IIJ bootCamp
MySQLを触ってみよう

IIJ s-fukuda
2021.07.30

MySQL を触ってみよう

本講義の目指すところ

- 技術職新卒採用者のうち、特定のミドルウェアに興味を持つ方へ基礎理解と経験値を上げる
- 発展的に、RDBMSへの興味を独力で深掘りしてくれるエンジニアへのきっかけづくり

MySQL を触ってみよう

想定する受講者

- 2021年 技術職の新卒採用者
- IIJグループ会社所属の技術職(あるみんで申し込み可能な方)
- BootCamp事前準備受講者
  - 環境OSは問わないが、LinuxVM環境を事前に構築が独力で可能
  - git, docker, docker-compose 等々の基本的コマンド操作が独力で行えること

MySQL を触ってみよう

予定する内容

- 事前の諸々、雑談、フリーディスカッション　15min程度
- ハンズズオン　残り全て

ハンズオン課題一覧

- 環境整備 git clone　… https://<hgoe>.git
- 起動、ログイン、パスワード
- Database / Table / Index … 各種作成
- Insert / Update / Delete …各種操作
- 応用編（やれる方/自由課題)

MySQL を触ってみる前に

みなさんの状況確認をさせてください

- RDBMSについて
  - 使った事ある (🖐)
  - RDBMSが好きで仕方ない (🖐)
- SQL/Queryについて
  - 何となく知っている/聞いた事はある (🖐)
  - 常にパフォーマンスを意識してSQLを書いている (🖐)
- RDBMSとその仲間
  - PostgreSQL、MySQLはいずれもRDBMSである (🖐)
  - MongoDBもRDBMSである (🖐)

