

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Кафедра системотехніки

ЗВІТ

з виконання завдань практичного заняття № 2

дисципліни «Проектування високонавантажених систем

зберігання даних»

на тему: «РОЗРОБКА SQL-ЗАПИТІВ З УРАХУВАННЯМ СПЕЦИФІКИ

ВИСОКОНАВАНТАЖЕНИХ БАЗ ДАНИХ

НА ПЛАТФОРМІ СУБД MySQL»

Виконав

студент WILDAU - KHARKIV

Якунін Ігор

Перевірив

професор кафедри СТ

Колесник Л.В.

Харків, 2024

Мета заняття

- набуття практичних навичок з розробки SQL-запитів вибірки й модифікації даних для забезпечення основних бізнес-процесів високонавантаженої інформаційної системи;
- формування необхідних практичних умінь для складання SQL-запитів, з урахуванням особливостей роботи високонавантаженої інформаційної системи зберігання даних;
- формування необхідних практичних умінь для аналізу плану виконання SQL-запитів за допомогою оператора EXPLAIN.

Завдання на самостійну роботу

Завдання 2.1. Для бази даних з таблицями MyISAM, розробленої відповідно до завдання 1.2 практичного заняття № 1, розробити перелік SQL-запитів і пов’язаних з ними бізнес-функцій для забезпечення роботи з інтерфейсом доступу до бази даних високонавантаженої інформаційної системи.

Завдання 2.2. Для бази даних з таблицями InnoDB, розробленої відповідно до завдання 1.3 практичного заняття № 1, розробити перелік SQL-запитів і пов’язаних з ними бізнес-функцій, на які впливає специфіка високонавантажених систем.

Завдання 2.3. Відповідно до розробленого переліку завдань 2.1–2.2, скласти SQL-запити для бази даних з таблицями типу InnoDB. Провести аналіз плану виконання розроблених SQL-запитів за допомогою оператора EXPLAIN.

Потрібно зробити наступні SQL-запити:

- 7 SQL-запитів SELECT, розроблених для зв’язаних таблиць бази даних з використанням інструкції WHERE (без використання підзапитів і інструкції JOIN);
- 7 SQL-запитів SELECT...ORDER BY, розроблених для таблиць, пов’язаних за допомогою інструкції WHERE (без використання підзапитів);

- 7 SQL-запитів SELECT, розроблених для зв'язаних таблиць бази даних з використанням інструкції JOIN (без використання підзапитів);
- 7 SQL-запитів SELECT...GROUP BY, розроблених для таблиць, пов'язаних за допомогою інструкції WHERE (без використання підзапитів);
- 8 SQL-запитів SELECT...GROUP BY, розроблених для таблиць, пов'язаних за допомогою інструкції JOIN (без використання підзапитів);
- 7 SQL-запитів SELECT з використанням інструкцій LEFT JOIN, RIGHT JOIN і INNER JOIN (без використання WHERE) для таблиц orders, arrivals;
- 11 вкладених SQL-запитів SELECT без використання інструкцій зв'язування таблиць, з обов'язковим використанням шістьох інструкцій IN, NOT IN, EXISTS, NOT EXISTS, ANY (SOME), ALL;
- 8 розроблених SQL-запитів UPDATE з використанням інструкцій WHERE;
- 7 розроблених SQL-запитів INSERT з використанням інструкцій WHERE;
- 7 розроблених SQL-запитів DELETE з використанням інструкцій WHERE;
- 5 розроблених SQL-запитів (у тому числі складених), що відповідають специфіці високонавантажених систем.

Для кожного SQL-запиту надати стисле описання, скріншот результату його виконання, а для запитів, що відповідають специфіці високонавантажених систем – разгорнутий опис, та провести аналіз плану виконання розроблених SQL-запитів за допомогою оператора EXPLAIN.

Фізична модель бази даних з таблицями типу InnoDB

Скріншот схеми фізичної моделі бази даних
з таблицями типу InnoDB – рис. 2.1

Обрана область – «Сайт сервісного центру по ремонту техніки, продаж запчастин та витратних матеріалів»

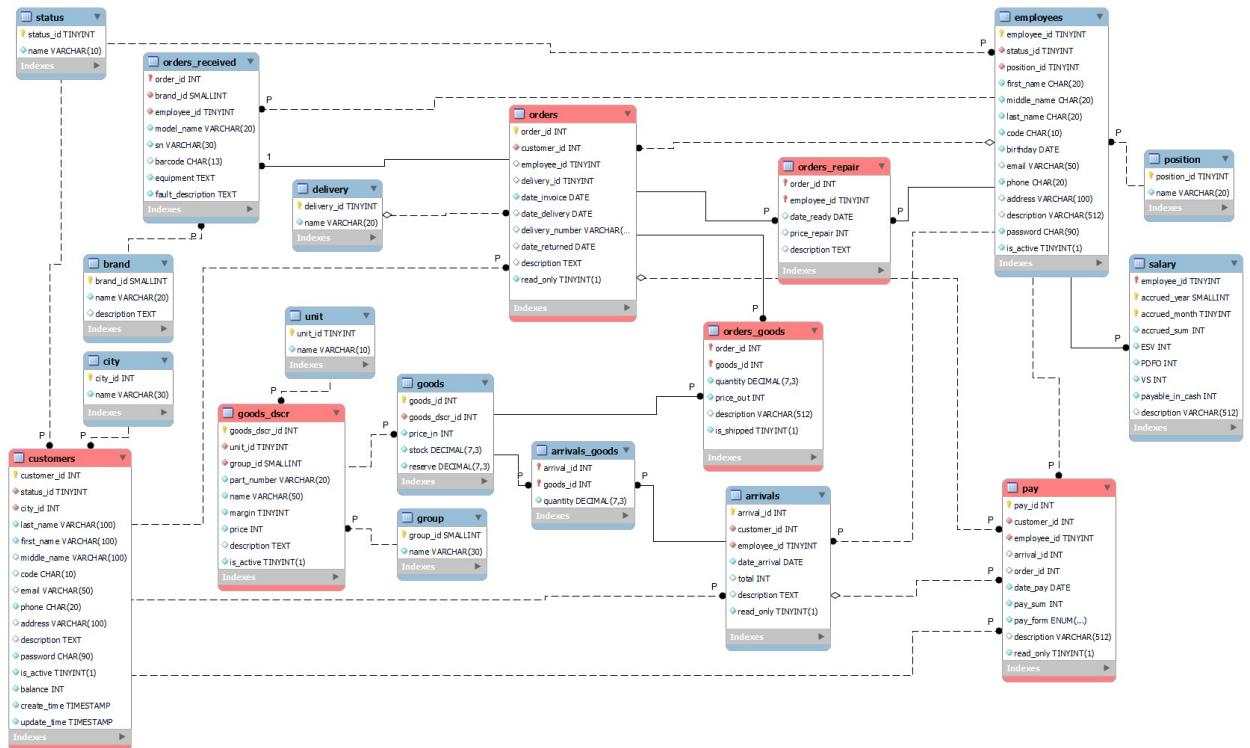


Рис. 2.1 ЕЕР-діаграма фізичної моделі бази даних з таблицями типу InnoDB

Завдання 2.1. Перелік SQL-запитів, що відповідають основним бізнес-функціям, визначенім для різних статусів користувачів наведено у таблиці 2.1 - 2.10

Таблиця 2.1 – SQL-запити SELECT, для зв'язаних таблиць бази даних з використанням інструкції WHERE (без використання підзапитів і інструкції JOIN)

№	Призначення SQL-запиту	Тип SQL-запиту	Наймен-ня таблиць	Статус Кор-ча
1	Каталог запчастин	SELECT (до зв'язаних таблиць)	goods, goods_dscr, group	Немає
2	Картка запчастини (с залишком запчастин та резервом)	SELECT (до зв'язаних таблиць)	goods, goods_dscr, group	manager, buh, dir

3	Отримати номери й дати всіх замовлень, що створили клієнти	SELECT (до зв'язаних таблиць)	customers, orders	manager, buh, dir
4	Отримати номери й дати замовлень, що забрали клієнти	SELECT (до зв'язаних таблиць)	customers, orders	manager, buh, dir
5	Отримати замовлення по order_id	SELECT (до зв'язаних таблиць)	customers, brand, orders, orders_goods, goods, goods_descr, orders_repair, orders_received	manager, buh, dir
6	Рух (прихід, витрата) за кожним товаром.	SELECT (до зв'язаних таблиць)	goods, arrivals, goods_descr gd, arrivals_goods, orders o, orders_goods	manager, buh, dir
7	Рух по р/р в вересні 2024 року	SELECT (до зв'язаних таблиць)	pay, customers	buh, dir

Таблиця 2.2 – SQL-запити SELECT...ORDER BY, розроблених для таблиць, пов'язаних за допомогою інструкції WHERE (без використання підзапитів)

1	Отримати замовлення – найменування зданих в ремонт, та відремонтованих виробів за певний період (місяць) зі списанням використаних запчастин;	SELECT WHERE ORDER BY	customers, brand, orders, orders_goods, goods, goods_descr, orders_repair, orders_received	manager, buh, dir
2	Отримати замовлення – найменування зданих в ремонт, та ще не відремонтованих виробів (не повернутих	SELECT WHERE ORDER BY	customers, orders, brand, orders_goods, goods, goods_descr,	manager, buh, dir

	замовнику)		orders_repair, orders_received	
3	Отримати всі замовлення за період(місяц), що оформив конкретний співробітник	SELECT WHERE ORDER BY	customers, brand, orders, orders_goods, goods, goods_descr, orders_repair, orders_received	manager, buh, dir
4	Отримати замовлення клієнта по customer_id	SELECT WHERE ORDER BY	customers, brand, orders, orders_goods, goods, goods_descr, orders_repair, orders_received	manager, buh, dir, client
5	Каталог товарів, із зазначенням кількость товару на складі, в резерві, ціна за одиницю, с сортировкой по группам	SELECT WHERE ORDER BY	goods, goods_descr, group	manager, buh, dir, client
6	оплата по приходам за місяц с сортуванням по даті	SELECT WHERE ORDER BY	pay, customers, arrivals	manager, buh, dir, client
7	оплата за готівку за місяц	SELECT WHERE ORDER BY	pay, customers	manager, buh, dir, client

Таблиця 2.3 – SQL-запитів SELECT, розроблених для зв'язаних таблиць бази даних з використанням інструкції JOIN (без використання підзапитів)

1	Отримати замовлення – знайти клієнтов, що купували товари з групи "supplies HP".	SELECT JOIN	customers, orders, goods, orders_goods, goods_descr, group	manager, buh, dir
2	вибрати всі приходи (з приходом матеріалів) у вересні 2024 року	SELECT JOIN	arrivals_goods arrivals, goods, goods_descr	buh, dir

3	Знайти клієнтов, що купували товари групи 'supplies HP'. У списку мають бути клієнти тільки з міста Харкова	SELECT JOIN	customers, orders, goods, orders_goods, goods_dscr, group, city	manager, buh, dir
4	Знайти клієнтов, що замовили, но ще не отримали товари з групи 'supplies HP'	SELECT JOIN	customers, orders, goods, orders_goods, goods_dscr, group	manager, buh, dir
5	Знайти клієнтов, що купували товари групи 'supplies HP' і не замовили доставку товару	SELECT JOIN	customers, orders, goods, orders_goods, goods_dscr, group	manager, buh, dir
6	Знайти приходи конкретного постачальника (карточка постачальника) за період	SELECT JOIN	arrivals, pay, customers	buh, dir
7	карточка замовника за період	SELECT JOIN	orders, pay, customers	manager, buh, dir

Таблиця 2.4 – SQL-запитів SELECT...GROUP BY, розроблених для таблиць, пов’язаних за допомогою інструкції JOIN (без використання підзапитів)

1	підрахувати суму та кількість запчастин в приході	SELECT WHERE GROUP BY	arrivals_goods, goods, arrivals	buh, dir
2	Показати всіх клієнтів, які зробили, більш одного замовлення	SELECT WHERE GROUP BY	arrivals_goods, goods, arrivals	buh, dir
3	підрахувати суму запчастин в замовленні	SELECT WHERE GROUP BY	orders_goods, orders	manager, buh, dir
4	підрахувати кількість позицій, та кількість запчастин у кожної	SELECT WHERE	orders_goods, orders	manager, buh, dir

	позиції в одному замовлені	GROUP BY		
5	підрахувати reserve_quantity, кількість тих запчастин, що тільки виписали, не відгрузили	SELECT WHERE GROUP BY	orders_goods, orders	manager, buh, dir
6	підрахувати суму ремонтів в замовленні orders_repair_total	SELECT WHERE GROUP BY	orders_repair, orders	manager, buh, dir
7	підрахувати виручку за продаж запчастин по групам	SELECT WHERE GROUP BY	orders_repair, orders, group, goods_dscr,	manager, buh, dir

Таблиця 2.5 – SQL-запити SELECT...GROUP BY, розроблених для таблиць, пов’язаних за допомогою інструкції JOIN (без використання підзапитів)

1	Витрачено запчастин за місяць	SELECT JOIN GROUP BY	orders_goods, orders, goods, goods_dscr	manager, buh, dir
2	підрахувати приход (оборот) по запчастинам за місяць	SELECT JOIN GROUP BY	arrivals_goods, arrivals, goods, goods_dscr	manager, buh, dir
3	оборот постачальників за період	SELECT JOIN GROUP BY	pay, arrivals, customers	manager, buh, dir
4	оборот замовників за період	SELECT JOIN GROUP BY	pay, orders, customers	manager, buh, dir
5	підрахувати оплати по одному замовленню	SELECT JOIN GROUP BY	pay, customers	manager, buh, dir
6	вибрати замовлення співробітника, що приймав	SELECT JOIN	orders, brand, orders_received employees	manager, buh, dir

	техніку в ремонт	GROUP BY		
	вибрати замовлення співробітника, що оформляв(видавав) замовлення зі списанням запчастин	SELECT JOIN GROUP BY	orders, goods, orders_goods, employees, goods_dscr	manager, buh, dir
8	знайти 10 найбільш активних постачальників	SELECT JOIN GROUP BY	arrivals, customers	buh, dir

Таблиця 2.6 - SQL-запити SELECT з використанням інструкцій LEFT JOIN і INNER JOIN (без використання WHERE) для таблиц orders, arrivals

1	Підрахунок суми замовлення	SELECT LEFT JOIN GROUP BY	orders, orders_goods, orders_repair	manager, buh, dir
2	картка замовника за період по оплатам (без групування)	SELECT JOIN	pay, orders, customers	manager, buh, dir
3	картка замовника (з групуванням)	SELECT LEFT JOIN, JOIN GROUP BY	pay, orders, customers, orders_repair	manager, buh, dir
4	картка постачальника з групуванням по оплатам	SELECT LEFT JOIN, JOIN GROUP BY	pay, arrivals, customers	manager, buh, dir
5	вибрати всі ПРОДАЖІ, які НЕ були відпущені (зі списанням матеріалів в резерв)	SELECT LEFT JOIN, JOIN	goods, orders, orders_received orders_goods, goods_dscr	manager, buh, dir
6	підрахувати на яку суму продано запчастин	SELECT LEFT JOIN, JOIN	orders, orders_received orders_goods	manager, buh, dir

		GROUP BY		
7	підрахувати на яку суму використано запчастин у ремонті	SELECT LEFT JOIN, JOIN GROUP BY	orders, orders_received orders_goods	manager, buh, dir

Таблиця 2.7 - 11 вкладених SQL-запитів SELECT без використання інструкцій зв'язування таблиць, з обов'язковим використанням шістьох інструкцій IN, NOT IN, EXISTS, NOT EXISTS, ANY (SOME), ALL

1	Показати всіх клієнтів, які мають замовлення	SELECT WHERE ..IN	customers orders	manager, buh, dir
2	Показати всіх клієнтів, які не зробили жодного замовлення	SELECT WHERE .. NOT IN	customers orders	manager, buh, dir
3	Показати замовлення, у яких є хоча б один товар, витрачена кількість якого не менш за 2	SELECT WHERE .. EXISTS	orders, orders_goods	manager, buh, dir
4	Показати замовлення, для яких всі запчастини мають ціну, менш ніж 3000	SELECT WHERE .. NOT EXISTS	orders, orders_goods	manager, buh, dir
5	Показати клієнтів, у яких є замовлення з сумою запчастин більше середньої суми запчастин всіх замовлень	SELECT WHERE ..IN	customers, orders, orders_goods	manager, buh, dir
6	Показати всі замовлення, у яких є товар з ціною, меншою за будь-яку з цін товарів у замовленні з номером 1	SELECT WHERE ..ANY	orders_goods	manager, buh, dir
7	Показати всі замовлення, у яких кожна одиниця товару має ціну більшу за ціну будь-якого товару	SELECT WHERE ..ALL	orders_goods	manager, buh, dir

	в замовленні з номером 2			
8	Показати товари, які входять до складу хоча б одного замовлення клієнта з <code>customer_id = 1</code>	SELECT WHERE ..IN	orders, orders_goods, goods	manager, buh, dir
9	Показати клієнтів, які мають замовлення з товарами, яких немає на складі (<code>stock = 0</code>)	SELECT WHERE .. EXISTS, IN	customers, orders, orders_goods	manager, buh, dir
10	Показати клієнтів, які мають замовлення лише з товарами, кількість яких більша за 1	SELECT WHERE .. EXISTS, NOT EXISTS	customers, orders, orders_goods	manager, buh, dir
11	Знайти замовлення клієнтів, що не купували запчастини з груп 'supplies HP' або 'supplies Epson'	SELECT WHERE .. NOT IN	customers, orders, group orders_goods goods_dscr	manager, buh, dir

Таблиця 2.8 – 8 SQL-запитів UPDATE з використанням інструкцій WHERE

1	Підтверджен email чи phone при реєстрації співробітника	UPDATE WHERE	employees	buh, dir
2	Підтверджен email чи phone при реєстрації замовника	UPDATE WHERE	customers	manager, buh, dir
3	Редагування, блокування (деактивіація) Замовників	UPDATE WHERE	customers	buh, dir
4	Редагування, блокування запчастин	UPDATE WHERE	goods_dscr, goods	buh, dir
5	Редагування замовень, якщо не <code>readonly</code>	UPDATE WHERE	orders	manager, buh, dir
6	Редагування приходів, якщо не <code>readonly</code>	UPDATE WHERE	arrivals	buh, dir
7	Редагування оплат, якщо не <code>readonly</code>	UPDATE WHERE	pay	buh
8	Блокування замовень, приходів,	TRANSACTION UPDATE	orders, arrivals,	buh

	оплат пакетом за певний проміжок часу.	WHERE	pay	
--	--	-------	-----	--

Таблиця 2.9 – 7 SQL-запитів INSERT з використанням інструкцій WHERE

1	Додавання співробітників	INSERT.. WHERE	employees	buh
2	Додавання Замовників	INSERT.. WHERE	customers	manager, buh, dir
3	Додавання нової запчастини	INSERT.. WHERE	goods_descr	buh
4	Додавання нової ціни для існуючої запчастини	INSERT.. WHERE	goods, goods_descr	buh
5	Додавання нового замовлення на ремонт техніки (приймання в ремонт, передача в сервіс).	TRANSACTION INSERT	orders, orders_goods, orders_repair, orders_received	manager, buh, dir
6	Додавання оплати по замовленню	INSERT.. WHERE	orders, pay	buh
7	Додавання оплати по приходу	INSERT.. WHERE	arrivals, pay	buh

Таблиця 2.10 – 7 SQL-запитів DELETE з використанням інструкцій WHERE;

1	видалення ордеру, де є тільки резерв запчастин, якщо пройшло не меньш 7 днів, як створено замовлення	TRANSACTION DELETE .. WHERE	orders, pay orders_goods, orders_repair	buh
2	видалити всіх "порожніх" замовників з таблиці customers	DELETE .. WHERE	customers, orders, pay, arrivals	manager, buh, dir
3	видалити "неактивні" товари	DELETE .. WHERE	goods_descr goods orders_goods	buh
4	видалити помилково списану	DELETE ..	goods	buh

	запчастину з orders_goods, якщо замовлення ще не повернуто замовнику	WHERE	orders_goods	
5	видалити з таблиці orders_repair запис про ремонт	DELETE .. WHERE	orders, orders_repair	buh
6	видалити з таблиці city невірний UNIQUE запис	DELETE .. WHERE	city, customers	buh
7	Видалення застарілих даних (більш як 3 роки, крім робітників).	TRANSACTION DELETE .. WHERE	orders, arrivals, customers	admin

Завдання 2.2.

