

Міністерство науки і освіти, молоді та спорту України
Національний технічний університет України
«Київський політехнічний інститут»
Кафедра АСОІУ

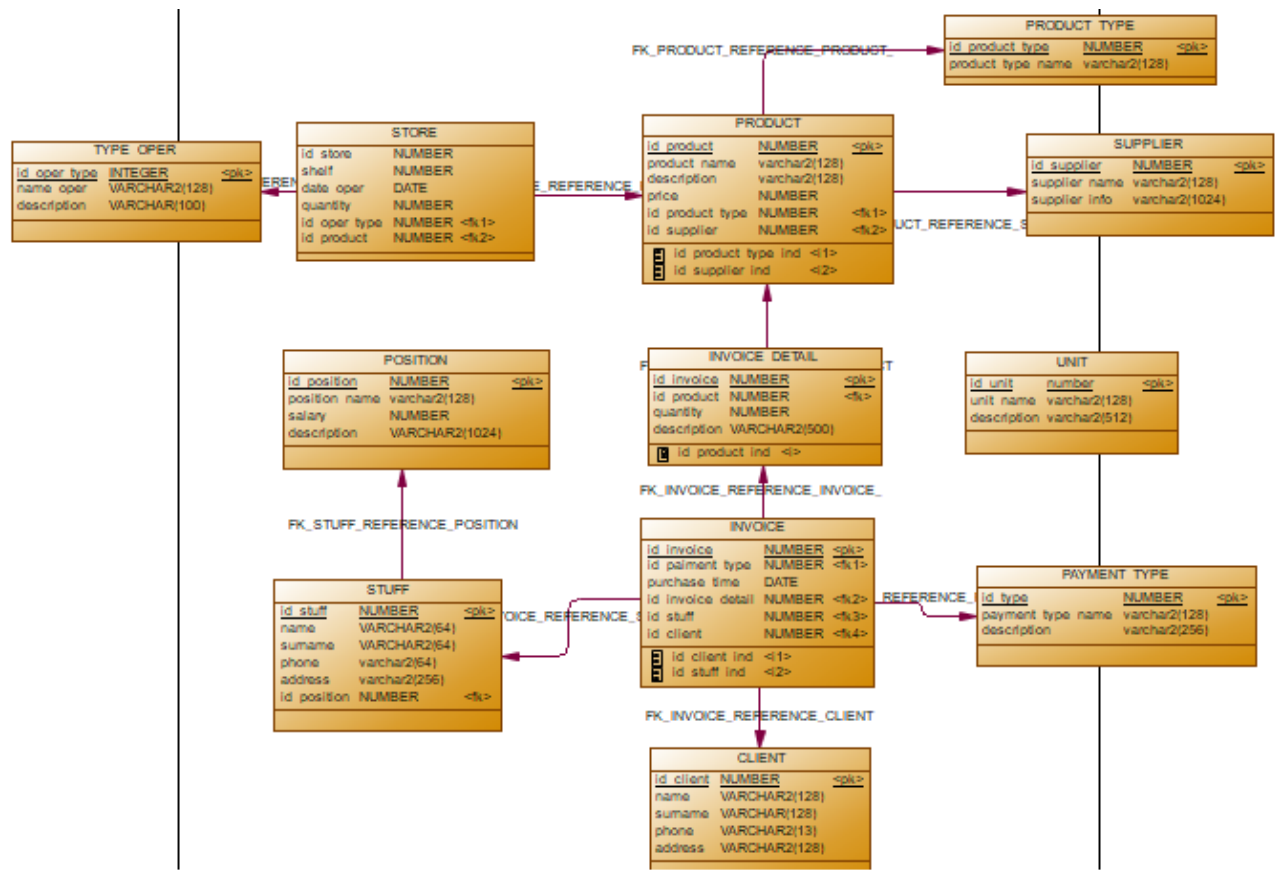
ЗВІТ
Про виконання лабораторної роботи №2
З дисципліни
«OLAP та сховища даних»

Прийняв:
Олійник Ю.О.

