TICKET BOOKING SYSTEM

```
create database ticketbookingsystem;
CREATE SCHEMA IF NOT EXISTS `ticketbooking1` DEFAULT CHARACTER SET utf8 ;
USE `ticketbooking1` ;
-- -----------------------------------------------------
-- Table `ticketbooking1`.`venue`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `ticketbooking1`.`venue` (
 `venue_id` INT NOT NULL AUTO_INCREMENT,
 `venue_name` VARCHAR(255) NULL,
 `address` VARCHAR(255) NULL,
 PRIMARY KEY (`venue_id`))
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `ticketbooking1`.`event`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `ticketbooking1`.`event` (
 `event_id` INT NOT NULL AUTO_INCREMENT,
 `event_name` VARCHAR(255) NULL,
 `event_date` DATE NULL,
 `event_time` TIME NULL,
 `total_seats` INT NULL,
 `available_seats` INT NULL,
 `ticket_price` INT NULL,
 `event_type` VARCHAR(255) NULL,
 `venue_id` INT NOT NULL,
 PRIMARY KEY (`event_id`, `venue_id`),
 INDEX `fk_event_venue1_idx` (`venue_id` ASC) ,
 CONSTRAINT `fk_event_venue1`
```

```
 FOREIGN KEY (`venue_id`)
 REFERENCES `ticketbooking1`.`venue` (`venue_id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `ticketbooking1`.`customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `ticketbooking1`.`customer` (
 `id` INT NOT NULL AUTO_INCREMENT,
 `customer_name` VARCHAR(255) NULL,
 `email` VARCHAR(255) NULL,
 `phone_number` BIGINT NULL,
 PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `ticketbooking1`.``
-- -----------------------------------------------------



-- -----------------------------------------------------
-- Table `ticketbooking1`.`booking`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `ticketbooking1`.`booking` (
 `event_id` INT NULL,
 `customer_id` INT NULL,
 `num_tickets` INT NULL,
 `total_cost` INT NULL,
 `booking_date` DATE NULL,
```

```
 `booking_id` INT NOT NULL AUTO_INCREMENT,
 INDEX `fk_event_has_customer_customer1_idx` (`customer_id` ASC) ,
 INDEX `fk_event_has_customer_event1_idx` (`event_id` ASC) ,
 PRIMARY KEY (`booking_id`),
 CONSTRAINT `fk_event_has_customer_event1`
 FOREIGN KEY (`event_id`)
 REFERENCES `ticketbooking1`.`event` (`event_id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION,
 CONSTRAINT `fk_event_has_customer_customer1`
 FOREIGN KEY (`customer_id`)
 REFERENCES `ticketbooking1`.`customer` (`id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;
show databases;
use information_schema;
show databases;
-- INSERTIONS--
use TicketBookingSystem1;
describe event;
insert into venue(venue_name,address) values
('mumbai', 'marol andheri(w)'),
('chennai', 'IT Park'),
('pondicherry ', 'state beach');
insert into customer(customer_name,email,phone_number)
values
('harry potter','harry@gmail.com','45454545'),
('ronald weasley','ron@gmail.com','45454545'),
```

```
('hermione granger','her@gmail.com','45454545'),
('draco malfoy','drac@gmail.com','45454545'),
('ginni weasley','ginni@gmail.com','45454545'),
('severus snape','sev@gmail.com','56556');
insert into
event(event_name,event_date,event_time,total_seats,available_seats,ticket_price,event_type,venue_id
)
values
('Late Ms. Lata Mangeshkar Musical', '2021-09-12','20:00',320,270,600,'concert',3),
('CSK vs RCB', '2024-04-11','19:30',23000,3,3600,'sports',2),
('CSK vs RR', '2024-04-19','19:30',23000,10,3400,'sports',2),
('MI vs KKR', '2024-05-01','15:30',28000,100,8000,'sports',1);
insert into booking(event_id,customer_id,num_tickets,total_cost,booking_date) values
(4,1,2,640,'2021-09-12'),
(4,3,5,3000,'2024-03-15'),