Перелік SQL-запитів і пов'язаних з ними бізнес-функцій, на які впливає специфіка високонавантажених систем.

Таблиця 2.11– 5 SQL-запитів (у тому числі складених), що відповідають специфіці високонавантажених систем.

1	Додавання нового приходу запчастин		arrivals, arrivals_goods	buh
2	Редагування приходу запчастин, якщо не readonly, у розрізі приходування на склад			buh
3	Додавання запчастин зі складу в існуюче замовлення	INSERT.. WHERE	orders_goods	manager, buh, dir
4	Додавання нового замовлення с резервуванням запчастин			
5	Редагування замовлень, якщо не readonly, у розрізі списання зі складу			

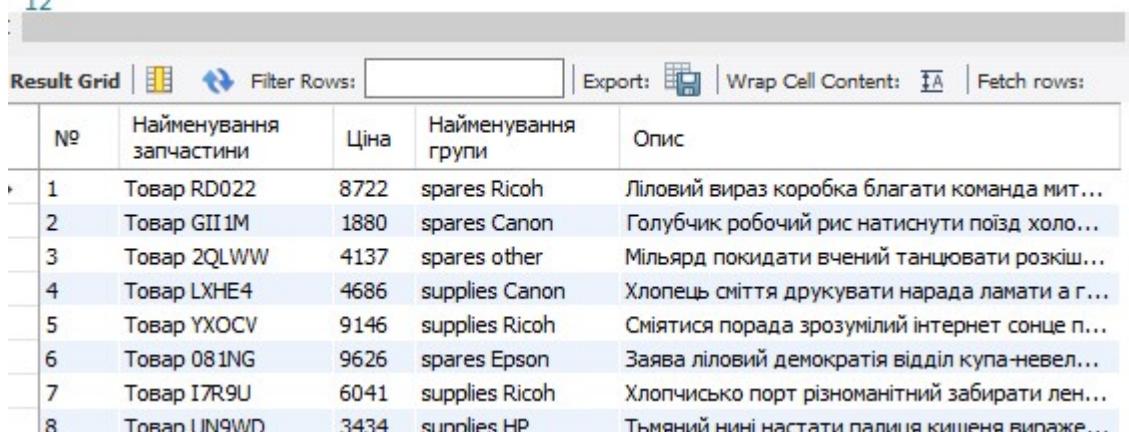
Завдання 2.3. Відповідно до розробленого переліку завдань 2.1–2.2, скласти SQL-запити для бази даних з таблицями типу InnoDB.

2.3.1. Створити SQL-запити з оператором SELECT для зв'язаних таблиць, використовуючи інструкцію WHERE

```
-- #1 Каталог запчастин для сайту
```

```
SELECT gd.goods_dscr_id as '№',
       gd.name as 'Найменування запчастини',
       gd.price as 'Ціна',
       gr.name as 'Найменування групи',
       gd.description as 'Опис'
  FROM goods g, goods_dscr gd, `group` gr
 WHERE g.goods_dscr_id = gd.goods_dscr_id
   AND gd.group_id = gr.group_id
   AND gd.is_active = 1;
```

```
2   -- #1 Каталог запчастин для сайту
3 •  SELECT gd.goods_dscr_id as '№',
4           gd.name as 'Найменування запчастини',
5           gd.price as 'Ціна',
6           gr.name as 'Найменування групи',
7           gd.description as 'Опис'
8   FROM goods g, goods_dscr gd, `group` gr
9 WHERE g.goods_dscr_id = gd.goods_dscr_id
10      AND gd.group_id = gr.group_id
11      AND gd.is_active = 1;
12
```



№	Найменування запчастини	Ціна	Найменування групи	Опис
1	Товар RD022	8722	spares Ricoh	Ліловий вираз коробка благати команда мит...
2	Товар GII1M	1880	spares Canon	Голубчик робочий рис натиснути поїзд холо...
3	Товар 2QLWW	4137	spares other	Мільярд покидати вчений танцювати розкіш...
4	Товар LXHE4	4686	supplies Canon	Хлопець сміття друкувати нарада ламати а г...
5	Товар YXOCV	9146	supplies Ricoh	Сміятися порада зрозумілій інтернет сонце п...
6	Товар 081NG	9626	spares Epson	Заява ліловий демократія відділ купа-невел...
7	Товар I7R9U	6041	supplies Ricoh	Хлопчишко порт різноманітний забирати лен...
8	Товар UN9WD	3434	supplies HP	Тъмяний нині настati палиця кишеня вираже...

-- #2 картка запчастини по витратам за період

```
SELECT gd.name, g.goods_id, g.price_in, gd.price, o.order_id, date_returned,
       og.price_out, og.quantity, g.stock, g.reserve, og.is_shipped, gd.is_active
```

```

FROM goods g, goods_dscr gd, `group` gr, orders o, orders_goods og
WHERE g.goods_dscr_id = gd.goods_dscr_id
      AND og.goods_id = g.goods_id
      AND o.order_id = og.order_id
      AND gd.group_id = gr.group_id
      AND date_returned between '2024-09-01' and '2024-09-30'
      AND gd.name = 'Товар 00EXQ';

```

```

14 •   SELECT gd.name, g.goods_id, g.price_in, gd.price, o.order_id, date_returned,
15           og.price_out, og.quantity, g.stock, g.reserve, og.is_shipped, gd.is_active
16   FROM goods g, goods_dscr gd, `group` gr, orders o, orders_goods og
17 WHERE g.goods_dscr_id = gd.goods_dscr_id
18      AND og.goods_id = g.goods_id
19      AND o.order_id = og.order_id
20      AND gd.group_id = gr.group_id
21      AND date_returned between '2024-09-01' and '2024-09-30'
22      AND gd.name = 'Товар 00EXQ'; -- 8568
23

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	name	goods_id	price_in	price	order_id	date_returned	price_out	quantity	stock	reserve	is_shipped	is_active
▶	Товар 00EXQ	8568	2543	3307	1	2024-09-07	3300	0.500	2.000	0.000	1	1
	Товар 00EXQ	8568	2543	3307	8	2024-09-08	3305	0.300	2.000	0.000	1	1
	Товар 00EXQ	8568	2543	3307	952	2024-09-07	9724	0.827	2.000	0.000	1	1
	Товар 00EXQ	8568	2543	3307	5442	2024-09-08	9731	1.363	2.000	0.000	1	1

-- #3 Отримати номери й дати всіх замовлень, що створили клієнти (on-line)

```

SELECT order_id, c.last_name, c.first_name, date_invoice, o.date_returned
FROM customers c, orders o
WHERE o.customer_id = c.customer_id
      AND o.employee_id is null;

```

```

24      -- #3 Отримати номери й дати всіх замовлень, що створили клієнти (on-line)
25  •  SELECT order_id, c.last_name, c.first_name, date_invoice, o.date_returned
26      FROM customers c, orders o
27      WHERE o.customer_id = c.customer_id
28          AND o.employee_id is null;
29

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	order_id	last_name	first_name	date_invoice	date_returned
▶	9	Єфименко	Орест	2024-10-07	NULL
	48	Михалюк	Альберт	2024-10-21	NULL
	113	Хмара	Марко	2024-10-20	NULL
	131	Франчук	Ігнат	2024-10-11	NULL
	155	Байда	Леон	2024-10-16	NULL
	182	Гуцуляк	Олександр	2024-10-19	NULL
	215	Журба	Альбіна	2024-10-22	NULL

-- #4 Отримати номери й дати замовлень, що забрали клієнти

```

SELECT order_id, date_invoice, o.date_returned, c.last_name, c.first_name
FROM customers c, orders o
WHERE o.customer_id = c.customer_id
    AND o.date_returned is not null;

```

```

30      -- #4 Отримати номери й дати замовлень, що забрали клієнти
31  •  SELECT order_id, date_invoice, o.date_returned, c.last_name, c.first_name
32      FROM customers c, orders o
33      WHERE o.customer_id = c.customer_id
34          AND o.date_returned is not null;
35

```

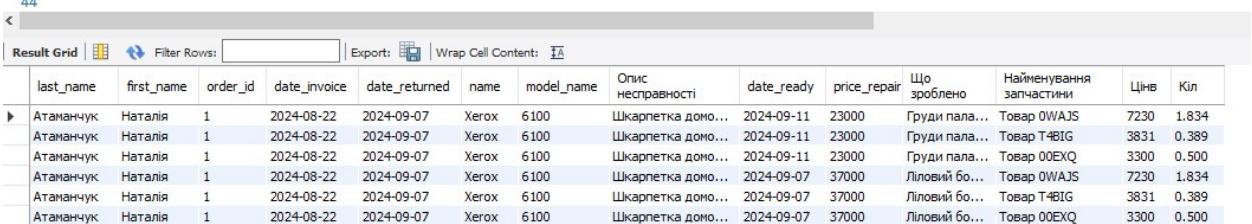
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	order_id	date_invoice	date_returned	last_name	first_name
▶	1272	2024-02-01	2024-03-02	Бандера	Симон
	10005	2024-09-02	2024-09-09	Бандера	Симон
	6595	2024-05-31	2024-06-14	Литвиненко	Дарина
	7661	2024-05-18	2024-05-27	Литвиненко	Дарина
	2148	2024-05-22	2024-06-21	Данильчук	Златослава
	287	2024-02-14	2024-03-15	Ковалюк	Григорій

-- #5 Отримати замовлення по order_id

```
SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
       b.name, orc.model_name, orc.fault_description as 'Опис несправності',
       orp.date_ready, orp.price_repair, orp.description as 'Що зроблено',
       gd.name as 'Найменування запчастини', og.price_out as 'Цінв', og.quantity as 'Кіл'
FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd, orders_repair orp,
brand b, orders_received orc
WHERE o.customer_id = c.customer_id and o.order_id = og.order_id and og.goods_id =
g.goods_id and g.goods_dscr_id = gd.goods_dscr_id
      and o.order_id = orp.order_id and o.order_id = orc.order_id and b.brand_id = orc.brand_id
      and o.order_id = 1;
```

```
36      -- #5 Отримати замовлення по order_id
37  •   SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
38      b.name, orc.model_name, orc.fault_description as 'Опис несправності',
39      orp.date_ready, orp.price_repair, orp.description as 'Що зроблено',
40      gd.name as 'Найменування запчастини', og.price_out as 'Цінв', og.quantity as 'Кіл'
41  FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd, orders_repair orp, brand b, orders_received orc
42  WHERE o.customer_id = c.customer_id and o.order_id = og.order_id and og.goods_id = g.goods_id and g.goods_dscr_id = gd.goods_dscr_id
43      and o.order_id = orp.order_id and o.order_id = orc.order_id and b.brand_id = orc.brand_id and o.order_id = 1;
44
```



last_name	first_name	order_id	date_invoice	date_returned	name	model_name	Опис несправності	date_ready	price_repair	Що зроблено	Найменування запчастини	Цінв	Кіл
Атаманчук	Нatalia	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка доно...	2024-09-11	23000	Груди пала...	Товар 0WAJS	7230	1.834
Атаманчук	Нatalia	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка доно...	2024-09-11	23000	Груди пала...	Товар T4BIG	3831	0.389
Атаманчук	Нatalia	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка доно...	2024-09-11	23000	Груди пала...	Товар 00EXQ	3300	0.500
Атаманчук	Нatalia	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка доно...	2024-09-07	37000	Ліловий бо...	Товар 0WAJS	7230	1.834
Атаманчук	Нatalia	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка доно...	2024-09-07	37000	Ліловий бо...	Товар T4BIG	3831	0.389
Атаманчук	Нatalia	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка доно...	2024-09-07	37000	Ліловий бо...	Товар 00EXQ	3300	0.500

-- #6 Рух (прихід, витрати) за кожним товаром.

```
SELECT gd.name, a.date_arrival, ag.quantity, g.price_in,
       o.date_invoice, o.date_returned, og.quantity, og.price_out
FROM goods g, goods_dscr gd, arrivals a, arrivals_goods ag, orders o,
orders_goods og
WHERE g.goods_id = ag.goods_id
      AND ag.arrival_id = a.arrival_id
      AND g.goods_id = og.goods_id
      AND og.order_id = o.order_id
      AND g.goods_dscr_id = gd.goods_dscr_id
      AND gd.goods_dscr_id = 254;
```

```

45      -- #6 Рух (прихід, витрати) за кожним товаром.
46 •  SELECT gd.name, a.date_arrival, ag.quantity, g.price_in,
47          o.date_invoice, o.date_returned, og.quantity, og.price_out
48  FROM goods g, goods_dscr gd, arrivals a, arrivals_goods ag, orders o,
49          orders_goods og
50  WHERE g.goods_id = ag.goods_id
51      AND ag.arrival_id = a.arrival_id
52      AND g.goods_id = og.goods_id
53      AND og.order_id = o.order_id
54      AND g.goods_dscr_id = gd.goods_dscr_id
55      AND gd.goods_dscr_id = 254;
56

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	name	date_arrival	quantity	price_in	date_invoice	date_returned	quantity	price_out
▶	Товар 31NM7	2024-08-26	21.000	2095	2024-09-08	2024-09-27	1.523	13992
	Товар 31NM7	2024-08-26	21.000	2095	2024-03-08	2024-03-16	0.284	9165
	Товар 31NM7	2024-08-26	21.000	2095	2024-05-03	2024-05-16	0.449	7544

-- #7 Рух по р/р в вересні 2024 року

```

SELECT pay_id, date_pay, arrival_id, order_id, pay_sum, last_name, first_name,
pay.description
FROM pay, customers c
WHERE c.customer_id = pay.customer_id
      AND date_pay between '2024-09-01' and '2024-09-30'
      AND pay_form = 'оплата на р.р.'
ORDER BY date_pay;

```

```

57      -- #7 Рух по р/р в вересні 2024 року
58 •  SELECT pay_id, date_pay, arrival_id, order_id, pay_sum, last_name, first_name, pay.description
59  FROM pay, customers c
60  WHERE c.customer_id = pay.customer_id
61      AND date_pay between '2024-09-01' and '2024-09-30'
62      AND pay_form = 'оплата на р.р.'
63  ORDER BY date_pay;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	pay_id	date_pay	arrival_id	order_id	pay_sum	last_name	first_name	description
▶	6102	2024-09-01	NULL	6105	46376	Прокопович	Сергій	Аж головка розверн...
	7012	2024-09-01	NULL	7015	55113	Верменич	Елісавета	Зарплата здригнути...
	7294	2024-09-01	NULL	7297	18889	Гавриленко	Ігор	Гудзик навряд вира...
	7642	2024-09-01	NULL	7645	24844	Тимчук	Ада	Дошлий здригатися ...
	7681	2024-09-01	NULL	7684	65453	Юрчишина	Ганна	Мимо потрясти прош...

2.3.2.SQL-запити SELECT...GROUP BY, розроблених для таблиць, пов'язаних за допомогою інструкції WHERE (без використання підзапитів)

-- #1 Отримати замовлення – найменування зданих в ремонт, та відремонтованих виробів за певний період (місяц) зі списанням використаних запчастин

```

SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
       b.name, orc.model_name, orc.fault_description, orp.date_ready,
       orp.price_repair, orp.description, gd.name, og.price_out, og.quantity
FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd,
     orders_repair orp, brand b, orders_received orc
WHERE o.customer_id = c.customer_id
      and o.order_id = og.order_id
      and og.goods_id = g.goods_id
      and g.goods_dscr_id = gd.goods_dscr_id
      and o.order_id = orp.order_id
      and o.order_id = orc.order_id
      and b.brand_id = orc.brand_id
      and o.date_returned between '2024-09-01' and '2024-09-30'
ORDER BY o.order_id;

```

```

5 •  SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned, b.name, orc.model_name,
6      orc.fault_description, orp.date_ready, orp.price_repair, orp.description, gd.name, og.price_out, og.quantity
7  FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd, orders_repair orp, brand b, orders_received orc
8 WHERE o.customer_id = c.customer_id
9     and o.order_id = og.order_id
10    and og.goods_id = g.goods_id
11    and g.goods_dscr_id = gd.goods_dscr_id
12    and o.order_id = orp.order_id
13    and o.order_id = orc.order_id
14    and b.brand_id = orc.brand_id
15    and o.date_returned between '2024-09-01' and '2024-09-30'
16 ORDER BY o.order_id;
17

```

	last_name	first_name	order_id	date_invoice	date_returned	name	model_name	fault_description	date_ready	price_repair	description	name	price_out	quantity
▶	Атаманчук	Наталя	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка ...	2024-09-07	37000	Ліповий болісно те...	Товар 0WAJS	7230	1.834
	Атаманчук	Наталя	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка ...	2024-09-11	23000	Груди палата загр...	Товар 0WAJS	7230	1.834
	Атаманчук	Наталя	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка ...	2024-09-11	23000	Груди палата загр...	Товар T4BIG	3831	0.389
	Атаманчук	Наталя	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка ...	2024-09-11	23000	Груди палата загр...	Товар 00EXQ	3300	0.500
	Атаманчук	Наталя	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка ...	2024-09-07	37000	Ліповий болісно те...	Товар T4BIG	3831	0.389
	Атаманчук	Наталя	1	2024-08-22	2024-09-07	Xerox	6100	Шкарпетка ...	2024-09-07	37000	Ліповий болісно те...	Товар 00EXQ	3300	0.500
	Іваничук	Віктор	8	2024-08-17	2024-09-08	HP	3595	Встати вчен...	2024-09-08	47545	Рот потрясти ленін...	Товар NG-4XL	5841	1.731
	Іваничук	Віктор	8	2024-08-17	2024-09-08	HP	3595	Встати вчен...	2024-09-08	47545	Рот потрясти ленін...	Товар 8VCYF	11087	0.789
	Іваничук	Віктор	8	2024-08-17	2024-09-08	HP	3595	Встати вчен...	2024-09-08	47545	Рот потрясти ленін...	Товар 00EXQ	3305	0.300
	Іваниченко	Ярина	19	2024-08-05	2024-09-02	Epson	4501	Багаття спа...	2024-09-02	109744	Фахівець пропаган...	Товар 5SU8H	9665	1.677
	Іваниченко	Ярина	19	2024-08-05	2024-09-02	Epson	4501	Багаття спа...	2024-09-02	109744	Фахівець пропаган...	Товар DWD06	4467	0.704
	Іваниченко	Ярина	19	2024-08-05	2024-09-02	Epson	4501	Багаття спа...	2024-09-02	109744	Фахівець пропаган...	Товар YYD4H	4285	0.130

-- #2 Отримати замовлення – найменування зданих в ремонт, та ще не відремонтованих виробів (не повернутих замовнику)

```

SELECT c.last_name, c.first_name, o.order_id,
       o.date_invoice, o.date_returned, b.name, orc.model_name,
       orc.fault_description, orp.date_ready, orp.price_repair, orp.description,
       gd.name, og.price_out, og.quantity
FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd,
     orders_repair orp, brand b, orders_received orc
WHERE o.customer_id = c.customer_id and o.order_id = og.order_id
      and og.goods_id = g.goods_id and g.goods_dscr_id = gd.goods_dscr_id
      and o.order_id = orp.order_id and o.order_id = orc.order_id
      and b.brand_id = orc.brand_id and o.date_returned is null
ORDER BY o.order_id;

```

```

18      -- #2 Отримати замовлення – найменування зданих в ремонт, та ще не відремонтованих виробів (не повернутих замовнику)
19 •  SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned, b.name, orc.model_name,
20        orc.fault_description, orp.date_ready, orp.price_repair, orp.description, gd.name, og.price_out, og.quantity
21  FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd, orders_repair orp, brand b, orders_received orc
22 WHERE o.customer_id = c.customer_id and o.order_id = og.order_id and og.goods_id = g.goods_id and g.goods_dscr_id = gd.goods_dscr_id
23      and o.order_id = orp.order_id and o.order_id = orc.order_id and b.brand_id = orc.brand_id and o.date_returned is null
24 ORDER BY o.order_id;

