

# Project II

**Course:** CSGY6083 B Principles of Database Systems

**Section:** B

**Submission Date:** December 14, 2021

**Student Names and IDs:** Youwen Zhang (yz6999), Run Yu (ry2068)

## Table of Contents

a) [Execute Summary](#)

[summary](#)

[Logical Model](#)

[Relational model](#)

[Assumptions](#)

b) [Brief details of software, programming language, and database used](#)

c) [DDL file content.](#)

d) [List of tables, and total number of records of each table](#)

e) [Screenshots of some sessions, pages, menus of your Web Application](#)

[Sessions](#)

[Pages](#)

[Menu](#)

f) [Details of security features that you have implemented on Web Application development](#)

g) [Lesson learned](#)

h) [Business analysis with 6 SQLs using your project data.](#)

[1\)Table joins with at least 3 tables in join](#)

[2\)Multi-row subquery](#)

[3\)Correlated subquery](#)

[4\)SET operator query](#)

[5\)Query with in line view or WITH clause](#)

[6\)TOP-N query](#)

## a) Execute Summary

summary

First, we have a strong authentication and authorization system which can help the company efficiently to manage their user's account login, logout, signup and manage the different pages and data viewed by different users with different permissions, like normal users account and staff account. To be more precisely, we use the django.contrib.auth model to automatically manage the sessions and permissions of each user, this model would create tables django\_session table: store each user's encrypted session and expired time; django.group: store user's type django.permission: store each group's permission.

Second, the customer can update the user's own information and create, update, read, delete the passenger's information. The website will check each input field fully and separately, like if the newly created passenger's passport id is the same as one in the database, if the input phone number is a legal phone number. We do this part based on the automatic type checking and some business rule checking.

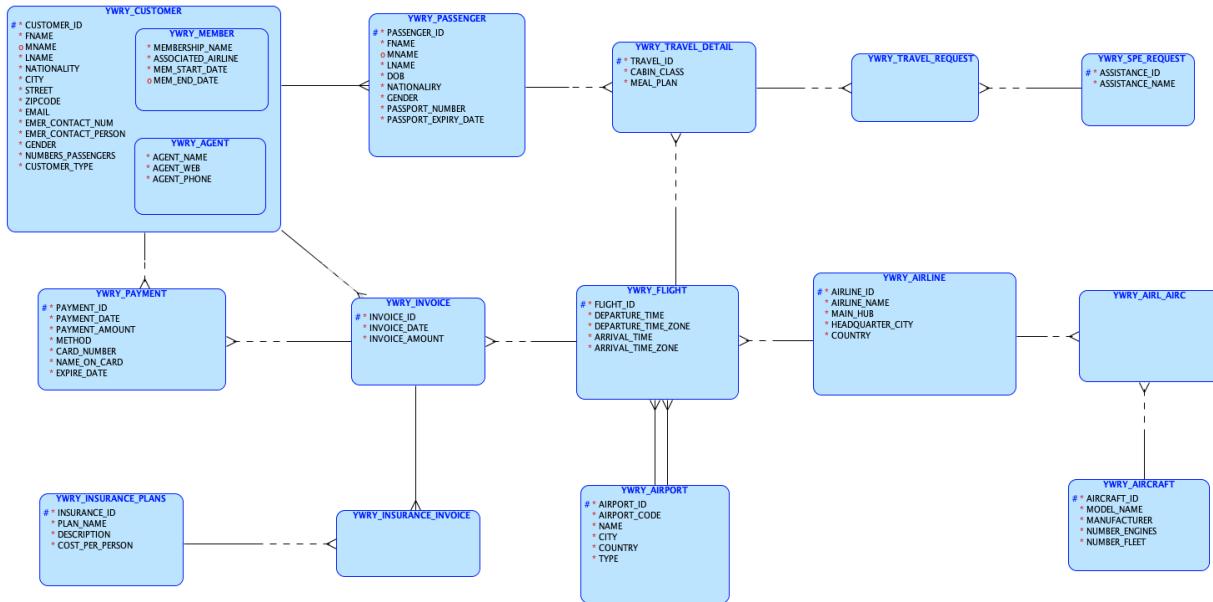
Third, the customer can buy different types of insurance, and all types of insurance bought by this customer would be listed in the invoice list page. Each invoice has a parameter 'expiry time' which is caculated exact three days before the passenger's departure time(business rule). For convenience, the invoices are ordered by its expiration time descending so that customers can see the most recent invoice. And if one invoice expired time is smaller than the current time, then this invoice would be marked expired, you can't do any actions for this invoice. If this invoice expiry time is greater than the current time, you can click a button the view the payment history of this invoice(some digit of card number is masked for security), if this invoice is not fully paid, you can continue to do payment by both debit card or credit card method. If this invoice is fully paid, the website would hide the pay button and tell you that you are done with this invoice

Fourth, we have some special accounts for the company staff. If company staff use the staff account to login in the website, they can view an analyse page which is hidden from customer users.In the analyse page, staffs can view the total pairs sales amount of each month of this year and last year, they can do some analyses based the trend of sales amount or year-on-year comparison. This page also list the top 10 spenders in this year, list the spenders' total insurance spending, order numbers, name and so on, staff can use this cart to do some customer lever analyse

