

Database Final Project Report

Yuxuan Wang, Peiyang Wen
Databases Spring 2021

1. User Cases Overview

1.1 Public Search

Introduction:

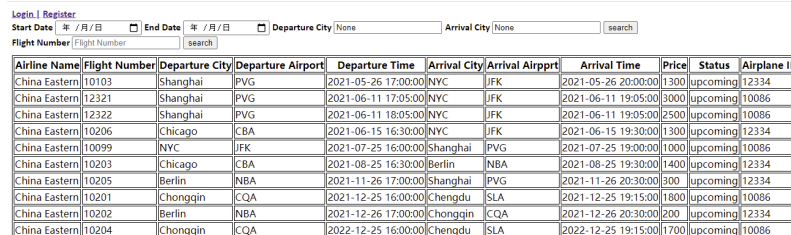
On this page, all the users can search for upcoming flights (we determine that based on departure time, not on flight status) using either a single flight number or a combination of departure/arrival city and time. Start date and end date are required, while departure/arrival city is optional.

Query Code:

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
from flight f join ticket using (flight_num) join purchases p using (ticket_id) JOIN airport
a on a.airport_name=f.departure_airport JOIN airport b on
b.airport_name=f.arrival_airport WHERE f.flight_num = '{}'
```

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
from flight f join ticket using (flight_num) join purchases p using (ticket_id) JOIN airport
a on a.airport_name=f.departure_airport JOIN airport b on
b.airport_name=f.arrival_airport
WHERE a.airport_city = '{}' and b.airport_city = '{}' and f.departure_time >= '{}'
and f.departure_time <= '{}' order by f.departure_time
```

Screenshot:



The screenshot shows a web interface for flight search. At the top, there are input fields for 'Start Date' (MM/DD/YYYY), 'End Date' (MM/DD/YYYY), 'Departure City' (dropdown), and 'Arrival City' (dropdown), followed by a 'search' button. Below these fields is a table with the following columns: Airline Name, Flight Number, Departure City, Departure Airport, Departure Time, Arrival City, Arrival Airport, Arrival Time, Price, Status, and Airplane ID. The table contains 15 rows of flight data.

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10203	Chicago	CBA	2021-08-25 16:30:00	Berlin	NBA	2021-08-25 19:30:00	1400	upcoming	12334
China Eastern	10205	Berlin	NBA	2021-11-26 17:00:00	Shanghai	PVG	2021-11-26 20:30:00	300	upcoming	12334
China Eastern	10201	Chongqing	CQA	2021-12-25 16:00:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqing	CQA	2021-12-26 20:30:00	200	upcoming	12334
China Eastern	10204	Chongqing	CQA	2022-12-25 16:00:00	Chengdu	SLA	2022-12-25 19:15:00	1700	upcoming	10086

1.2 Customer

(1) Home Page: View My Flights

Introduction:

On this page, customers will be able to see the upcoming flights he/she purchased. The “upcoming” flights here are defined as all flights with departure time larger than current date. Additionally, customers can search for flights based on Departure/Arrival Date and Departure/Arrival City.

Query Code:

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
From flight f join ticket using (flight_num) join purchases p using (ticket_id) JOIN
airport a on a.airport_name=f.departure_airport JOIN airport b on
b.airport_name=f.arrival_airport
```

WHERE f.departure_time > '{ }'
ORDER BY f.departure_time

Screenshot:

Purchase Tickets | My Spending

Start Date: 年 / 月 / 日 End Date: 年 / 月 / 日 Departure City: None Arrival City: None search

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10205	Berlin	NBA	2021-11-26 17:00:00	Shanghai	PVG	2021-11-26 20:30:00	300	upcoming	12334
China Eastern	10201	Chongqin	CQA	2021-12-25 16:00:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqin	CQA	2021-12-26 20:30:00	200	upcoming	12334
China Eastern	10204	Chongqin	CQA	2022-12-25 16:00:00	Chengdu	SLA	2022-12-25 19:15:00	1700	upcoming	10086

Customer: wpy@163.com Logout

(2) Purchase Tickets Page: Purchase Ticket & Search For Flights

Introduction:

On this page, customers will be able to see all the upcoming flights. Additionally, customers can search for flights based on Departure/Arrival Date and Departure/Arrival City. A buy button is attached to each flight. Customers can purchase the flight by clicking on the button. If the flight is sold out, an error message will be displayed once the customer clicks on the button. Past flights have no buy buttons attached to them.

Query Code:

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
from flight f join ticket using (flight_num) join purchases p using (ticket_id) JOIN airport
a on a.airport_name=f.departure_airport JOIN airport b on
b.airport_name=f.arrival_airport
```

