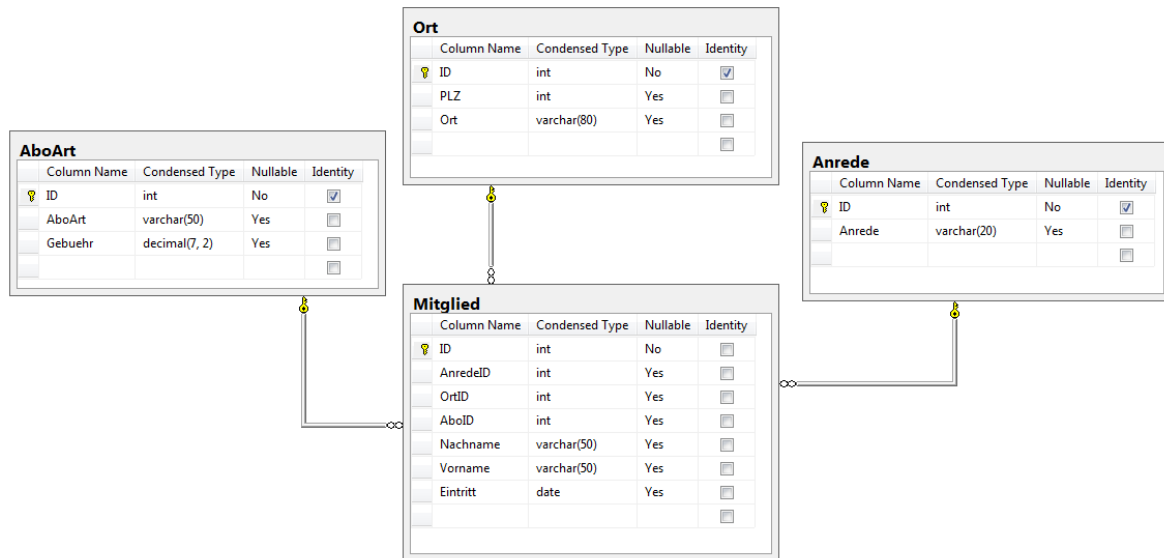


# SQL-Abfragen Teil 1

## Datenmodell



## 1. Einfache Abfragen

### A1.1

Selektieren Sie die alle Mitglieder mit Nachname, Vorname, Eintritt. Sortieren Sie die Ausgabe nach Eintritt Datum aufsteigend.

```
select nachname, vorname, eintritt
  from Mitglied
 order by eintritt asc;
```

### A1.2

Selektieren Sie die alle Mitglieder mit Nachname, Vorname, Eintritt welche seit dem 01.01.2000 eingetreten sind. Sortieren Sie die Ausgabe nach Nachname, Vorname aufsteigend.

```
select nachname, vorname, eintritt
  from Mitglied
 where eintritt > convert(date, '01.01.2000', 104)
 order by nachname asc, vorname asc
```

### A1.3

Selektieren Sie die alle Mitglieder deren Nachname mit „B“ beginnt und zeigen Sie den Vornamen, Nachname und Eintrittsdatum an.

```
select nachname, vorname, eintritt
  from Mitglied
 where nachname like 'B%'
 order by nachname desc, vorname desc
```

### A1.4

Selektieren Sie die alle Mitglieder deren Mitgliedsnummer kleiner 50 ist und der Eintritt vor 01.01.1995 liegt.

```
select nachname, vorname, eintritt
  from Mitglied
 where id < 50
    and eintritt < convert(date, '01.01.1995', 104)
 order by nachname desc, vorname desc
```

**A1.5**

Selektieren Sie die alle Mitglieder deren Nachname ein „zz“ enthält und das Eintrittsdatum nach dem 01.01.2000 liegt.

```
select nachname, vorname, eintritt
  from Mitglied
 where nachname like '%zz%'
    and eintritt > convert(date, '01.01.2000', 104)
 order by nachname desc, vorname desc
```

