

1. QLDB のマネージメントコンソールに移動します

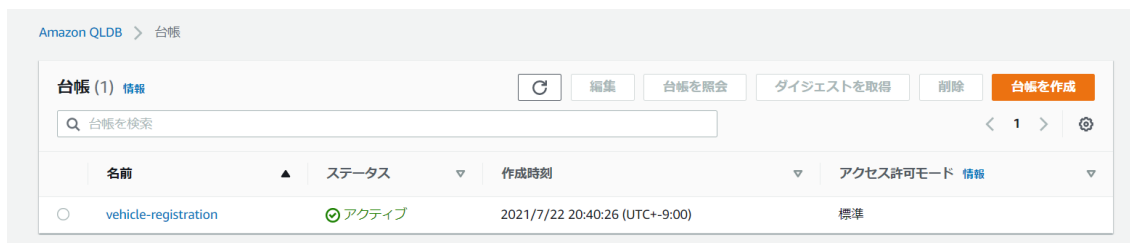


2. 左ペインの[開始方法]をクリックし[台帳を作成]をおします

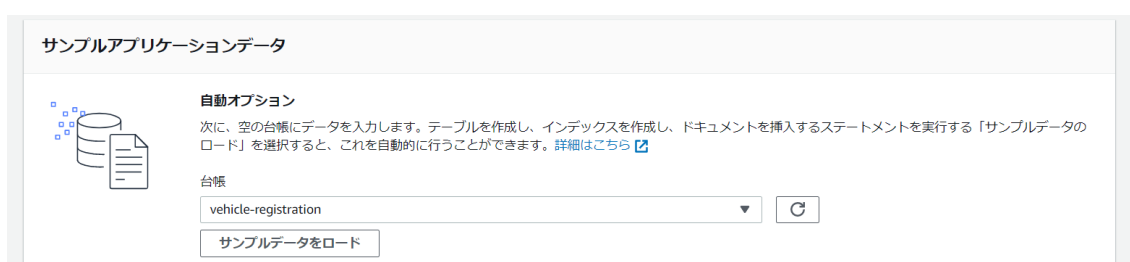


3. すべてデフォルトのまま[台帳を作成]をおします

4. 数分待つと以下のように台帳が作成されます



5. 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



6. 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

🟢 サンプルデータがロードされました

7. 左ペインから[クエリエディタ]をクリックします



8. 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
9. Command.txt の 5 番を実行し、ある特定の車の情報を出力します
10. 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
11. 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。(###の部分置換します)
12. 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。

出力

結果 (1)

行 (1)

イオンとして表示

🔍 行を検索

< 1 >

⚙️

<input type="checkbox"/>	FirstName	LastName	DOB	GovId	GovIdType	Address
<input type="checkbox"/>	"Raul"	"Lewis"	1963-08-19T	"LEWISR261LL"	"Driver License"	"1719 University Street, Seattle, WA, 98109"

13. 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

14. 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。

```
{
  blockAddress: {
    strandId: "2E0krLUtQa226ddeoU7ja3",
    sequenceNo: 36
  },
  hash: {{LCynMYTJiPwmLoTOL0x13KB+Ikju5jbfb39Ynsw240g=}},
  data: {
    FirstName: "Brent",
    LastName: "Logan",
    DOB: 1967-07-03T,
    GovId: "LOGANB486CG",
    GovIdType: "Driver License",
    Address: "43 Stockert Hollow Road, Everett, WA, 98203"
  },
  metadata: {
    id: "JuvoLmXrENBEFUDWacVuOb",
    version: 0,
    txTime: 2019-09-05T18:16:50.542Z,
    txId: "DZ1zjlG5gGm6m3Jf9JJBhK"
  }
}
```

15. 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)
16. 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。

<input type="checkbox"/>	PrimaryOwner	City
<input type="checkbox"/>	{PersonId:"DdM1XCcxNyJM7V8NlGT4A"}	"Everett"

17. 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

出力

結果 (1)

行 (1)

イオンとして表示

🔍 行を検索

< 1 >

⚙

<input type="checkbox"/>	PersonId	City
<input type="checkbox"/>	"DdM1XCcexNyJM7V8NlGT4A"	"Everett"

18. 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。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

19. 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)
20. 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。

