Project Title: ETSY-4

Course number and section number: CS6360.003

Team number: 24

Team member: Jinglun Zhang with Netid: jxz200002

Data requirements for ETSY:

- The ETSY web_users in ETSY are identified by unique u_id, there are two unique attributes except the primary key which are login name and email. Other attributes include birthday, status, member date, password, description, gender, first name, and last name. Each user has a list of his/her favorite materials. Also, information for the credit cards the user wish to user can also be saved. The attributes for the credit cards include zip, country, street address, city, states, expiration date, and owner name. A user can send messages to other users, each massage has a date and texts related to it. The message date of messages between two users should be unique for each message. User also can have a list of favorite users.
- A web_user can have several user addresses, the user address is identified by a unique user_address_id, other attributes of user_address include receiver_name, first_line of address, second line of address, city, states, zip code, country and if this address is default address of its user.
- A web_user can have one account of ETSY_Credit, each ETSY_Credit has a unique credit_id and a balance related to this account.
- Web_Users are categorized to seller and buyer based on the behavior. A web_user can be a buyer and a seller at the same time.
- A Seller have a bank_account associated to him/her to receive money. The attributes of bank_account of seller include account_number, name on the account, the type of the account, and the routing number of the account. Seller will also have a unique SSN number. A seller have a derived attribute which is the sale amount earned by the seller.
- A seller can open several shops on ETSY. A shop must be owned by a seller. Shops are identified a unique shop id. Other attributes of shop include open_year, description, and the name of the shop.
- A shop can have list several items. Item is called listing on ETSY and each listing must be listed by a shop. Listing is identified by a unique listing id, other attributes of listing include states, title, description, creation_date, ending_date if the listing is no longer available, renewal option, weight, processing time , country, and the inventory information about the list include price and stock. List can have several tags, categories, and materials related to it.
- A user can have a list of favorite shops.
- Listing can have several shipping information related to it. Each shipping information is identified by a unique id. The shipping information have attributes as destination country and primary_cost which record the shipping fee for shipping one listing to destination country.
- A user can have a list of favorite listings.
- Listing are categorized to craft supplies, vintage item, and handmade item. A listing must belong to at least one of the three subtypes.
- A craft supply will have a purpose for this item to be used.

- A vintage item will include age and certification attribute. A vintage item should be at least 20 years old.
- A handmade can have a production partner if the seller can not produce the item by himself. The attributes of production partner include description of the parter, the reason for having partner, the role of the seller, the role of the partner, the name of the partner, the city, state, and country of the partner.
- A buyer will have a derived attribute which record the amount of money he has spent on ETSY. A buyer can have several shopping carts with one shopping cart corresponds to one seller. Each shopping cart will have the destination_country of the cart, two derived attribute which record the total cost of the cart and the discount of the cart.
- A shopping cart can have several cart lists which record the items included in the cart. The cart list also records the quantity and the price of the listing added to the cart.
- A buyer can place several orders. Orders are identified by a unique order_id. The information of the order includes the shipping address of the order which include street address, zip, state, city, country and receiver. Other information includes the date of the order, the payment method, payment account, and the status of the order. A derived attribute of the order is the total cost of the order.
- An order is separated to several receipt with one receipt corresponds to one shop. The system will also record the status of the receipt. A derived attribute of receipt is the cost of the receipt.
- Each receipt will contain several transactions which correspond to one listing in the receipt. The transaction will have the information about the status of the transaction, the shipped_date of the transaction, the quantity, feedback regarding this transaction and the purchase price of the listing.

u_id
birthday
status
login_name
email
member_date
password
description
gender
favorite_materials
name
first_name
last_name
billing_address
zip
country
street_address
city
states
card_number
expiration_date
owner_name
credit_cards
User_Address
user_address_id
receiver_name
first_line
second_line
city
states
zip
country
is_default
1
Has
N
N
Favorite_User
M
message_date
text
N
Message
M
credit_id
balance
ETSY_Credit
1
credit
1
Paypal
Web_User
destination_country
total_cost
discount
1
Cart
N
Shopping_Cart
o
behavior
Cart
N
1
street
zip
states
city
country
receiver
shipping
order_id
total_cost
order_date
payment
status
method
account_no
transaction_sold_count
account_number
account_full_name
account_type
routing_number
bank_account
Seller
ssn
transaction_buy_count
Buyer
1
Place_Order
N
Orders
M
Favorite_Listing
N
1
Receipt_create
N
1
Receipt_create
N
Receipt
cost
status
1
Own
N
open_year
description
shop_id
name
Shop
M
Favorite_Shop
N
N
Transaction
M
status
shipped_date
quantity
feedback
price
1
List
N
quantity
price_sold
N
Cart_List
M
listing_id
states
title
description
creation_date
ending_date
stock
price
Inventory_And_Pricing
listing_tag
listing_category
listing_material
renewal_option
weight
processing_time
country
Listing
1
Shipping_Information
N
Shipping_Info
shipping_info_id
destination_country
primary_cost
type
o
Craft_Supplies
Vintage_Item
Handmade_Item
purpose
age
certification
1
Producer
1
Production_Partner
description
partnership
reason
seller_role
partner_role
name
location
city
states
country

Web_User

u_id	birthday	status	login_name	email	member_date	password	description	gender	first_name	last_name

User_Material

u_id	*description*

u_id to Web_User.u_id

Paypal

u_id	account_number

u_id to Web_User.u_id

Credit_Cards

u_id	card_number	zip	country	city	state	expiration_date	owner_name

u_id to Web_User.u_id

User_Addresses u_id to Web_User.u_id

u_id	user_address_id	receiver_name	first_line	second_line	city	state	zip	country	is_default

Favorite_User

like_user_id	liked_user_id

like_user_id, liked_user_id to Web_User.u_id

Message

sender_id	receiver_id	message_date	text

sender_id, receiver_id to Web_User.u_id

Seller

u_id	ssn	account_number	account_full_name	account_type	routing_number

u_id to Web_user.u_id

Shop

shop_id	open_year	description	name	owner_id

owner_id to Seller.u_id

Favorite_Shop

u_id	shop_id

u_id to Web_User.u_id

Listing

listing_id	state	title	description	creation_date	ending_date	stock	price	renewal_option	weight	processing_time	shop_id	country

