

APAC 2. Connectors

Accés a Dades



Continguts

Connexió i mode de funcionament general	3
Mode de connexió a la base de dades	3
Aclaracions i consells d'implementació	4
1. Estructuració de codi	4
2. Insercions en la taula. El mètode insertIntoTable.	6
3. Sobre les shells i la interacció amb l'usuari	7

En aquesta segona APAC d'Accés a dades, anem a posar en pràctica allò après respecta a connectors implementant una senzilla shell client d'accés a base de dades, a l'estil del client de mysql.

El nostre client ens permetrà principalment connectar-nos a un SGBD indicat, consultar les bases de dades que hi ha en ell, i realitzar algunes operacions amb elles.

Connexió i mode de funcionament general

En arrancar el nostre SGBD ens demanarà les dades del servidor i les nostres credencials:

```
1 $ gradle run --console plain
2 ...
3 # Server: 127.0.0.1
4 # Port: 3308
5 # Username: root
6 # Password:
7 # (root) on 127.0.0.1:3308>
```

Com veiem, una vegada connectats ens mostra el prompt de tipus: `(usuari)on IP:port>`. Aquest serà el mode de funcionament general, i des d'aquest podrem executar les següents ordres:

Ordre	Descripció
<code>show databases</code> o <code>sh db</code>	Mostra una llista amb les diferents bases de dades del sistema
<code>info</code>	Mostra informació sobre el SGBD i la connexió
<code>import Nom_del_script</code>	Permetrà executar un script sql indicant la ubicació del fitxer
<code>use Nom_de_la_BD</code>	Canvia al mode de connexió a la base de dades
<code>quit</code>	Ix de l'aplicació

Mode de connexió a la base de dades

Una vegada ens hem connectat a una base de dades, entrem en el mode de connexió a la base de dades, el qual s'indica amb un prompt com aquest:

```
1 # (root) on 127.0.0.1:3308> use BDJocs
2 # (root) on 127.0.0.1:3308[BDJocs] >
```

En aquest mode, i sobre la base de dades en la que ens trobem, podrem fer les següents operacions:

Ordre	Descripció
<code>sh tables</code>	Mostra les taules de la base de dades
<code>describe Nom_de_la_Taula</code>	Mostra una descripció dels camps de la taula indicada
<code>insert Nom_de_la_Taula</code>	Pregunta a l'usuari camp per camp de la taula, i inserix un registre a la base de dades
<code>select *</code>	Executarà una consulta select qualsevol en la BD
<code>quit</code>	Torna al mode general

Aclaracions i consells d'implementació

1. Estructuració de codi

Es recomana la implementació de les següents classes i mètodes:

Classe `com.ieseljust.ad.myDBMS.DBMan` Gestiona la connexió inicial a la base de dades, i implementa els següents mètodes:

Mètode	Descripció
<code>Main</code>	Pregunta les dades de connexió i crea un objecte de tipus <code>connectionManager</code> per iniciar la shell general

Classe `connectionManager` Gestiona la shell general, i conté els següents atributs i mètodes:

Atribut	Descripció
<code>String server;</code>	Adreça IP del servidor
<code>String port;</code>	port de MySQL
<code>String user;</code>	Usuari de la BD
<code>String pass;</code>	Password de l'usuari de la BD

Mètode	Descripció
<code>connectionManager()</code>	Constructor per defecte
<code>connectionManager(String server, String port, String user, String pass)</code>	Constructor parametritzat
public <code>Connection connectDBMS()</code>	Inicia una connexió a la BD i ens torna un objecte de tipus <code>Connection</code> que representa aquesta connexió
public void <code>showInfo()</code>	Mostra la informació del servidor
public void <code>showDatabases()</code>	Mostra les bases de dades del servidor
public void <code>importScript(String script)</code>	Carrega i executa un script SQL
public void <code>startShell()</code>	Mostra el prompt i gestiona les ordres dle mode general

Classe `databaseManager` Gestiona la shell de gestió de base de dades, i conté els següents atributs i mètodes:

Atribut	Descripció
<code>String server;</code>	Adreça IP del servidor
<code>String port;</code>	port de MySQL
<code>String user;</code>	Usuari de la BD
<code>String pass;</code>	Password de l'usuari de la BD
<code>String dbname;</code>	Nom de la base de dades a la que estem connectats

