

## KOCHSCHULE

### **Meilenstein 1: Anforderungsanalyse & Konzeptioneller Entwurf**

Die Kursteilnehmer besuchen die Kochschule. Jeder Kursteilnehmer besitzt Kursteilnehmernummer, Vorname, Nachname, E-Mail-Adresse und Telefonnummer. Die Kochschule zeichnet sich durch aus, dass sie ein Name, eine Adresse(bestehend aus Straße, Postleitzahl, Ort) und eine Abteilungsnummer hat. Die Kochschule hat die Küchen und von jeder Küche sind die Nummer, das Fassungsvermögen und die Ausstattung bekannt.

Die Kochschule beschäftigt viele MitarbeiterInnen. Alle in der Datenbank verwalteten Mitarbeiter werden durch die Mitarbeiter-ID eindeutig identifiziert. Außerdem haben sie auch eine physische Adresse, ein Nachname, ein Vorname, ein Gehalt und ein Geburtsdatum. Ein Mitarbeiter leitet den anderen Mitarbeiter. Sie sind in verschiedenen Positionen angestellt, etwa als Manager oder Koch. Jeder Koch wird durch eine eindeutige Koch-ID identifiziert. Er hat ein Rang und eine Information über Ausbildung.

Jeder Manager besitzt Sozialversicherungsnummer, E-Mail-Adresse und Telefonnummer. Ein Manager organisiert die Kochkurse(für diese soll die Kursnummer, das Thema und der Preis gespeichert werden).

Die Kursteilnehmer nehmen an dem Kochkurs teil. In der Küche können beliebig viele Kochkurse zu dem angegebenen Datum und Zeitblock stattfinden. Ein Kochkurs kann in genau einer Küche stattfinden. Ein Koch kann beliebig viele Kurse führen. Jeder Kochkurs wird von einer oder mehreren Köche geleitet.