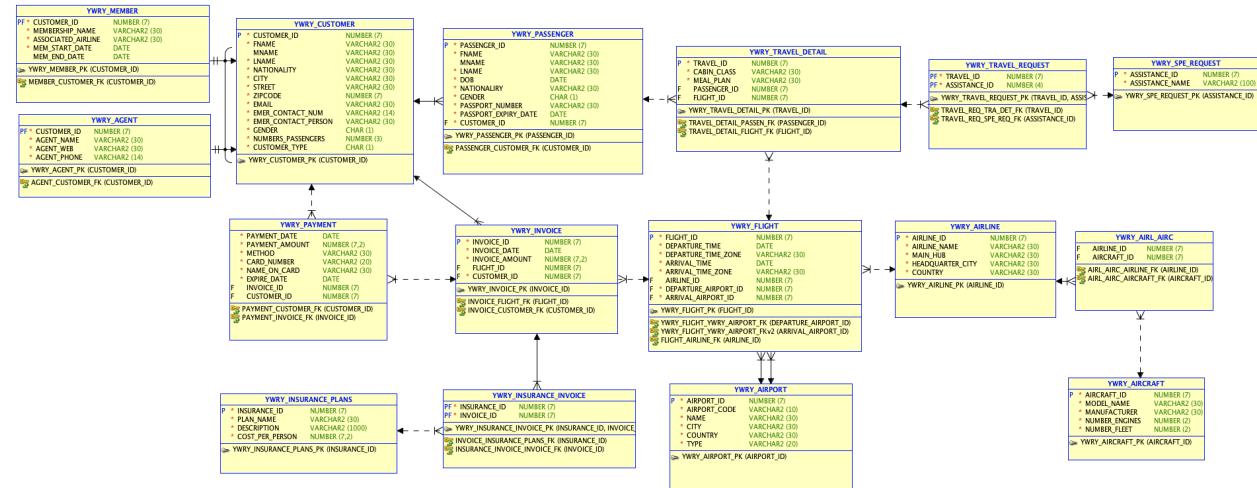
Fifth, For website and sql security, We use some libraries and code rules to prevent SQL injection and Cross Site Scripting. In more detail, we use django's model object and use

parameters in raw sql , two ways to prevent sql injection. Using React's dompurify package to prevent cross site scripting.

## Logical Model



## Relational model



## Assumptions

1. A passenger can have more than one invoice (for different flights), and an invoice can contain multiple insurance plans, e.g. people could buy Trip Cancellation and Medical Insurance at the same time.
2. A passenger can have more than one flight when traveling, each stop has their own meal plan and cabin class. And each travel detail (each stop) can have more than one assistant request.
3. An airline has multiple flights and multiple aircrafts. Any type of aircraft can be employed by different airlines.
4. A flight has a departure airport and an arrival airport. And an airport provides services to different flights.

b) Brief details of software, programming language, and database used

For the backend, we write Python and use django website model, import django.contrib.auth package to complete user authentication and authorization, using rest\_framework package to implement RESTful APIs web structure and url routers

For frontend, we use React model + axios + bootstrap 5 to implement web views, frontend logical functions and http request

For database, we use MYSQL. We add some index on some often used columns like customer email to increase the searching speed, and the django's model object can help us to do the sql transaction automatically

c) DDL file content.

```
CREATE DATABASE IF NOT EXISTS ywry_same;
USE ywry_same;
-- SQLINES DEMO *** le SQL Developer Data Modeler 21.1.0.092.1221
-- SQLINES DEMO *** -11-08 22:18:13 EST
-- SQLINES DEMO *** le Database 11g
-- SQLINES DEMO *** le Database 11g
-- SQLINES DEMO *** no DDL - MDSYS.SDO_GEOMETRY
-- SQLINES DEMO *** no DDL - XMLTYPE
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_agent (
    customer_id INT NOT NULL,
    agent_name VARCHAR(30) NOT NULL,
    agent_web VARCHAR(30) NOT NULL,
    agent_phone VARCHAR(14) NOT NULL
);
ALTER TABLE ywry_agent ADD CONSTRAINT ywry_agent_pk PRIMARY KEY (
customer_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_aircraft (
aircraft_id
model_name
manufacturer
number_engines TINYINT NOT NULL,
number_fleet TINYINT NOT NULL
INT NOT NULL,
VARCHAR(30) NOT NULL,
VARCHAR(30) NOT NULL,
);
ALTER TABLE ywry_aircraft ADD CONSTRAINT ywry_aircraft_pk PRIMARY KEY (
aircraft_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_airl_airc (
    airline_id INT,
    aircraft_id INT
);
-- SQLINES LICENSE FOR EVALUATION USE ONLY
```

```
CREATE TABLE ywry_airline (
    airline_id
    airline_name
    main_hub
    headquarter_city  VARCHAR(30) NOT NULL,
    country          VARCHAR(30) NOT NULL
    INT NOT NULL,
    VARCHAR(30) NOT NULL,
    VARCHAR(30) NOT NULL,
);
ALTER TABLE ywry_airline ADD CONSTRAINT ywry_airline_pk PRIMARY KEY (
    airline_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_airport (
    airport_id      INT NOT NULL,
    airport_code    VARCHAR(10) NOT NULL,
    name            VARCHAR(30) NOT NULL,
    city            VARCHAR(30) NOT NULL,
    country         VARCHAR(30) NOT NULL,
    type            VARCHAR(20) NOT NULL
);
ALTER TABLE ywry_airport ADD CONSTRAINT ywry_airport_pk PRIMARY KEY (
    airport_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_customer (
    customer_id
    fname
    mname
    lname
    nationality
    city
    street
    zipcode
    email
    emer_contact_num
    INT NOT NULL,
    VARCHAR(30) NOT NULL,
    VARCHAR(30),
```

```

VARCHAR(30) NOT NULL,
VARCHAR(30) NOT NULL,
VARCHAR(30) NOT NULL,
VARCHAR(30) NOT NULL,
INT NOT NULL,
VARCHAR(30) NOT NULL,
VARCHAR(14) NOT NULL COMMENT 'contact number with
country code',
    emer_contact_person  VARCHAR(30) NOT NULL,
);
gender
numbers_passengers
customer_type
CHAR(1) NOT NULL,
SMALLINT NOT NULL,
CHAR(1) NOT NULL
ALTER TABLE ywry_customer
    ADD CONSTRAINT ch_inh_ywry_customer CHECK ( customer_type IN (
'B',
'D', 'M' ) );
/* Moved to CREATE TABLE
COMMENT ON COLUMN ywry_customer.emer_contact_num IS
    'contact number with country code'; */
ALTER TABLE ywry_customer ADD CONSTRAINT ywry_customer_pk PRIMARY KEY
(
customer_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_flight (
    flight_id                  INT NOT NULL COMMENT 'FLIGHT_ID is the
flight
number mentioned in the document',

    departure_time
    departure_time_zone
    arrival_time
    arrival_time_zone
    airline_id
    departure_airport_id  INT NOT NULL,
    arrival_airport_id   INT NOT NULL

