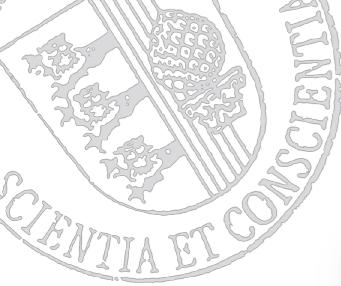


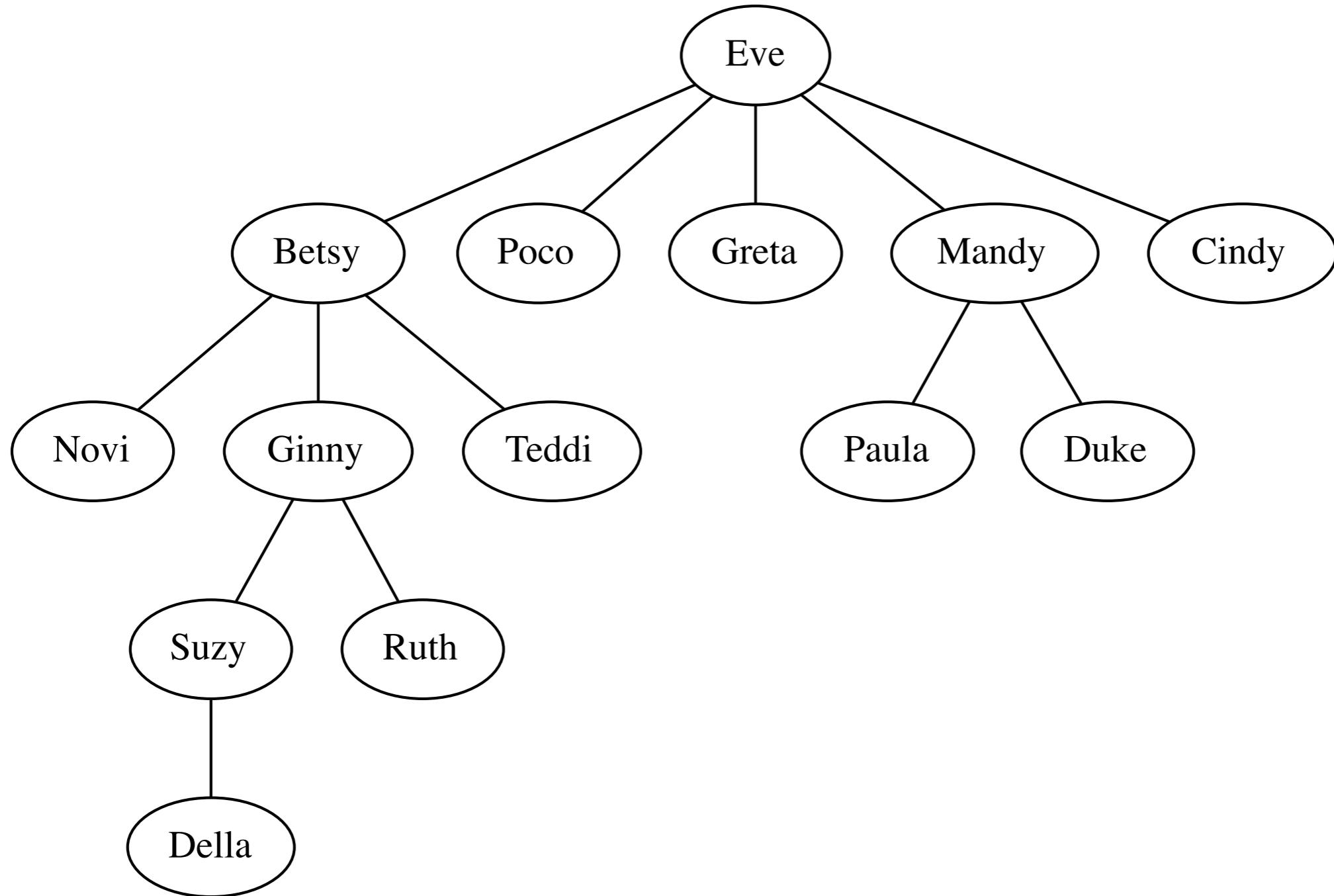
6. Baumstrukturen in SQL (Teil 2)





Bäume in Oracle – connect by

- **CONNECT BY** in Oracle ist eine **Report-Methode** zur Darstellung der Verzweigungen in Bäumen
- **CONNECT BY** bietet die Möglichkeit bestimmte Teilbäume oder Blätter auszublenden



