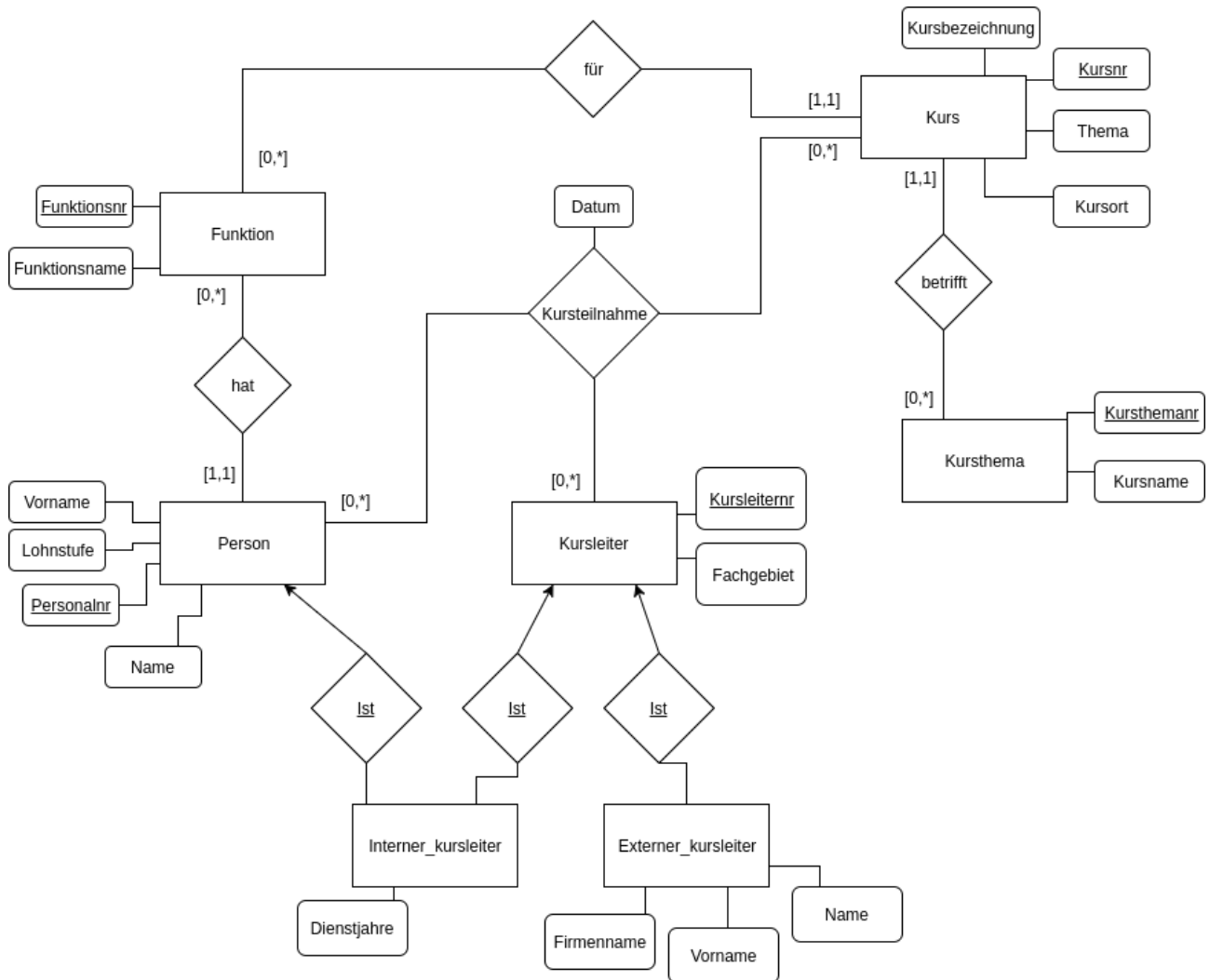


Trees Maximilian
Sinning Fabian

Aufgabe 1

a)



b) Interner und Externer Kursleiter besitzen beide die Kursleiternummer, obwohl es sich um nicht überlappende Objekte handelt.

c)

Alle Objekte des ER-Diagramms werden in Tabellen umgeformt.

Person

<u>Personalnr</u>	Name	Vorname	Lohnstufe

hat

<u>Personalnr</u>	<u>Funktionsnr</u>

Funktion

<u>Funktionsnr</u>	Funktionsname

für

<u>Funktionsnr</u>	<u>Kursnr</u>

Kurs

<u>Kursnr</u>	Thema	Kursort	Kurzbezeichnung

betrifft

<u>Kursnr</u>	Kursthemanr

Kursthema

<u>Kursthemanr</u>	Kursname

Kursteilnahme

<u>Personalnr</u>	<u>Kursleiternr</u>	<u>Kursnr</u>	Datum

Kursleiter

<u>Kursleiternr</u>	Fachgebiet

Interner_Kursleiter

<u>Kursleiternr</u>	Dienstjahre

Externer_Kursleiter

<u>Kursleiternr</u>	Name	Vorname	Firmenname

Aufgabe 2

Wir erstellen mit dem Tool *Oracle SQL Developer* alle Tabellen mit SQL Befehlen in der Konsole.

