

Fitnesszentrum

Meilenstein 1: Anforderungsanalyse & Konzeptioneller Entwurf

Die Kunden besuchen ein Fitnesszentrum. Jeder Kunde besitzt Kundennummer, Adresse und Telefonnummer. Das Fitnesszentrum wird mit der Abteilungsnummer, dem Namen, der PLZ, dem Ort bezeichnet. Das Fitnesszentrum beschäftigt viele MitarbeiterInnen, die eine Mitarbeiternummer, ein Geburtsdatum und eine Adresse haben und sind in verschiedenen Positionen angestellt, etwa als ReceptionistIn oder Personal Trainer. Ein Mitarbeiter leitet andere Mitarbeiter. ReceptionistInnen haben eine Sozialversicherungsnummer und eine E-mail. Die Personal Trainer haben ein TrainerID und einen Vor- und Nachnamen. Ein/Eine ReceptionistIn verkauft verschiedene Abonnementstypen an Kunden. Der Abonnementstyp wird durch die Abonummer, die Kündigungsfrist und die Geschäftsbedingungen beschrieben. Ein Personal Trainer trainiert Kunden. Das Fitnesszentrum hat Umkleidekabinen, die sich durch eine Zimmernummer, Damen oder Herren unterscheiden. Ein/Eine Kunde/Kundin nutzt eine/oder mehrere Umkleidekabinen. Eine Umkleidekabine wird von mehreren Kunden benutzt.