(4,4,3,960,'2021-09-12'),
(5,1,3,10800,'2024-04-11'),
(5,3,5,18000,'2024-04-10'),
(6,5,10,34000,'2024-04-15'),
(7,2,4,32000,'2024-05-01'),
(7,6,1,8000,'2024-03-15');
set foreign_key_checks=0;
select * from booking;
select * from booking;
select * from customer;
select * from venue;
select * from event;
update event set event_name='conference cup' where event_id=4;
-- writing queries solution of the task 2--
```

```
-- Tasks 2: Select, Where, Between, AND, LIKE:--

-- 1st query solution from task 2--

-- 1. Write a SQL query to Insert at least 10 sample records into each table.--

-- inserting 10 sample records in venue table--

INSERT INTO venue (venue_id, venue_name, address) VALUES

(4, 'Venue 4', 'Address 4'),

(5, 'Venue 5', 'Address 5'),

(6, 'Venue 6', 'Address 6'),

(7, 'Venue 7', 'Address 7'),

(8, 'Venue 8', 'Address 8'),

(9, 'Venue 9', 'Address 9'),

(10, 'Venue 10', 'Address 10'),

(11, 'Venue 11', 'Address 11'),

(12, 'Venue 12', 'Address 12'),

(13, 'Venue 13', 'Address 13');

-- inserting 10 sample records in event table--

INSERT INTO event (event_id, event_name, event_date, event_time, total_seats, available_seats,

ticket_price, ype, venue_id) VALUES

(5, 'Event 5', '2024-04-10', '14:00', 300, 300, 30.00, 'Type 5', 5),

(6, 'Event 6', '2024-04-11', '15:00', 350, 350, 35.00, 'Type 6', 6),

(7, 'Event 7', '2024-04-12', '16:00', 400, 400, 40.00, 'Type 7', 7),

(8, 'Event 8', '2024-04-13', '17:00', 450, 450, 45.00, 'Type 8', 8),

(9, 'Event 9', '2024-04-14', '18:00', 500, 500, 50.00, 'Type 9', 9),

(10, 'Event 10', '2024-04-15', '19:00', 550, 550, 55.00, 'Type 10', 10),

(11, 'Event 11', '2024-04-06', '10:00', 100, 100, 10.00, 'Type 11', 11),

(12, 'Event 12', '2024-04-07', '11:00', 150, 150, 15.00, 'Type 12', 12),

(13, 'Event 13', '2024-04-08', '12:00', 200, 200, 20.00, 'Type 13', 13),

(14, 'Event 14', '2024-04-09', '13:00', 250, 250, 25.00, 'Type 14', 14);

-- inserting 10 sample records in customer table--
```

```
INSERT INTO customer (customer_id, customer_name, email, phone_number) VALUES
(7, 'Customer 7', 'customer7@example.com', '1234567896'),
(8, 'Customer 8', 'customer8@example.com', '1234567897'),
(9, 'Customer 9', 'customer9@example.com', '1234567898'),
(10, 'Customer 10', 'customer10@example.com', '1234567899'),
(11, 'Customer 1', 'customer11@example.com', '1234567890'),
(12, 'Customer 2', 'customer12@example.com', '1234567891'),
(13, 'Customer 3', 'customer13@example.com', '1234567892'),
(14, 'Customer 4', 'customer14@example.com', '1234567893'),
(15, 'Customer 5', 'customer15@example.com', '1234567894' ),
(16, 'Customer 6', 'customer16@example.com', '1234567895' );
-- inserting 10 sample records in booking table--
INSERT INTO booking (booking_id, num_tickets, total_cost, booking_date, customer_id, event_id)
VALUES
(11, 2, 640, '2024-04-01', 4, 1),
(12, 3, 960, '2024-04-02', 2, 2),
(13, 4, 800, '2024-04-03', 5, 2),
(14, 5, 1205, '2024-04-04', 4, 4),
(15, 6, 1800, '2024-04-05', 5, 5),
(16, 7, 2405, '2024-04-06', 2, 4),
(17, 8, 3200, '2024-04-07', 7, 7),
(18, 9, 4050, '2024-04-08', 7, 2),
(19, 10, 5000, '2024-04-09', 2, 3),
(20, 11, 6050, '2024-04-10', 4, 3);
-- 2nd query solution from task 2--
-- 2. Write a SQL query to list all Events.--
update event set event_name= 'conference cup' where event_id=4;
select * from event;
/*1 Late Ms. Lata Mangeshkar Musical 2021-09-12 20:00:00 320 270 600
```

```
concert 3
2 CSK vs RCB 2024-04-11 19:30:00 23000 3 3600 sports 2
3 CSK vs RR 2024-04-19 19:30:00 23000 10 3400 sports 2
