

PROJECT PART -II

PROJECT REPORT

COURSE: PRINCIPLES OF DATABASE
MANAGEMENT SYSTEMS

SECTION: B

SUBMISSION DATE: 12-18-2021

STUDENT NAME: Sai Kiran Uriti

STUDENT ID: su2092

STUDENT NAME: Utkarsh Saboo

STUDENT ID: US452

TABLE OF CONTENTS

- 1) SUMMARY AND ASSUMPTIONS
- 2) LOGICAL MODEL
- 3) RELATIONAL MODEL
- 4) DDL CODE
- 5) ABOUT OUR PROJECT
- 6) TABLE SCREENSHOTS
- 7) PROJECT EXECUTION
- 8) SECURITY FEATURES
- 9) OUR LEARNINGS AND EXPERIENCE

1) SUMMARY AND ASSUMPTIONS:

Bussiness case: SAME is a company which offers insurances to domestic and international passengers. Initially they relied on manually entering the data and all. However, due to the growth of the business, they need to implement a database to store the data and actively make necessary changes whenever needed.

So, we have implemented a database model and made sure to implement all the requested features and data.

In our model there is a ternary relationship between passenger, insurance and flight tables. That is because one passenger can have multiple insurances(he could purchase a life insurance and a lost luggage insurance) and 1 insurance can be taken by many customers. 1 passenger can have multiple flights(connecting flights with different flight numbers) and 1 flight can have multiple passengers. Finally 1 insurance can be had by people on many flights and many flights can fly people with the same insurance.

Most importantly, we have created a separate table for agents. And that has a one to many relationship with the customer_agent subtype in ussk_customer. We do that because we realize that tens of thousands of people make the booking and lots of bookings will have the same agents. Thus it would be a lot of repeated work to write down the agent_name, agent_address, agent_email for each row in the customer_agent table. In order to remove that redundancy we create separate agent table(master table) which will store all the agent details only once and pass the agentID to the customer_agent as foreign key.

The ternary intersect entity between the three entities mentioned above has the cabin class, meal plan and special request as its attributes. Thus we will always have a record for a passenger travelling on a flight with all of their preferences(including no insurance and no special request).

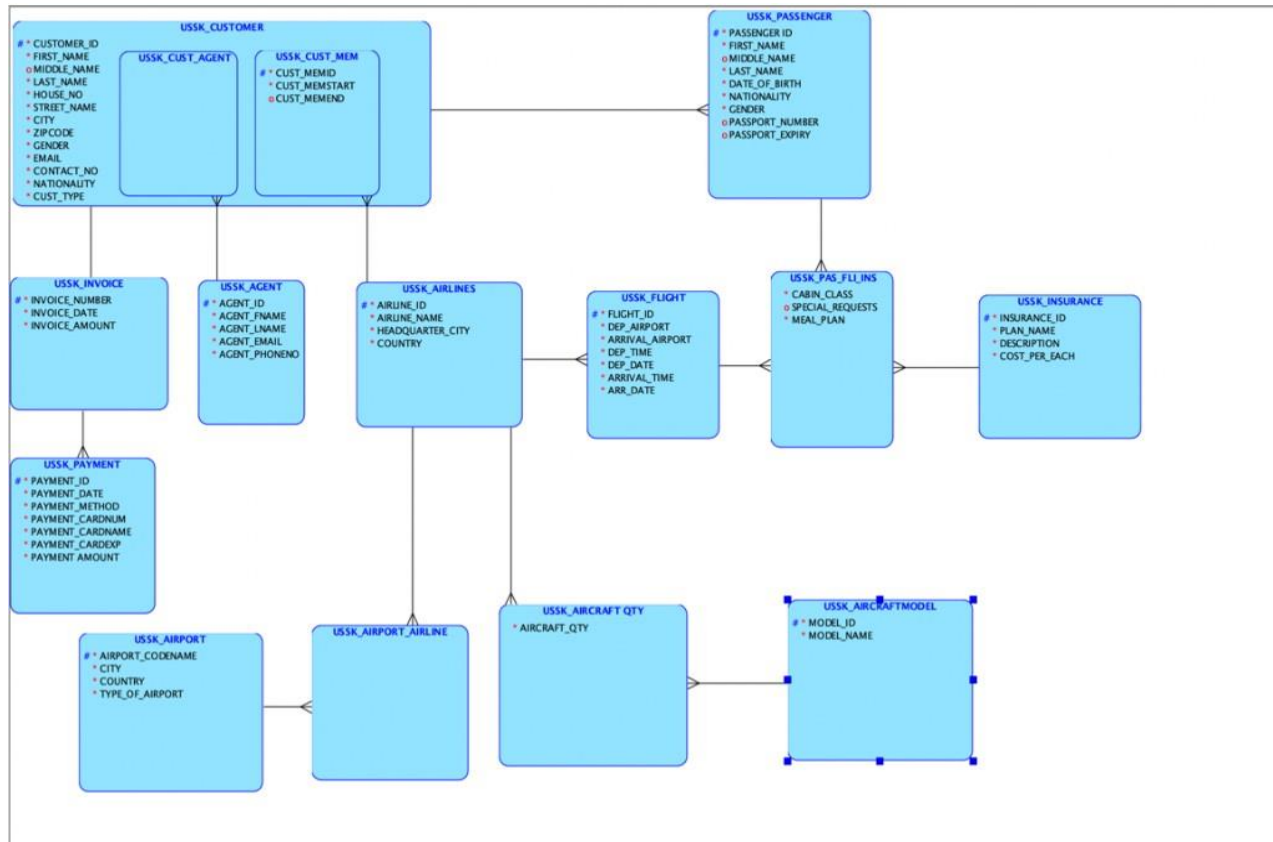
The default insurance value is Null, that is because we do not want to lose the record of a passenger who does not have insurance.