WHERE f.departure_time > '{ }'

order by f.departure_time

Screenshot:

My Flights | My Spending

Departure City: departure city Arrival City: arrival city Start Date: 年 / 月 / 日 End Date: 年 / 月 / 日 search

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	Buy
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086	Buy
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086	Buy
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334	Buy
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086	Buy
China Eastern	10203	Chicago	CBA	2021-08-25 16:30:00	Berlin	NBA	2021-08-25 19:30:00	1400	upcoming	12334	Buy
China Eastern	10205	Berlin	NBA	2021-11-26 17:00:00	Shanghai	PVG	2021-11-26 20:30:00	300	upcoming	12334	Buy
China Eastern	10201	Chongqin	CQA	2021-12-25 16:00:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086	Buy
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqin	CQA	2021-12-26 20:30:00	200	upcoming	12334	Buy
China Eastern	10204	Chongqin	CQA	2022-12-25 16:00:00	Chengdu	SLA	2022-12-25 19:15:00	1700	upcoming	10086	Buy

Customer: wpy@163.com Logout

(3) My Spending Page: Track my Spending

Introduction:

In this page, customers will be presented with their month-wise spendings in the past six months in a bar chart. Additionally, customers can select an arbitrary range of dates to further explore the month-wise spendings.

Query Code:

```
select year(g.purchase_date), month(g.purchase_date), sum(g.price) from
(SELECT distinct t.ticket_id, f.flight_num, f.price, p.purchase_date
from flight f join ticket t using (flight_num) join purchases p using (ticket_id) WHERE
p.customer_email = '{ }' and p.booking_agent_id IS NULL) g
where g.purchase_date >= '{ }' and g.purchase_date <= '{ }'
```

group by year(g.purchase_date), month(g.purchase_date)
order by year(g.purchase_date) desc, month(g.purchase_date) desc;

select year(g.purchase_date), month(g.purchase_date), sum(g.price)*1.1 from
(SELECT distinct t.ticket_id, f.flight_num, f.price, p.purchase_date
from flight f join ticket t using (flight_num) join purchases p using (ticket_id) WHERE
p.customer_email = '{}' and p.booking_agent_id IS NOT NULL) g
where g.purchase_date >= '{}' and g.purchase_date <= '{}'
group by year(g.purchase_date), month(g.purchase_date)
order by year(g.purchase_date) desc, month(g.purchase_date) desc;

Screenshot:



1.3 Booking Agent

(1) Home Page: View My Flights

Introduction:

On this page, agents will be able to see the upcoming flights he/she purchased. The “upcoming” flights here are defined as all flights with departure time larger than current date. Additionally, agents can search for flights based on Departure/Arrival Date and Departure/Arrival City.

Query Code:

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id \
from flight f join ticket using (flight_num) join purchases p using (ticket_id) JOIN airport
a on a.airport_name=f.departure_airport JOIN airport b on
b.airport_name=f.arrival_airport\
WHERE p.booking_agent_id = '{}' and f.departure_time > '{}' \
order by f.departure_time
```

Screenshot:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10203	Chicago	CBA	2021-08-25 16:30:00	Berlin	NBA	2021-08-25 19:30:00	1400	upcoming	12334
China Eastern	10201	Chongqing	CQA	2021-12-25 16:00:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqing	CQA	2021-12-26 20:30:00	200	upcoming	12334

(2) Purchase Tickets Page: Purchase Tickets & Search for Flights

Introduction:

On this page, agents will be able to see all the upcoming flights. Additionally, agents can search for flights based on Departure/Arrival Date and Departure/Arrival City. A buy button is attached to each flight. Agents can purchase the flight by inputting the right customer email and clicking on the button. If the flight is sold out or the customer email

is invalid, an error message will be displayed once the customer clicks on the button. Past flights have no buy buttons attached to them.

Query Code:

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
from flight f join airport a on a.airport_name=f.departure_airport JOIN airport b on
b.airport_name=f.arrival_airport
WHERE f.departure_time > '{ }'
order by f.departure_time
```

Screenshot:

Purchase Tickets | My Commissions | Top Customers

Start Date: 年 / 月 / 日 End Date: 年 / 月 / 日 Departure City: None Arrival City: None search

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NVC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NVC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10099	NVC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10203	Chicago	CBA	2021-08-25 16:30:00	Berlin	NBA	2021-08-25 19:30:00	1400	upcoming	12334
China Eastern	10201	Chongqin	CQA	2021-12-25 16:00:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqin	CQA	2021-12-26 20:30:00	200	upcoming	12334

Agency ID: 12344 Logout

(4) My Commission Page: View My Commissions

Introduction:

In this page, customers will be presented with their commissions earned/ tickets sold/ average sold ticket price in the past 30 days. Additionally, agents can select an arbitrary range of dates to further explore the commission statistics.

Query Code:

```
select sum(g.price), count(g.ticket_id) from
(SELECT distinct t.ticket_id, f.flight_num, f.price, p.purchase_date
from flight f join ticket t using (flight_num) join purchases p using (ticket_id) WHERE
p.booking_agent_id = '{ }' ) g
where g.purchase_date >= '{ }' and g.purchase_date <= '{ }'
```

Screenshot:

Purchase Tickets | My Flights | Top Customers

Start Date: 年 / 月 / 日 End Date: 年 / 月 / 日 search

Total Commission Fees in Past 30 Days: 5440.00
Number of Tickets Sold in Past 30 Days: 19
Average Sold Ticket Price in Past 30 Days: 286.32

Customer: 12344 Logout

(5) Top Customer Page: View Top Customers

Introduction:

In this page, customers will be presented with the top five customers in the past six months in a bar chart, regarding commission fees and number of tickets bought.

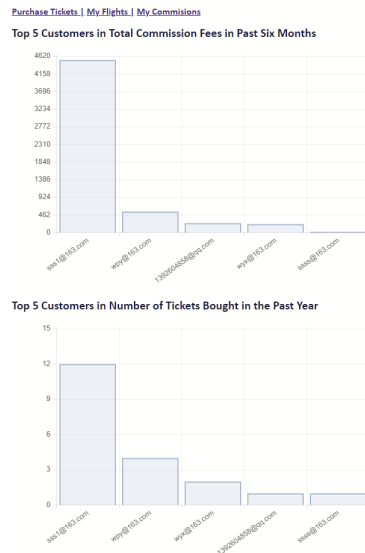
Query Code:

```
select g.customer_email, sum(g.price)/10 total_commission from
(SELECT distinct t.ticket_id, f.flight_num, f.price, p.purchase_date, p.customer_email
from flight f join ticket t using (flight_num) join purchases p using (ticket_id) WHERE
p.booking_agent_id = '{ }' ) g
where g.purchase_date >= '{ }' and g.purchase_date <= '{ }'
group by g.customer_email
order by sum(g.price) desc
limit 5;
```

```
select g.customer_email, count(g.ticket_id) total_commission from
(SELECT distinct t.ticket_id, f.flight_num, f.price, p.purchase_date, p.customer_email
from flight f join ticket t using (flight_num) join purchases p using (ticket_id) WHERE
p.booking_agent_id = '{ }' ) g
where g.purchase_date >= '{ }' and g.purchase_date <= '{ }'
```

group by g.customer_email
order by count(g.ticket_id) desc
limit 5;

Screenshot:



1.4 Airline Staff

(1) Homepage: View my flights and change flight status

Introduction:

This is the homepage for staff after login. Flights of staff's airline within the next 30 days will be displayed by default. Staff can also search for flights in the airline through start/end date and departure/arrival city. We combined the change flight status function into a homepage as staff can enter new status and click submit for the flight to be changed.

