Заявки от 10 занятие

```
SELECT
    t1.firstname || ' ' || t1.lastname employee
,   COUNT(*) n_orders
FROM
-- АЛТЕРНАТИВЕН INNER JOIN --
    employees t1
,   orders t2
WHERE
--  свързване на таблиците --
    t2.employee_id = t1.employee_id
    AND
    t2.shipped_date IS NOT NULL
GROUP BY
    t1.firstname || ' ' || t1.lastname
ORDER BY
    2 DESC
```

Задача

Да се изчисли стойността на продадените продукти в хиляди за 2015г. като продуктите с продажби под 7 хил. се обединят в категория "Други".

```
--1
SELECT
    t1.product_name
,   SUM(t2.unit_price * t2.quantity) product_sales
FROM
    products t1
        INNER JOIN
    order_details t2
        ON t2.product_id = t1.product_id
        INNER JOIN
    orders t3
        ON t2.order_id = t3.order_id
WHERE
    EXTRACT(YEAR FROM t3.order_date) = 2015
GROUP BY
    t1.product_name
ORDER BY
    2 DESC
```

```
WITH
    prod_sales
    AS (SELECT
            t1.product_name
        ,   SUM(t2.unit_price * t2.quantity) product_sales
        FROM
            products t1
                INNER JOIN
            order_details t2
                ON t2.product_id = t1.product_id
                INNER JOIN
            orders t3
                ON t2.order_id = t3.order_id
        WHERE
            EXTRACT(YEAR FROM t3.order_date) = 2015
        GROUP BY
            t1.product_name)
SELECT
    (CASE WHEN ps.product_sales < 7000 THEN NULL ELSE ' ' END) S
,   (CASE WHEN ps.product_sales < 7000 THEN 'Other' ELSE ps.product_name END)
product_name
,   ROUND(SUM(ps.product_sales)/1000,2) product_sales
FROM
    prod_sales ps
GROUP BY
    (CASE WHEN ps.product_sales < 7000 THEN NULL ELSE ' ' END)
,   (CASE WHEN ps.product_sales < 7000 THEN 'Other' ELSE ps.product_name END)
ORDER BY
    1 NULLS LAST
,   3 DESC
```

Задача

Изчисляване на ТОП N

Да се класират продуктите по продадени единици в рамките на съответната категория и глобално за всички продукти независимо от категорията.

1. Класическо решение

```
--- SELF JOIN
SELECT
    ...
FROM
    customers t1
        ...
    customers t2
        ...
```

```
-- 1
SELECT
    t1.category_name
```

```
,   t2.product_name
,   SUM(t3.quantity) units_sold
FROM
    categories t1
        INNER JOIN
    products t2
        ON t2.category_id = t1.category_id
        INNER JOIN
    order_details t3
        ON t3.product_id = t2.product_id
GROUP BY
    t1.category_name
,   t2.product_name
ORDER BY
    1
,   3 DESC
```

```
-- 2
WITH
    prod_sales
    AS (SELECT
            t1.category_name
        ,   t2.product_name
        ,   SUM(t3.quantity) units_sold
        FROM
            categories t1
                INNER JOIN
            products t2
                ON t2.category_id = t1.category_id
                INNER JOIN
            order_details t3
                ON t3.product_id = t2.product_id
        GROUP BY
            t1.category_name
        ,   t2.product_name)
SELECT
    ps.category_name
,   ps.product_name
,   ps.units_sold
,   (SELECT
        COUNT(*) + 1
    FROM
        prod_sales ns
    WHERE
        ns.units_sold > ps.units_sold
            AND
        ns.category_name = ps.category_name
    ) cat_rank
,   (SELECT
        COUNT(*) + 1
    FROM
        prod_sales ns
    WHERE
        ns.units_sold > ps.units_sold
    ) glob_rank
FROM
    prod_sales ps
ORDER BY
    5
```