Each customer will only have one invoice associated with their booking, as the amount has to be paid in full. Furthermore 1 invoice can have many payments associated with it, since the customer is allowed to pay using multiple cards, credit or debit.

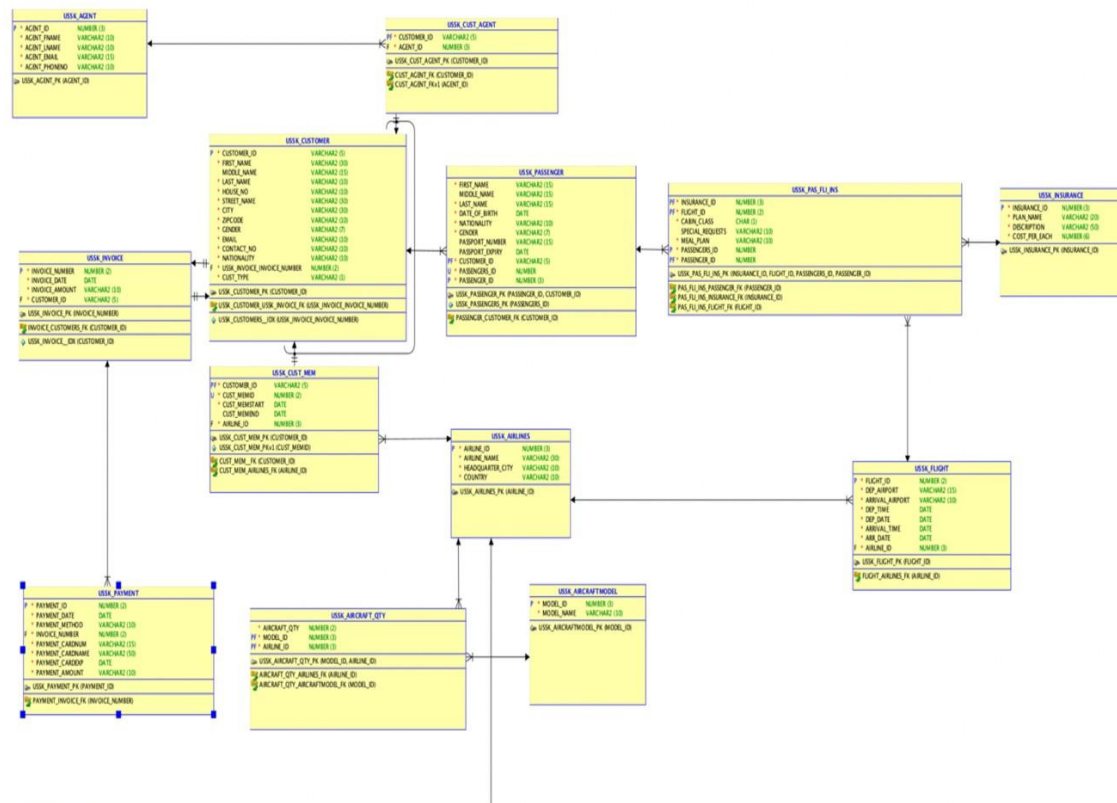
A passenger may or may not have a passport, since we have not specified whether the flight is exclusively international, a passenger may travel domestically with their domestic ID. Airlines and airports have a many to many relationship as 1 airport can host many airlines and 1 airline can be flying at many airports.

Database is very important for any company. Therefore, implementing this project will definitely benefit the company especially in a long run. There are certain features like limiting the viewers to just only read the data, update the data of a particular group of passengers. This saves lot of time. Therefore, implementing this database is very important.