Shipping_Information

shipping_info_id	destination_country	primary_cost	listing_id

listing_id to Listing.listing_id

Buyer

u_id

u_id to Web_User.u_id

Shopping_Cart

buyer_id	seller_id	destination_country

buyer_id to Buyer.u_id

Cart_List

buyer_id	seller_id	listing_id	quantity	price_sold

Favorite_Lisitng

u_id	listing_id

u_id to Web_User.u_id, listing_id to Listing.listing_id

ETSY_Credit

balance	credit_id	u_id

Transactions

price	feedback	quantity	shipped_date	status	listing_id	shop_id	order_id

Receipt

	status	shop_id	order_id

u_id to Buyer.u_id

Orders

u_id	receiver	city	states	zip	street	account_no	method	order_date	status	order_id

Listing_Tag

tag_description	listing_id

Listing_Category

category_description	listing_id

Listing_Material

material_description	listing_id

Craft_Supplies

purpose	listing_id

Vintage_Item

certification	age	listing_id

Handmade_Item

listing_id

Production_Partner

listing_id	name	description	reason	seller_role	partner_role	city	states	country

Normal Form Analysis:

All non-prime attributes are fully dependent on every key, no violations of 2nd normal form

Web_User table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

User_Material table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. No non-prime attributes thus no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Paypal table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. No non-prime attributes thus no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Credit_Cards table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Zip, country, city, state, expiration_date and owner_name attributes are partially dependent on card_number which violate 2nd normal form. This table need to be normalized by separated into two tables.

User_Address table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Favorite_User table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. No non-prime attributes thus no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Message Table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. All non-prime attributes are fully dependent on every key, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Seller Table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Shop Table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Favorite Shop table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. No non-prime attributes thus no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Listing table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Shipping_Information table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Buyer table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Shopping_Cart table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. All non-prime attributes are fully dependent on every key, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Cart_List table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Although price_sold looks partially dependent on listing_id, the sold price can be different from what listed in the listing table, only knowing listing_id can't determine the price sold. All non-prime attributes are fully dependent on every key, no violations of 2nd normal form. Although seller_id can be indirectly determined by listing_id, this is not a violation of 3rd normal form since seller_id is a prime attribute.

ETSY_Credit table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Every non-prime attribute is fully functionally dependent on every key, no violation of 2NF. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Transaction table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Although price seems partially dependent on listing_id, the sold price can be different from what listed in the listing table, only knowing listing_id can't determine the price sold. All non-prime attributes are fully dependent on every key, no violations of 2nd normal form. Although shop_id can be determined by listing_id, there is no violation of 3rd normal form since shop_id is a prime attribute.

Receipt table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. All non-prime attributes are fully dependent on every key, no violations of 2nd normal form. Although shop_id functional dependent on order_id, there is no violation of 3rd normal form since shop_id is a prime attribute.

Orders table: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Listing Tag, Listing_Category, and Listing_Material tables: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. No non-prime attributes thus no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Craft_Supplies, Vintage_Item, and Handmade_Item, and Production_Partner tables: No composite attribute, multivalued attributes, and nested relations, no violations of 1st normal form. Only one prime

attribute, no violations of 2nd normal form. For every FD X -> A in the table, either X is a superkey of R or A is a prime attribute, thus no violation of 3rd normal form.

Web_User

u_id	birthday	status	login_name	email	member_date	password	description	gender	first_name	last_name

User_Material

u_id	*description*

u_id to Web_User.u_id

Paypal

u_id	account_number

u_id to Web_User.u_id

User_Credit_Cards u_id to Web_User.u_id

u_id	card_number

Credit_Card

card_number	zip	country	city	states	expiration_date	owner_name	street_address

User_Address u_id to Web_User.u_id

u_id	user_address_id	receiver_name	first_line	second_line	city	states	zip	country	is_default

Favorite_User

like_user_id	liked_user_id

like_user_id, liked_user_id to Web_User.u_id

Message

sender_id	receiver_id	message_date	text

sender_id, receiver_id to Web_User.u_id

Seller

u_id	ssn	account_number	account_full_name	account_type	routing_number

u_id to Web_user.u_id

Shop

shop_id	open_year	description	name	owner_id

owner_id to Seller.u_id

Favorite_Shop

u_id	shop_id

u_id to Web_User.u_id

Listing

listing_id	states	title	description	creation_date	ending_date	stock	price	renewal_option	weight	processing_time	shop_id	country

Shipping_Information

shipping_info_id	destination_country	primary_cost	listing_id

listing_id to Listing.listing_id

Buyer

u_id

u_id to Web_User.u_id

Shopping_Cart

buyer_id	seller_id	destination_country

buyer_id to Buyer.u_id

Cart_List

buyer_id	seller_id	listing_id	quantity	price_sold

Favorite_Lisitng

u_id	listing_id

u_id to Web_User.u_id, listing_id to Listing.listing_id

Favorite_Listing

listing_id	u_id

ETSY_Credit

balance	credit_id	u_id

Transactions

price	feedback	quantity	shipped_date	status	listing_id	shop_id	order_id

Receipt

	status	shop_id	order_id

u_id to Buyer.u_id

Orders

country	u_id	receiver	city	states	zip	street	account_no	method	order_date	status	order_id