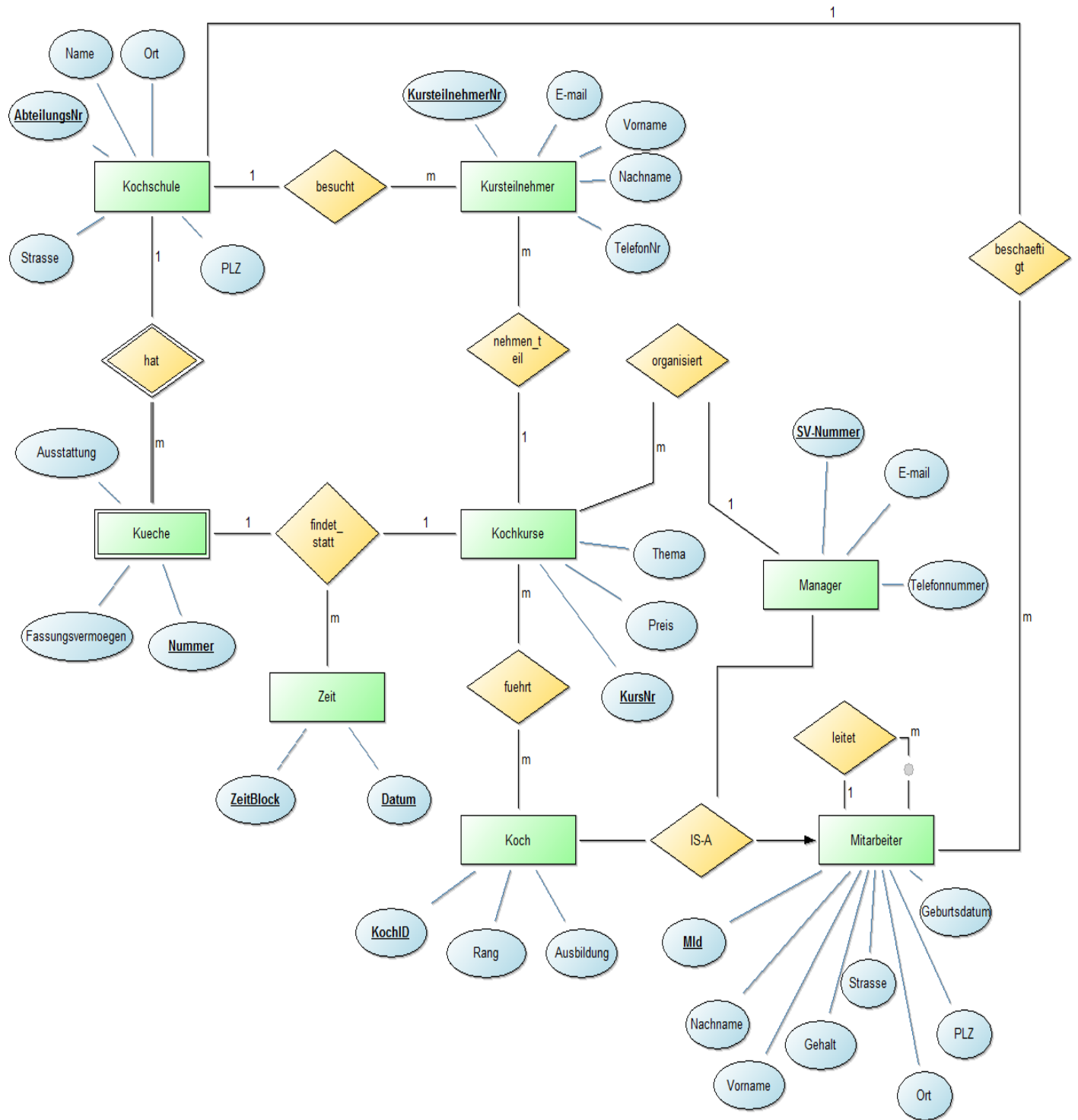


Abbildung 1: Entity Relationship Diagramm

## Meilenstein 2: Logischer Entwurf

**Kochschule** (AbteilungsNr, Name, Ort, PLZ, Strasse)

PK: {AbteilungsNr}

**Kueche** (AbteilungsNr, Nummer, Fassungsvermoegen, Ausstattung)

PK: { AbteilungsNr, Nummer }

FK: Kueche.AbcilungsNr<>Kochschule.AbcilungsNr

**Kursteilnehmer**(KursteilnehmerNr, Vorname, Nachname, E-Mail, TelefonNr, *AbteilungsNr*, *KursNr*)

PK: {KursteilnehmerNr}

FK: Kursteilnehmer.AbcilungsNr<>Kochschule.AbcilungsNr

FK: Kursteilnehmer.KursNr<>Kochkurse.KursNr

**Mitarbeiter**(Mld, Nachname, Vorname, Gehalt, Strasse, Ort, PLZ, Geburtsdatum, *Leiter-Mld*, *AbteilungsNr*)

PK: {Mld}

FK: Mitarbeiter.Leiter-Mld<>Mitarbeiter.Mld

FK: Mitarbeiter.AbcilungsNr<>Kochschule.AbcilungsNr

**Manager**(SV-Nummer, E-Mail, Telefonnummer, *Mld*)

PK: {SV-Nummer}

FK: Manager.Mld<> Mitarbeiter.Mld

**Koch**(KochID, Rang, Ausbildung, *Mld*)

PK: {KochID}

FK: Manager.Mld<> Mitarbeiter.Mld

**Kochkurse**(KursNr, Preis, Thema, *SV-Nummer*)

PK: {KursNr}

FK: Kochkurse.SV-Nummer<>Manager.SV-Nummer

**Zeit** (ZeitBlock, Datum)

PK: { ZeitBlock, Datum }

**Findet\_statt**(ZeitBlock, Datum, KursNr, *Nummer*, *AbteilungsNr*)

PK: { ZeitBlock, Datum, KursNr }

FK: Findet\_statt.ZeitBlock<>Zeit.ZeitBlock

FK: Findet\_statt.Datum<>Zeit.Datum

FK: Findet\_statt.KursNr<>Kochkurse.KursNr

FK: Findet\_statt.Nummer<>Kueche.Nummer

FK: Findet\_statt.AbcilungsNr<>Kueche.AbcilungsNr

**Fuehrt**(KochID, KursNr)

PK: {Koch, KursNr}

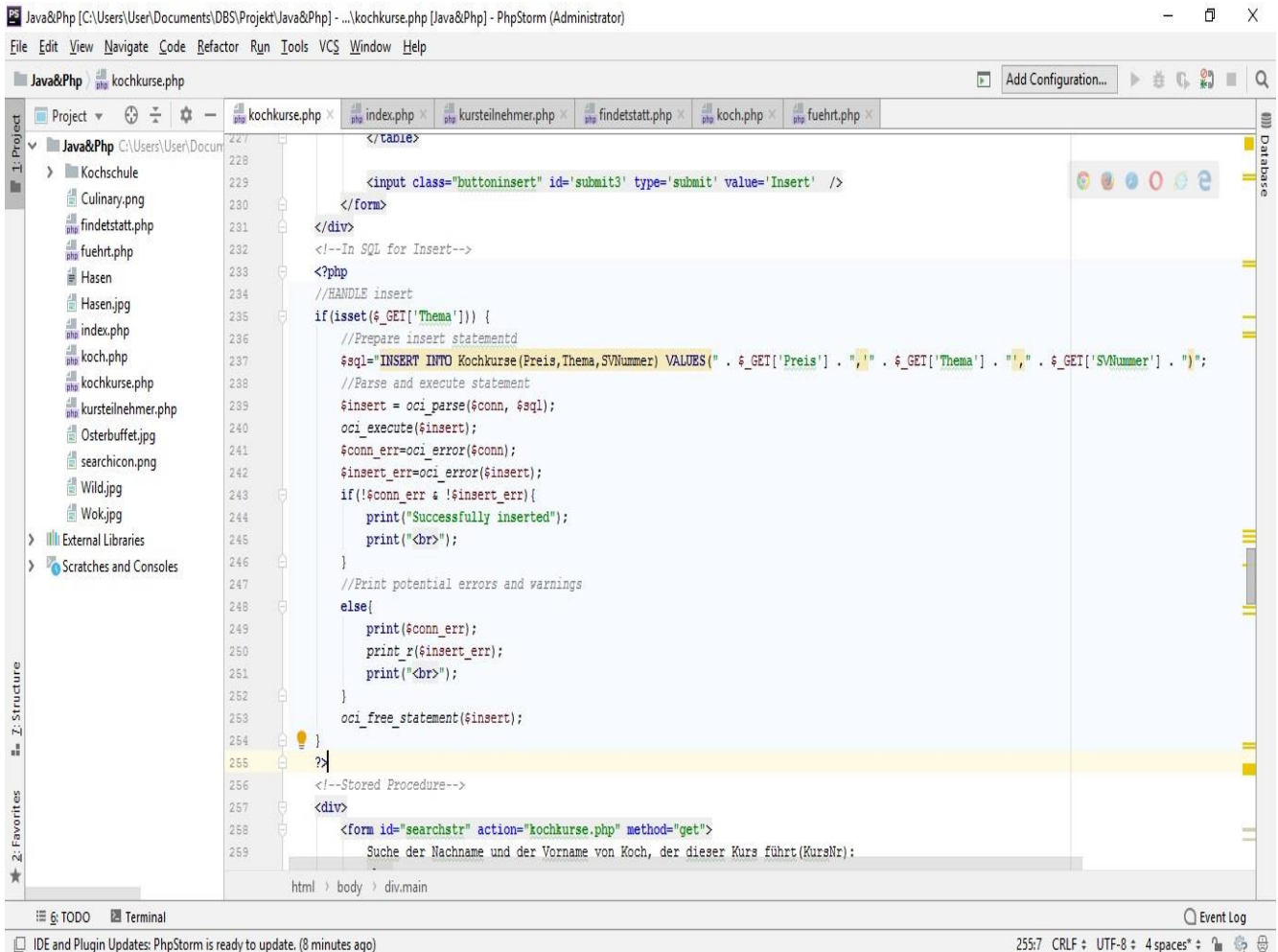
FK: Fuehrt.KochID<>Koch.KochID

FK: Fuehrt.KursNr<>Kochkurse.KursNr



## PHP Implementierung

Die meisten Tabellen werden als eigene Seite gemacht. Bei jeder angezeigter Tabelle kann man neue Datensätze eintragen und nach dem bestimmten Attribut suchen. Die Anzahl der Datensätze in der Tabelle wird entsprechend angezeigt. In Datei „kochkurse.php“ und „findetstatt.php“ wird Stored Procedure aufgerufen. Auf linke Seite steht das vertikale Menü für Navigation. In „index.php“ befinden sich die Bildern und Linke auf die Seite mit diesen Kochkursen.




