**Amazon QLDB**

[https://us-east-2.console.aws.amazon.com/qldb/home?region=us-east-2#gettingStarted](https://us-east-2.console.aws.amazon.com/qldb/home?region=us-east-2" \l "gettingStarted)

**Steps Using Console**

**1.Create Ledger**

* In the navigation pane, choose **Getting started**.
* On the **Create your first ledger** card, choose **Create Ledger**.
* **Add Ledger information**—The **Ledger name**
* Tags **(optional)-** Meta Data to ledger Key : value pair -to help organize and identify them

Used For: AWS uses the tags to categorize your costs and deliver a monthly cost allocation report to you

<https://docs.aws.amazon.com/qldb/latest/developerguide/tagging.html>

* Choose create Ledger – Then the status changes to Active.

# **2. Create Tables, Indexes, and Sample Data:**

<https://docs.aws.amazon.com/qldb/latest/developerguide/getting-started-step-2.html>

1.In the navigation pane, choose **Query editor**.

2. Choose your ledger

3. In the query editor window, enter the following statement, and then choose **Run**.

CREATE TABLE VehicleRegistration

CREATE TABLE Vehicle

4. create indexes that help speed up queries against each table.

CREATE INDEX ON VehicleRegistration (VIN)

**5.Insert Data to your Table:**

INSERT INTO Person << { 'FirstName' : 'Raul', 'LastName' : 'Lewis', 'DOB' : `1963-08-19T`, 'GovId' : 'LEWISR261LL', 'GovIdType' : 'Driver License', 'Address' : '1719 University Street, Seattle, WA, 98109' }>>

**3: Query the Tables in a Ledger PartiQL**

<https://docs.aws.amazon.com/qldb/latest/developerguide/getting-started-step-3.html>

SELECT v.VIN, r.LicensePlateNumber, r.State, r.City, r.Owners

FROM Vehicle AS v, VehicleRegistration AS r

WHERE v.VIN = '1N4AL11D75C109151'

AND v.VIN = r.VIN

# **4: Modify Documents in a Ledger:**

1.From Person Table Get document id of the user

SELECT metadata.id FROM \_ql\_committed\_Person AS p

WHERE p.data.FirstName = 'Raul' and p.data.LastName = 'Lewis’

>> "0VwGVLIiO484tPg1sQfPKT"

Select \* from \_ql\_committed\_Person

From Committed view you can see the entire block data

**prefix \_ql\_committed\_ For accessing from committed view**



Use this id in an UPDATE statement to modify the appropriate document in the VehicleRegistration table

UPDATE VehicleRegistration AS r

SET r.Owners.PrimaryOwner.PersonId = '294jJ3YUoH1IEEm8GSabOs' --replace with your id

WHERE r.VIN = '1N4AL11D75C109151'

# **5: View the Revision History for a Document: (Modification History )**

1.To query the history of a document, start by finding its unique id. In addition to querying the committed view, another way of getting a document id is to use the BY keyword in the table's default user view. To learn more, see [Using the BY Clause to Query Document ID](https://docs.aws.amazon.com/qldb/latest/developerguide/working.metadata.by-clause.html).

In the query editor window, enter the following statement, and then choose **Run**.

SELECT r\_id FROM VehicleRegistration AS r BY r\_idWHERE r.VIN = '1N4AL11D75C109151'

History Does not Provide information in an sequential Order

2.SELECT h.data.VIN, h.data.City, h.data.Owners

FROM history(VehicleRegistration) AS h

WHERE h.metadata.id = 'ADR2LQq48kB9neZDupQrMm' --replace with your id

>>

|  |  |  |
| --- | --- | --- |
| **VIN** | **City** | **Owners** |
| "1N4AL11D75C109151" | "Seattle" | {PrimaryOwner:{PersonId:""},SecondaryOwners:[]} |
| "1N4AL11D75C109151" | "Seattle" | {PrimaryOwner:{PersonId:"294jJ3YUoH1IEEm8GSabOs"}, SecondaryOwners:[]} |
| "1N4AL11D75C109151" | "Everett" | {PrimaryOwner:{PersonId:"7NmE8YLPbXc0IqesJy1rpR"}, SecondaryOwners:[]} |
| "1N4AL11D75C109151" | "Everett" | {PrimaryOwner:{PersonId:"7NmE8YLPbXc0IqesJy1rpR"}, SecondaryOwners:[{PersonId:"5Ufgdlnj06gF5CWcOIu64s"}]} |

3. To inspect the document metadata of each revision. Enter the following statement, and then choose **Run**. Again, be sure to replace the id value with your own document ID as appropriate.