2) LOGICAL MODEL



3) RELATIONAL MODEL



4) DDL CODE

```
CREATE TABLE ussk_agent (  
agent_id NUMBER(3) NOT NULL, agent_fname VARCHAR2(10) NOT NULL, agent_lname VARCHAR2(10) NOT NULL,  
agent_email VARCHAR2(15) NOT NULL, agent_phoneno VARCHAR2(10) NOT NULL  
);  
  
COMMENT ON COLUMN ussk_agent.agent_id IS 'ID OF THE AGENT';  
  
COMMENT ON COLUMN ussk_agent.agent_fname IS 'FIRST NAME OF THE AGENT';  
  
COMMENT ON COLUMN ussk_agent.agent_lname IS 'LAST NAME OF THE AGENT';  
  
COMMENT ON COLUMN ussk_agent.agent_email IS 'AGENT'S EMAIL ID';  
  
COMMENT ON COLUMN ussk_agent.agent_phoneno IS 'CONTACT NUMBER OF THE AGENT';  
  
ALTER TABLE ussk_agent ADD CONSTRAINT ussk_agent_pk PRIMARY KEY ( agent_id );  
  
CREATE TABLE ussk_aircraft_qty ( aircraft_qty NUMBER(2) NOT NULL, model_id  NUMBER(3)  NOT  NULL, airline_id  
NUMBER(3) NOT NULL  
);  
  
COMMENT ON COLUMN ussk_aircraft_qty.aircraft_qty IS 'NUMBER OF AIRCRAFTS IN THE FLEET';  
  
ALTER TABLE ussk_aircraft_qty ADD CONSTRAINT ussk_aircraft_qty_pk PRIMARY KEY ( model_id,  
airline_id );  
  
CREATE TABLE ussk_aircraftmodel ( model_id NUMBER(3) NOT NULL, model_name VARCHAR2(10) NOT NULL  
);  
  
COMMENT ON COLUMN ussk_aircraftmodel.model_id IS 'MODEL ID OF THE AIRCRAFT';  
  
COMMENT ON COLUMN ussk_aircraftmodel.model_name IS 'NAME OF THE AIRCRAFT MODEL';  
  
ALTER TABLE ussk_aircraftmodel ADD CONSTRAINT ussk_aircraftmodel_pk PRIMARY KEY ( model_id );  
  
CREATE TABLE ussk_airlines (  
airline_id      NUMBER(3) NOT NULL, airline_name      VARCHAR2(30)   NOT   NULL,   headquarter_city  
VARCHAR2(10) NOT NULL, country      VARCHAR2(10) NOT NULL  
);  
  
COMMENT ON COLUMN ussk_airlines.airline_id IS 'ID OF THE AIRLINE';  
  
COMMENT ON COLUMN ussk_airlines.airline_name IS 'NAME OF THE AIRLINE';  
  
COMMENT ON COLUMN ussk_airlines.headquarter_city IS 'CITY OF AIRLINES HEADQUARTERS';  
  
COMMENT ON COLUMN ussk_airlines.country IS 'COUNTRY OF THE AIRLINES';  
  
ALTER TABLE ussk_airlines ADD CONSTRAINT ussk_airlines_pk PRIMARY KEY ( airline_id );
```

```
CREATE TABLE ussk_airport ( airport_codename VARCHAR2(5) NOT NULL, city   VARCHAR2(15) NOT NULL, country
                           VARCHAR2(10) NOT NULL,
type_of_airport VARCHAR2(10) NOT NULL
```

```
);
```

```
COMMENT ON COLUMN ussk_airport.airport_codename IS 'CODENAME OF THE AIRPORT';
```

```
COMMENT ON COLUMN ussk_airport.city IS 'CITY OF THE AIRPORT';
```

```
COMMENT ON COLUMN ussk_airport.country IS 'AIRPORTS COUNTRY';
```

```
COMMENT ON COLUMN ussk_airport.type_of_airport IS 'TYPE OF THE AIRPORT';
```

```
ALTER TABLE ussk_airport ADD CONSTRAINT ussk_airport_pk PRIMARY KEY ( airport_codename );
```

```
CREATE TABLE ussk_airport_airline ( ussk_airlines_airline_id      NUMBER(3)          NOT          NULL,
ussk_airport_airport_codename VARCHAR2(5) NOT NULL
);
```

```
ALTER TABLE ussk_airport_airline ADD CONSTRAINT ussk_airport_airline_pk PRIMARY KEY ( ussk_airlines_airline_id,
ussk_airport_airport_codename );
```

```
CREATE TABLE ussk_cust_agent ( customer_id VARCHAR2(5) NOT NULL, agent_id NUMBER(3) NOT NULL
);
```

```
COMMENT ON COLUMN ussk_cust_agent.customer_id IS 'ID OF THE CUSTOMER';
```

```
ALTER TABLE ussk_cust_agent ADD CONSTRAINT ussk_cust_agent_pk PRIMARY KEY ( customer_id );
```

```
CREATE TABLE ussk_cust_mem ( customer_id VARCHAR2(5) NOT NULL, cust_memid      NUMBER(2)  NOT   NULL,
cust_memstart DATE NOT NULL, cust_memend DATE,
airline_id      NUMBER(3) NOT NULL
```

```
);
```

```
COMMENT ON COLUMN ussk_cust_mem.customer_id IS 'ID OF THE CUSTOMER';
```

```
COMMENT ON COLUMN ussk_cust_mem.cust_memid IS 'MEMBERSHIP ID OF THE CUSTOMER';
```

```
COMMENT ON COLUMN ussk_cust_mem.cust_memstart IS 'CUSTOMERS MEMBERSHIPS START DATE';
```

```
COMMENT ON COLUMN ussk_cust_mem.cust_memend IS 'CUSTOMER"S MEMBERSHIP"S END DATE';
```



```

ALTER TABLE ussk_cust_mem ADD CONSTRAINT ussk_cust_mem_pk PRIMARY KEY ( customer_id );

ALTER TABLE ussk_cust_mem ADD CONSTRAINT ussk_cust_mem_pkv1 UNIQUE ( cust_memid );

CREATE TABLE ussk_customer (

customer_id      VARCHAR2(5) NOT NULL,

first_name       VARCHAR2(30) NOT NULL, middle_name          VARCHAR2(15),
last_name        VARCHAR2(10) NOT NULL,

house_no         VARCHAR2(10) NOT NULL,

street_name      VARCHAR2(30) NOT NULL, city VARCHAR2(30) NOT NULL, zipcode
                 VARCHAR2(10) NOT NULL,
gender  VARCHAR2(7) NOT NULL,

email   VARCHAR2(10) NOT NULL,

contact_no      VARCHAR2(10) NOT NULL,

nationality     VARCHAR2(10) NOT NULL, ussk_invoice_invoice_number  NUMBER(2) NOT NULL, cust_type
                 VARCHAR2(1) NOT NULL
);


ALTER TABLE ussk_customer

ADD CONSTRAINT ch_inh_ussk_customer CHECK ( cust_type IN ( 'A', 'D', 'M' ) );

COMMENT ON COLUMN ussk_customer.customer_id IS 'ID OF THE CUSTOMER';

COMMENT ON COLUMN ussk_customer.first_name IS 'CUSTOMERS FIRST NAME';

COMMENT ON COLUMN ussk_customer.middle_name IS 'CUSTOMERS MIDDLE NAME';

COMMENT ON COLUMN ussk_customer.last_name IS 'CUSTOMERS LAST NAME';

COMMENT ON COLUMN ussk_customer.house_no IS 'CUSTOMERS HOUSE NO';

COMMENT ON COLUMN ussk_customer.street_name IS 'NAME OF THE CUSTOMERS STREET';

COMMENT ON COLUMN ussk_customer.city IS 'CUSTOMERS CITY';

COMMENT ON COLUMN ussk_customer.zipcode IS 'CUSTOMERS AREA ZIPCODE';

COMMENT ON COLUMN ussk_customer.gender IS 'CUSTOMERS GENDER TYPE';

COMMENT ON COLUMN ussk_customer.email IS 'CUSTOMERS EMAIL';

COMMENT ON COLUMN ussk_customer.contact_no IS 'CUSTOMERS CONTACT NO';

COMMENT ON COLUMN ussk_customer.nationality IS 'CUSTOMERS NATIONALITY';

COMMENT ON COLUMN ussk_customer.cust_type IS 'DISCRIMINATOR FOR THE TYPE OF CUSTOMER';

CREATE UNIQUE INDEX ussk_customers_idx ON ussk_customer (
ussk_invoice_invoice_number ASC );

ALTER TABLE ussk_customer ADD CONSTRAINT ussk_customer_pk PRIMARY KEY ( customer_id );

