Amazon QLDB ワークショップ

2021/07/22

シニアエバンジェリスト 亀田治伸

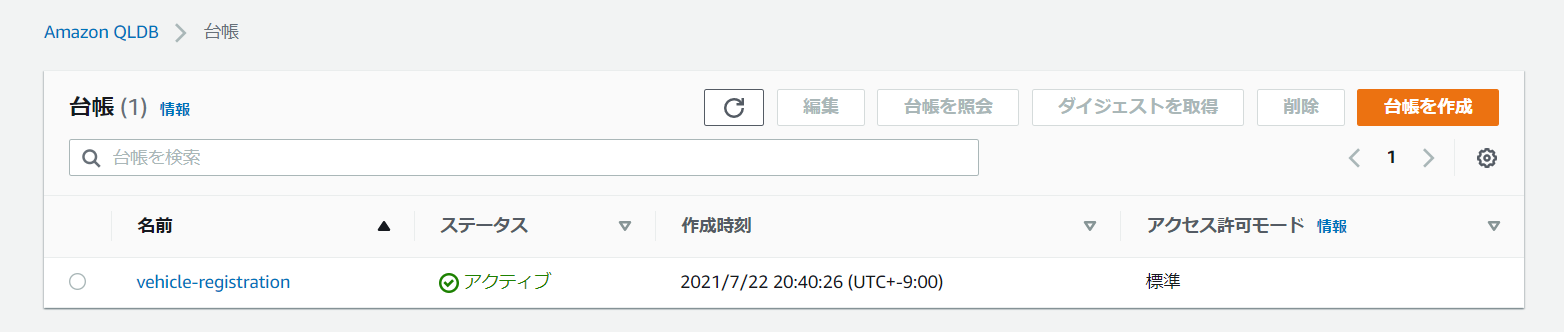
1. QLDBのマネージメントコンソールに移動します



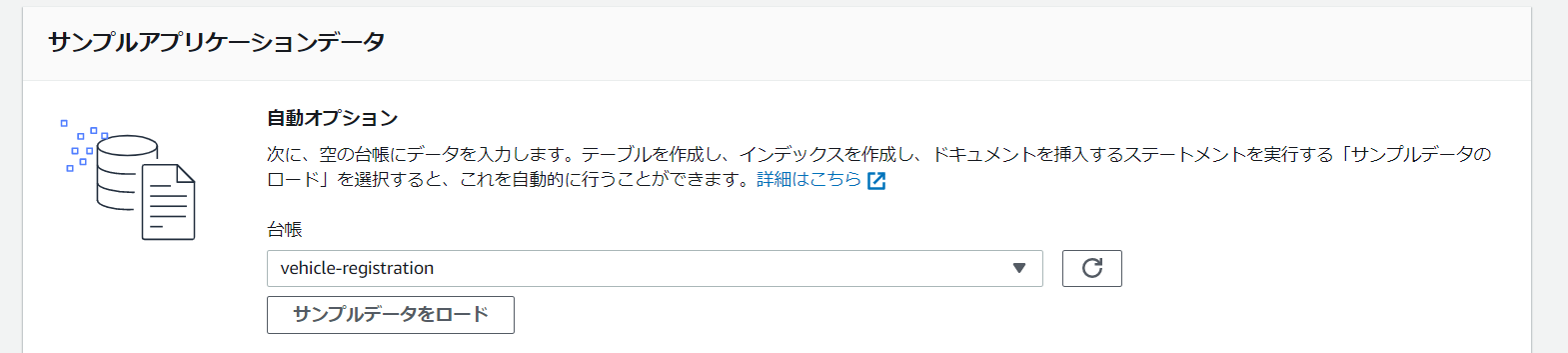
1. 左ペインの[開始方法]をクリックし[台帳を作成]をおします



1. すべてデフォルトのまま[台帳を作成]をおします
2. 数分待つと以下のように台帳が作成されます



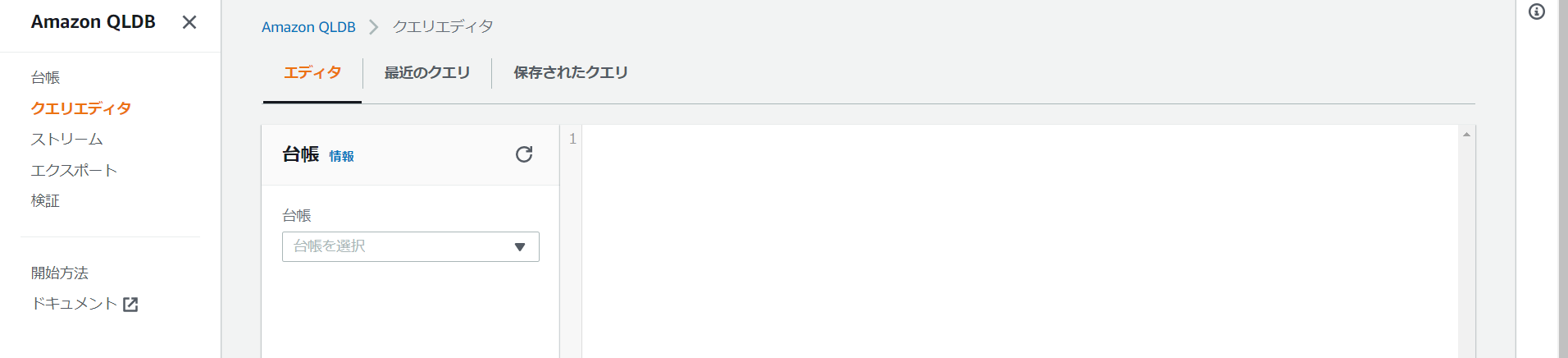
1. 再度左ペインの[開始方法]をクリックし、[サンプルアプリケーションデータ]から、作成された台帳を選び、[サンプルデータのロード]をおします



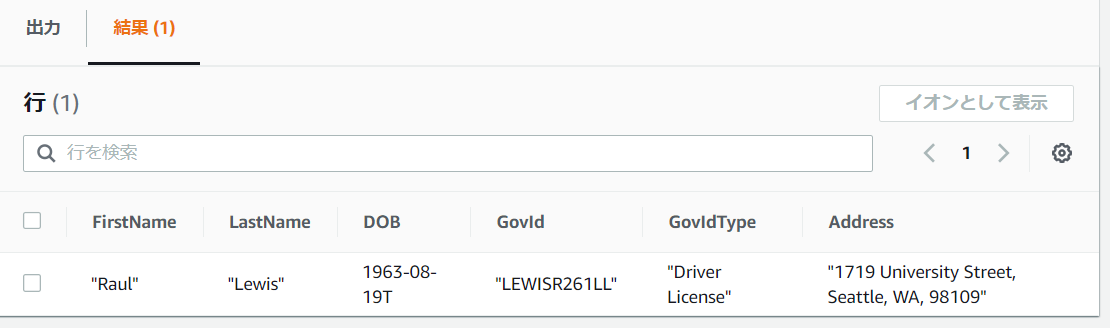
1. 以下が表示されれば正しくデータがロードされています。これにより台帳に4つのテーブルができています。VehicleRegistration、Vehicle,、Person、DriversLicense



1. 左ペインから[クエリエディタ]をクリックします



1. Commands.txtの1番から4番を順に実行し、それぞれのテーブルの中に何が含まれているかを確認しておきます
2. Command.txtの5番を実行し、ある特定の車の情報を出力します
3. Command.txtの6番を実行し、だれに所有されているかを確認します
4. 上記で表示された[PersonId]を用いて7番のクエリを実行すると、そのVehicleを保有しているLicenseNumberがわかります。（###の部分を置換します）
5. 表示された [LicenseNumber]の値を8番の###に置換して実行すると所有者情報が出力されます。



