CSI 2132 [A] Final Report

April 6th, 2020

Paula Branco

University of Ottawa, Ontario, Canada

Abdullah Morrison || 300057314 || amorr128@uottawa.ca
Aiden Stevenson Bradwell || 300064655 || abrad060@uottawa.ca
Bryan Edler || 7161340 || bryanedler0@gmail.com

1.0 Overview

The TripAdvisor java program runs on an Apache Tomcat server under the address *localhost:XXXX/TripAdvisor/*

It is connected to the PgAdmin 4 Database used in the laboratories, under the sub-database abrad060. It automatically connects and does not require a local database be installed. Provided with the application is a file under the name DataBaseFill.java, which automatically fills the database with information, in the case that you would like to empty and refill it to assure valid design.

All tables and SQL are working within the schema *finalproject*, including the initial database code included in the following sections.

Please find attached with this report the TripAdvisor Project, as well as the SQL extracted from it, and lastly the database initial DDL file in text format.

2.0 Installation and Implementation

To install and run the program, the user is required to launch a local server, and run the java application through it.

STEPS:

1) Install a local sever software (We used Apache Tomcat, however another server application would work)
2) Run the TripAdvisor java program through it (we used Eclipse using the Java Web Plugins, and then chose Run As > Run on Server > CHOOSE SERVER)
3) Load the application using a browser through the localhost port chosen (using localhost:XXXX/TripAdvisor) where XXXX is the port you've chosen to run the server on
4) It will launch the default.html file, which works as the '*home*' for the program