Query Code:

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
from flight f join airline_staff s using (airline_name) JOIN airport a on
a.airport_name=f.departure_airport JOIN airport b on b.airport_name=f.arrival_airport
WHERE f.airline_name = '{}' and f.departure_time >= '{}' and f.departure_time <=
{}' order by f.departure_time
```

```
SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport,
f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status,
f.airplane_id
from flight f join airline_staff s using (airline_name) JOIN airport a on
a.airport_name=f.departure_airport JOIN airport b on b.airport_name=f.arrival_airport
WHERE f.airline_name = '{}' and f.flight_num = '{}'
```

Screenshot:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airprt	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	<input type="text" value="new status"/> <input type="button" value="Submit"/>
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086	<input type="text" value="new status"/> <input type="button" value="Submit"/>
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086	<input type="text" value="new status"/> <input type="button" value="Submit"/>
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334	<input type="text" value="new status"/> <input type="button" value="Submit"/>

(2) Create new flights

Introduction:

The staff can create a new flight for the airline, and this page can be accessed from the homepage. After inputting the necessary information for flight creation, the staff will be directed back to the homepage. Check whether it's really a staff operating.

Query Code:

INSERT INTO flight VALUES('','','','','','','','','','')\n

Screenshot:

Back to Homepage | View the Customers of a Flight | Add New Airplanes | Add New Airports | View all Booking Agents | View Most Frequent Customers | View Reports | Comparison of Revenue earned | View Top Destinations

Customer: Arthur Logout

airline name

flight num

departure airport

arrival airport

price

flight status

airplane id

Create

(3) Add new airplanes

Introduction:

The staff can add new airplanes for the airline. Similarly, the staff will input information and then be directed back to the homepage. Check whether it's really a staff operating and prevent adding airplanes of other airlines.

Query Code:

INSERT INTO airplane VALUES('','','')\n

Screenshot:

Back to Homepage | View the Customers of a Flight | Create New Flights | Add New Airports | View all Booking Agents | View Most Frequent Customers | View Reports | Comparison of Revenue earned | View Top Destinations

Customer: Arthur Logout

airline name

airplane id

seat number

Add

(4) Add new airports

Introduction:

The staff can add new airports. Similarly, the staff will input information and then be directed back to the homepage. Check whether it's really a staff operating.

Query Code:

INSERT INTO airport VALUES('','')\n

Screenshot:

Back to Homepage | View the Customers of a Flight | Create New Flights | Add New Airplanes | View all Booking Agents | View Most Frequent Customers | View Reports | Comparison of Revenue earned | View Top Destinations

Customer: Arthur Logout

airport name

airport city

Add

(5) View all booking agents

Introduction:

Top 5 agents based on number of tickets sold in the past month/year, and based on total commission earned in the past year. .

Query Code:

Select p.customer_email, count(booking_agent_id) from purchases p join ticket t using (ticket_id) where p.booking_agent_id is not null and t.airline_name = '' and p.purchase_date >= '' and p.purchase_date <= ''\n

group by p.customer_email order by count(p.booking_agent_id) desc limit 5

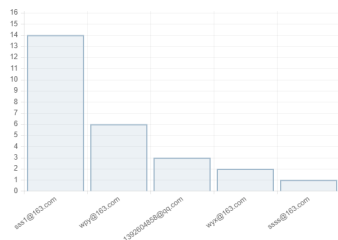
Select p.booking_agent_id, sum(f.price)/10 from purchases p join ticket t using (ticket_id) join flight f using (flight_num) where t.airline_name = '\{\}' and p.purchase_date >= '\{\}' and p.purchase_date <= '\{\}' and p.booking_agent_id is not Null group by p.booking_agent_id order by sum(f.price) desc limit 5

Screenshot:

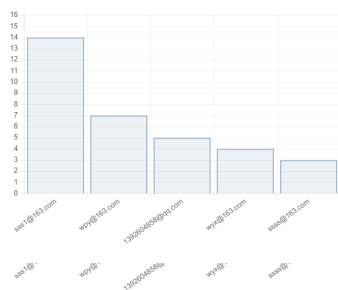
[Back to Homepage](#) | [View the Customers of a Flight](#) | [Create New Flights](#) | [Add New Airplanes](#) | [Add New Airports](#) | [View Most Frequent Customers](#) | [View Reports](#) | [Comparison of Revenue earned](#) | [View Top Destinations](#)

Customer: Arthur Logout

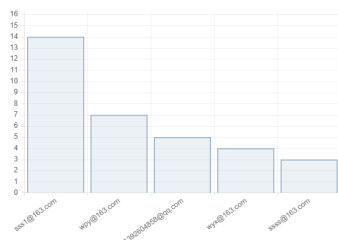
Top 5 Agents in Total Ticket Sales in the Past Month



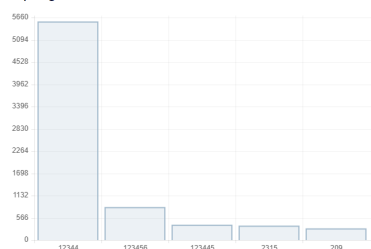
Top 5 Agents in Total Ticket Sales in the Past Year



Top 5 Agents in Total Ticket Sales in the Past Year