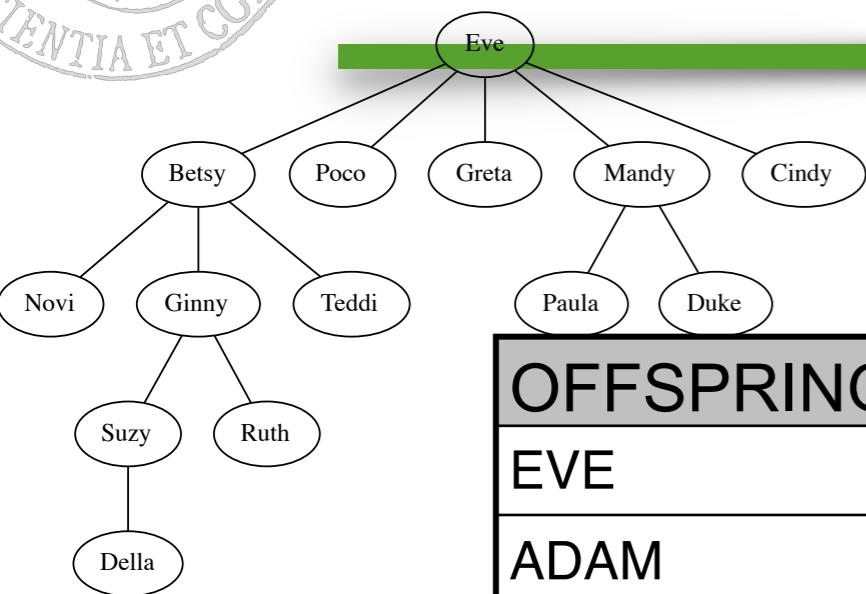


Tabelle breeding

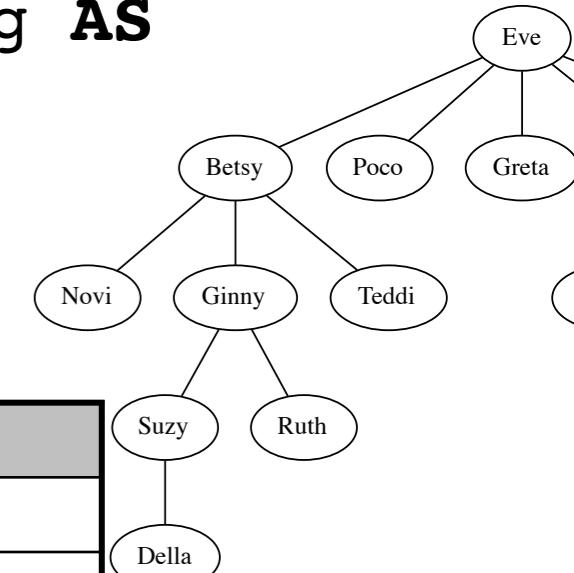
OFFSPRING	SEX	COW	BULL	BIRTHDATE
EVE	F			
ADAM	M			
BANDIT	M			
BETSY	F	EVE	ADAM	02-JAN-00
POCO	M	EVE	ADAM	15-JUL-00
GRETA	F	EVE	BANDIT	12-MAR-01
MANDY	F	EVE	POCO	22-AUG-01
CINDY	F	EVE	POCO	09-FEB-03
NOVI	F	BETSY	ADAM	30-MAR-03
GINNY	F	BETSY	BANDIT	04-DEC-03
DUKE	M	MANDY	BANDIT	24-JUL-04
TEDDI	F	BETSY	BANDIT	12-AUG-05
SUZY	F	GINNY	DUKE	03-APR-06
PAULA	F	MANDY	POCO	21-DEC-06
RUTH	F	GINNY	DUKE	25-DEC-06
DELLA	F	SUZY	BANDIT	11-OCT-08

```

SELECT cow,
       bull,
       lpad(' ', 6 * (LEVEL - 1), '.') || offspring AS
       offspring
FROM breeding
START WITH offspring = 'EVE'
CONNECT BY PRIOR offspring = cow;
    
```