Tabelle Person

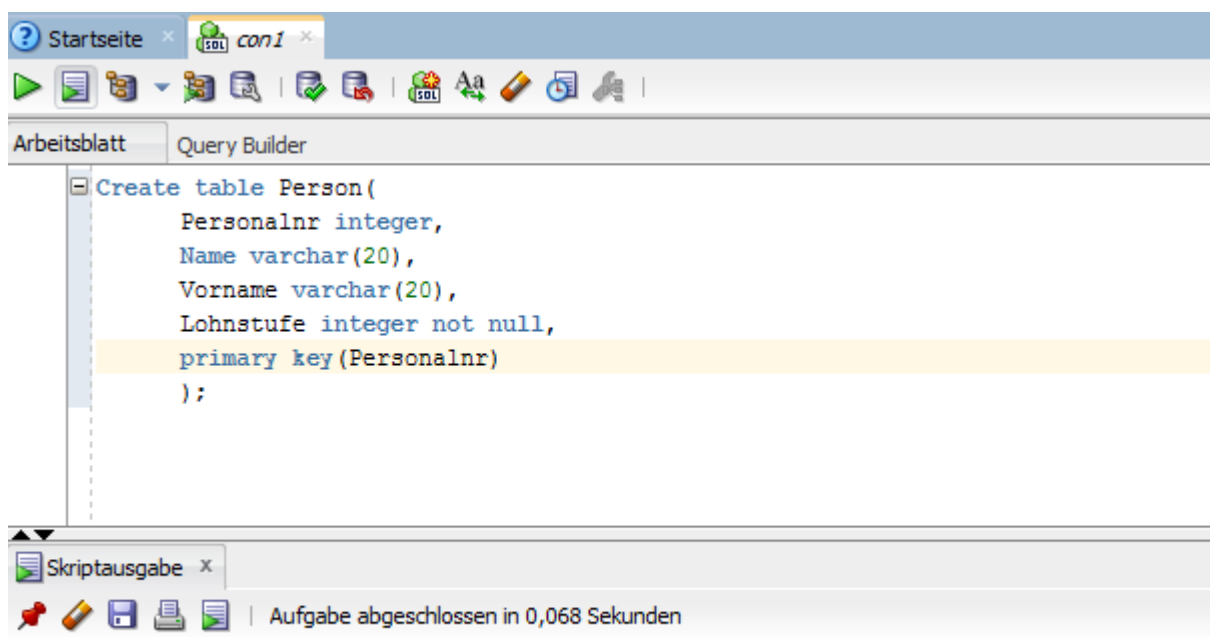


Table PERSON erstellt.

Tabelle Funktion

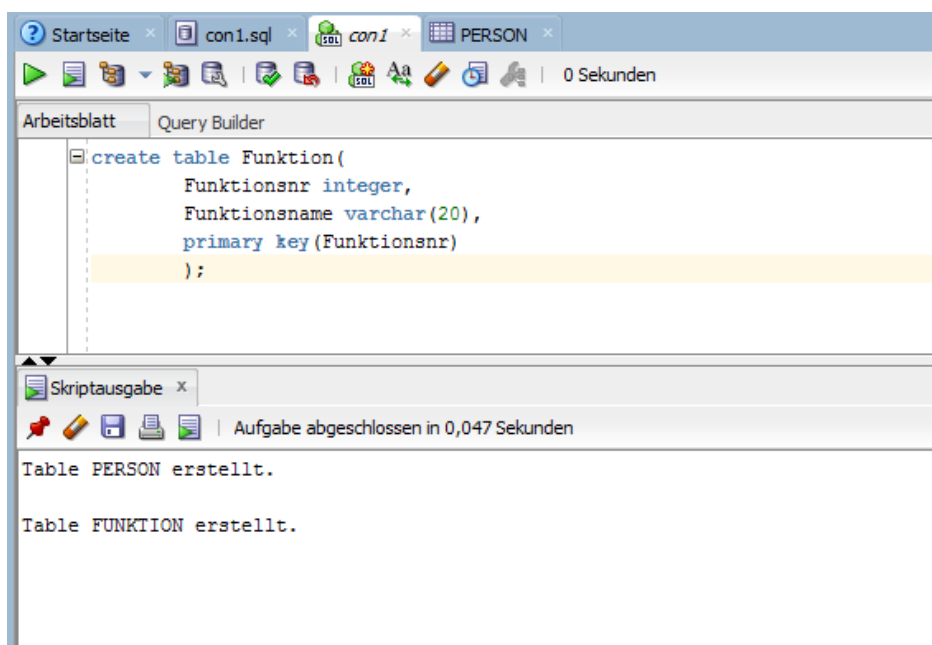


Table PERSON erstellt.

Table FUNKTION erstellt.

Tabelle hat

The screenshot shows a database management interface with a top toolbar containing icons for execution, saving, and other database operations. The main window is titled 'Arbeitsblatt' and 'Query Builder'. It displays the SQL command to create a table named 'hat' with the following structure:

```
create table hat(  
    Personalnr integer,  
    Funktionsnr integer,  
    foreign key (Personalnr) references Person(Personalnr),  
    foreign key (Funktionsnr) references Funktion(Funktionsnr)  
);
```

Below the query editor, a status bar indicates 'Aufgabe abgeschlossen in 0,14 Sekunden'. The output window, titled 'Skriptausgabe', shows the message 'Table HAT erstellt.'

Tabelle Kurs

The screenshot shows the same database management interface, but now the 'HAT' table is selected in the top toolbar. The 'Query Builder' window displays the SQL command to create a table named 'Kurs' with the following structure:

```
create table Kurs(  
    Kursnr integer,  
    Thema varchar(30),  
    Kursort varchar(20),  
    Kursbezeichnung varchar(40),  
    primary key (Kursnr)  
);
```

The status bar again shows 'Aufgabe abgeschlossen in 0,063 Sekunden'. The output window, titled 'Skriptausgabe', shows the message 'Table KURS erstellt.'

Tabelle für

The screenshot shows a database management interface with a top toolbar containing icons for Startseite, con1.sql, con1, and HAT. Below the toolbar is a tabbed interface with 'Arbeitsblatt' and 'Query Builder'. The 'Query Builder' tab displays the following SQL code:

```
create table für(  
    Funktionsnr integer,  
    Kursnr integer,  
    foreign key (Funktionsnr) references Funktion(Funktionsnr),  
    foreign key (Kursnr) references Kurs(Kursnr)  
);
```

Below the query editor is a 'Skriptausgabe' (Script Output) window. It shows the status 'Aufgabe abgeschlossen in 0,078 Sekunden' and the message 'Table FÜR erstellt.' (Table FÜR created).

Tabelle Kursthema

The screenshot shows the same database management interface. The 'Query Builder' tab displays the following SQL code:

```
create table Kursthema(  
    Kursthemanr integer,  
    Kursname varchar(20),  
    primary key (Kursthemanr)  
);
```

Below the query editor is the 'Skriptausgabe' window, which shows the status 'Aufgabe abgeschlossen in 0,125 Sekunden' and the message 'Table KURSTHEMA erstellt.' (Table KURSTHEMA created).

