SQL TASK GTTRAIN

Credentials

//User inputs $username,$password,$confirmpassword into Log In screen //application ensures $password = $confirmpassword.

SELECT * FROM User WHERE Username = $Username AND Password = $Password;

Make Account

Create Account //user inputs $username, $password, $confirmpassword into New User Registration screen //make sure that $password = $confirmpassword.

//make sure all forms are filled in //check to ensure $username isn't already in use.

SELECT Username FROM User WHERE Username = $Username;

SELECT * FROM Customer WHERE Email = $Email;

//If $username or $email is already in use, application asks for different username or email from user, otherwise

INSERT INTO User(Username, Password) VALUES $Username, $Password;

INSERT INTO Customer(Username, Email) VALUES $Username, $Email;

Add School Info

//Make sure all forms are filled in

//Determine if the customer is a student, if customer is a student, $isStudent = 1;

Update Customer SET isStudent = 1 WHERE Username = $Username;

View Train Schedule

//Make sure all forms are filled in

//Check if the TrainNum is correct.

SELECT TrainNum FROM TrainRoute WHERE TrainNum = $TrainNum;

//Display train schedule

SELECT * FROM Stop WHERE TrainNum = $TrainNum ORDER BY ArrivalTime;

Search Train

//Get stop information from database

SELECT StationName From Stop GROUP BY StationName;

//display the selected train schedule details

SELECT tr.TrainNum, s.DepartureTime, s.ArrivalTime, tr.FstClassPrice, tr.SndClassPrice FROM

    (SELECT departure.TrainNum, departure.DepartureTime, arrival.ArrivalTime FROM

    (SELECT TrainNum, DepartureTime

    FROM Stop

    WHERE StationName = $StationName

    AND DepartureTime IS NOT NULL) departure

    JOIN

    (SELECT TrainNum, ArrivalTime

    FROM Stop

    WHERE StationName = $StationName

    AND ArrivalTime IS NOT NULL) arrival

    ON departure.TrainNum = arrival.TrainNum AND departure.DepartureTime < arrival.ArrivalTime) s

    JOIN TrainRoute tr

    ON tr.TrainNum = s.TrainNum";

Make Reservation

//check if the customer is a student, if yes, then apply student discount

SELECT IsStudent FROM Customer WHERE Username = $Username;

//display all the cards that are owned by the customer

SELECT CardNum FROM PaymentInfo WHERE Username = $Username;

Add Card

//check if the card has been added to the system

SELECT * FROM PaymentInfo WHERE CardNum = $CardNum;

//If not add the new card

INSERT INTO PaymentInfo VALUES($CardNum, $CVV, $ExpirationDate, $NameOnCard);

Delete Card

//delete selected card from the customer

DELETE FROM PaymentInfo WHERE CardNum = $CardNum;

Confirm

//display the latest reservation ID

SELECT ReserveID FROM Reservation ORDER BY ReserveID DESC LIMIT 1;

// add new reservation information into Reservation table

INSERT INTO Reservation (IsCancelled, Username, CardNum) VALUES ($IsCancelled, $Username, $CardNum);

// add new reservation information into Reserve table

INSERT INTO Reserve VALUES ($ReserveID, $TrainNum, $Class, $DepartureDate, $PassengerName, $NumOfBaggage, $DepartFrom, $ArriveAt, $UpdateTimes);

Update

//check if the ReserveID and Username are valid to get

SELECT ReserveID FROM Reservation WHERE ReserveID = $ReserveID and Username = $Username;

//get the reservation information related to the ReserveID

SELECT * FROM Reserve WHERE ReserveID = $ReserveID;

//get departure time and arrival time from Stop table, and combine with the reservation information from Reserve table to display a new table.

SELECT departure.DepartureTime, arrival.ArrivalTime FROM

        (SELECT TrainNum, DepartureTime

        FROM Stop

        WHERE TrainNum = $TrainNum AND StationName = $StationName)

departure

        JOIN

        (SELECT TrainNum, ArrivalTime

        FROM Stop

        WHERE TrainNum = TrainNum AND StationName = $StationName)

arrival

        ON departure.TrainNum = arrival.TrainNum;

//get the price of first class if the customer choose first class when making reservation.

SELECT FstClassPrice FROM TrainRoute WHERE TrainNum = $TrainNum;

// get the price of the second class if the customer choose second class when making reservation.

SELECT SndClassPrice FROM TrainRoute WHERE TrainNum = $TrainNum;

// check if the customer a student, if yes, apply student discount on the total cost.

SELECT Customer.IsStudent FROM (Customer JOIN Reservation ON Customer.Username = Reservation.Username) WHERE Reservation.ReserveID = $ReserveID;

//check if the reservation has been cancelled, if yes, customer is not allowed to update the reservation.

SELECT IsCancelled FROM Reservation WHERE ReserveID = $ReserveID;

//update the departure date on the selected ticket.

UPDATE Reserve SET DepartureDate = $DepartureDate, UpdateTimes = $UpdateTimes WHERE ReserveID = $ReserveID AND TrainNum = $TrainNum;

Cancel

//display the reservation information related to the ReserveID

SELECT ReserveID FROM Reservation WHERE ReserveID = $ReserveID;

// get the information from Reserve table related to that reserce ID.