1. また、9番の###をそのままLicenseNumberに置換して実行しても、上記11及び12を一度に行った結果が出力されます。9番のコマンドは2つのテーブルを連結したのち検索をかけているためです。この辺りの処理は通常のRDBと同様に行うことができます。

[所有者の移転]

今までの作業で、乗り物’ 1N4AL11D75C109151’は’ Raul Lewis’が所有していることがわかりました。これからこの所有者を’ Brent Logan’に移転します

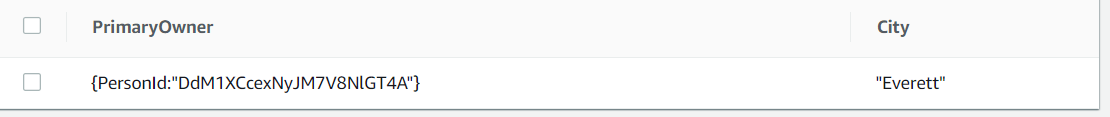
1. 再度、“SELECT \* FROM Person”を実行します。このテーブルにはIDカラムが存在していないことがわかります。QLDBではテーブル名の頭に[\_ql\_committed\_]を付与することでテーブルのメタデータを取得することができます。Commandの10番、11番を順に実行してIDを取得してください。

10番で、全てのメタデータ、11番でメタデータ内に含まれるIDが出力されました。

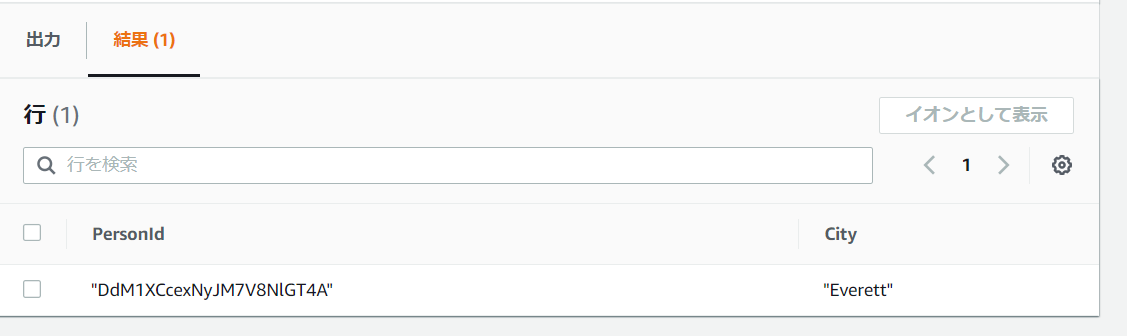
データ構造はこのようなNested JSONになっています。通常のクエリはdata部分に対してのみ行われています。



1. 12番の<<replace with your id>>を先程取得したIDに置き換え実行します。（ダブルクォーテーションは含まないので注意してください）
2. これで乗り物の所有者が移転しています。13番を実行して確認してください。



1. この状態では、Nestされた値が出力されているため、13番を少し改修した14番を実行してください。今度は[PersonId]のみが出力されました。13番と14番の違いを見ておいてください



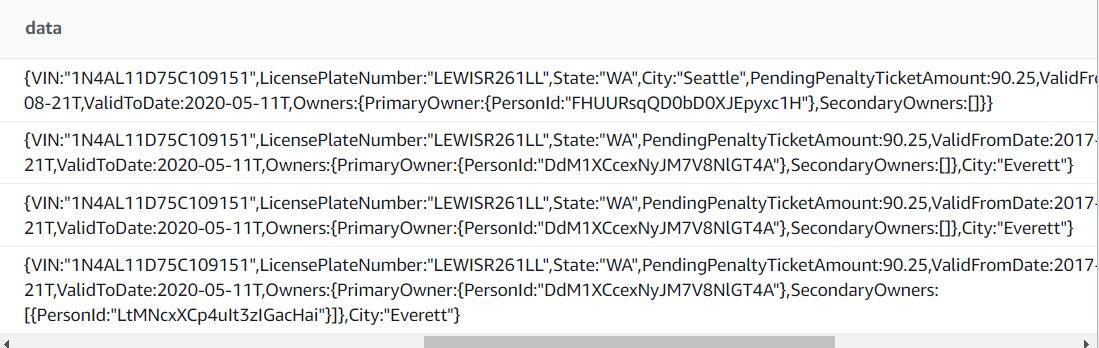
1. 次に’Alex Pena’をこの乗り物の共同所有者として追加登録します。先の手順では、所有者を書き換えましたが、今度の手順では追加（Secondary Owner）します。15番を実行してIDを取得します
2. 次に16番を実行して、登録を行います。[<<replace with your id>>]を上で表示されたIDに置きかえます。（ダブルクォーテーションは含みません）
3. 17番を実行して、SecondaryOwnerが追加されたか確認してください。