SELECT VALUE h.metadataFROM history(VehicleRegistration) AS hWHERE h.metadata.id = '*ADR2LQq48kB9neZDupQrMm*'

|  |  |  |  |
| --- | --- | --- | --- |
| **version** | **id** | **txTime** | **txId** |
| 0 | "ADR2LQq48kB9neZDupQrMm" | 2019-05-23T19:20:360d-3Z | "FMoVdWuPxJg3k466Iz4i75" |
| 1 | "ADR2LQq48kB9neZDupQrMm" | 2019-05-23T21:40:199d-3Z | "KWByxe842Xw8DNHcvARPOt" |
| 2 | "ADR2LQq48kB9neZDupQrMm" | 2019-05-23T21:44:432d-3Z | "EKwDOJRwbHpFvmAyJ2Kdh9" |
| 3 | "ADR2LQq48kB9neZDupQrMm" | 2019-05-23T21:49:254d-3Z | "96EiZd7vCmJ6RAvOvTZ4YA" |

# **6: Verify a Document in a Ledger**

**To Request Digest:**

* 1. Click on Verification under task panel
  2. Choose Get digest under prerequisites
     1. An ion format file .ion.txt will be downloaded

{ "digest": "42zaJOfV8iGutVGNaIuzQWhD5Xb/5B9lScHnvxPXm9E=", "digestTipAddress": "{strandId:\"BlFTjlSXze9BIh1KOszcE3\",sequenceNo:73}", "ledger": "vehicle-registration", "date": "2019-04-17T16:57:26.749Z" }

Save the file

**To Verify document Revision :**

* 1. Choose the Ledger
  2. Get id and blockaddress of the table you want to verify :

SELECT r.metadata.id, r.blockAddress FROM \_ql\_committed\_VehicleRegistration AS r WHERE r.data.VIN = '1N4AL11D75C109151'

* 1. Paste the details in the verify a document dialog
  2. Click Verify

# **7 (Optional): Clean Up Resources:**

**To delete the ledger**

* Open the Amazon QLDB console at <https://console.aws.amazon.com/qldb>.
* In the navigation pane, choose **Ledgers**.
* In the list of ledgers, select vehicle-registration.
* Choose **Delete ledger**. Confirm this by entering **Ledgername** in the field provided.
* If deletion protection is enabled for this ledger, you must also choose the option to **Override deletion protection**.

**-------------------------------------------------------------------------------------------------------------------------------------**

# **Amazon QLDB Driver for Python:**

<https://docs.aws.amazon.com/qldb/latest/developerguide/getting-started.python.html>

**Configuration:**

## **1.Creating an Administrator IAM User and Group (Console)**

This procedure describes how to use the AWS Management Console to create an IAM user for yourself and add that user to a group that has administrative permissions from an attached managed policy.

**To create an administrator user for yourself and add the user to an administrators group (console)**

1. Use your AWS account email address and password to sign in as the *[AWS account root user](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html) to the IAM console at <https://console.aws.amazon.com/iam/>.*
2. **Note**
3. We strongly recommend that you adhere to the best practice of using the **Administrator** IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few [account and service management tasks](https://docs.aws.amazon.com/general/latest/gr/aws_tasks-that-require-root.html).
4. Enable access to billing data for the IAM admin user that you will create.
   1. On the navigation bar, choose your account name, and then choose **My Account**.
   2. Next to **IAM User and Role Access to Billing Information**, choose **Edit**.
   3. Select the check box to **Activate IAM Access** and choose **Update**.
   4. On the navigation bar, choose **Services** and then **IAM** to return to the IAM dashboard.
5. In the navigation pane, choose **Users** and then choose **Add user**.
6. For **User name**, type **Administrator**.
7. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type your new password in the text box. By default, AWS forces the new user to create a new password when first signing in. You can optionally clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
8. Choose access key checkbox
9. Choose **Next: Permissions**.
10. On the **Set permissions** page, choose **Add user to group**.
11. Choose **Create group**.
12. In the **Create group** dialog box, for **Group name** type **Administrators**.
13. Choose **Filter policies**, and then choose **AWS managed - job function** to filter the table contents.
14. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
15. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
16. Choose **Next: Tags**.
17. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see [Tagging IAM Users and Roles](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_tags.html).
18. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