Mètode	Descripció
<code>databaseManager(String server, String port, String user, String pass, String dbname)</code>	Constructor parametritzat
public <code>Connection connectDatabase()</code>	Es connecta a la base de dades especificada en l'atribut <code>dbname</code>

Mètode	Descripció
<code>public void showTables()</code>	Mostra les taules de la base de dades
<code>public void insertIntoTable(String table)</code>	Inicia l'assistent d'inserció de dades en la taula indicada
<code>public void showDescTable(String table)</code>	Mostra la informació de les metadades de la taula
<code>public void executeSelect(String query)</code>	Mostra els resultats d'una consulta indicada per la línia d'ordres
<code>public void startShell()</code>	Gestiona les ordres de la shell de manipulació de la base de dades

2. Insercions en la taula. El mètode `insertIntoTable`.

Aquest possiblement siga el mètode més complicat de tota l'aplicació, ja que cal consultar primerament les metadades per saber l'estructura de la taula, demanar-les a l'usuari, i inserir-les a la base de dades, tot respectant els diferents tipus i formats.

La recomanació general és mantenir al menys tres vectors o `ArrayLists` per controlar:

- La llista de camps que caldrà inserir a la taula
- La llista de valors que els anem a assignar
- La llista de tipus de dada que tindrà cadascun

A més, caldrà generar la llista de «placeholders», amb tants «?» com camp anem a afegir (pot ser un `arraylist` que després convertim amb `toString`, o bé generar directament la cadena de caràcters).

Per a tot açò ens seran d'utilitat les següents columnes que obté `getCols`:

- Columna 4: Nom de la columna de la BD
- Columna 6: El tipus de dada de la columna
- Columna 18: Indica si és nullable o no
- Columna 23: Indica si és autoincremental

En cas de tractar-se d'una columna autoincremental, com que a aquesta li s'assigna valor automàticament, no hem d'emmagatzemar cap informació sobre ella, ni incloure-la a la consulta.

Un altre punt complicat d'aquest mètode serà l'assignació de paràmetres, ja que en funció del tipus, haurem d'utilitzar una o altra funció en el statement. Com a mínim, haurem de tindre en compte el següent tipus de dades:

- Si la dada és de tipus «INT», caldrà utilitzar `statement.setInt(posició, Integer.parseInt(valor))`
- Si la dada és de tipus «DATETIME», haurem d'utilitzar `statement.setDate(posició, java.sql.Date.valueOf(valor))`;
- En altre cas (podem comprovar també altres tipus abans), si no és de cap dels tipus anterior, la tractem com a String (pot ser VARCHAR, TINYINT...), pel que haurem d'utilitzar: `Statement.setString(i+1, LlistaValors.get(i))`;

3. Sobre les shells i la interacció amb l'usuari

Com hem vist, hi ha un parell de shells que caldrà implementar, per tal de capturar les diferents ordres possibles en cadascuna.

Cal tindre en compte que l'estructura de control `switch..case` no accepta expressions regulars, pel que potser, en algun moment, tinguem problemes a l'hora de gestionar ordres com `use Nom_de_la_BD` o `describe nom_de_la_taula`

Per tal de comprovar aquestes ordres, podem fer ús de switches anidats, on el segon s'ubicaria en el default del primer. L'estructura seria semblant a aquesta:

```
1  command = keyboard.nextLine();
2
3  switch (command){
4      case "sh tables":
5      case "show tables":
6          this.showTables();
7          break;
8
9      case "quit"
10         break;
11
12     default:
13         // Si no s'ha complert cap dels casos anteriors, dividim l'ordre i
14         // analitzem les parts:
15         String[] subcommand=command.split(" ");
16         switch(subcommand[0]){
17             case "describe":
18                 this.showDescTable(subcommand[1]);
19                 break;
20             case "insert":
21                 this.insertIntoTable(subcommand[1]);
22                 break;
23             ...
24             default:
25                 System.out.println(ConsoleColors.RED+"Unknown db option"+
26                                     ConsoleColors.RESET);
```

```
25     }  
26 }
```