Виконав:
студент 3-го курсу
Гр. ІІІ-52 ФІОТ
Набоков Едуард

Київ 2018

Модель бази даних:



Запити:

1. Вивести сумарну вартість поставки по кожному продукту.

```
SELECT p."product_name", SUM(inv_d."quantity" * p."price") TOTAL_PRICE FROM
invoice inv
JOIN invoice_detail inv_d ON
inv."id_invoice_detail" = inv_d."id_invoice"
JOIN product p ON
p."id_product" = inv_d."id_product"
GROUP BY p."product_name";
```

Screenshot of the PostgreSQL Query Builder interface:

- Query Builder Tab:** Shows a SQL query for calculating total prices by product name.

```
-- 1. Вивести сумарну вартість поставки по кожному продукту.
SELECT p."product_name", SUM(inv_d."quantity" * p."price") TOTAL_PRICE FROM invoice inv
JOIN invoice_detail inv_d ON
inv."id_invoice_detail" = inv_d."id_invoice"
JOIN product p ON
p."id_product" = inv_d."id_product"
GROUP BY p."product_name";

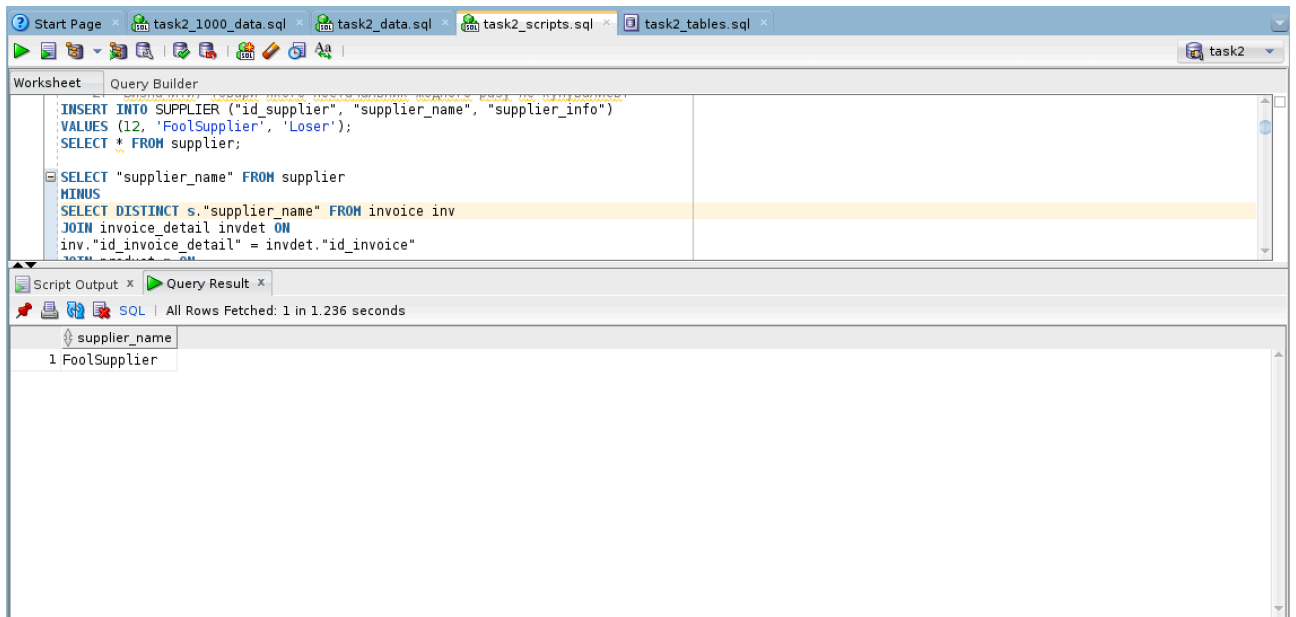
select * from product p where p."product_name" = 'Clock-2';
```

The result shows 12 rows of data:

	product_name	TOTAL_PRICE
1	Lamp-34	175606.44
2	Lamp-68	364968.73
3	Lamp-70	361371.57
4	TV-40	412992.16
5	Remote-42	38914.66
6	Clock-14	128725.96
7	TV-34	79478.86
8	TV-32	15934.16
9	Clock-7	194581.66
10	Remote-72	73368.72
11	Clock-57	51774.9
12	Clock-1	195983.54

