## SWP\_Übung

#### Beispiel 1: Zugriff auf eine Adress-Datenbank mit JDBC

#### Lösungsidee:

Für diese Übung wird der DBeaver verwendet.

Zuerst muss über das Programm die derbyrun.jar verwenden. Dazu wurde der lib Ordner über die Projekt Struktur eingebunden, damit man die .jar-Dateien auch verwenden kann:

Markieren von Dependencies:

File -> Project Structure -> Modules -> + (ganz rechts an der Leiste) -> JARs or directories -> in das Projekt hinein navigieren und den lib Ordner auswählen

Dann wurde die run()-Methode von DerbyStart zum Starten festgelegt.

Festlegen einer Start-Klasse:

Rechts oben in der Projektansicht den Reiter ausklappen (neben dem sich der Hammer befindet) -> Edit Configurations -> + (links oben in der Ecke) -> Application -> (optional) einen Namen hergeben -> Main class auf die drei Punkte daneben -> Project -> auf derbyrun.jar navigieren und auswählen -> danach noch im Configurationsfenster bei Program arguments ‚server start‘ angeben und speichern.

Zum Erstellen der Connection:

Im DBeaver auf ‚New Database Connection’ -> Derby Server -> Host: localhost -> Port: 1527 -> Schema: Persons -> User name: user -> Password user  
WICHTIG: Es darf nicht der neueste Treiber verwendet werden, sondern nur 10.14.2.0  
Edit Driver Setting -> Download/Update -> auf die Version klicken und ändern, sollte eine andere installiert sein.

Das folgende CREATE TABLE Statement müsste bitte in der neuen Connection ausgeführt werden. Es wird auch als Textfile dabei sein!

**CREATE** **TABLE** APP.PERSONS (

ID **INT** **NOT** **NULL** **GENERATED** **ALWAYS** **AS** **IDENTITY**,

FIRSTNAME **VARCHAR**(50) **NOT** **NULL**,

LASTNAME **VARCHAR**(50) **NOT** **NULL**,

CITY **VARCHAR**(50) **NOT** **NULL**,

POSTALCODE **INT** **NOT** **NULL**,

ADDRESS **VARCHAR**(50) **NOT** **NULL**,

PHONENUMBER **VARCHAR**(50) **NOT** **NULL**,

**CONSTRAINT** ID\_PK **PRIMARY** **KEY** (ID)

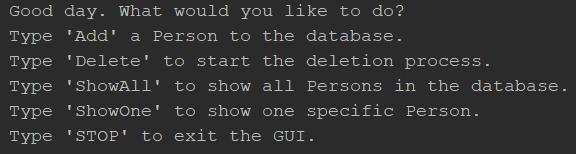
);

Wie man nun das Programm startet:

Zuerst den derbyserver starten. Dann im DBeaver die Connection anlegen und das CREATE TABLE Statement ausführen. Dann zu guter Letzt PseudoGUI ausführen! (Alias test01).

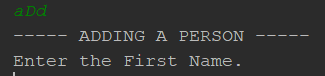
#### Testfälle:

Das ganze Beispiel basiert auf einem einzigen Testfall (test01) weil man von dort aus über die Konsole alle Entscheidungen treffen kann.

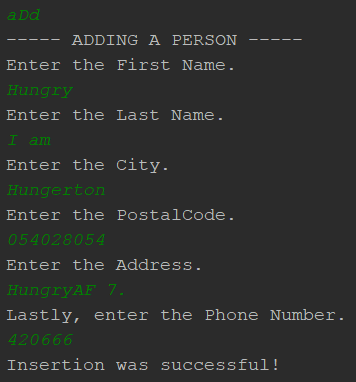


Diese Nachricht wird solange kommen, bis man eine der gezeigten Wörter eintippt. Man kann beliebig oft hintereinander Add, Delete, ShowAll und ShowOne wählen, aber bei STOP hört das Programm auf zu laufen.

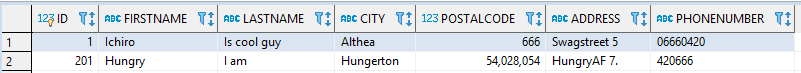
Anmerkung: NICHT Case-Sensitive! Aber dafür empfindlich, wenn was anderes eingetippt wird 😊



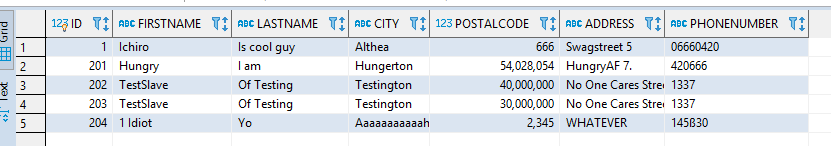
Wirklich nicht case-sensitive.