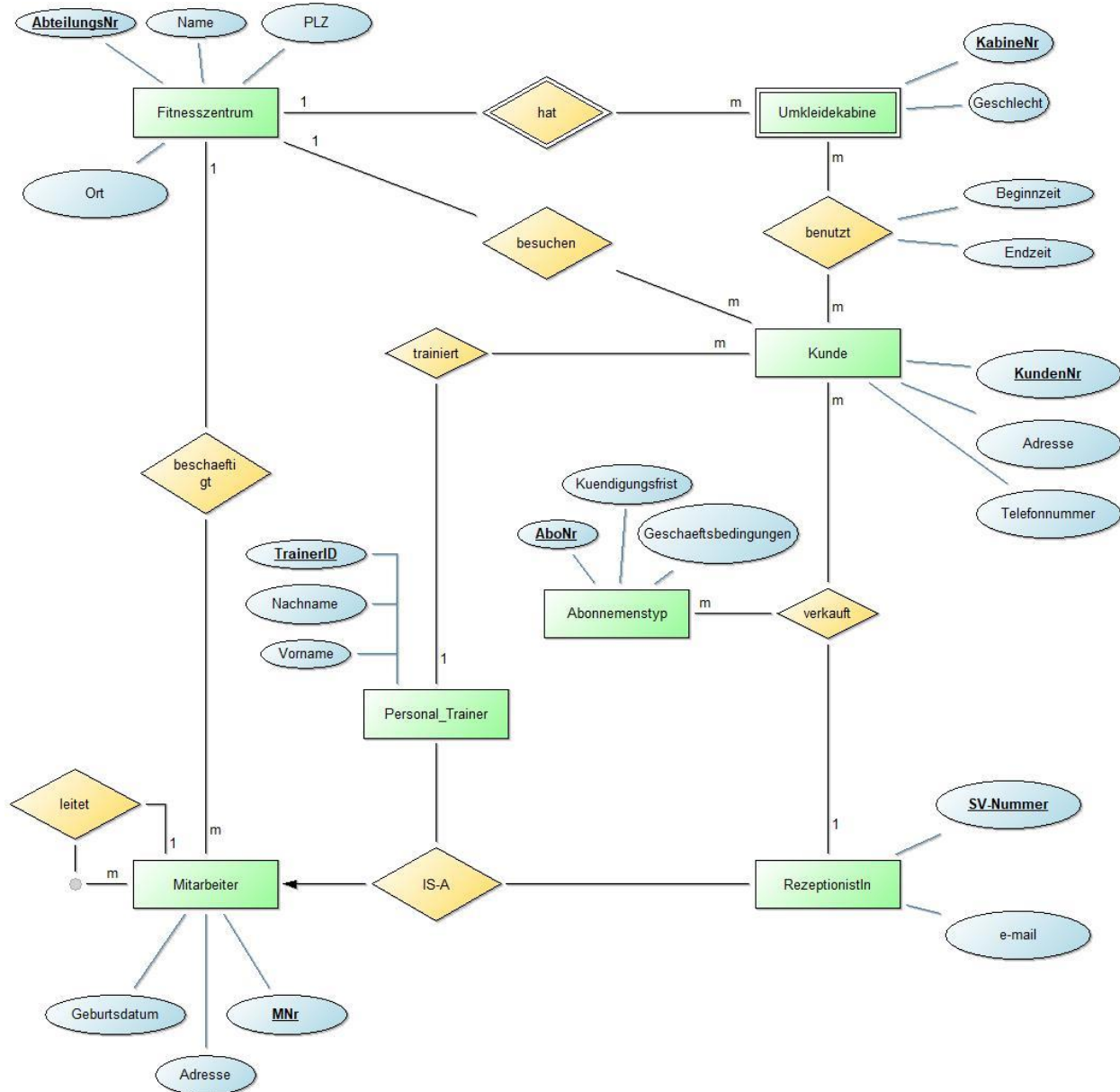


Abbildung 1: Entity Relationship Diagramm

Meilenstein 2: Logischer Entwurf

Fitnesszentrum (AbteilungsNr, Name, PLZ, Ort)
SK: {AbteilungsNr}

Kunde (KundenNr, Adresse, Telefonnummer, *AbteilungsNr*, *TrainerID*)
SK: {KundenNr}
FK: *AbteilungsNr* <> Fitnesszentrum.*AbteilungsNr*
FK: *TrainerID* <> Personal_Trainer.*TrainerID*

Umkleidekabine (AbteilungsNr, KabineNr, Geschlecht)
SK: {AbteilungsNr, KabineNr}
FK: *AbteilungsNr* <> Fitnesszentrum.*AbteilungsNr*

Benutzung (AbteilungsNr, KundenNr, KabineNr, Beginnzeit, Endzeit)
SK: {KundenNr, KabineNr}
FK: *AbteilungsNr* <> Fitnesszentrum.*AbteilungsNr*
FK: *KundenNr* <> Kunde.*KundenNr*
FK: *KabineNr* <> Umkleidekabine.*KabineNr*

Mitarbeiter (MNr, Adresse, Geburtsdatum, *Leiter-MNr*, *AbteilungsNr*)
SK: {MNr}.
FK: *Leiter-MNr* <> Mitarbeiter.*MNr*
FK: *AbteilungsNr* <> Fitnesszentrum.*AbteilungsNr*

Abonnementstyp (AboNr, Kuendigungsfrist, Geschaeftsbedingungen)
SK: {AboNr}

RezeptionistIn (MNr, SV-Nummer, e-mail)
SK: {MNr}
FK: *MNr* <> Mitarbeiter.*MNr*

Personal_Trainer (MNr, *TrainerID*, Nachname, Vorname)
SK: {MNr}.
FK: *MNr* <> Mitarbeiter.*MNr*

Verkauf (AboNr, KundenNr, *SV-Nummer*)
SK: {AboNr, KundenNr}
FK: *AboNr* <> Abonnement.*AboNr*
FK: *KundenNr* <> Kunde.*KundenNr*
FK: *SV-Nummer* <> RezeptionistIn.*SV-Nummer*

