OLAP PLF

Abgeänderte Tabellen:

drop table person cascade constraints;  
drop table country cascade constraints;  
drop table calls cascade constraints;  
drop sequence seq\_calls;

create sequence seq\_calls;

create table person(pnrp integer primary key, pname varchar2(20));

create table country (country\_idp integer primary key, country\_name varchar2(20));

create table calls(call\_nrp integer, pnrp integer references person(pnrp),

country\_idp integer references country (country\_idp),

duration integer,

call\_date date);

Aufgabe 1:

select decode(grouping\_id(pname),1,'== all Persons', pname) as Person,

decode(grouping\_id(country\_name),1,'== all countries', country\_name) as Country,

decode(grouping\_id(to\_char(call\_date, 'YYYY')),1,'== all years',

to\_char(call\_date, 'YYYY')) as Year,sum(duration) as SUMME

from calls c join person p on c.pnrp = p.pnrp join country co on c.country\_idp = co.country\_idp

group by cube(pname, country\_name, to\_char(call\_date, 'YYYY'))

order by year DESC, country\_name;

PERSON COUNTRY YEAR SUMME

-------------------- -------------------- ------------ ----------

Einstein Austria 2017 100

Zweistein Austria 2017 160

Vierstein Austria 2017 10

== all Persons Austria 2017 280

Dreistein Austria 2017 10

Vierstein Germany 2017 10

Einstein Germany 2017 10

== all Persons Germany 2017 30

Dreistein Germany 2017 10

Dreistein Great Britain 2017 320

== all Persons Great Britain 2017 420

Zweistein Great Britain 2017 30

Einstein Great Britain 2017 50

Vierstein Great Britain 2017 20

Einstein == all countries 2017 160

Zweistein == all countries 2017 190

== all Persons == all countries 2017 730

Dreistein == all countries 2017 340

Vierstein == all countries 2017 40

Vierstein Austria 2016 10

== all Persons Austria 2016 480

Einstein Austria 2016 470

Einstein Germany 2016 250

Vierstein Germany 2016 10

== all Persons Germany 2016 260

Einstein Great Britain 2016 530

== all Persons Great Britain 2016 650

Zweistein Great Britain 2016 110

Vierstein Great Britain 2016 10

Vierstein == all countries 2016 30

Einstein == all countries 2016 1250

Zweistein == all countries 2016 110

== all Persons == all countries 2016 1390

== all Persons Austria == all years 760

Vierstein Austria == all years 20

Dreistein Austria == all years 10

Zweistein Austria == all years 160

Einstein Austria == all years 570

== all Persons Germany == all years 290

Vierstein Germany == all years 20

Dreistein Germany == all years 10

Einstein Germany == all years 260

Vierstein Great Britain == all years 30

== all Persons Great Britain == all years 1070

Einstein Great Britain == all years 580

Dreistein Great Britain == all years 320

Zweistein Great Britain == all years 140

Einstein == all countries == all years 1410

Vierstein == all countries == all years 70

Dreistein == all countries == all years 340

== all Persons == all countries == all years 2120

Zweistein == all countries == all years 300

52 Zeilen gewählt

Aufgabe 2:

SELECT to\_char(call\_date, 'MM') as month, duration,

FIRST\_VALUE(duration) OVER (ORDER BY to\_char(call\_date, 'MM') ROWS BETWEEN 1 PRECEDING AND 0 FOLLOWING ) AS "Duration former month",

to\_char(

duration -

FIRST\_VALUE(duration) OVER (ORDER BY to\_char(call\_date, 'MM') ROWS BETWEEN 1 PRECEDING AND 0 FOLLOWING ),'999999.9') || ' C' AS "Change"

FROM calls

where to\_char(call\_date, 'YYYY') = '2017' AND country\_idp = (select country\_idp from country where country\_name = 'Austria');

MONTH DURATION Duration former month Change

----- ---------- --------------------- -----------

03 10 10 .0 C

03 10 10 .0 C

03 100 10 90.0 C

05 10 100 -90.0 C

05 10 10 .0 C

06 70 10 60.0 C

06 10 70 -60.0 C

06 10 10 .0 C

06 10 10 .0 C

06 20 10 10.0 C

08 10 20 -10.0 C

08 10 10 .0 C

12 Zeilen gewählt

Aufgabe 3:

CREATE VIEW view3 AS

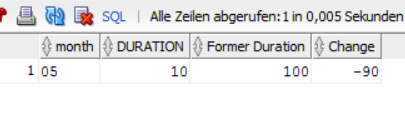
select to\_char(call\_date, 'MM') as "month", duration,

FIRST\_VALUE(duration) over (order by calls.call\_date ROWS BETWEEN 1 PRECEDING AND 0 FOLLOWING) AS "Former Duration",

duration - FIRST\_VALUE(duration) over (order by calls.call\_date rows between 1 preceding and 0 following) as "Change"

from calls where country\_idp = (select country\_idp from country where to\_char(call\_date, 'YYYY') = 2017 AND country\_name = 'Austria');

SELECT \* FROM view3 WHERE "Change" = (SELECT MIN("Change") FROM view3);



Aufgabe 4:

CREATE VIEW view4 AS

select country\_name, pname, duration,

DENSE\_RANK() OVER (ORDER BY duration DESC) AS rank from calls c

join country co on c.country\_idp = co.country\_idp join person p on p.pnrp = c.pnrp

GROUP BY country\_name, pname, duration;

SELECT \* FROM view4 WHERE rank <= 2;