Top 5 Agents in Total Commission in the Past Year



(6) View frequent customers

Introduction:

The staff can view the 3 most frequent customers of the airline for the past year. The staff can also use the customer's email to search for the flights this customer has taken in the airline. Check whether it's really a staff operating.

Query Code:

Select c.name, p.customer_email, count(distinct ticket_id) from purchases p join customer c on c.email = p.customer_email join ticket t using (ticket_id)

where t.airline_name = '{}' and p.purchase_date >= '{}' and p.purchase_date <= '{}' group by p.customer_email order by count(ticket_id) desc limit 3

SELECT distinct f.airline_name, f.flight_num, a.airport_city, f.departure_airport, f.departure_time, b.airport_city, f.arrival_airport, f.arrival_time, f.price, f.status, f.airplane_id from flight f join ticket using (flight_num) join purchases p using (ticket_id) JOIN airport a on a.airport_name=f.departure_airport JOIN airport b on b.airport_name=f.arrival_airport where f.airline_name = '{}' and f.arrival_time <= '{}' and p.customer_email = '{}' order by f.departure_time desc

Screenshot:

[Back to Homepage](#) | [View the Customers of a Flight](#) | [Create New Flights](#) | [Add New Airplanes](#) | [Add New Airports](#) | [View all Booking Agents](#) | [View Reports](#) | [Comparison of Revenue earned](#) | [View Top Destinations](#)

Customer: Arthur Logout

Customer Email

Customer Name	Customer Email	Number of Tickets Bought in Last Year
Arthur	wpy@163.com	33
kaxf	sss1@163.com	17
kaxf	1392604858@qq.com	9

[View the Customers of a Flight](#) | [Create New Flights](#) | [Add New Airplanes](#) | [Add New Airports](#) | [View all Booking Agents](#) | [View Most Frequent Customers](#) | [View Reports](#) | [Comparison of Revenue earned](#) | [View Top Destinations](#)

Customer: Arthur Logout

Customer Email

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airprt	Arrival Time	Price	Status	Airplane ID
China Eastern	10000	NYC	JFK	2020-12-25 16:00:00	Shanghai	PVG	2020-12-25 19:00:00	1200	delayed	10086

(7) View reports

Introduction:

The staff can select a range of months and see the month-wise number of tickets sold. Default is to show this for the past 6 months. Check whether it's really a staff operating.

Query Code:

select year(g.purchase_date), month(g.purchase_date), count(distinct g.ticket_id) from (SELECT distinct t.ticket_id, t.airline_name, f.flight_num, f.price, p.purchase_date from flight f join ticket t using (flight_num) join purchases p using (ticket_id)) g where g.airline_name = '{}' and g.purchase_date >= '{}' and g.purchase_date < '{}' group by year(g.purchase_date), month(g.purchase_date) order by year(g.purchase_date) desc, month(g.purchase_date) desc

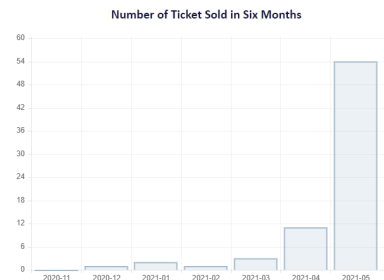
Screenshot:

[Back to Homepage](#) | [View the Customers of a Flight](#) | [Create New Flights](#) | [Add New Airplanes](#) | [Add New Airports](#) | [View all Booking Agents](#) | [View Most Frequent Customers](#) | [Comparison of Revenue earned](#) | [View Top Destinations](#)

Customer: Arthur Logout

Number of Ticket Sold in Six Months: 72

Start Date End Date



(8) Comparison of revenue

Introduction:

The staff can view the percentage of direct sales (customers buy tickets themselves) and indirect sales (customers buy through agents) of total revenue in two pie charts, one for the past month and the other for the past year. Check whether it's really a staff operating.

Query Code:

Select sum(f.price) from purchases p join ticket t using (ticket_id) join flight f using (flight_num)
 where p.booking_agent_id is not null and f.airline_name = '{}' and p.purchase_date >= '{}' and p.purchase_date <= '{}'

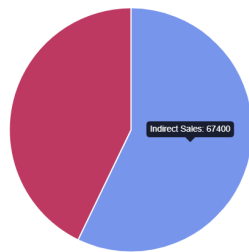
Select sum(f.price) from purchases p join ticket t using (ticket_id) join flight f using (flight_num)
 where p.booking_agent_id is null and f.airline_name = '{}' and p.purchase_date >= '{}' and p.purchase_date <= '{}'

Screenshot:

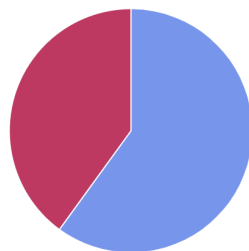
[Back to Homepage](#) | [View the Customers of a Flight](#) | [Create New Flights](#) | [Add New Airplanes](#) | [Add New Airports](#) | [View all Booking Agents](#) | [View Most Frequent Customers](#) | [View Reports](#) | [View Top Destinations](#)

Customer: Arthur [Logout](#)

Revenue Comparison for The Past Month



Revenue Comparison for the Past Year



(9) View top destinations

Introduction:

The staff can view the top 3 destinations in the airline for the past year and for the past 3 months. Check whether it's really a staff operating.

Query Code:

```
SELECT a.airport_city, count(t.ticket_id) from flight f join ticket t using (flight_num)
join purchases p using (ticket_id) join airport a on f.arrival_airport = a.airport_name
where f.airline_name = '{}' and p.purchase_date >= '{}' and p.purchase_date <= '{}'
group by a.airport_city order by count(distinct t.ticket_id) desc limit 3
```

Screenshot:

Top 3 Popular Destinations for The Past three Months



Top 3 Popular Destinations for the Past Year



1.5 Advanced Functions

(1) Security: Invalid User Identity Check

Introduction:

Users are not allowed to visit pages of other identities by typing in the url. For instance, a customer cannot visit an agent's pages by typing in the url link of agents. Every such behavior will direct the user to the index page (public search page) and trigger an error message "Invalid Identity. Please return to your own page."

Pseudo Code:

```
if identity == "agent":
```

```
else:
```

```
    (Code that fetches data for the index page).
```

```
    return render_template('index.html', data=data, error = "Invalid Identity. Please return.")
```

Screenshot:

[Login](#) | [Register](#)
Error: Invalid Identity. Please return.

Start Date: 年 / 月 / 日 ☐ End Date: 年 / 月 / 日 ☐ Departure City: None Arrival City: None
 Flight Number:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airpprt	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10203	Chicago	CBA	2021-08-25 16:30:00	Berlin	NBA	2021-08-25 19:30:00	1400	upcoming	12334
China Eastern	10205	Berlin	NBA	2021-11-26 17:00:00	Shanghai	PVG	2021-11-26 20:30:00	300	upcoming	12334
China Eastern	10201	Chongqin	CQA	2021-12-25 16:00:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqin	CQA	2021-12-26 20:30:00	200	upcoming	12334
China Eastern	10204	Chongqin	CQA	2022-12-25 16:00:00	Chengdu	SLA	2022-12-25 19:15:00	1700	upcoming	10086

(2) Security: Log-out Users cannot Go Back to Log-in Pages

Introduction:

Once a user has logged out, the user cannot return to the previous logged-in pages by clicking on the "return" button in the browser. Every such behavior will force the user to stay in the index page (public search page) and trigger an error message "Invalid Identity. Please log in again."

Pseudo Code:

```
try:
```

```
username = session['username']
identity = session["identity"]
```

except:

(Code that fetches data for the index page).

