

# Konzeptuelles

1. Die Authentifikation und Autorisation lassen sich mit dem Einbinden des Moduls Spring-Security in das System hinzufügen. Dabei wird die Applikation sofort mit einer Standard Sicherheitskonfiguration geschützt, wobei der Benutzername und Passwort bei jedem Serverstart erneut erzeugt werden und in der Konsole ausgegeben.

Die Authentifizierung kann erweitert werden, indem die Authentifizierungsdaten verschlüsselt werden und in der Datenbank gespeichert werden.

Bezüglich der Autorisation kann sie so erweitert werden, dass mehrere Gruppen definiert werden, z.B. Administratoren und Verwalter und dass bestimmte Bereiche bzw. Funktionen im Code nur für bestimmte Gruppen zugelassen werden.

Mit dem Spring Framework lässt sich relativ einfach diese Anbindung zwischen den Gruppen und den zugelassenen Bereichen realisieren. Dabei muss das Framework konfiguriert werden, um die Gruppeninformationen aus der Datenbank lesen zu können. Anhand Annotationen auf der jeweiligen Funktion/Klasse wird der Zugriff für den authentifizierten Benutzer freigeschaltet oder gesperrt.

2. Anhand eines Java Profiler, z.B. YourKit, kann man sich mit der Applikation in Laufzeit verbinden. Dabei können verschiedene Parameters analysiert werden, u.a. Speichernutzung, Klassen-Load und laufende Threads. Dabei werden auch auftretende Deadlocks und Memory Leaks identifiziert und ausgegeben.

3. Die Applikation besteht aus einem Frontend und einem Backend, die miteinander asynchron kommunizieren. Das heißt im Fall der Create-Methode wartet das Frontend nicht aktiv auf eine Antwort und der Benutzer kann weiterarbeiten. Erst wenn das Backend eine erfolgreiche Antwort zurückschickt, fügt das Frontend die Daten der erzeugten Person in die vorhandene Personen-Liste. Das System kann erweitert werden, sodass der Benutzer Protokolleinträge über den Zustand der Vorgänge sehen kann.

4. Unten sind die Skizzen für ein mögliches ERM abgebildet, sowie das entsprechende Datenbankmodell:

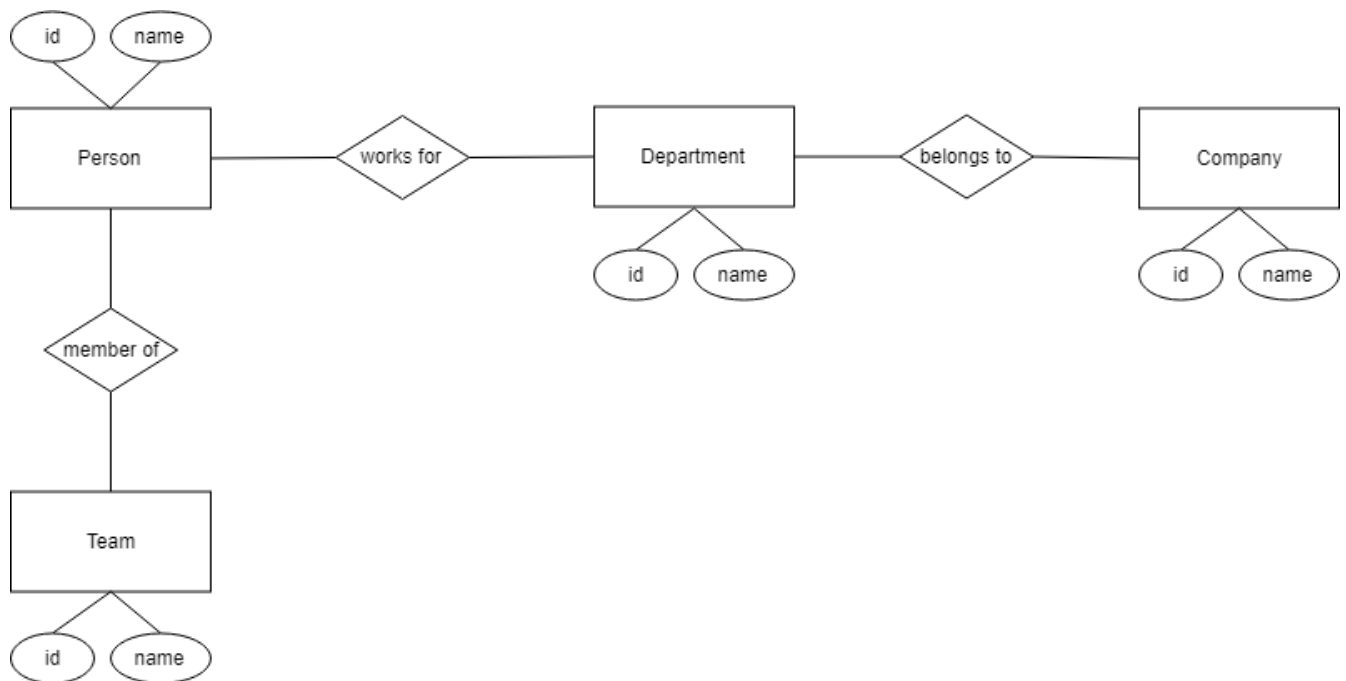


Abbildung 1: Das ERM Modell

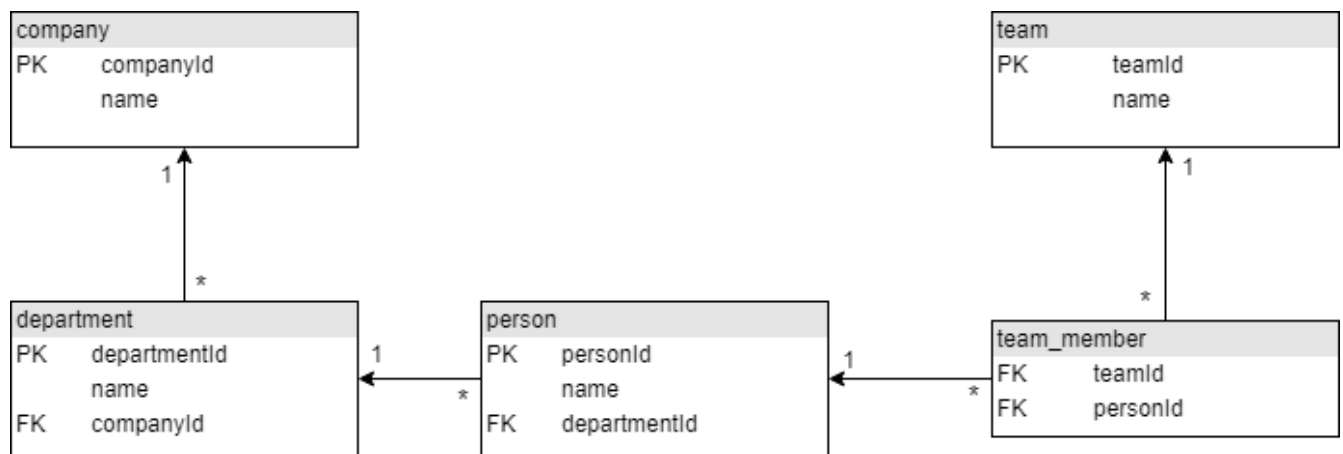


Abbildung 2: Das relationale Datenbankmodell