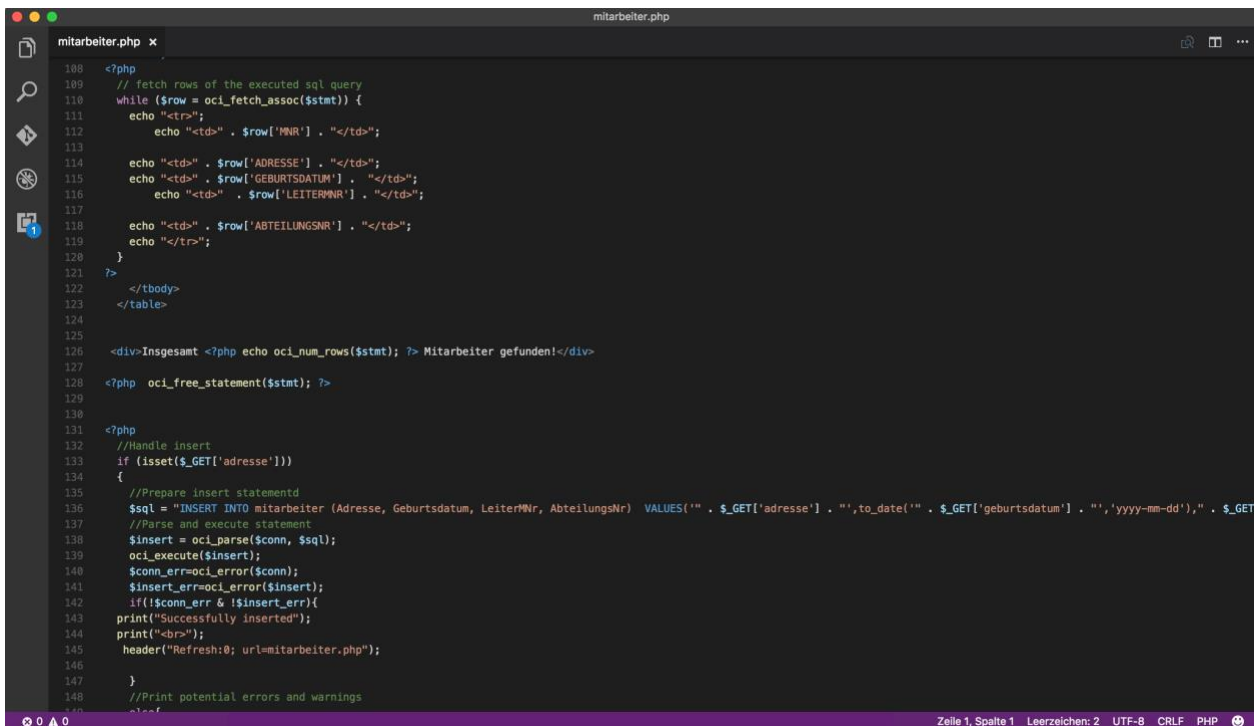
Meilenstein 4: Implementierung

Java Implementierung

Der Hauptteil von Java Code besteht aus for-Schleifen, die die Datensätze in einer Datenbank einfügen. Die Daten unterscheiden sich durch Zahlen der for-Schleife (z.B Adresse_1, Adresse_2 usw.). Die Datums (yyyy-mm-dd), Telefonnummer, E-mails usw. werden mit Hilfe von math.Random gemacht.

PHP Implementierung

Alle Tabellen werden als eigene Seite gemacht. Bei jeder Tabelle darf man neue Daten eintragen und nach dem bestimmten Attribut suchen. Die Anzahl der Daten in einer Datenbank wird unter diesen Daten angezeigt. Die Daten darf man oben einfügen. Das erleichtert einem User die Dateneingabe, weil ein User nicht nach unten scrollen muss, wenn er die neuen Daten einfügen will.



```
108 <?php
109 // fetch rows of the executed sql query
110 while ($row = oci_fetch_assoc($stmt)) {
111     echo "<tr>";
112     echo "<td>" . $row['MNR'] . "</td>";
113
114     echo "<td>" . $row['ADRESSE'] . "</td>";
115     echo "<td>" . $row['GEBURTSDATUM'] . "</td>";
116     echo "<td>" . $row['LEITERMNR'] . "</td>";
117
118     echo "<td>" . $row['ABTEILUNGSNR'] . "</td>";
119     echo "</tr>";
120 }
121 ?>
122 </tbody>
123 </table>
124
125 <div>Insgesamt <?php echo oci_num_rows($stmt); ?> Mitarbeiter gefunden!</div>
126
127 <?php oci_free_statement($stmt); ?>
128
129
130
131 <?php
132 //Handle insert
133 if (isset($_GET['adresse']))
134 {
135     //Prepare insert statement
136     $sql = "INSERT INTO mitarbeiter (Adresse, Geburtsdatum, LeiterMnr, AbteilungsNr) VALUES('" . $_GET['adresse'] . "',to_date('" . $_GET['geburtsdatum'] . "','yyyy-mm-dd'),' " . $_GET
137 //Parse and execute statement
138 $insert = oci_parse($conn, $sql);
139 oci_execute($insert);
140 $conn_err=oci_error($conn);
141 $insert_err=oci_error($insert);
142 if(!$conn_err & !$insert_err){
143     print("Successfully inserted");
144     print("<br>");
145     header("Refresh:0; url=mitarbeiter.php");
146 }
147 //Print potential errors and warnings
148 }
```

Abbildung 2: PHP Code

wwwlab.cs.univie.ac.at

DBS WS2017 - Übungsprojekt

wwwlab.cs.univie.ac.at/~jelbuldin87/dbs/kunde.php

[Alle Fitnesszentrum](#) --- Suche nach Adresse: [Los!](#) [Alle Mitarbeiter](#) --- [Alle Trainers](#) --- [Alle Kabinen](#) --- [Alle Rezeptionistinnen](#) --- [Alle Benutzungen](#) --- [Alle Abos](#) --- [Alle Verkäufe](#) ---

Neue Kunde einfügen:

kundenr	adresse	telefonnummer	abteilungsnr	trainerid
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Insert!

kundenr	adresse	telefonnummer	abteilungsnr	trainerid
1	Adresse1	10000001	1	1
2	Adresse2	10000002	1	2
3	Adresse3	10000003	1	3
4	Adresse4	10000004	1	4
5	Adresse5	10000005	1	5
6	Adresse6	10000006	1	6
7	Adresse7	10000007	1	7
8	Adresse8	10000008	1	8
9	Adresse9	10000009	1	9
10	Adresse10	10000010	1	10
11	Adresse11	10000011	1	11
12	Adresse12	10000012	1	12
13	Adresse13	10000013	1	13
14	Adresse14	10000014	1	14
15	Adresse15	10000015	1	15
16	Adresse16	10000016	1	16
17	Adresse17	10000017	1	17
18	Adresse18	10000018	1	18
19	Adresse19	10000019	1	19
20	Adresse20	10000020	1	20
21	Adresse21	10000021	1	21
22	Adresse22	10000022	1	22
23	Adresse23	10000023	1	23
24	Adresse24	10000024	1	24
25	Adresse25	10000025	1	25
26	Adresse26	10000026	1	26
27	Adresse27	10000027	1	27

Abbildung 3: Design