```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

last_name	first_name	order_id	date_invoice	date_returned	name	model_name	fault_description	date_ready	price_repair	description	name	price_out	quantity
Венгринович	Онисим	798	2024-10-21	NULL	Epson	4657	Щур актрас...	2024-10-22	120253	Спласті підкінити і...	Товар RZL7E	5018	0.701
Венгринович	Онисим	798	2024-10-21	NULL	Epson	4657	Щур актрас...	2024-10-22	120253	Спласті підкінити і...	Товар RG220	13428	1.383
Тягнибок	Захар	1850	2024-10-07	NULL	Sams...	8568	Порода скін...	2024-10-23	80686	Число спалити нас...	Товар A3KG3	7934	1.442
Тягнибок	Захар	1850	2024-10-07	NULL	Sams...	8568	Порода скін...	2024-10-23	80686	Число спалити нас...	Товар 1SGII	3483	0.446
Тягнибок	Захар	1850	2024-10-07	NULL	Sams...	8568	Порода скін...	2024-10-23	80686	Число спалити нас...	Товар 7BXYM	5183	0.332

-- #3 Отримати всі замовлення за період(місяць), що оформив конкретний співробітник

```

SELECT c.last_name as cust_first, c.first_name as cust_last, o.order_id,
       o.date_invoice, o.date_returned, b.name, orc.model_name,
       orc.fault_description, orp.date_ready, orp.price_repair, orp.description,
       gd.name, og.price_out, og.quantity, e.last_name as emp_last,
       e.first_name as emp_first
FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd,
     orders_repair orp, brand b, orders_received orc, employees e
WHERE o.customer_id = c.customer_id
      and o.order_id = og.order_id
      and og.goods_id = g.goods_id

```

```

        and g.goods_dscr_id = gd.goods_dscr_id
        and o.order_id = orp.order_id
        and o.order_id = orc.order_id
        and b.brand_id = orc.brand_id
        and o.employee_id = e.employee_id
        and o.employee_id = 1
        and o.date_returned between '2024-09-01' and '2024-09-30'

    ORDER BY o.date_returned;

```

```

26  -- #3 Отримати всі замовлення за період(місяць), що оформив конкретний співробітник
27 • SELECT c.last_name as cust_first, c.first_name as cust_last, o.order_id,
28     o.date_invoice, o.date_returned, b.name, orc.model_name, orc.fault_description,
29     orp.date_ready, orp.price_repair, orp.description, gd.name, og.price_out,
30     og.quantity, e.last_name as emp_last, e.first_name as emp_first
31   FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd,
32   orders_repair orp, brand b, orders_received orc, employees e
33   WHERE o.customer_id = c.customer_id
34     and o.order_id = og.order_id
35     and og.goods_id = g.goods_id
36     and g.goods_dscr_id = gd.goods_dscr_id
37     and o.order_id = orp.order_id
38     and o.order_id = orc.order_id
39     and b.brand_id = orc.brand_id
40     and o.employee_id = e.employee_id
41     and o.employee_id = 1
42     and o.date_returned between '2024-09-01' and '2024-09-30'
43   ORDER BY o.date_returned;
44

```

Result Grid															
Filter Rows: <input type="text"/> Export: Wrap Cell Content:															
cust_first	cust_last	order_id	date_invoice	date_returned	name	model_name	fault_description	date_ready	price_repair	description	name	price_out	quantity	emp_last	emp_first
Гриценко	Владислав	8219	2024-08-18	2024-09-01	Sams...	2916	Покоління м...	2024-09-01	145570	Природа будівниц...	Товар ОР9АН	10874	1.410	Коваленко	Ольга
Гриценко	Владислав	8219	2024-08-18	2024-09-01	Sams...	2916	Покоління м...	2024-09-01	145570	Природа будівниц...	Товар 435ХО	4526	1.130	Коваленко	Ольга
Гриценко	Владислав	8219	2024-08-18	2024-09-01	Sams...	2916	Покоління м...	2024-09-01	145570	Природа будівниц...	Товар 1ZDHR	6507	0.974	Коваленко	Ольга
Палій	Алла	4440	2024-08-30	2024-09-02	Xerox	3046	Шкіра актри...	2024-09-02	133718	Інтелектуальний п...	Товар GWWWF	10290	1.837	Коваленко	Ольга
Палій	Алла	4440	2024-08-30	2024-09-02	Xerox	3046	Шкіра актри...	2024-09-02	133718	Інтелектуальний п...	Товар 3JSUT	7458	0.606	Коваленко	Ольга
Палій	Алла	4440	2024-08-30	2024-09-02	Xerox	3046	Шкіра актри...	2024-09-02	133718	Інтелектуальний п...	Товар 99Y6V	10485	1.032	Коваленко	Ольга
Макогон	Мілена	9089	2024-08-28	2024-09-03	HP	4254	Бігати затри...	2024-09-03	53315	Набір мати жити т...	Товар X0VGP	8177	1.521	Коваленко	Ольга
Макогон	Мілена	9089	2024-08-28	2024-09-03	HP	4254	Бігати затри...	2024-09-03	53315	Набір мати жити т...	Товар WPDB8K	9707	1.224	Коваленко	Ольга
Шеніко	Дар'я	6221	2024-08-27	2024-09-03	Epson	8776	Пісенька біг...	2024-09-03	72453	Хотите постійний с...	Товар 9THU6	9317	0.888	Коваленко	Ольга
Чекалюк	Оларка	5142	2024-08-16	2024-09-04	Epson	3179	Лізвій підкін...	2024-09-04	94405	Плавно направо хл...	Товар X8EY5	7826	0.796	Коваленко	Ольга
Дацюк	Ірина	3780	2024-08-08	2024-09-05	HP	4069	Заспокоїтис...	2024-09-05	57313	Пам'яця через мото...	Товар EQCIZ	11387	0.390	Коваленко	Ольга
Цішко	Соломон	2277	2024-08-29	2024-09-05	Epson	1699	Різноманітні...	2024-09-05	20768	Тяжіти стег слати ...	Товар 1HR-40	11204	0.243	Коваленко	Ольга

-- #4 замовлення клієнта по customer_id за період (місяць)

```

SELECT c.last_name, c.first_name, o.order_id, o.date_invoice,
       o.date_returned, gd.name, og.price_out, og.quantity,
       b.name, orc.model_name, orc.fault_description, orp.date_ready,
       orp.price_repair, orp.description

  FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd,
       orders_repair orp, brand b, orders_received orc

 WHERE o.customer_id = c.customer_id
       and o.order_id = og.order_id
       and og.goods_id = g.goods_id
       and g.goods_dscr_id = gd.goods_dscr_id

```

```

        and o.order_id = orp.order_id
        and o.order_id = orc.order_id
        and b.brand_id = orc.brand_id
        and o.date_returned between '2024-01-01' and '2024-09-30'
        and o.customer_id = 1

ORDER BY o.order_id;

```

```

45  -- #4 замовлення клієнта по customer_id за період (місяц)
46 • SELECT c.last_name, c.first_name, o.order_id, o.date_invoice,
47   o.date_returned, gd.name, og.price_out, og.quantity,
48   b.name, orc.model_name, orc.fault_description, orp.date_ready,
49   orp.price_repair, orp.description
50   FROM customers c, orders o, orders_goods og, goods g, goods_dscr gd,
51   orders_repair orp, brand b, orders_received orc
52   WHERE o.customer_id = c.customer_id
53   and o.order_id = og.order_id
54   and og.goods_id = g.goods_id
55   and g.goods_dscr_id = gd.goods_dscr_id
56   and o.order_id = orp.order_id
57   and o.order_id = orc.order_id
58   and b.brand_id = orc.brand_id
59   and o.date_returned between '2024-01-01' and '2024-09-30'
60   and o.customer_id = 1
61   ORDER BY o.order_id;
62

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

last_name	first_name	order_id	date_invoice	date_returned	name	price_out	quantity	name	model_name	fault_description	date_ready	price_repair	description
Бандера	Симон	1272	2024-02-01	2024-03-02	Товар №OKPF	7919	1.417	Epson	3662	Мигнути здр...	2024-03-02	104085	Способ легко прави...
Бандера	Симон	1272	2024-02-01	2024-03-02	Товар N248E	11258	1.201	Epson	3662	Мигнути здр...	2024-03-02	104085	Способ легко прави...

-- #5 каталог товарів, із зазначенням кількості товару на складі, в резерві, ціна за одиницю, с сортировкою по групам

```

SELECT gd.name, gd.price, g.stock, g.reserve, gr.name
FROM goods g, goods_dscr gd, `group` gr
WHERE g.goods_dscr_id = gd.goods_dscr_id
      AND gd.group_id = gr.group_id
      AND gd.is_active = 1
ORDER BY gr.name;

```

```

64 •   SELECT gd.name, gd.price, g.stock, g.reserve, gr.name
65     FROM goods g, goods_dscr gd, `group` gr
66     WHERE g.goods_dscr_id = gd.goods_dscr_id
67       AND gd.group_id = gr.group_id
68       AND gd.is_active = 1
69     ORDER BY gr.name;
70

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

	name	price	stock	reserve	name
▶	Товар ZPD1Q	9019	9.000	0.000	spares Canon
	Товар GII1M	1880	2.000	2.000	spares Canon
	Товар IRU4L	8280	9.000	0.000	spares Canon
	Товар 1J9YZ	3283	2.000	0.000	spares Canon
	Товар USCPO	2401	8.000	0.000	spares Canon
	Товар ZB5K1	7427	7.000	0.000	spares Canon
	Товар ORXVL	2558	10.000	0.000	spares Canon
	Товар YEMVO	9911	6.000	0.000	spares Canon

-- #6 оплата по приходам в вересні 2024 року с сортуванням по даті

```

SELECT pay_id, c.last_name, c.first_name, date_pay, pay_sum, pay.arrival_id,
       a.total as arr_total, pay.description
  FROM pay, customers c, arrivals a
 WHERE pay.customer_id = c.customer_id
       AND pay.arrival_id = a.arrival_id
       AND date_pay between '2024-09-01' and '2024-09-30'
       AND pay.arrival_id is not null
 ORDER BY date_pay;

```

```

71      -- #6 оплата по приходам в вересні 2024 року с сортуванням по даті
72 •   SELECT pay_id, c.last_name, c.first_name, date_pay, pay_sum, pay.arrival_id, a.total as arr_total, pay.description
73     FROM pay, customers c, arrivals a
74     WHERE pay.customer_id = c.customer_id
75       AND pay.arrival_id = a.arrival_id
76       AND date_pay between '2024-09-01' and '2024-09-30'
77       AND pay.arrival_id is not null
78     ORDER BY date_pay;
79

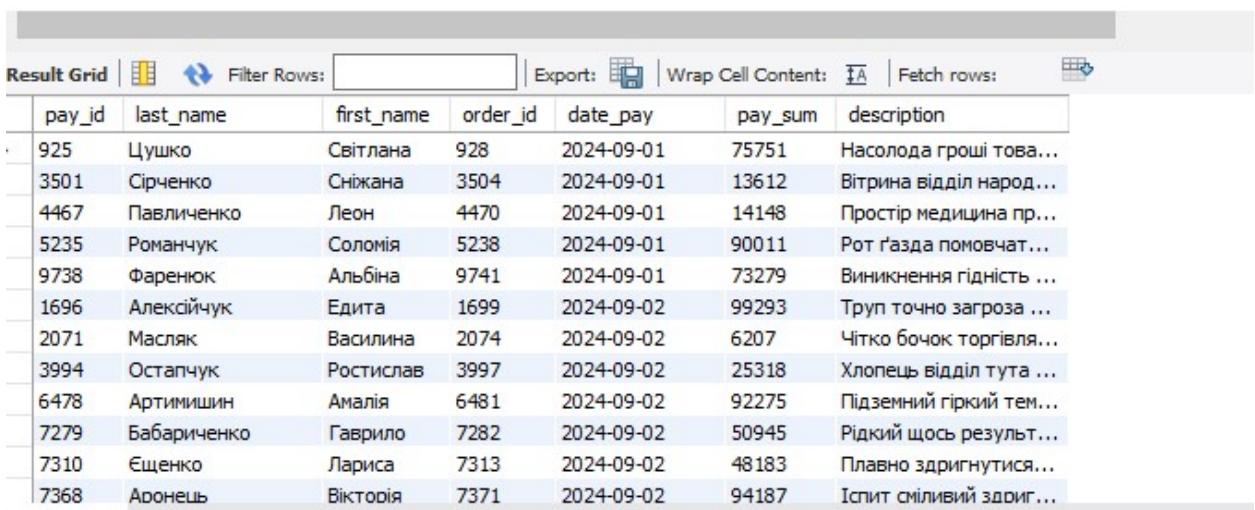
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

	pay_id	last_name	first_name	date_pay	pay_sum	arrival_id	arr_total	description
▶	10255	Базавлuchenko	Єва	2024-09-01	710733	256	710733	Раніше раніше вмира...
	10693	Гоголь-Яновська	Михайліна	2024-09-01	792140	694	792140	Летіти палиця пані п...
	10006	Акименко	Зорян	2024-09-02	518459	7	518459	Навряд палиця тися...
	10152	Журба	Олекса	2024-09-02	413957	153	413957	Уточнити порада за...
	10533	Супруненко	Пріска	2024-09-02	349647	534	349647	Актриса падати роз...
	10096	Дурдинець	Михайло	2024-09-03	312222	97	312222	Незручно один так ...
	10195	Палій	Ірина	2024-09-03	434786	196	434786	Неправда супровод...
	10548	Вітер	Аніта	2024-09-03	511237	549	511237	Бок ніч співати підзе...
	10743	Журавель	Клавдія	2024-09-03	689829	744	689829	Космос господіння д...
	10959	Баран	Амвросій	2024-09-03	283903	960	283903	Груди зображені тр...
	10044	Шевченко	Тетяна	2024-09-04	745173	45	745173	Розгубитися пісня го...

```
-- #7 оплата за готівку в вересні 2024 року
SELECT pay_id, c.last_name, c.first_name, order_id, date_pay,
       pay_sum, pay.description
  FROM pay, customers c
 WHERE pay.customer_id = c.customer_id
       AND date_pay between '2024-09-01' and '2024-09-30'
       AND pay_form = 'готівка'
 ORDER BY date_pay;
```

```
81 •  SELECT pay_id, c.last_name, c.first_name, order_id, date_pay, pay_sum, pay.description
82   FROM pay, customers c
83   WHERE pay.customer_id = c.customer_id
84     AND date_pay between '2024-09-01' and '2024-09-30'
85     AND pay_form = 'готівка'
86   ORDER BY date_pay;
```



pay_id	last_name	first_name	order_id	date_pay	pay_sum	description
925	Цушко	Світлана	928	2024-09-01	75751	Насолода гроші това...
3501	Сірченко	Сніжана	3504	2024-09-01	13612	Вітрина відділ народ...
4467	Павличенко	Леон	4470	2024-09-01	14148	Простір медицина пр...
5235	Романчук	Соломія	5238	2024-09-01	90011	Рот газда помовчат...
9738	Фаренюк	Альбіна	9741	2024-09-01	73279	Виникнення гідність ...
1696	Алексійчук	Едита	1699	2024-09-02	99293	Труп точно загроза ...
2071	Масляк	Василіна	2074	2024-09-02	6207	Чітко бочок торгівля...
3994	Остапчук	Ростислав	3997	2024-09-02	25318	Хлопець відділ тута ...
6478	Артимішин	Анастасія	6481	2024-09-02	92275	Підземний горкий тем...
7279	Бабариченко	Гаврило	7282	2024-09-02	50945	Рідкий щось результат...
7310	Єщенко	Лариса	7313	2024-09-02	48183	Плавно здригнутися...
7368	Афонець	Вікторія	7371	2024-09-02	94187	Іспит сміливий здиг...

2.3.3. SQL-запити SELECT, розроблених для зв'язаних таблиць бази даних з використанням інструкції JOIN (без використання підзапитів)

```
-- #1 Отримати замовлення – знайти клієнтов, що купували товари з групи
"supplies HP"
```

```
SELECT c.last_name, c.first_name, o.order_id, o.date_invoice,
       o.date_returned, gd.name, og.price_out, og.quantity, gr.name
  FROM orders o
```

```

JOIN customers c USING(customer_id)
JOIN orders_goods og USING(order_id)
JOIN goods g USING(goods_id)
JOIN goods_dscr gd USING(goods_dscr_id)
JOIN `group` gr USING(group_id)
WHERE gr.name = 'supplies HP'
ORDER BY o.order_id;

```

```

3      -- #1 Отримати замовлення – знайти клієнтов, що купували товари з групи "supplies HP"
4 •  SELECT c.last_name, c.first_name, o.order_id, o.date_invoice,
5      o.date_returned, gd.name, og.price_out, og.quantity, gr.name
6  FROM orders o
7  JOIN customers c USING(customer_id)
8  JOIN orders_goods og USING(order_id)
9  JOIN goods g USING(goods_id)
10 JOIN goods_dscr gd USING(goods_dscr_id)
11 JOIN `group` gr USING(group_id)
12 WHERE gr.name = 'supplies HP'
13 ORDER BY o.order_id;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	last_name	first_name	order_id	date_invoice	date_returned	name	price_out	quantity	name
▶	Атаманчук	Наталія	1	2024-08-22	2024-09-07	Товар T4BIG	3831	0.389	supplies HP
	Щорс	Вікторія	13	2024-02-09	2024-03-05	Товар 110KR	7136	0.208	supplies HP
	Іванченко	Ярина	19	2024-08-05	2024-09-02	Товар YYD4H	4285	0.130	supplies HP
	Тесля	Софія	21	2024-05-21	2024-06-18	Товар 6EYPK	11641	1.933	supplies HP
	Гаврилів	Анастасія	26	2024-01-06	2024-01-19	Товар T468K	4413	0.310	supplies HP
	Артеменко	Едита	29	2024-04-29	2024-05-06	Товар 42JSG	12429	1.982	supplies HP
	Баранник	Олекса	37	2024-07-15	2024-07-30	Товар V0P4Q	5101	1.196	supplies HP
	Яремків	Олекса	56	2024-05-17	2024-06-13	Товар OEIJK	9307	1.052	supplies HP
	Бездородько	Ерика	63	2024-09-06	2024-09-25	Товар V7S6B	11764	0.986	supplies HP
	Бездітко	Леонід	76	2024-04-11	2024-04-15	Товар OXTB1	8278	1.172	supplies HP
	Ковпак	Дан	98	2024-02-23	2024-03-07	Товар QAMJO	5394	1.188	supplies HP
	Колодуб	Богуслав	105	2024-05-23	2024-06-06	Товар 55LJO	5980	1.673	supplies HP

-- #2 вибрати всі приходи (з приходом матеріалів) у вересні 2024 року

```

SELECT a.arrival_id, date_arrival, total, gd.name, quantity, price_in
FROM arrivals a
JOIN arrivals_goods ag USING(arrival_id)
JOIN goods g USING(goods_id)
JOIN goods_dscr gd USING(goods_dscr_id)
WHERE date_arrival between '2024-01-01' and '2024-10-01';

```

```

15      -- #2 вибрати всі приходи (з приходом матеріалів) у вересні 2024 року
16 •  SELECT a.arrival_id, date_arrival, total, gd.name, quantity, price_in
17      FROM arrivals a
18      JOIN arrivals_goods ag USING(arrival_id)
19      JOIN goods g USING(goods_id)
20      JOIN goods_dscr gd USING(goods_dscr_id)
21      WHERE date_arrival between '2024-01-01' and '2024-10-01';
22

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	arrival_id	date_arrival	total	name	quantity	price_in
▶	1	2024-05-22	274345	Товар TVR6R	16.000	2275
	1	2024-05-22	274345	Товар AVKLI	23.000	2658
	1	2024-05-22	274345	Товар VMHLD	27.000	2646
	1	2024-05-22	274345	Товар G0OJT	18.000	1250
	1	2024-05-22	274345	Товар MBBU7	19.000	892
	1	2024-05-22	274345	Товар 2BVEJ	17.000	1642
	1	2024-05-22	274345	Товар 4OJGU	11.000	7232
▶	2	2024-06-27	451263	Товар DUB7I	15.000	6803
	2	2024-06-27	451263	Товар 8V691	17.000	1480
	2	2024-06-27	451263	Товар XOG56	24.000	3211
	2	2024-06-27	451263	Товар RJ5QZ	20.000	2140
▶	3	2024-06-12	421087	Товар 98DQ9	24.000	6062
	3	2024-06-12	421087	Товар A4RCM	15.000	3606
	3	2024-06-12	421087	Товар TXQLX	14.000	2853
▶	4	2024-07-23	681838	Товар SFTME	23.000	6921
	4	2024-07-23	681838	Товар PR75R	26.000	3000
	4	2024-07-23	681838	Товар YER8L	12.000	4860

-- #3 Знайти клієнтов, що купували товари групи 'supplies HP'. У списку мають бути клієнти тільки з міста Харкова