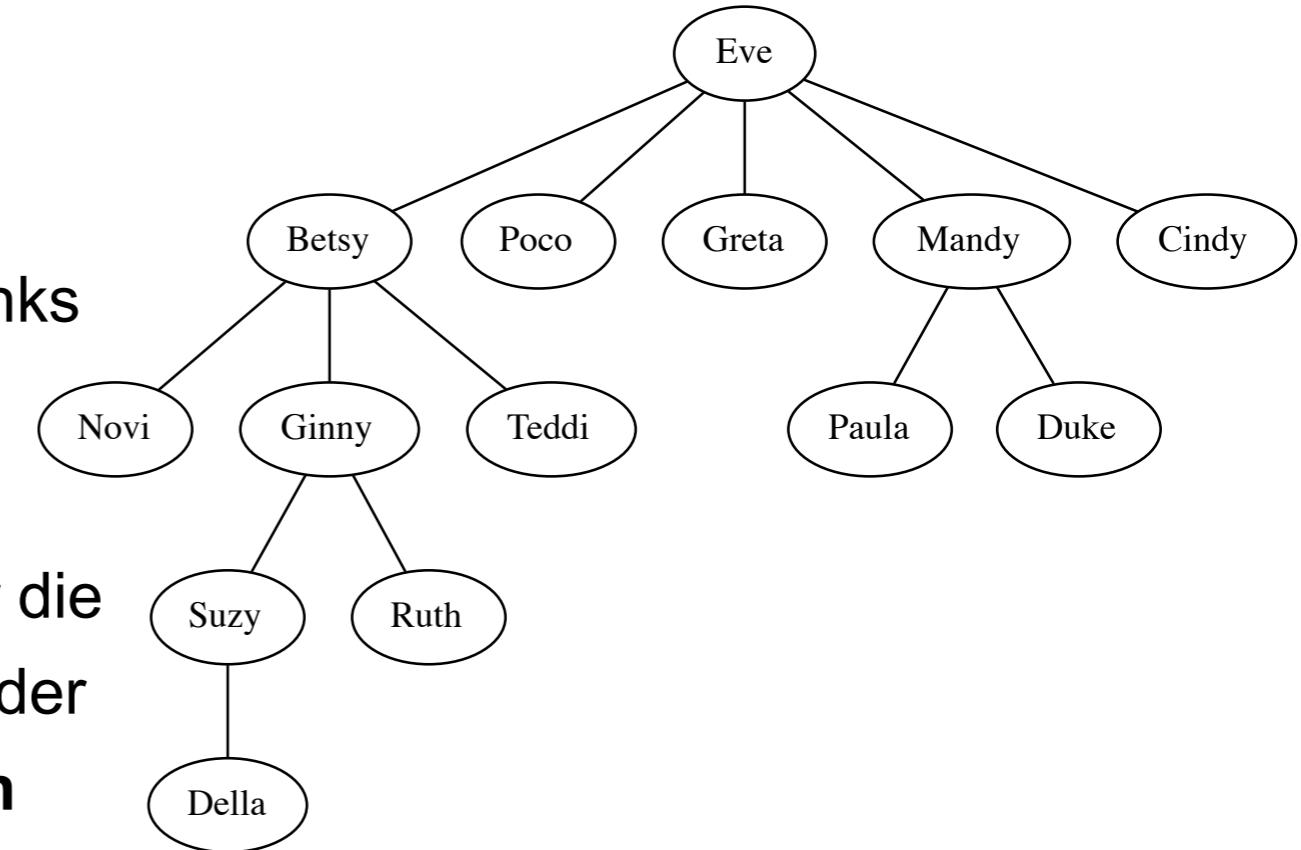
COW	BULL	OFFSPRING
		EVE
EVE	ADAM BETSY
BETSY	BANDIT GINNY
GINNY	DUKE RUTH
GINNY	DUKE SUZY
SUZY	BANDIT DELLA
BETSY	ADAM NOVI
BETSY	BANDIT TEDDI
EVE	POCO CINDY
EVE	BANDIT GRETA
EVE	POCO MANDY
MANDY	BANDIT DUKE
MANDY	POCO PAULA
EVE	ADAM POCO

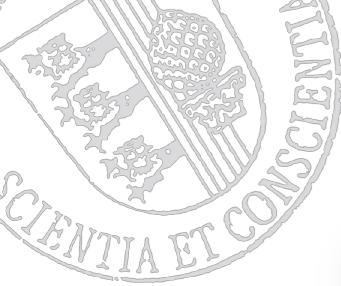
Visualisierung der familiären Beziehungen



```
SELECT cow,  
       bull,  
       lpad(' ', 6 * (LEVEL - 1), '.') || offspring AS  
       offspring  
  FROM breeding  
 START WITH offspring = 'EVE'  
CONNECT BY PRIOR offspring = cow;
```