4 conference cup 2024-05-01 15:30:00 28000 100 8000 sports 1
*/
-- 3rd query solution from task 2--
-- 3. Write a SQL query to select events with avallable tickets.--
select event_name,event_date,event_time,ticket_price,event_type
from event
where available_seats>0;
/*Late Ms. Lata Mangeshkar Musical 2021-09-12 20:00:00 600 concert
CSK vs RCB 2024-04-11 19:30:00 3600 sports
CSK vs RR 2024-04-19 19:30:00 3400 sports
conference cup 2024-05-01 15:30:00 8000 sports*/
-- 4th query solution from task 2--
-- 4. Write a SQL query to select events name partial match with 'cup'.--
select * from event
where event_name like '%cup%';
/* 4 conference cup 2024-05-01 15:30:00 28000 100 8000 sports 1
*/
-- 5th query solution from task 2--
-- 5. Write a SQL query to select events with ticket price range is between 1000 to 2500.--
select * from event
where ticket_price between 1000 and 2500;
/* NO OUTPUT
*/
-- 6th query solution from task 2--
-- 6. Write a SQL query to retrieve events with dates falling within a specific range.--
select * from event
```

```
where event_date between '2024-01-01' and '2024-04-30';
/*
2 CSK vs RCB 2024-04-11 19:30:00 23000 3 3600 sports 2
3 CSK vs RR 2024-04-19 19:30:00 23000 10 3400 sports 2
*/
-- 7th query solution from task 2--
-- 7. Write a SQL query to retrieve events with available tickets that also have "Concert" in their name.--
select * from event
where available_seats>0
and event_name like '%concert%';
/* NO OUTPUT
*/
-- 8th query solution from task 2--
-- 8. Write a SQL query to retrieve users in batches of 5, starting from the 6th user.--
select * from customer
order by id limit 5,5;
/*
6 severus snape sev@gmail.com56556
*/
-- 9th query solution from task 2--
-- 9. Write a SQL query to retrieve bookings details contains booked no of ticket more than 4.--
SELECT *
FROM booking
WHERE num_tickets > 4;
/*
4 3 5 3000 2024-03-15 18
5 3 5 18000 2024-04-10 21
6 5 10 34000 2024-04-15 22
*/
```

```
-- 10th query solution from task 2--
-- 10. Write a SQL query to retrieve customer information whose phone number end with '000--
select * from customer
where phone_number like '%000';
/* NO OUTPUT
*/
-- 11th query solution from task 2--
-- 11. Write a SQL query to retrieve the events in order whose seat capacity more than 15000.--
select * from event
where total_seats>15000
order by total_seats;
/*
2 CSK vs RCB 2024-04-11 19:30:00 23000 3 3600 sports 2
3 CSK vs RR 2024-04-19 19:30:00 23000 10 3400 sports 2
4 conference cup 2024-05-01 15:30:00 28000 100 8000 sports 1
*/
-- 12th query solution from task 2--
-- 12. Write a SQL query to select events name not start with 'x', 'Y', 'Z'--
select event_name from event
where event_name not like 'x%'
and event_name not like 'y%'
and event_name not like 'z%';
/*
Late Ms. Lata Mangeshkar Musical
CSK vs RCB
CSK vs RR
conference cup*/
-- writing queries solution for task 3--
-- Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:
```

```
-- 1st query solution from task 3--
-- 1. Write a SQL query to List Events and Their Average Ticket Prices.--
select event_id,avg(ticket_price) as average_ticket_price
from event group by event_id;
/*
1 600.0000
2 3600.0000
3 3400.0000
4 8000.0000*/
-- 2nd query solution from task 3--
-- 2. Write a SQL query to Calculate the Total Revenue Generated by Events.--
select sum(total_cost)as total_revenue
from booking;
/*107400*/
-- 3rd query solution from task 3--
-- 3. Write a SQL query to find the event with the highest ticket sales.--
