

goit-rdb-hw-05

Завантажити базу даних

```
mysql -u root -p --local-infile=1 < db.sql
```

Завдання 1

Напишіть SQL запит, який буде відображати таблицю order_details та поле customer_id з таблиці orders відповідно для кожного поля запису з таблиці order_details.

```
SELECT
    od.*,
    (SELECT o.customer_id FROM orders o WHERE o.id = od.order_id) AS customer_id
FROM order_details od;
```

	id	order_id	product_id	quantity	customer_id
	1	10248	11	12	90
	2	10248	42	10	90
	3	10248	72	5	90
	4	10249	14	9	81
	5	10249	51	40	81
	6	10250	41	10	34
	7	10250	51	35	34
	8	10250	65	15	34
	9	10251	22	6	84
	10	10251	57	15	84
	11	10251	65	20	84
	12	10252	20	40	76
	13	10252	33	25	76
	14	10252	60	40	76
	15	10253	31	20	34
	16	10253	39	42	34
	17	10253	49	40	34
	18	10254	24	15	14
	19	10254	55	21	14
	20	10254	74	21	14

Завдання 2

Напишіть SQL запит, який буде відображати таблицю order_details. Відфільтруйте результати так, щоб відповідний запис із таблиці orders виконував умову shipper_id=3.

```
SELECT od.*
FROM order_details od
WHERE od.order_id IN (SELECT o.id FROM orders o WHERE o.shipper_id = 3);
```

	id	order_id	product_id	quantity
	1	10248	11	12
	2	10248	42	10
	3	10248	72	5
	21	10255	2	20
	22	10255	16	35
	23	10255	36	25
	24	10255	59	30
	27	10257	27	25
	28	10257	39	6
	29	10257	77	15
	33	10259	21	10
	34	10259	37	1
	41	10262	5	12
	42	10262	7	15
	43	10262	56	2
	44	10263	16	60
	45	10263	24	28
	46	10263	30	60
	47	10263	74	36
	48	10264	2	35

Завдання 3

Напишіть SQL запит, вкладений в операторі FROM, який буде обирати рядки з умовою `quantity>10` з таблиці `order_details`. Для отриманих даних знайдіть середнє значення поля `quantity` — групувати слід за `order_id`.

```

SELECT
    order_id,
    AVG(quantity) AS avg_quantity
FROM (
    SELECT * FROM order_details WHERE quantity > 10
) AS filtered_details
GROUP BY order_id;

```

	order_id	avg_quantity
	10248	12.0000
	10249	40.0000
	10250	25.0000
	10251	17.5000
	10252	35.0000
	10253	34.0000
	10254	19.0000
	10255	27.5000
	10256	13.5000
	10257	20.0000
	10258	57.5000
	10260	25.5000
	10261	20.0000
	10262	13.5000
	10263	46.0000
	10264	30.0000
	10265	25.0000
	10266	12.0000
	10267	45.0000
	10269	40.0000

Завдання 4

Розв'яжіть завдання 3, використовуючи оператор WITH для створення тимчасової таблиці temp.

```
WITH temp AS (  
    SELECT * FROM order_details WHERE quantity > 10  
)  
SELECT  
    order_id,  
    AVG(quantity) AS avg_quantity  
FROM temp  
GROUP BY order_id;
```

	order_id	avg_quantity
	10248	12.0000
	10249	40.0000
	10250	25.0000
	10251	17.5000
	10252	35.0000
	10253	34.0000
	10254	19.0000
	10255	27.5000
	10256	13.5000
	10257	20.0000
	10258	57.5000
	10260	25.5000
	10261	20.0000
	10262	13.5000
	10263	46.0000
	10264	30.0000
	10265	25.0000
	10266	12.0000
	10267	45.0000
	10269	40.0000

Завдання 5

Створіть функцію з двома параметрами, яка буде ділити перший параметр на другий. Обидва параметри та значення, що повертається, повинні мати тип FLOAT.

```
DROP FUNCTION IF EXISTS divide_floats;
```

```
DELIMITER //
```

```
CREATE FUNCTION divide_floats(a FLOAT, b FLOAT)
```

```
RETURNS FLOAT
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE result FLOAT;
```

```
    SET result = a / b;
```

```
    RETURN result;
```

```
END //
```

```
DELIMITER ;
```

```
-- Застосування функції до атрибута quantity з order_details
```

```
SELECT
```

```
    id,
```

```
    order_id,
```

```
    product_id,
```

```
    quantity,
```

```
    divide_floats(quantity, 2.0) AS quantity_divided_by_2
```

```
FROM order_details;
```

	id	order_id	product_id	quantity	quantity_divided_by...
	1	10248	11	12	6
	2	10248	42	10	5
	3	10248	72	5	2.5
	4	10249	14	9	4.5
	5	10249	51	40	20
	6	10250	41	10	5
	7	10250	51	35	17.5
	8	10250	65	15	7.5
	9	10251	22	6	3
	10	10251	57	15	7.5
	11	10251	65	20	10
	12	10252	20	40	20
	13	10252	33	25	12.5
	14	10252	60	40	20
	15	10253	31	20	10
	16	10253	39	42	21
	17	10253	49	40	20