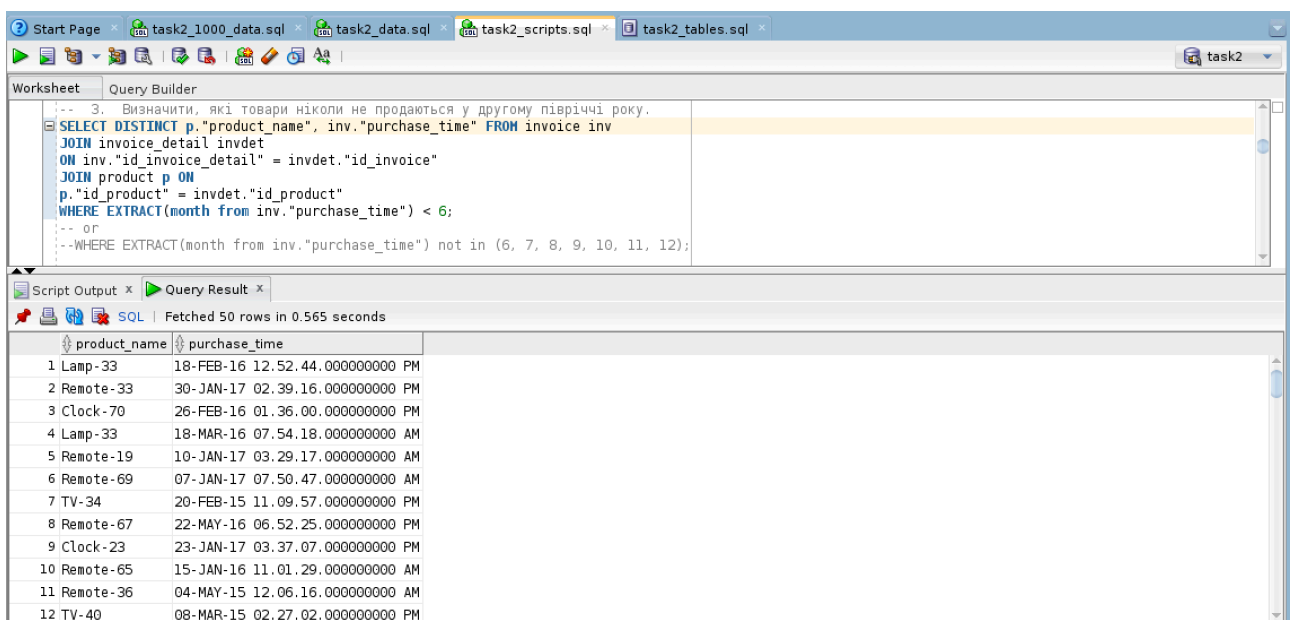
2. Визначити, товари якого постачальника жодного разу не купувались.

```
SELECT "supplier_name" FROM supplier
MINUS
SELECT DISTINCT s."supplier_name" FROM invoice inv
JOIN invoice_detail invdet ON
inv."id_invoice_detail" = invdet."id_invoice"
JOIN product p ON
p."id_product" = invdet."id_product"
JOIN supplier s ON
s."id_supplier" = p."id_supplier";
```



