**CSCI-SHU 213 Databases**

**Final Project – Use Cases Explanation**

Please see below for the queries for each use case.

Note: Everything inside curly bracket {} is a variable!

Public

1. View Public Info

* Look up all info of upcoming flights

select \* from flight where status = 'Upcoming'

1. Register

* Check if email inputted is already registered

select \* from customer where email = '{email}'

select \* from booking\_agent where email = '{email}'

select \* from airline\_staff where email = '{username}'

* Insertion for successful registration

insert into customer values ('{email}', '{name}', 'MD5{password}', {building\_number}, '{street}', '{city}', '{state}', {phone\_number}, '{passport\_number}', '{passport\_expiration}', '{passport\_country}', '{date\_of\_birth}')

insert into booking\_agent values ('{email}', 'MD5{password}', {booking\_agent\_id})

insert into customer values ('{username}', 'MD5{password}', {first\_name}, '{last\_name}', '{date\_of\_birth}', ‘{airline\_name}’)

1. Login

* Check if email inputted is already registered

select \* from customer where email = '{email}'

select \* from booking\_agent where email = '{email}'

select \* from airline\_staff where email = '{username}'

Customer Use Cases

1. View My Flights

* Select all upcoming purchased flights for user

select distinct \* from flight natural join purchases natural join ticket where purchases.customer\_email = '{customer}' and flight.status = 'Upcoming'

1. Purchase Tickets

* Find flights with inputted flight number and airline. Also to check if the customer has bought the selected ticket before.

select \* from ticket natural join purchases where flight\_num={flight\_num} and airline\_name='{airline\_name}'

* Check if there is available seats for selected flight

select seats from flight natural join airplane where flight.flight\_num = {flight\_num}

* Update seat (subtract by 1) number if there is available seat

update flight natural join airplane set seats = {int(avail\_seats)-1} where flight.flight\_num = {flight\_num}

* Get current maximum number of ticket id

select max(ticket\_id) as max from ticket

* Insert into ticket

insert into ticket values ({new\_id}, '{airline\_name}', {flight\_num})

* Insert into purchases

insert into purchases values ({new\_id}, '{customer}', null, '{today}')

1. Search For Flights

* Select all upcoming flights if input value is none

select \* from flight where status = 'Upcoming'

* Select filtered results if input value is not none

select \* from flight where departure\_airport= "{departure}" and arrival\_airport= "{arrival}" and date(departure\_time)=’{date}’

1. Track My Spending

* Get the customer email and the sum of all tickets purchased in the past year if input value is none

select customer\_email, sum(price) from purchases natural join ticket natural join flight where customer\_email = '{email}' and purchase\_date between date\_sub(date(now()), interval 1 year) and date(now())

* Get the total spending for each year and date for the past 6 months

select year(date\_sub(date(now()), interval {} month)) as year, month(date\_sub(date(now()), interval {} month)) as month, sum(price) from ticket natural join purchases natural join flight where customer\_email= \'{}\' AND year(purchase\_date) = year(date\_sub(date(now()), interval {} month)) and month(purchase\_date)= month(date\_sub(date(now()), interval {} month))

* Get the sum of all tickets purchased in the past year if input value is not none

select sum(flight.price) as spending, month(purchase\_date) as month from purchases natural join ticket natural join flight where customer\_email='{email}' and (purchase\_date between date('{start\_year}-{start\_month}-01') and date('{end\_year + 1 if end\_month + 1 > 12 else end\_year}-{end\_month + 1 if end\_month + 1 <= 12 else 1}-01')) group by month

Booking Agent Use Cases

1. View My Flights

* Get all upcoming flights bought by the booking agent for a customer

select distinct \* from flight natural join purchases natural join ticket where purchases.booking\_agent\_id = {agent\_id} and flight.status = 'Upcoming'

1. Purchase Tickets

* Get the airline name of the booking agent

select airline\_name from booking\_agent\_work\_for where email = '{email}'

* Get the ID of the booking agent

select booking\_agent\_id from booking\_agent where email = '{email}'

* Check if customer exists in the database

select \* from customer where email=\'{customer}\'

* Find the flights and check if the customer has bought the selected ticket before.

select \* from ticket natural join purchases where flight\_num={flight\_num} and airline\_name='{airline\_name}' and customer\_email='{customer}'

* Check if there is available seats for selected flight

select seats from flight natural join airplane where flight.flight\_num = {flight\_num}

* Update seat (subtract by 1) number if there is available seat

update flight natural join airplane set seats = {int(avail\_seats)-1} where flight.flight\_num = {flight\_num}

