPORTS	2021-10-22 10:30:00	3:00:00	2000
E0103	FOLK	ET01	DANCE	2021-09-30 10:00:00	2:00:00	3000

6.CREATE 5 INDEXES

1.To Search The Events That User Wanted By Event_Type And Event Name

> create index search_event
> using btree
> on sys.event_details (event_type , event_name)

2.For Admin To Search For A User With Mail Id , Unique Is Used So If User Is Applied 2 Times , Duplicate Values Can Be Eliminated

> CREATE UNIQUE INDEX users_emailID
> USING BTREE
> ON sys.user (email);

3.To Search For The User To See What All Booking One Have Made

> CREATE INDEX user_booking
> using btree
> ON sys.user_booking (UID)

4.For User To Search For Discounts Available While Booking

> CREATE INDEX search_discount_available
> using btree

on sys.event_discount(discount_name ASC)

5.For Users To Check If A Event Is Availablle For Given Amount

➢ create index EVENT_AMOUNT
  using btree
  on sys.event_details (e_amount)

sys.event_details

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Visible	Expression
event_details	0	PRIMARY	1	EID	A	7	NULL	NULL		BTREE	YES	NULL
event_details	1	search_event	1	EVENT_TYPE	A	4	NULL	NULL	YES	BTREE	YES	NULL
event_details	1	search_event	2	EVENT_NAME	A	7	NULL	NULL		BTREE	YES	NULL
event_details	1	EVENT_AMOUNT	1	E_AMOUNT	A	5	NULL	NULL		BTREE	YES	NULL

sys.user

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Visible	Comment	Index
user	0	PRIMARY	1	UID	A	7	NULL	NULL		BTREE	YES		
user	0	users_emailID	1	EMAIL	A	7	NULL	NULL		BTREE	YES		

sys.user_booking

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Visible	Expression
user_booking	0	PRIMARY	1	BID	A	6	NULL	NULL		BTREE	YES	NULL
user_booking	1	fk_uid_idx	1	UID	A	5	NULL	NULL		BTREE	YES	NULL
user_booking	1	fk_eid_idx	1	EID	A	5	NULL	NULL		BTREE	YES	NULL
user_booking	1	fk_did_idx	1	DID	A	4	NULL	NULL	YES	BTREE	YES	NULL
user_booking	1	fk_sid_idx	1	SID	A	4	NULL	NULL	YES	BTREE	YES	NULL
user_booking	1	user_booking	1	UID	A	5	NULL	NULL		BTREE	YES	NULL

sys.event_discount

Table	Non_unique	Key_name	Seq_in_inde:	Column_name	Collation	Cardinality	Sub_part	Packed	Index_type	Expr	Visible
event_discount	0	PRIMARY	1	DID	A	5	NULL	NULL	BTREE	NULL	YES
event_discount	1	search_discount_available	1	DISCOUNT_NAME	A	5	NULL	NULL	BTREE	NULL	YES