

# Beispiele zu SQL

# Insert und Update

- **Machen Sie die notwendigen Einträge für die Hochzeit von zwei Personen am 26. Mai 2017 in 73447 Oberkochen.**
  - `select max(paarid) from paar; -- (max. PaarID)`
  - `insert into paar  
 (paarid, hdatum, plz, ort) values  
 (6, '2017-05-26', '73447', 'Oberkochen');`
  - `select * from ggr_hoch_kunde where paarid is  
 null; -- Singles raussuchen – bspw. Kunde 14 und 15`
  - `update kunde set paarid=6 where kundenr in  
 (14,15)`

# Scheidung durch SQL

- **Machen Sie die notwendigen Änderungen für die Scheidung eines Paares Ihrer Wahl.**

*bspw. Paar 5*

- `update kunde set paarid=null  
where paarid=5;`

**Ergebnis: Abfrage erfolgreich: 2 Zeilen**

- **Scheidung, aber Paar selbst noch nicht gelöscht**
- **Wäre auch möglich, einfach das Paar zu löschen?**

# Select-Ausdruck

- **Liste die Orte, in denen Kunden wohnen.**
  - `Select Ort from kunde`
  - sinnvolles Ergebnis?
- **Liste die Orte, in denen verheiratete Kunden wohnen.**
  - `Select distinct Ort  
from kunde where PaarId is not null`

# Select-Element und Order-by

- **Einschub Konkatenation**
  - Zeichenketten verketteten durch: “||”
- **Listen Sie die vollständigen Namen der Kunden in alphabetischer Reihenfolge (Nach-, Vorname)**
  - **Select**  
  
`KundenNr, VName || ' ' || NName as Kd_Name`  
  
`from Kunde`  
  
`order by NName, VName`

# Natural Join (NJ) Versuch 1

- Nenne Kunden mit Name und Hochzeitsdatum.

- ```
SELECT KundenNr,  
       VName || ' ' || NName as Kd_Name, HDatum  
from Kunde NJ Paar  
order by NName, VName
```

Gewünschtes Ergebnis?

# Natural Join (NJ) Versuch 1

- Nenne Kunden mit Name und Hochzeitsdatum.

- ```
Select KundenNr, VName || ' ' || NName as  
Kd_Name, hdatum  
from kunde join paar  
using (paarid)  
order by nname, vname
```

# Natural Join (NJ) Versuch 2

- **Nenne Geschenkwünsche von Paar 1.**

- ```
SELECT ProduktNr, PName
from Geschenkliste NJ Produkt
where PaarId = 1
```



# Join durch Konkatenation

- **Nenne Geschenkwünsche von Paar 1.**
  - `Select ProduktNr, PName  
from Geschenkliste, Produkt  
where PaarId = 1 and  
Geschenkliste.ProduktNr = Produkt.ProduktNr;`
  - `Select Geschenkliste.ProduktNr, PName  
from Geschenkliste, Produkt  
where PaarId = 1 and  
Geschenkliste.ProduktNr = Produkt.ProduktNr;`

# Aliase

- Nenne Geschenkwnünsche von Paar 1.
  - ```
Select p.ProduktNr, PName  
from Geschenkliste as g, Produkt p  
where PaarId = 1 and  
p.ProduktNr = g.ProduktNr;
```
  - ```
Select Produkt.ProduktNr, PName  
from Geschenkliste as g, Produkt p  
where PaarId = 1 and  
p.ProduktNr = g.ProduktNr;
```

Diese  
Variante  
geht jetzt  
nicht  
mehr!

# (Inner) Join

- Nenne Geschenkwünsche von Paar 1.

- ```
Select p.ProduktNr, PName
from Geschenkliste as g join Produkt p
                                using (ProduktNr)
where PaarId = 1
```

- ```
Select p.ProduktNr, PName
from Geschenkliste as g join Produkt p
                                on p.ProduktNr = g.ProduktNr
where PaarId = 1
```

# Outer Join

- **Nenne *alle* Paare mit ihren Geschenkwünschen.**
- ```
Select PaarId, ProduktNr, PName  
from Paar left join  
        (Geschenkliste natural join Produkt)  
        using (paarid)  
order by PaarId
```

# Outer Join

- **Liste alle Paare mit den Geschenken, die sie bekommen haben (→ 'abgegeben')**
  - `Select PaarId, ProduktNr, PName`  
`from Paar xxx`