# **2.Installing the AWS CLI version 2 on Linux:**

[https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html#cliv2-linux-install](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html" \l "cliv2-linux-install)

curl "<https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip>" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

**3.Configure aws**

$ **aws configure**

AWS Access Key ID [None]: *AKIAIOSFODNN7EXAMPLE*

AWS Secret Access Key [None]: *wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY* Default

region name [None]: *us-west-2* (your code should match with the configuration location)

Default output format [None]: *json*

**Steps:**

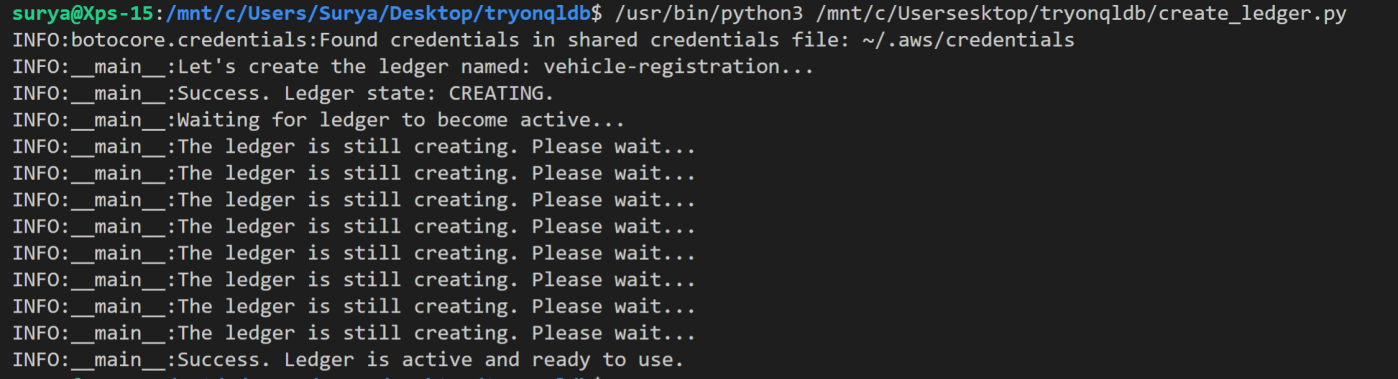
1. **To create a Ledger**

1.Constants.py contains constant values that are used by all of the other programs in this tutorial.

<https://docs.aws.amazon.com/qldb/latest/developerguide/getting-started.python.step-1.html>

2. create\_ledger.py

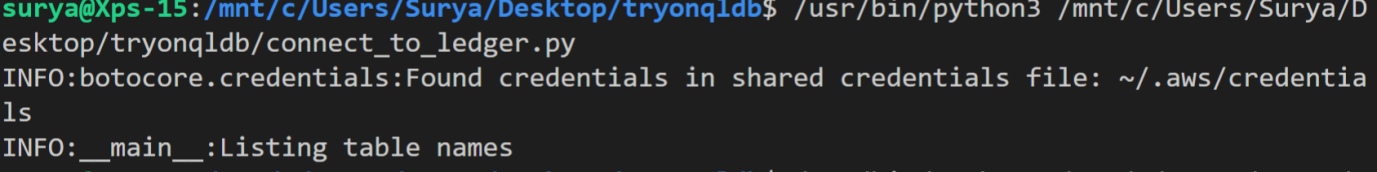
1. run python create\_ledger.py



# **2.Test Connectivity to the Ledger**

connect\_to\_ledger.py to create a data session connection to the vehicle-registration ledger.