[Historyの確認]

今までの手順で、乗り物の所有権が一度移転し、さらにSecondaryOwnerが追加されています。

1. まず18番を実行し乗り物のID（メタデータ上に存在）を取得します
2. 19番を上で出力されたIDで置換し実行します。[FROM history(VehicleRegistration)]に対してSQLを発行していますが、これはQLDB専用の機能でありトランザクションの履歴を保持しているテーブルです。以下のように履歴が出力されます



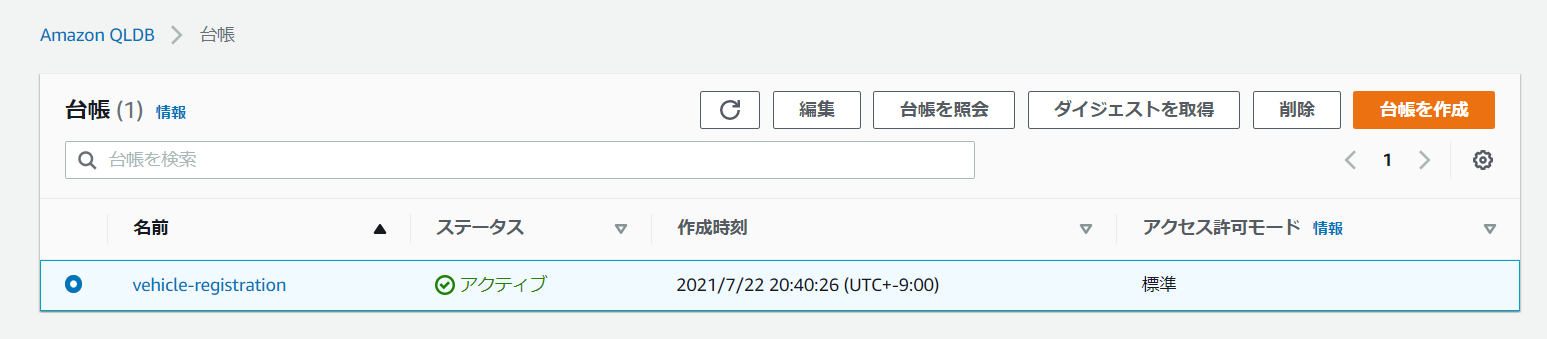


（データ個数は作業した分だけ増えていくので人により異なりますので気にしなくて大丈夫です）

[トランザクションの検証]

QLDBはSHA-256ハッシュツリー構造をベースとしてトランザクションのイミュータブル性（不変性）が提供されるため、トランザクションの改ざんなどを防ぐことができます。ここから今でおこなった作業が正しく検証できるかを確認します。

