

Online Jobsuche Plattform: CareerSuccess

Meilenstein 1: Anforderungsanalyse & Konzeptioneller Entwurf

Online Jobsuche Plattform bietet Arbeitgebern neue Stellenangebote zu posten und Bewerbern auf diese Stellen sich zu bewerben. Online Jobsuche Plattform hat AdministratorInnen und BenutzerInnen.

Erforderliche Entitäten:

- Administrator (stark Entity)
- Benutzer (stark Entity)
- Bewerber (stark Entity)
- Arbeitgeber (stark Entity)
- Stellenangebote (schwach Entity)
- Lebenslauf (schwach Entity)
- Kategorie (stark Entity)

Wenn das nötig ist, der Administrator kann mehreren Benutzern blockieren, also die Kardinalität ist 1:n. „Administrator“ Entity hat folgende Attributen:

- Login
- Kennwort
- eindeutige Personalnummer (Schlüssel Attribute)

„Benutzer“ Entity hat folgende Attributen:

- Login
- Kennwort
- Benutzer_id (Schlüssel Attribute)

Benutzer können miteinander kommunizieren. Das bedeutet, dass zwischen Benutzern eine Unäre Beziehung n zu m gibt.

Benutzern sind entweder BewerberInnen oder ArbeitgeberInnen. Es ist eine „is-a“ Generalisierungsbeziehung.

„Arbeitgeber“ Entity hat folgende Attributen:

- Firma Namen
- Adresse
- Bereich

- E-Mail-Adresse
- Telefonnummer
- Ort

„Bewerber“ Entity hat folgende Attributen:

- Nachname
- Vorname
- Geburtsdatum
- E-Mail-Adresse
- Telefonnummer
- Adresse
- Ort

Ein Arbeitgeber kann mehrere neue Stellenangebote erstellen und diese haben eine Beginn- und eine Endzeit. Es gibt Binäre Beziehungen 1: n zwischen Arbeitgeber und Stellenangebote. Diese Beziehung hat zwei Attribute:

- Beginn Zeit
- Endzeit

„Stellenangebot“ Entity hat folgende Attributen:

- vacancyid (Schlüssel Attribute)
- Beschreibung
- Job Titel
- Gehalt
- Anstellungsart

Mehrere Bewerber können sich auf mehreren offene Stellen bewerben. Es ist eine Binäre Beziehung n: m zwischen „Bewerber“ und „Stellenangebote“ Entitys.

Ein Bewerber darf nur ein Lebenslauf hochladen. Es ist eine Binäre Beziehung 1: 1 zwischen „Bewerber“ und „Lebenslauf“ Entitys.

„Lebenslauf“ Entity hat folgende Attributen:

- CVid (Schlüssel Attribute)
- Beschreibung
- Schule Name

- Hochschule Name
- datum
- Position

Attribute „Datum “ zeigt wann Bewerber das Lebenslauf eingefügt hat. Attribute „Position“ hat Information über gewünschte Position von Bewerbern.

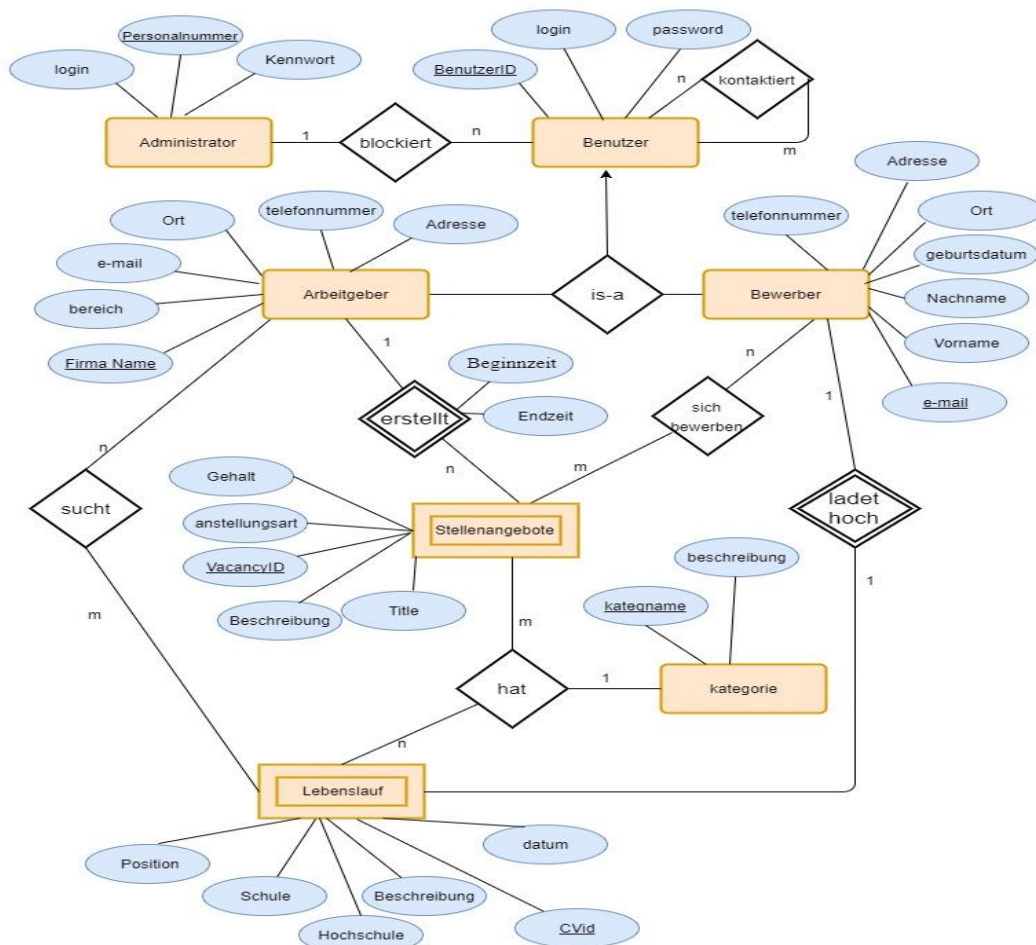
ArbeitgeberInnen können mehrere Lebenslaufen suchen. Es ist eine Binäre Beziehung n: m zwischen „Arbeitgeber“ und „Lebenslauf“ Entitys.