21. 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
22. 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

行 (4)		イオンとして表示
Q 行を検索		< 1 > ⚙
<input type="checkbox"/>	blockAddress	hash
<input type="checkbox"/>	{strandId:"6RDxhfJujhPHh2hReBDfHW",sequenceNo:3}	{{LvyFHgbjXSv3YDjIQ7r6j29SkiWeqzfydEuJQ7v7+Dw=}}
<input type="checkbox"/>	{strandId:"6RDxhfJujhPHh2hReBDfHW",sequenceNo:42}	{{yawCXagQNaJ9nKnKZWwkTzY6SurIBkgHFlj9kbBf/Iw=}}
<input type="checkbox"/>	{strandId:"6RDxhfJujhPHh2hReBDfHW",sequenceNo:49}	{{BcbTAsvtxrNKNivWDIDMEnm6YtUj+lf5ZL9sRcMFEw=}}
<input type="checkbox"/>	{strandId:"6RDxhfJujhPHh2hReBDfHW",sequenceNo:51}	{{ZgUpnY47UdDPGuzXc9Bg80vepvJw8T3R0NP1z+K+9QY=}}

data
{VIN:"1N4AL11D75C109151",LicensePlateNumber:"LEWISR261LL",State:"WA",City:"Seattle",PendingPenaltyTicketAmount:90.25,ValidFrom:"2017-08-21T",ValidToDate:"2020-05-11T",Owners:{PrimaryOwner:{PersonId:"FHUURsqQD0bD0XJEpyxc1H"},SecondaryOwners:[]}}
{VIN:"1N4AL11D75C109151",LicensePlateNumber:"LEWISR261LL",State:"WA",PendingPenaltyTicketAmount:90.25,ValidFromDate:"2017-21T",ValidToDate:"2020-05-11T",Owners:{PrimaryOwner:{PersonId:"DdM1XCcexNyJM7V8NlGT4A"},SecondaryOwners:[]},City:"Everett"}
{VIN:"1N4AL11D75C109151",LicensePlateNumber:"LEWISR261LL",State:"WA",PendingPenaltyTicketAmount:90.25,ValidFromDate:"2017-21T",ValidToDate:"2020-05-11T",Owners:{PrimaryOwner:{PersonId:"DdM1XCcexNyJM7V8NlGT4A"},SecondaryOwners:[]},City:"Everett"}
{VIN:"1N4AL11D75C109151",LicensePlateNumber:"LEWISR261LL",State:"WA",PendingPenaltyTicketAmount:90.25,ValidFromDate:"2017-21T",ValidToDate:"2020-05-11T",Owners:{PrimaryOwner:{PersonId:"DdM1XCcexNyJM7V8NlGT4A"},SecondaryOwners:[{PersonId:"LtMNCxXCp4ult3zlGacHai"}]},City:"Everett"}

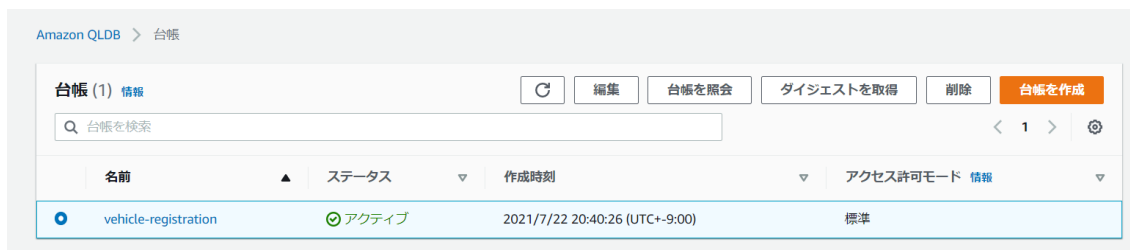
(データ個数は作業した分だけ増えていくので人により異なりますので気にしないで大丈夫です)

[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。

23. 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

24. 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



25. 確認画面で[保存]をおします




26. 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

```
{
  "digest": "A++4/NCrQJfP/Cn6MPAacZT7NvDx2iYUEMI774TgZkI=",
  "digestTipAddress": "{strandId: \"7YQHR0n2kVV4FMARuLKaHF\", sequenceNo: 6}",
  "ledger": "vehicle-registration",
  "date": "2021-03-30T14:52:50.948Z"
}
```