MySQL を触ってみる前に

さて、RDBMSってそもそもなに？

- Database とは

  データベース（英: **database**, DB）とは、検索や蓄積が容易にできるよう整理された情報の集まり via Wikipedia

- Relational とは

  「関係」とは果たして何と何との関係でしょう？

MySQL を触ってみる前に

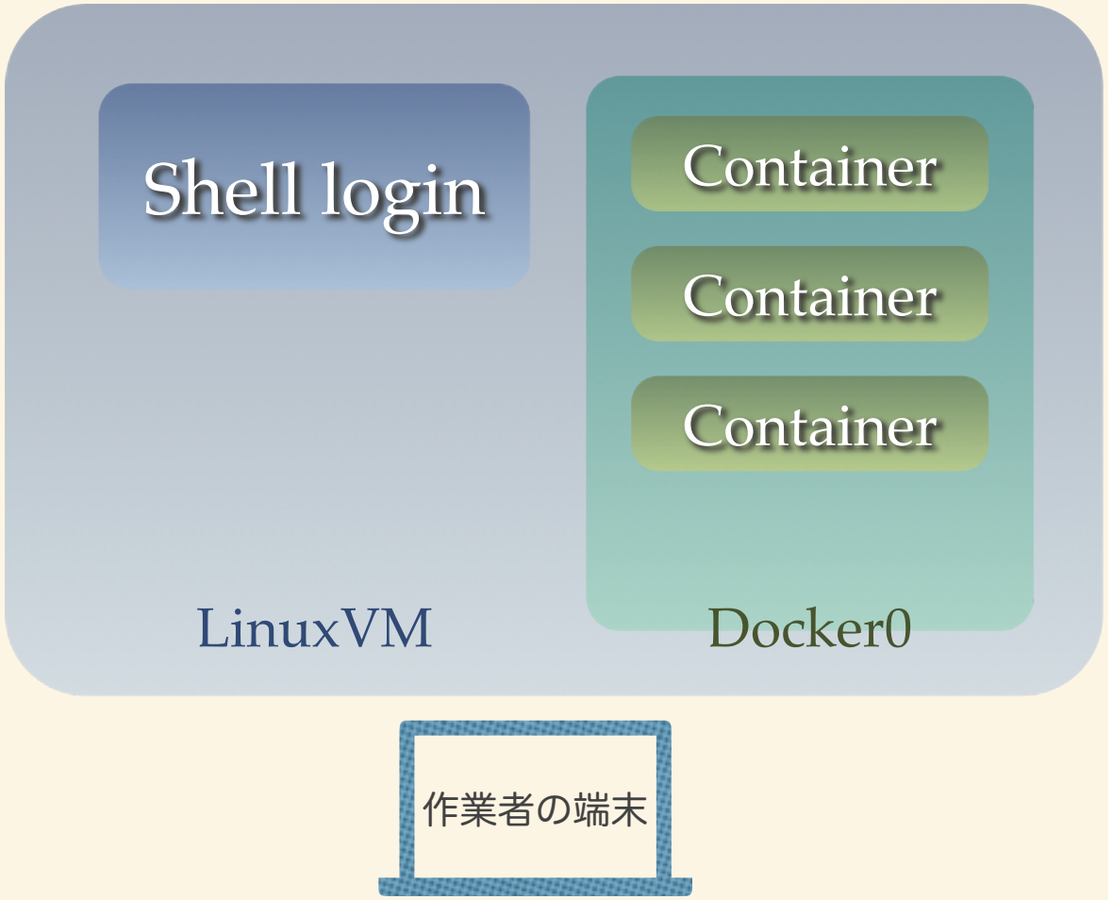
人事部がもしRDBMSをつかうとしたら？

- 社員表(社員番号、指名、年齢、入社日、部門id、給与id)
- 部門表(部門id、部門名、内線番号、所在地)
- 給与表(給与id、給与額)

この表とデータ構造において、何が適切(あるいは不適切か)を様々な視点/要件で捉え、データを使う側の状況応じて柔軟に”関係モデル”を構築するのに適切な「データ保管庫」が _____

カリキュラムの具体的な進め方

- Curlコマンドを操作する場合は右の図のLinuxVM上の「Shell」から行う事になります
- Docker コマンド、docker-composeコマンドも同様です
- MySQLサーバはコンテナとして立ち上げます
- MySQLサーバへアクセスするクライアント用にサーバとは別のコンテナを立ち上げて作業します



MySQL 環境整備

カリキュラム用Repository をCloneする // CP#1

- Proxy設定を確認の上、下記のURLへアクセス

```
$ git clone https://gh.iiji.jp/s-fukuda/bootcamp_mysql.git
```

```
$ cd bootcamp_mysql
```

- Group 会社の方は、下記のURLへアクセス

```
$ git clone https://github.com/isfukuda/bootcamp_mysql.git
```

MySQL 起動、ログイン

MySQLサーバを起動　// CP#1

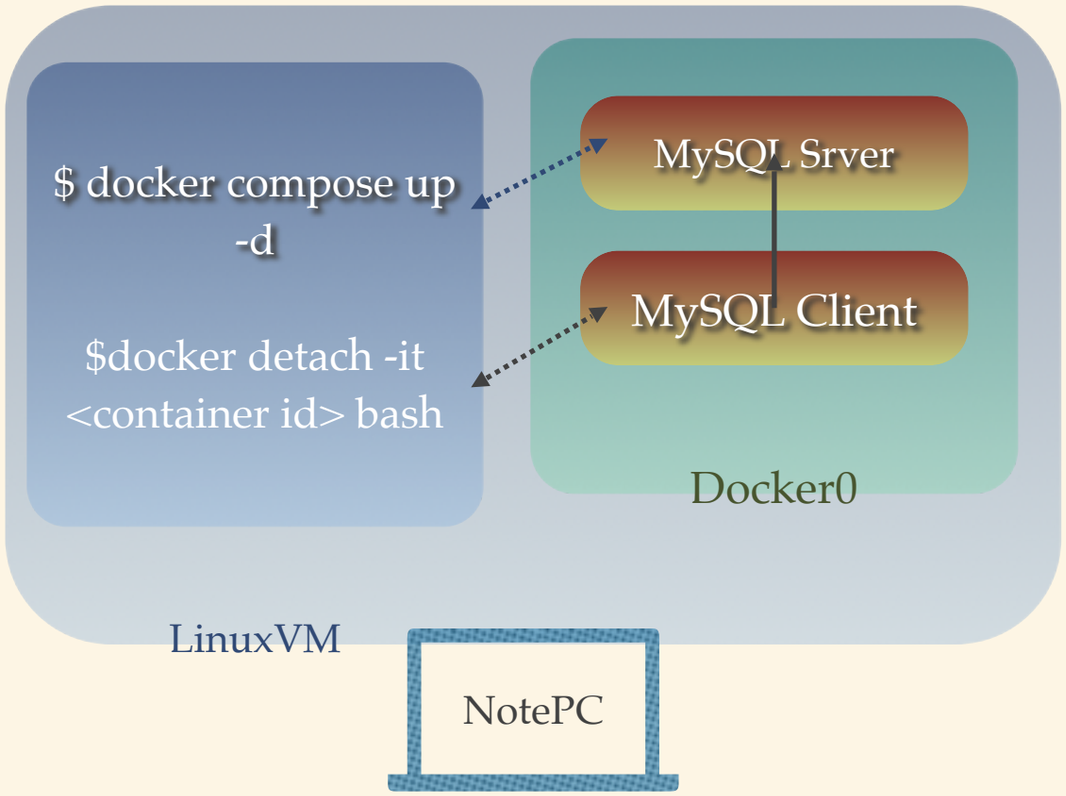
- docker-compose コマンドで起動

```
$ cd bootcamp_mysql

# docker-compose up -d

# docker ps
```