```

```
DATETIME NOT NULL,
VARCHAR(30) NOT NULL,
DATETIME NOT NULL,
VARCHAR(30) NOT NULL,
INT,
);
/* Moved to CREATE TABLE
COMMENT ON COLUMN ywry_flight.flight_id IS
    'FLIGHT_ID is the flight number mentioned in the document'; */
ALTER TABLE ywry_flight ADD CONSTRAINT ywry_flight_pk PRIMARY KEY (
flight_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_insurance_invoice (
    insurance_id INT,
    invoice_id     INT NOT NULL
);
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_insurance_plans (
insurance_id
plan_name
description
cost_per_person DECIMAL(7, 2) NOT NULL
INT NOT NULL,
VARCHAR(30) NOT NULL,
VARCHAR(1000) NOT NULL,
);
ALTER TABLE ywry_insurance_plans ADD CONSTRAINT
ywry_insurance_plans_pk
PRIMARY KEY ( insurance_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_invoice (
    invoice_id     INT NOT NULL,
    invoice_date   DATETIME NOT NULL,
    invoice_amount DECIMAL(7, 2) NOT NULL COMMENT 'USD',
    flight_id      INT,
    passenger_id   INT
);
/* Moved to CREATE TABLE
COMMENT ON COLUMN ywry_invoice.invoice_amount IS
```

```
'USD'; */
ALTER TABLE ywry_invoice ADD CONSTRAINT ywry_invoice_pk PRIMARY KEY (
invoice_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_member (
    customer_id      INT NOT NULL,
    membership_name  VARCHAR(30) NOT NULL,
    associated_airline VARCHAR(30) NOT NULL,
    mem_start_date   DATETIME NOT NULL,
    mem_end_date     DATETIME
);
ALTER TABLE ywry_member ADD CONSTRAINT ywry_member_pk PRIMARY KEY (
customer_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_passenger (
);
passenger_id
fname
mname
lname
                INT NOT NULL,
                VARCHAR(30) NOT NULL,
                VARCHAR(30),
                VARCHAR(30) NOT NULL,
                DATETIME NOT NULL,
                VARCHAR(30) NOT NULL,
                CHAR(1) NOT NULL,
                VARCHAR(30) NOT NULL,
ALTER TABLE ywry_passenger ADD CONSTRAINT ywry_passenger_pk PRIMARY
KEY (
passenger_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_payment (
    payment_date     DATETIME NOT NULL,
    payment_amount   DECIMAL(7, 2) NOT NULL,
dob
nationaliry
gender
```

```
passport_number
passport_expiry_date  DATETIME NOT NULL,
customer_id           INT NOT NULL
method
card_number
name_on_card
expire_date
invoice_id
customer_id      INT
VARCHAR(30) NOT NULL,
VARCHAR(20) NOT NULL,
VARCHAR(30) NOT NULL,
DATETIME NOT NULL,
INT,
);
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_spe_request (
    assistance_id      INT NOT NULL,
    assistance_name    VARCHAR(100) NOT NULL
);
ALTER TABLE ywry_spe_request ADD CONSTRAINT ywry_spe_request_pk
PRIMARY KEY
( assistance_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_travel_detail (
    travel_id          INT NOT NULL,
    cabin_class        VARCHAR(2) NOT NULL COMMENT '''E'' for Economy
class,
'''EP'' for Economy Plus class, ''B'' for Business class, ''F'' for
first
class',
    meal_plan          VARCHAR(10) NOT NULL,
    passenger_id       INT,
    flight_id          INT
);
/* Moved to CREATE TABLE
COMMENT ON COLUMN ywry_travel_detail.cabin_class IS
'''E'' for Economy class, ''EP'' for Economy Plus class, ''B''
```

```
for
Business class, ''F'' for first class'; */
ALTER TABLE ywry_travel_detail ADD CONSTRAINT ywry_travel_detail_pk
PRIMARY
KEY ( travel_id );
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE ywry_travel_request (
    travel_id      INT,
    assistance_id  INT
);
ALTER TABLE ywry_agent
    ADD CONSTRAINT agent_customer_fk FOREIGN KEY ( customer_id )
        REFERENCES ywry_customer ( customer_id );
ALTER TABLE ywry_airl_airc
    ADD CONSTRAINT airl_airc_aircraft_fk FOREIGN KEY ( aircraft_id )
        REFERENCES ywry_aircraft ( aircraft_id );
ALTER TABLE ywry_airl_airc
    ADD CONSTRAINT airl_airc_airline_fk FOREIGN KEY ( airline_id )
        REFERENCES ywry_airline ( airline_id );
ALTER TABLE ywry_flight
    ADD CONSTRAINT flight_airline_fk FOREIGN KEY ( airline_id )
        REFERENCES ywry_airline ( airline_id );
ALTER TABLE ywry_insurance_invoice
    ADD CONSTRAINT insurance_invoice_invoice_fk FOREIGN KEY (
invoice_id )
        REFERENCES ywry_invoice ( invoice_id );
ALTER TABLE ywry_invoice
    ADD CONSTRAINT invoice_flight_fk FOREIGN KEY ( flight_id )
        REFERENCES ywry_flight ( flight_id );

ALTER TABLE ywry_insurance_invoice
    ADD CONSTRAINT invoice_insurance_plans_fk FOREIGN KEY (
insurance_id )
        REFERENCES ywry_insurance_plans ( insurance_id );
ALTER TABLE ywry_invoice
    ADD CONSTRAINT invoice_passenger_fk FOREIGN KEY ( passenger_id )
        REFERENCES ywry_passenger ( passenger_id );
ALTER TABLE ywry_member
    ADD CONSTRAINT member_customer_fk FOREIGN KEY ( customer_id )
```

```

        REFERENCES ywry_customer ( customer_id );
ALTER TABLE ywry_passenger
    ADD CONSTRAINT passenger_customer_fk FOREIGN KEY ( customer_id )
        REFERENCES ywry_customer ( customer_id );
ALTER TABLE ywry_payment
    ADD CONSTRAINT payment_customer_fk FOREIGN KEY ( customer_id )
        REFERENCES ywry_customer ( customer_id );
ALTER TABLE ywry_payment
    ADD CONSTRAINT payment_invoice_fk FOREIGN KEY ( invoice_id )
        REFERENCES ywry_invoice ( invoice_id );
ALTER TABLE ywry_travel_detail
    ADD CONSTRAINT travel_detail_flight_fk FOREIGN KEY ( flight_id )
        REFERENCES ywry_flight ( flight_id );
ALTER TABLE ywry_travel_detail
    ADD CONSTRAINT travel_detail_passen_fk FOREIGN KEY ( passenger_id
)
        REFERENCES ywry_passenger ( passenger_id );
ALTER TABLE ywry_travel_request
    ADD CONSTRAINT travel_req_spe_req_fk FOREIGN KEY ( assistance_id
)
        REFERENCES ywry_spe_request ( assistance_id );
ALTER TABLE ywry_travel_request
    ADD CONSTRAINT travel_req_tra_det_fk FOREIGN KEY ( travel_id )
        REFERENCES ywry_travel_detail ( travel_id );
ALTER TABLE ywry_flight
    ADD CONSTRAINT ywry_flight_ywry_airport_fk FOREIGN KEY (
departure_airport_id )
        REFERENCES ywry_airport ( airport_id );
ALTER TABLE ywry_flight
    ADD CONSTRAINT ywry_flight_ywry_airport_fkv2 FOREIGN KEY (
arrival_airport_id )

REFERENCES ywry_airport ( airport_id );
ALTER TABLE ywry_payment
    ADD CONSTRAINT payment_method_check CHECK ( method IN ( 'Debit ',
'Credit ' ) );