**A1.6**

Selektieren Sie die alle Mitglieder welche zwischen dem 01.01.1994 und 01.01.1997 eingetreten sind. Zeigen Sie die Ausgabe sortiert nach Eintritt Datum an.

```
select nachname, vorname, eintritt
  from Mitglied
 where eintritt between convert(date, '01.01.1994', 104) and convert(date, '01.01.1997', 104)
 order by nachname desc, vorname desc
```

**A1.7**

Selektieren Sie die alle Mitglieder deren Nachname ein „e“ enthält oder der Vorname mit „a“ endet.

```
select nachname, vorname, eintritt
  from Mitglied
 where nachname like '%e%' or vorname like '%a'
 order by nachname desc, vorname desc
```

## 2. Join Abfragen

### A2.1

Selektieren Sie die Abos, bei welchen die Gebühr kleiner oder gleich 500 ist und die Anrede „Frau“ ist. Sortieren Sie die Ausgabe aufsteigend nach Nachname, Vorname.

```
select nachname, vorname, eintritt
  from Mitglied
    inner join AboArt
      on mitglied.AboID = AboArt.ID
    inner join Anrede
      on Mitglied.AnredeID = Anrede.ID
 where AboArt.Gebuehr <= 500
    and Anrede.Anrede = 'Frau'
 order by Nachname asc, Vorname asc
```

### A2.2

Selektieren Sie alle Abos, bei welchen die Anrede „Herr“ steht und der Ort „Bern“ und die Aboart „Student“ ist.

```
select nachname, vorname, eintritt
  from Mitglied
    inner join AboArt
      on mitglied.AboID = AboArt.ID
    inner join Anrede
      on Mitglied.AnredeID = Anrede.ID
    inner join Ort
      on Mitglied.OrtID = Ort.ID
 where AboArt.AboArt = 'Student'
    and Anrede.Anrede = 'Herr'
 order by Mitglied.Nachname asc,
        Mitglied.Vorname asc
```

### A2.3

Listen Sie alle Abos, bei welchen die Anrede „Herr“ ist und der Ort „Zürich“ oder „Bern“ ist.

```
select nachname, vorname, eintritt
  from Mitglied
    inner join Anrede
      on Mitglied.AnredeID = Anrede.ID
    inner join Ort
      on Mitglied.OrtID = Ort.ID
 where Anrede.Anrede = 'Herr'
    and (Ort.Ort = 'Bern' or Ort.Ort = 'Zürich')
 order by Mitglied.Nachname asc,
        Mitglied.Vorname asc
```

### A2.4

Zeige alle Abos, bei welchen die Aboart nicht „Student“, die Anrede „Herr“ und der Ort nicht „Bern“ ist.

```
select nachname, vorname, eintritt
  from Mitglied
    inner join AboArt
      on mitglied.AboID = AboArt.ID
    inner join Anrede
      on Mitglied.AnredeID = Anrede.ID
    inner join Ort
      on Mitglied.OrtID = Ort.ID
 where AboArt.AboArt <> 'Student'
    and Anrede.Anrede = 'Herr'
    and Ort.Ort <> 'Bern'
```

```
order by Mitglied.Nachname asc,  
         Mitglied.Vorname asc
```

### A2.5

Zeige alle Abos an, bei welchen das Eintrittsdatum > 1.1.2005 oder die Anrede „Frau“ ist.

```
select nachname, vorname, eintritt  
from Mitglied  
     inner join Anrede  
           on Mitglied.AnredeID = Anrede.ID  
where Anrede.Anrede = 'Frau'  
     and Mitglied.Eintritt > convert(date, '01.01.2005', 104)  
order by Mitglied.Nachname asc,  
         Mitglied.Vorname asc
```

