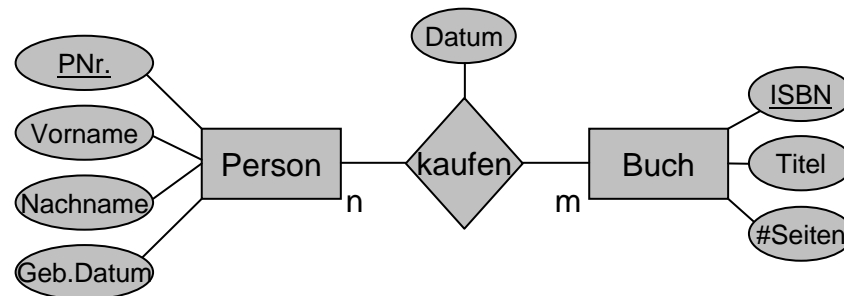
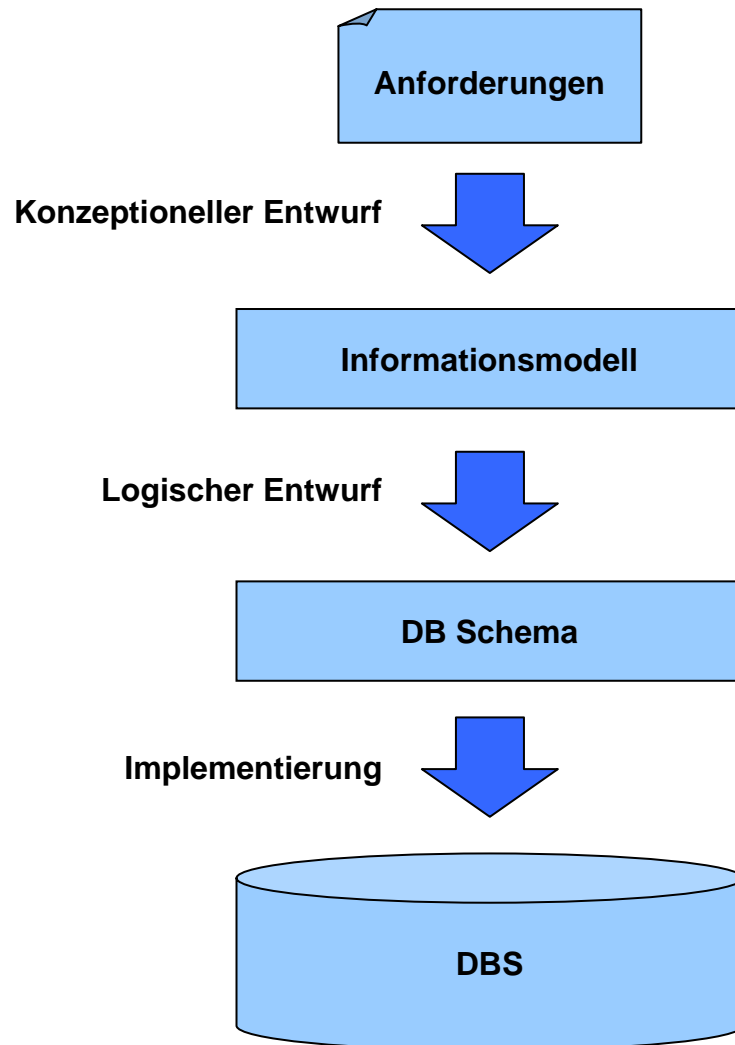

Grundlagen von Datenbanken

Aufgabenzettel 4

SQL-Anfragen



Überblick: DB-Entwurf und Modellierung



Person(PNR, Vorname, Nachname, GebDatum)
Buch(ISBN, Titel, Seiten)
Kauf(Person, Buch, Datum)

SQL DDL

```
CREATE TABLE Person (  
    PNR          int          PRIMARY KEY,  
    Vorname      varchar(50),  
    Nachname     varchar(50) NOT NULL,  
    GebDatum     date         NOT NULL  
);
```

SQL DDL: Basisrelationen

■ Syntax

```
<table definition> ::=  
    CREATE TABLE <table name>  
    (<table element> [ { <comma> <table element> }... ] )
```

```
<table element> ::=  
    <column definition>  
    | <table constraint definition>
```

```
<column definition> ::=  
    <column name> { <data type> | <domain name> }  
    [ <default clause> ]  
    [ <column constraint definition>... ]
```

Datentypen:
siehe Skript
Kapitel 5 Folie 54

■ Beispiel

```
CREATE TABLE Person (  
    PNR          int,  
    Vorname      varchar(50),  
    Nachname     varchar(50),  
    GebDatum     date,  
    Titel        varchar(50) DEFAULT 'None',  
);
```

SQL DDL: Column Constraints

■ Syntax

```
<column-constraint-def> ::=  
    [CONSTRAINT <constraint name>]  
    { NOT NULL | { PRIMARY KEY | UNIQUE } | <referential> | CHECK (<cond>) }
```

■ Beispiel

```
CREATE TABLE Person (  
    PNR          int          PRIMARY KEY,  
    Vorname      varchar(50),  
    Nachname     varchar(50) NOT NULL,  
    Alter        int          CHECK(Alter>0),  
    Titel        varchar(50) DEFAULT 'None',  
);
```

