Team: Akıllı Arayüzler (Group 9)

Orçun Özdemir

İsmet Erdem Demir

Emir Atışay

DATABASE MANAGEMENT SYSTEMS TERM PROJECT: Bestflight.com

Introduction

We implemented a web-based application design for flight ticket-supply company. Our database system tries to form a connection between the customers who are looking for flight tickets and airline companies. In our database, simply we have two users which are customers and airway companies. Subsequently, they are both identified with their unique ID’s. Customers should register with their mail addresses, passwords, user names and phone numbers. In addition, airway companies should register with their user names, passwords, mail addresses and their phone numbers.

Problem Description

When customers want to buy flight tickets, they have to go to airports or corresponding sell points. Creating a database system that will allow users to buy flight tickets from online, saves the users’ time and energy. Thus, Bestflight.com provides this service to users.

Design

While designing our database, we have urge upon 10 entities, which are Airway, Customer, Address, Payment, Flight, Airplane, UserType, Discpunt, Ticket and Country. While doing it, we try to have simplified database and in order to do that, we made collection of basic attributes. Followingly, we form a 12 relation. Those relations are Buys, UsesDiscount, isKindOfUser, PaysWith, FliesWith,AddressOf, FliesTo, FliesFrom HiresPlane, BelongsTo, PlaceIn and Distance. However, while forming schema in the light of our entities and relations, we exclude some of the relations and added some foreign key to entities, while relation is one-to-one or one-to-many regarding to advise of our TA about the normalizations. Than as a result of it, table implementations are shaped, which is going to be discussed in the following part. Also, since we have large population of relations, we found a way to reduce it by using some queries. For example, instead of having a Distance relation, we use the following command;

select (arrival\_date-departure\_date) as Distance, T.flight\_id

from FliesTo natural join FliesFrom natural join Ticket as FF natural join Ticket as T

By using those kind of functions and with a new version of tables, we then decided on our queries. As we are an mediator between the customer and airway company, we design queries that includes some important measures of a flight like the duration of flight, price comparisons for a customer. Also, it outputs data that important for companies to hold statistics like most loyal customer (ie. Customer has the most flight) and cities that has the most airport activities. We decided to have those queries available and include them in the corresponding php and html files.

Implementation

We used WinSCP application to reach our istavrit.eng.ku.edu.tr server. In addition, we created a public HTML file for assistants reaching our PHP and HTML files. We created a PHP file called connect.php to connect istavrit.eng.ku.edu.tr with our project group user name and password and we threw an exception whether the connection failed. Then, we created a customer.html file for users that wants to register our site. Subsequently, the HTML file connects through ticket.php file for entering the values to database. The ticket.php calls connect.php and customer.html to reach the database and the registration page. Finally, when the user fill the form, specific information will be uploaded to the database.Then, we created PHP files which returns queries that we proposed. To connect each PHP file, we used the following code:

$con=mysqli\_connect("localhost","group9","<passwordofgroup9>","group9");

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

}

(*Figure 1:* Implementation of the connection between the server and MYSQL using PHP files.)

Subsequently, queries have taken data from our database whom we created for the BestFlight.com. For example, to find the customer who has bought the longest duration of flight ticket, we used the following code:

$result = mysqli\_query($con,"select C.user\_name,SSK.Distance as D

from

Customer as C,(select (arrival\_date-departure\_date) as Distance, T.customer\_id,T.date as date

from FliesTo natural join FliesFrom natural join Ticket as FF natural join Ticket as T) as SSK

where SSK.customer\_id=C.customer\_id

order by SSK.Distance desc limit 1

");

(*Figure 2:* Query of the customer who has bought the longest duration of flight ticket.)

Consequently, we had to display the output within a table form which led us to writing the following code:

echo "

<th>Find the customer who has bought the longest duration of flight ticket.</th>

<table border='1'>

<tr>

<th>user\_name</th>

<th>D</th>

</tr>";

while($row = mysqli\_fetch\_array($result))

{

echo "<tr>";

echo "<td>" . $row['user\_name'] . "</td>";

echo "<td>" . $row['D'] . "</td>";

echo "</tr>";

}

echo "</table>";

(*Figure 3:* Table implementation for the given query by using PHP.)

Results

To sum up, we’ve sustained proper ticket supply for the customers who are willing to buy ticket for a specific flight. To do that, we create a panel for ticket transaction. And we also checked it in our databases whether the new ticket is imported to the “Tickets” entity or not. At the end, we saw that, as user inputs the correct parts in the ticket, the sample ticket outputted to the screen and new ticket is inserted to the database. On the other hand, queries are also outputted, which are specified in the previous parts in the report. Tabularized output values are prompted as it’s clicked. And those prompted values have an heading that gives information about the query.