* Get current maximum number of ticket id

select max(ticket\_id) as max from ticket

* Insert into ticket

insert into ticket values ({new\_id}, '{airline\_name}', {flight\_num})

* Insert into purchases

insert into purchases values ({new\_id}, '{customer}', {booking\_agent\_id}, '{today}')

1. Search For Flights

* Select all upcoming flights if input value is none

select \* from flight where status = 'Upcoming'

* Select all upcoming flights if input value is not none

select \* from flight where status = 'Upcoming' and {criteria} = '{value}'

1. View My Commission

* Assuming the commission is 10% of ticket price, get the sum of commissions from the past 30 days

select sum(flight.price)\*0.1 as total\_commission, (sum(flight.price)\*0.1)/count(ticket.flight\_num) as average\_commission, count(ticket.flight\_num) as number\_of\_tickets from purchases natural join booking\_agent natural join flight natural join ticket where booking\_agent.email = '{agent\_email}' and purchases.purchase\_date >= adddate(date(now()), interval -30 day)

* Assuming the commission is 10% of ticket price, get the sum of commissions from and until two selected days

select sum(flight.price)\*0.1 as total\_commission, (sum(flight.price)\*0.1)/count(ticket.flight\_num) as average\_commission, count(ticket.flight\_num) as number\_of\_tickets from purchases natural join booking\_agent natural join flight natural join ticket where booking\_agent.email = '{agent\_email}' and purchases.purchase\_date > '{start\_date}' and purchases.purchase\_date < '{end\_date}'

1. View Top Customers

* Get customer’s name

select name from customer where email = '{cust\_email}'

* Select top 5 customers based on number of ticket sales in the past 6 months

select customer\_email, count(\*) as ticket\_count from purchases natural join booking\_agent where booking\_agent.email = '{agent\_email}' and purchases.purchase\_date >= adddate(date(now()), interval -6 month) group by customer\_email order by count(\*) desc limit 5

* Select top 5 customers based on commissions in the past 6 months

select customer\_email, sum(flight.price)\*0.1 as total\_commission from purchases natural join booking\_agent natural join flight natural join ticket where booking\_agent.email = '{agent\_email}' and purchases.purchase\_date >= adddate(date(now()), interval -6 month) group by customer\_email order by total\_commission desc limit 5

Airline Staff Use Cases

1. View My Flights

* Get all flight info with specified input values

select \* from flight where airline\_name='{staff\_airline}' and arrival\_airport like '{arrival}' and departure\_airport like '{departure}' and (departure\_time between {startDate} AND {endDate})

1. Create New Flight

* Check if the staff has admin permission

select permission\_type from permission where username = '{staff\_username}' and permission\_type='Admin'

* Insert all flight info into flight table

insert into flight values('{crit\_value[0]}', {crit\_value[1]}, '{crit\_value[2]}', '{crit\_value[3]}', '{crit\_value[4]}', '{crit\_value[5]}', {crit\_value[6]}, '{crit\_value[7]}', {crit\_value[8]})

1. Change Status of Flights

* Check if the staff has operator permission

select permission\_type from permission where username = '{staff\_username}' and permission\_type='Operator'

* Update flight status in flight table

update flight set status = '{new\_status}' where airline\_name = '{staff\_airline}' and flight\_num = '{flight\_num}' and airline\_name='{staff\_airline}'

1. Add airplanes

* Check if the staff has admin permission

select permission\_type from permission where username = '{staff\_username}' and permission\_type='Admin'

* Insert new airplane info into airplane table

insert into airplane values('{staff\_airline}', '{airplane\_id}', '{seats}')

1. Add new airport

* Check if the staff has admin permission

select permission\_type from permission where username = '{staff\_username}' and permission\_type='Admin'

* Insert new airport info into airport table

insert into airport values('{airport\_name}', '{airport\_city}')

1. View all the booking agents

* Select top 5 booking agents based on number of ticket sales in the past month

select booking\_agent.email, count(ticket.ticket\_id) as number\_of\_ticket\_sales from booking\_agent natural join ticket natural join purchases where airline\_name = '{staff\_airline}' and purchases.purchase\_date >= adddate(date(now()), interval -1 month) group by booking\_agent.email order by number\_of\_ticket\_sales desc limit 5

* Select top 5 booking agentes based on commission in the past year

select booking\_agent.email, 0.1\*sum(flight.price) as commission from booking\_agent natural join flight natural join ticket natural join purchases where airline\_name = '{staff\_airline}' and purchases.purchase\_date >= adddate(date(now()), interval -12 month) group by booking\_agent.email order by commission desc limit 5