```

d) List of tables, and total number of records of each table

Name	Rows	Data Len:
account_emailaddress	6	
account_emailconfirmation	0	
auth_group	0	
auth_group_permissions	0	
auth_permission	116	
authtoken_token	1	
django_admin_log	0	
django_content_type	29	
django_migrations	30	
django_session	96	
django_site	1	
socialaccount_socialaccount	0	
socialaccount_socialapp	0	
users_customuser	43	
users_customuser_groups	0	
users_customuser_user_per...	0	
ywry_agent	10	
ywry_aircraft	20	
ywry_airl_airc	30	
ywry_airline	20	
ywry_airport	20	
ywry_flight	30	
ywry_insurance_invoice	30	
ywry_insurance_plans	6	
ywry_invoice	30	
ywry_member	9	
ywry_passenger	22	
ywry_payment	30	
ywry_spe_request	6	
ywry_travel_detail	30	
ywry_travel_request	34	

## e) Screenshots of some sessions, pages, menus of your Web Application

### Sessions

#### Session used in http request header

**Referer:** `http://localhost:3000/`

**sec-ch-ua:** "Google Chrome";v="95", "Chromium";v="95",  
";Not A Brand";v="99"

**sec-ch-ua-mobile:** ?0

**sec-ch-ua-platform:** "macOS"

**Token:** `0vy9ggrb0vtatagyrlv0mc306pvo4gkz`

**User-Agent:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10

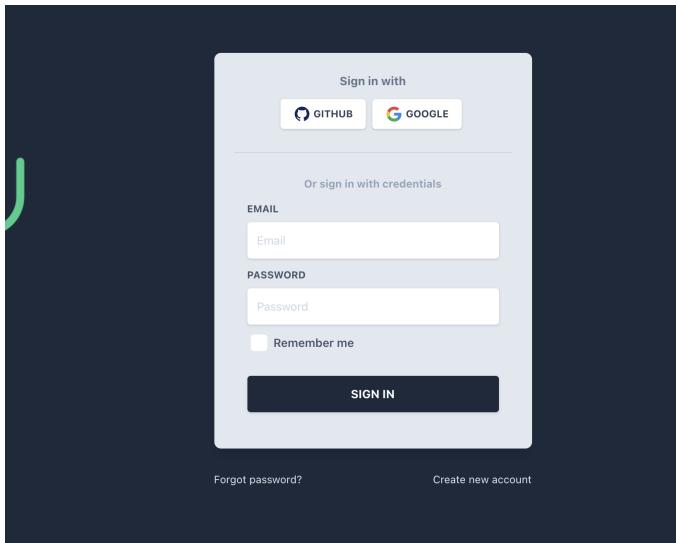
\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrom

### Sessions stored in database:

session_key	session_data	expire_date
037v2x9pt0a29t4q99oney3k	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 16:38:09.217134
0n9vlnxgvb0vhhsd7y1n5us8q80bjm	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-24 17:09:37.556895
0vy9ggrb0vtatagyrlv0mc306pvo4gkz	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 19:23:91.5495815
0zuy8am3vmlwiwpe3q3z2otl23lm	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 16:30:40.926220
10pu8s6kgpo0fqsnp1a9v8y0bgjhyh	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-26 18:55:01.515569
1kxcnwncupbgfap36bre5op5qn72i	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-26 03:43:47.533426
1cup09pe6463dn76515xuwvtxdfj	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-24 16:21:57.751675
1dy09pe65e463dn76515xuwvtxdfj	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 17:00:26.670983
23u0nfnozzoufn4010b5a40axn728omk9	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 08:34:44.422582
2gn266uxgg0iqxkm17wv7a9b6nh45xr9	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 05:59:57.28972
2umvwmpa4xeu5ey8nvelepe8w8t5gqc	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-27 19:17:35.641382
3m22fj9q7yd3z7v4hh6r243g1u8m	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 07:57.59154600
3m5tnnmwwgyu6a64rw970d0uj0cx121	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 04:12:50.260705
3s482yphwr055b7a52z33b278ujpoda	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-20 02:55:33.825391
48evmbmufuvz16rsf16esfxgvck9j	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 06:18:45.675141
4103zehlq4wvapg9ihkd7h035ugd7v	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-26 03:24:04.924830
5f64vcz40ua4v8n2cm5rcs4p1bfv6af	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-24 17:00:14.407605
6cm42zsg9euvwrfj6esgek7w36su	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-22 02:05:55.666590
6ftvqqcahm6ra8s35ja1qlbh6qjkl7	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 16:36:53.595369
6h55z2f5hmc6ra8s35ja1qlbh6qjkl7	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-23 19:34:45.184212
7ynlmvq9y8zksw27o09hwe4n4oc7	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 05:54:33.74422
8bu08if1d42d2h0khdm0592ko15	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 03:55:54.898597
8rz75103nb1zw56bd7204mpcrz19	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-26 06:32:38.875736
810d1p3aolsccscsizwxwlgf59gjwv7j	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-26 06:36:15.179747
Amwmcr7f?2?wv14swf1rfinh7lnknw	.eJxVJEKwAURO-SdSipYp005xfuBEFBXIWf5LeNikSaHtTx7zQZQZc2z-a9ilZp7PSUcnDekZowSrLfo0C9yVgWd4XQxtGMA7e5Usr2vK99Fhv13ZP0EHqZvfhujpOXWmpksSVHUiISFgKcEVlw	2021-12-21 08:23:16.862714

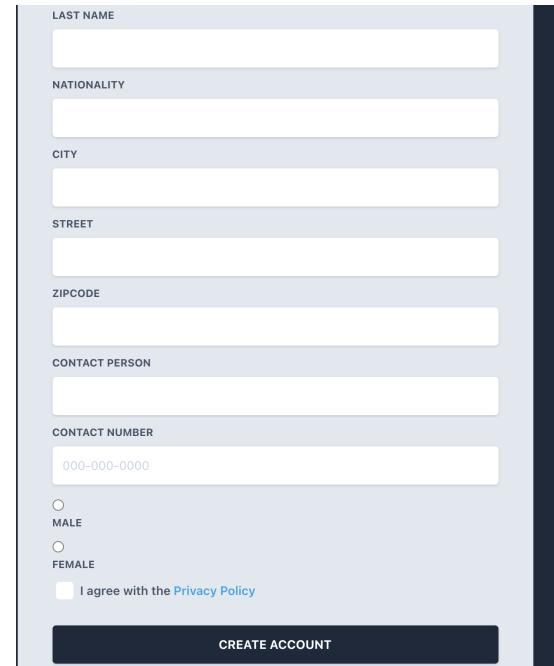
## Pages

Login page:



The login page features a dark background with a central white login form. At the top, there's a "Sign in with" section with "GITHUB" and "GOOGLE" buttons. Below this, a "Or sign in with credentials" section contains "EMAIL" and "PASSWORD" fields. A "Remember me" checkbox is located next to the password field. A large "SIGN IN" button is at the bottom. At the very bottom of the page are two small links: "Forgot password?" and "Create new account".

Register page:



The register page has a light gray background with a dark blue header bar at the top. It contains several input fields: "LAST NAME", "NATIONALITY", "CITY", "STREET", "ZIPCODE", "CONTACT PERSON", "CONTACT NUMBER" (containing "000-000-0000"), and gender selection fields ("MALE" and "FEMALE"). There is also a checkbox for "I agree with the [Privacy Policy](#)". A large "CREATE ACCOUNT" button is at the bottom.