```
return render_template('index.html', data=data, error = "Invalid Identity. Please log in again.")
```

Screenshot:

[Login](#) | [Register](#)

Error: You have logged out. Please log in again.

Start Date 年 / 月 / 日 ☐ End Date 年 / 月 / 日 ☐ Departure City None Arrival City None

Flight Number

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10203	Chicago	CBA	2021-08-25 16:30:00	Berlin	NBA	2021-08-25 19:30:00	1400	upcoming	12334
China Eastern	10205	Berlin	NBA	2021-11-26 17:00:00	Shanghai	PVG	2021-11-26 20:30:00	300	upcoming	12334
China Eastern	10201	Chongqin	CQA	2021-12-25 19:15:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqin	CQA	2021-12-26 20:30:00	200	upcoming	12334
China Eastern	10204	Chongqin	CQA	2022-12-25 16:00:00	Chengdu	SLA	2022-12-25 19:15:00	1700	upcoming	10086

(3) Drop Down Search Box: Search Convenience for Users

Introduction:

In each search box, a drop down box will appear, listing all possible options to the users. When the user types in any letters, the drop down box will update automatically to filter out all options containing the letters. For instance, the departure city search box will display the drop down box listing all cities contained in the database. This function will save users time and effort in searching for flights.

Pseudo Code:

- (1. Fetch all cities/flight numbers from database.)
- (2. Transfer the cities/flight numbers data to the html through render_template.)
- (3. In html, create a datalist function for each search box which presents all cities/flight numbers in the transferred data.)

Screenshot:

Purchase Tickets | My Spending

Start Date 年 / 月 / 日 ☐ End Date 年 / 月 / 日 ☐ Departure City Arrival City None

Customer: wpy@163.com [Logout](#)

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086
China Eastern	10099	NYC	JFK	2021-07-25 16:00:00	Shanghai	PVG	2021-07-25 19:00:00	1000	upcoming	10086
China Eastern	10206	Berlin	NBA	2021-08-25 16:30:00	Shanghai	PVG	2021-11-26 20:30:00	300	upcoming	12334
China Eastern	10201	Chongqin	CQA	2021-12-25 19:15:00	Chengdu	SLA	2021-12-25 19:15:00	1800	upcoming	10086
China Eastern	10202	Berlin	NBA	2021-12-26 17:00:00	Chongqin	CQA	2021-12-26 20:30:00	200	upcoming	12334
China Eastern	10204	Chongqin	CQA	2022-12-25 16:00:00	Chengdu	SLA	2022-12-25 19:15:00	1700	upcoming	10086

(4) Purchase Check: All past flights not allowed to be bought

Introduction:

For all past flights, no buy button will be attached to them. Instead, a message “Stop selling tickets” will be displayed.

Pseudo Code:

- (1. Fetch all past flight numbers from database.)
- (2. Transfer the flight numbers data to the html through render_template.)
- (3. In html table, for each flight with past flight number, display no buy buttons and show the message “Stop selling tickets”).)

Screenshot:

My Flights | My Seending

Departure City Arrival City Start Date End Date Search

Customer: wpy@163.com Logout

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10000	NYC	JFK	2020-12-25 16:00:00	Shanghai	PVG	2020-12-25 19:00:00	1200	delayed	10086	Stop selling tickets.
China Eastern	10200	NYC	JFK	2020-12-25 16:30:00	Shanghai	PVG	2020-11-25 19:30:00	1200	delayed	12334	Stop selling tickets.

(5) Purchase Check: Invalid customer email not allowed to be bought

Introduction:

For booking agents, purchase behaviors are not allowed if input customer emails are invalid. Instead, a message “Error: Invalid Customer Email” will be displayed.