Listing_Tag

tag_description	listing_id

Listing_Category

category_description	listing_id

Listing_Material

material_description	listing_id

Craft_Supplies

purpose	listing_id

Vintage_Item

certification	age	listing_id

Handmade_Item

listing_id

Production_Partner

listing_id	name	description	reason	seller_role	partner_role	city	states	country

```
/*SQL commands to create the tables*/
create table Web_User(
      u_id integer,
      birthday date,
      status varchar(10),
      login_name varchar(10) not null unique,
      email varchar(50) not null unique,
      member_date date,
      password varchar(20),
      description varchar(300),
      gender char,
      first_name varchar(20),
      last_name varchar(20),
      primary key (u_id)
);

create table User_Material(
      u_id integer,
      description varchar(50),
      primary key (u_id, description)
);

create table Paypal(
      u_id integer,
      account_name varchar(30),
      primary key (u_id, account_name)
);

create table User_Credit_Cards(
      u_id integer,
      card_number integer,
      primary key (u_id, card_number)
);

create table Credit_Card(
      card_number integer,
      zip varchar(10) not null,
      country varchar(20) not null,
      city varchar(20) not null,
      states varchar(20) not null,
      expiration_date date not null,
      owner_name varchar(20) not null,
      primary key (card_number)
);

create table User_Address(
      u_id integer,
      user_address_id integer,
      receiver_name varchar(20) not null,
      first_line varchar(50) not null,
      second_line varchar(50),
      states varchar(20) not null,
      zip varchar(10) not null,
      country varchar(20) not null,
      is_default char(1),
      primary key (user_address_id)
);

create table Favorite_User(
```

```
      like_user_id integer,
      liked_user_id integer,
      primary key (like_user_id, liked_user_id)
);

create table Message(
      sender_id integer,
      receiver_id integer,
      message_date date,
      texts varchar(200) not null,
      primary key (sender_id, receiver_id, message_date)
);

create table Seller(
      u_id integer,
      ssn char(9) not null unique,
      account_number varchar(20) not null,
      account_full_name varchar(20) not null,
      account_type varchar(10) not null,
      routing_number varchar(20) not null,
      primary key (u_id)
);

create table Shop(
      shop_id integer,
      open_year char(4),
      description varchar(200),
      name varchar(20),
      owner_id integer,
      primary key (shop_id)
);

create table Favorite_Shop(
      u_id integer,
      shop_id integer,
      primary key (u_id, shop_id)
);

create table Listing(
      listing_id integer,
      states varchar(10),
      title varchar(30) not null,
      description varchar(200),
      creation_date date not null,
      ending_date date,
      stock integer not null,
      price decimal not null,
      renewal_option varchar (10),
      weight decimal,
      process_time integer,
      shop_id integer,
      country varchar(20) not null,
      primary key (listing_id)
);

create table Shipping_Information(
      shipping_info_id integer,
      destination_country varchar(20) not null,
      primary_cost decimal,
```

```
      listing_id integer,
      primary key (shipping_info_id)
);

create table Buyer(
      u_id integer,
      primary key (u_id)
);

create table Shopping_Cart(
      buyer_id integer,
      seller_id integer,
      destination_country varchar(20) not null,
      primary key (buyer_id, seller_id)
);

create table Cart_List(
      buyer_id integer,
      seller_id integer,
      listing_id integer,
      quantity integer not null,
      price_sold decimal not null,
      primary key (buyer_id, seller_id, listing_id)
);

create table Orders(
      order_id integer,
      status varchar(20),
      order_date date not null,
      method varchar(10) not null,
      account_no varchar(20) not null,
      street varchar(20) not null,
      zip varchar (20) not null,
      states varchar(20) not null,
      city varchar(20) not null,
      receiver varchar(20) not null,
      u_id integer,
      country varchar(20) not null,
      primary key (order_id)
);

create table Receipt(
      order_id integer,
      shop_id integer,
      status varchar(10),
      primary key (order_id, shop_id)
);

create table Transactions(
      order_id integer,
      shop_id integer,
      listing_id integer,
      status varchar(20),
      shipped_date date,
      quantity integer not null,
      feedback varchar(200),
      price decimal not null,
      primary key (order_id, shop_id, listing_id)
);
```

```
create table ETSY_Credit(
      u_id integer,
      credit_id integer,
      balance decimal,
      primary key (credit_id)
);

create table Favorite_Listing(
      u_id integer,
      listing_id integer,
      primary key (u_id, listing_id)
);

create table Listing_Tag(
      listing_id integer,
      tag_description varchar(20),
      primary key (listing_id, tag_description)
);

create table Listing_Category(
      listing_id integer,
      category_description varchar(20),
      primary key (listing_id, category_description)
);

create table Listing_Material(
      listing_id integer,
      material_description varchar(20),
      primary key (listing_id, material_description)
);

create table Craft_Supplies(
      listing_id integer,
      purpose varchar(20) not null,
      primary key (listing_id)
);

create table Vintage_Item(
      listing_id integer,
      age integer,
      certification varchar(100) not null,
      primary key (listing_id)
);

create table Handmade_Item(
      listing_id integer,
      primary key (listing_id)
);

create table Production_Partner(
      listing_id integer,
      name varchar(20) not null,
      description varchar(50),
      reason varchar(50),
      seller_role varchar(50) not null,
      partner_role varchar(50) not null,
      city varchar(20) not null,
      states varchar(20) not null,
```

```
      country varchar(20) not null,
      primary key (listing_id)
);

/*Foreign key constraints*/
alter table User_Material add constraint fkum foreign key (u_id) references
Web_User(u_id) on delete cascade;
alter table Paypal add constraint fkp foreign key (u_id) references Web_User(u_id)
on delete cascade;
alter table User_Credit_Cards add constraint fkucc foreign key (u_id) references
Web_User(u_id) on delete cascade;
alter table User_Credit_Cards add constraint fkuccc foreign key (card_number)
references Credit_Card(card_number) on delete cascade;
alter table User_Address add constraint fkua foreign key (u_id) references
Web_User(u_id) on delete cascade;
alter table Favorite_User add constraint fkfulike foreign key (like_user_id)
references Web_User(u_id) on delete cascade;
alter table Favorite_User add constraint fkfuliked foreign key (liked_user_id)
references Web_User(u_id) on delete cascade;
alter table Message add constraint fkmsender foreign key (sender_id) references
Web_User(u_id) on delete cascade;
alter table Message add constraint fkmreceiver foreign key (receiver_id) references
Web_User(u_id) on delete set null;
alter table Seller add constraint fks foreign key (u_id) references Web_User(u_id)
on delete cascade;
alter table Shop add constraint fksp foreign key (owner_id) references Seller(u_id)
on delete cascade;
alter table Favorite_Shop add constraint fkfsu foreign key (u_id) references
Web_User(u_id) on delete cascade;
alter table Favorite_Shop add constraint fkfss foreign key (shop_id) references
Shop(shop_id) on delete cascade;
alter table Listing add constraint fkl foreign key (shop_id) references
Shop(shop_id) on delete cascade;
alter table Shipping_Information add constraint fksi foreign key (listing_id)
references Listing(listing_id) on delete cascade;
alter table Buyer add constraint fkb foreign key (u_id) references Web_User(u_id)
on delete cascade;
alter table Shopping_Cart add constraint fkscb foreign key (buyer_id) references
Buyer(u_id) on delete cascade;
alter table Shopping_Cart add constraint fkscs foreign key (seller_id) references
Seller(u_id) on delete cascade;
alter table Cart_List add constraint fkclbs foreign key (buyer_id, seller_id)
references Shopping_Cart(buyer_id, seller_id) on delete cascade;
alter table Cart_List add constraint fkcll foreign key (listing_id) references
Listing(listing_id) on delete cascade;
alter table Favorite_Listing add constraint fkflu foreign key (u_id) references
Web_User(u_id) on delete cascade;
alter table Favorite_Listing add constraint fkfll foreign key (listing_id)
references Listing(listing_id) on delete cascade;
alter table ETSY_Credit add constraint fkec foreign key (u_id) references
Web_User(u_id) on delete cascade;
alter table Orders add constraint fko foreign key (u_id) references Buyer(u_id) on
delete cascade;
alter table Receipt add constraint fkro foreign key (order_id) references
Orders(order_id);
alter table Receipt add constraint fkrs foreign key (shop_id) references
Shop(shop_id) on delete set null;
alter table Transactions add constraint fkto foreign key (order_id, shop_id)
references Receipt(order_id, shop_id) on delete set null;
```

```
alter table Transactions add constraint fktl foreign key (listing_id) references
Listing(listing_id) on delete set null;
alter table Listing_Tag add constraint fklt foreign key (listing_id) references
Listing(listing_id) on delete cascade;
alter table Listing_Category add constraint fklc foreign key (listing_id)
references Listing(listing_id) on delete cascade;
alter table Listing_Material add constraint fklm foreign key (listing_id)
references Listing(listing_id) on delete cascade;
alter table Craft_Supplies add constraint fkcs foreign key (listing_id) references
Listing(listing_id) on delete cascade;
alter table Vintage_Item add constraint fkvi foreign key (listing_id) references
Listing(listing_id) on delete cascade;
alter table Handmade_Item add constraint fkhi foreign key (listing_id) references
Listing(listing_id) on delete cascade;
alter table Production_Partner add constraint fkpp foreign key (listing_id)
references Handmade_Item(listing_id) on delete cascade;
```

```
/*Place an order*/
create or replace procedure place_order(b_id in integer, o_id in integer, o_status
in Orders.status%type, o_date in date, o_method in Orders.method%type, o_account in
Orders.account_no%type, o_street in Orders.street%type, o_zip in Orders.zip%type,
o_state in Orders.states%type, o_city in Orders.city%type, o_receiver in
Orders.receiver%type, o_country in Orders.country%type) as

cursor cart_items is
select *
from Cart_List
where buyer_id = b_id;

current_item Cart_List%rowtype;
current_list integer;
current_shop integer;
item_counter integer;
buyer_counter integer;
shop_counter integer;

begin
      select count(*) into buyer_counter from Shopping_Cart where buyer_id = b_id;
      if(buyer_counter = 0) then
            RAISE_APPLICATION_ERROR(-20000, 'The shopping cart of this user is
empty, can not place order');
      end if;
      select count(*) into item_counter from Cart_List where buyer_id = b_id;
      if(item_counter = 0) then
            delete from Shopping_Cart where buyer_id = b_id;
            RAISE_APPLICATION_ERROR(-20000, 'The shopping cart of this user is
empty, can not place order');
      end if;
      insert into Orders values(o_id, o_status, o_date, o_method, o_account,
o_street, o_zip, o_state, o_city, o_receiver, b_id, o_country);
      open cart_items;
            loop
                  fetch cart_items into current_item;
                  exit when (cart_items%notfound);
                  current_list := current_item.listing_id;
                  select shop_id into current_shop from Listing where listing_id =
current_list;
                  select count(*) into shop_counter from Receipt where order_id =
o_id and shop_id = current_shop;
                  if(shop_counter = 0) then
                        insert into Receipt values (o_id, current_shop,
'Initiate');
                  end if;
                  insert into Transactions values (o_id, current_shop,
current_list, 'Initiate', '', current_item.quantity, ' ', current_item.price_sold);
            end loop;
      close cart_items;

      /*Clear the shopping carts and cart_lists of the check out user*/
      delete from Shopping_Cart where buyer_id = b_id;
      delete from Cart_List where buyer_id = b_id;
end;
/

/*Procedure to get the default shipping contry of the user, if no default, return
'USA'*/
```

```
create or replace procedure default_country(us_id in integer, result_country out
User_Address.country%type) as

cursor us_addresses is
select *
from user_address
where u_id = us_id;

current_address User_Address%rowtype;

begin
      result_country := 'USA';
      open us_addresses;
            loop
                  fetch us_addresses into current_address;
                  exit when(us_addresses%notfound);
                  if(current_address.is_default = 'Y') then
                        result_country := current_address.country;
                  end if;
            end loop;
      close us_addresses;
end;
/

/*Procedure to give a discount to all the items in one user's shopping cart*/
create or replace procedure discount_cart(b_id in integer, discount in decimal) as

cursor cart_items is
select *
from Cart_List
where buyer_id = b_id
for update;

current_item Cart_List%rowtype;

begin
      open cart_items;
            loop
                  fetch cart_items into current_item;
                  exit when (cart_items%notfound);
                  update Cart_List set price_sold = price_sold * discount where
current of cart_items;
            end loop;
      close cart_items;
end;
/

/*Proceture to add item to cart_list, also create entry in Shopping_Cart based on
differnet conditions*/
create or replace procedure add_to_cart(b_id in integer, l_id in integer, buy_num
in integer, buy_price in decimal) as

se_id integer;
so_id integer;
s_c_count integer;
def_country varchar(20);

begin
      def_country := 'USA';
```

