

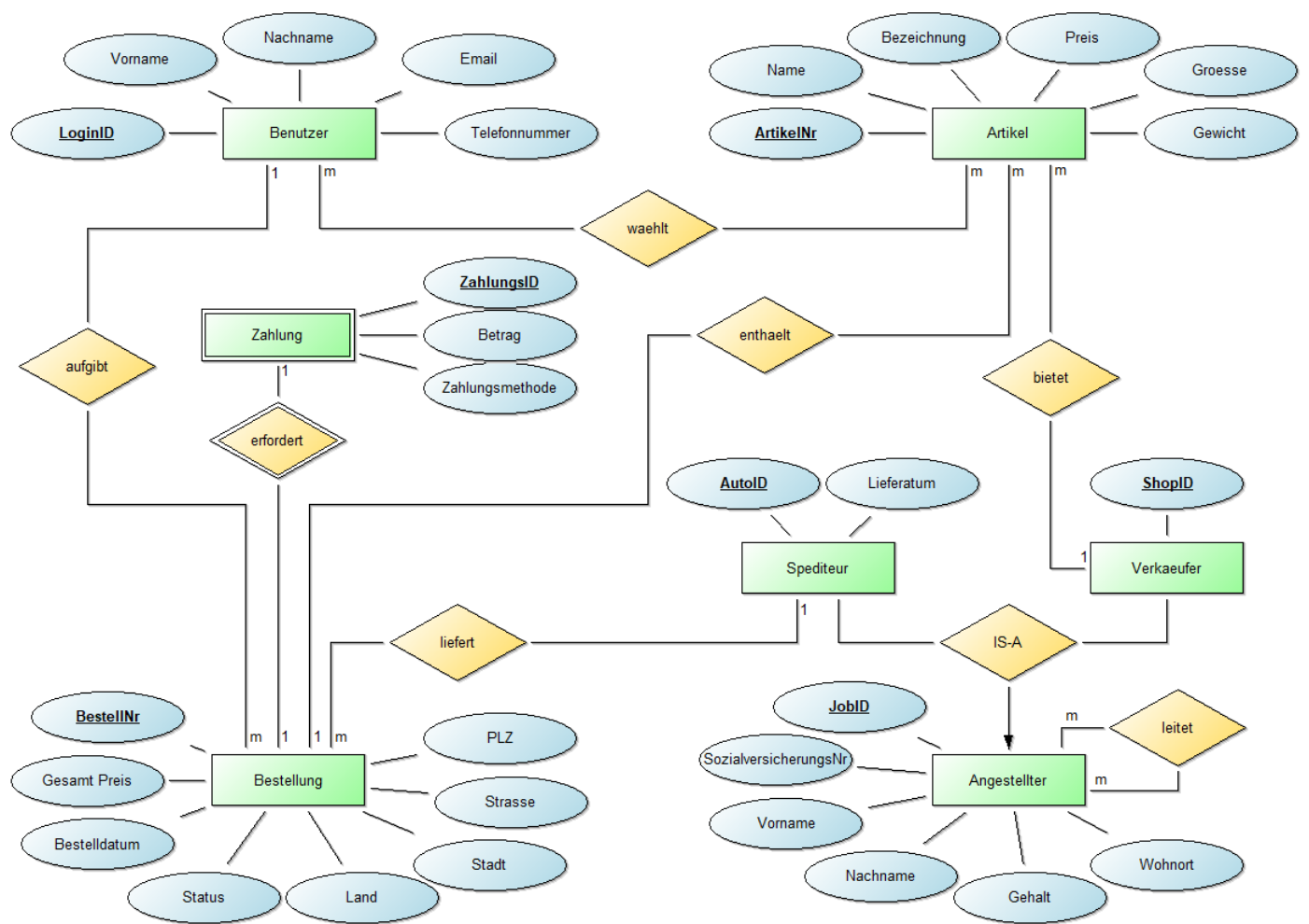
Online Shop

Anforderungsanalyse

Ein ausgereiftes und gut funktionierendes Online-Shop-System erfordert eine zuverlässige Kommunikation zwischen den Kunden, Verkäufern und Spediteuren und dem System selbst. Ausserdem muss es die verfügbaren Bestände und Artikel veranschaulichen, Daten der Benutzer sowie Zahlungs- und Bestelldetails sicher halten, und auch viele andere Bequemlichkeiten bieten. Ein Abstraktionsmodell, um ein Konzept aller Aufgaben zu erstellen, die zusammenarbeiten ist erforderlich. Diese Abstraktion habe ich mithilfe vom ER-Modell realisiert.