Tabelle betrifft

The screenshot shows a database management interface with a top toolbar containing icons for Startseite, con1.sql, con1, and HAT. Below the toolbar is a tabbed interface with 'Arbeitsblatt' and 'Query Builder'. The 'Query Builder' tab is active, displaying a SQL script to create a table named 'betrifft'. The script is as follows:

```
create table betrifft(  
    Kursnr integer,  
    Kursthemanr integer,  
    foreign key (Kursnr) references Kurs (Kursnr),  
    foreign key (Kursthemanr) references Kursthema (Kursthemanr)  
);
```

Below the script editor is a 'Skriptausgabe' (Script Output) window. It shows the message 'Aufgabe abgeschlossen in 0,047 Sekunden' and 'Table BETRIFFT erstellt.' (Table BETRIFFT created).

Tabelle Kursleiter

The screenshot shows the same database management interface as the previous one. The 'Query Builder' tab is active, displaying a SQL script to create a table named 'Kursleiter'. The script is as follows:

```
create table Kursleiter(  
    Kursleiternr integer,  
    Fachgebiet varchar(30),  
    primary key (Kursleiternr)  
);
```

Below the script editor is a 'Skriptausgabe' (Script Output) window. It shows the message 'Aufgabe abgeschlossen in 0,078 Sekunden' and 'Table KURSLEITER erstellt.' (Table KURSLEITER created).

Tabelle Interner_Kursleiter

The screenshot shows the SQL Developer interface. The top toolbar includes icons for Startseite, con1.sql, con1, and HAT. Below the toolbar, the 'Arbeitsblatt' (Worksheet) tab is active, displaying the 'Query Builder' with the following SQL code:

```
create table Interner_Kursleiter(  
    Kursleiternr integer,  
    Dienstjahre integer,  
    primary key(Kursleiternr)  
);
```

The bottom pane, labeled 'Skriptaussgabe' (Script Output), shows the execution result: 'Aufgabe abgeschlossen in 0,125 Sekunden' and 'Table INTERNER_KURSLEITER erstellt.' (Table INTERNER_KURSLEITER created).

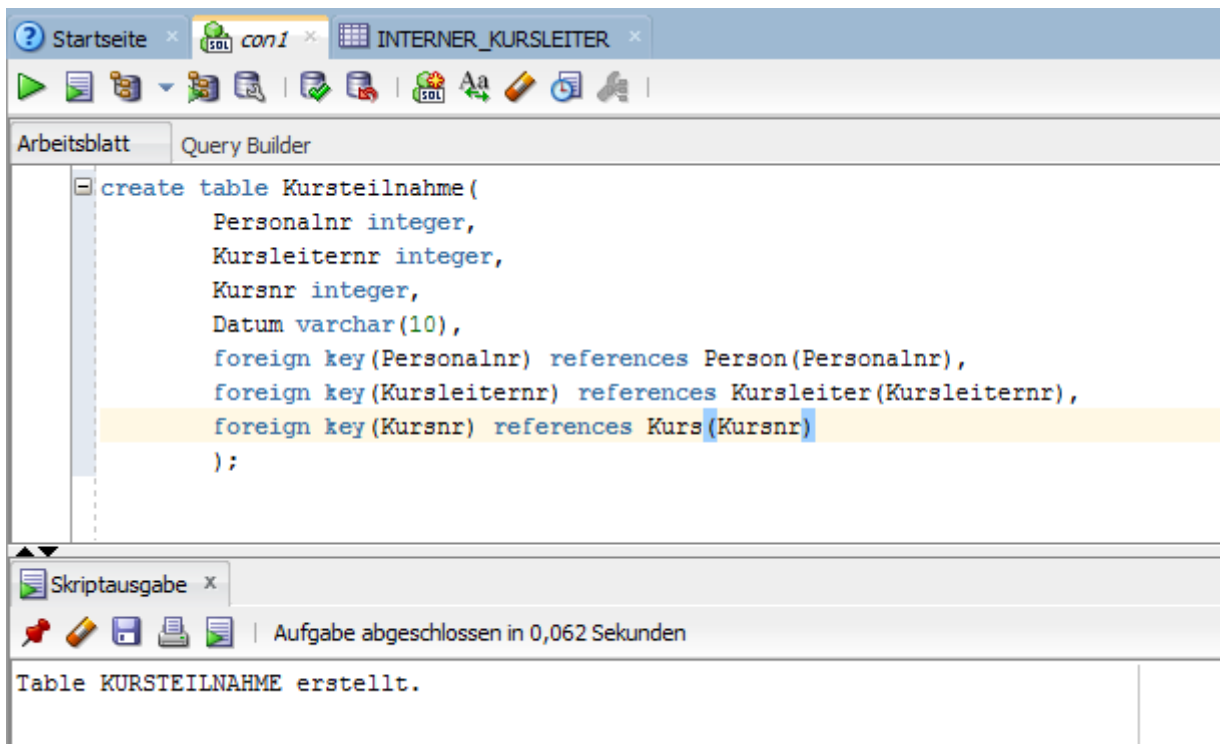
Tabelle Externer_Kursleiter

The screenshot shows the SQL Developer interface. The top toolbar includes icons for Startseite, con1, and INTERNER_KURSLEITER. Below the toolbar, the 'Arbeitsblatt' (Worksheet) tab is active, displaying the 'Query Builder' with the following SQL code:

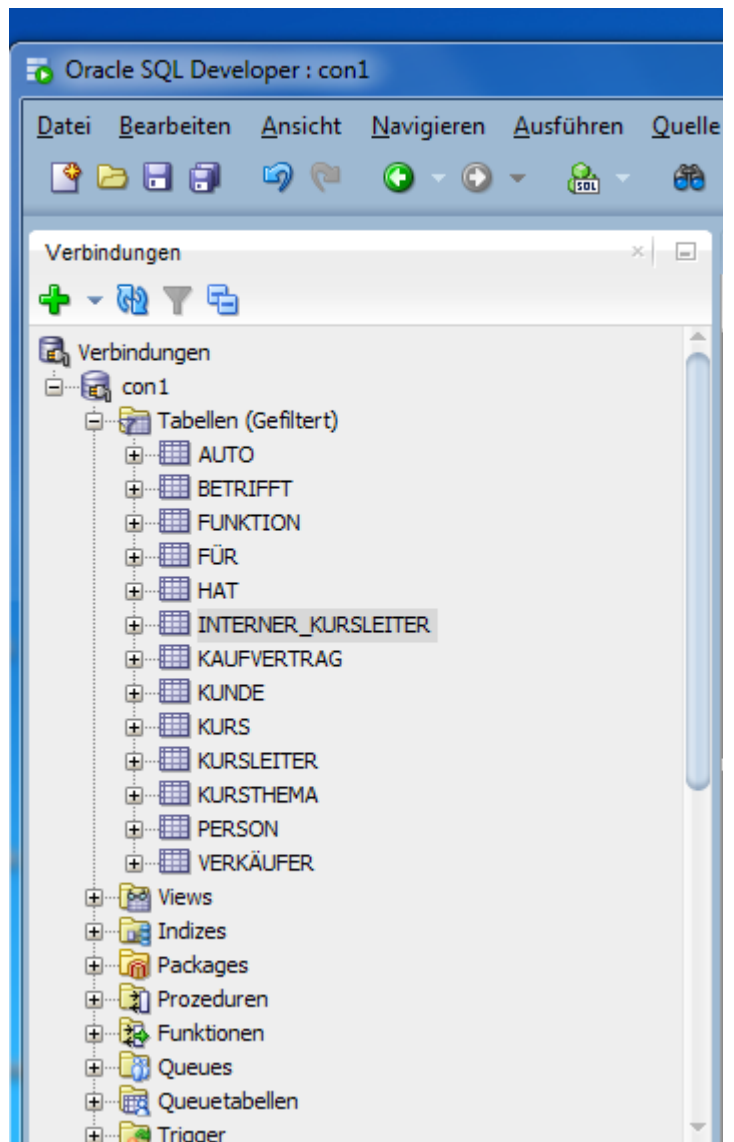
```
create table Externer_Kursleiter(  
    Kursleiternr integer,  
    Name varchar(20),  
    Vorname varchar(20),  
    Firmenname varchar(30),  
    primary key(Kursleiternr)  
);
```

The bottom pane, labeled 'Skriptaussgabe' (Script Output), shows the execution result: 'Aufgabe abgeschlossen in 0,047 Sekunden' and 'Table EXTERNER_KURSLEITER erstellt.' (Table EXTERNER_KURSLEITER created).

Tabelle Kursteilnahme



Einen Überblick über alle angelegten Tabellen, sowie alle Tabellen aus Blatt 3 Aufgabe 4 zeigt folgender Screenshot.



Aufgabe 3

Folgende Datensätze wollen wir als Beispiel in die Tabellen eintragen:
Alle Daten geben wir der Reihe nach mit dem SQL-Befehl