- **START WITH** definiert den “Wurzelknoten”
- Der *LPAD*-Aufruf hängt an das Leerzeichen links
 $6 * \text{LEVEL} - 1$ viele ‘.’ an
- **CONNECT BY PRIOR offspring = cow:**
Es soll die nächste Zeile gefunden werden, für die
der Wert der **cow-Spalte** gleich dem Wert in der
offspring-Spalte in der **vorherigen Generation**
(Level) ist
- Die Position des **PRIOR** ist also entscheidend!





```
... CONNECT BY PRIOR column1 = column2;
```

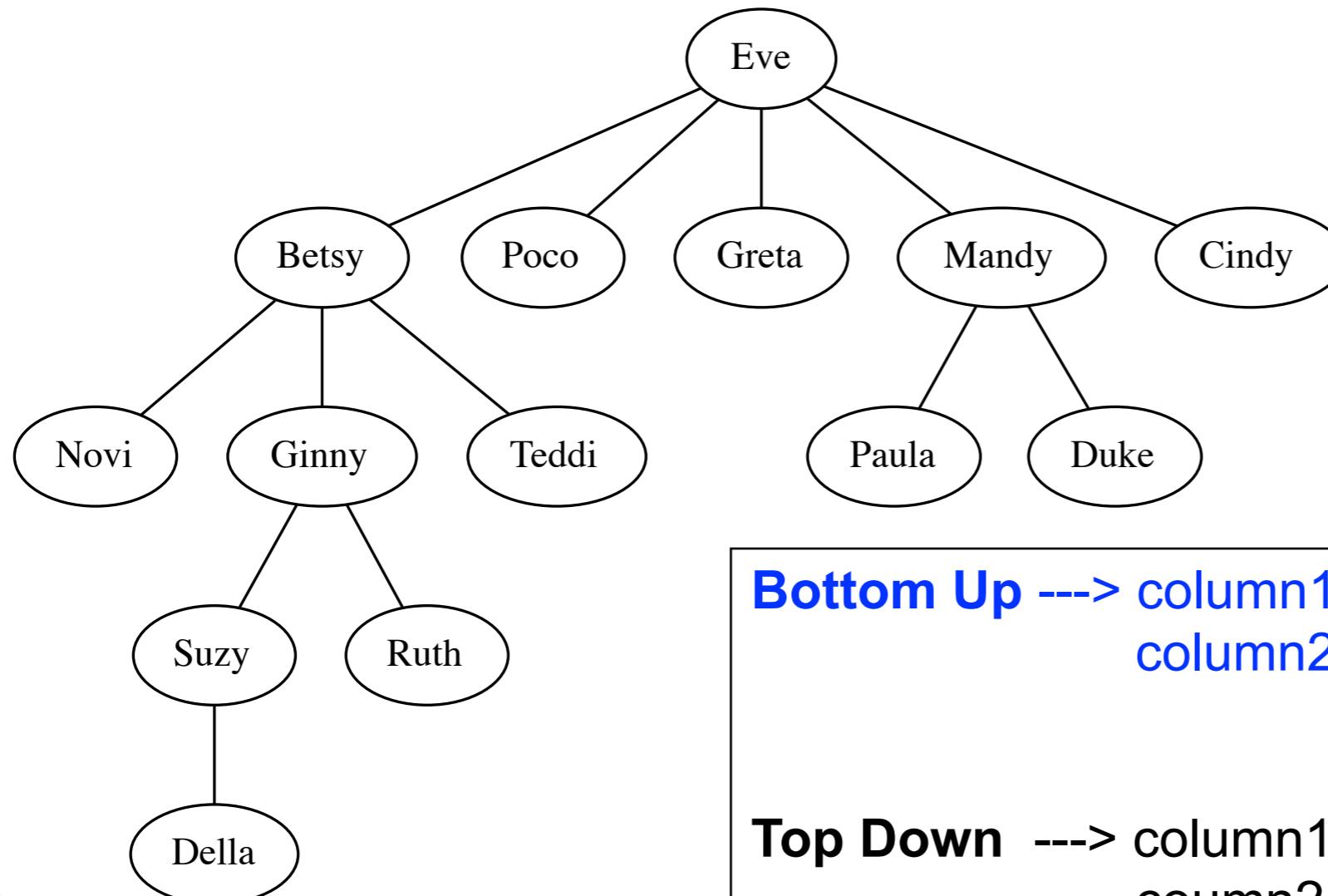
Richtung:

Bottom Up ---> column1 = Elternknoten
column2 = Kindknoten

Top Down ---> column1 = Kindknoten
column2 = Elternknoten

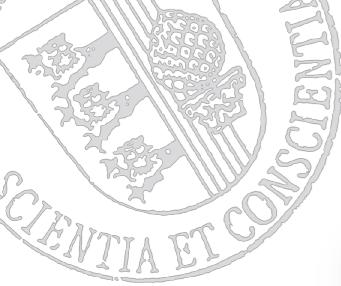
“From the bottom up”: Vorfahren von Della

```
SELECT cow, bull, offspring  
FROM breeding  
START WITH offspring = 'DELLA'  
CONNECT BY PRIOR cow = offspring;
```