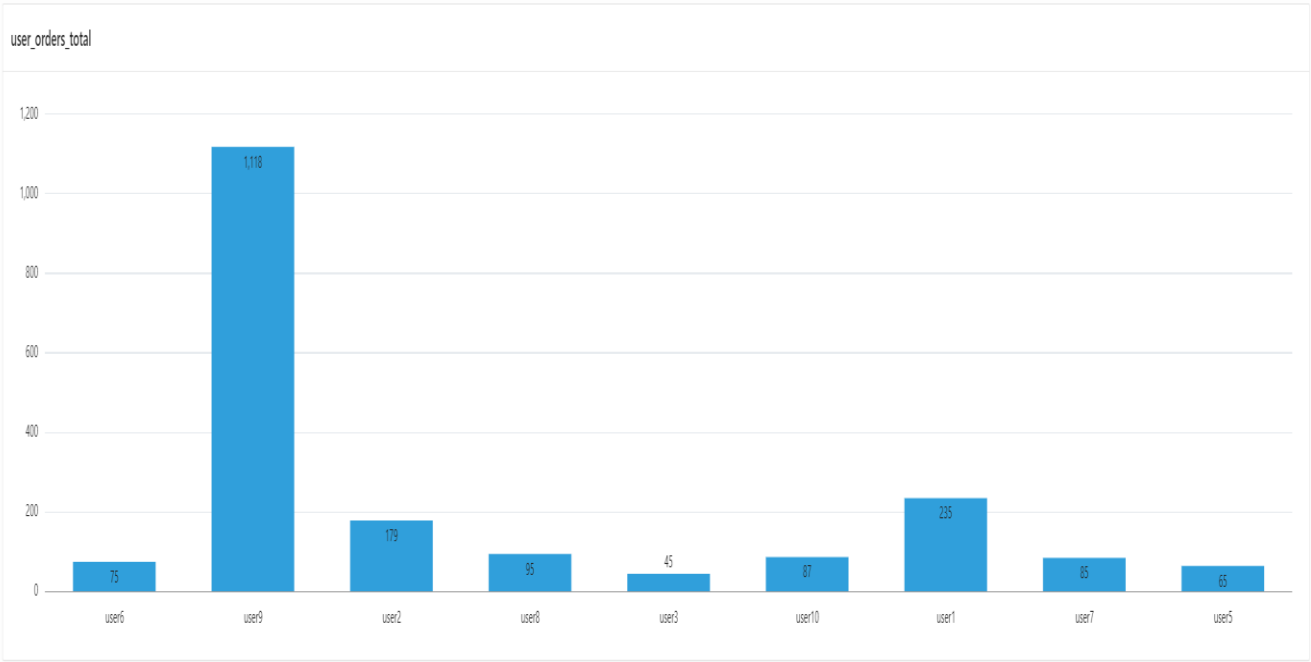
```
      /*Get the shop_id of the listing*/
      select shop_id into so_id from listing where listing_id = l_id;
      /*Get the seller_id of the shop(listing)*/
      select owner_id into se_id from Shop where shop_id = so_id;
      /*Check if already have the entry for the seller and buyer in Shopping_Cart
table. If not, insert a new entry to the Shopping_Cart table for the buyer and
seller*/
      select count(*) into s_c_count from Shopping_Cart where buyer_id = b_id and
seller_id = se_id;
      if(s_c_count = 0) then
            default_country(b_id, def_country);
            insert into Shopping_Cart values (b_id, se_id, def_country);
      end if;
      insert into Cart_List values(b_id, se_id, l_id, buy_num, buy_price);
end;
/
```

```
/*Compare the quantity of the purchased item in stock to the quantity added to the
cart, raise an error if the purchase quantity is bigger than the  quantity of the
item in stock.*/
create or replace trigger check_stock_cart
before insert or update of quantity on Cart_List
for each row

declare
      stock_num Cart_List.quantity%type;

begin
            select stock into stock_num from Listing where listing_id
= :new.listing_id;
            if(stock_num < :new.quantity) then
                  RAISE_APPLICATION_ERROR(-20000, 'Do not have enough stock for
this shopping cart.');
            end if;
end;
/

/*Compare the quantity of the purchased item in stock to the quantity added in the
transaction, raise an error if the purchase quantity is bigger than the  quantity
of the item in stock.*/
create or replace trigger check_stock_transaction
before insert or delete or update of quantity on Transactions
for each row

declare
      stock_num Transactions.quantity%type;
      diff integer;

begin
      if DELETING then
            update Listing set stock = stock + :old.quantity where listing_id
= :old.listing_id;
      else
            select stock into stock_num from Listing where listing_id
= :new.listing_id;
            diff := :new.quantity - :old.quantity;
            if(stock_num < diff) then
                  RAISE_APPLICATION_ERROR(-20000, 'Do not have enough stock for the
order.');
            else
                  update Listing
                  set stock = stock - diff
                  where listing_id = :new.listing_id;
            end if;
      end if;
end;
/

/*When add to cart_list of shopping_cart, insert a new row to shopping_cart table
if not existing. check if the item can be shipped to their country*/
create or replace trigger check_destination_cart
before insert or update of listing_id on Cart_List
for each row

declare
      desti Shopping_Cart.destination_country%type;
```

