**CSI2132 - Deliverable 2 - Report**

Udit Kulkarni - Jonathan Lam

**DBMS and Languages**

The DBMS that was used in the creation of this application was postgreSQL and the programming languages used were Java and SQL.

Within this folder are the .jar executable files, java code and the DDL and SQL code used in our project.

**DDL for Creating Database**

**Table Creation:**

Listed below is the DDL used in order to create the database.

CREATE TABLE dbproject.adventures

(

adventure\_id integer NOT NULL,

adventure\_type character varying(20) COLLATE pg\_catalog."default" NOT NULL,

price integer NOT NULL,

days\_long integer NOT NULL,

transportation boolean NOT NULL,

meals character varying(20)[] COLLATE pg\_catalog."default" NOT NULL,

location dbproject.location NOT NULL,

reviews integer,

owner\_id integer NOT NULL,

group\_size character varying(20) COLLATE pg\_catalog."default" NOT NULL,

language character varying(20) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Adventures\_pkey" PRIMARY KEY (adventure\_id),

CONSTRAINT "HOST" FOREIGN KEY (owner\_id)

REFERENCES dbproject.host (host\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE TABLE dbproject.branches

(

branch\_id integer NOT NULL,

country character varying(20) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Branches\_pkey" PRIMARY KEY (branch\_id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE TABLE dbproject.employees

(

employee\_id integer NOT NULL,

name dbproject.name NOT NULL,

address dbproject.address NOT NULL,

"position" character varying(20) COLLATE pg\_catalog."default" NOT NULL,

salary integer NOT NULL,

branch integer NOT NULL,

CONSTRAINT "Employees\_pkey" PRIMARY KEY (employee\_id),

CONSTRAINT "Branch" FOREIGN KEY (branch)

REFERENCES dbproject.branches (branch\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE INDEX "fki\_Branch"

ON dbproject.employees USING btree

(branch ASC NULLS LAST)

TABLESPACE pg\_default;

CREATE TABLE dbproject.experience

(

experience\_id integer NOT NULL,

experience\_type character varying(20) COLLATE pg\_catalog."default" NOT NULL,

language character varying(20) COLLATE pg\_catalog."default" NOT NULL,

price integer NOT NULL,

location dbproject.location NOT NULL,

reviews integer[],

owner\_id integer NOT NULL,

CONSTRAINT "Experience\_pkey" PRIMARY KEY (experience\_id),

CONSTRAINT "HOST" FOREIGN KEY (owner\_id)

REFERENCES dbproject.host (host\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE TABLE dbproject.guest

(

guest\_id integer NOT NULL,

booked\_properties integer[],

address dbproject.address NOT NULL,

name dbproject.name NOT NULL,

email\_addresses character varying(40)[] COLLATE pg\_catalog."default" NOT NULL,

accepted\_terms boolean NOT NULL,

phone\_numbers character varying(12)[] COLLATE pg\_catalog."default" NOT NULL,

dateofbirth date NOT NULL,

CONSTRAINT "Guest\_pkey" PRIMARY KEY (guest\_id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE TABLE dbproject.host

(

host\_id integer NOT NULL,

email\_addresses character varying(40)[] COLLATE pg\_catalog."default" NOT NULL,

name dbproject.name NOT NULL,

address dbproject.address,

phone\_numbers character varying(12)[] COLLATE pg\_catalog."default" NOT NULL,

dateofbirth date NOT NULL,

CONSTRAINT "Host\_pkey" PRIMARY KEY (host\_id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE TABLE dbproject.payment

(

transaction\_id integer NOT NULL,

host\_id integer NOT NULL,

guest\_id integer NOT NULL,

payment\_type character varying(20) COLLATE pg\_catalog."default" NOT NULL,

payment\_amount integer NOT NULL,

payment\_status character varying(20) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Payment\_pkey" PRIMARY KEY (transaction\_id),

CONSTRAINT "GUEST" FOREIGN KEY (guest\_id)

REFERENCES dbproject.guest (guest\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "HOST" FOREIGN KEY (host\_id)

REFERENCES dbproject.host (host\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE INDEX "fki\_CUSTOMER"

ON dbproject.payment USING btree

(guest\_id ASC NULLS LAST)

TABLESPACE pg\_default;

CREATE INDEX "fki\_HOST"

ON dbproject.payment USING btree

(host\_id ASC NULLS LAST)

TABLESPACE pg\_default;

CREATE TABLE dbproject.property

(

property\_id integer NOT NULL,

property\_type character varying(20) COLLATE pg\_catalog."default" NOT NULL,

price integer NOT NULL,

max\_occupancy integer NOT NULL,

min\_occupancy integer NOT NULL,

house\_rules character varying(20)[] COLLATE pg\_catalog."default",

amenities dbproject.amenities NOT NULL,

owner\_id integer NOT NULL,

reviews integer[],

fees dbproject.fees,

address dbproject.address NOT NULL,

dates\_booked integer[],

CONSTRAINT "Property\_pkey" PRIMARY KEY (property\_id),

CONSTRAINT "OWNER" FOREIGN KEY (owner\_id)

REFERENCES dbproject.host (host\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

CREATE INDEX "fki\_OWNER"

ON dbproject.property USING btree

(owner\_id ASC NULLS LAST)

TABLESPACE pg\_default;

**New type creation:**

Because of the nature of the project, composite types had to be made, and below is the SQL code used to create the new types.