User information:

The screenshot displays a dashboard interface for managing passengers. On the left, a sidebar lists categories: SAFE, INFORMATION PAGES (User Information, Passenger), ENSURANCE (Invoice), MANAGEMENT (Analyse). The main area is titled 'DASHBOARD' and contains four cards: TRAFFIC (350,897, +3.48% Since last month), NEW USERS (2,356, -3.48% Since last week), SALES (924, -1.10% Since yesterday), and PERFORMANCE (49,65%, +12% Since last month). Below this is a detailed 'User Information' section for a passenger named Youwen Zhang, female, from Jersey City, China, with an email of 2@nyu.edu. The contact information includes street address 605 Pavonia ave.

DASHBOARD

SAFE

INFORMATION PAGES

User Information

PASSINGER

ENSURANCE

INVOICE

MANAGEMENT

ANALYSE

TRAFFIC  
350,897

NEW USERS  
2,356

SALES  
924

PERFORMANCE  
49,65%

↑ 3.48% Since last month

↓ 3.48% Since last week

↓ 1.10% Since yesterday

↑ 12% Since last month

User Information

USER INFORMATION

FIRST NAME: Youwen

LAST NAME: Zhang

EMAIL: 2@nyu.edu

GENDER: Female

CONTACT INFORMATION

STREET: 605 Pavonia ave

CITY: jersey city

NATIONALITY: China

ZIPCODE: 17306

Passenger CURD:

INFOINATION PAGES

USER INFORMATION

PASSENGER

ENSURANCE

INVOICE

MANAGEMENT

**Passenger Information**

Add Passenger

NAME	BIRTHDAY	NATIONALITY	PASSPORT	PASSPORT EXPIRY DATE	GENDER	OPERATION
Yang Kuhlman	2003-08-01	USA	482099981	2022-08-27	Male	<a href="#">edit</a>
Ramon Padberg	1992-07-21	United Arab Emirates	492838591	2024-08-27	Female	<a href="#">edit</a>
Zhang Jack	1992-07-15	United Arab Emirates	959383825	2025-08-27	Female	<a href="#">edit</a>
Mallory Lowe	1992-07-11	Czech Republic	38485838272	2028-08-26	Female	<a href="#">edit</a>
Carly Ma	1989-08-11	Czech Republic	3848475123	2028-08-27	Female	<a href="#">edit</a>
Floyd Connelly	1972-08-22	Haiti	4837571	2028-08-27	Female	<a href="#">edit</a>
River Brakus	1972-09-11	Afghanistan	483828405	2028-08-27	Female	<a href="#">edit</a>
Jeff Smith	1989-01-12	Afghanistan	68583231	2030-08-27	Female	<a href="#">edit</a>
Jewell Skiles	1999-02-23	Pakistan	123857313	2032-08-27	Male	<a href="#">edit</a>
Cccc B	2021-12-07	USA	555444	2021-12-30	Female	<a href="#">edit</a>

← → ⌂ localhost:3000/admin/passengeredit/1000000

SAFE

INFOINATION PAGES

USER INFORMATION

PASSENGER

ENSURANCE

INVOICE

MANAGEMENT

**Passenger Information**

[delete](#)

**PASSENGER INFORMATION**

<b>FIRST NAME</b>	<b>LAST NAME</b>
Yang	Kuhlman
<b>DOB</b>	<b>GENDER</b>
2003/08/01	Male

**PASSPORT INFORMATION**

<b>PASSPORT NUMBER</b>	
482099981	
<b>PASSPORT EXPIRY DATE</b>	<b>NATIONALITY</b>
2022/08/27	USA

**SUBMIT CHANGE**

## Invoice list:

Invoice List						
INVOICEID	NAME	PASSPORT NUMBER	PAYMENT AMOUNT	PAYMENT DEADLINE	FLIGHTID	
1000015	Zhang Jack	959383825	867.82\$	2022-01-06 00:30:00	1000006	<a href="#">payment detail</a>
1000027	River Brakus	483828405	270.18\$	2021-12-28 00:00:00	1000023	<a href="#">payment detail</a>
1000004	Zhang Jack	959383825	943.25\$	2021-12-22 00:00:00	1000027	<a href="#">payment detail</a>
1000001	River Brakus	483828405	639.14\$	2021-12-17 08:00:00	1000013	<a href="#">payment detail</a>
1000002	Yang Kuhlman	482099981	937\$	2021-10-10 00:00:00	1000019	expired
1000026	Zhang Jack	959383825	579.89\$	2021-09-03 00:00:00	1000012	expired
1000028	Ramon Padberg	492838591	170.94\$	2021-08-30 00:00:00	1000018	expired
1000023	Carly Ma	3848475123	772.18\$	2021-08-27 00:00:00	1000003	expired
1000005	Yang Kuhlman	482099981	714.53\$	2021-07-16 00:00:00	1000014	expired
1000011	Jeff Smith	68583231	743.4\$	2021-07-01 00:00:00	1000025	expired
1000020	Floyd Connally	4837571	161.27\$	2021-05-15 00:00:00	1000021	expired
1000021	Yang Kuhlman	482099981	566.34\$	2021-04-22 00:00:00	1000001	expired
1000018	Jeff Smith	68583231	975.61\$	2021-04-12 00:00:00	1000010	expired
1000010	Zhang Jack	959383825	884.78\$	2021-03-22 00:00:00	1000006	expired