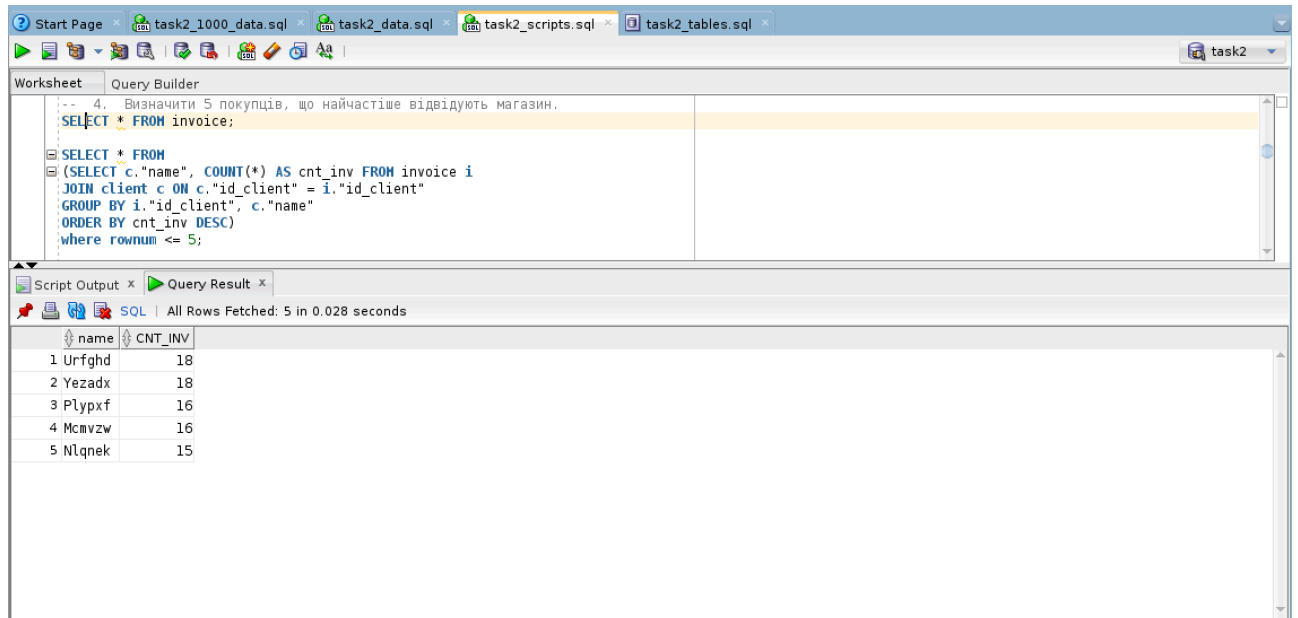
3. Визначити, які товари ніколи не продаються у другому півріччі року.

```
SELECT DISTINCT p."product_name", inv."purchase_time" FROM invoice inv
JOIN invoice_detail invdet
ON inv."id_invoice_detail" = invdet."id_invoice"
JOIN product p ON
p."id_product" = invdet."id_product"
WHERE EXTRACT(month from inv."purchase_time") < 6;
-- or
--WHERE EXTRACT(month from inv."purchase_time") not in (6, 7, 8, 9, 10, 11, 12);
```



4. Визначити 5 покупців, що найчастіше відвідують магазин.

```
SELECT * FROM
(SELECT c."name", COUNT(*) AS cnt_inv FROM invoice i
JOIN client c ON c."id_client" = i."id_client"
GROUP BY i."id_client", c."name"
ORDER BY cnt_inv DESC)
where rownum <= 5;
```



The screenshot shows the Oracle SQL Developer interface. The top pane displays a SQL query in the 'Worksheet' tab. The query is designed to find the top 5 clients by the number of invoices they have. The bottom pane shows the 'Query Result' tab with the results of the query. The results are displayed in a table with two columns: 'name' and 'CNT_INV'. The table contains five rows of data, representing the top 5 clients by invoice count.

	name	CNT_INV
1	Urfghd	18
2	Yezadx	18
3	Plypxf	16
4	Mcmvzw	16
5	Nlqnek	15

5. Визначити працівника, що продає найбільше товарів.

```
SELECT q1.* FROM
(
  SELECT DISTINCT i."id_stuff" stuff,
    SUM(idet."quantity") summa
    FROM invoice i
  JOIN invoice_detail idet ON
  i."id_invoice_detail" = idet."id_invoice"
  GROUP BY i."id_stuff"
) q1,
(
  SELECT MAX(q2.summa) max_quant FROM
  (
    SELECT DISTINCT i."id_stuff" stuff,
      SUM(idet."quantity") summa
      FROM invoice i
    JOIN invoice_detail idet ON
    i."id_invoice_detail" = idet."id_invoice"
    GROUP BY i."id_stuff"
  ) q2
) q3
WHERE q1.summa = q3.max_quant;
```

Worksheet: Query Builder

```
-- 5. Визначити працівника, що продає найбільше товарів.
SELECT q1.* FROM
(
  SELECT DISTINCT i."id_stuff" stuff,
    SUM(idet."quantity") summa
    FROM invoice i
  JOIN invoice_detail idet ON
    i."id_invoice_detail" = idet."id_invoice"
  GROUP BY i."id_stuff"
)
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.009 seconds

STUFF	SUMMA
1	1600

6. Визначити 3 працівника, що оформили найбільше чеків.