- MySQLコンテナが TCP/3306 portで立ち上がっている事を確認して下さい



Client用のcontaier起動、ログイン

```
# docker run -it —net=bootcamp_mysql_default —rm mysql:5.7 sh -c ‘exec  mysql  -u docker -h 172.17.0.1 -p’  //  Linux, MacOSX

# winpty docker run -it --net=bootcamp_mysql_default --rm mysql:5.7 sh -c 'exec mysql -u docker -h 172.17.0.1 -p’  // 一部windowsの方

mysql > show databases;
```

- コンテナ内部で上記のコマンドを打ち、test_db が含まれる事を確認します。以後の CP#3,4 はこのContainerログインのままで作業を続けてください

MySQL オペレーション

データベースオブジェクト作成 // CP#3

- Database 新規作成

  mysql > CREATE DATABASE class_of_2021 DEFAULT CHARSET utf8;

- Table 新規作成

  mysql > use class_of_2021

  mysql > CREATE TABLE people (

  ```
  id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
  name VARCHAR(8),
  age int(3),
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP);
  ```

  mysql > show tables

- Index 確認、作成

  mysql > CREATE INDEX ind_name ON people(name);

MySQL オペレーション

データ操作　// CP#4

- Insert

mysql > INSERT INTO people (name, age) VALUES (“kenji”, 14);

mysql > INSERT INTO people (name, age) VALUES (“yumiko”, 15);

mysql > SELECT count(*) FROM people;　　// Q. 結果はどうなりましたか？

- Update

mysql > UPDATE people SET age = 17 WHERE name = “kenji”;

mysql > SELECT name,age FROM people WHERE name = “kenji”; // Q. 結果はどうなりましたか？

- Delete

mysql > DELETE FROM people WHERE name = “kenji”;

mysql > SELECT count(*) FROM people;　// Q. 前回に比べ、今回のカウント結果はどうなりましたか？

先に進める方はこちらをどうぞ

どこまで出来るか？　// CP#5

1. Juan Manuel Lilloさん (53) が今日からclass_of_2021に参加されます。彼をメンバーに加えるSQLを教えてください
2. 2.を実行するSQLが書けけたら、実行してください
3. もし、3.が実行するに当たって何か問題があれば教えてください。可能であれば、その解決方法を教えてください
4. class_of_2021 を作成したデータベースですが、暗黙のうちに設定された多数のパラメータがあります。その中の一つである「トランザクション分離レベル」を確認するSQLを調べ、実行してみてください
5. カリキュラム中のDelete文、kenjiさんに何が起きたか日本語2文字で端的に答えてください