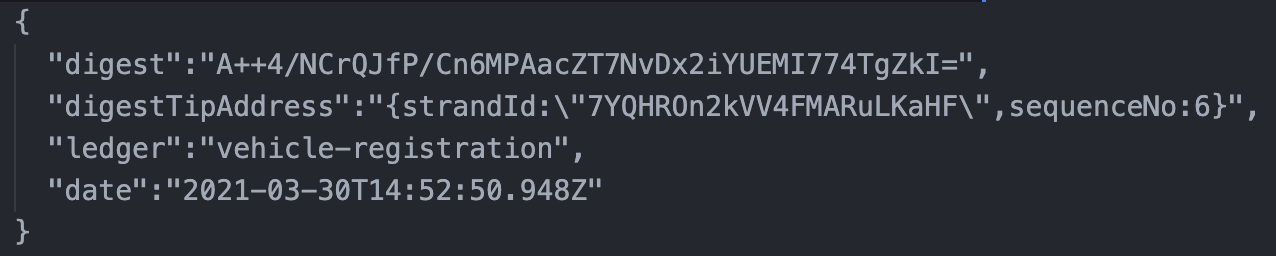
1. 左ペインから[検証]をクリックし、[ダイジェストを取得]をおします
2. 台帳一覧が表示されますので、台帳を選び[ダイジェストを取得]をおします



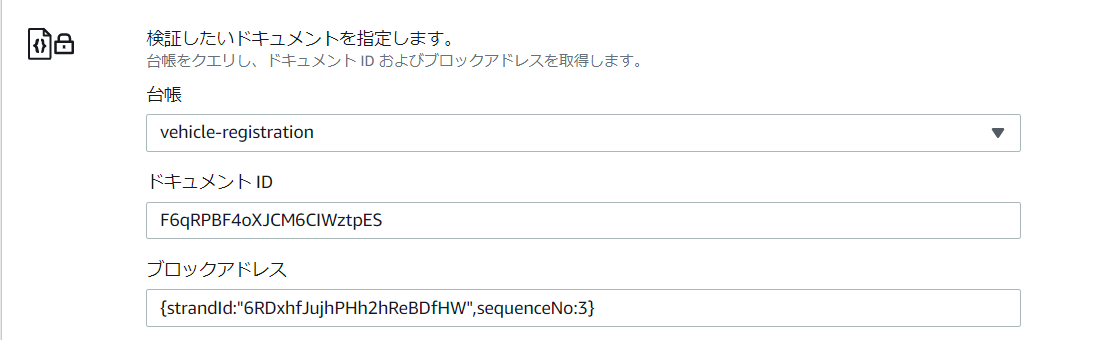
1. 確認画面で[保存]をおします



1. QLDBではすべてのトランザクションはハッシュツリー構造によるつながっています。（シンプルにはブロックチェーンと同じ構造です）このためデータも整合性検証は一番最後のトランザクションのポイントがあれば検証可能です。途中のトランザクションを改ざんしたとした場合、一番最後のトランザクションの計算結果が異なり別の値になってしまうためです。このためダウンロードされたファイルには非常にシンプルな一番最後のトランザクションの情報へのフィンガープリントのみが含まれています