Die Organisation vom Online-Shop ist in den folgenden Sektionen unterteilt.

Der User(Login-ID, Vorname, Nachname, Adresse, Email, Geburtsdatum, Telefonnummer) besitzt einen Profil wo kann er alle Artikel(ArtikelNr, Gewicht, Bezeichnung, Preis, Grösse, Verfügbarkeit), die gesucht werden, einsehen und speichern. Ein Angestellter(Job-ID, Versicherungsnummer, Vorname, Nachname, Gehalt). Das Unternehmen stellt Mitarbeiter mit unterschiedlichen Aufgaben ein. Einer der Typen sind die Verkäufer(ShopID, Adresse), der kann mehrere Artikel anbieten. Der andere Typ ist der Spediteur(AutoID, Lieferadresse, Datum), der die Lieferungen durchführt. Wenn ein Angestellter lange im Unternehmen arbeitet, ist er verpflichtet, die Neuankömmlinge am Anfang ihrer Karriere zu navigieren und unterstützen, damit sie sich leichter an den Job gewöhnen können. Wenn der Benutzer eine Bestellung(Bestellnummer Lieferdatum, Status, PLZ, Strasse, Stadt, Land, Zahlungsmethode) aufgibt, geht diese an den Spediteur mit der Lieferadresse, er kann auch mehrere Pakete an einem einzigen Tag zustellen. Ein Artikel benötigt Zahlung(ZahlungsID, Status).



Logischer Entwurf

Benutzer(LoginID, Vorname, Nachname, Email, Telefon)

SK={LoginID, Email, Telefon}

PK = LoginID

Angestellter(JobID, SozNr, Vorname, Nachname, Gehalt, Wohnort)

SK={JobID, SozNr}

PK = JobID

Spediteur(AutoID, *JobID*, Lieferadresse)

SK= {AutoID}

JobId must be candidate key

PK= AutoID

FK= Spediteur.JobID \diamond Angestellter

Verkaeufer(ShopID, *JobID*)

SK={ShopID}

JobId must be candidate key

PK= ShopID

FK =Verkaeufer.JobID \diamond Angestellter

Bestellung(BestellNr, *LoginID*, *AutoID*, Gesamtpreis, Bestelldatum, Status, Land, Stadt, Strasse, PLZ)

SK={BestellNr}

PK=BestellNr

FK1=Bestellung.LoginID \diamond Benutzer

FK2=Bestellung.AutoID \diamond Spediteur

Zahlung(BestellNr, ZahlungsID, Betrag, Zahlungsmethode)

SK={BestellNr,ZahlungsID}

PK=(BestellNr, ZahlungsID)

FK=Zahlung.BestellNr \diamond Bestellung

Artikel(ArtikelNr, *BestellNr*, *ShopID*, Name, Bezeichnung, Preis, Groesse, Gewicht)

SK={ArtikelNr}

PK=ArtikelNr

FK1=Artikel.BestellNr \diamond Bestellung

FK2=Artikel.ShopID \diamond Verkaeufer

waehlt(LoginID, *ArtikelNr*)

SK={LoginID, ArtikelNr}

PK=(LoginID, ArtikelNr)

FK1=waehlt.LoginID \diamond Benutzer

FK2=waehlt.ArtikelNr \diamond Artikel

leitet(JobID1, JobID2)

SK={JobID1,JobID2}

PK=(JobID1,JobID2)

FK1=leitet.JobID1 \diamond Angestellter

FK2=leitet.JobID2 \diamond Angestellter

Implementierung

Ich habe den Datenbank in SQL Developer erstellt. Basierend auf die ER-Diagramm habe ich alle Tabellen, deren Attributen und die Beziehungen zwischen ihnen zusammengestellt. Die Datenbank wird mit Daten mithilfe von Java befüllt. Für die Generierung von Daten habe ich CSV Files verwendet. Die Interaktion vom Frontend mit den Datenbank zu ermöglichen, habe ich PHP benutzt.