SELECT e.event_id,e.event_name, SUM(num_tickets) AS total_tickets_sold
FROM booking b join event e on e.event_id=b.event_id
GROUP BY event_id,e.event_name
ORDER BY total_tickets_sold DESC
LIMIT 1;
/*4 conference cup 10*/
-- 4th query solution from task 3--
-- 4. Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event.--
select event_name,abs(total_seats-available_seats) as number_of_tickets_sold
from event order by number_of_tickets_sold;
/*
Late Ms. Lata Mangeshkar Musical 50
CSK vs RR 22990
```

```
CSK vs RCB 22997

conference cup 27900*/

-- 5th query solution from task 3--

-- 5. Write a SQL query to Find Events with No Ticket Sales.--

select * from event

where event_id not in

(select e.event_id from event e, booking b

where e.event_id=b.event_id);

/*

1 Late Ms. Lata Mangeshkar Musical 2021-09-12 20:00:00 320 270 600

concert 3

2 CSK vs RCB 2024-04-11 19:30:00 23000 3 3600 sports 2

3 CSK vs RR 2024-04-19 19:30:00 23000 10 3400 sports 2

*/

-- 6th query solution from task 3--

-- 6. Write a SQL query to Find the User Who Has Booked the Most Tickets.--

select customer_name,max(num_tickets)

from event e ,customer c,booking b

where e.event_id=b.event_id and b.customer_id=c.id group by customer_name ;

/*harry potter 2

hermione granger 5

draco malfoy 3*/

-- 7th query solution from task 3--

-- 7. Write a SQL query to List Events and the total number of tickets sold for each month.--

-- not able to solve --

/*4 conference cup 2021-09 5

4 conference cup 2024-03 5*/

-- 8th query solution from task 3--

-- 8. Write a SQL query to calculate the average Ticket Price for Events in Each Venue.--
```

```
select venue_name ,avg(ticket_price) as average_ticket_price
from event e,venue v
where e.venue_id=v.venue_id
group by venue_name ;
/*mumbai 8000.0000
chennai3500.0000
pondicherry 600.0000*/
-- 9th query solution from task 3--
-- 9. Write a SQL query to calculate the total Number of Tickets Sold for Each Event Type.--
select event_type,event_name from event e, booking b
where e.event_id=b.event_id
group by event_type,event_name;
/*sports conference cup*/
-- 10th query solution from task 3--
-- 10. Write a SQL query to calculate the total Revenue Generated by Events in Each Year.--
-- not able to solve
/*2021 1600
2024 3000*/
-- 11th query solution from task 3--
-- 11. Write a SQL query to list users who have booked tickets for multiple events.--
select c.id,c.customer_name
from customer c,booking b
where c.id=b.customer_id
group by c.id,c.customer_name
having count(distinct b.event_id)>1;
/*1 harry potter
3 hermione granger*/
-- 12th query solution from task 3--
-- 12. Write a SQL query to calculate the Total Revenue Generated by Events for Each User.--
```

```
select c.id,c.customer_name,sum(b.total_cost) as total_revenue
from customer c , booking b
where c.id=b.customer_id
group by c.id,c.customer_name
order by c.id;
/*1 harry potter 11440
2 ronald weasley 32000
3 hermione granger 21000
4 draco malfoy 960
5 ginni weasley 34000
6 severus snape 8000*/
-- 13th query solution from task 3--
-- 13. Write a SQL query to calculate the Average Ticket Price for Events in Each Category and Venue--
select v.venue_id,e.event_type as event_category,
v.venue_name,avg(ticket_price)
from event e, venue v
where e.venue_id=v.venue_id
group by e.event_type,v.venue_id
order by venue_id;
/*1 sports mumbai 8000.0000
2 sports chennai3500.0000
3 concert pondicherry 600.0000*/
-- 14th query solution from task 3--