```

```
CREATE TABLE ussk_flight (  
    flight_id NUMBER(2) NOT NULL, dep_airport VARCHAR2(15) NOT NULL, arrival_airport  
    VARCHAR2(10) NOT NULL, dep_time DATE NOT NULL,  
    dep_date DATE NOT NULL, arrival_time DATE NOT NULL, arr_date DATE NOT  
    NULL, airline_id NUMBER(3) NOT NULL  
);
```

```
COMMENT ON COLUMN ussk_flight.flight_id IS 'FLIGHT ID NUMBER';
```

```
COMMENT ON COLUMN ussk_flight.dep_airport IS 'NAME OF THE DEPARTURE AIRPORT';
```

```
COMMENT ON COLUMN ussk_flight.arrival_airport IS 'NAME OF THE ARRIVAL AIRPORT';
```

```
COMMENT ON COLUMN ussk_flight.dep_time IS  
'DEPARTURE TIME';
```

```
COMMENT ON COLUMN ussk_flight.dep_date IS 'TIME ZONE OF THE DEPARTURE ';
```

```
COMMENT ON COLUMN ussk_flight.arrival_time IS 'TIME OF ARRIVAL';
```

```
COMMENT ON COLUMN ussk_flight.arr_date IS 'TIMEZONE OF ARRIVAL';
```

```
ALTER TABLE ussk_flight ADD CONSTRAINT ussk_flight_pk PRIMARY KEY ( flight_id );
```

```
CREATE TABLE ussk_insurance ( insurance_id NUMBER(3) NOT NULL, plan_name VARCHAR2(20) NOT NULL,  
description VARCHAR2(50) NOT NULL, cost_per_each NUMBER(6) NOT NULL  
);
```

```
COMMENT ON COLUMN ussk_insurance.insurance_id IS 'INSURANCE ID NUMBER';
```

```
COMMENT ON COLUMN ussk_insurance.plan_name IS 'INSURANCE PLAN NAME';
```

```
COMMENT ON COLUMN ussk_insurance.description IS 'INSURANCE DESCRIPTION ';
```

```
COMMENT ON COLUMN ussk_insurance.cost_per_each IS 'PRICE OF THE INSURANCE PER PERSON';
```

```
ALTER TABLE ussk_insurance ADD CONSTRAINT ussk_insurance_pk PRIMARY KEY ( insurance_id );
```

```
CREATE TABLE ussk_invoice ( invoice_number NUMBER(2) NOT NULL, invoice_date DATE NOT NULL,
invoice_amount VARCHAR2(10) NOT NULL, customer_id  VARCHAR2(5) NOT NULL
);
```

```
COMMENT ON COLUMN ussk_invoice.invoice_number IS 'INVOICE ID';
```

```
COMMENT ON COLUMN ussk_invoice.invoice_date IS 'DATE OF THE INVOICE';
```

```
COMMENT ON COLUMN ussk_invoice.invoice_amount IS 'AMOUNT IN THE INVOICE';
```

```
CREATE UNIQUE INDEX ussk_invoice_idx ON ussk_invoice (
customer_id ASC );
```

```
ALTER TABLE ussk_invoice ADD CONSTRAINT ussk_invoice_pk PRIMARY KEY ( invoice_number );
```

```
CREATE TABLE ussk_pas_fli_ins ( insurance_id          NUMBER(3)          NOT      NULL,      flight_id
                                NUMBER(2) NOT NULL, cabin_class          CHAR(1) NOT NULL, special_requests VARCHAR2(10),
meal_plan          VARCHAR2(10) NOT NULL, passengers_id          NUMBER NOT NULL, passenger_id
                                NUMBER NOT NULL
);
```

```
COMMENT ON COLUMN ussk_pas_fli_ins.cabin_class IS 'PASSENGERS CABIN CLASS';
```

```
COMMENT ON COLUMN ussk_pas_fli_ins.special_requests IS 'PASSENGERS SPECIAL REQUESTS IF THERE ARE
ANY';
```

```
COMMENT ON COLUMN ussk_pas_fli_ins.meal_plan IS 'PASSENGERS MEAL PLAN';
```

```
ALTER TABLE ussk_pas_fli_ins
```

```
ADD CONSTRAINT ussk_pas_fli_ins_pk PRIMARY KEY ( insurance_id,
flight_id, passengers_id, passenger_id );
```

```
CREATE TABLE ussk_passenger (
```

```
first_name          VARCHAR2(15) NOT NULL, middle_name          VARCHAR2(15),
last_name           VARCHAR2(15) NOT NULL, date_of_birth  DATE NOT NULL, nationality
                    VARCHAR2(10) NOT NULL, gender            VARCHAR2(7) NOT NULL,
passport_number VARCHAR2(15), passport_expiry DATE,
customer_id        VARCHAR2(5) NOT NULL,

passengers_id NUMBER NOT NULL, passenger_id  NUMBER(3) NOT NULL
);
```

