Technical Support Engineering Homework

Dear Shippo Support Engineering Candidate,

The first part of this two part homework assignment will test your familiarity with SQL against tables that are critical components of Shippo's label purchasing flow.

To understand what entities comprise a label purchase in Shippo's system, please review the following diagram: https://shippo-static.s3.amazonaws.com/img/illustrations/api-object-flow.png.

Note: The boxes in the diagram represent references to api_parcel, api_address, api_shipment, api_rate and api_transaction tables. The tables below represent a subset of those tables.

api_transaction

Name	Data Type	Description
id	bigserial	Internally facing unique identifier for record (not exposed to customers in API responses)
object_id	uuid	UUID (externally facing unique identifier unique to table, exposed to customers in API responses)
object_status	varchar	Status of the transaction
object_created	timestamp with time zone	Timestamp of when the transaction was created
object_updated	timestamp with time zone	Timestamp of the last transaction update
object_owner_id	bigint	FK to app_user
tracking_number	varchar	Carrier specific tracking number
eta	timestamp with timezone	Most recently updated eta from carrier
label_url	varchar	URL pointing to label
internal_metadata	varchar	
order_id	bigint	FK to user_order (Order table)
parcel_id	bigint	FK to api_parcel

manifest_id	bigint	FK to manifest
api_rate_id	bigint	FK to api_rate

api_rate

Name	Data Type	Description
id	bigserial	Internally facing unique identifier for record (not exposed to customers in API responses)
object_id	UUID	UUID (externally facing unique identifier unique to table, exposed to customers in API responses)
object_state	varchar	state of the Rate
object_purpose	varchar	purpose of the Rate
object_owner	int	FK to auth_user
object_used	boolean	whether this Rate has been used
account	int	FK to Account
amount	numeric	Stores the total cost of the rate in the currency of the sender address country
amount_local	numeric	Stores the total cost of the rate in the currency of the recipient address country
currency_rate_usd	numeric	Exchange rate from USD to the currency of amount (currency_id)
duration	int	FK to adapter.Duration

servicelevel_id	int	FK to adapter.servicelevel
servicelevel_name	varchar	Name of the service level eg: “Priority Mail”
servicelevel_token	varchar	Token representing service level eg “Priority”, “PriorityExpress”, “dhl_germany_paket”
shipment_id	bigint	FK to api_shipment. All proposed rates have a same shipment_id.

api_shipment

Name	Data Type	Description
id	bigserial	Internally facing unique identifier for record (not exposed to customers in API responses)
object_id	uuid	UUID (externally facing unique identifier unique to table, exposed to customers in API responses)
object_state	VARCHAR	State of the Shipment
object_purpose	VARCHAR	Purpose of the Shipment (PURCHASE, QUOTE)
object_created	timestamp with time zone	Timestamp of shipment creation
object_updated	timestamp with time zone	Timestamp of last shipment update
object_owner_id	bigint	FK to auth_user
address_from_id	bigint	FK to Api_Address
address_to_id	bigint	FK to Api_Address
address_return_id	bigint	FK to Api_Address
parcel_id	bigint	FK to Api_Parcel

is_return	boolean	indicates whether or not the shipment is a return
submission_date	timestamp with time zone	date shipment will be submitted to carrier
insurance_amount	decimal	amount of insurance paid
extra	VARCHAR	Additional JSON parameters sent with request
is_test	boolean	indicates whether the object is a test shipment
reference_1	VARCHAR	additional information to be printed on the label
reference_2	VARCHAR	additional information to be printed on the label
order_id	bigint	FK to user_order (Order table)

SQL Questions

Assume the database system being used is Postgres 9.6.

1. Write a query to find the shipment record associated to transaction object_id **‘618c313ed743ba2c7cdc3636’**.

2. Write a query to find the amount or amount_local the customer paid for a label with shipment object_id **‘618b1586fbabb3e56406d8d1’**.

3. Write a query to find the total amount spent on labels by object_owner_id 123 created between 01/01/2021 and today

4. Write a query to find all transactions for object_owner_id 123 that had shipments with the following extra key value pair: { "signature_confirmation": "ADULT", … } (a customer might have other misc keys in the extra object)

API Question

For the second part of this homework assignment, we would like you to prepare and submit the following project.

Build a tool to automate shipping related tasks. Your tool should allow your users to:

1. Create a shipment
2. Retrieve a shipment
3. Retrieve rates for a shipment
4. Retrieve either the sender or recipient address by specifying their object ids or retrieve both simultaneously by passing a shipment's object id
5. Retrieve a parcel by specifying its object id or by passing a shipment's object id

Please include a **readme** in your submission and list any dependencies required to run your project as well as installation instructions. Assume your audience will be running macOS Catalina version 10.15.7.

Your tool should minimally be able to handle the tasks described above. During the interview be prepared to walk through your code and answer questions like:

1. How would you convert your script to an API?
2. How would you build a simple UI to interface with your API?
3. What changes would you make to your API as the number of users scale and you have to start worrying about Shippo's rate limiting?

Additional Notes

1. Shippo's API reference can be found at: https://goshippo.com/docs/reference
2. To test, you will have to create a Shippo account and create an API token. Please also remember to enable one of our many master accounts in Test mode. USPS is the easiest carrier to test with.
3. We will be testing your solution with our own API key so please do not include hard coded API tokens in your submission.
4. A UI is not necessary for the submission.
5. **BONUS** points if you integrate with our SDKs and report any issues that you had with the integration experience:

Python: https://github.com/goshippo/shippo-python-client
Javascript: https://github.com/goshippo/shippo-node-client
Ruby: https://github.com/goshippo/shippo-ruby-client
PHP: https://github.com/goshippo/shippo-php-client