SQL DDL: Table Constraints

■ Syntax

```
<table-constraint-def> ::=  
    [ CONSTRAINT <constraint name> ]  
    { { PRIMARY KEY | UNIQUE } (<column list>) | <referential> | CHECK (cond) }  
  
<referential> ::=  
    FOREIGN KEY ( <referencing columns> )  
    REFERENCES <table name> [ ( <reference column list> ) ]  
    [ <referential triggered action> ]
```

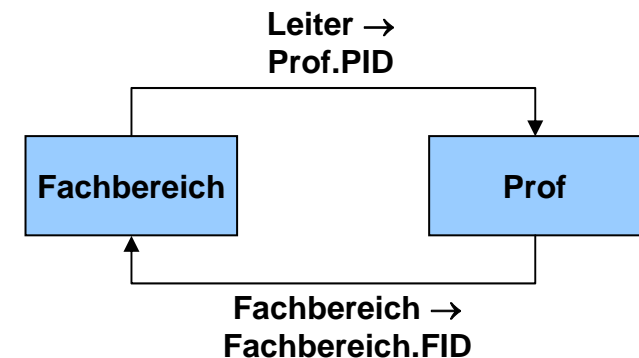
■ Beispiel

```
CREATE TABLE Kauf (  
    Person          int,  
    Buch            varchar(13),  
    CONSTRAINT pk_kauf PRIMARY KEY (Person, Buch),  
    CONSTRAINT fk_pers FOREIGN KEY (Person) REFERENCES Person (PNR),  
    CONSTRAINT fk_buch FOREIGN KEY (Buch) REFERENCES Buch (ISBN)  
);
```

SQL-DDL: Referenzzyklen

■ Zyklische Referenzen

- Beim Anlegen der ersten Tabelle ist andere Tabelle noch nicht bekannt
- Fremdschlüsselbeziehung muss nachträglich hinzugefügt werden



■ Beispiel

```
CREATE TABLE Fachbereich (  
    FID          int PRIMARY KEY,  
    Leiter       int);
```

```
CREATE TABLE Professor (  
    PID          int PRIMARY KEY,  
    Fachbereich  int,  
    CONSTRAINT fk_fb FOREIGN KEY (Fachbereich) REFERENCES Fachbereich(FID));
```

```
ALTER TABLE Fachbereich  
ADD CONSTRAINT fk_lt FOREIGN KEY (Leiter) REFERENCES Professor(PID);
```

SQL-DML: Anfragen (1)

■ Syntax

```
<select-exp> ::=  
    SELECT [ALL | DISTINCT] <select-item-commalist>  
    FROM <table-list>  
    [WHERE <cond>]  
    [GROUP BY <column-list>]  
    [HAVING <cond>]  
    [ORDER BY <column-list >]
```

■ Beispiele

```
SELECT Vorname, Nachname  
FROM Student  
WHERE Wohnort = 'Hamburg';
```

```
SELECT DISTINCT Wohnort  
FROM Student  
ORDER BY Wohnort ASC;
```

```
SELECT *  
FROM Fachbereich F, Student S  
WHERE S.Fach = F.FID;
```

```
SELECT F.FID, MAX(S.Semester)  
FROM Student S, Fachbereich F  
WHERE S.Fach = F.FID  
GROUP BY F.FID  
HAVING COUNT(*) > 10;
```

SQL-DML: Anfragen (2)

- Äußerer Verbund
 - Beispiele

R	A	B	C
	a ₁	b ₁	c ₁
	a ₂	b ₂	c ₂

S	C	D	E
	c ₁	d ₁	e ₁
	c ₃	d ₂	e ₂

Nat. Verbund

ERG	A	B	C	D	E
	a ₁	b ₁	c ₁	d ₁	e ₁

Right Outer Join

ERG	A	B	C	D	E
	a ₁	b ₁	c ₁	d ₁	e ₁
	--	--	c ₃	d ₂	e ₂

Left Outer Join

ERG	A	B	C	D	E
	a ₁	b ₁	c ₁	d ₁	e ₁
	a ₂	b ₂	c ₂	--	--

Full Outer Join

ERG	A	B	C	D	E
	a ₁	b ₁	c ₁	d ₁	e ₁
	a ₂	b ₂	c ₂	--	--
	--	--	c ₃	d ₂	e ₂

SQL-DML: Anfragen (3)

■ Syntax für (äußerer) Verbund

```
SELECT
  <select-item-commalist>
FROM
  <table: left> [<join-typ>] JOIN <table right> ON <cond>
```

■ Beispiele

```
SELECT * FROM
Mitarbeiter mi JOIN
Dienstwagen dw ON dw.MID = mi.ID
```

```
SELECT * FROM
Mitarbeiter mi LEFT [OUTER] JOIN
Dienstwagen dw ON dw.MID = mi.ID
```

```
SELECT * FROM
Mitarbeiter mi RIGHT [OUTER] JOIN
Dienstwagen dw ON dw.MID = mi.ID
```

```
SELECT * FROM
Mitarbeiter mi FULL [OUTER] JOIN
Dienstwagen dw ON dw.MID = mi.ID
```

SQL-DML: Anfragen (4)

- Geschachtelte Anfragen:

**äußere
Abbildung** **SELECT DISTINCT** FIGUR
FROM ROLLE
WHERE TITEL **IN** (**SELECT** TITEL
FROM DRAMA
WHERE AUTOR **IN** („Schiller“, „Goethe“));

aber auch...

```
SELECT    SUM(B.Betrag)
FROM      Bußgeldkatalog AS B,
            (SELECT * FROM Personen, FlenKartei WHERE PID = FahrerID) AS K
WHERE      B.BID = K.Vergehen
```