1. View Frequent Customers

* Select top 5 customers based on number of tickets bought in the past year

select customer\_email, count(ticket.ticket\_id) as tickets\_bought from purchases natural join ticket where airline\_name = '{staff\_airline}' and purchases.purchase\_date >= adddate(date(now()), interval -12 month) group by customer\_email order by tickets\_bought desc limit 5

* Select all flights a particular Customer has taken with the airline in the past year

select customer\_email, flight\_num from purchases natural join ticket where flight\_num ='{flight\_number}'

1. View report

* Select the number of tickets sold for each year and month in the past year if start date and end date is none

select year(date\_sub(date(now()), interval {month} month)) as year, month(date\_sub(date(now()), interval {month} month)) as month, count(\*) as tickets\_sold from purchases natural join ticket where ticket.airline\_name = '{staff\_airline}' and year(purchase\_date) = year(date\_sub(date(now()), interval {month} month)) and month(purchase\_date) = month(date\_sub(date(now()), interval {month} month))"

* Select the number of tickets sold for each year and month given the start date and end date

select year(date\_sub(date('{end\_date}'), interval {month} month)) as year, month(date\_sub(date('{end\_date}'), interval {month} month)) as month, count(\*) as tickets\_sold from purchases natural join ticket where ticket.airline\_name = '{staff\_airline}' and year(purchase\_date) = year(date\_sub(date('{end\_date}'), interval {month} month)) and month(purchase\_date) = month(date\_sub(date('{end\_date}'), interval {month} month))

1. Comparison of Revenue Earned

* Get the amount of sales without a booking agent in the past month

select sum(flight.price) as sales from flight natural join purchases natural join ticket where ticket.airline\_name = '{staff\_airline}' and purchases.booking\_agent\_id is null and purchase\_date >= adddate(date(now()), interval -30 day)

* Get the amount of sales with a booking agent in the past month

select sum(flight.price) as sales from flight natural join purchases natural join ticket where ticket.airline\_name = '{staff\_airline}' and purchases.booking\_agent\_id is not null and purchase\_date >= adddate(date(now()), interval -30 day)

* Get the amount of sales without a booking agent in the past year

select sum(flight.price) as sales from flight natural join purchases natural join ticket where ticket.airline\_name = '{staff\_airline}' and purchases.booking\_agent\_id is null and purchase\_date >= adddate(date(now()), interval -12 month)

* Get the amount of sales with a booking agent in the past year

select sum(flight.price) as sales from flight natural join purchases natural join ticket where ticket.airline\_name = '{staff\_airline}' and purchases.booking\_agent\_id is not null and purchase\_date >= adddate(date(now()), interval -12 month)

1. View top destinations

* Select top 3 destinations for the last 3 months

select arrival\_airport from flight natural join purchases where flight.airline\_name = '{staff\_airline}' and purchase\_date >= adddate(date(now()), interval -3 month) group by flight.arrival\_airport order by count(\*) desc limit 3

* Select top 3 destinations for the last year

select arrival\_airport from flight natural join purchases where flight.airline\_name = '{staff\_airline}' and purchase\_date >= adddate(date(now()), interval -12 month) group by flight.arrival\_airport order by count(\*) desc limit 3

1. Grant new permission

* Check if the staff has admin permission

select permission\_type from permission where username = '{staff\_username}' and permission\_type='Admin'

* Check if other\_username (other staff) is in the same airline

select airline\_name from airline\_staff where username = '{other\_username}'

* Grant the other staff a new permission

insert into permission values ('{other\_username}', '{new\_permission}')

* Project the updated list of airline staff with their permission

select username, permission\_type from airline\_staff natural join permission where airline\_name = '{staff\_airline}'

1. Add booking agents

* Check if the staff has admin permission

select permission\_type from permission where username = '{staff\_username}' and permission\_type='Admin'

* Check if the inputted booking agent email is already in the database

select \* from booking\_agent where email='{agent\_email}'

* Add booking agent info into booking\_agent\_work\_for table

insert into booking\_agent\_work\_for values ('{agent\_email}','{staff\_airline}')

Additional Queries

1. Search Customer by email

select \* from customer where email like '{email}%'

1. Search airline staff by username

select \* from airline\_staff where username like '{username}%'

1. Get permission of airline staff

select permission\_type from permission where username='{username}'

1. Get airplane info of a particular flight

select \* from airplane where airline\_name='{airline\_name}'