Bottom Up ---> column1 = Elternknoten
column2 = Kindknoten

Top Down ---> column1 = Kindknoten
column2 = Elternknoten



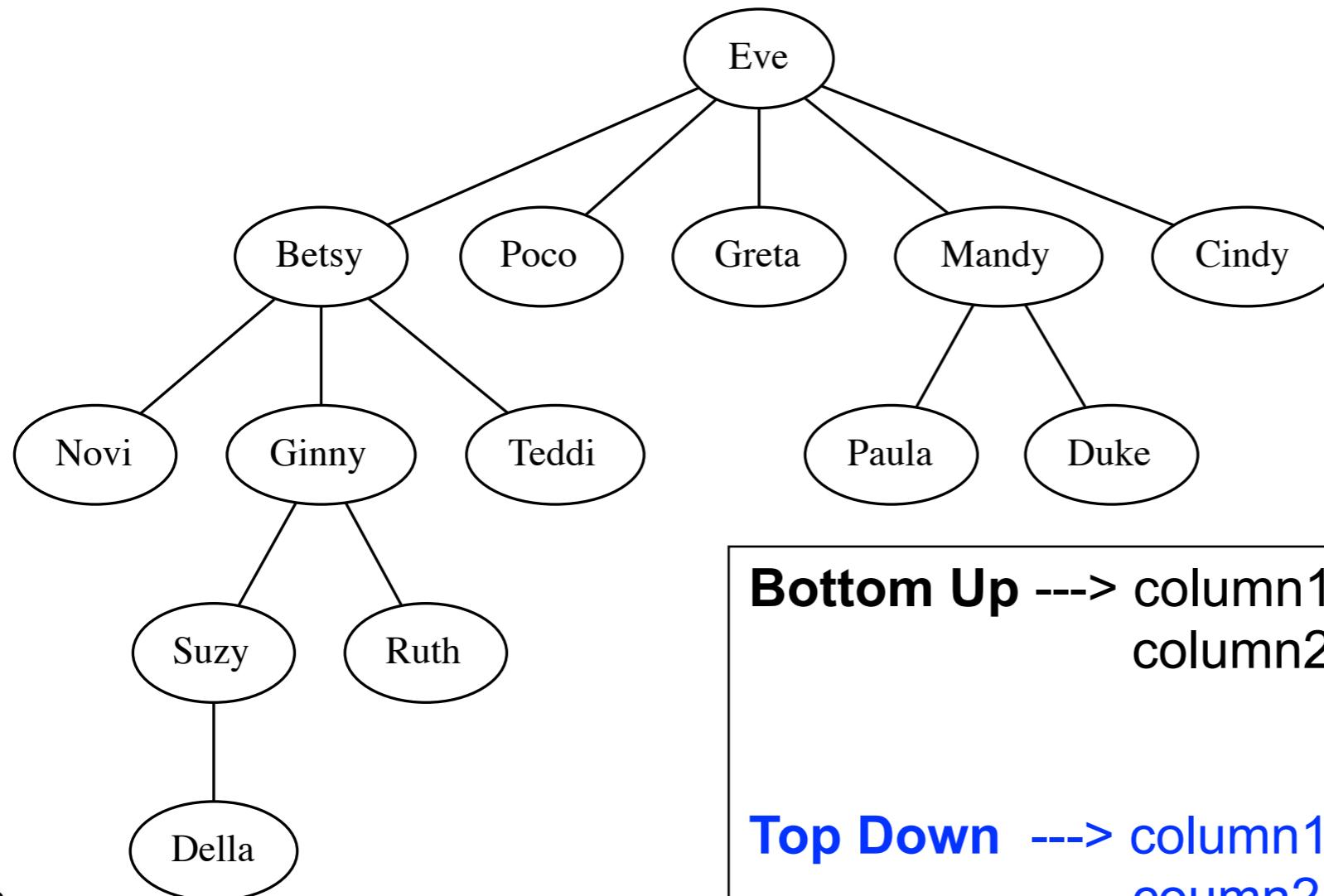
“From the bottom up”: Vorfahren von Della

```
SELECT cow,  
       bull,  
       lpad(' ', 6 * (LEVEL - 1), '.') || offspring AS  
       offspring  
  FROM breeding  
 START WITH offspring = 'DELLA'  
CONNECT BY PRIOR cow = offspring;
```