Hier wurde eine Person eingefügt. ID gibt man nicht selbst an, die wird von der Tabelle aus automatisch generiert!

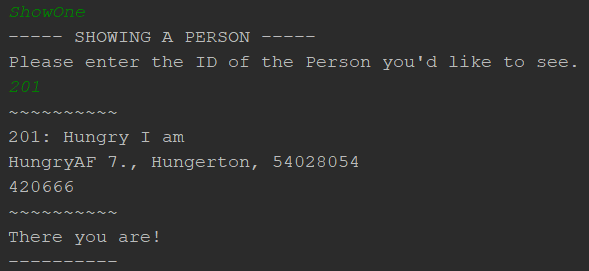


Der Eintrag ist nun in der Datenbank! (SELECT \* FROM APPS.PERSONS)

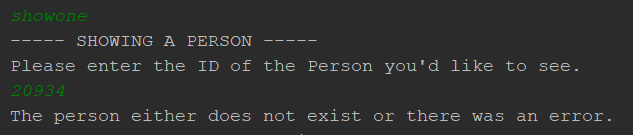


In der Datenbank sind nun all diese Personen.

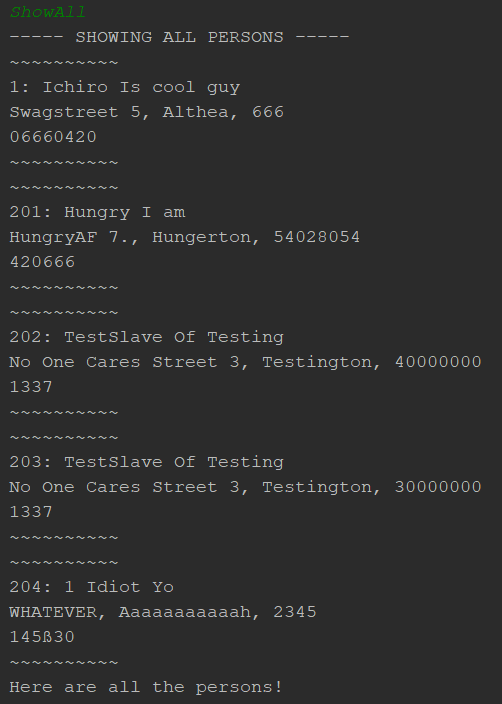
Nun wird mit ShowOne eine Person angezeigt.



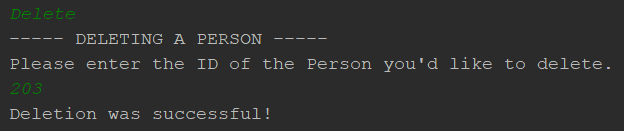
Hier wird erfolgreich eine Person angezeigt, die es gibt.



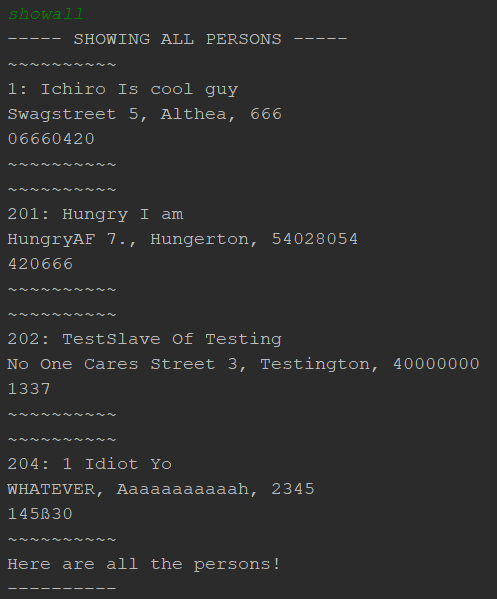
Hier wird versucht eine Person anzuzeigen, die es nicht gibt.



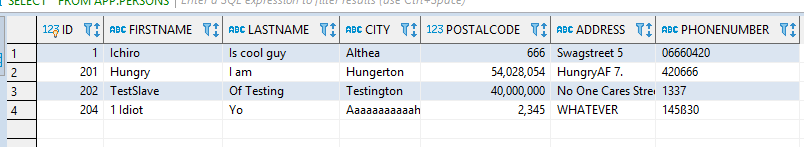
Hier werden alle Personen in der Datenbank angezeigt.



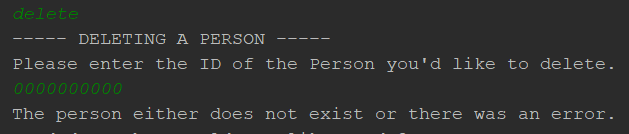
Hier wird die Person mit der ID 203 gelöscht.



203 ist nicht mehr zu sehen, da diese Person gelöscht wurde!



Auch im DBeaver ist sie weg.



Wenn man versucht eine Person zu löschen, die es nicht gibt.