```
227 </table>
228
229 <input class="buttoninsert" id='submit3' type='submit' value='Insert' />
230 </form>
231 </div>
232 <!--In SQL for Insert-->
233 <?php
234 //HANDLE insert
235 if(isset($_GET['Thema'])) {
236     //Prepare insert statementd
237     $sql="INSERT INTO Kochkurse(Preis,Thema,SVNummer) VALUES(" . $_GET['Preis'] . "," . $_GET['Thema'] . "," . $_GET['SVNummer'] . ")";
238     //Parse and execute statement
239     $insert = oci_parse($conn, $sql);
240     oci_execute($insert);
241     $conn_err=oci_error($conn);
242     $insert_err=oci_error($insert);
243     if(!$conn_err & !$insert_err){
244         print("Successfully inserted");
245         print("<br>");
246     }
247     //Print potential errors and warnings
248     else{
249         print($conn_err);
250         print_r($insert_err);
251         print("<br>");
252     }
253     oci_free_statement($insert);
254 }
255 ?>
256 <!--Stored Procedure-->
257 <div>
258 <form id="searchstr" action="kochkurse.php" method="get">
259     Suche der Nachname und der Vorname von Koch, der dieser Kurs führt (KursNr):
```

Abbildung 3: PHP Code

D8S WS2018 - Übungsprojekt x SQL - Structured Query Language x Die Kochschule x +

Nicht sicher | wwwlab.cs.univie.ac.at/~karynav98/dbs/kochkurse.php?search=Festlicher+Osterbrunch



Die Kochkurse

Neuer Kochkurs einfügen:

Preis	Thema	SVNummer
<input type="text"/>	<input type="text"/>	<input type="text"/>

Suche der Nachname und der Vorname von Koch, der dieser Kurs führt(KursNr):

und Kontakte von Manager, der dieser Kurs organisiert(SVNummer):

[Alle Kochkurse](#)

Suche nach Thema des Kochkurses:

Abbildung 4: Design