## Payment List

PAYMENT AMOUNT	METHOD	CARD NUMBER	NAME ON CARD	CAR EXPIRE DATE	PAYMENT DATE
101.12\$	Debit	*****3834	Susana Bruen	2024-08-29	2021-12-30 00:00:00
688.96\$	Credit Card	*****4455	Amir Braun	2024-05-13	2021-01-31 00:00:00

invoice amount:867.82\$

balance: 77.74\$ To Pay

## Payment List

PAYMENT AMOUNT	METHOD	CARD NUMBER	NAME ON CARD	CAR EXPIRE DATE	PAYMENT DATE
243.63\$	Debit	*****3619	Desmond Mertz	2029-06-13	2021-02-21 00:00:00
395.51\$	Credit Card	*****4651	Dr. Katelin Toy	2024-09-13	2021-09-15 00:00:00

invoice amount:639.14\$

balance: 0\$ (this invoice is finished)

## Analyses:

## OVERVIEW

### Ensurance Sale(all completed invoice)



### TOP 10 SPENDERS

PASSENGER NAME	PASSENGER ID	TOTAL AMOUNT	ORDER NUMBER
Zhang Jack	1000003	3275.74	4
Yang Kuhlman	1000000	2217.87	3
Jeff Smith	1000008	1719.01	2
Zin Shin	1000018	1606.45	3
Brock Jakubowski	1000015	1410.35	2
River Brakus	1000007	909.32	2
Dilly Dal	1000017	775.89	1
Carly Ma	1000005	772.18	1
Jaron Lockman	1000019	688.47	3
Peggie Donnelly	1000016	623.72	3

## Menu

SAFE

---

INFOMATION PAGES

 USER INFORMATION

 PASSENGER

---

ENSURANCE

 INVOICE

---

MANAGEMENT

 ANALYSE

### f) Details of security features that you have implemented on Web Application development

For website and sql security, We use some libraries and code rules to prevent SQL injection and Cross Site Scripting. In more detail, first we check each input's type and validity for the first defense. Second we use django's model object to do a mapping between mysql and django, then we can write sql by calling functions of this object to prevent sql injection. But for some complex sql, like using join in many tables, we need to write raw sql but use parameters in sql string to prevent sql injection. For cross site scripting, we use React's dompurify package to prevent cross site scripting, for example, when inserting some javascript labels "<111>", it will show "&lt;111&gt;" as legal html output in the front side

### g) Lesson learned

We learn a lot from this database class, like how to implement a third normalization of a database, how to first analyze a business model and then change it to a logical and relational model which can be used in databases. So we first write the logical and relational model learned in class for this business model, and then write data insert sql we learn in class. Then we learned how to write complex query sql using join and sub select, we use it to do our analyse function by joining many tables.

The difficulties we faced were usually the unfamiliarity of django and react model, but for the willingness of learning a new technology, we spent much time on searching the document and dealing with bugs in the project coding.

## h) Business analysis with 6 SQLs using your project data.

### 1)Table joins with at least 3 tables in join

The screenshot shows a MySQL database interface with the following details:

Session: root | Website2 | ▶ | ⌂

```
1 select ywry_invoice.invoice_id, ywry_passenger.fname,ywry_passenger.lname, ywry_passenger.passport_number,
2 ywry_invoice.invoice_amount, ywry_invoice.invoice_date, ywry_invoice.flight_id
3 from users_customuser,ywry_passenger,ywry_invoice
4 where users_customuser.customer_id = ywry_passenger.customer_id AND
5 ywry_passenger.passenger_id = ywry_invoice.passenger_id AND users_customuser.customer_id = 37
6 ORDER BY invoice_date desc
```

Result 1

invoice_id	fname	lname	passport_number	invoice_amount	invoice_date	flight_id
1000015	Zhang	Jack	959383825	867.82	2022-01-06 00:30:00	1000006
1000027	River	Brakus	483828405	270.18	2021-12-28 00:00:00	1000023
1000004	Zhang	Jack	959383825	943.25	2021-12-22 00:00:00	1000027
1000001	River	Brakus	483828405	639.14	2021-12-17 08:00:00	1000013
1000002	Yang	Kuhlman	482099981	937.00	2021-10-10 00:00:00	1000019
1000026	Zhang	Jack	959383825	579.89	2021-09-03 00:00:00	1000012
1000028	Ramon	Padberg	492838591	170.94	2021-08-30 00:00:00	1000018
1000023	Carly	Ma	3848475123	772.18	2021-08-27 00:00:00	1000003
1000005	Yang	Kuhlman	482099981	714.53	2021-07-16 00:00:00	1000014
1000011	Jeff	Smith	68583231	743.40	2021-07-01 00:00:00	1000025
1000020	Floyd	Connelly	4837571	161.27	2021-05-15 00:00:00	1000021

This query shows all the invoices information for a customer whose id is 37. This customer can check all the passengers' information from him, like how much this passenger spend, what's the flight id of this passenger.