```
SELECT * FROM
(
  SELECT i."id_stuff" STUFF, COUNT(i."id_stuff") AMOUNT_RECEIPT from invoice
  i
  GROUP BY i."id_stuff"
  ORDER BY AMOUNT_RECEIPT desc
)
WHERE rownum <= 3;
```

Worksheet: Query Builder

```
-- 6. Визначити 3 працівника, що оформили найбільше чеків.
SELECT * FROM
(
  SELECT i."id_stuff" STUFF, COUNT(i."id_stuff") AMOUNT_RECEIPT from invoice i
  GROUP BY i."id_stuff"
  ORDER BY AMOUNT_RECEIPT desc
)
where rownum <= 3;
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.012 seconds

STUFF	AMOUNT_RECEIPT
1	122
2	119
3	115

7. Яка сума в середньому сплачує покупець за одне відвідування магазину.

```
SELECT client, round(sum_inv_price / cnt_inv, 2) average FROM
(
  SELECT i."id_client" AS client, COUNT(*) AS cnt_inv, SUM(p."price" *
  id."quantity") AS sum_inv_price FROM invoice i
  JOIN invoice_detail id
  ON i."id_invoice_detail" = id."id_invoice"
  JOIN product p
```

```

ON id."id_product" = p."id_product"
GROUP BY "id_client"
);

```

The screenshot shows a SQL IDE with a query in the 'Query Builder' tab. The query calculates the average invoice price for each client. The 'Script Output' and 'Query Result' tabs show the results of the query.

Query:

```

-- 7. Яка сума в середньому сплачує покупець за одне відвідування магазину.
SELECT client, round(sum_inv_price / cnt_inv, 2) average FROM
(
  SELECT i."id_client" AS client, COUNT(*) AS cnt_inv, SUM(p."price" * id."quantity") AS sum_inv_price FROM invoice i
  JOIN invoice_detail id
  ON i."id_invoice_detail" = id."id_invoice"
  JOIN product p
  ON id."id_product" = p."id_product"
  GROUP BY "id_client"
)

```

Query Result:

CLIENT	AVERAGE
1	54 32736.54
2	1 20095.54
3	83 32463.92
4	91 35027.4
5	25 23765.2
6	51 39708.25
7	100 29271.92
8	43 15855.37
9	78 30246.36
10	34 37293.32
11	30 15996.68
12	42 30790.92

8. Скільки покупок оформлює кожен продавець за день.

```

SELECT q2.stuff1, round(q2.sum_purch / q2.cnt_days) as avg_purch_amount FROM
(
  SELECT q1.stuff stuff1, COUNT(*) cnt_days, SUM(q1.cnt) sum_purch FROM
  (
    SELECT "id_stuff" stuff, COUNT(*) cnt FROM invoice
    GROUP BY "id_stuff", trunc("purchase_time")
  ) q1
  GROUP BY q1.stuff
) q2;

```

The screenshot shows a SQL IDE with a query in the 'Query Builder' tab. The query calculates the average purchase amount for each product. The 'Script Output' and 'Query Result' tabs show the results of the query.