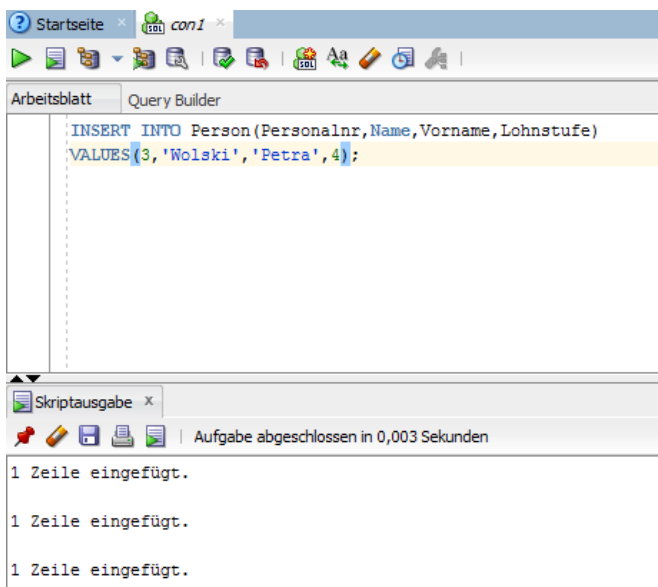
```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

in der Konsle ein.

Tabelle Person

Personalnr Name Vorname Lohnstufe

1	Hins	Albert	2
2	Kala	Simon	2
3	Wolski	Petra	4



The screenshot shows the 'PERSON' table data view. The table has four columns: PERSONALNR, NAME, VORNAME, and LOHNSTUFE. The data is as follows:

PERSONALNR	NAME	VORNAME	LOHNSTUFE
1	Hins	Albert	2
2	Kala	Simon	2
3	Wolski	Petra	4

oben Tabelle *Person* mit allen Daten
links Insert Befehl

Tabelle Funktion

Funktionsnr Funktionsname

1	Qualitätscheck
2	Web-Entwicklung
3	Architektur

The screenshot shows the 'FUNKTION' table data view. The table has two columns: FUNKTIONSNR and FUNKTIONSNAMEN. The data is as follows:

FUNKTIONSNR	FUNKTIONSNAMEN
1	Qualitätscheck
2	Web-Entwicklung
3	Architektur

oben Tabell *Funktion* mit allen Daten
rechts Insert Befehl

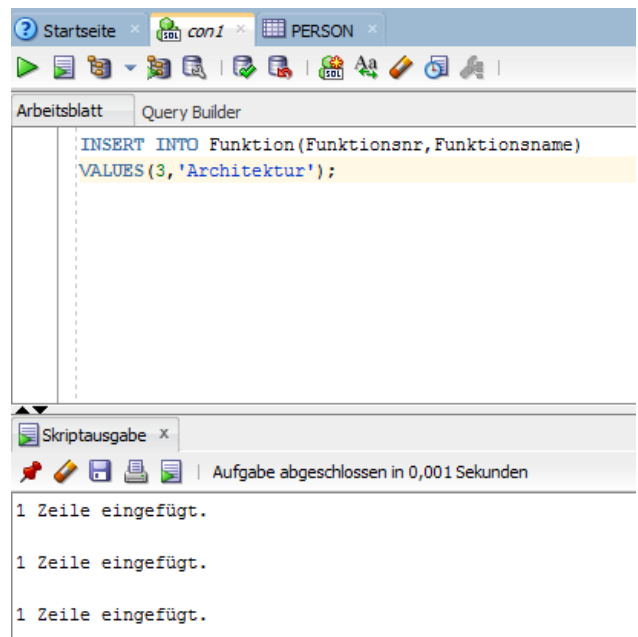
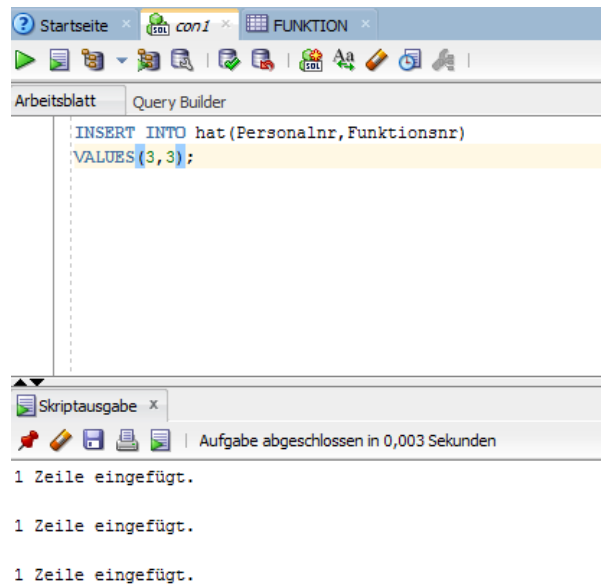


Tabelle hat

Personalnr Funktionsnr

1	1
2	2
3	3

PERSONALNR	FUNKTIONSNR
1	3
2	1
3	2

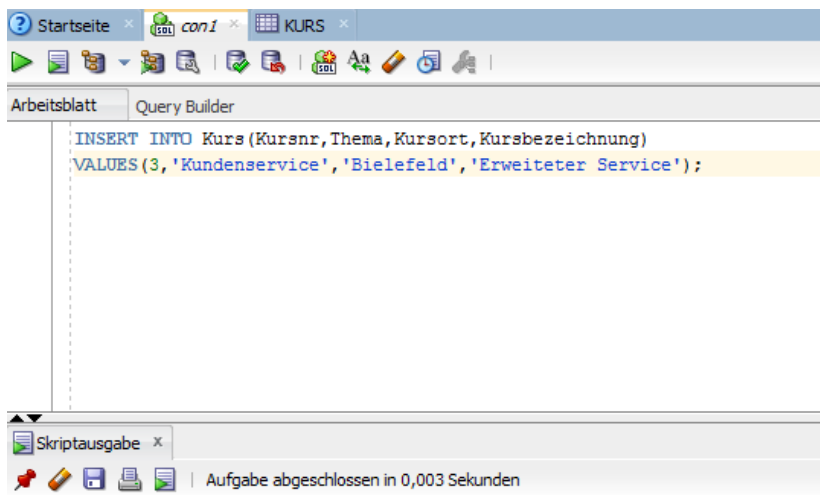


links Tabelle **hat** mit allen Daten

rechts Insert Befehl

Tabelle Kurs

Kursnr	Thema	Korsort	Kursbezeichnung
1	Design	München	Neues Design
2	Benutzerfreundliche Designs	Berlin	für Einsteiger
3	Kundenservice	Bielefeld	Erweiterter Service



Insert Befehl

KURSNR	THEMA	KURSORT	KURSBEZEICHNUNG
1	1 Design	München	Neus Design
2	2 Benutzerfreundliche Designs	Berlin	für Einsteiger
3	3 Kundenservice	Bielefeld	Erweiterter Service

Tabelle **Kurs** mit allen Daten

Tabelle für
Funktionsnr Kursnr

1	1
2	2

FÜR		
FUNKTIONSNR	KURSNR	
1	1	1
2	2	2

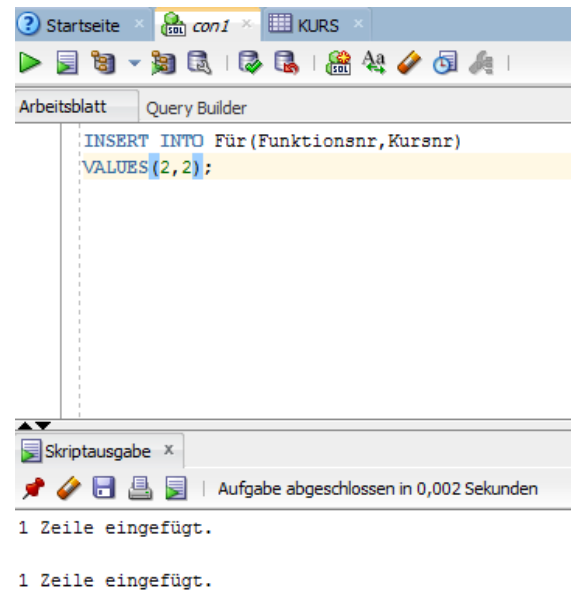


Tabelle **für** mit allen Daten

Insert Befehl

Tabelle Kursthema