CREATE TYPE dbproject."Review" AS

(

communication integer,

cleanliness integer,

value integer,

location integer

);

CREATE TYPE dbproject.address AS

(

house\_number integer,

street\_name character varying(20),

city character varying(20),

state\_province character varying(20),

country character varying(20)

);

CREATE TYPE dbproject.amenities AS

(

bathrooms integer,

bedrooms integer,

beds integer,

kitchens integer,

internet\_access boolean

);

CREATE TYPE dbproject.fees AS

(

cleaning\_fees integer,

transaction\_fees integer,

restocking\_fees integer

);

CREATE TYPE dbproject.location AS

(

city character varying(20),

state\_province character varying(20),

country character varying(20)

);

CREATE TYPE dbproject.name AS

(

first\_name character varying(20),

middle\_name character varying(20),

last\_name character varying(20)

);

**Table Insertion:**

In order to insert into the database, this code is used.

\*These are all general cases for insertion, they were run multiple times with different values in order to propagate the database.

INSERT INTO guest(guest\_id, address, name, email\_addresses, phone\_numbers, dateofbirth,accepted\_terms)

VALUES (3, (580,'Pine St', 'Rimbey','Alberta','Canada'), ('Joseph', 'D','Stephens'), ARRAY['jdstephs8@hotmail.com'], ARRAY['4038433231]'],'1984-9-30',true);

INSERT INTO host(host\_id, address, name, email\_addresses, phone\_numbers, dateofbirth)

VALUES (1000, (3178,'Adelaide St', 'Toronto','Ontario','Canada'), ('Rickie', 'M','Carbonell'), ARRAY['rickmc@hotmail.com'], ARRAY['4165206090'],'1968-1-13');

INSERT INTO property(property\_id,property\_type,price,max\_occupancy, min\_occupancy, house\_rules, address, amenities, owner\_id, fees)

VALUES (2000, 'Apartment',200,5,1, Array['No smoking', 'No parties'],(847,'Sheppard Ave', 'Toronto','Ontario','Canada'),(1,2,2,1,true) ,1000, (10,10,10));

INSERT INTO adventures(adventure\_id,adventure\_type,price,language,days\_long,transportation,group\_size,meals,location,offering\_host\_id)

Values (1001, 'Greece Tour', 500, 'English', 2, true, '1 or more', ARRAY['Souvlaki', 'Spanokopita', 'Dolmas'],('Colonus','Athens','Greece'),1001);

INSERT INTO experience(experience\_id,experience\_type,language,price,location,owner\_id)

VALUES (4001, 'Cooking lessons', 'English', 40, ('Ottawa','Ontario','Canada'),1002);

INSERT INTO branches(branch\_id,country)

VALUES(5001,'Canada');

INSERT INTO employees(employee\_id,name,address,position,salary,branch)

VALUES(6001,('Laura', 'W', 'Herman'), (4950, 'Burdett Ave', 'Victoria', 'British Colombia', 'Canada'), 'Receptionist',45000,5001);

INSERT INTO payment(transaction\_id, host\_id, guest\_id, payment\_type,payment\_amount,payment\_status)

VALUES (7001, 1001, 1, 'Credit', '626', 'Complete');

INSERT INTO rental\_contract(agreement\_id,start\_date,end\_date,guest\_id,host\_id,property\_id,total\_cost)

VALUES (8001,'2020-04-23','2020-04-29',1,1000,2000,1230);

INSERT INTO reviews(review\_id,guest\_id,reviewed\_id,review,additional\_comments)

VALUES (9001,1,4001,(5,4,4,5),'fun cooking experience, would recommend!');

**Table Value Grabbers**

This SQL code is used to grab information from the database to achieve certain tasks in the interfaces.

SELECT \* FROM property WHERE owner\_id = 1000;

SELECT \* FROM adventures WHERE owner\_id = 1000;

SELECT \* FROM experience WHERE owner\_id = 1000;

SELECT \* FROM employees WHERE branch = 5001;

SELECT \* FROM employees WHERE position = 'Manager';

**How to Install the Applications**

The applications are all java executable files. There are three applications, one for administrators, one for hosts and guests and the last one for employees. In order to run the application you must download the files and save them to any location. Since they are a command line application you will have to open the command prompt and enter the following, **java -jar filepath.rar** , the filepath.rar is to be replaced by the actual file path of the application. In order for the guest/host interface to work you must enter a valid id here are some valid Host ids(1000, 1001 and 1002) and some valid Guest Ids(1,2,3 and 5).