### A2.6

Suchen Sie die Abos, bei welchen die Aboart „Student“ oder „Jahresabo“ ist und die Anrede „Herr“ und Ort „Zürich“ ist

```
select nachname, vorname, eintritt  
from Mitglied  
     inner join AboArt  
           on mitglied.AboID = AboArt.ID  
     inner join Anrede  
           on Mitglied.AnredeID = Anrede.ID  
     inner join Ort  
           on Mitglied.OrtID = Ort.ID  
where (AboArt.AboArt = 'Student' or AboArt.AboArt = 'Jahresabo')  
     and Anrede.Anrede = 'Herr'  
     and Ort.Ort = 'Zürich'  
order by Mitglied.Nachname asc,  
         Mitglied.Vorname asc
```

## 3. Abfragen mit Funktionen

### A3.1

Erstellen Sie eine Abfrage, welche die Vornamen und Nachnamen der Mitglieder kommasetrennt zusammengefügt und als eine Spalte listet.

```
select vorname + ', ' + nachname as Fullname  
from Mitglied  
order by 1
```

### A3.2

Ändern Sie die Abfrage aus der A3.1 und zeigen Sie dabei Vorname u. Nachname in Großschrift an.

```
select upper(vorname) + ', ' + upper(nachname) as Fullname  
from Mitglied  
order by 1
```

### A3.3

Listen Sie alle Mitglieder mit Nachname u. Vorname und ordnen Sie die Ausgabe nach der Zeichenlänge (Anzahl Zeichen) der Nachnamen absteigend.

```
select vorname, nachname  
from Mitglied  
order by len(nachname) desc
```

### A3.4

Listen Sie alle Mitglieder mit der Kurzbezeichnung (Erster Buchstabe aus Nachname u. Vorname) aufsteigend.

```
select substring(nachname,1,1) + substring(vorname,1,1)
  from Mitglied
 order by 1
```

**A3.5**

Listen Sie alle Mitglieder mit Vorname, Nachname und dem Eintrittsjahr (z.B. 1990). Sortieren Sie die Ausgabe nach dem Eintrittsjahr absteigend.

```
select vorname, nachname, eintritt
  from Mitglied
 where YEAR(eintritt) = 1990
 order by Eintritt desc
```

**A3.6**

Ermitteln Sie mit einer Abfrage die Mitglieder (Vorname, Nachname, Eintritt) welche im Jahr 2005 eingetreten sind.

```
select vorname, nachname, eintritt
  from Mitglied
 where YEAR(eintritt) = 2005
 order by Eintritt desc
```

## 4. Abfragen mit Aggregatfunktionen

### A4.1

Ermitteln Sie mit einer Abfrage den tiefsten, höchsten und Durchschnittspreis der Abo Arten.

```
select min(gebuehr), max(gebuehr), avg(gebuehr)
from aboart
```

### A4.2

Ermitteln Sie mit einer Abfrage die Anzahl der weiblichen Mitglieder.

```
select count(*)
from Mitglied
    inner join Anrede
        on Mitglied.AnredeID = Anrede.ID
where Anrede.Anrede = 'Frau'
```

### A4.3

Ermitteln Sie mit einer Abfrage die Anzahl der Mitglieder (Vorname, Nachname, Eintritt) welche im Jahr 2005 eingetreten sind.

```
select count(*)
from Mitglied
where YEAR(eintritt) = 2005
```

### A4.4

Ermitteln Sie mit einer Abfrage die Anzahl der Mitglieder, welche in „Zürich“ (Ort) wohnen.

```
select count(*)
from Mitglied
    inner join Ort
        on Mitglied.OrtID = Ort.ID
where Ort.Ort = 'Zürich'
```

### A4.5

Ermitteln Sie mit einer Abfrage die Anzahl der Mitglieder pro Abo Art (d.h. für jede Abo Arten wird die Mitgliederanzahl ausgegeben).

```
select AboArt.AboArt, count(*)
from Mitglied
    inner join AboArt
        on mitglied.AboID = AboArt.ID
group by AboArt.AboArt
order by count(*)
```

### A4.6

Ermitteln Sie mit einer Abfrage das Total der Gebühren sämtlicher Mitglieder.

```
select sum(AboArt.Gebuehr) as Total
from Mitglied
    inner join AboArt
        on mitglied.AboID = AboArt.ID
```