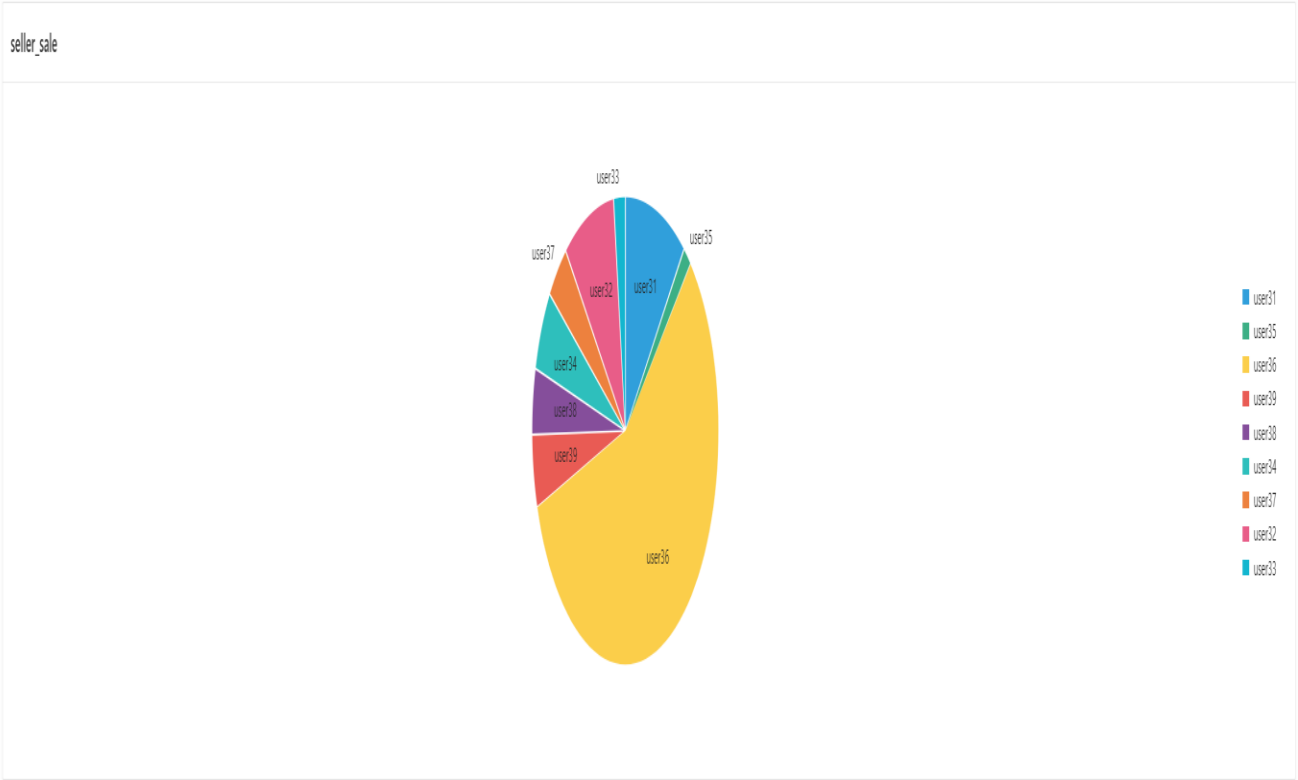
```
      resultCounter integer;
      scCounter integer;
begin
      select count(*) into scCounter from Shopping_Cart where buyer_id
= :new.buyer_id and seller_id = :new.seller_id;
      if(scCounter = 0) then
            insert into Shopping_Cart values (:new.buyer_id, :new.seller_id,
'USA');
      end if;
      select destination_country into desti from Shopping_Cart where buyer_id
= :new.buyer_id and seller_id = :new.seller_id;
      select count(*) into resultCounter from Shipping_Information where listing_id
= :new.listing_id and destination_country = desti;
      if(resultCounter = 0) then
            RAISE_APPLICATION_ERROR(-20000, 'This item can not be shipped to your
country!');
      end if;
end;
/

/*When update the destination_country of their order, check if the item can be
shipped to the new country*/
create or replace trigger check_destination_order
before insert or update of country on Orders
for each row

declare
      current_listing Listing.listing_id%type;
      cursor listing_list is select listing_id from transactions where order_id
= :new.order_id;
      resultCounter integer;
begin
      open listing_list;
            loop
                  fetch listing_list into current_listing;
                  exit when (listing_list%notfound);
                  select count(*) into resultCounter from Shipping_Information
where listing_id = current_listing and destination_country = :new.country;
                  if(resultCounter = 0) then
                        RAISE_APPLICATION_ERROR(-20000, 'Item can not be shipped to
the new destination.');
                  end if;
            end loop;
      close listing_list;
end;
/

/*when add vintage_item, make sure that the age of this item is at least 20 years
old*/
create or replace trigger vintage_age
before insert or update of age on Vintage_Item
for each row

begin
      if(:new.age < 20) then
            RAISE_APPLICATION_ERROR(-20000, 'Vintage item should be at least 20
years old.');
      end if;
end;
```