COW	BULL	OFFSPRING
SUZY	BANDIT	DELLA
GINNY	DUKE SUZY
BETSY	BANDIT GINNY
EVE	ADAM BETSY
	 EVE

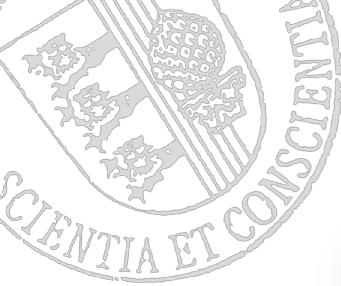
“From the top down”: Nachfahren von Betsy

```
SELECT cow, bull, offspring  
FROM breeding  
START WITH offspring = 'BETSY'  
CONNECT BY PRIOR offspring = cow;
```



**Bottom Up ---> column1 = Elternknoten
column2 = Kindknoten**

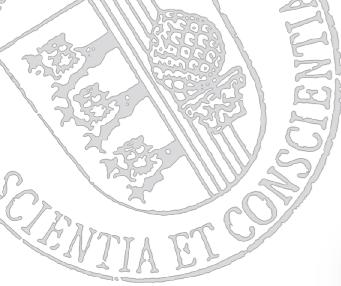
**Top Down ---> column1 = Kindknoten
coumn2 = Elternknoten**



“From the top down”: Nachfahren von Betsy

```
SELECT cow,  
       bull,  
       lpad(' ', 6 * (LEVEL - 1), '.') || offspring AS  
       offspring  
  FROM breeding  
 START WITH offspring = 'BETSY'  
CONNECT BY PRIOR offspring = cow;
```

COW	BULL	OFFSPRING
Eve	Adam	Betsy
Betsy	Bandit Ginny
Ginny	Duke Ruth
Ginny	Duke Suzy
Suzy	Bandit Della
Betsy	Adam Novi
Betsy	Bandit Teddi

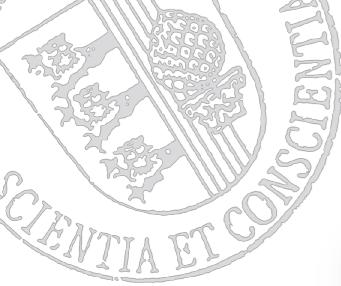


Bäume in Oracle – connect by – Individuen und Zweige ausblenden

```
SELECT cow, bull, lpad(' ', 6 * (LEVEL - 1), '.') || offspring
      AS offspring
  FROM breeding
 START WITH offspring = 'EVE'
CONNECT BY PRIOR offspring = cow
      AND offspring != 'BETSY';
```

Schließt BETSY und alle
ihre Nachkommen aus

COW	BULL	OFFSPRING
		EVE
EVE	ADAM POCO
EVE	BANDIT GRETA
EVE	POCO MANDY
MANDY	BANDIT DUKE
MANDY	POCO PAULA
EVE	POCO CINDY

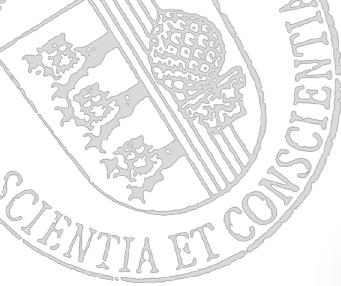


Bäume in Oracle – connect by – Individuen und Zweige ausblenden

```
SELECT cow, bull, lpad(' ', 6 * (LEVEL - 1), '.') || offspring
      AS offspring
  FROM breeding
 WHERE offspring != 'BETSY'
START WITH offspring = 'EVE'
CONNECT BY PRIOR offspring = cow;
```

Schließt BETSY aber
keine Nachkommen aus

COW	BULL	OFFSPRING
		EVE
BETSY	ADAM NOVI
BETSY	BANDIT GINNY
GINNY	DUKE SUZY
SUZY	BANDIT DELLA
GINNY	DUKE RUTH
BETSY	BANDIT TEDDI
EVE	ADAM POCO
EVE	BANDIT GRETA
EVE	POCO MANDY
MANDY	BANDIT DUKE
MANDY	POCO PAULA
EVE	POCO CINDY