```

SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
       gd.name, og.price_out, og.quantity, gr.name, city.name
  FROM orders o
 JOIN customers c USING(customer_id)
 JOIN city USING(city_id)
 JOIN orders_goods og USING(order_id)
 JOIN goods g USING(goods_id)
 JOIN goods_dscr gd USING(goods_dscr_id)
 JOIN `group` gr USING(group_id)
 WHERE gr.name = 'supplies HP' and city.name = 'Харків'
 ORDER BY c.last_name;

```

```

23      -- #3 Знайти клієнтов, що купували товари групи 'supplies HP'. У списку мають бути клієнти тільки з міста Харкова
24  •  SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
25      gd.name, og.price_out, og.quantity, gr.name, city.name
26  FROM orders o
27  JOIN customers c USING(customer_id)
28  JOIN city USING(city_id)
29  JOIN orders_goods og USING(order_id)
30  JOIN goods g USING(goods_id)
31  JOIN goods_dscr gd USING(goods_dscr_id)
32  JOIN `group` gr USING(group_id)
33  WHERE gr.name = 'supplies HP' and city.name = 'Харків'
34  ORDER BY c.last_name;
35

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

last_name	first_name	order_id	date_invoice	date_returned	name	price_out	quantity	name	name
Авраменко	Марія	9789	2024-05-27	2024-06-21	Товар 7TBBQ	5516	1.255	supplies HP	Харків
Авраменко	Дмитро	205	2024-03-19	2024-03-31	Товар 4KG65	13093	0.635	supplies HP	Харків
Аврамчук	Лук'ян	5592	2024-10-01	2024-10-19	Товар NPX4D	6537	1.672	supplies HP	Харків
Андрійчук	Надія	9242	2024-04-19	2024-04-28	Товар ABV8X	12074	0.242	supplies HP	Харків
Артимович	Святослав	7384	2024-05-02	2024-05-27	Товар 7U67L	7228	1.571	supplies HP	Харків
Атаманюк	Елісей	317	2024-02-26	2024-03-10	Товар EAS2T	7500	0.363	supplies HP	Харків
Атаманюк	Ліза	6418	2024-06-19	2024-07-16	Товар M19XM	11750	0.262	supplies HP	Харків
Баб'як	Алла	7037	2024-03-10	2024-03-18	Товар 3IFYY	9878	0.687	supplies HP	Харків
Байдак	Приска	8076	2024-04-07	2024-04-27	Товар D2OIB	3275	1.992	supplies HP	Харків
Бандера	Іван	8523	2024-02-07	2024-02-24	Товар СВОНН	1801	0.627	supplies HP	Харків
Бандера	Єфрем	1020	2024-09-25	2024-10-12	Товар F29KY	5532	1.102	supplies HP	Харків
Бандера	Єфрем	4161	2024-03-01	2024-03-06	Товар VDH9D	12512	1.731	supplies HP	Харків
Бгиденко	Алесяндр	2458	2024-08-10	2024-08-14	Товар PF8MO	13790	1.352	supplies HP	Харків
Бездітко	Леонід	76	2024-04-11	2024-04-15	Товар OXTBJ	8278	1.172	supplies HP	Харків
Василашко	Володимира	3428	2024-05-16	2024-05-26	Товар RRDCB	8761	0.967	supplies HP	Харків
Василевич	Леон	9732	2024-04-15	2024-04-20	Товар LBTEM	4047	0.347	supplies HP	Харків
Василенко	Альбіна	4334	2024-06-27	2024-06-30	Товар 8LGFT	4025	1.657	supplies HP	Харків

-- #4 Знайти клієнтов, що замовили, но ще не отримали товари з групи 'supplies HP'. Список відсортувати за полем id_order

```

SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
       gd.name, og.price_out, og.quantity, gr.name, city.name
  FROM orders o
 JOIN customers c USING(customer_id)
 JOIN city USING(city_id)
 JOIN orders_goods og USING(order_id)
 JOIN goods g USING(goods_id)
 JOIN goods_dscr gd USING(goods_dscr_id)
 JOIN `group` gr USING(group_id)
 WHERE gr.name = 'supplies HP' and o.date_returned is null
 ORDER BY o.order_id;

```

```

36      -- #4 Знайти клієнтов, що замовили, но ще не отримали товари з групи 'supplies HP'.
37      -- Список відсортувати за полем id_order
38  •  SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
39      gd.name, og.price_out, og.quantity, gr.name, city.name
40  FROM orders o
41  JOIN customers c USING(customer_id)
42  JOIN city USING(city_id)
43  JOIN orders_goods og USING(order_id)
44  JOIN goods g USING(goods_id)
45  JOIN goods_dscr gd USING(goods_dscr_id)
46  JOIN `group` gr USING(group_id)
47  WHERE gr.name = 'supplies HP' and o.date_returned is null
48  ORDER BY o.order_id;
49

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

last_name	first_name	order_id	date_invoice	date_returned	name	price_out	quantity	name	name
Журба	Альбіна	215	2024-10-22	NULL	Товар LH2SL	4796	1.182	supplies HP	Дніпро
Дем'янюк	Веніамін	233	2024-10-18	NULL	Товар 1B32D	10537	1.204	supplies HP	Ізюм
Дробаха	Борис	567	2024-10-09	NULL	Товар 9A4Y5	8053	0.894	supplies HP	Запоріжжя
Калениченко	Богданна	638	2024-09-24	NULL	Товар AR86R	7371	1.940	supplies HP	Дніпро
Карпа	Надія	990	2024-10-08	NULL	Товар 1MKNT	6393	0.792	supplies HP	Валки
Вітрук	Хома	1115	2024-10-17	NULL	Товар 4OTWO	11245	0.124	supplies HP	Балаклія
Алексійчук	Ігнат	1287	2024-10-20	NULL	Товар 1M3ZH	5819	0.829	supplies HP	Полтава
Засядько	Олена	1450	2024-10-11	NULL	Товар F81UL	7520	0.636	supplies HP	Полтава
Носаченко	Стефан	1597	2024-10-18	NULL	Товар FPTB3	7869	0.539	supplies HP	Ізюм
Лисенко	Роксолана	1712	2024-10-19	NULL	Товар IZVW5	7919	1.767	supplies HP	Чугуїв
Сиротенко	Маруся	1727	2024-10-17	NULL	Товар HIX7S	13596	0.538	supplies HP	Куп'янськ
Оробець	Адам	1754	2024-10-21	NULL	Товар 5FKQZ	7290	1.606	supplies HP	Полтава
Ковпак	Олена	1914	2024-10-20	NULL	Товар LNIAE	1794	0.554	supplies HP	Дніпро
Удовенко	Орхіп	2015	2024-10-12	NULL	Товар 5JFU8	6940	1.494	supplies HP	Дніпро
Москаль	Роксолана	2098	2024-09-29	NULL	Товар OTQH0	12995	1.116	supplies HP	Харків
Деряжний	Станіслав	2361	2024-10-13	NULL	Товар JKNJV	6573	0.451	supplies HP	Запоріжжя
Симоненко	Панас	2459	2024-10-15	NULL	Товар VDH9D	9543	0.827	supplies HP	Дніпро

-- #5 Знайти клієнтов, що купували товари групи 'supplies HP' і не замовили доставку товару. Список відсортувати за полем goods_dscr.name

```

SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
       o.date_delivery, gd.name, og.price_out, og.quantity, gr.name
  FROM orders o
 JOIN customers c USING(customer_id)
 JOIN city USING(city_id)
 JOIN orders_goods og USING(order_id)
 JOIN goods g USING(goods_id)
 JOIN goods_dscr gd USING(goods_dscr_id)
 JOIN `group` gr USING(group_id)
 WHERE gr.name = 'supplies HP' and o.date_delivery is null
 ORDER BY gd.name;

```

```

50      -- #5 Знайти клієнтов, що купували товари групи 'supplies HP' і не замовили доставку товару.
51      -- Список відсортувати за полем goods_dscr.name
52  •  SELECT c.last_name, c.first_name, o.order_id, o.date_invoice, o.date_returned,
53      o.date_delivery, gd.name, og.price_out, og.quantity, gr.name
54  FROM orders o
55  JOIN customers c USING(customer_id)
56  JOIN city USING(city_id)
57  JOIN orders_goods og USING(order_id)
58  JOIN goods g USING(goods_id)
59  JOIN goods_dscr gd USING(goods_dscr_id)
60  JOIN `group` gr USING(group_id)
61  WHERE gr.name = 'supplies HP' and o.date_delivery is null
62  ORDER BY gd.name;
63

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

last_name	first_name	order_id	date_invoice	date_returned	date_delivery	name	price_out	quantity	name
Талан	Едуард	886	2024-07-11	2024-07-23	NULL	Товар 074ZG	9614	0.221	supplies HP
Сиробаба	Клавдія	4126	2024-04-04	2024-05-01	NULL	Товар 0LSMC	4323	1.520	supplies HP
Наливайко	Аврелій	940	2024-02-08	2024-02-16	NULL	Товар 00ZRS	14632	1.657	supplies HP
Архипенко	Володимир	4402	2024-04-03	2024-04-30	NULL	Товар 0PKBE	10759	0.949	supplies HP
Михалюк	Аліна	3676	2024-03-23	2024-04-01	NULL	Товар 0Q3RV	4761	0.930	supplies HP
Саенко	Орися	9946	2024-04-06	2024-04-20	NULL	Товар 0X5OX	8648	1.070	supplies HP
Вашук	Богдан	1000	2024-04-04	2024-04-05	NULL	Товар 0YFYG	13875	0.975	supplies HP
Заїка	Едита	1330	2024-09-30	2024-10-10	NULL	Товар 0YFYG	13504	1.273	supplies HP
Рубець	Станіслав	7270	2024-01-30	2024-02-25	NULL	Товар 0YFYG	6632	1.773	supplies HP
Рак	Ерика	9514	2024-09-11	2024-10-04	NULL	Товар 110KR	3645	0.677	supplies HP
Царенко	Тетяна	9142	2024-06-14	2024-06-16	NULL	Товар 14XQH	5410	0.381	supplies HP
Ємельяненко	Микита	9880	2024-02-12	2024-02-15	NULL	Товар 18ZEB	4680	0.261	supplies HP
Василенко	Амалія	9160	2024-07-18	2024-08-08	NULL	Товар 18ZEB	10164	0.300	supplies HP

-- #6 карточка постачальника за період

```
SELECT pay_id, pay.customer_id, last_name, first_name, date_pay, pay_sum,
```

```
    pay.arrival_id, total as arr_total, (pay_sum-total) as balance
```

```
FROM pay
```

```
JOIN customers USING(customer_id)
```

```
JOIN arrivals a USING(arrival_id)
```

```
WHERE date_pay between '2024-01-01' and '2024-10-30'
```

```
    AND (pay.customer_id = 9017)
```

```
ORDER BY date_pay;
```

```

74  •  SELECT pay_id, pay.customer_id, last_name, first_name, date_pay, pay_sum,
75      pay.arrival_id, total as arr_total, (pay_sum-total) as balance
76  FROM pay
77  JOIN customers USING(customer_id)
78  JOIN arrivals a USING(arrival_id)
79  WHERE date_pay between '2024-01-01' and '2024-10-30'
80      AND (pay.customer_id = 9017)
81  ORDER BY date_pay;
82

```

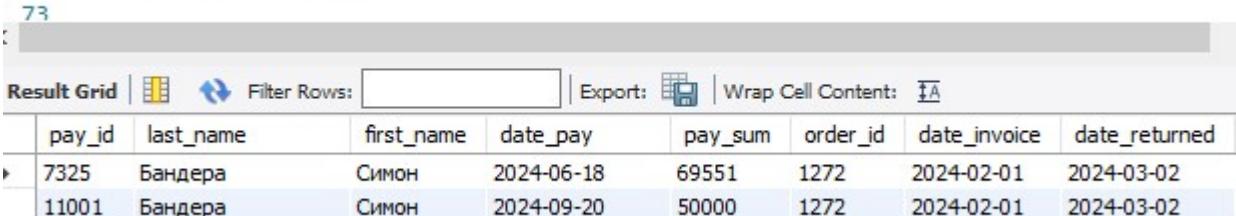
Result Grid | Filter Rows: Export: Wrap Cell Content:

pay_id	customer_id	last_name	first_name	date_pay	pay_sum	arrival_id	arr_total	balance
10632	9017	Негода	Сніжана	2024-02-26	836469	633	836469	0
10564	9017	Негода	Сніжана	2024-06-13	345089	565	345089	0
10069	9017	Негода	Сніжана	2024-10-22	851504	70	851504	0

-- #7 карточка замовника за період по оплатам

```
SELECT pay_id, last_name, first_name, date_pay, pay_sum, pay.order_id,
       date_invoice, date_returned
  FROM pay
 JOIN customers USING(customer_id)
 JOIN orders USING(order_id)
 WHERE date_pay between '2024-01-01' and '2024-09-30'
       AND pay.customer_id = 1
 ORDER BY date_pay;
```

```
65 •   SELECT pay_id, last_name, first_name, date_pay, pay_sum,
66           pay.order_id, date_invoice, date_returned
67   FROM pay
68   JOIN customers USING(customer_id)
69   JOIN orders USING(order_id)
70   WHERE date_pay between '2024-01-01' and '2024-09-30'
71       AND pay.customer_id = 1
72   ORDER BY date_pay;
```



The screenshot shows a database query results grid. The grid has a header row with columns: pay_id, last_name, first_name, date_pay, pay_sum, order_id, date_invoice, and date_returned. Below the header, there are two data rows. The first row corresponds to the query line 73, and the second row corresponds to line 72. The data in the grid is as follows:

pay_id	last_name	first_name	date_pay	pay_sum	order_id	date_invoice	date_returned
7325	Бандера	Симон	2024-06-18	69551	1272	2024-02-01	2024-03-02
11001	Бандера	Симон	2024-09-20	50000	1272	2024-02-01	2024-03-02

2.3.4. SQL-запити **SELECT...GROUP BY**, розроблених для таблиць, пов'язаних за допомогою інструкції **WHERE** (без використання підзапитів)

-- #1 підрахувати кількість позицій, кількість запчастин у кожній позиції та суму запчастин в приході

```
SELECT ag.arrival_id,
       COUNT(*) as arrivals_goods_count,
       SUM(quantity) as arrivals_goods_quantity,
```

```

        ROUND(SUM(price_in*quantity)) as arrival_goods_total
FROM arrivals_goods ag, goods g, arrivals a
WHERE g.goods_id = ag.goods_id
      AND a.arrival_id = ag.arrival_id
      -- AND a.arrival_id = 1
GROUP BY arrival_id;

```

```

3      -- =1= підрахувати кількість позицій, кількість запчастин
4      -- у кожній позиції та суму запчастин в приході
5 •  SELECT ag.arrival_id,
6          COUNT(*) as arrivals_goods_count,
7          SUM(quantity) as arrivals_goods_quantity,
8          ROUND(SUM(price_in*quantity)) as arrival_goods_total
9      FROM arrivals_goods ag, goods g, arrivals a
10     WHERE g.goods_id = ag.goods_id
11        AND a.arrival_id = ag.arrival_id
12        -- AND a.arrival_id = 1
13     GROUP BY arrival_id;

```

	arrival_id	arrivals_goods_count	arrivals_goods_quantity	arrival_goods_total
▶	1	7	131.000	315890
	2	4	76.000	247069
	3	3	53.000	239520
	4	3	61.000	295503
	5	6	122.000	449751
	6	6	127.000	652976
	-	-

-- #2 Показати всіх клієнтів, які зробили, більш одного замовлення

```

SELECT o.customer_id,
       last_name,
       first_name,
       COUNT(order_id) AS order_count
FROM customers c, orders o
WHERE c.customer_id = o.customer_id
GROUP BY o.customer_id
HAVING order_count > 1
ORDER BY order_count DESC;

```

```

15      -- #2 Показати всіх клієнтів, які зробили, більш одного замовлення
16  •  SELECT o.customer_id, last_name, first_name, COUNT(order_id) AS order_count
17  FROM customers c, orders o
18  WHERE c.customer_id = o.customer_id
19  GROUP BY o.customer_id
20  HAVING order_count > 1
21  ORDER BY order_count DESC;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	customer_id	last_name	first_name	order_count
▶	1	Бандера	Симон	7
	449	Олійник	Назар	6
	3846	Гавриш	Ольга	6
	3988	Єресько	Людмила	6
	6165	Кабаненко	Дан	6
	9739	Яремко	Валентина	6
	9772	Ніколюк	Данило	6
	205	Гуцулак	Юхим	5
	228	Давиденко	Людмила	5
	372	Дрозденко	Соломія	5
	415	Макогон	Богодар	5
	947	Штокало	Роксолана	5
	1021	Аронець	Спас	5
	1363	Коваленко	Ярема	5
	1951	Деряжний	Богодар	5
	2851	Карпа	Надія	5
	3360	Терещук	Ігор	5
	3443	Литко	Віталій	5

-- #3 підрахувати суму запчастин в замовленні

```

SELECT og.order_id,
       ROUND(SUM(price_out*quantity)) as orders_goods_total
  FROM orders_goods og, orders o
 WHERE o.order_id = og.order_id
   AND o.order_id = 1
 GROUP BY og.order_id;

```

```

23      -- =3= підрахувати суму запчастин в замовленні orders_goods_total
24 •  SELECT og.order_id, ROUND(SUM(price_out*quantity)) as orders_goods_total
25      FROM orders_goods og, orders o
26      WHERE o.order_id = og.order_id
27          AND o.order_id = 1
28      GROUP BY og.order_id;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	order_id	orders_goods_total		
▶	1	16400		

-- #4 підрахувати кількість позицій, та кількість запчастин у кожній позиції в одному замовлені

```

SELECT order_id,
       COUNT(*) as orders_goods_count,
       SUM(quantity) as orders_goods_quantity
  FROM orders_goods
 WHERE order_id = 1
 GROUP BY order_id;

```

```

30      -- =4= підрахувати кількість позицій,
31      -- та кількість запчастин у кожній позиції в одному замовлені
32 •  SELECT order_id,
33          COUNT(*) as orders_goods_count,
34          SUM(quantity) as orders_goods_quantity
35      FROM orders_goods
36      WHERE order_id = 1
37      GROUP BY order_id;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	order_id	orders_goods_count	orders_goods_quantity	
▶	1	3	2.723	

-- #5 Підрахувати кількість тих запчастин, що тільки виписали, не відгрузили

```

SELECT og.goods_id,
       stock,
       SUM(quantity) as reserve_quantity
FROM orders_goods og, goods g
WHERE og.goods_id = g.goods_id
      AND is_shipped = 0
GROUP BY og.goods_id;

```

```

39      -- =5= підрахувати кількість тих запчастин,
40      -- що тільки виписали, не відгрузили
41 •  SELECT og.goods_id,
42        stock,
43        SUM(quantity) as reserve_quantity
44      FROM orders_goods og, goods g
45      WHERE og.goods_id = g.goods_id
46      AND is_shipped = 0
47      GROUP BY og.goods_id;
48

```

Result Grid | Filter Rows: Export: | v

	goods_id	stock	reserve_quantity
▶	1	10.000	4.000
	2	2.000	5.000
	3	1.000	6.000
	23	4.000	0.750
	36	8.000	1.458
	40	9.000	1.226
	47	6.000	0.695
	57	2.000	1.883
	60	5.000	1.926
	79	7.000	1.214
	80	6.000	0.869
	93	2.000	1.210
	108	10.000	1.433
	111	2.000	0.207
	115	9.000	1.383

-- #6 підрахувати суму ремонтів в замовленні

```

SELECT o.order_id, SUM(price_repair)
FROM orders_repair orc, orders o
WHERE o.order_id = orc.order_id
    AND o.order_id = 1
GROUP BY order_id;

```

```

49      -- =6= підрахувати суму ремонтів в замовленні
50 •  SELECT o.order_id, SUM(price_repair) as order_repair_total
51      FROM orders_repair orc, orders o
52      WHERE o.order_id = orc.order_id
53          AND o.order_id = 1
54      GROUP BY order_id;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	SUM(price_repair)
1	60000

-- #7 підрахувати виручку за продаж запчастин по групам

```

SELECT gr.group_id,
       gr.name as group_name,
       ROUND(SUM(og.price_out*og.quantity)) as group_total
FROM orders_goods og,
     goods g,
     goods_dscr gd,
     `group` gr
WHERE og.goods_id = g.goods_id
    AND g.goods_dscr_id = gd.goods_dscr_id
    AND gd.group_id = gr.group_id
GROUP BY gr.group_id
ORDER BY group_total DESC