出来た方は答えを下記URLのコメントに追記する形で書き込んで下さい

回答はこちら

グループ会社の方はこちら

まとめ

MySQLに触れてみて

- ### MySQLサーバ環境準備

  docker/docker-composeを使いMySQLサーバ構築を簡略化させてもらいました。MySQL on dockerについては考慮すべきことが多々あります、この点は頭の片隅に必ず置いて覚えておいて下さい

- ### 基本的なDatabase操作を経験

  データベースへのアクセスにはmysqlクライアントを使い、基本的な知識抜きに「実践形式」でデータベースオブジェクトを作成、それに対してQueryを実行しました
  なお、アプリケーション技術者を目指す方は別途、開発言語/Database Driver経由でデータベースの操作を行う事をお勧めします

- ### 本日触れなかった事

  - MySQLサーバ初期構築から、rootログインとその後のDatabaseユーザ管理はしっかりと設計と実装が別途必須です
  - MySQLサーバ設定については一切触れていません、ご了承ください
  - 今回の講義ではINDEXを作成したのみです、有効なINDEXであるかは (Q. INDEX貼ればQueryって早くなるのか？)
    利用するデータ、Query、条件などによって変わります。とても深い内容につき割愛しています
  - データベース要件に合わせた論理設計、物理設計、運用設計等々全般

RDBMSを継続して勉強したい方むけ

RDBMSの世界、勉強する事がいっぱい

今回はACID特性について取り上げ、課題と共に理解を深めます

- トランザクション内の処理は，全てが実行されるか，全てが取り消されるかのいずれかである
- 同じデータベースに対する同一処理は，何度実行しても結果は同じである
- トランザクションの処理結果は，他のトランザクション処理の影響を受けない
- トランザクション完了後、ハードウェア障害が発生しても，更新されたデータベースの内容は保証される

補足課題 その1

RDBMSがデータベースの更新に対して保証すべきAtomicityの単位は次のうちどれか？

1. RDBMSの起動から停止まで
2. データベースのバックアップ取得から媒体障害の発生時点まで
3. トランザクション開始からコミット又はロールバックまで
4. チェックポイントから次のチェックポイントまで

補足課題 その2

MySQL 5.7 (InnoDB) トランザクション分離レベルのデフォルト値の特徴を述べたものは次のどれが適当か？

1. Serializable でもっとも分離レベルが高いが一方、パフォーマンスが犠牲になることもしばしばある
2. Read Committed なのでダーティーリードは起きないが、コミットデータが他のトランザクションから参照される
3. Repetable Read でトランザクション中、読み取り一貫性が得られる
4. Read Uncommited なので、ダーティーリードが起きるが全体的なパフォーマンスを考慮した結果と言える

補足課題 その3

トランザクション分離レベルのRead Uncommitedがハマるケースがあるとすれば次のどれが適当か？

1. 時系列データかつ過去のデータが更新されたり、削除される事が無いデータであり、パフォーマンス要件も高い為に採用
2. データ「操作」の信頼性がかなり下がるリスクを抱えても一貫性あるデータを可能な限り保証しなければならない
3. トランザクション中、読み取り一貫性が必須
4. アプリトップ画面に表示する素材に使うデータ(概要把握に使う数字)で都度更新すれば良いので、とにかく素早く表示したい

その他

- SQL/Queryをたくさん書く
  基本はSQL/Queryです、これ無くして何も始まらない。何が「正しい」SQL/Queryなのか？常に意識して描きましょう
- トランザクション分離レベル、もっと深掘りしたい
  「A Critique of ANSI SQL Isolation Levels」の論文を読む
- ”Hybrid”なデータベース要件に対応するデータベース管理者(DBA)
  何を最優先にするのか？何をDBAとして「守る」のか、を常に見極める
  DBAの現場：
  「更新され続けるData、Queryは不特定多数かつ同時多発的・だがパフォーマンスは落とすな！」
  「DB止めるな」
- MySQL 8.0 (筆者も未体験ゾーン)
  「InnoDB Cluster 」
  「ドキュメントストア」
  「Window関数のサポート」… etc