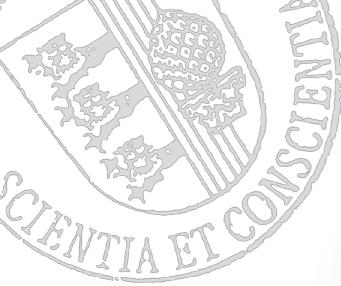


```
SELECT cow, bull, lpad(' ', 6 * (LEVEL - 1), '.') || offspring
      AS offspring
  FROM breeding
 START WITH offspring = 'EVE'
CONNECT BY PRIOR offspring = cow;
```

ist nur eine Kurzform für

```
SELECT cow, bull, lpad(' ', 6 * (LEVEL - 1), '.') || offspring
      AS offspring
  FROM breeding
 WHERE CONNECT_BY_ROOT(offspring) = 'EVE'
CONNECT BY PRIOR offspring = cow;
```

CONNECT_BY_ROOT gibt immer den Wert der Baumwurzel zurück und ermöglicht somit den Zugriff auf das jeweilige Wurzelelement der Zeile.

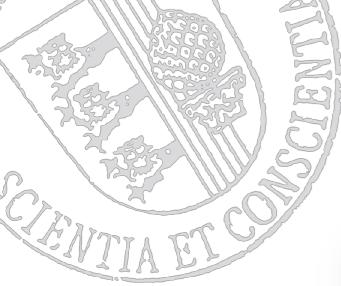


Ohne CONNECT_BY_ROOT, bzw. START WITH würden alle Teilbäume aufgezählt.

Das kann durchaus interessant sein.

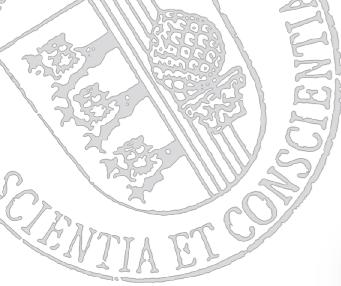
Zu jedem Rind sollen alle weiblichen Vorfahren u. das jeweilige Verwandtschaftsverhältnis ausgegeben werden:

```
SELECT CONNECT_BY_ROOT(offspring) kuh,  
       offspring vorfahr,  
       CASE  
           WHEN LEVEL = 2 THEN 'Mutter'  
           WHEN LEVEL = 3 THEN 'Großmutter'  
           ELSE RPAD('Ur', 2 * (LEVEL - 3), 'ur') ||  
                'großmutter'  
       END beziehung  
FROM breeding  
WHERE offspring != CONNECT_BY_ROOT(offspring)  
CONNECT BY PRIOR cow = offspring  
ORDER BY kuh, LEVEL;
```



Bäume in Oracle – connect by – Details

Kuh	Vorfahr	Beziehung
BETSY	EVE	Mutter
CINDY	EVE	Mutter
DELLA	SUZY	Mutter
DELLA	GINNY	Großmutter
DELLA	BETSY	Urgroßmutter
DELLA	EVE	Ururgroßmutter
DUKE	MANDY	Mutter
DUKE	EVE	Großmutter
GINNY	BETSY	Mutter
GINNY	EVE	Großmutter
GRETA	EVE	Mutter
MANDY	EVE	Mutter
NOVI	BETSY	Mutter
NOVI	EVE	Großmutter
PAULA	MANDY	Mutter
PAULA	EVE	Großmutter
POCO	EVE	Mutter
RUTH	GINNY	Mutter
RUTH	BETSY	Großmutter
RUTH	EVE	Urgroßmutter
SUZY	GINNY	Mutter
SUZY	BETSY	Großmutter
SUZY	EVE	Urgroßmutter
TEDDI	BETSY	Mutter
TEDDI	EVE	Großmutter



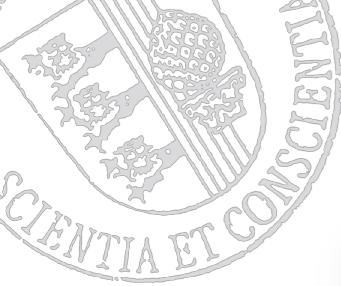
Zyklen:

Fehler beim Eintragen können zu Zyklen führen.

Fehler bei der Ausführung von **CONNECT BY**.

Die Angabe von **NOCYCLE** verhindert dies. Zyklen werden dann genau 1x durchlaufen (wichtig bei **Graphen**).

Die Pseudospalte **CONNECT_BY_ISCYCLE** enthält dann den Wert 1.



