

---

## Praktikum

# Datamining zur automatischen Spielanalyse im Fußball

---

Auf diesem Aufgabenblatt geht es um die Visualisierung der Daten. Einige Visualisierungstypen sind bereits vorhanden und können einfach ausprobiert werden.

Einige einfache Visualisierungen kann man durch Rechtsklick auf das Resultat einer Query anzeigen. Holen sie sich dazu z.B. alle Daten der Episoden und probieren sie die verschiedenen Möglichkeiten der Visualisierungen aus.

Eine andere Möglichkeit um Visualisierungen zu erzeugen und diese dann auch zu speichern ist der Visualisierungsdialog. Öffnen sie dazu das Kontextmenü zu **Visualisations** und wählen Sie **Manage...**

Es erscheint ein Fenster, das in 3 Teile eingeteilt ist.

Im oberen Teil sehen Sie links den Namen der Visualisierung und rechts den Typ (einfach oder kombiniert).

Im mittleren Teil sehen Sie für einfache Visualisierungen links unter **Stored Query** die Abfrage, die zur Visualisierung gehört. Rechts können Sie auswählen welche Art der Visualisierung Sie haben wollen.

Im unteren Teil sehen Sie im ersten Block die Ausgabefelder (Spalten) der SQL-Abfrage mit Namen und ihren Typen. Im zweiten Block sehen Sie die Spalten, die von der Visualisierung pro Datensatz erwartet werden. Zu jeder erwarteten Spalte gibt es vier Felder. Das erste Feld gibt den Namen der Spalte in der Visualisierung an, das zweite Feld gibt den Datentyp der Spalte an (z.B. **FLOAT** oder **TEXT**). Das dritte Feld gibt an, ob die Spalte erforderlich (!) oder optional (\*) ist. Die vierte Spalte gibt einen Repräsentationstyp an, der die Art der Eingabe (z.B. **PlayerID**) beschreibt. Diese Typen folgen einer Hierarchie, die man unter **Tools**→**Type Domain Viewer** einsehen kann. Der Datentyp der Query muss mit dem Datentyp der Spalte übereinstimmen, der Repräsentationstyp ist nur als Hilfestellung für das automatische Generieren der Visualisierungen gedacht. Im dritten Block sehen Sie die Zuordnung der Ausgabefelder der SQL Abfrage und der erwarteten Felder der Visualisierung.

Für kombinierte Visualisierungen können sie im mittleren Teil die zu kombinierenden Visualisierungen auswählen. Momentan können allerdings nur Feldvisualisierungen (**Association**, **Lines**, **Piecharts**, **Positions**, **Spheres**) kombiniert werden. Ausserdem kann die Reihenfolge der Visualisierungen geändert werden um die angezeigten Schichten zu ordnen.

Im unteren Teil können Variablen, die in Queries mittels  $\${<name>}$  eingeführt werden können einer kombinierten Visualisierungsvariablen zugewiesen werden.

Sobald Sie **Execute** drücken, sehen Sie das Resultat der Visualisierung.

## Aufgaben:

1. Machen Sie sich etwas vertraut mit den verschiedenen Modulen der Visualisierung, indem Sie einige einfache SQL-Abfragen erstellen und mit den Modulen darstellen.
2. Erstellen Sie eine eigene Visualisierung mit den vorgegebenen Modulen. Denken Sie sich etwas interessantes aus. Beispiele sind:
  - Ballbesitz als Piechart
  - Positionen der Torschüsse als Positions
  - erfolgreiche Pässe als Lines
  - ...

Natürlich müssen Sie vorher die passende SQL-Abfrage erstellen.

3. Erstellen Sie mit Java eine eigenen Feldvisualisierung. Gehen sie hierbei von der Klasse `LineVisualisation` aus, die sie als Sourcecode unter `~p019/fipm/LineVisualisation.java` finden. Die Datei muss irgendwo im `CLASSPATH` zu finden sein (dazu muss die Datei `build.xml` verändert werden) und der Paketname der Verzeichnissruktur angepasst werden.

Die Signatur `signature` beschreibt die im mittleren Block der einfachen Visualisierungen angegebenen Spalteneigenschaften, diese müssen sie ihren Bedürfnissen anpassen (`owl:Thing` bezeichnet dabei einen noch nicht verwendeten Repräsentationstyp in OWL). Die Methode `setDataImpl` ist der Hauptbestandteil dieser Klasse und ist für das Einlesen der Daten verantwortlich. Die Daten werden als Query-Results (siehe JavaDocs) in Form einer Tabelle übergeben. Die Methode liest die Daten zeilenweise ein und fügt `Shapes` (Grafikobjekte) zum Vector `shapes` hinzu, der von der Superklasse `FieldVisualisation` ausgegeben wird. Es können in JAVA definierte `Shapes` oder selbst generierte von `Shape` abgeleitete Klassen verwendet werden.

Die Klasse `AttributedShape` und `colorCols` können für das Praktikum ignoriert werden. Ersteres definiert eine Semantik auf den dargestellten Daten. Mit `colorCols` können Farben definiert werden, die zur Darstellung je nach Spaltentyp/-name verwendet werden.

Ihre Aufgabe ist nun eine Visualisierung zu generieren, die statistische Daten wie z.B. Ballbesitz eines Spielers an einer in der Query vorgegebenen Position z.B. als Balken auszugeben.

Die fertige Visualisierung muss dann noch in die Hauptdatenbank (siehe z.B. `Preferences...`) eingetragen werden um im Visualisierungsdialog zu erscheinen. Dazu muss ein Name und die Klasse mit ihrem Paket in die Tabelle `visualisation_modules` eingetragen werden.