Query:

```

-- 8. Скільки покупок оформлює кожен продавець за день
SELECT q2.stuff1, round(q2.sum_purch / q2.cnt_days) as avg_purch_amount FROM
(
  SELECT q1.stuff stuff1, COUNT(*) cnt_days, SUM(q1.cnt) sum_purch FROM
  (
    SELECT "id_stuff" stuff, COUNT(*) cnt FROM invoice
    GROUP BY "id_stuff", trunc("purchase_time")
  ) q1
  GROUP BY q1.stuff
) q2;

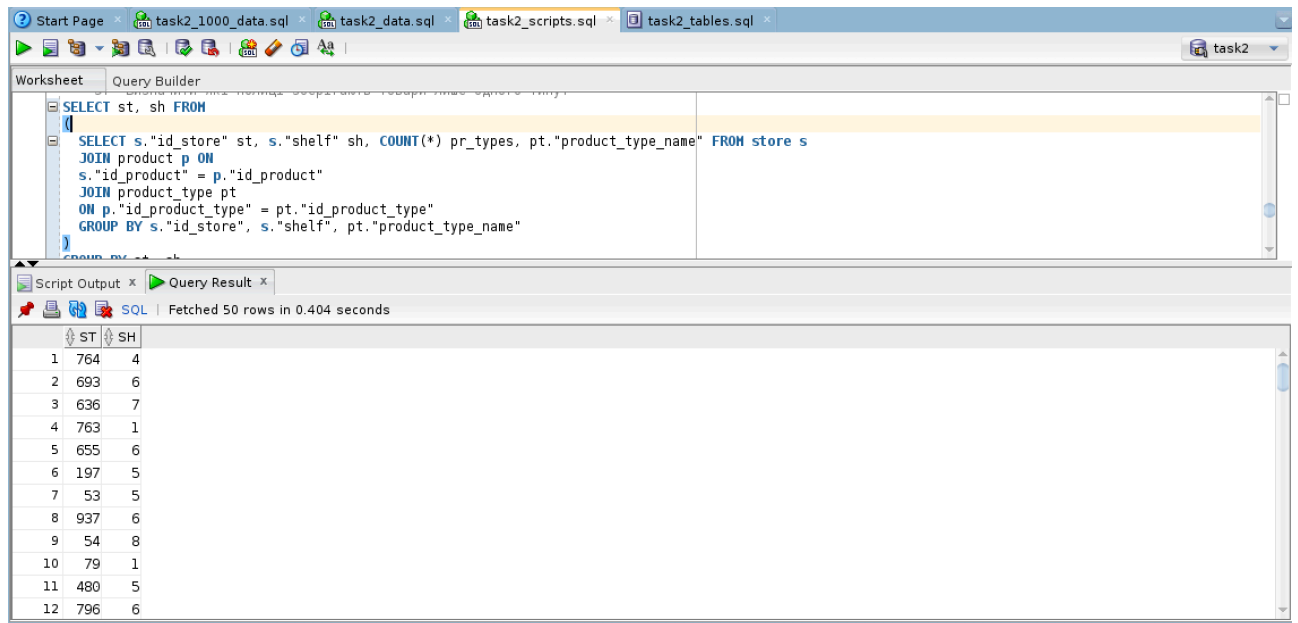
```

Query Result:

STUFF1	AVG_PURCH_AMOUNT
1	1
2	6
3	2
4	4
5	5
6	8
7	7
8	3
9	9
10	10

9. Визначити які полиці зберігають товари лише одного типу.

```
SELECT st, sh FROM
(
  SELECT s."id_store" st, s."shelf" sh, COUNT(*) pr_types,
  pt."product_type_name" FROM store s
  JOIN product p ON
  s."id_product" = p."id_product"
  JOIN product_type pt
  ON p."id_product_type" = pt."id_product_type"
  GROUP BY s."id_store", s."shelf", pt."product_type_name"
)
GROUP BY st, sh
HAVING COUNT(*) = 1;
```



The screenshot shows a database query tool interface. The 'Query Builder' tab displays the SQL query for task 9. The 'Query Result' tab shows the results of the query, which is a table with two columns: ST and SH. The results are as follows:

ST	SH
1	764
2	693
3	636
4	763
5	655
6	197
7	53
8	937
9	54
10	79
11	480
12	796

10. Визначити товари, що мають найбільшу ціну та найчастіше продаються.

```
SELECT q1.id_prod, q1.price, q1.cnt_inv FROM
(
  SELECT id."id_product" id_prod, COUNT(*) cnt_inv, p."price" price, COUNT(*)
  * p."price" mult FROM invoice i
  JOIN invoice_detail id
  ON i."id_invoice_detail" = id."id_invoice"
  JOIN product p
  ON id."id_product" = p."id_product"
  GROUP BY id."id_product", p."price"
  ORDER BY price DESC, cnt_inv
) q1,
(
  SELECT MAX(q2.mult) max_mult FROM
  (
    SELECT id."id_product" id_prod, COUNT(*) cnt_inv, p."price" price,
    COUNT(*) * p."price" mult FROM invoice i
    JOIN invoice_detail id
    ON i."id_invoice_detail" = id."id_invoice"
    JOIN product p
    ON id."id_product" = p."id_product"
    GROUP BY id."id_product", p."price"
  ) q2
) q3
WHERE q1.mult = q3.max_mult;
```

The screenshot shows a SQL IDE with several tabs: task2_1000_data.sql, task2_data.sql, task2_scripts.sql, and task2_tables.sql. The 'task2_scripts.sql' tab is active, displaying a SQL query in the Query Builder. The query is as follows:

```

GROUP BY st, sh
HAVING COUNT(*) = 1;

-- 10. Визначити товари, що мають найбільшу ціну та найчастіше продаються.
SELECT ql.id_prod, ql.price, ql.cnt_inv FROM
(
  SELECT id."id_product" id_prod, COUNT(*) cnt_inv, p."price" price, COUNT(*) * p."price" mult FROM invoice i
  JOIN invoice_detail id
  ON i."id_invoice_detail" = id."id_invoice"
)
ORDER BY mult desc

```

Below the query, the 'Query Result' pane shows the results of the query. It indicates 'All Rows Fetched: 1 in 1.797 seconds'. The result is displayed in a table with the following columns: ID_PROD, PRICE, and CNT_INV.

ID_PROD	PRICE	CNT_INV
1	449 4873.77	6

Додаткове завдання

```

SELECT * FROM (
  SELECT "id_client", purchases FROM (
    SELECT c."id_client", COUNT(*) as purchases FROM invoice i
    JOIN client c ON c."id_client" = i."id_client"
    GROUP BY i."id_client"
  )
  GROUP BY purchases, "id_client"
  HAVING purchases in (
    SELECT * FROM (
      SELECT purchases FROM (
        SELECT c."id_client", COUNT(*) as purchases FROM invoice i
        JOIN client c ON c."id_client" = i."id_client"
        GROUP BY i."id_client"
      )
      GROUP BY purchases
      ORDER BY purchases desc
    )
    WHERE rownum <= 5
  )
  ORDER BY purchases desc

```


) t2

JOIN client c2 ON c2."id_client" = t2."id_client";

```
SELECT * FROM (
  SELECT "id_client", purchases FROM (
    SELECT c."id_client", COUNT(*) as purchases FROM invoice i
    JOIN client c ON c."id_client" = i."id_client"
    GROUP BY i."id_client"
  )
  GROUP BY purchases, "id_client"
  HAVING purchases in (
    SELECT * FROM (
      SELECT purchases FROM (
```

Query Result x Query Result 1 x

SQL | All Rows Fetched: 10 in 0.006 seconds

	id_client	PURCHASES	id_client_1	name	surname	phone	address
1	25	22	25	Mhxojo	Idxrccn	+380197265785	W0UGLX2C76
2	13	18	13	Koyxsh	Inquqjk	+380868165025	0FHRI6P76I
3	36	18	36	Dygdwb	Ayridix	+380168260689	6XYH1IBGVE
4	66	17	66	Ntacee	Xsbuqgn	+380365368183	03JQPX4DS1
5	38	16	38	Rlaqnh	Wmsizjd	+380348807199	HVDM5MVA25
6	42	16	42	Waomcu	Icepwsx	+380938536655	J029BDW4EK
7	51	16	51	Dgwkcj	Zrwcyyj	+380772228392	5ZIEH41NNR
8	15	15	15	Rixqzf	Oapymfd	+380495907644	G18ZKIY6KZ
9	55	15	55	Bdzbnv	Xuatwek	+380433765447	SDV6AVDLYG
10	92	15	92	Jwzmwn	Yrarncg	+380283897106	Y3M1CY6PV5

| Line 92 Column 36 | Insert | Modified | Unix/Mac: LF