Kursthemamn Kursname

1	Design
2	SQL
3	Java Beginning

KURSTHEMA	
KURSTHEMANR	KURSNAME
1	1 Design
2	2 SQL
3	3 Java Beginning

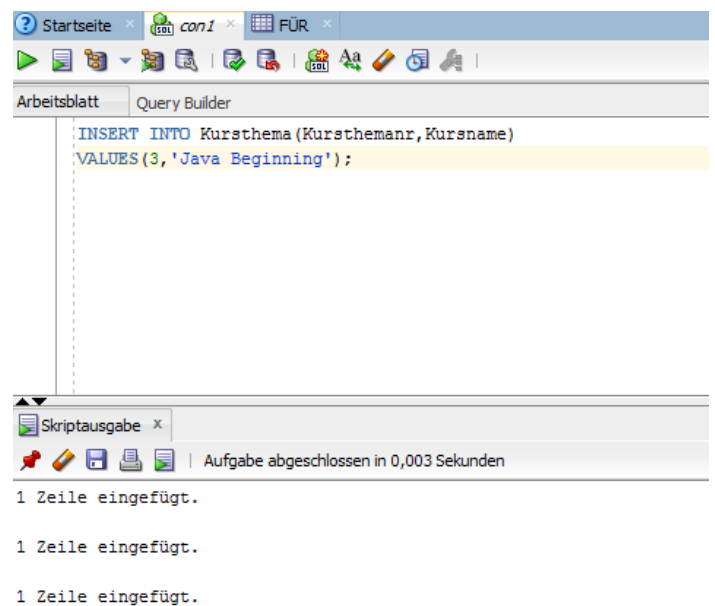


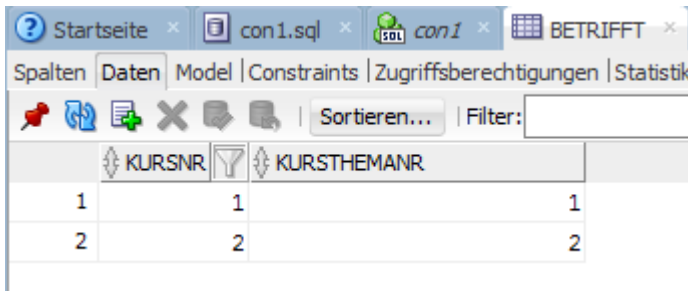
Tabelle **Kursthema** mit allen Daten

Insert Befehl

Tabelle *betrifft*

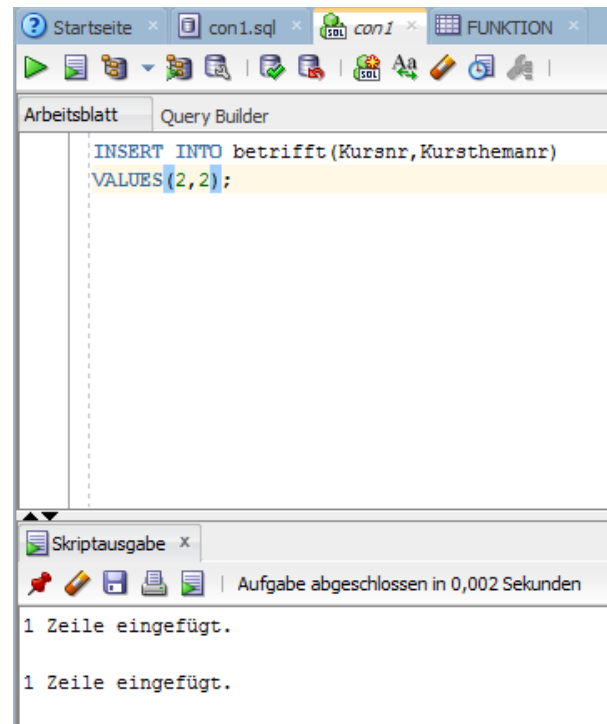
Kursnr **Kursthemanr**

1	1
2	2



KURSNR	KURSTHEMANR
1	1
2	2

Tabelle ***betrifft*** mit allen Daten

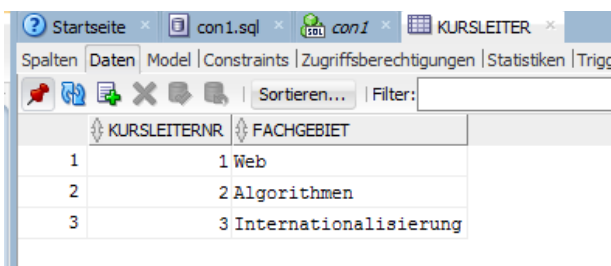


Insert Befehl

Tabelle *Kursleiter*

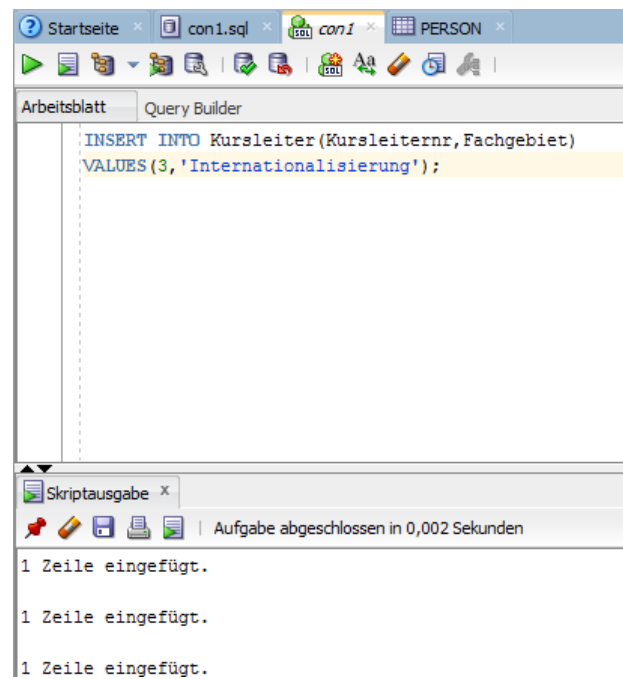
Kursleiternr **Fachgebiet**

1	Web
2	Algorithmen
3	Internationalisierung



KURSLEITERNR	FACHGEBIET
1	1 Web
2	2 Algorithmen
3	3 Internationalisierung

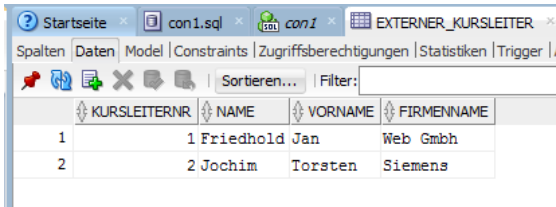
Tabelle ***Kursleiter*** mit allen Daten



Insert Befehl

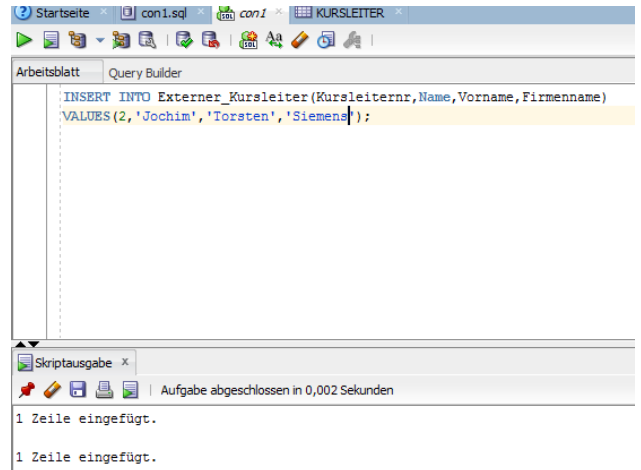
Tabelle Externer_Kursleiter

<u>Kursleiternr</u>	<u>Name</u>	<u>Vorname</u>	<u>Firmenname</u>
1	Friedhold	Jan	Web Gmbh
2	Jochim	Torsten	Siemens



KURSLEITERNR	NAME	VORNAME	FIRMENNAME
1	Friedhold	Jan	Web Gmbh
2	Jochim	Torsten	Siemens

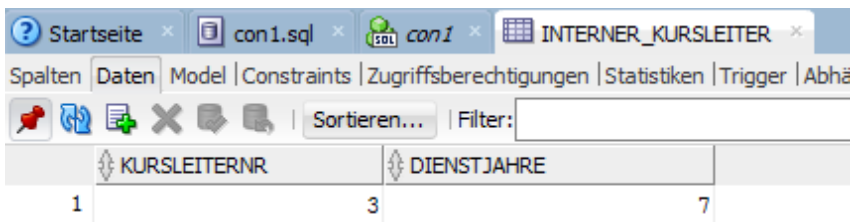
Tabelle *Externer_Kursleiter* mit allen Daten



Insert Befehl

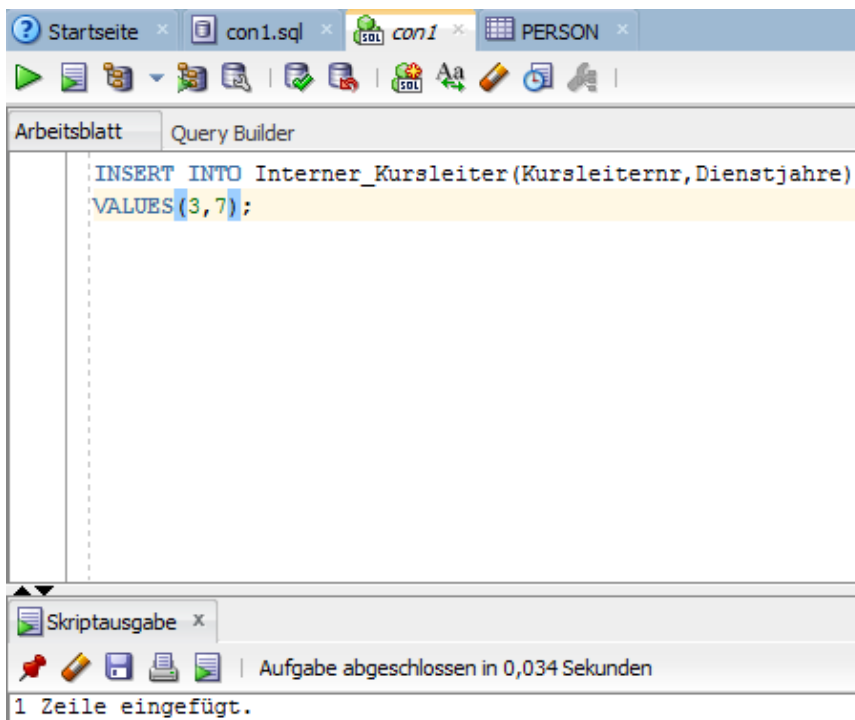
Tabelle Interner_Kursleiter

<u>Kursleiternr</u>	<u>Dienstjahre</u>
3	7



KURSLEITERNR	DIENTJAHRE
3	7

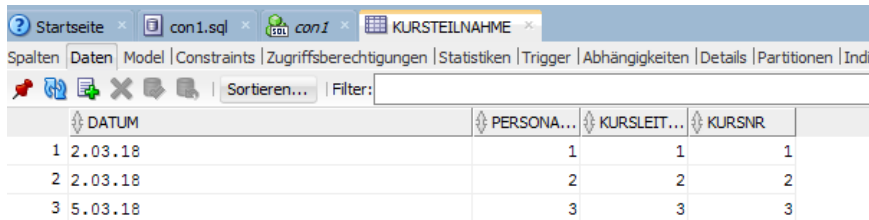
Tabelle *Interner_Kursleiter* mit allen Daten



Insert Befehl

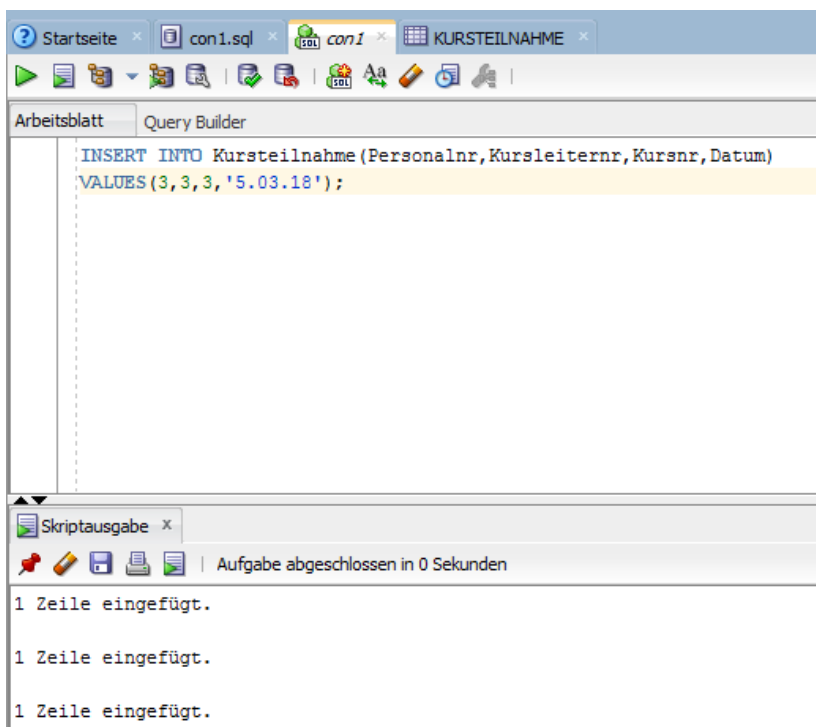
Tabelle Kursteilnahme

Datum	Personalnr	Kursleiternr	Kursnr
2.03.18	1	1	1
2.03.18	2	2	2
5.03.18	3	3	3



DATUM	PERSONA...	KURSLEIT...	KURSNR
1 2.03.18	1	1	1
2 2.03.18	2	2	2
3 5.03.18	3	3	3

Tabelle **Kursteilnahme** mit allen Daten



Arbeitsblatt Query Builder