2. С аналитична функция

```
WITH
    prod_sales
    AS (SELECT
            t1.category_name
        ,   t2.product_name
        ,   SUM(t3.quantity) units_sold
        FROM
            categories t1
                INNER JOIN
            products t2
                ON t2.category_id = t1.category_id
                INNER JOIN
            order_details t3
                ON t3.product_id = t2.product_id
        GROUP BY
            t1.category_name
        ,   t2.product_name)
SELECT
    ps.category_name
,   ps.product_name
,   ps.units_sold
,   RANK()
        OVER(
            PARTITION BY ps.category_name
            ORDER BY ps.units_sold DESC
        ) cat_rank
,   RANK()
        OVER(
            ORDER BY ps.units_sold DESC
        ) glob_rank
FROM
    prod_sales ps
```

```
WITH
    prod_sales
    AS (SELECT
            t1.category_name
        ,   t2.product_name
        ,   SUM(t3.quantity) units_sold
        FROM
```

```
            categories t1
                INNER JOIN
            products t2
                ON t2.category_id = t1.category_id
                INNER JOIN
            order_details t3
                ON t3.product_id = t2.product_id
        GROUP BY
            t1.category_name
        ,   t2.product_name)
,   prod_ranks
    AS (SELECT
            ps.category_name
        ,   ps.product_name
        ,   ps.units_sold
        ,   RANK()
                OVER(
                    PARTITION BY ps.category_name
                    ORDER BY ps.units_sold DESC
                ) cat_rank
        ,   RANK()
                OVER(
                    ORDER BY ps.units_sold DESC
                ) glob_rank
        FROM
            prod_sales ps)
SELECT
    pr.category_name
,   pr.product_name
,   pr.units_sold
,   pr.cat_rank
,   pr.glob_rank
FROM
    prod_ranks pr
WHERE
    pr.cat_rank <= 3
```

```
WITH
    prod_sales
    AS (SELECT
            t1.category_name
        ,   t2.product_name
        ,   SUM(t3.quantity) units_sold
        FROM
            categories t1
                INNER JOIN
            products t2
                ON t2.category_id = t1.category_id
                INNER JOIN
            order_details t3
                ON t3.product_id = t2.product_id
        GROUP BY
```

```
            t1.category_name
        ,   t2.product_name)
SELECT
    ps.category_name
,   ps.product_name
,   ps.units_sold
,   DENSE_RANK()
        OVER(
            PARTITION BY ps.category_name
            ORDER BY ps.units_sold DESC
        ) cat_rank
,   DENSE_RANK()
        OVER(
            ORDER BY ps.units_sold DESC
        ) glob_rank
FROM
    prod_sales ps
```

Задача

Кои продукти формират X % от продажбите по стойност за 2015г. (кумулативна диаграма).

```
WITH
    sales
    AS (SELECT
            t1.product_name
        ,   ROUND(SUM(t2.unit_price * t2.quantity)/1000,2) product_sales
        FROM
            products t1
                INNER JOIN
            order_details t2
                ON t2.product_id = t1.product_id
                INNER JOIN
            orders t3
                ON t2.order_id = t3.order_id
        WHERE
            EXTRACT( YEAR FROM t3.order_date) = 2015
        GROUP BY
            t1.product_name)
,   summary
    AS (SELECT
            s.product_name
        ,   s.product_sales
        ,   SUM(s.product_sales)
                OVER(
                    ORDER BY s.product_sales DESC
                    ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
                ) cum_sales
        ,   SUM(s.product_sales)
                OVER(
                    ORDER BY s.product_sales DESC
                    ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
                ) total_sales
        FROM
            sales s)
SELECT
    sm.product_name
,   sm.product_sales
,   sm.cum_sales
,   ROUND(sm.cum_sales / sm.total_sales * 100,2) cum_perc
FROM
    summary sm
```

общо: 658.39

40%: 263,35

RANGE vs ROWS

```
WITH
    sales
    AS (SELECT
            t1.product_name
        ,   ROUND(SUM(t2.unit_price * t2.quantity)/1000) product_sales
        FROM
            products t1
                INNER JOIN
            order_details t2
                ON t2.product_id = t1.product_id
                INNER JOIN
            orders t3
                ON t2.order_id = t3.order_id
        WHERE
            EXTRACT( YEAR FROM t3.order_date) = 2015
        GROUP BY
            t1.product_name)
,   summary
    AS (SELECT
            s.product_name
        ,   s.product_sales
        ,   SUM(s.product_sales)
                OVER(
                    ORDER BY s.product_sales DESC
                    RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
                ) cum_sales
        ,   SUM(s.product_sales)
                OVER(
                    ORDER BY s.product_sales DESC
                    RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
                ) total_sales
        FROM
            sales s)
SELECT
    sm.product_name
,   sm.product_sales
,   sm.cum_sales
,   ROUND(sm.cum_sales / sm.total_sales * 100,2) cum_perc
FROM
    summary sm
```