/

Buyer_order_total: Chart which shows the total money each user has spent.



Seller_sale: Chart which shows how much sale made by each seller.



Etsy_user: Interactive Grid which shows the information about users in the system.

ETSY_Users_Interactive Grid

Search: All Text Columns | Go | Actions | Edit | Save | Add Row | Reset

User Name	Email	Member Date	Status	First Name	Last Name	Gender	Description	Birthday
user1	user1@gmail.com	7/1/2020	active	Andy	Vile	M	normal user	6/21/1944
user2	user2@gmail.com	7/2/2020	active	Brad	Knight	M	normal user	2/13/1968
user3	user3@gmail.com	7/3/2020	active	Evan	Wallis	M	normal user	1/16/1958
user4	user4@gmail.com	7/4/2020	active	Josh	Zell	M	normal user	5/22/1954
user5	user5@gmail.com	7/5/2020	active	Jared	James	F	normal user	10/10/1966
user6	user6@gmail.com	7/6/2020	active	Justin	Mark	F	normal user	1/12/1966
user7	user7@gmail.com	7/7/2020	active	Jon	Jones	F	normal user	11/14/1967
user8	user8@gmail.com	7/8/2020	active	John	James	F	normal user	6/30/1975
user9	user9@gmail.com	7/9/2020	active	Alex	Freed	F	normal user	10/9/1950
user10	user10@gmail.com	7/10/2020	active	Ahmad	Jabbar	M	normal user	3/29/1959
user11	user11@gmail.com	7/11/2020	active	Joyce	English	M	normal user	7/31/1962
user12	user12@gmail.com	7/12/2020	active	Ramesh	Narayan	M	normal user	9/15/1952
user13	user13@gmail.com	7/13/2020	active	Alicia	Zelaya	M	normal user	7/19/1958
user14	user14@gmail.com	7/14/2020	active	John	Smith	F	normal user	1/9/1955
user15	user15@gmail.com	7/15/2020	active	Jennifer	Wallace	F	normal user	6/20/1931
user16	user16@gmail.com	7/16/2020	active	Franklin	Wong	F	normal user	12/8/1945
user17	user17@gmail.com	7/17/2020	active	James	Borg	F	normal user	11/10/1927
user18	user18@gmail.com	7/18/2020	active	Tom	Brand	F	normal user	12/16/1966
user19	user19@gmail.com	7/19/2020	active	Jenny	Vos	M	normal user	11/11/1967
user20	user20@gmail.com	7/20/2020	active	Chris	Carter	M	normal user	3/21/1960
user21	user21@gmail.com	7/21/2020	active	Kim	Grace	M	normal user	10/23/1970
user22	user22@gmail.com	7/22/2020	active	Jeff	Chase	M	normal user	1/7/1970
user23	user23@gmail.com	7/23/2020	active	Bonnie	Rays	F	normal user	6/19/1956