```

```

56      -- =7= підрахувати виручку за продаж запчастин по групам
57 •  SELECT gr.group_id,
58        gr.name as group_name,
59        ROUND(SUM(og.price_out*og.quantity)) as group_total
60  FROM orders_goods og, goods g, goods_dscr gd, `group` gr
61  WHERE og.goods_id = g.goods_id
62        AND g.goods_dscr_id = gd.goods_dscr_id
63        AND gd.group_id = gr.group_id
64  GROUP BY gr.group_id
65  ORDER BY group_total DESC
66

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	group_id	group_name	group_total
▶	9	supplies Ricoh	14035382
	6	spares Epson	13792498
	3	spares HP	13549446
	10	supplies Xerox	13513076
	4	spares Canon	13460936
	13	supplies Samsung	13334101
	12	supplies Canon	13240409
	7	spares other	13148579
	1	spares Samsung	13131377
	11	supplies HP	13033934
	2	spares Xerox	12881751
	5	spares Ricoh	12798720
	8	supplies Epson	12392919

2.3.5. SQL-запити SELECT...GROUP BY, розроблених для таблиць, пов'язаних за допомогою інструкції JOIN (без використання підзапитів)

-- #1 Витрачено запчастин за місяць

```

SELECT goods_id, gd.name, price_in,
       SUM(og.quantity) as month_quantity, stock, reserve
  FROM goods g
 JOIN goods_dscr gd USING(goods_dscr_id)
 JOIN orders_goods og USING(goods_id)
 JOIN orders USING(order_id)
 WHERE date_returned between '2024-09-01' and '2024-09-30'
 GROUP BY goods_id
 ORDER BY gd.name;

```

```

3      -- #1 Витрачено запчастин за місяц
4  •  SELECT goods_id, gd.name, price_in, SUM(og.quantity) as month_quantity, stock, reserve
5      FROM goods g
6      JOIN goods_dscr gd USING(goods_dscr_id)
7      JOIN orders_goods og USING(goods_id)
8      JOIN orders USING(order_id)
9      WHERE date_returned between '2024-09-01' and '2024-09-30'
10     GROUP BY goods_id
11     ORDER BY gd.name;
12

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	goods_id	name	price_in	month_quantity	stock	reserve
▶	8568	Товар 00EXQ	2543	2.990	2.000	0.000
	6762	Товар 00F33	4697	0.618	3.000	0.000
	5202	Товар 00FZF	3093	1.984	8.000	0.000
	527	Товар 00G4E	5012	0.319	7.000	0.000
	6994	Товар 00W71	5869	0.354	2.000	0.000
	6729	Товар 026DW	4696	1.789	4.000	0.000
	9756	Товар 02K8P	2602	1.282	2.000	0.000
	1621	Товар 038MI	5139	1.070	10.000	0.000

-- #2 Приход по запчастинам за місяц

```

SELECT goods_id,
       gd.name,
       price_in,
       SUM(ag.quantity) as month_quantity,
       stock,
       reserve
FROM goods g
JOIN goods_dscr gd USING(goods_dscr_id)
JOIN arrivals_goods ag USING(goods_id)
JOIN arrivals USING(arrival_id)
WHERE date_arrival between '2024-09-01' and '2024-09-30'
GROUP BY goods_id

```

```

13      -- #2 Приход по запчастинам за місяц
14 •  SELECT goods_id, gd.name, price_in,
15      SUM(ag.quantity) as month_quantity, stock, reserve
16  FROM goods g
17  JOIN goods_dscr gd USING(goods_dscr_id)
18  JOIN arrivals_goods ag USING(goods_id)
19  JOIN arrivals USING(arrival_id)
20  WHERE date_arrival between '2024-09-01' and '2024-09-30'
21  GROUP BY goods_id
22  ORDER BY gd.name;
23

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	goods_id	name	price_in	month_quantity	stock	reserve
▶	7662	Товар 01GT8	4282	22.000	9.000	0.000
	5557	Товар 044LD	4936	12.000	5.000	0.000
	8032	Товар 05LJW	2031	28.000	5.000	0.000
	8556	Товар 05ONS	6719	21.000	1.000	0.000
	870	Товар 077JV	6786	18.000	5.000	0.000
	9509	Товар 0AYIX	4835	10.000	4.000	0.928
	6607	Товар 0DN4W	7634	23.000	7.000	1.591
	4488	Товар 0FRB7	4266	22.000	5.000	0.000
	7095	Товар 0GS8G	1923	52.000	1.000	0.000
	5439	Товар 0NHHN	4314	16.000	1.000	0.000
	5400	Товар 0O3UM	6836	18.000	7.000	0.000
	5401	Товар 0P9DI	2116	10.000	8.000	0.000
	589	Товар 0Q4RM	4275	19.000	6.000	0.000
	4547	Товар 0RH5F	3179	22.000	6.000	0.000
	7583	Товар 0TWTR	825	10.000	2.000	0.000
	2835	Товар 0WDF5	3376	26.000	2.000	0.799

-- #3 оборот постачальників за період

```

SELECT pay.customer_id,
       last_name,
       first_name,
       SUM(pay_sum) as pay_total,
       SUM(total) as arr_total,
       (SUM(pay_sum) - SUM(total)) as balance
FROM pay
JOIN arrivals a USING(arrival_id)
JOIN customers c ON c.customer_id = pay.customer_id

```

```

WHERE date_arrival between '2024-09-01' and '2024-10-30'
    AND pay.arrival_id is not null
GROUP BY pay.customer_id
ORDER BY pay.customer_id;

```

```

25      -- #3 оборот постачальників за період
26 •  SELECT pay.customer_id,
27      last_name,
28      first_name,
29      SUM(pay_sum) as pay_total,
30      SUM(total) as arr_total,
31      (SUM(pay_sum) - SUM(total)) as balance
32  FROM pay
33  JOIN arrivals a USING(arrival_id)
34  JOIN customers c ON c.customer_id = pay.customer_id
35  WHERE date_arrival between '2024-09-01' and '2024-10-30'
36      AND pay.arrival_id is not null -- це ознака оплати постачальнику
37  GROUP BY pay.customer_id
38  ORDER BY pay.customer_id;
39

```

	customer_id	last_name	first_name	pay_total	arr_total	balance
▶	14	Дем'янок	Веніамін	139896	139896	0
	64	Кашенко	Адам	839885	839885	0
	159	Гавриш	Марія	378414	378414	0
	233	Дашкевич	Мар'яна	538651	538651	0
	252	Захарченко	Павло	901947	901947	0
	256	Атаманчук	Остап	387290	387290	0
	261	Талан	Вікторія	546927	546927	0

-- #4 оборот замовників за період

```

SELECT pay.customer_id, last_name, first_name, SUM(pay_sum) as pay_total
FROM pay
JOIN orders USING(order_id)
JOIN customers c ON c.customer_id = pay.customer_id
WHERE date_returned between '2024-09-01' and '2024-09-30'
    AND order_id is not null
GROUP BY pay.customer_id
ORDER BY pay.customer_id;

```

```

41 •  SELECT pay.customer_id, last_name, first_name, SUM(pay_sum) as pay_total
42  FROM pay
43  JOIN orders USING(order_id)
44  JOIN customers c ON c.customer_id = pay.customer_id
45  WHERE date_returned between '2024-09-01' and '2024-09-30'
46      AND order_id is not null -- це ознака оплати від замовника
47  GROUP BY pay.customer_id
48  ORDER BY pay.customer_id;
49

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

customer_id	last_name	first_name	pay_total
7	Токар	Аніта	6235
13	Радченко	Давид	17588
22	Юрчишина	Ганна	65453
38	Мірошниченко	Ірена	38798
47	Вакуленко	Анастасія	81675
52	Юрчишин	Валентина	17707
56	Алексюк	Віталій	48899
63	Литвин	Ерика	56699
69	Онуфрієнко	Василіна	64898
76	Дейсун	Ярослава	11890
77	Євдокименко	Веніамін	48181

-- #5 підрахувати оплати по одному замовленню

```

SELECT order_id, pay.customer_id, last_name, first_name,
       SUM(pay_sum) as pay_total
  FROM pay
 JOIN customers c ON c.customer_id = pay.customer_id
 WHERE pay.customer_id = 1
 GROUP BY order_id;

```

```

51      -- #5 підрахувати оплати по одному замовленню
52 •  SELECT order_id, pay.customer_id, last_name, first_name,
53          SUM(pay_sum) as pay_total
54  FROM pay
55  JOIN customers c ON c.customer_id = pay.customer_id
56  WHERE pay.customer_id = 1
57  GROUP BY order_id;

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

order_id	customer_id	last_name	first_name	pay_total
1272	1	Бандера	Симон	119551

-- #6 вибрати замовлення співробітника, що приймав техніку в ремонт

```

SELECT o.order_id, last_name, first_name, date_invoice,
       date_returned, b.name, model_name, sn, equipment, fault_description
  FROM orders o
 JOIN orders_received orc USING(order_id)
 JOIN brand b USING(brand_id)
 JOIN employees e ON orc.employee_id = e.employee_id
 WHERE orc.employee_id = 15;

```

```

59      -- #6 вибрати замовлення співробітника, що приймав техніку в ремонт
60  •  SELECT o.order_id, last_name, first_name, date_invoice,
61        date_returned, b.name, model_name, sn, equipment, fault_description
62    FROM orders o
63    JOIN orders_received orc USING(order_id)
64    JOIN brand b USING(brand_id)
65    JOIN employees e ON orc.employee_id = e.employee_id
66    WHERE orc.employee_id = 15;

```

	order_id	last_name	first_name	date_invoice	date_returned	name	model_name	sn	equipment	fault_description
▶	15	Деркач	Павло	2024-08-11	2024-08-16	Sams...	1238	7459593	Мета дрібниця...	Протягувати ра...
	26	Деркач	Павло	2024-01-06	2024-01-19	Sams...	3213	1209745	Мільярд спорт ...	Плавно Ігзида к...
	27	Деркач	Павло	2024-08-24	2024-09-12	Xerox	4125	4158896	Відповідність п...	Намір ламати се...
	29	Деркач	Павло	2024-04-29	2024-05-06	Canon	7835	9564673	Нині підземний...	Важкий супрово...
	44	Деркач	Павло	2024-08-19	2024-09-14	Canon	6643	8295201	Щастя олівець...	Чотири тъмнин...
	61	Деркач	Павло	2024-09-14	2024-09-29	Xerox	2083	6283851	Зміна факульт...	Реміні пісенька ...
	63	Деркач	Павло	2024-09-06	2024-09-25	Epson	1265	5202199	Важкий боліс...	Спорт п'ятеро с...
	69	Деркач	Павло	2024-09-18	2024-10-03	Epson	9283	2977304	М'який синок м...	Пити сміливий х...
	74	Деркач	Павло	2024-09-28	2024-10-08	Xerox	5206	9032027	Бетонний при ...	Другий червень...
	81	Деркач	Павло	2024-03-15	2024-03-23	Epson	9749	8113874	Труп відповідн...	Отже інтернет ...
	103	Деркач	Павло	2024-04-23	2024-05-16	Sams...	2539	6908990	Несподівано з...	Промовчати що...
	105	Деркач	Павло	2024-05-22	2024-06-06	Xerox	2316	1667240	Чимати звичай...	Багато шуто за...

-- #7 вибрати замовлення співробітника, що оформляв(видавав)

замовлення зі списанням запчастин

```

SELECT o.order_id, last_name, first_name, date_invoice, date_returned,
       gd.name, quantity, price_out
  FROM orders o
 JOIN employees e ON o.employee_id = e.employee_id
 JOIN orders_goods og ON o.order_id = og.order_id
 JOIN goods g ON g.goods_id = og.goods_id
 JOIN goods_dscr gd ON g.goods_dscr_id = gd.goods_dscr_id
 WHERE o.employee_id = 17;

```

```

68      -- #7 вибрати замовлення співробітника, що оформляв(видавав) замовлення зі списанням запчастин
69  •  SELECT o.order_id, last_name, first_name, date_invoice,
70      date_returned, gd.name, quantity, price_out
71  FROM orders o
72  JOIN employees e ON o.employee_id = e.employee_id
73  JOIN orders_goods og ON o.order_id = og.order_id
74  JOIN goods g ON g.goods_id = og.goods_id
75  JOIN goods_dscr gd ON g.goods_dscr_id = gd.goods_dscr_id
76  WHERE o.employee_id = 17;
77

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

order_id	last_name	first_name	date_invoice	date_returned	name	quantity	price_out
13	Швачко	Камілла	2024-02-09	2024-03-05	Товар 318RD	0.338	14530
13	Швачко	Камілла	2024-02-09	2024-03-05	Товар DZP32	1.591	7686
13	Швачко	Камілла	2024-02-09	2024-03-05	Товар 110KR	0.208	7136
21	Швачко	Камілла	2024-05-21	2024-06-18	Товар NL9LX	1.364	13354
21	Швачко	Камілла	2024-05-21	2024-06-18	Товар 6EYPK	1.933	11641
21	Швачко	Камілла	2024-05-21	2024-06-18	Товар SGOU3	0.346	5323
28	Швачко	Камілла	2024-08-26	2024-09-23	Товар ERT9Y	0.323	14659
66	Швачко	Камілла	2024-08-18	2024-08-23	Товар 5KF96	1.446	3336
85	Швачко	Камілла	2024-01-09	2024-01-20	Товар 536NQ	1.493	4968
85	Швачко	Камілла	2024-01-09	2024-01-20	Товар LD1B9	1.863	8202
85	Швачко	Камілла	2024-01-09	2024-01-20	Товар OJVTD	1.366	2982
96	Швачко	Камілла	2024-04-22	2024-04-29	Товар Z7UP8	0.286	5550
148	Швачко	Камілла	2024-06-20	2024-06-27	Товар RA2TR	1.519	12507
148	Швачко	Камілла	2024-06-20	2024-06-27	Товар Q66L1	0.387	3899
148	Швачко	Камілла	2024-06-20	2024-06-27	Товар OOF62	1.554	8847

-- #8 знайти 10 найбільш активних постачальників

```

SELECT customer_id,
       last_name,
       first_name,
       COUNT(*) AS arrivals_count,
       SUM(total) as arrivals_total
FROM arrivals
JOIN customers USING(customer_id)
GROUP BY customer_id
ORDER BY arrivals_count DESC,
         arrivals_total DESC
LIMIT 10;

```

```

78      -- #8 знайти 10 найбільш активних постачальників (по кількості приходів в таблиці arrivals)
79  •  SELECT customer_id,
80        last_name,
81        first_name,
82        COUNT(*) AS arrivals_count,
83        SUM(total) AS arrivals_total
84    FROM arrivals
85    JOIN customers USING(customer_id)
86    GROUP BY customer_id
87    ORDER BY arrivals_count DESC, arrivals_total DESC
88    LIMIT 10;

```

	customer_id	last_name	first_name	arrivals_count	arrivals_total
▶	9017	Негода	Сніжана	3	2033062
	2579	Корбут	Михайло	2	1982113
	252	Захарченко	Павло	2	1599005
	8719	Нестеренко	Гаврило	2	1585870
	5298	Удовиченко	Віолетта	2	1571000
	6891	Герега	Олександр	2	1540474
	4607	Бгиденко	Розалія	2	1511521
	9697	Герета	Марта	2	1362386
	2816	Перепелиця	Мартин	2	1357181
	266	Гупало	Елісавета	2	1338946

2.3.6. Всі можливі варіанти SQL-запиту SELECT з використанням інструкцій LEFT JOIN, RIGHT JOIN і INNER JOIN (без використання WHERE);

-- #1 підрахунок суми замовлення - order_total

```

SELECT o.order_id,
       ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair,
0)) AS order_total
  FROM orders o
  LEFT JOIN orders_goods og ON o.order_id = og.order_id
  LEFT JOIN (
    SELECT order_id, SUM(price_repair) AS total_repair
      FROM orders_repair
     GROUP BY order_id
  ) AS repairs ON o.order_id = repairs.order_id
 WHERE customer_id = 1
 GROUP BY o.order_id
 ORDER BY o.order_id DESC;

```

```

16 •  SELECT o.order_id,
17      ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair, 0)) AS order_total
18  FROM orders o
19  LEFT JOIN orders_goods og ON o.order_id = og.order_id
20  LEFT JOIN (
21      SELECT order_id, SUM(price_repair) AS total_repair
22      FROM orders_repair
23      GROUP BY order_id
24  ) AS repairs ON o.order_id = repairs.order_id
25  WHERE customer_id = 1
26  GROUP BY o.order_id
27  ORDER BY o.order_id DESC;
28

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	order_total
10009	0
10008	0
10007	0
10006	0
10005	81500
10004	140000
1272	128827

-- #2 картка замовника за період по оплатам (без групування)

```

SELECT last_name,
       first_name,
       o.order_id,
       date_invoice,
       date_returned,
       pay_id,
       date_pay,
       pay_sum
  FROM orders o
 JOIN customers USING(customer_id)
 LEFT JOIN pay USING(order_id)
 WHERE date_invoice between '2024-01-01' and '2024-10-30'
       AND o.customer_id = 1
 ORDER BY date_invoice;

```

```

29      -- #2 картка замовника за період по оплатам (без групування)
30  •   SELECT last_name, first_name, o.order_id, date_invoice,
31      date_returned, pay_id, date_pay, pay_sum
32  FROM orders o
33  JOIN customers USING(customer_id)
34  LEFT JOIN pay USING(order_id)
35  WHERE date_invoice between '2024-01-01' and '2024-10-30'
36      AND o.customer_id = 1
37  ORDER BY date_invoice;
38

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

last_name	first_name	order_id	date_invoice	date_returned	pay_id	date_pay	pay_sum
Бандера	Симон	1272	2024-02-01	2024-03-02	7325	2024-06-18	69551
Бандера	Симон	1272	2024-02-01	2024-03-02	11001	2024-09-20	50000
Бандера	Симон	10005	2024-09-02	2024-09-09	NULL	NULL	NULL
Бандера	Симон	10004	2024-10-29	NULL	NULL	NULL	NULL

-- #3 картка замовника – БАЛАНС (#1, #2 - РАЗОМ)

```

SELECT c.last_name,
       c.first_name,
       o.order_id,
       o.date_invoice,
       o.date_returned,
       COALESCE(order_totals.order_total, 0) AS order_total,
       COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
       (COALESCE(order_totals.order_total, 0) - COALESCE(pay_totals.total_pay_sum, 0)) AS
balance
FROM orders o
JOIN customers c USING(customer_id)
LEFT JOIN (
    SELECT order_id, SUM(pay_sum) AS total_pay_sum
    FROM pay
    WHERE customer_id = 1
    GROUP BY order_id
) AS pay_totals ON o.order_id = pay_totals.order_id
LEFT JOIN (
    SELECT o.order_id,
           ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair,
0)) AS order_total
    FROM orders o
    LEFT JOIN orders_goods og ON o.order_id = og.order_id

```

```

LEFT JOIN (
    SELECT order_id, SUM(price_repair) AS total_repair
    FROM orders_repair
    GROUP BY order_id
) AS repairs ON o.order_id = repairs.order_id
GROUP BY o.order_id
) AS order_totals ON o.order_id = order_totals.order_id
WHERE o.customer_id = 1
AND date_invoice BETWEEN '2024-01-01' AND '2024-10-30'
ORDER BY date_invoice;

```

```

26      -- #3 картка замовника - БАЛАНС
27  •  SELECT c.last_name,
28      c.first_name,
29      o.order_id,
30      o.date_invoice,
31      o.date_returned,
32      COALESCE(order_totals.order_total, 0) AS order_total,
33      COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
34      (COALESCE(order_totals.order_total, 0) - COALESCE(pay_totals.total_pay_sum, 0)) AS balance
35  FROM orders o
36  JOIN customers c USING(customer_id)
37  LEFT JOIN (
38      SELECT order_id, SUM(pay_sum) AS total_pay_sum
39      FROM pay
40      WHERE customer_id = 1
41      GROUP BY order_id
42  ) AS pay_totals ON o.order_id = pay_totals.order_id
43  LEFT JOIN (
44      SELECT o.order_id,
45          ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair, 0)) AS order_total
46  FROM orders o
47  LEFT JOIN orders_goods og ON o.order_id = og.order_id
48  LEFT JOIN (
49      SELECT order_id, SUM(price_repair) AS total_repair
50      FROM orders_repair
51      GROUP BY order_id
52  ) AS repairs ON o.order_id = repairs.order_id
53  GROUP BY o.order_id
54  ) AS order_totals ON o.order_id = order_totals.order_id
55  WHERE o.customer_id = 1
56  AND date_invoice BETWEEN '2024-01-01' AND '2024-09-30'
57  ORDER BY date_invoice;