COMMENT ON COLUMN ussk_passenger.first_name IS 'PASSENGERS FIRST NAME';

COMMENT ON COLUMN ussk_passenger.last_name IS 'PASSENGERS LAST NAME';

COMMENT ON COLUMN ussk_passenger.date_of_birth IS 'DATE OF BIRTH OF THE PASSENGER';

COMMENT ON COLUMN ussk_passenger.nationality IS 'PASSENGERS NATIONALITY';

COMMENT ON COLUMN ussk_passenger.gender IS 'PASSENGERS GENDER TYPE';

COMMENT ON COLUMN ussk_passenger.passport_number IS 'PASSENGERS PASSPORT NUMBER';

COMMENT ON COLUMN ussk_passenger.passport_expiry IS
'PASSENGERS PASSPORT EXPIRY';

COMMENT ON COLUMN ussk_passenger.customer_id IS 'ID OF THE CUSTOMER';

COMMENT ON COLUMN ussk_passenger.passenger_id IS 'ID OF THE PASSENGER';

ALTER TABLE ussk_passenger ADD CONSTRAINT ussk_passengers_pk UNIQUE (passengers_id);

CREATE TABLE ussk_payment (payment_id NUMBER(2) NOT NULL, payment_date DATE NOT NULL,
payment_method VARCHAR2(10) NOT NULL, invoice_number NUMBER(2) NOT NULL, payment_cardnum
VARCHAR2(15) NOT NULL, payment_cardname VARCHAR2(50) NOT NULL, payment_cardexp DATE NOT NULL,
payment_amount VARCHAR2(10) NOT NULL
);

COMMENT ON COLUMN ussk_payment.payment_id IS 'PAYMENT ID NUMBER';

COMMENT ON COLUMN ussk_payment.payment_date IS 'DATE OF THE PAYMENT';

COMMENT ON COLUMN ussk_payment.payment_method IS 'PAYMENT TYPE';

COMMENT ON COLUMN ussk_payment.payment_cardnum IS 'NUMBER ON THE CARD';

COMMENT ON COLUMN ussk_payment.payment_cardname IS 'NAME OF THE PERSON ON THE CARD';

COMMENT ON COLUMN ussk_payment.payment_cardexp IS 'EXPIRY DATE ON THE CARD';

COMMENT ON COLUMN ussk_payment.payment_amount IS 'PAYMENT AMOUNT';

ALTER TABLE ussk_payment ADD CONSTRAINT ussk_payment_pk PRIMARY KEY (payment_id);

ALTER TABLE ussk_aircraft_qty

ADD CONSTRAINT aircraft_qty_aircraftmodel_fk FOREIGN KEY (model_id)

REFERENCES ussk_aircraftmodel (model_id);

ALTER TABLE ussk_aircraft_qty

ADD CONSTRAINT aircraft_qty_airlines_fk FOREIGN KEY (airline_id) REFERENCES ussk_airlines (airline_id);

ALTER TABLE ussk_airport_airline

ADD CONSTRAINT airline_airport_fk FOREIGN KEY (ussk_airport_airport_codename) REFERENCES ussk_airport (airport_codename);

ALTER TABLE ussk_airport_airline

ADD CONSTRAINT airport_airlines_fk FOREIGN KEY (ussk_airlines_airline_id) REFERENCES ussk_airlines (airline_id);

ALTER TABLE ussk_cust_agent

ADD CONSTRAINT cust_agent_fk FOREIGN KEY (customer_id) REFERENCES ussk_customer (customer_id);

ALTER TABLE ussk_cust_agent

ADD CONSTRAINT cust_agent_fk1 FOREIGN KEY (agent_id) REFERENCES ussk_agent (agent_id);

ALTER TABLE ussk_cust_mem

ADD CONSTRAINT cust_mem_fk FOREIGN KEY (customer_id) REFERENCES ussk_customer (customer_id);

ALTER TABLE ussk_cust_mem

ADD CONSTRAINT cust_mem_airlines_fk FOREIGN KEY (airline_id) REFERENCES ussk_airlines (airline_id);

ALTER TABLE ussk_flight

ADD CONSTRAINT flight_airlines_fk FOREIGN KEY (airline_id) REFERENCES ussk_airlines (airline_id);