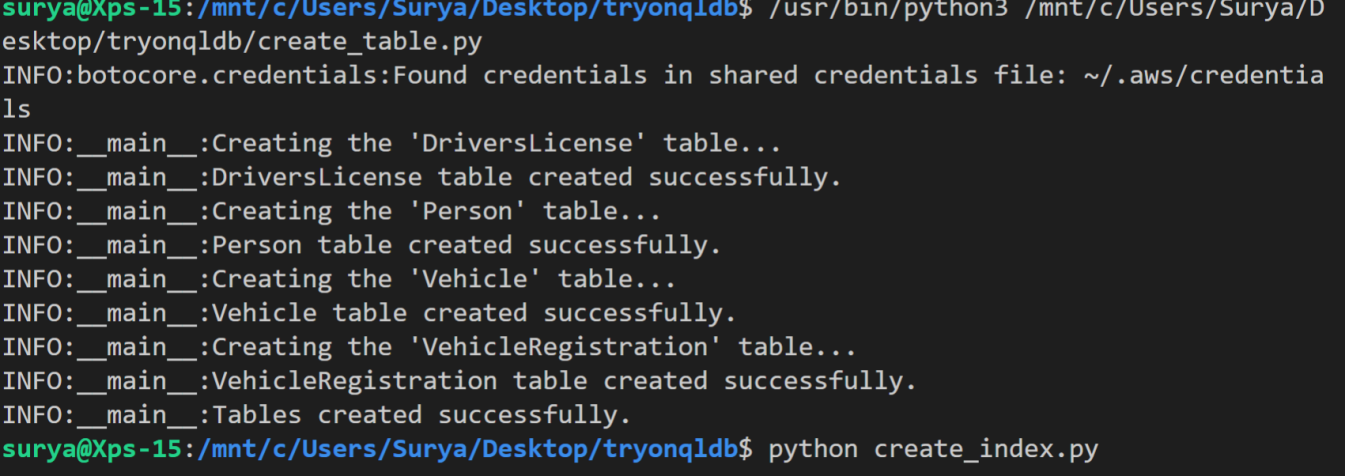
python connect\_to\_ledger.py



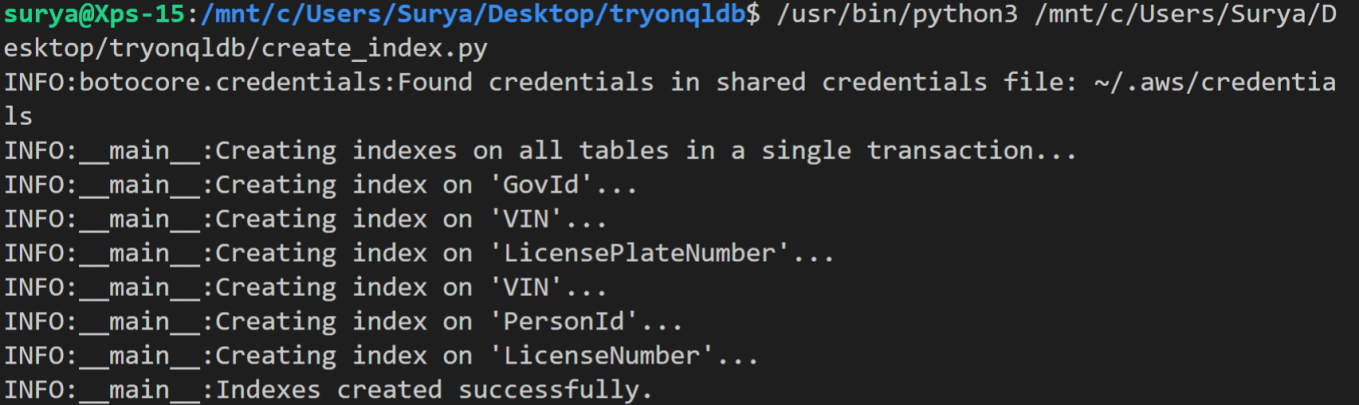
# **3. Create Tables, Indexes, and Sample Data**