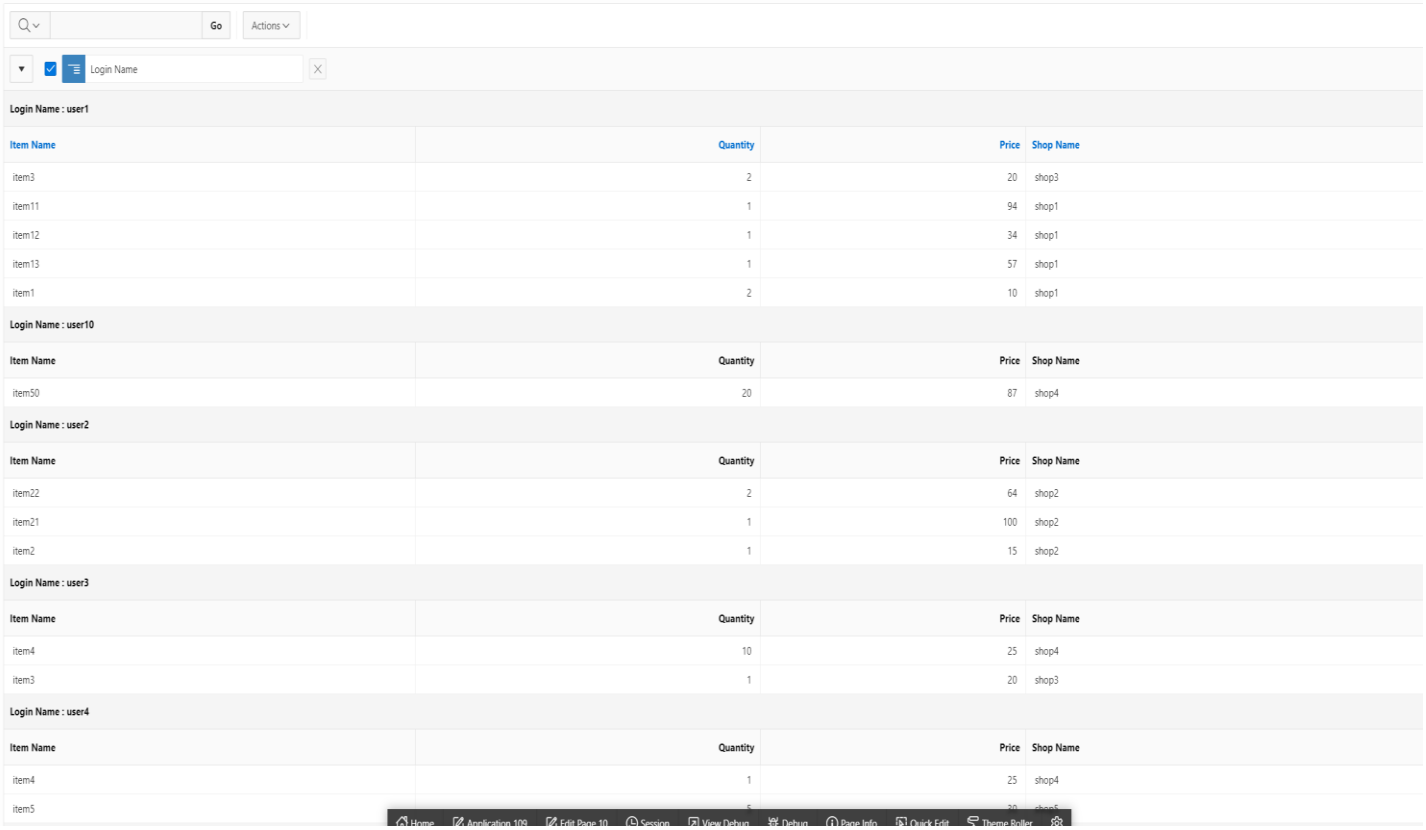
Item_Info: Interactive Report which shows the item information.

Item_Info_Interactive Report

Go Actions Reset

Item Name	States	Description	Date Published	Stock	Price	Renewal Option	Weight	Country	Process Time	Name
item1	active	description1	1/1/2020	7	10	weekly	1	USA	7	shop1
item100	end	description100	1/30/2020	10	10	monthly	2	USA	7	shop1
item11	active	description11	3/15/2020	94	94	weekly	1	USA	23	shop1
item12	active	description12	4/17/2020	86	34	weekly	4	USA	28	shop1
item13	active	description13	6/1/2020	73	57	weekly	1	USA	2	shop1
item14	active	description14	3/1/2020	12	29	weekly	7	USA	22	shop1
item15	active	description15	3/31/2020	99	42	weekly	4	USA	11	shop1
item16	active	description16	4/29/2020	49	20	weekly	6	USA	12	shop1
item17	active	description17	2/18/2020	76	71	weekly	4	USA	21	shop1
item18	active	description18	5/5/2020	29	40	weekly	6	USA	17	shop1
item19	active	description19	3/20/2020	84	91	weekly	4	USA	4	shop1
item20	active	description20	3/6/2020	45	94	weekly	5	USA	4	shop1
item2	active	description2	1/2/2020	9	15	weekly	1	USA	7	shop2
item21	active	description21	2/1/2020	86	100	weekly	5	USA	18	shop2
item22	active	description22	3/12/2020	22	64	weekly	9	USA	18	shop2
item23	active	description23	3/8/2020	42	74	weekly	9	USA	19	shop2
item24	active	description24	3/5/2020	51	33	weekly	4	USA	5	shop2
item25	active	description25	2/11/2020	33	75	weekly	5	USA	19	shop2
item26	active	description26	4/8/2020	51	56	weekly	8	USA	17	shop2
item27	active	description27	1/3/2020	1	51	weekly	2	USA	5	shop2
item28	active	description28	3/11/2020	83	92	weekly	6	USA	16	shop2
item29	active	description29	2/20/2020	62	41	weekly	1	USA	21	shop2
item30	active	description30	5/29/2020	74	71	weekly	2	USA	8	shop2