SELECT * FROM Reserve WHERE ReserveID = $ReserveID;

//get departure time and arrival time from Stop table, and combine with the reservation information from Reserve table to display a new table.

SELECT departure.DepartureTime, arrival.ArrivalTime FROM

        (SELECT TrainNum, DepartureTime

        FROM Stop

        WHERE TrainNum = $TrainNum AND StationName = $TrainNum)

departure

        JOIN

        (SELECT TrainNum, ArrivalTime

        FROM Stop

        WHERE TrainNum = $TrainNum AND StationName = $StationName)

arrival

        ON departure.TrainNum = arrival.TrainNum;

//get the price of first class if the customer choose first class when making reservation.

SELECT FstClassPrice FROM TrainRoute WHERE TrainNum = $TrainNum;

// get the price of the second class if the customer choose second class when making reservation.

SELECT SndClassPrice FROM TrainRoute WHERE TrainNum = $TrainNum;

// check if the customer a student, if yes, apply student discount on the total cost.

SELECT Customer.IsStudent FROM (Customer JOIN Reservation ON Customer.Username = Reservation.Username) WHERE Reservation.ReserveID = $ReserveID;

//check if the reservation has been cancelled, if yes, customer is not allowed to cancel the

reservation.

SELECT IsCancelled FROM Reservation WHERE ReserveID = $ReserveID;

// cancel the selected reservation

UPDATE Reservation SET IsCancelled = 1 WHERE ReserveID = $ReserveID;

View Review

//check if the TrainNum is valid

SELECT * FROM TrainRoute WHERE TrainNum = $TrainNum;

//display the comment and rating related to the TrainNum

SELECT Comment, Rating FROM Review WHERE TrainNum = $TrainNum;

Give Review

// select the TrainNum to give a particular train review

SELECT TrainNum From TrainRoute WHERE TrainNum = $TrainNum;

// add new review information into Review table

INSERT INTO Review(Comment, Rating, Username, TrainNum) VALUES ($Comment, $Rating, $Username, $TrainNum);

Generating Revenue Report

//create view that contains the Month, EarliestDate, TotalPrice, IsCancelled information which will be utilized in the next step

CREATE VIEW TotalRevenueReport (Month, EarliestDate, TotalPrice, IsCancelled) AS SELECT MONTH(DepartureDate), MIN(DepartureDate), SUM((50 * UpdateTimes + IF(IsStudent = 0,(IF(Class = '1st Class', FstClassPrice, SndClassPrice) + IF(NumOfBaggage > 2, 30 * NumOfBaggage, 0)), 0.8 * (IF(Class = '1st Class', FstClassPrice, SndClassPrice) + IF(NumOfBaggage > 2, 30 * NumOfBaggage, 0))))) AS TotalPrice, IsCancelled FROM Reservation NATURAL JOIN Customer NATURAL JOIN Reserve NATURAL JOIN TrainRoute GROUP BY ReserveID;

//create view that contains the Month, Refund information from the TotalRevenueReport

CREATE VIEW CancelledRevenueReport (Month, Refund) AS SELECT Month, (CASE WHEN DATEDIFF(CURDATE(), EarliestDate) > 7 THEN 0.8 * TotalPrice - 50 WHEN

(DATEDIFF(CURDATE(), EarliestDate) <= 7) AND DATEDIFF(CURDATE(), EarliestDate) >= 1 THEN 0.5 * TotalPrice -50 ELSE 0 END) AS Refund FROM TotalRevenueReport WHERE IsCancelled = 1;

//display the total revenue of each month (Only need to show 3 months)

SELECT MONTHNAME(STR_TO_DATE(T.Month, '%m')), SUM(T.Revenue) FROM (SELECT Month AS Month,TotalPrice AS Revenue FROM TotalRevenueReport UNION ALL SELECT Month AS Month, (-1) * Refund AS Revenue FROM CancelledRevenueReport) AS T GROUP BY T.Month ORDER BY T.Month LIMIT 3;

//drop the previous views

DROP VIEW TotalRevenueReport;

DROP VIEW CancelledRevenueReport;

Generating Popular Train Report

//create view that contains the Date, TrainNum, NumOfReservation information which are not cancelled and ordered by DepartureDate and ReserveNum.

CREATE VIEW PopularRouteReport (Date, TrainNum, NumOfReservation) AS SELECT DepartureDate, TrainNum, COUNT(ReserveID) AS ReserveNum FROM Reservation NATURAL JOIN Reserve WHERE isCancelled = 0 GROUP BY TrainNum, MONTH(DepartureDate) ORDER BY MONTH(DepartureDate) ASC, ReserveNum DESC;

//select specific month, TrainNum, NumOfReservation information from the PopularRouteReport view and display the top 3 most popular train (calculated by number of reservations) of each month.

SELECT MONTHNAME(Date), TrainNum, NumOfReservation FROM PopularRouteReport WHERE STRCMP(substring(Date,1,4),$currentYear) = 0 AND STRCMP(substring(Date,6,2),$month)= 0 LIMIT 3;

//drop the PopularRouteReport view

DROP VIEW PopularRouteReport;