

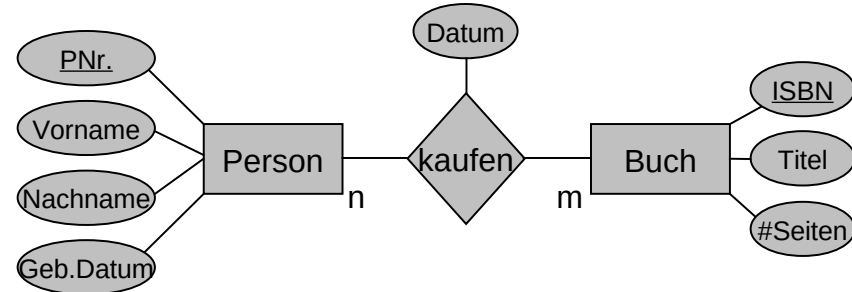
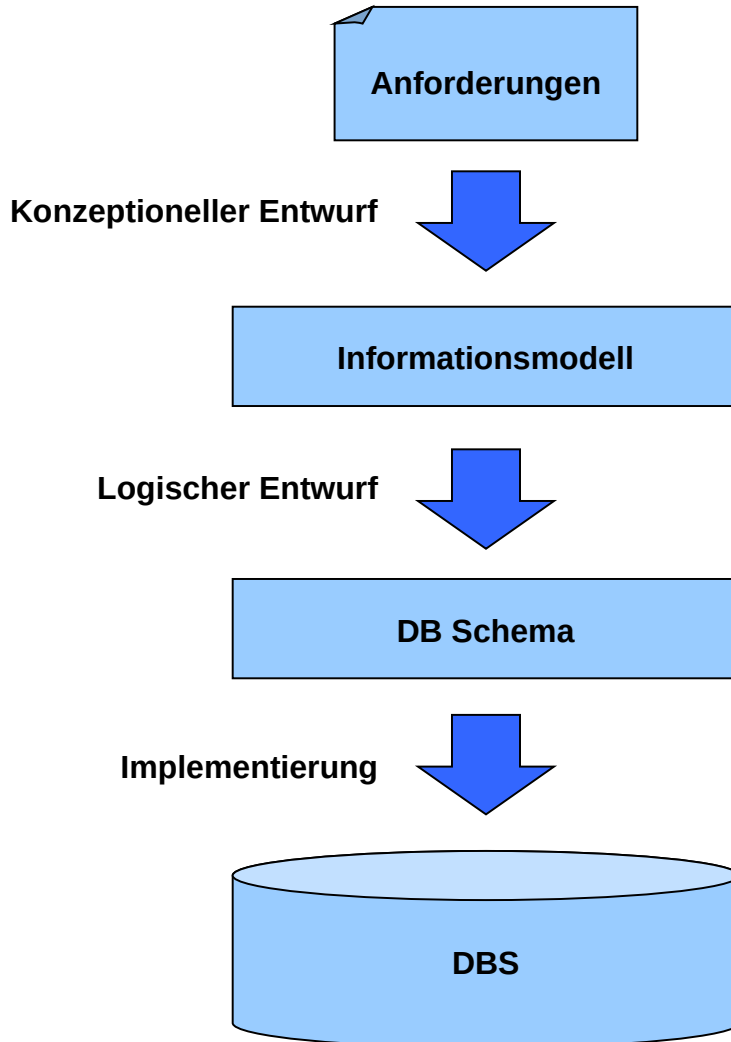
Grundlagen von Datenbanken

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

■ Beispiel

```
CREATE TABLE Person (  
    PNR          int          PRIMARY KEY,  
    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> ]
```

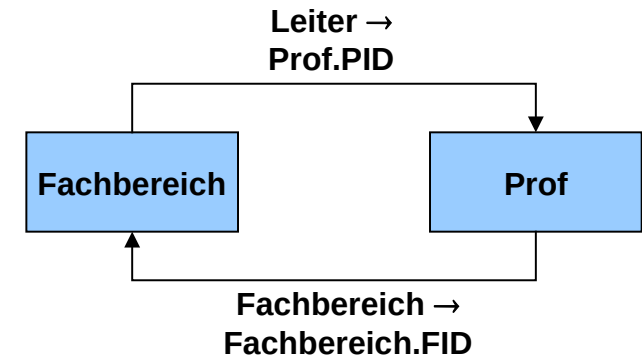
■ Beispiel

```
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

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

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

```
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: Geschachtelte Anfragen mit IN-Prädikat

row-constr [NOT] IN (table-exp)

- **x IN (a, b, . . . , z)** \Leftrightarrow **x = a OR x = b . . . OR x = z**
- **row-constr IN (table-exp)** \Leftrightarrow **row-constr = ANY (table-exp)**
- **x NOT IN erg** \Leftrightarrow **NOT (x IN erg)**

▪ **Beispiel:**

Finde die Namen der Schauspieler, die den Faust gespielt haben.

```
SELECT S.NAME
FROM   SCHAUSPIELER S
WHERE  'Faust' IN
      (SELECT D.FIGUR
       FROM   DARSTELLER D
       WHERE  D.PNR = S.PNR)
```

```
SELECT S.NAME
FROM   SCHAUSPIELER S
WHERE  S.PNR IN
      (SELECT D.PNR
       FROM   DARSTELLER D
       WHERE  D.FIGUR = 'Faust')
```

```
SELECT S.NAME
FROM   SCHAUSPIELER S,
      DARSTELLER D
WHERE  S.PNR = D.PNR AND
      D.FIGUR = 'Faust'
```


SQL-DML: Weitere Prädikate

- Nicht-quantifizierte Prädikate
 - Vergleichsprädikate
 - **BETWEEN-Prädikate**
 - IN-Prädikate
 - **Ähnlichkeitssuche**
 - **Prädikate über Nullwerten**
- Quantifizierte Prädikate
 - **ALL**
 - **ANY**
 - **EXISTS**
- Sonstige Prädikate
 - UNIQUE
 - ...