```
INSERT INTO Kursteilnahme (Personalnr, Kursleiternr, Kursnr, Datum)
VALUES (3, 3, 3, '5.03.18');
```

Skriptausgabe x

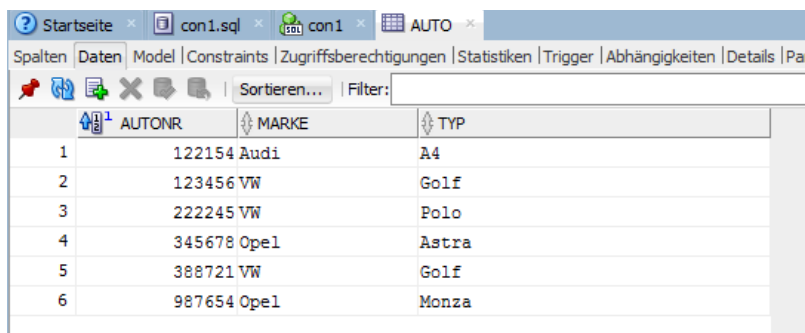
Aufgabe abgeschlossen in 0 Sekunden

1 Zeile eingefügt.
1 Zeile eingefügt.
1 Zeile eingefügt.

Insert Befehl

Aufgabe 3b

Wir verfahren für jede Tabellen mit dem Insert Befehl wie bei den Tabellen aus Aufgabe 3a und geben daher nur Screenshots von den fertigen Tabellen (Auto, Kaufvertrag und Verkäufer) an.



AUTONR	MARKE	TYP
1	122154 Audi	A4
2	123456 VW	Golf
3	222245 VW	Polo
4	345678 Opel	Astra
5	388721 VW	Golf
6	987654 Opel	Monza

Spalten	Daten	Model	Constraints	Zugriffsberechtigungen	Statistiken	Trigger	Abhängigkeiten	Details	Partitionen	Indizes	SQL
Sortieren...	Filter:										
	VERTRAGSNR	KUNDENNR	AUTONR	DATUM	VERKÄUFERNAME						
1	1	1	123456	23.4.04	Schmid						
2	2	1	345678	7.8.04	Plüss						
3	3	2	388721	17.6.04	Frey						
4	4	3	22245	15.7.04	Schmid						
5	5	4	122154	13.11.04	Frey						

Spalten	Daten	Model	Constraints	Zugriffsberechtigungen	Statistiken	Trigger	Abhängigkeiten	Details	Partitionen	Indizes	SQL
Sortieren...	Filter:										
	VERKÄUFERNAME	VERTRAGSNR									
1	Schmid	(null)									
2	Plüss	(null)									
3	Frey	(null)									
4	Schenk	(null)									

Aufgabe 4

a)

Die Tabelle VW wird mit dem create Table Statement eingefügt.