ALTER TABLE ussk_invoice

ADD CONSTRAINT invoice_customers_fk FOREIGN KEY (customer_id) REFERENCES ussk_customer (customer_id);

```
ALTER TABLE ussk_pas_fli_ins
```

```
ADD CONSTRAINT pas_fli_ins_flight_fk FOREIGN KEY ( flight_id ) REFERENCES ussk_flight ( flight_id );
```

```
ALTER TABLE ussk_pas_fli_ins
```

```
ADD CONSTRAINT pas_fli_ins_insurance_fk FOREIGN KEY ( insurance_id ) REFERENCES ussk_insurance ( insurance_id );
```

```
ALTER TABLE ussk_pas_fli_ins
```

```
ADD CONSTRAINT pas_fli_ins_passenger_fk FOREIGN KEY ( passenger_id ) REFERENCES ussk_passenger ( passengers_id );
```

```
ALTER TABLE ussk_passenger
```

```
ADD CONSTRAINT passenger_customer_fk FOREIGN KEY ( customer_id ) REFERENCES ussk_customer ( customer_id );
```

```
ALTER TABLE ussk_payment
```

```
ADD CONSTRAINT payment_invoice_fk FOREIGN KEY ( invoice_number ) REFERENCES ussk_invoice ( invoice_number );
```

```
ALTER TABLE ussk_customer
```

```
ADD CONSTRAINT ussk_customer_ussk_invoice_fk FOREIGN KEY ( ussk_invoice_invoice_number )
```

```
REFERENCES ussk_invoice ( invoice_number );
```

```
CREATE OR REPLACE TRIGGER arc_fkarc_1_ussk_cust_agent BEFORE INSERT OR UPDATE OF customer_id ON  
ussk_cust_agent  
FOR EACH ROW DECLARE  
d VARCHAR2(1);
```

```
BEGIN
```

```
SELECT
```

```
a.cust_type INTO d FROM  
ussk_customer a WHERE  
a.customer_id = :new.customer_id;
```

```
IF ( d IS NULL OR d <> 'A' ) THEN
```

```
raise_application_error(-20223,
```

```
'FK CUST_AGENT_FK in Table USSK_CUST_AGENT violates Arc constraint on Table USSK_CUSTOMER - discriminator  
column CUST_TYPE doesn't have value "A");
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN no_data_found THEN NULL;
WHEN OTHERS THEN RAISE;
END;
```

```
/
```

```
CREATE OR REPLACE TRIGGER arc_fkarc_1_ussk_cust_mem BEFORE INSERT OR UPDATE OF customer_id ON
ussk_cust_mem
```

```
FOR EACH ROW DECLARE
d VARCHAR2(1);
```

```
BEGIN
```

```
SELECT
```

```
a.cust_type INTO d FROM
ussk_customer a WHERE
a.customer_id = :new.customer_id;
```

```
IF ( d IS NULL OR d <> 'M' ) THEN
```

```
raise_application_error(-20223,
```

```
'FK CUST_MEM FK in Table USSK_CUST_MEM violates Arc constraint on Table USSK_CUSTOMER - discriminator
column CUST_TYPE doesn't have value "M"');
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN no_data_found THEN NULL;
WHEN OTHERS THEN RAISE;
END;
```

```
/
```

```
CREATE SEQUENCE ussk_passenger_passengers_id START WITH 1 NOCACHE ORDER;
```

```
CREATE OR REPLACE TRIGGER ussk_passenger_passengers_id BEFORE INSERT ON ussk_passenger
FOR EACH ROW
```

```
WHEN ( new.passengers_id IS NULL ) BEGIN
:new.passengers_id := ussk_passenger_passengers_id.nextval; END;
```

5) ABOUT OUR PROJECT:

We used node.js and react.js both of which are the latest web development frameworks around. React is based on the concept of components where each feature of the webpage is a separate component, as you may see from the files submitted that the home page, the login page, the abouts us page, the payments page are all implemented as different components. Thus it breaks the code into "modules" each of which function independantly from each other but can interact with each other by passing props which are to components what arguments are to functions.

Most of our code is written in Javascript, however it would look like the code for the front end is written in some combination of javascript and html, which would probably not make sense to someone new to react at first. But the key thing to remember here is that we writing the code in "jsx" which essentially allows the user to describe the UI that they want in "html" like code, however that code is just syntatic sugar which is then converted to Javascript using the Babbler compiler. It allows the coder to write javascript code which is not too different from html.

6) TABLE SCREENSHOTS

SQL Worksheet

Clear

Find

Actions

Save

Run

1

SELECT COUNT(*) FROM USSK_AIRCRAFTMODEL;

COUNT(*)
10

[Download](#) [CSV](#)

SQL Worksheet

Clear

Find

Actions

Save

Run





1

SELECT COUNT(*) FROM USSK_AIRLINES;

COUNT(*)
19

[Download](#) [CSV](#)

SQL Worksheet





 Clear  Find Actions ▾  Save  Run

```
1 SELECT COUNT(*) FROM USSK_PASSENGER;
```

COUNT (*)
23

[Download CSV](#)

SQL Worksheet





 Clear  Find Actions ▾  Save  Run

```
1 SELECT COUNT(*) FROM USSK_CUSTOMERS;
```

COUNT (*)
28

[Download CSV](#)

SQL Worksheet





 Clear  Find Actions ▾  Save  Run

```
1 SELECT COUNT(*) FROM USSK_CUST_MEM;
```

COUNT (*)
6

[Download CSV](#)

SQL Worksheet





 Clear  Find Actions ▾  Save  Run

```
1 SELECT COUNT(*) FROM USSK_AGENT;
```

COUNT (*)
6

[Download CSV](#)