<https://docs.aws.amazon.com/qldb/latest/developerguide/getting-started.python.step-3.html>

create\_table.py



create\_index.py

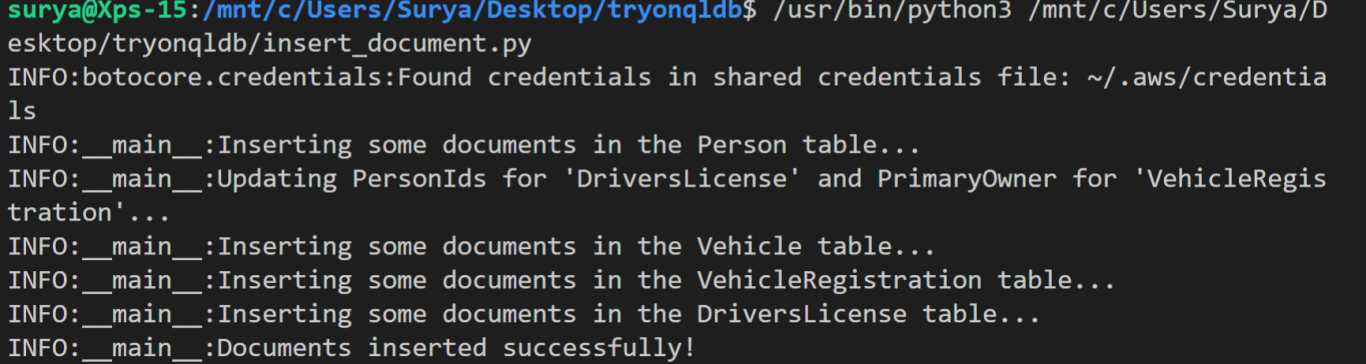


To insert sample Data:

1.Sampledata.py uses Amazon Ion Format

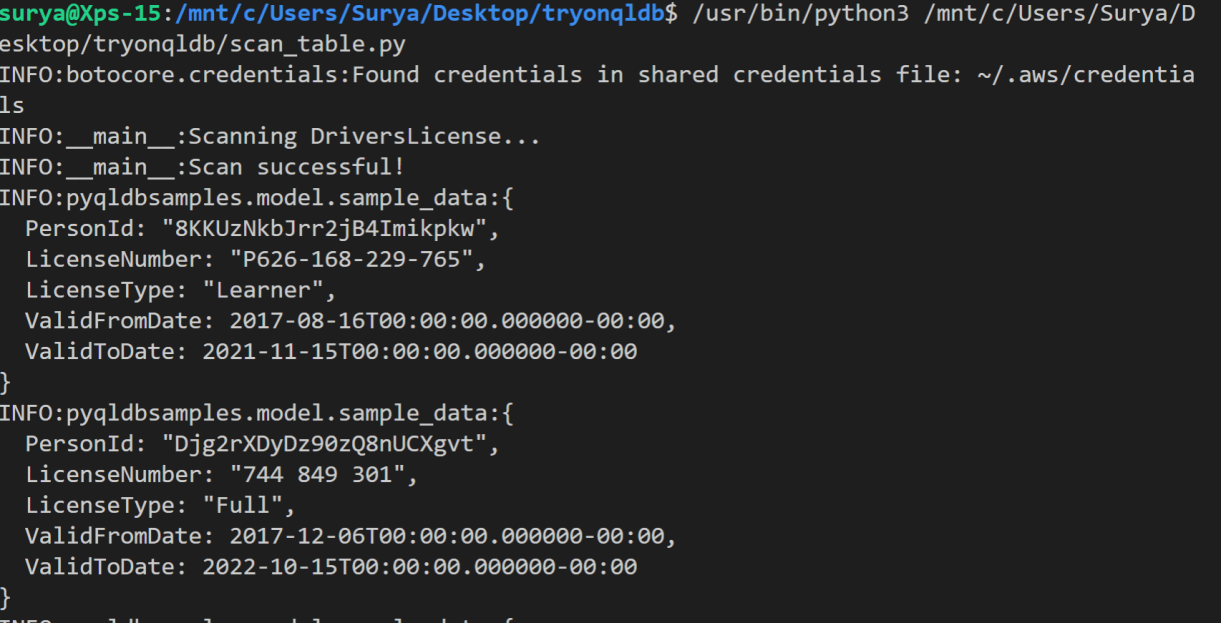
2.InsertDocument.py

3.Run python insertDocument.py

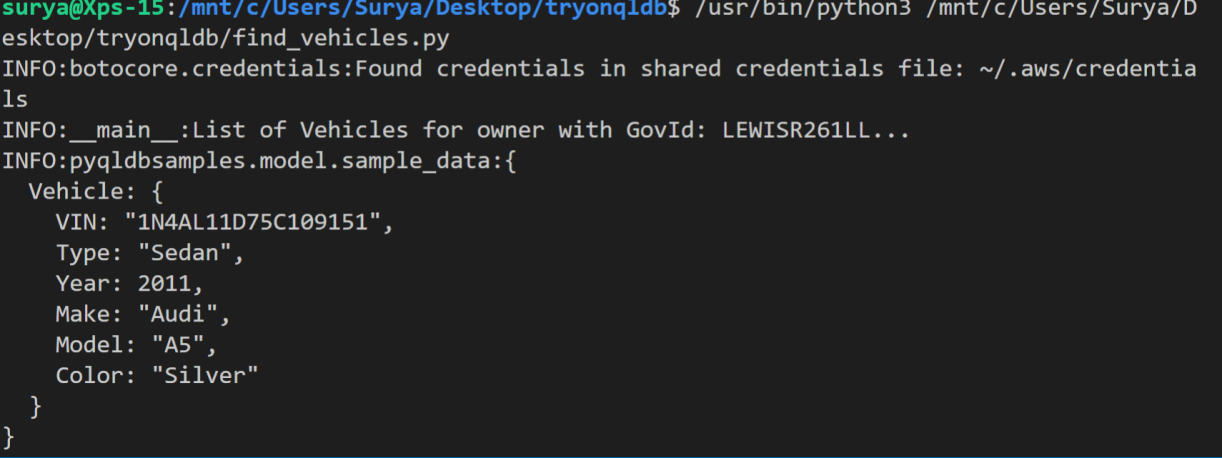


# **4: Query the Tables in a Ledger**

1.scan\_table.py to scan all documents in all tables in your ledger

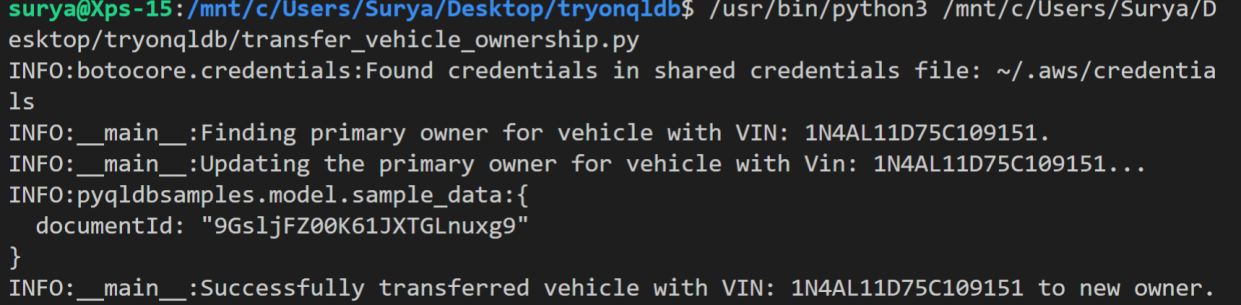


2.Find\_vehicles.py to query all vehicles registered under a person in your ledger.