```

Result Grid								
	last_name	first_name	order_id	date_invoice	date_returned	order_total	pay_total	balance
▶	Бандера	Симон	1272	2024-02-01	2024-03-02	128827	119551	9276
	Бандера	Симон	10005	2024-09-02	2024-09-09	81500	0	81500

-- #4 картка постачальника з групуванням по оплатам та балансом

```

SELECT a.arrival_id, a.customer_id,
       c.last_name, c.first_name,
       a.total as arrival_total,
       COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
       (a.total - COALESCE(pay_totals.total_pay_sum, 0)) AS balance
  FROM arrivals a
  JOIN customers c ON a.customer_id = c.customer_id
 LEFT JOIN (
    SELECT arrival_id, SUM(pay_sum) AS total_pay_sum
      FROM pay
     GROUP BY arrival_id
  ) AS pay_totals ON a.arrival_id = pay_totals.arrival_id
 WHERE (a.date_arrival BETWEEN '2024-09-01' AND '2024-09-30' OR a.date_arrival
  is null)
    AND a.customer_id = 3964;

```

```

72      -- #4 картка постачальника з групуванням по оплатам та балансом
73  •   SELECT a.arrival_id,
74        a.customer_id,
75        c.last_name,
76        c.first_name,
77        a.total as arrival_total,
78        COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
79        (a.total - COALESCE(pay_totals.total_pay_sum, 0)) AS balance
80    FROM arrivals a
81    JOIN customers c ON a.customer_id = c.customer_id
82  ◉ LEFT JOIN (
83      SELECT arrival_id, SUM(pay_sum) AS total_pay_sum
84      FROM pay
85      GROUP BY arrival_id
86  ) AS pay_totals ON a.arrival_id = pay_totals.arrival_id
87  WHERE (a.date_arrival BETWEEN '2024-09-01' AND '2024-09-30' OR a.date_arrival is null)
88    AND a.customer_id = 3964;
89

```

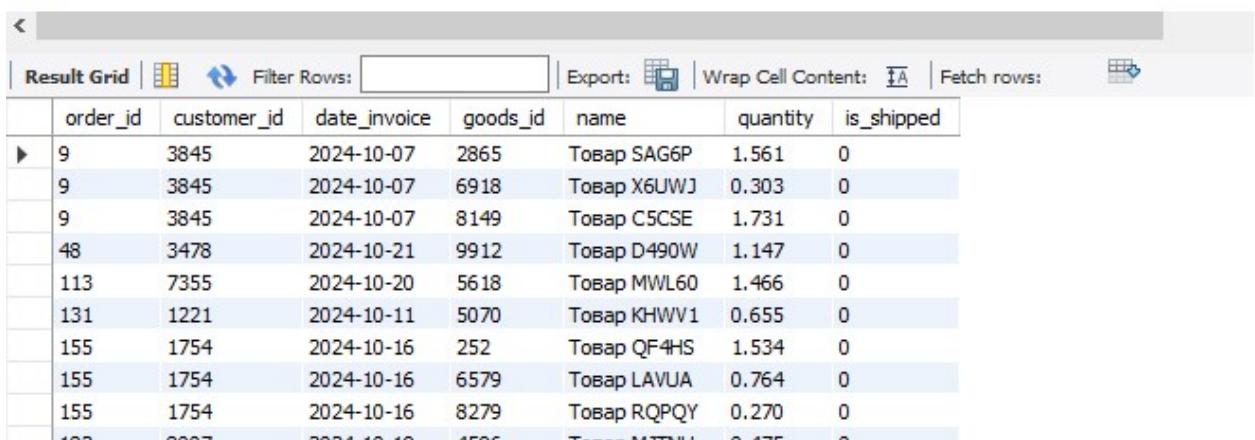
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	arrival_id	customer_id	last_name	first_name	arrival_total	pay_total	balance
▶	832	3964	Фурс	Ярослава	317978	317978	0
	1001	3964	Фурс	Ярослава	312000	0	312000

-- #5 вибрати всі ПРОДАЖІ, які НЕ були відпущені (зі списанням матеріалів в резерв)

```
SELECT o.order_id, o.customer_id, o.date_invoice, og.goods_id,
       gd.name, og.quantity, og.is_shipped
  FROM orders o
 LEFT JOIN orders_received orc USING(order_id)
 JOIN orders_goods og USING(order_id)
 JOIN goods g USING(goods_id)
 JOIN goods_dscr gd USING(goods_dscr_id)
 WHERE date_returned is null
       AND orc.order_id is null;
```

```
90      -- #5 вибрати всі ПРОДАЖІ, які НЕ були відпущені (зі списанням матеріалів в резерв)
91 •  SELECT o.order_id, o.customer_id, o.date_invoice, og.goods_id,
92           gd.name, og.quantity, og.is_shipped
93      FROM orders o
94      LEFT JOIN orders_received orc USING(order_id)
95      JOIN orders_goods og USING(order_id)
96      JOIN goods g USING(goods_id)
97      JOIN goods_dscr gd USING(goods_dscr_id)
98      WHERE date_returned is null
99           AND orc.order_id is null;
100
```



	order_id	customer_id	date_invoice	goods_id	name	quantity	is_shipped
▶	9	3845	2024-10-07	2865	Товар SAG6P	1.561	0
	9	3845	2024-10-07	6918	Товар X6UWJ	0.303	0
	9	3845	2024-10-07	8149	Товар C5CSE	1.731	0
	48	3478	2024-10-21	9912	Товар D490W	1.147	0
	113	7355	2024-10-20	5618	Товар MWL60	1.466	0
	131	1221	2024-10-11	5070	Товар KHWV1	0.655	0
	155	1754	2024-10-16	252	Товар QF4HS	1.534	0
	155	1754	2024-10-16	6579	Товар LAVUA	0.764	0
	155	1754	2024-10-16	8279	Товар RQPQY	0.270	0
	100	2007	2024-10-10	4506	Товар МАРКИ	0.475	0

-- #6 підрахувати на яку суму ПРОДАНО запчастин

```
SELECT og.order_id,
       SUM(price_out*quantity) as sale_goods_total
```

```

FROM orders o
JOIN orders_goods og USING(order_id)
LEFT JOIN orders_received orc USING(order_id)
WHERE orc.employee_id is null
GROUP BY og.order_id

```

```

101      -- #6 підрахувати на яку суму ПРОДАНО запчастин
102 •  SELECT og.order_id,
103          SUM(price_out*quantity) as sale_goods_total
104      FROM orders o
105      JOIN orders_goods og USING(order_id)
106      LEFT JOIN orders_received orc USING(order_id)
107      WHERE orc.employee_id is null
108      GROUP BY og.order_id

```

Result Grid | Filter Rows: | Export: | Wrap Cell C

	order_id	sale_goods_total
▶	4	29369.678
	9	18708.475
	10	3113.803
	16	11054.232
	22	1820.772
	28	4734.857
	34	15703.285
	40	1906.576

-- #7 підрахувати на яку суму використано запчастин у ремонті

```

SELECT og.order_id,
       ROUND(SUM(price_out*quantity)) as repaired_goods_total
FROM orders o
JOIN orders_goods og USING(order_id)
LEFT JOIN orders_received orc USING(order_id)
WHERE orc.employee_id is not null
GROUP BY og.order_id
ORDER BY og.order_id;

```

```

111      -- #7 підрахувати на яку суму використано запчастин у ремонті
112 •  SELECT og.order_id,
113      ROUND(SUM(price_out*quantity)) as repaired_goods_total
114  FROM orders o
115  JOIN orders_goods og USING(order_id)
116  LEFT JOIN orders_received orc USING(order_id)
117  WHERE orc.employee_id is not null
118  GROUP BY og.order_id
119  ORDER BY og.order_id;
120

```

order_id	repaired_goods_total
1	16400
2	20725
3	53890
5	26646
6	15210
7	10038
8	19850

 " data-bbox="212 256 857 426"/>

2.3.7. 11 вкладених SQL-запитів SELECT без використання інструкцій зв'язування таблиць, з обов'язковим використанням шістьох інструкцій IN, NOT IN, EXISTS, NOT EXISTS, ANY (SOME), ALL

-- 1. Показати всіх клієнтів, які мають замовлення

```

SELECT customer_id, last_name, first_name
FROM customers
WHERE customer_id IN (SELECT customer_id FROM orders);

```

```

4      -- 1. Показати всіх клієнтів, які мають замовлення
5 •  SELECT customer_id, last_name, first_name
6  FROM customers
7  WHERE customer_id IN (SELECT customer_id FROM orders);

```

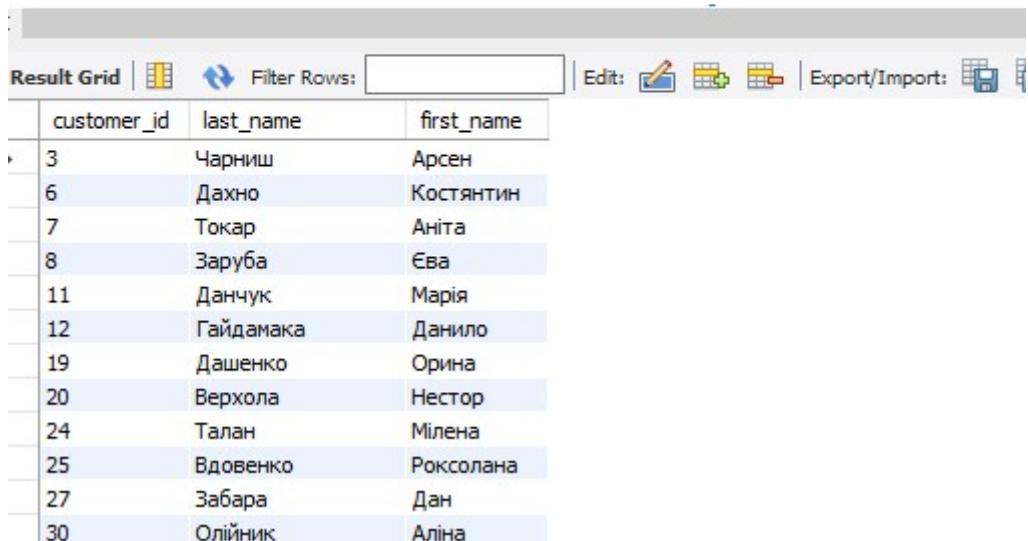
customer_id	last_name	first_name
1	Бандера	Симон
2	Литвиненко	Дарина
4	Данильчук	Златослава
5	Ковалюк	Григорій
9	Фаренюк	Владислав
10	Пустовіт	Алла
13	Радченко	Давид
14	Дем'янюк	Веніамін
15	Яремків	Анжела

 " data-bbox="242 727 826 925"/>

-- 2. Показати всіх клієнтів, які не зробили жодного замовлення

```
SELECT customer_id, last_name, first_name
FROM customers
WHERE customer_id NOT IN (SELECT customer_id FROM orders);
```

```
9      -- 2. Показати всіх клієнтів, які не зробили жодного замовлення
10 •  SELECT customer_id, last_name, first_name
11   FROM customers
12 WHERE customer_id NOT IN (SELECT customer_id FROM orders);
13
```



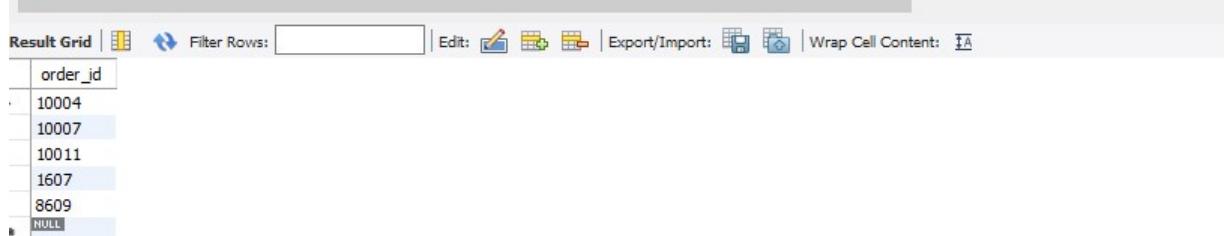
	customer_id	last_name	first_name
•	3	Чарниш	Арсен
	6	Дахно	Костянтин
	7	Токар	Аніта
	8	Заруба	Єва
	11	Данчук	Марія
	12	Гайдамака	Данило
	19	Дашенко	Орина
	20	Верхола	Нестор
	24	Талан	Мілена
	25	Вдовенко	Роксолана
	27	Забара	Дан
	30	Олійник	Аліна

-- 3. Показати замовлення, у яких є хоча б одна запчастиня, кількість якої не менш за 2

```
SELECT order_id
FROM orders
```

```
WHERE EXISTS (SELECT * FROM orders_goods WHERE orders_goods.order_id =
orders.order_id AND quantity >= 2);
```

```
14      -- 3. Показати замовлення, у яких є хоча б одна запчастиня, кількість якої не менш за 2
15 •  SELECT order_id
16   FROM orders
17 WHERE EXISTS (SELECT * FROM orders_goods WHERE orders_goods.order_id = orders.order_id AND quantity >= 2);
18
```



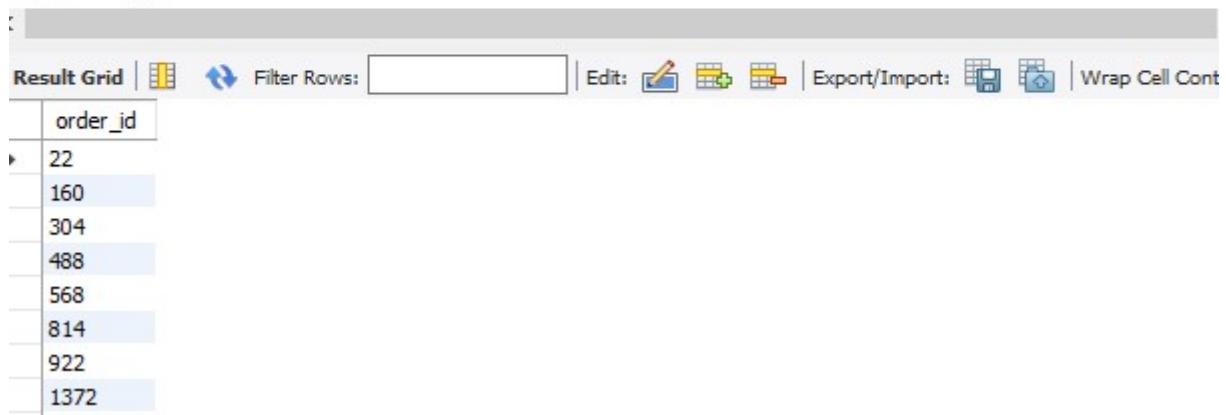
	order_id
	10004
	10007
	10011
	1607
	8609
	NULL

-- 4. Показати замовлення, для яких всі запчастини мають ціну, меншу ніж 3000

```
SELECT order_id
FROM orders
WHERE NOT EXISTS (
    SELECT *
    FROM orders_goods
    WHERE orders_goods.order_id = orders.order_id
    AND price_out >= 3000
);
```

19 -- 4. Показати замовлення, для яких всі запчастини мають ціну, меншу ніж 3000

```
20 •  SELECT order_id
21   FROM orders
22   WHERE NOT EXISTS (
23       SELECT *
24       FROM orders_goods
25       WHERE orders_goods.order_id = orders.order_id
26       AND price_out >= 3000
27   );
```



order_id
22
160
304
488
568
814
922
1372

-- 5. Показати клієнтів, у яких є замовлення з сумою запчастин більше середньої суми запчастин всіх замовлень

```
SELECT customer_id, last_name, first_name
FROM customers
WHERE customer_id IN (
    SELECT o.customer_id
```

```

FROM orders o
WHERE (
    SELECT SUM(og.price_out * og.quantity)
    FROM orders_goods og
    WHERE og.order_id = o.order_id
) > (
    SELECT AVG(total_sum)
    FROM (
        SELECT SUM(price_out * quantity) AS total_sum
        FROM orders_goods
        GROUP BY order_id
    ) AS order_totals
)
);

```

```

31      -- 5. Показати клієнтів, у яких є замовлення з сумою запчастин більше середньої суми запчастин всіх замовлень
32  •  SELECT customer_id, last_name, first_name
33  FROM customers
34  WHERE customer_id IN (
35      SELECT o.customer_id
36      FROM orders o
37  WHERE (
38      SELECT SUM(og.price_out * og.quantity)
39      FROM orders_goods og
40      WHERE og.order_id = o.order_id
41  ) > (
42      SELECT AVG(total_sum)
43      FROM (
44          SELECT SUM(price_out * quantity) AS total_sum
45          FROM orders_goods
46          GROUP BY order_id
47      ) AS order_totals
48  )
49 );
50

```

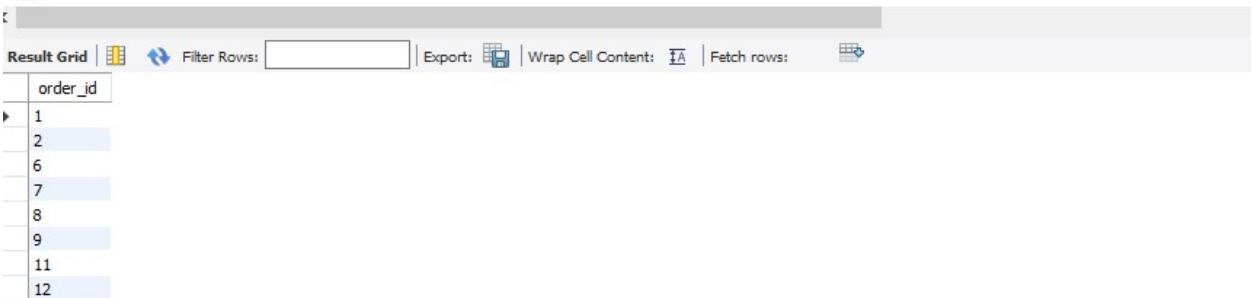
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

	customer_id	last_name	first_name
▶	1	Бандера	Симон
	2	Литвиценко	Даріна
	10	Пустовіт	Алла
	14	Дем'янок	Веніамін
	15	Яремків	Анжела
	17	Рябовіл	Марія
	21	Рак	Богданна
	22	Юрчишина	Ганна
	23	Бевз	Владислав

-- 6. Показати всі замовлення, у яких є товар з ціною, меншою за будь-яку з цін товарів у замовленні з номером 1

```
SELECT DISTINCT order_id
FROM orders_goods
WHERE price_out < ANY (SELECT price_out FROM orders_goods WHERE order_id = 1);
```

```
52      -- 6. Показати всі замовлення, у яких є товар з ціною, меншою за будь-яку з цін товарів у замовленні з номером 1
53 •  SELECT DISTINCT order_id
54    FROM orders_goods
55    WHERE price_out < ANY (SELECT price_out FROM orders_goods WHERE order_id = 1);
56
```

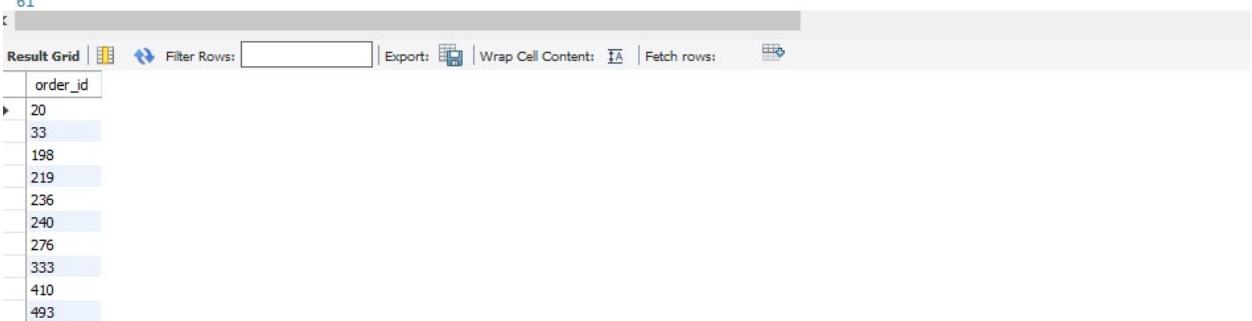


order_id
1
2
6
7
8
9
11
12

-- 7. Показати всі замовлення, у яких кожна одиниця товару має ціну більшу за ціну будь-якого товару в замовленні з номером 2

```
SELECT DISTINCT order_id
FROM orders_goods
WHERE price_out > ALL (SELECT price_out FROM orders_goods WHERE order_id = 2);
```

```
57      -- 7. Показати всі замовлення, у яких кожна одиниця товару має ціну більшу за ціну будь-якого товару в замовленні з номером 2
58 •  SELECT DISTINCT order_id
59    FROM orders_goods
60    WHERE price_out > ALL (SELECT price_out FROM orders_goods WHERE order_id = 2);
61
```