-- 14. Write a SQL query to list user and the Total Number of Tickets Thev've Purchased in the last 30days--
-- task 4 query solutions--
-- 1st query solution in task 4--
-- 1. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery.--
select v.venue_name,avg(e.ticket_price) as Average_Ticket_Price
```

```
from venue v join event e on v.venue_id=e.venue_id
group by v.venue_name;
/*mumbai 8000.0000
chennai3500.0000
pondicherry 600.0000*/
-- 2nd query solution task 4--
-- 2. Find Events with More Than 50% of Tickets Sold using subquery.--
select * from event
where (total_seats-available_seats)>(select (total_seats/2)
from event) ;
-- 3rd query solution task 4--
-- 3. Calculate the Total Number of Tickets Sold for Each Event.--
select event_name,sum(total_seats-available_seats) as total_tickets_sold from event group by
event_name;
/*Late Ms. Lata Mangeshkar Musical 50
CSK vs RCB 22997
CSK vs RR 22990
conference cup 27900*/
-- 4th query solution task 4--
-- 4. Find Users Who Have Not Booked Any Tickets Using a NOT EXISTS Subquery. --
select * from customer where id not in(select distinct c.id from customer c join booking b on
c.id=b.customer_id);
/* NO OUTPUT
*/
-- 5th query solution task 4--
-- 5. List Events with No Ticket Sales Using a NOT IN Subquery.--
select event.* from event
where event_id not in(
select e.event_id from event e, booking b where e.event_id=b.event_id);
```

```
/*1 Late Ms. Lata Mangeshkar Musical 2021-09-12 20:00:00 320 270 600

concert 3

2 CSK vs RCB 2024-04-11 19:30:00 23000 3 3600 sports 2

3 CSK vs RR 2024-04-19 19:30:00 23000 10 3400 sports 2

*/

-- 6th query solution--

-- 6. Calculate the Total Number of Tickets Sold for Each Event Type Using a Subquery in the FROMClause--

select event_type,sum(total_seats-available_seats) as total_tickets_sold from event group by

event_type;

/*concert 50

sports 73887*/

-- 7th query solution--

-- 7. Find Events with Ticket Prices Higher Than the Average Ticket Price Using a Subquery in the WHEREClause.--

select * from event where ticket_price>(select avg(ticket_price) from event);

-- 8th query solution--

-- 8. Calculate the Total Revenue Generated by Events for Each User Using a Correlated Subquery.--

select c.customer_name,sum(total_cost) from customer c ,booking b where c.id=b.customer_id group by

c.customer_name;

/*harry potter 11440

ronald weasley 32000

hermione granger 21000

draco malfoy 960

ginni weasley 34000

severus snape 8000*/

-- 9th query solution--

-- 9. List Users Who Have Booked Tickets for Events in a Given Venue Using a Subquery in the WHEREClause.--

select c.*
```

```
FROM customer c
join booking b ON c.id = b.customer_id
join event e on e.event_id=b.event_id
join venue v on v.venue_id=e.venue_id
where b.event_id in (select e.event_id from venue where venue_name = 2)
group by c.id, c.customer_name;



/* NO OUTPUT
*/
-- 10th query solution task 4--
-- 10. Calculate the Total Number of Tickets Sold for Each Event Category Using a Subquery with Group By--
select event_type,sum(total_seats-available_seats) as total_number_of_tickets from event group by
event_type;
/*concert 50
sports 73887*/
-- 12th query solution task 4--
-- 12. Calculate the Average Ticket Price for Events in Each Venue Using a Subquery--
select v.venue_id,v.venue_name,
avg(e.ticket_price) AS average_ticket_price
from venue v
join event e on v.venue_id = e.venue_id
group by v.venue_id, v.venue_name;



/*1 mumbai 8000.0000
2 chennai3500.0000
3 pondicherry 600.0000*/
```