SQL Worksheet





 Clear  Find Actions ▾  Save  Run

```
1 SELECT COUNT(*) FROM USSK_INSURANCE;
```

COUNT (*)
6

[Download CSV](#)

SQL Worksheet

 Clear  Find Actions ▾  Save  Run

```
1 SELECT COUNT(*) FROM USSK_INVOICE;
```

COUNT (*)
12

[Download CSV](#)

SQL Worksheet

1 `SELECT COUNT(*) FROM USSK_FLIGHT;`

COUNT(*)

23

[Download CSV](#)

SQL Worksheet

 Clear

 Find

Actions ▾

 Save

 Run

1 `SELECT COUNT(*) FROM USSK_PAYMENT;`

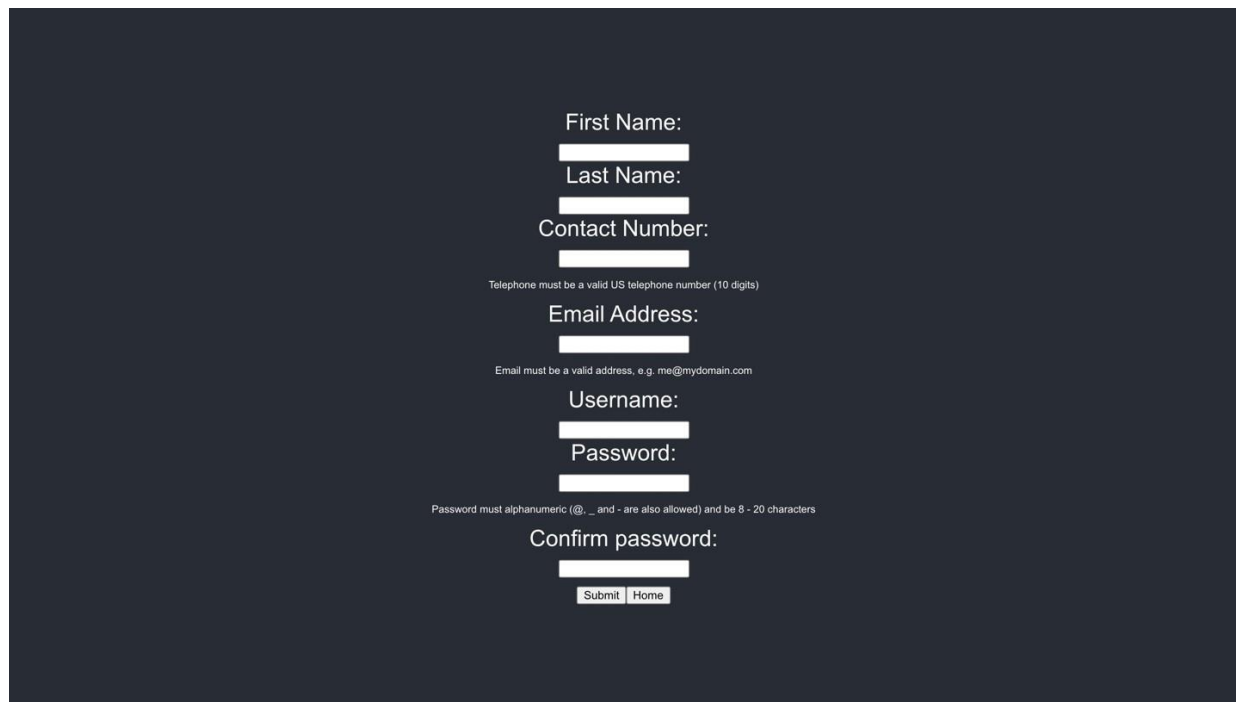
COUNT(*)

13

[Download CSV](#)

7) PROJECT EXECUTION

USER SIGNUP PAGE



A user signup form on a dark blue background. The form is centered and consists of several input fields with labels above them. Below the 'Contact Number' field is a small error message. Below the 'Email Address' field is another small error message. Below the 'Password' field is a third small error message. At the bottom, there are two buttons: 'Submit' and 'Home'.

First Name:

Last Name:

Contact Number:

Telephone must be a valid US telephone number (10 digits)

Email Address:

Email must be a valid address, e.g. me@mydomain.com

Username:

Password:

Password must alphanumeric (@, _ and - are also allowed) and be 8 - 20 characters

Confirm password:

USER LOGIN

Username:
asbcdkjsb

Password:

USER LOGIN CRITERIA

Password does not match set criterion

First Name:
jshadvcb
Last Name:
samcn
Contact Number:
2348994320
Telephone must be a valid US telephone number (10 digits)
Email Address:
ussaboooukarsh@gmail.
Email must be a valid address, e.g. me@mydomain.com
Username:
saikiran98
Password:

Password must alphanumeric (@, _ and - are also allowed) and be 8 - 20 characters
Confirm password:

[Submit](#) [Home](#)