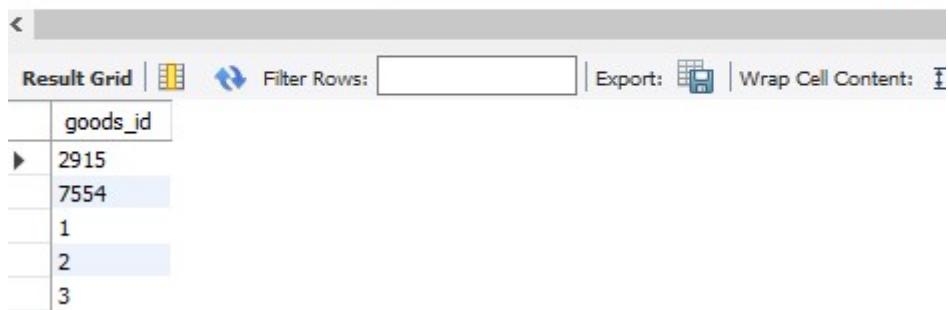


order_id
20
33
198
219
236
240
276
333
410
493

-- 8. Показати товари, які входять до складу хоча б одного замовлення клієнта з `customer_id = 1`

```
SELECT DISTINCT goods_id
FROM orders_goods
WHERE order_id IN (
    SELECT order_id FROM orders WHERE customer_id = 1);
```

```
69 •  SELECT DISTINCT goods_id
70      FROM orders_goods
71      WHERE order_id IN (
72          SELECT order_id FROM orders WHERE customer_id = 1);
```



goods_id
2915
7554
1
2
3

-- 9. Показати клієнтів, які мають замовлення з товарами, яких немає на складі (`stock = 0`)

```
SELECT customer_id, last_name, first_name
FROM customers
WHERE EXISTS (
    SELECT *
    FROM orders
    WHERE orders.customer_id = customers.customer_id AND orders.order_id IN(
        SELECT order_id
        FROM orders_goods
        WHERE goods_id IN (
            SELECT goods_id FROM goods WHERE stock = 0)));
```

```

74      -- 9. Показати клієнтів, які мають замовлення з товарами, яких немає на складі (stock = 0)
75  •  SELECT customer_id, last_name, first_name
76    FROM customers
77    WHERE EXISTS (
78      SELECT *
79      FROM orders
80      WHERE orders.customer_id = customers.customer_id AND orders.order_id IN(
81        SELECT order_id
82        FROM orders_goods
83        WHERE goods_id IN (
84          SELECT goods_id FROM goods WHERE stock = 0)));

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	customer_id	last_name	first_name
▶	7943	Опанасенко	Степан
1	Бандера	Симон	
*	NULL	NULL	NULL

-- 10. Показати клієнтів, які мають замовлення лише з товарами, кількість яких в замовленні не меньш за 1

```

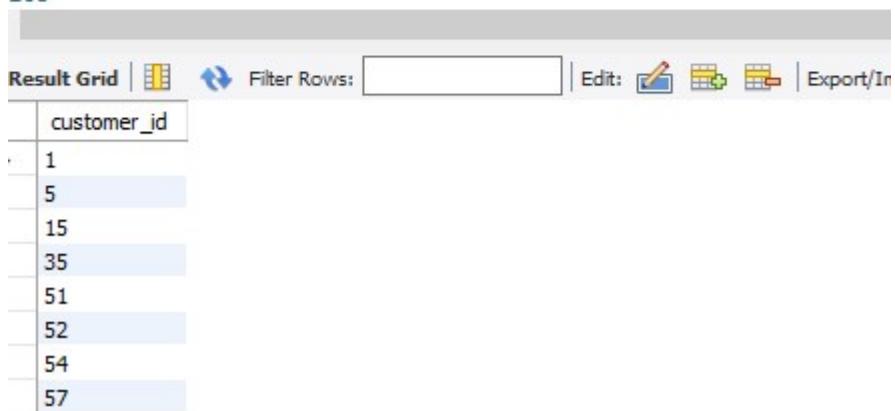
SELECT customer_id
FROM customers c
WHERE EXISTS (
  SELECT 1
  FROM orders o
  WHERE o.customer_id = c.customer_id
)
AND NOT EXISTS (
  SELECT 1
  FROM orders o
  JOIN orders_goods og ON o.order_id = og.order_id
  WHERE o.customer_id = c.customer_id
  AND og.quantity < 1
);

```

```

91 •   SELECT customer_id
92     FROM customers c
93     WHERE EXISTS (
94         SELECT 1
95         FROM orders o
96         WHERE o.customer_id = c.customer_id
97     )
98     AND NOT EXISTS (
99         SELECT 1
100        FROM orders o
101       JOIN orders_goods og ON o.order_id = og.order_id
102       WHERE o.customer_id = c.customer_id
103         AND og.quantity < 1
104     );
105

```



The screenshot shows a database result grid with a single column labeled 'customer_id'. The data is as follows:

customer_id
1
5
15
35
51
52
54
57

-- #11 Отримати замовлення – знайти клієнтів, що не купували запчастини з
груп 'supplies HP' або 'supplies Epson'

```

SELECT DISTINCT c.last_name,
    c.first_name,
    o.order_id,
    o.date_invoice,
    o.date_returned,
    gd.name AS goods_name,
    og.price_out,
    og.quantity,
    gr.name AS group_name
FROM orders o
JOIN customers c USING(customer_id)
JOIN orders_goods og USING(order_id)
JOIN goods g USING(goods_id)
JOIN goods_dscr gd ON g.goods_dscr_id = gd.goods_dscr_id

```

```

JOIN `group` gr ON gd.group_id = gr.group_id
WHERE c.customer_id NOT IN (
    SELECT DISTINCT c2.customer_id
    FROM customers c2
    JOIN orders o2 ON c2.customer_id = o2.customer_id
    JOIN orders_goods og2 ON o2.order_id = og2.order_id
    JOIN goods g2 ON og2.goods_id = g2.goods_id
    JOIN goods_dscr gd2 ON g2.goods_dscr_id = gd2.goods_dscr_id
    JOIN `group` gr2 ON gd2.group_id = gr2.group_id
    WHERE gr2.name IN ('supplies HP', 'supplies Epson')
)
ORDER BY o.order_id;

```

```

107 •   SELECT DISTINCT c.last_name,
108                 c.first_name,
109                 o.order_id,
110                 o.date_invoice,
111                 o.date_returned,
112                 gd.name AS goods_name,
113                 og.price_out,
114                 og.quantity,
115                 gr.name AS group_name
116             FROM orders o
117             JOIN customers c USING(customer_id)
118             JOIN orders_goods og USING(order_id)
119             JOIN goods g USING(goods_id)
120             JOIN goods_dscr gd ON g.goods_dscr_id = gd.goods_dscr_id
121             JOIN `group` gr ON gd.group_id = gr.group_id
122             WHERE c.customer_id NOT IN (
123                 SELECT DISTINCT c2.customer_id
124                 FROM customers c2
125                 JOIN orders o2 ON c2.customer_id = o2.customer_id
126                 JOIN orders_goods og2 ON o2.order_id = og2.order_id
127                 JOIN goods g2 ON og2.goods_id = g2.goods_id
128                 JOIN goods_dscr gd2 ON g2.goods_dscr_id = gd2.goods_dscr_id
129                 JOIN `group` gr2 ON gd2.group_id = gr2.group_id
130                 WHERE gr2.name IN ('supplies HP', 'supplies Epson')
131             )
132             ORDER BY o.order_id;

```

Result Grid | Filter Rows: | Export: Wrap Cell Content: Fetch rows:

	last_name	first_name	order_id	date_invoice	date_returned	goods_name	price_out	quantity	group_name
	Щербань	Нестор	5	2024-03-05	2024-03-09	Товар OLW43	9533	1.772	supplies Ricoh
	Щербань	Нестор	5	2024-03-05	2024-03-09	Товар B9NF5	13939	0.132	spares Canon
	Щербань	Нестор	5	2024-03-05	2024-03-09	Товар W7DYB	8026	0.986	spares Canon
	Зінчук	Клавдія	6	2024-03-07	2024-03-22	Товар 887E2	1976	0.312	supplies Ricoh
	Зінчук	Клавдія	6	2024-03-07	2024-03-22	Товар 6NHLR	1746	1.058	spares Ricoh
	Зінчук	Клавдія	6	2024-03-07	2024-03-22	Товар KFNWQ	10857	1.174	spares Ricoh

2.3.8. 8 розроблених SQL-запитів UPDATE з використанням інструкцій WHERE

-- #1 Підтверджен email чи phone при реєстрації співробітника

UPDATE employees

SET is_confirmed = 1

WHERE employee_id = 1;

До виконання запиту

```
1 •  SELECT employee_id,
2          is_confirmed
3  FROM employees
4 WHERE employee_id = 1;
```

	employee_id	is_confirmed
▶	1	0
*	NULL	NULL

Після виконання запиту

```
1 •  SELECT employee_id,
2          is_confirmed
3  FROM employees
4 WHERE employee_id = 1;
```

	employee_id	is_confirmed
▶	1	1
*	NULL	NULL

70 19:19:55 UPDATE employees SET is_confirmed = 1 WHERE employee_id = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.016 sec

-- #2 Підтверджен email чи phone при реєстрації Замовника

UPDATE customers

SET is_confirmed = 1

WHERE customer_id = 1;

До виконання запиту

```
1 •  SELECT customer_id,
2          is_confirmed
3  FROM customers
4 WHERE customer_id = 1;
```

	customer_id	is_confirmed
▶	1	0
*	NULL	NULL

Після виконання запиту

```
1 •  SELECT customer_id,
2          is_confirmed
3  FROM customers
4 WHERE customer_id = 1;
```

	customer_id	is_confirmed
▶	1	1
*	NULL	NULL

73 19:24:28 UPDATE customers SET is_confirmed = 1 WHERE customer_id = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec

-- #3 Редагування, блокування (деактивізація) Замовників

UPDATE customers

SET is_active = 0

WHERE customer_id = 1;

До виконання запиту

```
1 •   SELECT customer_id,  
2           is_active  
3   FROM customers  
4  WHERE customer_id = 1;
```

Result Grid	
customer_id	is_active
1	1
NULL	NULL

Після виконання запиту

```
1 •   SELECT customer_id,  
2           is_active  
3   FROM customers  
4  WHERE customer_id = 1;
```

Result Grid	
customer_id	is_active
1	0
NULL	NULL

```
✓ 76 19:30:41 UPDATE customers SET is_active = 0 WHERE customer_id = 1  
1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
```

-- #4 Редагування, блокування (деактивізація, якщо виконані умови:

goods.stock=0 and goods.reserve=0) запчастин

UPDATE goods_dscr

SET is_active = 0

WHERE goods_dscr_id = 5

AND NOT EXISTS (

SELECT 1

FROM goods

WHERE goods.goods_dscr_id = goods_dscr.goods_dscr_id

AND (goods.stock > 0 OR goods.reserve > 0)

);

До виконання запиту

```
1 •   SELECT gd.goods_dscr_id, stock, reserve, is_active  
2   FROM goods_dscr gd, goods  
3  WHERE goods.goods_dscr_id = gd.goods_dscr_id  
4    AND gd.goods_dscr_id = 5  
5    AND NOT EXISTS (  
6      SELECT 1  
7      FROM goods g  
8     WHERE goods.goods_dscr_id = gd.goods_dscr_id  
9       AND (goods.stock > 0 OR goods.reserve > 0)  
10    );
```

Result Grid			
goods_dscr_id	stock	reserve	is_active
5	0.000	0.000	1

Після виконання запиту

```
1 •   SELECT gd.goods_dscr_id, stock, reserve, is_active  
2   FROM goods_dscr gd, goods  
3  WHERE goods.goods_dscr_id = gd.goods_dscr_id  
4    AND gd.goods_dscr_id = 5  
5    AND NOT EXISTS (  
6      SELECT 1  
7      FROM goods g  
8     WHERE goods.goods_dscr_id = gd.goods_dscr_id  
9       AND (goods.stock > 0 OR goods.reserve > 0)  
10    );
```

Result Grid			
goods_dscr_id	stock	reserve	is_active
5	0.000	0.000	0

```
88 19:44:06 UPDATE goods_dscr SET is_active = 0 WHERE goods_dscr_id = 5 AND NOT EXISTS ( ... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
```

-- #5 Редагування замовлень, якщо не readonly

UPDATE orders

SET date_returned = '2024-11-04'

WHERE arrival_id = 9

AND `read_only` = 0;

До виконання запиту

```
1 •  SELECT order_id,
2      date_returned,
3      read_only
4  FROM orders
5 WHERE order_id = 9;
```

Result Grid			
	order_id	date_returned	read_only
▶	9	NULL	0
*	NULL	NULL	NULL

Після виконання запиту

```
1 •  SELECT order_id,
2      date_returned,
3      read_only
4  FROM orders
5 WHERE order_id = 9;
```

Result Grid			
	order_id	date_returned	read_only
▶	9	2024-11-04	0
*	NULL	NULL	NULL

```
104 20:03:15 UPDATE orders SET date_returned = '2024-11-04' WHERE order_id = 9 AND `read_only` = 0 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.016 sec
```

-- #6 Редагування приходів, якщо не readonly

UPDATE arrivals

SET date_arrival = '2024-11-05'

WHERE arrival_id = 1002

AND `read_only` = 0;

До виконання запиту

```
1 •  SELECT arrival_id,
2      date_arrival,
3      read_only
4  FROM arrivals
5 WHERE arrival_id = 1002;
```

Result Grid			
	arrival_id	date_arrival	read_only
▶	1002	NULL	0
*	NULL	NULL	NULL

Після виконання запиту

```
1 •  SELECT arrival_id,
2      date_arrival,
3      read_only
4  FROM arrivals
5 WHERE arrival_id = 1002;
```

Result Grid			
	arrival_id	date_arrival	read_only
▶	1002	2024-11-05	0
*	NULL	NULL	NULL

```
107 20:09:21 UPDATE arrivals SET date_arrival = '2024-11-05' WHERE arrival_id = 1002 AND `read_only` ... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
```

-- #7 Редагування оплат, якщо не readonly

```
UPDATE pay
SET pay_form = 'готівка'
WHERE pay_id = 1
    AND `read_only` = 0;
```

До виконання запиту

1	•	SELECT	pay_id,
2			pay_form,
3			read_only
4		FROM	ss_innodb.pay
5		WHERE	pay_id = 9962;

Після виконання запиту

1	•	SELECT	pay_id,
2			pay_form,
3			read_only
4		FROM	ss_innodb.pay
5		WHERE	pay_id = 9962;

114 20:20:07 UPDATE pay SET pay_form = 'готівка' WHERE pay_id = 9962 AND `read_only` = 0 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.016 sec

-- #8 Блокування для запису (read_only=1) замовень, приходів, оплат

ПАКЕТОМ за певний проміжок часу.

START TRANSACTION;

SET SQL_SAFE_UPDATES = 0; -- Вимкнути безпечний режим (поки немає індексу на даті!)

UPDATE orders

SET read_only = 1

WHERE date_returned between '2024-02-01' and '2024-02-01';

UPDATE arrivals

SET read_only = 1

WHERE date_arrival between '2024-02-01' and '2024-02-01';

UPDATE pay

SET read_only = 1

WHERE date_pay between '2024-02-01' and '2024-02-01';

SET SQL_SAFE_UPDATES = 1; -- Увімкнути безпечний режим назад

COMMIT;

До виконання запиту

```

1 •  SELECT order_id,
2      date_returned,
3      read_only
4  FROM orders
5 WHERE date_returned
6      between '2024-02-01'
7      and '2024-02-01'
8 ORDER BY order_id;

```

order_id	date_returned	read_only
32	2024-02-01	0
41	2024-02-01	0
640	2024-02-01	0
1131	2024-02-01	0
1237	2024-02-01	0
1918	2024-02-01	0
3329	2024-02-01	0
3431	2024-02-01	0
3702	2024-02-01	0
3835	2024-02-01	0
NULL	NULL	NULL

```

1 •  SELECT arrival_id,
2      date_arrival,
3      read_only
4  FROM arrivals
5 WHERE date_arrival
6      between '2024-02-01'
7      and '2024-02-01'
8 ORDER BY arrival_id;

```

arrival_id	date_arrival	read_only
80	2024-02-01	0
125	2024-02-01	0
196	2024-02-01	0
355	2024-02-01	0
538	2024-02-01	0
912	2024-02-01	0
NULL	NULL	NULL

```

1 •  SELECT pay_id,
2      date_pay,
3      read_only
4  FROM ss_innodb.pay
5 WHERE date_pay
6      between '2024-02-01'
7      and '2024-02-01'
8 ORDER BY pay_id;

```

pay_id	date_pay	read_only
216	2024-02-01	0
886	2024-02-01	0
956	2024-02-01	0
1141	2024-02-01	0
1762	2024-02-01	0
2264	2024-02-01	0
2532	2024-02-01	0
2538	2024-02-01	0
2973	2024-02-01	0
3211	2024-02-01	0
NULL	NULL	NULL

```

128 20:55:48 START TRANSACTION
129 20:55:48 SET SQL_SAFE_UPDATES = 0
130 20:55:48 UPDATE orders SET read_only = 1 WHERE date_returned between '2024-02-01' and '2024-02-01' 33 row(s) affected Rows matched: 33 Changed: 33 Warnings: 0 0.031 sec
131 20:55:48 UPDATE arrivals SET read_only = 1 WHERE date_arrival between '2024-02-01' and '2024-02-01' 6 row(s) affected Rows matched: 6 Changed: 6 Warnings: 0 0.000 sec
132 20:55:48 UPDATE pay SET read_only = 1 WHERE date_pay between '2024-02-01' and '2024-02-01' 45 row(s) affected Rows matched: 45 Changed: 45 Warnings: 0 0.015 sec
133 20:55:48 SET SQL_SAFE_UPDATES = 1
134 20:55:48 COMMIT

```

Після виконання запиту

```

1 •  SELECT order_id,
2      date_returned,
3      read_only
4  FROM orders
5 WHERE date_returned
6      between '2024-02-01'
7      and '2024-02-01'
8 ORDER BY order_id;

```

order_id	date_returned	read_only
32	2024-02-01	1
41	2024-02-01	1
640	2024-02-01	1
1131	2024-02-01	1
1237	2024-02-01	1
1918	2024-02-01	1
3329	2024-02-01	1
3431	2024-02-01	1
3702	2024-02-01	1
3835	2024-02-01	1
NULL	NULL	NULL

```

1 •  SELECT arrival_id,
2      date_arrival,
3      read_only
4  FROM arrivals
5 WHERE date_arrival
6      between '2024-02-01'
7      and '2024-02-01'
8 ORDER BY arrival_id;

```

arrival_id	date_arrival	read_only
80	2024-02-01	1
125	2024-02-01	1
196	2024-02-01	1
355	2024-02-01	1
538	2024-02-01	1
912	2024-02-01	1
NULL	NULL	NULL

```

1 •  SELECT pay_id,
2      date_pay,
3      read_only
4  FROM ss_innodb.pay
5 WHERE date_pay
6      between '2024-02-01'
7      and '2024-02-01'
8 ORDER BY pay_id;

```

pay_id	date_pay	read_only
216	2024-02-01	1
886	2024-02-01	1
956	2024-02-01	1
1141	2024-02-01	1
1762	2024-02-01	1
2264	2024-02-01	1
2532	2024-02-01	1
2538	2024-02-01	1
2973	2024-02-01	1
3211	2024-02-01	1
NULL	NULL	NULL

2.3.9. 7 SQL-запитів INSERT з використанням інструкцій WHERE