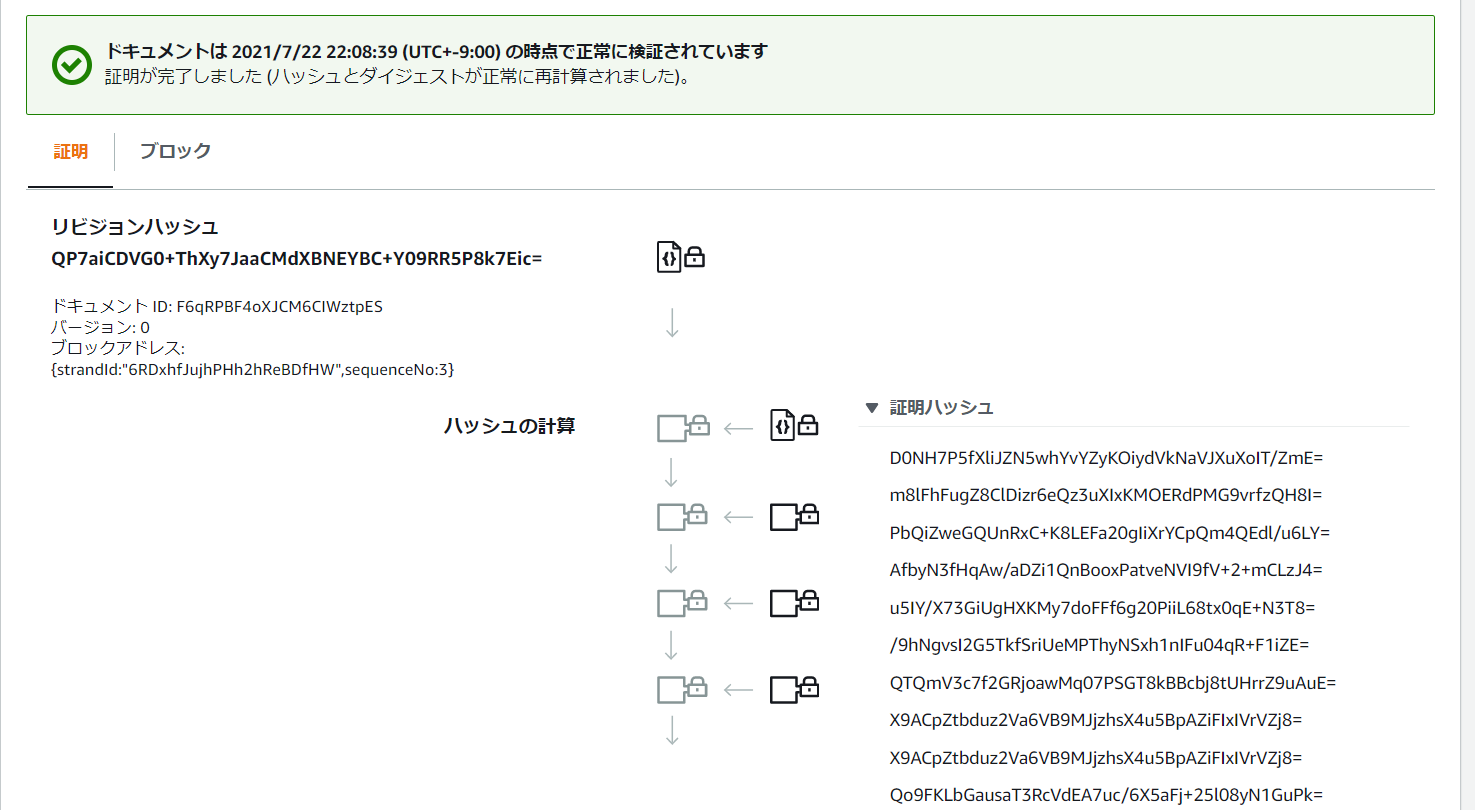
1. 再度クエリエディタに戻り、20番を実行します。このコマンドは今まで全く登場してこなかった乗り物に関する情報（IDをブロックアドレス）を取得します。ブロックアドレスとは、ハッシュツリー構造に組み込まれるそのデータが持つ、QLDBが計算に用いる値が格納されている場所をさします。
2. この2つの値をメモ帳にコピーしておきます。(ダブルクォーテーションは不要です)
3. 再度左ペインから[検証]をクリックし、[ドキュメントID]と[ブロックアドレス]にさっきコピーしたものをペーストします



1. [検証に使用するダイジェストを指定します。] で先程ダウンロードしたファイルをアップロードします。値は自動で読み込まれ入力が行われます。



1. [検証]をおします。以下のように正当性が検証されました。





[テーブルからデータの削除]

では最後にデータをテーブルから削除してみます

1. 21番を実行して、IDを取得しておきます。（HistoryレコードにこのIDで後ほど検索を行います）
2. 22番を実行し、乗り物情報を消します。
3. 23番を実行すると、データが消えていることがわかります。
4. 24番を実行すると、[history(VehicleRegistration)]には、その履歴が引き続き残っていることがわかります。何も出力されない場合、上記手順32で取得したIDに置き換えて実行してみてください。

おつかれさまでした！

削除は以下を行ってください。

・QLDB 台帳