```

CREATE TABLE VW(
  Autonr integer,
  Marke varchar(50),
  Typ varchar(50),
  primary key(Autonr)
);

```

Skriptausgabe x

Aufgabe abgeschlossen in 0,156 Sekunden

Table VW erstellt.

```

INSERT INTO VW (Autonr, Marke, Typ)
SELECT Autonr, Marke, Typ FROM Auto
WHERE Marke='VW';

```

Skriptausgabe x

Aufgabe abgeschlossen in 0,031 Sekunden

3 Zeilen eingefügt.

Mit dem Befehl fügen wir aus Tabelle Auto alle VW Autos mit Marke VW ein.

b)

The screenshot shows a database query editor window with the following components:

- Top Bar:** Contains tabs for 'Startseite', 'con1', and 'AUTO'. Below the tabs is a toolbar with various icons for query execution and editing.
- Arbeitsblatt (Worksheet) Tab:** Displays the SQL query:

```
UPDATE Auto
SET Autonr=987654
WHERE Autonr=445321;
```
- Skriptausgabe (Script Output) Tab:** Shows the execution result: 'Aufgabe abgeschlossen in 0,003 Sekunden' and '1 Zeile aktualisiert.'

The screenshot shows a database table view with the following components:

- Top Bar:** Contains tabs for 'Startseite', 'con1', and 'AUTO'. Below the tabs is a toolbar with various icons for table manipulation.
- Table View:** Displays a table with three columns: 'AUTONR', 'MARKE', and 'TYP'. The table contains six rows of data.

	AUTONR	MARKE	TYP
1	122154	Audi	A4
2	123456	VW	Golf
3	222245	VW	Polo
4	345678	Opel	Astra
5	388721	VW	Golf
6	987654	Opel	Monza