# Outer Join

- Liste alle Paare mit den Geschenken, die sie bekommen haben ( $\rightarrow$  'abgegeben')
  - ```
Select PaarId, ProduktNr, PName  
from Paar left join  
  (Geschenkliste natural join Produkt) using  
  (paarid)  
where status = 'abgegeben'  
order by PaarId
```

# Outer Join

- Liste alle Paare mit den Geschenken, die sie bekommen haben (→ 'abgegeben')
  - ```
Select pa.PaarId, ProduktNr, PName, status  
from Paar pa left join (Geschenkliste  
natural join Produkt) as glp  
on pa.paarid=glp.paarid and  
status='abgegeben'  
order by PaarId
```

# Outer Join

- Liste alle Kunden mit ihrem Hochzeitsdatum.

- ```
Select Kd.KundenNr,  
      Kd.VName || ' ' || Kd.NName as Kd_Name,  
      HDatum  
  
from Kunde as Kd left join Paar Pa  
using (PaarId)  
  
order by Kd.NName, Kd.VName
```



# Outer Join

- Liste alle Kunden mit ihrem Hochzeitsdatum und ihrem Partner.
  - ```
Select Kd.KundenNr,  
      Kd.VName || ' ' || Kd.NName as Kd_Name,  
      P.VName || ' ' || P.NName as Partner_Name,  
      HDatum  
  
from Kunde as Kd xxxJ Paar Pa ...  
  
order by Kd.NName, Kd.VName
```

# Outer Join

- Liste alle Kunden mit ihrem Hochzeitsdatum und ihrem Partner.

- ```
Select Kd.KundenNr,  
      Kd.VName || ' ' || Kd.NName as Kd_Name,  
      P.VName || ' ' || P.NName as Partner_Name, HDatum  
from Kunde as Kd  
  
left join Paar pa using (PaarId)  
  
left join Kunde as p using (PaarId)  
  
order by Kd.NName, Kd.VName
```
- Problem?

# Outer Join

- Liste alle Kunden mit ihrem Hochzeitsdatum und ihrem Partner.

- ```
Select Kd.KundenNr,  
      Kd.VName || ' ' || Kd.NName as Kd_Name,  
      P.VName || ' ' || P.NName as Partner_Name, HDatum  
from Kunde as Kd  
left join Paar pa using (PaarId)  
left join Kunde as p  
on kd.PaarId=p.PaarId and kd.kundennr <> p.kundennr  
order by Kd.NName, Kd.VName
```

# Gruppierung

- **Liste Paare mit dem Preis ihres teuersten Geschenkwechs.**
- `Select PaarId, max(Preis)`  
`from Geschenkliste NJ Produkt`  
`group by PaarId`

# Summenbildung

- **Wie viel Geld wurde insgesamt für jedes Paar ausgegeben?** *(nur tatsächlich abgegebene Geschenke)*
  - `Select PaarId, sum(Preis)`  
`from Geschenkliste NJ Produkt`  
`where Status = 'abgegeben'`  
`group by PaarId`
  - was müsste geändert werden, damit auch Paare, für die gar nichts ausgegeben wurde, angezeigt werden?

# Summenbildung

- **Wie viel Geld wurde insgesamt für jedes Paar ausgegeben?** *(nur tatsächlich abgegebene Geschenke)*
  - ```
Select PaarId, sum(Preis)
from Paar as Pa left join
  (Geschenkliste natural join Produkt) as G
on Pa.PaarId = G.PaarId and Status =
  'abgegeben'
group by Pa.PaarId
```

# Zählen

- **Wie viele Geschenkwünsche hatte jedes Paar.**
  - `Select PaarId, count(Preis)`  
`from Geschenkliste NJ Produkt`  
`group by PaarId`
  - was müsste geändert werden, damit auch Paare, die sich gar nichts gewünscht haben, angezeigt werden?

# Zählen

- **Wie viele Geschenkwünsche hatte jedes Paar.**

- ```
Select PaarId, count(Preis)
from Paar left join
(Geschenkliste natural join Produkt)
using(PaarId)
group by PaarId
```



# "Besten-Liste"

- Welches Paar bzw. welche Paare hatten die meisten Geschenkwünsche.

- ```
Select PaarId, count(Preis)
from Geschenkliste NJ Produkt
group by PaarId
having count(Preis)
                >= All
                (Select count(Preis)
                 from Geschenkliste NJ Produkt
                 group by PaarId)
```