27. 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。
28. 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）
29. 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

 検証したいドキュメントを指定します。
台帳をクエリし、ドキュメント ID およびブロックアドレスを取得します。

台帳

vehicle-registration ▼

ドキュメント ID

F6qRPBF4oXJCM6CIWztpES

ブロックアドレス

{strandId: "6RDxhfJujhPHh2hReBDfHW", sequenceNo: 3}

30. [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。



検証に使用するダイジェストを指定します。
以前に保存したか、または値を入力したダイジェストの .ion.txt ファイルを選択します。

ダイジェストを選択

vehicle-registration-2021-07-22.ion.txt

ファイルサイズ (KB): 0.2

ファイルの最終変更日: 2021/7/22 21:56:08 (UTC+-9:00)

ダイジェスト

VesgJFHtuYJ3hLRAuDUP93jKIMaYKpDRNMzq5rFTPps=

ダイジェストヒントアドレス

{strandId:"6RDxhfJujhPHh2hReBDfHW",sequenceNo:58}

台帳

vehicle-registration

日付

2021/7/22 21:56:08 (UTC+-9:00)

31. [検証]をおします。以下のように正当性が検証されました。



ドキュメントは 2021/7/22 22:08:39 (UTC+-9:00) の時点で正常に検証されています
証明が完了しました (ハッシュとダイジェストが正常に再計算されました)。

証明

ブロック

リビジョンハッシュ

QP7aiCDVG0+ThXy7JaaCMdXBNEYBC+Y09RR5P8k7Eic=

ドキュメント ID: F6qRPBF4oXJCM6CIWztpE5

バージョン: 0

ブロックアドレス:

{strandId:"6RDxhfJujhPHh2hReBDfHW",sequenceNo:3}



ハッシュの計算



▼ 証明ハッシュ

D0NH7P5fXliJZN5whYvYZyKoiydVkJNaVJXuXoIT/ZmE=
m8lFhFugZ8ClDizr6eQz3uXlxKMOERdPMG9vrfzQH8I=
PbQizweGQUrRxK+K8LEFa20gliXrYCPqm4QEdl/u6LY=
AfbyN3fhqAw/aDZi1QnBooxPatveNV19fV+2+mCLzJ4=
u5lY/X73GiUgHXKMy7doFFf6g20PiL68tx0qE+N3T8=
/9hNgvsl2G5TkfSriUeMPTHyNSxh1nIFu04qR+F1iZE=
QTQmV3c7f2GRjoawMq07PSGT8kBBcbj8tUHrrZ9uAuE=
X9ACpZtbduz2Va6VB9MJjzhsX4u5BpAZiFiXivVZj8=
X9ACpZtbduz2Va6VB9MJjzhsX4u5BpAZiFiXivVZj8=
Qo9FKLbGausaT3RcVdEA7uc/6X5afj+25l08yN1GuPk=

トランザクション ID
0VwGhP2kObp4tjDp6sx19g

ブロックハッシュ
QP7aiCDVG0+ThXy7JaaCMdXBNEYBC+Y09RR5P8k7Eic=

トランザクション時間
2021-07-22T11:44:01.620Z

ブロックアドレス
ストランド ID: 6RDxhfJujhPHh2hReBDfHW
シーケンス番号: 3

ドキュメント ID

F6qRPBF4oXJCM6CIWztpES (リビジョン 0) ▼

ステートメント

1 INSERT INTO VehicleRegistration ?

ドキュメントエントリ

```
1 {
2   blockAddress: {
3     strandId: "6RDxhfJujhPHh2hReBDfHW",
4     sequenceNo: 3
5   },
6   hash: {{QP7aiCDVG0+ThXy7JaaCMdXBNEYBC+Y09RR5P8k7Eic=}},
7   data: {
8     VIN: "KM8SRDHF6EU074761",
9     LicensePlateNumber: "CA762X",
10    State: "WA",
11    City: "Kent",
12    PendingPenaltyTicketAmount: 130.75,
13    ValidFromDate: 2017-09-14T,
14    ValidToDate: 2020-06-25T,
15    Owners: {
```

[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

32. 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)
33. 22 番を実行し、乗り物情報を消します。
34. 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。
35. 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