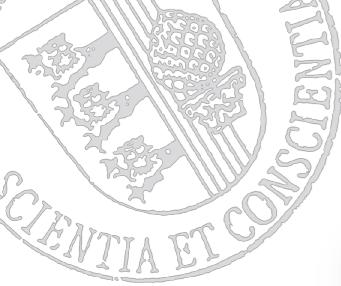
Bäume in Oracle – connect by – weitere Funktionen

SYS_CONNECT_BY_PATH(<Spalte>, <Separator>)

gibt den **Pfad** von der Baumwurzel zum Wert von
<Spalte> des aktuellen Tupel an

```
SELECT cow, bull, SYS_CONNECT_BY_PATH(offspring, '/') ospr
FROM breeding
START WITH offspring = 'EVE'
CONNECT BY PRIOR offspring = cow;
```

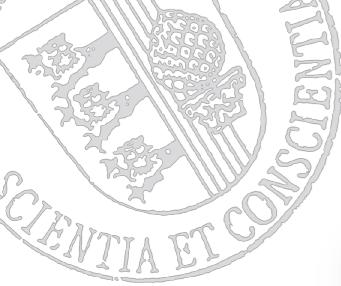
COW	BULL	OSPR
		/EVE
EVE	ADAM	/EVE/BETSY
BETSY	ADAM	/EVE/BETSY/NOVI
BETSY	BANDIT	/EVE/BETSY/GINNY
GINNY	DUKE	/EVE/BETSY/GINNY/SUZY
SUZY	BANDIT	/EVE/BETSY/GINNY/SUZY/DELLA
GINNY	DUKE	/EVE/BETSY/GINNY/RUTH
BETSY	BANDIT	/EVE/BETSY/TEDDI
EVE	ADAM	/EVE/POCO
EVE	BANDIT	/EVE/GRETA
EVE	POCO	/EVE/MANDY
MANDY	BANDIT	/EVE/MANDY/DUKE
MANDY	POCO	/EVE/MANDY/PAULA
EVE	POCO	/EVE/CINDY



CONNECT_BY_ISLEAF gibt 1 zurück, wenn es sich bei dem aktuellen Tupel um ein Blatt handelt (sonst 0).

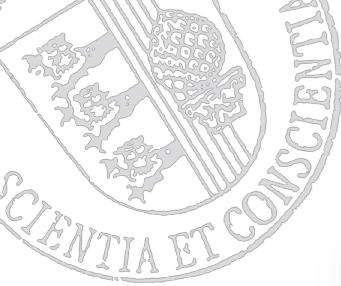
Aufgabe: Gib die mütterliche Abstammungsgeschichte aller kinderlosen Kühe wieder:

```
SELECT offspring,
       SUBSTR(SYS_CONNECT_BY_PATH(offspring,
                                     ' -> '), 5) abstammung
  FROM breeding
 WHERE CONNECT_BY_ISLEAF = 1 AND sex = 'F'
 START WITH offspring = 'EVE'
 CONNECT BY PRIOR offspring = cow;
```



Bäume in Oracle – connect by – weitere Funktionen

OFFSPRING	ABSTAMMUNG
NOVI	EVE -> BETSY -> NOVI
DELLA	EVE -> BETSY -> GINNY -> SUZY -> DELLA
RUTH	EVE -> BETSY -> GINNY -> RUTH
TEDDI	EVE -> BETSY -> TEDDI
GRETA	EVE -> GRETA
PAULA	EVE -> MANDY -> PAULA
CINDY	EVE -> CINDY

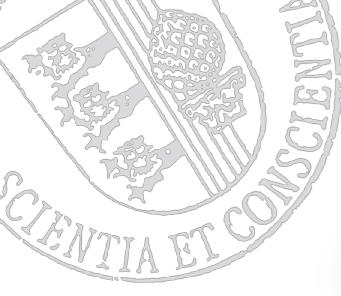


connect by – Grundregeln und Zusammenfassung

- Reihenfolge der Klauseln

SELECT; FROM; WHERE; START WITH; CONNECT BY [NOCYCLE]; ORDER BY

- **PRIOR** erzwingt die Ausführung von den Blättern in Richtung Ursprung (wenn die PRIOR-Spalte das Elternteil ist) oder vom Stamm in Richtung Blätter (Wenn die PRIOR-Spalte das Kind ist)
- Eine **WHERE**-Klausel eliminiert zwar einzelne Elemente, aber nicht deren Abkömmlinge bzw. Vorfahren (je nach Stellung von PRIOR) aus dem Baum
- Eine Einschränkung in **CONNECT BY** (insbesondere ein ‘ungleich’) eliminiert sowohl ein einzelnes Element als auch dessen Abkömmlinge (oder Vorfahren, in Abhängigkeit von der Laufrichtung innerhalb des Baums).
- **CONNECT BY** kann bei einer Tabellenverknüpfung nicht innerhalb der **WHERE**-Klausel verwendet werden



Mögliche Klausuraufgaben