3.0 SQL Data Definition Language

```
CREATE TABLE finalproject.branches (
    branch_id integer NOT NULL,
    country character varying(75) COLLATE pg_catalog."default" NOT NULL ,
    CONSTRAINT branches_pkey PRIMARY KEY (branch_id, country)
);

CREATE TABLE finalproject.reviews (
    review_id integer NOT NULL,
    rating integer,
    cleanliness integer,
    locat integer,
    comfort integer,
    facilities integer,
    overallvalue integer,
    communications character varying(500) COLLATE pg_catalog."default",
    building integer,
    food integer,
    CONSTRAINT reviews_pkey PRIMARY KEY (review_id),
    CONSTRAINT chk_reviews CHECK ((rating = NULL::integer OR rating <= 10 AND rating
>= 1) AND (cleanliness = NULL::integer OR cleanliness <= 10 AND cleanliness >= 1) AND
(locat = NULL::integer OR locat <= 10 AND locat >= 1) AND (comfort = NULL::integer OR
comfort <= 10 AND comfort >= 1) AND (facilities = NULL::integer OR facilities <= 10 AND
facilities >= 1) AND (overallvalue = NULL::integer OR overallvalue <= 10 AND overallvalue
>= 1) AND (building = NULL::integer OR building <= 10 AND building >= 1) AND (food =
NULL::integer OR food <= 10 AND food >= 1))
);

CREATE TABLE finalproject.employees (
    employee_id integer NOT NULL,
    hasposition character varying(50) COLLATE pg_catalog."default",
    salary money,
    CONSTRAINT employees_pkey PRIMARY KEY (employee_id),
    CONSTRAINT employees_salary_check CHECK (salary >= '$0.00'::money)
);

CREATE TABLE finalproject.addresses (
    house_num integer NOT NULL,
```

```
    street character varying(50) COLLATE pg_catalog."default" NOT NULL,
    city character varying(50) COLLATE pg_catalog."default" NOT NULL,
    province character varying(50) COLLATE pg_catalog."default" NOT NULL,
    country character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT addresses_pkey PRIMARY KEY (house_num, street, city, province, country)
);

CREATE TABLE finalproject.people (
    person_id integer NOT NULL,
    branch_id integer NOT NULL,
    branch_country character varying(75) NOT NULL,
    first_name character varying(35) COLLATE pg_catalog."default" NOT NULL,
    middle_name character varying(35) COLLATE pg_catalog."default",
    last_name character varying(35) COLLATE pg_catalog."default" NOT NULL,
    email character varying(50) COLLATE pg_catalog."default" NOT NULL,
    phone_number integer NOT NULL,
    ad_house_number integer NOT NULL,
    ad_street character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_city character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_province character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_country character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT people_pkey PRIMARY KEY (person_id),
    CONSTRAINT people_email_key UNIQUE (email),
    CONSTRAINT people_branch_id_fkey FOREIGN KEY (branch_id, branch_country)
REFERENCES finalproject.branches(branch_id, country),
    CONSTRAINT people_ad_house_number_fkey FOREIGN KEY (ad_house_number, ad_city,
ad_province, ad_street, ad_country)
        REFERENCES finalproject.addresses (house_num, city, province, street, country) MATCH
SIMPLE
);

CREATE TABLE finalproject.properties (
    property_id integer NOT NULL,
    host_id integer NOT NULL,
    rating integer,
    property_type character varying(50) COLLATE pg_catalog."default" NOT NULL,
    amenities character varying(50) COLLATE pg_catalog."default" NOT NULL,
    bathrooms integer NOT NULL,
    bedrooms integer NOT NULL,
    room_type character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_house_number integer NOT NULL,
```

```
    ad_street character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_city character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_province character varying(50) COLLATE pg_catalog."default" NOT NULL,
    ad_country character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT properties_pkey PRIMARY KEY (property_id),
    CONSTRAINT properties_ad_house_number_fkey FOREIGN KEY (ad_house_number,
ad_city, ad_province, ad_street, ad_country)
        REFERENCES finalproject.addresses (house_num, city, province, street, country) MATCH
SIMPLE,
    CONSTRAINT properties_bathrooms_check CHECK (bathrooms > 0),
    CONSTRAINT properties_bedrooms_check CHECK (bedrooms > 0)
);

CREATE TABLE finalproject.hosts (
    host_id integer NOT NULL,
    CONSTRAINT hosts_pkey PRIMARY KEY (host_id),
    CONSTRAINT hosts_host_id_fkey FOREIGN KEY (host_id)
        REFERENCES finalproject.people (person_id) MATCH SIMPLE
);

CREATE TABLE finalproject.guests (
    guest_id integer NOT NULL,
    CONSTRAINT guests_pkey PRIMARY KEY (guest_id),
    CONSTRAINT guests_guest_id_fkey FOREIGN KEY (guest_id)
        REFERENCES finalproject.people (person_id) MATCH SIMPLE
);

CREATE TABLE finalproject.employs (
    country character varying(75) COLLATE pg_catalog."default",
    branch_id integer NOT NULL,
    employee_id integer NOT NULL,
    CONSTRAINT employs_pkey PRIMARY KEY (branch_id, employee_id),
    CONSTRAINT employs_branch_id_fkey FOREIGN KEY (branch_id, country)
        REFERENCES finalproject.branches (branch_id, country) MATCH SIMPLE,