Pseudo Code:

- Check if the input customer email exists in the database.)
- Generate an error message to the html.)

Screenshot:

My Flights | My Commissions | Top Customers

Departure City Arrival City Start Date End Date Search

Agency ID: 12344 Logout

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	customer_email <input type="text"/> Submit
											Error: Invalid Customer Email.
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086	customer_email <input type="text"/> Submit

(6) Flight Status Update Check: Inform the staff of whether the status is updated or not

Introduction:

For staff, if the status input is valid, a message showing the flight status successfully updated will be displayed. If the status is invalid, a message showing the flight status invalid will be displayed and the staff can type in the new status to keep on updating the status.

Pseudo Code:

- Check if the input status exists in the database. If yes, update the database.)
- Generate a message to the html.)

Screenshot:

View the Customers of a Flight | Create New Flights | Add New Airplanes | Add New Airports | View all Booking Agents | View Most Frequent Customers | View Reports | Comparison of Revenue Earned | View Top Destinations

Start Date End Date Departure City Arrival City Search

Staff: Arthur Logout

Last updated flight:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	new status <input type="text"/> Submit
											Flight status successfully updated.

Regular flight:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	new status <input type="text"/> Submit
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086	new status <input type="text"/> Submit
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086	new status <input type="text"/> Submit
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334	new status <input type="text"/> Submit

View the Customers of a Flight | Create New Flights | Add New Airplanes | Add New Airports | View all Booking Agents | View Most Frequent Customers | View Reports | Comparison of Revenue Earned | View Top Destinations

Start Date End Date Departure City Arrival City Search

Staff: Arthur Logout

Last updated flight:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	new status <input type="text"/> Submit
											Error: Invalid Status Input.

Regular flight:

Airline Name	Flight Number	Departure City	Departure Airport	Departure Time	Arrival City	Arrival Airport	Arrival Time	Price	Status	Airplane ID	Action
China Eastern	10103	Shanghai	PVG	2021-05-26 17:00:00	NYC	JFK	2021-05-26 20:00:00	1300	upcoming	12334	new status <input type="text"/> Submit
China Eastern	12321	Shanghai	PVG	2021-06-11 17:05:00	NYC	JFK	2021-06-11 19:05:00	3000	upcoming	10086	new status <input type="text"/> Submit
China Eastern	12322	Shanghai	PVG	2021-06-11 18:05:00	NYC	JFK	2021-06-11 19:05:00	2500	upcoming	10086	new status <input type="text"/> Submit
China Eastern	10206	Chicago	CBA	2021-06-15 16:30:00	NYC	JFK	2021-06-15 19:30:00	1300	upcoming	12334	new status <input type="text"/> Submit

2. Team Members and Responsibilities

Tasks	Team Member In Charge
-------	-----------------------

Public Search (code+html)	Peiyang Wen
Register (code+html)	Yuxuan Wang
Login (code+html)	Peiyang Wen
Staff (code+html)	Yuxuan Wang
Customer (code+html)	Peiyang Wen
Agent (code+html)	Peiyang Wen
Advanced Functions	Yuxuan Wang, Peiyang Wen