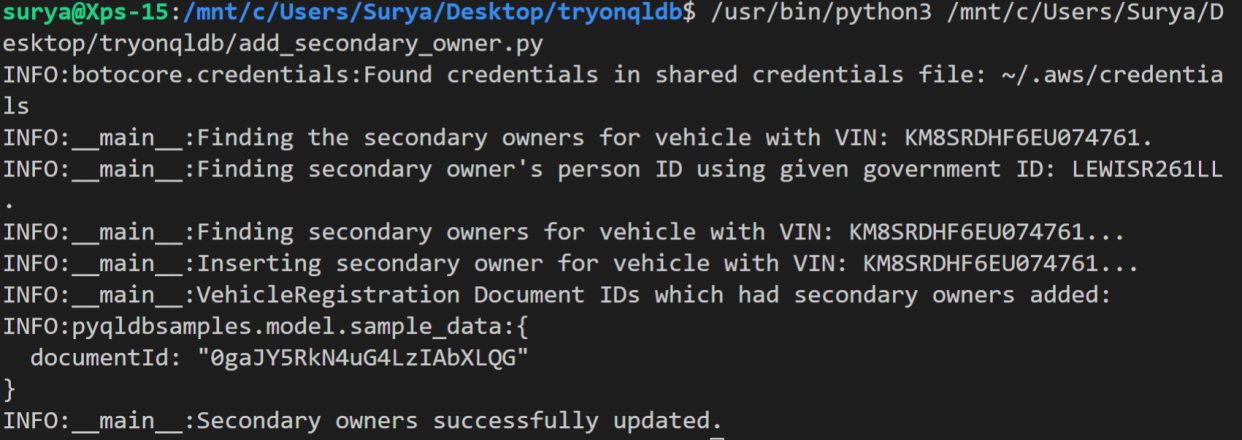


# **5: Modify Documents in a Ledger**

1. transfer\_vehicle\_ownership.py to update the primary owner of the vehicle with VIN 1N4AL11D75C109151 in your ledger.

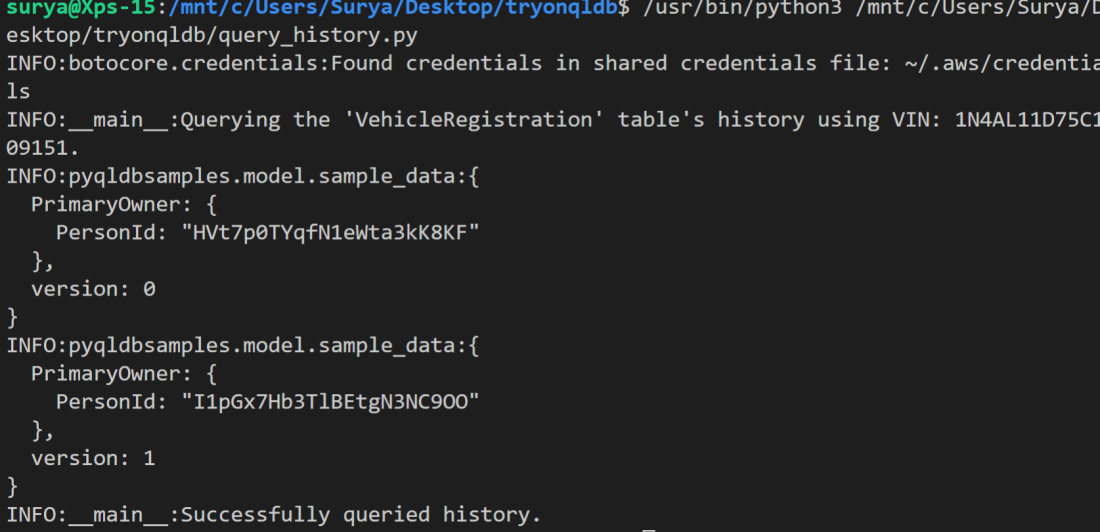


2. add\_secondary\_owner.py to add a secondary owner to the vehicle with VIN KM8SRDHF6EU074761 in your ledger.



# **6: View the Revision History for a Document**

1. query\_history.py) to query the revision history of the VehicleRegistration document with VIN 1N4AL11D75C109151.



# **7: Verify a Document in a Ledger**

**To verify a document revision:**

1.QLDB objects that are required for verification and a utility module with helper functions to convert QLDB response types to strings.

block\_address.py

Verifier.py

qldb\_string\_utils.py

2.Use two .py programs (get\_digest.py and get\_revision.py) to perform the following steps:

* 1. Request a new digest from the vehicle-registration ledger.
  2. Request a proof for each revision of the document with VIN 1N4AL11D75C109151 from the VehicleRegistration table.
  3. Verify the revisions using the returned digest and proof by recalculating the digest.

get\_digest.py

get\_revision.py