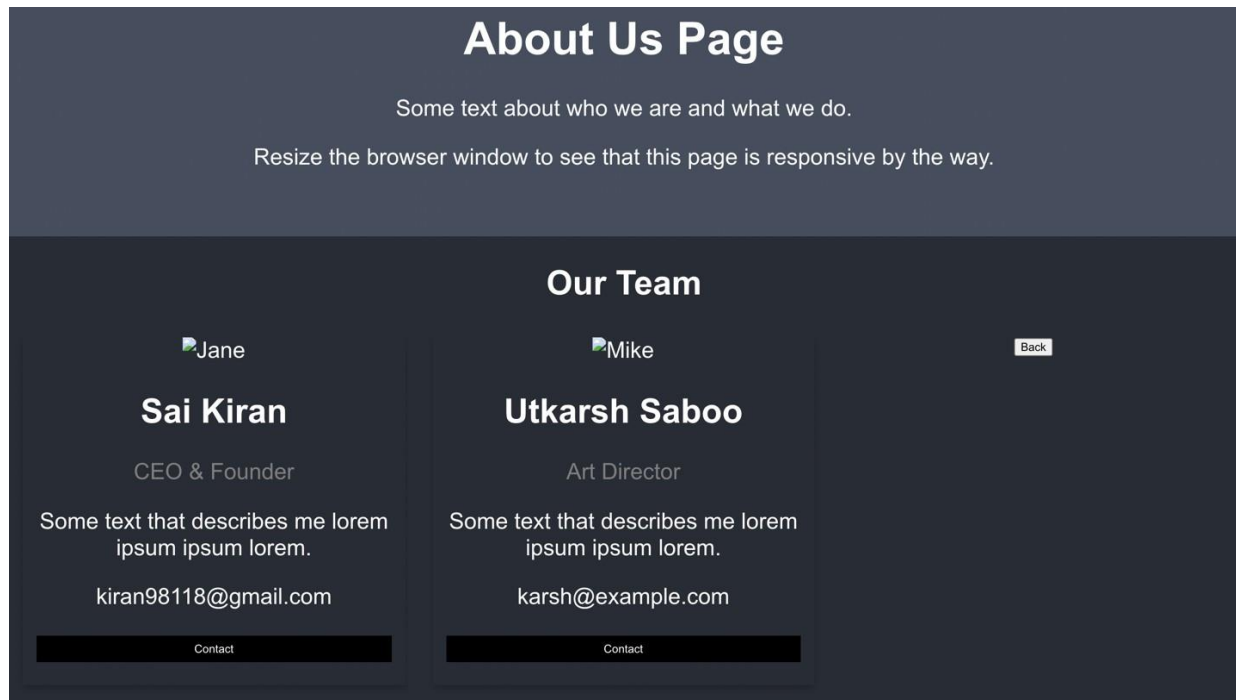
Please enter a valid telephone number

First Name:
jshadvcb
Last Name:
samcn
Contact Number:
234899432
Telephone must be a valid US telephone number (10 digits)
Email Address:
ussaboooukarsh@gmail.
Email must be a valid address, e.g. me@mydomain.com
Username:
saikiran98
Password:

Password must alphanumeric (@, _ and - are also allowed) and be 8 - 20 characters
Confirm password:

[Submit](#) [Home](#)

ABOUT US



phpMyAdmin

Server: localhost - Database: nodemysql - Table: Users

Showing rows 0 - 12 (13 total, Query took 0.0004 seconds.) [Username: BABYBABES... - VARCHAR2...]

`SELECT * FROM `Users` ORDER BY `Username` ASC`

Options

	Username	FName	LName	Email	Password	Contact	admin
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	babybabes	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	zzzzsssss	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	babybabess	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	zzzzsssss	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	babyless	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	zzzzsssss	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	bgf	rgfjnerfjk	jrnfkjdtgnv	tuymerkgmn@gmail.com	zzzzflll	5559595955	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	darknight	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	sqlsqlsql	9999999999	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	thivbfhjghbfhiugbfxnh	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	vvvvvvvv	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	homehome	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	zzzzsssss	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	jislam	jamina	islam	jislam@gmail.com	champion	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	misterkarsh	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	zzzzsssss	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	saikiran98	sai	kiran	sai@gmail.com	wanwares	8681987066	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	salik123	Salli	Kulkarni	sk@gmail.com	qwertyui	9999999999	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Utkarsh	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	zzzzsssss	9999999999	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	varchar2	Utkarsh	Saboo	ussaboooutkarsh@gmail.com	champion	9999999999	0

Query results operations

phpMyAdmin

Server: localhost - Database: nodemysql

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
payments	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Empty <input type="checkbox"/> Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
Users	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Empty <input type="checkbox"/> Drop	13	InnoDB	utf8mb4_general_ci	32.0 KiB	-
2 tables	Sum	13	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Create table

Name: Number of columns: 4

Go

8) SECURITY FEATURES:

SQL INJECTION:

In our project, We have implemented an sql injection to protect databases from malicious attacks. It helps to prevent unexpected and undesired results and thus, adds a layer of security in the project.

a. 9) OUR LEARNINGS AND EXPERIENCE:

SAI KIRAN

Overall, I am very much satisfied with the work that we have done. I know that we could have added more features in our project and make it look better. Though, I have developed a good front end pages, we could not add that in the project due to lack of time. Initially, my partner has used a sample code and worked on back end part. However, lot of time being spent on connecting to a database as we both are having m1 macbook air, we had trouble connecting the database to xampp. It took us 3 days. On the project submission day, we decided not to add the front end code (which is in normal css file) in the react as it may disrupt our entire work. And there are more features we know and could have implemented in back end part as well. Hopefully, we will develop this project in the winter break as this project will add a great addition to our career. Thanks to the professor for giving us this opportunity and to all the TA's for the valuable feedback and doubt sessions.

UTKARSH SABOO

I feel like there was so much more we could add to the project to make it look more user friendly and elegant. However despite our best efforts to start early, which we did, we could not implement everything that we wanted. Learning node and react took a lot of our time too. However it was really fruitful experience which has certainly enhanced my knowledge in database design and software development in general. Me and my project partner plan to finish the project during the winter breaks make it an actual functioning website.