Shopping_cart: Interactive Report which shows the items in the shopping cart of each buyer.



Go Actions

Login Name

Login Name : user1

Item Name	Quantity	Price	Shop Name
item3	2	20	shop3
item11	1	94	shop1
item12	1	34	shop1
item13	1	57	shop1
item1	2	10	shop1

Login Name : user10

Item Name	Quantity	Price	Shop Name
item50	20	87	shop4

Login Name : user2

Item Name	Quantity	Price	Shop Name
item22	2	64	shop2
item21	1	100	shop2
item2	1	15	shop2

Login Name : user3

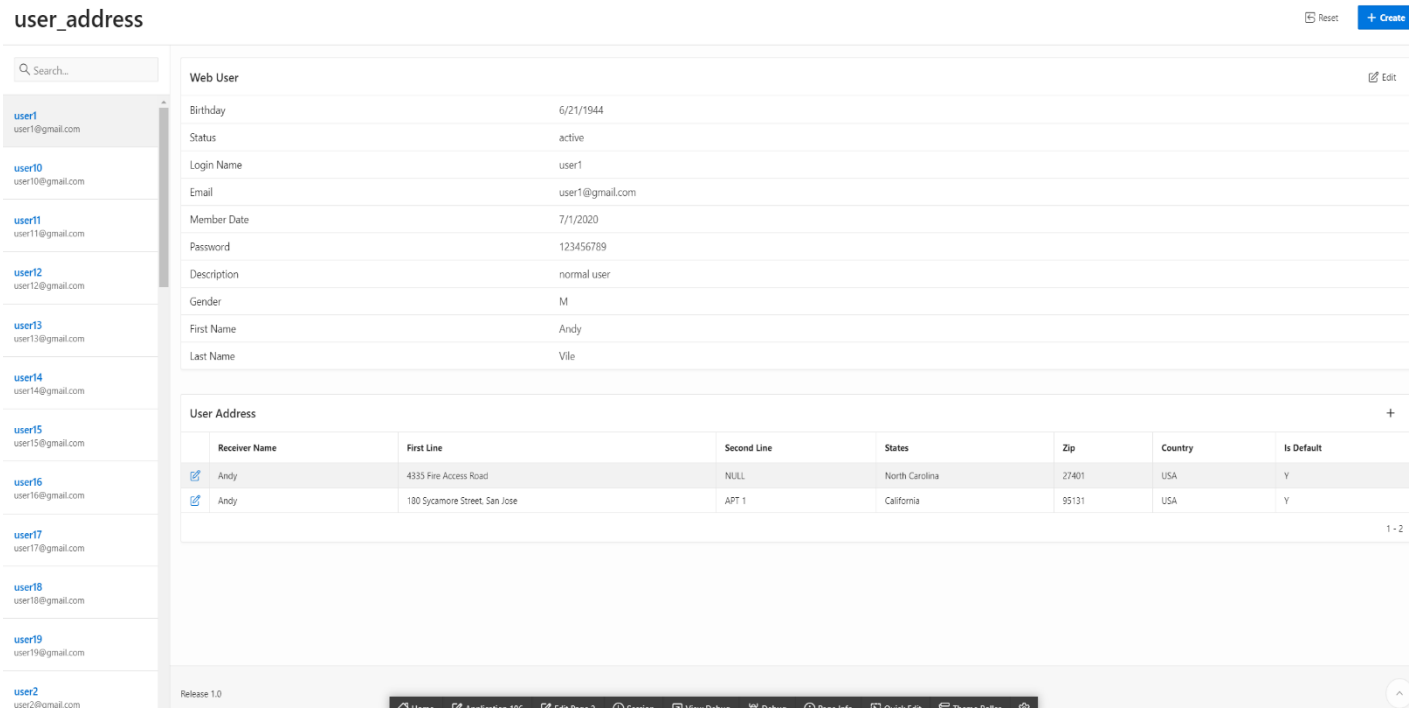
Item Name	Quantity	Price	Shop Name
item4	10	25	shop4
item3	1	20	shop3

Login Name : user4

Item Name	Quantity	Price	Shop Name
item4	1	25	shop4
item5			

Home Application 109 Edit Page 10 Session View Debug Debug Page Info Quick Edit Theme Roller

User_address: Master Detail which shows the addresses saved for each user.



user_address

Reset + Create

Search...

user1 user1@gmail.com
user10 user10@gmail.com
user11 user11@gmail.com
user12 user12@gmail.com
user13 user13@gmail.com
user14 user14@gmail.com
user15 user15@gmail.com
user16 user16@gmail.com
user17 user17@gmail.com
user18 user18@gmail.com
user19 user19@gmail.com
user2 user2@gmail.com

Web User Edit

Birthday	6/21/1944
Status	active
Login Name	user1
Email	user1@gmail.com
Member Date	7/1/2020
Password	123456789
Description	normal user
Gender	M
First Name	Andy
Last Name	Vile

User Address +

Receiver Name	First Line	Second Line	States	Zip	Country	Is Default
Andy	4335 Fire Access Road	NULL	North Carolina	27401	USA	Y
Andy	180 Sycamore Street, San Jose	APT 1	California	95131	USA	Y

1 - 2

Release 1.0

Calendar_listing: Calendar which shows the publish date of each item in the system.

calendar_listing