## 2)Multi-row subquery

The screenshot shows a MySQL Workbench interface with a query editor and a results pane. The query is:

```
1 select email,first_name,last_name from users_customuser
2 where zipcode = (
3   select zipcode from users_customuser where email = 'yz6912@nyu.edu')
4
```

The results pane displays a table with columns: email, first\_name, last\_name. The data is:

email	first_name	last_name
yz6912@nyu.edu	Cris	Yang
yz69125@nyu.edu	Sherry	Zi
yz6905@nyu.edu	Zhang	Wen

Search who has the same zip code as user '[yz6912@nyu.edu](#)'

## 3)Correlated subquery

The screenshot shows a MySQL Workbench interface with a query editor and a results pane. The query is:

```
1 SELECT *FROM ywry_payment pay
2 WHERE payment_amount = (
3   SELECT MAX(payment_amount)
4   FROM ywry_payment
5   WHERE customer_id = pay.customer_id);
6
```

The results pane displays a table with columns: payment\_date, payment\_amount, method, card\_number, name\_on\_card, expire\_date, invoice\_id, customer\_id. The data is:

payment_date	payment_amount	method	card_number	name_on_card	expire_date	invoice_id	customer_id
2021-09-16 00:00:00	620.07	Credit Card	3176206260	Melyssa Waters	2024-05-06 00:00:00	1000019	1000025
2021-07-04 00:00:00	510.65	Credit Card	6792968314	Mrs. Novella Hamill PhD	2029-03-22 00:00:00	1000013	1000022
2021-02-21 00:00:00	243.63	Debit	3918963619	Desmond Mertz	2029-06-13 00:00:00	1000001	1000012
2021-09-09 00:00:00	185.68	Debit	7848785162	Mr. Orion Bayer	2024-07-13 00:00:00	1000004	1000026
2021-04-17 00:00:00	234.01	Debit	4086681532	Dr. Arnaldo Ziemo V	2024-08-06 00:00:00	1000004	1000013
2021-07-21 00:00:00	970.00	Credit Card	117934703	Miss Donna Raynor	2024-09-21 00:00:00	1000011	1000000
2021-02-14 00:00:00	715.43	Debit	6083640823	Dr. Bunice Bauch MD	2024-06-12 00:00:00	1000014	1000020
2021-05-11 00:00:00	629.32	Credit Card	6239810544	Willie Brown	2024-02-10 00:00:00	1000020	1000017
2021-07-27 00:00:00	163.89	Credit Card	6573271449	Bryon White	2024-10-29 00:00:00	1000016	1000019
2021-02-02 00:00:00	369.52	Debit	7569543881	Sallie Mertz	2024-07-12 00:00:00	1000022	1000021
2021-12-30 00:00:00	101.12	Debit	3190713834	Susana Bruen	2024-08-29 00:00:00	1000015	1000018
2021-01-31 00:00:00	688.68	Credit Card	5082444545	Amir Braun	2024-06-12 00:00:00	1000015	1000027

Find the biggest once payment amount of each customer

#### 4)SET operator query

```
1 select count(*) as different_customer_fname from
2 (select first_name from users_customuser
3 UNION
4 select first_name from users_customuser) customer
5
6
```

different_customer_fname
9

How many different first name customers have of all customers

#### 5)Query with in line view or WITH clause

```
1 WITH temporaryTable(avgValue) as
2 (SELECT avg(payment_amount)
3   from ywry_payment)
4   SELECT *
5     FROM ywry_payment, temporaryTable
6    WHERE ywry_payment.payment_amount > temporaryTable.avgValue;
```

payment_date	payment_amount	method	card_number	name_on_card	expire_date	invoice_id	customer_id	avgValue
2021-09-16 00:00:00	620.07	Credit Card	3176206260	Melyssa Waters	2024-05-06 00:00:00	1000019	1000025	505.541667
2021-07-04 00:00:00	510.65	Credit Card	6792968314	Mrs. Novella Hamill PhD	2029-03-22 00:00:00	1000013	1000022	505.541667
2021-07-21 00:00:00	970.00	Credit Card	1177934703	Miss Donna Raynor	2024-09-21 00:00:00	1000011	1000000	505.541667
2021-02-14 00:00:00	715.43	Debit	6083640823	Dr. Burnice Bauch MD	2024-06-12 00:00:00	1000014	1000020	505.541667
2021-05-11 00:00:00	629.32	Credit Card	6239510544	Willie Brown	2024-02-10 00:00:00	1000020	1000017	505.541667
2021-01-31 00:00:00	688.96	Credit Card	5982444455	Amir Braun	2024-05-13 00:00:00	1000015	1000027	505.541667
2021-08-15 00:00:00	971.00	Credit Card	1493238272	Cleve Rutherford IV	2024-05-23 00:00:00	1000022	1000028	505.541667
2021-12-20 00:00:00	977.63	Debit	6360252395	Prof. April Connell V	2024-01-08 00:00:00	1000002	1000009	505.541667
2021-04-10 00:00:00	664.68	Debit	6750541274	Sydney Wisoky Jr.	2024-01-01 00:00:00	1000000	1000008	505.541667

List the payment record whose payment amount is greater than the average payment amount

## 6)TOP-N query

The screenshot shows a MySQL command-line interface with the following details:

- Session path: root / website2
- Query entered:

```
1 select sum(invoice_amount) as amount,ywry_passenger.passenger_id,ywry_passenger.fname,ywry_passenger.lname,count(*)  
2      from ywry_passenger, ywry_invoice where ywry_passenger.passenger_id = ywry_invoice.passenger_id  
3      GROUP BY ywry_passenger.passenger_id ORDER BY amount DESC limit 10
```
- Result table:

amount	passenger_id	fname	lname	count(*)
3275.74	1000003	Zhang	Jack	4
2217.87	1000000	Yang	Kuhlman	3
1719.01	1000008	Jeff	Smith	2
1606.45	1000018	Zin	Shin	3
1410.35	1000015	Brock	Jakubowski	2
909.32	1000007	River	Brakus	2
775.89	1000017	Dilly	Dal	1
772.18	1000005	Carly	Ma	1
688.47	1000019	Jaron	Lockman	3
623.72	1000016	Peggie	Donnelly	3
- Result tab selected: Result 1

Select the top 10 spenders