    CONSTRAINT employs_employee_id_fkey FOREIGN KEY (employee_id)
        REFERENCES finalproject.employees (employee_id) MATCH SIMPLE
);

CREATE TABLE finalproject.login_credentials (
    username character varying(50) COLLATE pg_catalog."default" NOT NULL,
    passw character varying(50) COLLATE pg_catalog."default" NOT NULL,
```

```
    person_id integer NOT NULL,
    CONSTRAINT login_credentials_pkey PRIMARY KEY (username, passw, person_id),
    CONSTRAINT login_credentials_person_id_fkey FOREIGN KEY (person_id)
        REFERENCES finalproject.people (person_id) MATCH SIMPLE
);

CREATE TABLE finalproject.makes (
    guest_id integer,
    property_id integer,
    review_id integer,
    CONSTRAINT makes_guest_id_fkey FOREIGN KEY (guest_id)
        REFERENCES finalproject.guests (guest_id) MATCH SIMPLE,
    CONSTRAINT makes_property_id_fkey FOREIGN KEY (property_id)
        REFERENCES finalproject.properties (property_id) MATCH SIMPLE,
    CONSTRAINT makes_review_id_fkey FOREIGN KEY (review_id)
        REFERENCES finalproject.reviews (review_id) MATCH SIMPLE
);

CREATE TABLE finalproject.manages (
    manager_id integer NOT NULL,
    employee_id integer NOT NULL,
    CONSTRAINT manages_pkey PRIMARY KEY (manager_id, employee_id),
    CONSTRAINT manages_employee_id_fkey FOREIGN KEY (employee_id)
        REFERENCES finalproject.employees (employee_id) MATCH SIMPLE,
    CONSTRAINT manages_manager_id_fkey FOREIGN KEY (manager_id)
        REFERENCES finalproject.employees (employee_id) MATCH SIMPLE
);

CREATE TABLE finalproject.rental_agreements (
    rental_id integer NOT NULL,
    property_id integer,
    rental_type character varying(40) COLLATE pg_catalog."default",
    rental_price money,
    signing character varying(25) COLLATE pg_catalog."default",
    signing_date timestamp without time zone NOT NULL,
    start_date timestamp without time zone NOT NULL,
    end_date timestamp without time zone NOT NULL,
    CONSTRAINT rental_agreements_pkey PRIMARY KEY (rental_id),
    CONSTRAINT rental_agreements_property_id_fkey FOREIGN KEY (property_id)
        REFERENCES finalproject.properties (property_id) MATCH SIMPLE
);
```

```
CREATE TABLE finalproject.signs (
    rental_id integer not null,
    host_id integer not null,
    guest_id integer not null,

    CONSTRAINT signs_pkey PRIMARY KEY (rental_id),

    CONSTRAINT signs_rental_id_fkey FOREIGN KEY (rental_id)
        REFERENCES finalproject.rental_agreements (rental_id) MATCH SIMPLE,

    CONSTRAINT signs_host_id_fkey FOREIGN KEY (host_id)
        REFERENCES finalproject.hosts (host_id) MATCH SIMPLE,

    CONSTRAINT signs_guest_id_fkey FOREIGN KEY (guest_id)
        REFERENCES finalproject.guests (guest_id) MATCH SIMPLE
);

CREATE TABLE finalproject.payments(
    payment_id integer NOT NULL,
    host_id integer NOT NULL,
    rental_id integer NOT NULL UNIQUE,
    payment_type character varying(50) COLLATE pg_catalog."default" NOT NULL,
    payment_amount money not null,
    status character varying(25) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT payments_pkey PRIMARY KEY (payment_id),
    CONSTRAINT payments_host_id_fkey FOREIGN KEY (host_id)
        REFERENCES finalproject.hosts (host_id) MATCH SIMPLE,
    CONSTRAINT chk_payments CHECK (
        (payment_type::text = 'Cash'::text OR payment_type::text = 'Debit'::text OR
            payment_type::text = 'Credit'::text)
        AND (payment_type::text = 'Cash'::text AND status::text = 'Completed'::text OR
            payment_type::text = 'Debit'::text AND status::text = 'Approved'::text OR
            payment_type::text = 'Credit'::text AND status::text = 'Approved'::text OR
            status::text = 'Pending'::text)),
     CONSTRAINT payments_rental_id_fkey FOREIGN KEY (rental_id)
        REFERENCES finalproject.rental_agreements (rental_id) MATCH SIMPLE
);

CREATE TABLE finalproject.pays (
    guest_id integer NOT NULL,
```

```
    host_id integer NOT NULL,
    payment_id integer NOT NULL,
    CONSTRAINT pays_pkey PRIMARY KEY (payment_id),
    CONSTRAINT pays_guest_id_fkey FOREIGN KEY (guest_id)
        REFERENCES finalproject.guests (guest_id) MATCH SIMPLE,
    CONSTRAINT pays_host_id_fkey FOREIGN KEY (host_id)
        REFERENCES finalproject.hosts (host_id) MATCH SIMPLE,
    CONSTRAINT pays_payment_id_fkey FOREIGN KEY (payment_id)
        REFERENCES finalproject.payments (payment_id) MATCH SIMPLE

);

CREATE TABLE finalproject.books (
    booking_id integer NOT NULL,
    guest_id integer,
    host_id integer,
    rental_id integer,
    property_id integer,
    CONSTRAINT books_pkey PRIMARY KEY (booking_id),
    CONSTRAINT books_guest_id_fkey FOREIGN KEY (guest_id)
        REFERENCES finalproject.guests (guest_id) MATCH SIMPLE,
    CONSTRAINT books_host_id_fkey FOREIGN KEY (host_id)
        REFERENCES finalproject.hosts (host_id) MATCH SIMPLE,
    CONSTRAINT books_property_id_fkey FOREIGN KEY (property_id)
        REFERENCES finalproject.properties (property_id) MATCH SIMPLE,
    CONSTRAINT books_rental_id_fkey FOREIGN KEY (rental_id)
        REFERENCES finalproject.rental_agreements (rental_id) MATCH SIMPLE
);
```