Screencast 5-9

Common table expressions (non-recursive)

Database - ADWLT

CTE

WITH **a** as (select …)
SELECT …
FROM **a** …

Task

Prepare a query for a report with the following structure:

- Country code (billing address)
- Product category
- Number of orders
- Taxes payed

Show only orders made in 2012. If there were no orders for a particular combination of category and year then display 0 instead.

Find all countries that had 100 or less orders in clothes category in 2012.

Original query:

```
select
	p.productid,
	sh.taxamt,
	sh.duedate,
	sh.billtoaddressid
from salesorderheader as sh
	join salesorderdetail as sd using (salesorderid)
	join product p using(productid)
	join productsubcategory ps using (productsubcategoryid)
	join productcategory pc using (productcategoryid)
```

CTE with country-category combinations:

```
with c_c as (
select * from
(select distinct countryregioncode from address) c
cross join
(select productcategoryid id, name cname from productcategory) p)
select * from c_c
```

Better query:

```
with c_c as (
select * from
(select distinct countryregioncode country from address) c
cross join
(select productcategoryid id, name cname from productcategory) p),
orders as (
select
	--p.productid,
	sum(sh.taxamt) as taxes,
	count(distinct sh.salesorderid) as ordersnr,
	--sh.duedate,
	--sh.billtoaddressid,
	a.countryregioncode as country,
	pc."name" as cname,
	pc.productcategoryid as id
from salesorderheader as sh
	join salesorderdetail as sd using (salesorderid)
	join product p using(productid)
	join productsubcategory ps using (productsubcategoryid)
	join productcategory pc using (productcategoryid)
	join address a on a.addressid = sh.billtoaddressid
where extract(year from sh.duedate ) = 2012
group by 3, 4, 5
having count(distinct sh.salesorderid) <= 100
)
select * from
	c_c left join orders
		on c_c.country = orders.country
			and c_c.id = orders.id
```

Filter conditions are still insufficient. Correct filter should check the number of orders in ‘Clothing’ category:

```
with c_c as (
select * from
(select distinct countryregioncode country from address) c
cross join
(select productcategoryid id, name cname from productcategory) p),
orders as (
select
	--p.productid,
	sum(sh.taxamt) as taxes,
	count(distinct sh.salesorderid) as ordersnr,
	--sh.duedate,
	--sh.billtoaddressid,
	a.countryregioncode as country,
	pc."name" as cname,
	pc.productcategoryid as id,
	sum(count(distinct case pc."name" when 'Clothing' then sh.salesorderid else null
end))
		over (partition by a.countryregioncode) as cl_ord_nr
from salesorderheader as sh
```

```
	join salesorderdetail as sd using (salesorderid)
	join product p using(productid)
	join productsubcategory ps using (productsubcategoryid)
	join productcategory pc using (productcategoryid)
	join address a on a.addressid = sh.billtoaddressid
where extract(year from sh.duedate ) = 2012
group by 3, 4, 5
)
select * from
	c_c left join orders
		on c_c.country = orders.country
			and c_c.id = orders.id
where cl_ord_nr <= 100 or cl_ord_nr is null
```

Final version of this query with NULLs replaced by 0 is shown below:

```
with c_c as (
select * from
(select distinct countryregioncode country from address) c
cross join
(select productcategoryid id, name cname from productcategory) p),
orders as (
select
	sum(sh.taxamt) as taxes,
	count(distinct sh.salesorderid) as ordersnr,
	a.countryregioncode as country,
	pc."name" as cname,
	pc.productcategoryid as id,
	sum(count(distinct case pc."name" when 'Clothing' then sh.salesorderid else null
end))
		over (partition by a.countryregioncode) as cl_ord_nr
from salesorderheader as sh
	join salesorderdetail as sd using (salesorderid)
	join product p using(productid)
	join productsubcategory ps using (productsubcategoryid)
	join productcategory pc using (productcategoryid)
	join address a on a.addressid = sh.billtoaddressid
where extract(year from sh.duedate ) = 2012
group by 3, 4, 5
)
select c_c.country,
	c_c.cname,
	coalesce(ordersnr, 0) ordersnr,
	coalesce(taxes, 0) taxes
from
	c_c left join orders
		on c_c.country = orders.country
			and c_c.id = orders.id
where cl_ord_nr <= 100 or cl_ord_nr is null
```