Zwischen „Stellenangebote“ und „Lebenslauf“ und „Kategorie“ Entitys gibt es eine trinäre Beziehung. Mehreren Stellenangeboten und Lebenslaufen dürfen eine Kategorie haben.

„Kategorie“ Entity hat folgende Attributen:

- Beschreibung
- Kategname (Schlüssel Attribute)

Jobsuche Plattform



Meilenstein 2: Logischer Entwurf

- Administrator (Login, Kennwort, Personalnummer)
 - PK: {Personalnummer}

- Benutzer (Login, Kennwort, Benutzer_id, *Blockierung-Personalnummer*)
 - PK: { Benutzer_id }
 - FK: Benutzer. Blockierung-Personalnummer <> Administrator.
Blockierung-Personalnummer

- Arbeitgeber (Firma_name, *Benutzer_id*, Adresse, telefonnummer, Ort, Bereich, e-mail)
 - PK: {Firma_name}
 - FK: Arbeitgeber. Benutzer_id<> Benutzer. Benutzer_id

- Bewerber (*Benutzer_id*, Geburtsdatum, Nachname, Vorname, Adresse, telefonnummer, Ort, e-mail)
 - PK: { e-mail }
 - FK: Bewerber. Benutzer_id <> Benutzer. Benutzer_id

- Kategorie (Beschreibung, kategname)
 - PK: { kategname }

- Lebenslauf (CVid, updatedatum, position, Beschreibung, Schule, Hochschule, *gehören- kategname, e-mail*)
 - PK: {CVid}
 - FK: Lebenslauf. E-mail <> Bewerber. E-mail
 - FK: Lebenslauf.kategname<>Kategorie.kategname

- Stellenangebote (Ort, VacancyID, Beschreibung, Title, Gehalt, *erstellt-firma_name, gehören- kategname*, Endzeit, Beginzeit)
 - PK: {VacancyID}
 - FK: Stellenangebote. firma_name <> Arbeitgeber. firma_name

- FK: Stellenangebote. *kategname* <> *Kategorie.kategname*
- StellenangeboteBewerben (*e-mail*, *VacancyID*)
 - FK1: StellenangeboteBewerben. E-mail <> Bewerber. E-mail
 - FK2: StellenangeboteBewerben.VacancyID <> Stellenangebote.VacancyID
- LebenslaufSuchen (*firma_name*, *CVid*)
 - FK1: LebenslaufSuchen. *_Benutzer_id* <> Arbeitgeber. *Firma_name*
 - FK2 : LebenslaufSuchen.CVid <> Lebenslauf.CVid
- Zugehoerigkeit (*VacancyID*, *CVid*, *kategname*)
 - FK1: Zugehoerigkeit.VacancyID <> Stellenangebote.VacancyID
 - FK2 : Zugehoerigkeit.CVid <> Lebenslauf.CVid
 - FK2 : Zugehoerigkeit.kategname <> Kategorie.kategname
- Kontakt (*E-mail*)
 - FK: Kontakt.E-mail <> Benutzer.E-mail

Meilenstein 4: Implementierung

Java Implementierung

Das Programm implementiert eine Klasse namens "TestDataGenerator". Die "main"-Funktion stellt dann die Verbindung zur Datenbank her, indem sie `Connection con= DriverManager.getConnection (database, user, pass)` verwendet und erstellt dann eine Anweisungsvariable (`Statement stmt = con.createStatement()`). Mit diesem Anweisungsvariable und mittels «executeUpdate» wird später die INSERT-Anweisungen ausgeführt werden, um die Datenbank mit Testwerten zu füllen. Für jede Ausfüllung den Tabellen wird for-Schleifen benutzt. Unterschied zwischen Datensätzen für Tabellen wird durch verschiedene Zahlen implementiert.

„TestDataGenerator“ erstellt insgesamt 7 Tabellen mittels 7 for-Schleife:

benutzer – 2000 Rows;

bewerber – 100 Rows;

arbeitgeber – 100 Rows;
kategorie – 16 Rows;
stellenangebote – 1010 Rows;
lebenslauf – 100 Rows;
stellenangeboteBewerben – 25 Rows;

PHP Implementierung

Das PHP Projekt besteht aus 9 PHP-Dateien. Für 8 PHP-datei gibt es „select“ und „insert“. Noch ein PHP-Datei zeigt VIEWS.