```
-- #1 Додавання Співробітника с перевіркою по його ІПН (код)
START TRANSACTION;
SELECT employee_id
FROM employees
WHERE code = "0000001234" -- Перевірка, чи є запис з таким кодом
LIMIT 1;

INSERT INTO employees (
    status_id,
    position_id,
    last_name,
    first_name,
    middle_name,
    code,
    birthday,
    phone,
    password)
SELECT 5, 6,
    "Федорків", "Петро", "Іванович",
    "0000001234",
    "2002-12-12",
    '+38066 123-4567',
    "asdanhlsk4j634lk5tr23qr"

WHERE NOT EXISTS (
    SELECT 1
    FROM employees
    WHERE code = "0000001234" -- Якщо записів немає додаємо нового Співробітника
);
COMMIT;
```

```

1      -- Добавлення співробітника з перевіркою по його ІПН (код)
2 •  START TRANSACTION;
3
4      -- Перевірка, чи є запис з таким кодом
5 •  SELECT employee_id
6      FROM employees
7      WHERE code = "0000001234"
8      LIMIT 1;
9
10     -- Якщо записів немає (тобто жоден рядок не повернувся), додаємо нового співробітника
11 •  INSERT INTO employees (status_id, position_id, last_name, first_name, middle_name, code, birthday, phone, password)
12      SELECT 5, 6, "Федорків", "Петро", "Іванович", "0000001234", "2002-12-12", '+38066 123-4567', "asdanhlsk4j634lk5tr23qr"
13      WHERE NOT EXISTS (
14          SELECT 1
15          FROM employees
16          WHERE code = "0000001234"
17      );
18
19 •  COMMIT;

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
*	employee_id	NULL			

employees 2 x		
Output		
Action Output		
#	Time	Action
141	22:34:33	START TRANSACTION
142	22:34:33	SELECT employee_id FROM employees WHERE code = "0000001234" LIMIT 1
143	22:34:33	INSERT INTO employees (status_id, position_id, last_name, first_name, middle_name, code, b... 1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
144	22:34:33	COMMIT

```

1 •  SELECT *
2      FROM ss_innodb.employees
3      WHERE code = "0000001234";

```

Result Grid												
employee_id	status_id	position_id	first_name	middle_name	last_name	code	birthday	email	phone	address	description	password
21	5	6	Петро	Іванович	Федорків	0000001234	2002-12-12	HULL	+38066 123-4567	HULL	HULL	asdanhlsk4j634lk5tr23qr
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

-- #2 Добавлення замовника по його номеру телефона (або email)

START TRANSACTION;

```

SELECT customer_id
FROM customers
WHERE phone = '+38067 123-4567' -- Перевірка, чи є запис з таким телефоном
LIMIT 1;

INSERT INTO customers (
    status_id,
    city_id,
    last_name,
    first_name,
    phone,
    password)

SELECT 6, 1,
    "Федорків", "Павло",
    '+38067 123-4567',
    "asdanhlsk4j634lk5tr23qr"

WHERE NOT EXISTS (
    SELECT 1
    FROM customers
    WHERE phone = '+38067 123-4567' -- Якщо записів немає, додаємо нового замовника
);

COMMIT;

```

До виконання запиту

```

1 •  SELECT customer_id,
2           last_name, first_name, phone
3   FROM customers
4  WHERE last_name = 'Федорків';

```

Result Grid				
	customer_id	last_name	first_name	phone
*	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL

Після виконання запиту

```

1 •  SELECT customer_id,
2           last_name, first_name, phone
3   FROM customers
4  WHERE last_name = 'Федорків';

```

Result Grid				
	customer_id	last_name	first_name	phone
▶	10011	Федорків	Павло	+38067 123-4567
*	NULL	NULL	NULL	NULL

```

1  -- #2 Добавлення Замовника по його номеру телефона (або email)
2 • START TRANSACTION;
3
4  -- Перевірка, чи є запис з таким телефоном або кодом
5 • SELECT customer_id
6  FROM customers
7  WHERE phone = '+38067 123-4567'
8  LIMIT 1;
9
10 -- Якщо записів немає (тобто жоден рядок не повернувся), додаємо нового замовника
11 • INSERT INTO customers (status_id, city_id, last_name, first_name, phone, password)
12  SELECT 6, 1, "Федорків", "Павло", '+38067 123-4567', "asdanhlsk4j634lk5tr23qr"
13  WHERE NOT EXISTS (
14    SELECT 1
15    FROM customers
16    WHERE phone = '+38067 123-4567'
17  );
18
19 • COMMIT;

```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
*	customer_id	NULL			

customers 1 ×			
Output			
Action Output			
#	Time	Action	Message
147	22:49:56	START TRANSACTION	0 row(s) affected
148	22:49:56	SELECT customer_id FROM customers WHERE phone = '+38067 123-4567' LIMIT 1	0 row(s) returned
149	22:49:56	INSERT INTO customers (status_id, city_id, last_name, first_name, phone, password) SELEC...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
150	22:49:56	COMMIT	0 row(s) affected

-- #3 Добавлення нової запчастини (part_number, name - UNIQUE)

START TRANSACTION;

```

SELECT goods_dscr_id, part_number, name
FROM goods_dscr
WHERE name = 'Товар 4XAWK' -- Перевірка, чи є запис з таким name
      OR part_number = "2345125-21413"
LIMIT 1;

```

INSERT INTO goods_dscr (unit_id, group_id, part_number, name)

```

SELECT 1, 1, "2345125-21413", "Товар 4XAWK"
WHERE NOT EXISTS (
    SELECT 1
    FROM goods_dscr
    WHERE name = "Товар 4XAWK" -- Якщо записів немає, додаємо нову запчастину
        OR part_number = "2345125-21413"
);
COMMIT;

```

```

1      -- #3 Добавлення нової запчастини (part_number, name - UNIQUE)
2 • START TRANSACTION;
3
4      -- Перевірка, чи є запис з таким name
5 • SELECT goods_dscr_id, part_number, name
6      FROM goods_dscr
7      WHERE name = 'Товар 4XAWK'
8          OR part_number = "2345125-21413"
9      LIMIT 1;
10
11      -- Якщо записів немає (тобто жоден рядок не повернувся), додаємо нову запчастину
12 • INSERT INTO goods_dscr (unit_id, group_id, part_number, name)
13     SELECT 1, 1, "2345125-21413", "Товар 4XAWK"
14     WHERE NOT EXISTS (
15         SELECT 1
16         FROM goods_dscr
17         WHERE name = "Товар 4XAWK"
18             OR part_number = "2345125-21413"
19     );
20

```

Result Grid		
goods_dscr_id	part_number	name
*	NULL	NULL

goods_dscr2 x			
Output			
Action Output			Message
#	Time	Action	Message
158	23:09:32	START TRANSACTION	0 row(s) affected
159	23:09:32	SELECT goods_dscr_id, part_number, name FROM goods_dscr WHERE name = 'Товар 4XAWK'	0 row(s) returned
160	23:09:32	INSERT INTO goods_dscr (unit_id, group_id, part_number, name) SELECT 1, 1, "2345125-21413", "Товар 4XAWK"	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
161	23:09:32	COMMIT	0 row(s) affected

До виконання запиту

```
18 •  SELECT goods_dscr_id, name
19   FROM goods_dscr
20  WHERE name = 'Товар 4ХAWK';
```

Result Grid	
goods_dscr_id	name
NULL	NULL

Після виконання запиту

```
18 •  SELECT goods_dscr_id, name
19   FROM goods_dscr
20  WHERE name = 'Товар 4ХAWK';
```

Result Grid	
goods_dscr_id	name
10001	Товар 4ХAWK

-- #4 Додавання нової ціни до існуючої запчастини

```
INSERT INTO goods (goods_dscr_id, price_in)
SELECT gd.goods_dscr_id, 10000
FROM goods_dscr gd
LEFT JOIN goods g
  ON g.goods_dscr_id = gd.goods_dscr_id
  AND g.price_in = 10000
WHERE gd.name = 'Товар 4ХAWK'
  AND g.goods_dscr_id IS NULL; -- Умова, щоб не було запису з такою ціною
```

```
170 23:38:13 INSERT INTO goods (goods_dscr_id, price_in) SELECT gd.goods_dscr_id, 10000 FROM goo... 1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
```

До виконання запиту

```
1 •  SELECT g.goods_dscr_id, price_in
2   FROM goods_dscr gd
3  LEFT JOIN goods g ON g.goods_dscr_id = gd.goods_dscr_id AND g.price_in = 10000
4  WHERE gd.name = 'Товар 4ХAWK' AND g.goods_dscr_id IS NULL;
```

Result Grid	
goods_dscr_id	price_in
NULL	NULL

Після виконання запиту

```

1 •  SELECT g.goods_dscr_id, price_in
2   FROM goods_dscr gd
3   LEFT JOIN goods g ON g.goods_dscr_id = gd.goods_dscr_id AND g.price_in = 10000
4 WHERE gd.name = 'Товар 4XAWK' AND g.goods_dscr_id IS NULL;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
goods_dscr_id	price_in			

```

1 •  SELECT g.goods_dscr_id, price_in
2   FROM goods_dscr gd
3   LEFT JOIN goods g ON g.goods_dscr_id = gd.goods_dscr_id -- AND g.price_in = 10000
4 WHERE gd.name = 'Товар 4XAWK' -- AND g.goods_dscr_id IS NULL;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
goods_dscr_id	price_in			
10001	10000			

-- #5 Додавання нового замовлення на ремонт техніки (приймання в ремонт, передача в сервіс).

START TRANSACTION;

-- 1. Додаємо нове замовлення в таблицю orders

```

INSERT INTO orders (customer_id, date_invoice)
VALUES ('1', '2024-11-06');

```

-- 2. Отримуємо order_id останнього вставленого замовлення

```
SET @order_id = LAST_INSERT_ID();
```

-- 3. Додаємо запис у таблицю orders_received, використовуючи отриманий order_id

```

INSERT INTO orders_received (order_id, employee_id, brand_id, model_name, sn,
equipment, fault_description)
VALUES (@order_id, 1, 1, 2015, '3455-456456-456', 'без шнура живлення',
'погана подача паперу, неякісний друк');

```

-- 4. Додаємо запис у таблицю orders_repair, використовуючи отриманий order_id

```
INSERT INTO orders_repair (order_id, employee_id)
VALUES (@order_id, '1');
```

```
COMMIT;
```

До виконання запиту

```
1 •  SELECT order_id
2   FROM orders
3   ORDER BY order_id desc
4   LIMIT 1;
```

Result Grid	
order_id	
10011	
NULL	

Після виконання запиту

```
1 •  SELECT order_id
2   FROM orders
3   ORDER BY order_id desc
4   LIMIT 1;
```

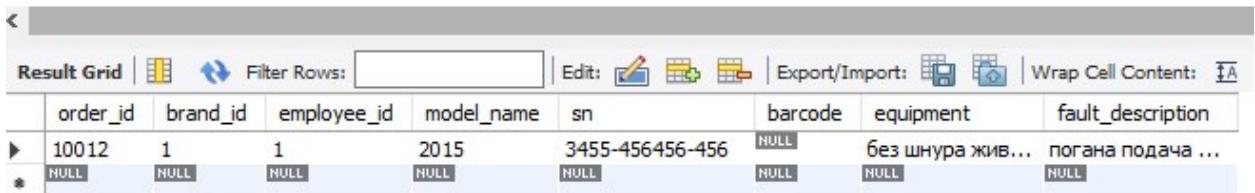
Result Grid	
order_id	
10012	
NULL	

```
1   -- #5 Додавання нового замовлення на ремонт техніки (приймання в ремонт, передача в сервіс).
2 •  START TRANSACTION;
3
4   -- 1. Додаємо нове замовлення в таблицю orders
5 •  INSERT INTO orders (customer_id, date_invoice)
6   VALUES ('1', '2024-11-06');
7
8   -- 2. Отримуємо order_id останнього вставленого замовлення
9 •  SET @order_id = LAST_INSERT_ID();
10
11  -- 3. Додаємо запис у таблицю orders_received, використовуючи отриманий order_id
12 •  INSERT INTO orders_received (order_id, employee_id, brand_id, model_name, sn, equipment, fault_description)
13   VALUES (@order_id, 1, 1, 2015, '3455-456456-456', 'без шнура живлення', 'погана подача паперу, неякісний друк');
14
15  -- 4. Додаємо запис у таблицю orders_repair, використовуючи отриманий order_id
16 •  INSERT INTO orders_repair (order_id, employee_id)
17   VALUES (@order_id, '1');
18
19 •  COMMIT;
20
```

Output		
#	Time	Action
1	22:34:29	START TRANSACTION
2	22:34:29	INSERT INTO orders (customer_id, date_invoice) VALUES ('1', '2024-11-06')
3	22:34:29	SET @order_id = LAST_INSERT_ID()
4	22:34:29	INSERT INTO orders_received (order_id, employee_id, brand_id, model_name, sn, equipment, fault_descri...)
5	22:34:29	INSERT INTO orders_repair (order_id, employee_id) VALUES (@order_id, '1')
6	22:34:29	COMMIT

Після виконання запиту

```
1 •  SELECT *
2   FROM orders_received
3   WHERE order_id = 10012;
```



	order_id	brand_id	employee_id	model_name	sn	barcode	equipment	fault_description
▶	10012	1	1	2015	3455-456456-456	NULL	NULL	без шнура жив...
*	HULL	NULL	NULL	NULL	NULL	NULL	NULL	погана подача ...

```
1 •  SELECT *
2   FROM orders_repair
3   WHERE order_id = 10012;
```



	order_id	employee_id	date_ready	price_repair	description
▶	10012	1	NULL	0	NULL
*	HULL	NULL	NULL	NULL	NULL

-- #6 додавання оплати по замовленню

```
START TRANSACTION;
SET @order_id = 10005;
SET @customer_id = 1;
-- Перевірка, чи існує замовлення для цього клієнта
SELECT 1
FROM orders
WHERE order_id = @order_id
AND customer_id = @customer_id
LIMIT 1;
-- Додавання запису про оплату, якщо замовлення існує
INSERT INTO pay (customer_id, employee_id, order_id, date_pay, pay_sum)
SELECT @customer_id, 1, @order_id, '2024-11-06', '83000'
WHERE EXISTS (
  SELECT 1
  FROM orders
  WHERE order_id = @order_id)
```

```

        AND customer_id = @customer_id
);

COMMIT;

```

До виконання запиту

```

1 •  SELECT pay_id,
2      date_pay
3  FROM pay
4 WHERE order_id = 10005;
5
6

```

Result Grid		Filter Rows:
	pay_id	date_pay
*	NULL	NULL

Після виконання запиту

```

1 •  SELECT pay_id, date_pay,
2      order_id, customer_id
3  FROM pay
4 WHERE order_id = 10005;
5

```

Result Grid				Filter Rows:	Export:
	pay_id	date_pay	order_id	customer_id	Fetch rows:
▶	11002	2024-11-06	10005	1	

```

2 •  START TRANSACTION;
3 •  SET @order_id = 10005;
4 •  SET @customer_id = 1;
5
6  -- Перевірка, чи існує замовлення для цього клієнта
7 •  SELECT 1
8  FROM orders
9  WHERE order_id = @order_id
10     AND customer_id = @customer_id
11  LIMIT 1;
12
13  -- Додавання запису про оплату, якщо замовлення існує
14 •  INSERT INTO pay (customer_id, employee_id, order_id, date_pay, pay_sum)
15  SELECT @customer_id, 1, @order_id, '2024-11-06', '83000'
16  WHERE EXISTS (
17      SELECT 1
18      FROM orders
19      WHERE order_id = @order_id
20      AND customer_id = @customer_id
21  );
22
23 •  COMMIT;
24

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	1				
▶	1				

Result 3			
Output			Message
#	Time	Action	
1	22:52:59	START TRANSACTION	0 row(s) affected
2	22:52:59	SET @order_id = 10005	0 row(s) affected
3	22:52:59	SET @customer_id = 1	0 row(s) affected
4	22:52:59	SELECT 1 FROM orders WHERE order_id = @order_id AND customer_id = @customer_id LIMIT 1	1 row(s) returned
5	22:52:59	INSERT INTO pay (customer_id, employee_id, order_id, date_pay, pay_sum) SELECT @customer_id, 1, ..., 1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0	1 row(s) affected
6	22:52:59	COMMIT	0 row(s) affected

```
-- #7 додавання оплати по приходу
```

```
START TRANSACTION;
```

```
SET @arrival_id = 1001;
```

```
SET @customer_id = 3964;
```

```
-- Перевірка, чи існує приход для цього клієнта
```

```
SELECT 1
```

```
FROM arrivals
```

```
WHERE arrival_id = @order_id
```

```
AND customer_id = @customer_id
```

```
LIMIT 1;
```

```
-- Додавання запису про оплату, якщо приход існує
```

```
INSERT INTO pay (customer_id, employee_id, arrival_id, date_pay, pay_sum)
```

```
SELECT @customer_id, 1, @arrival_id, '2024-11-06', '83000'
```

```
WHERE EXISTS (
```

```
SELECT 1
```

```
FROM arrivals
```

```
WHERE arrival_id = @arrival_id
```

```
AND customer_id = @customer_id
```

```
);
```

```
COMMIT;
```

До виконання запиту

```
6 •  SELECT pay_id, date_pay,  
7      arrival_id, customer_id  
8      FROM pay  
9      WHERE arrival_id = 1001;
```



pay_id	date_pay	arrival_id	customer_id
NULL	NULL	NULL	NULL

Після виконання запиту

```
6 •  SELECT pay_id, date_pay,  
7      arrival_id, customer_id  
8      FROM pay  
9      WHERE arrival_id = 1001;
```



pay_id	date_pay	arrival_id	customer_id
11003	2024-11-06	1001	3964

```

2 • START TRANSACTION;
3 • SET @arrival_id = 1001;
4 • SET @customer_id = 3964;
5
6 -- Перевірка, чи існує приход для цього клієнта
7 • SELECT 1
8   FROM arrivals
9   WHERE arrival_id = @arrival_id
10  AND customer_id = @customer_id
11  LIMIT 1;
12
13 -- Добавлення запису про оплату, якщо приход існує
14 • INSERT INTO pay (customer_id, employee_id, arrival_id, date_pay, pay_sum)
15   SELECT @customer_id, 1, @arrival_id, '2024-11-06', '83000'
16   WHERE EXISTS (
17     SELECT 1
18     FROM arrivals
19     WHERE arrival_id = @arrival_id
20     AND customer_id = @customer_id
21   );
22
23 • COMMIT;
24

```

		Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	1					
	1					

Result 1			
Output			
Action Output			Message
#	Time	Action	
1	23:08:08	START TRANSACTION	0 row(s) affected
2	23:08:08	SET @arrival_id = 1001	0 row(s) affected
3	23:08:08	SET @customer_id = 3964	0 row(s) affected
4	23:08:08	SELECT 1 FROM arrivals WHERE arrival_id = @arrival_id AND customer_id = @customer_id LIMIT 1	1 row(s) returned
5	23:08:08	INSERT INTO pay (customer_id, employee_id, arrival_id, date_pay, pay_sum) SELECT @customer_id, 1, @arrival_id, '2024-11-06', '83000'	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0
6	23:08:08	COMMIT	0 row(s) affected

2.3.10.7 SQL-запитів DELETE з використанням інструкцій WHERE

-- #1 Видалення ордеру, де є тільки резерв запчастин, якщо пройшло не менше 7 днів, як створено замовлення

START TRANSACTION;

-- зберігаємо order_id в змінну

SET @order_id = 10007;

-- Перевірка умов для видалення

```

DELETE FROM orders
WHERE order_id = @order_id
    AND date_returned IS NULL -- заказ ще не повернули замовнику
    AND DATEDIFF(NOW(), date_invoice) > 7 -- минуло більше 7 днів з моменту
    створення замовлення
    AND NOT EXISTS (
        SELECT 1
        FROM orders_received
        WHERE order_id = @order_id
            AND employee_id IS NOT NULL -- техніку не було прийнято в ремонт
    )
    AND NOT EXISTS (
        SELECT 1
        FROM orders_goods
        WHERE order_id = @order_id
            AND is_shipped = 1 -- нічого не було відвантажено за цим замовленням
    )
    AND NOT EXISTS (
        SELECT 1
        FROM pay
        WHERE order_id = @order_id -- немає оплат по замовленню c order_id:
    );

```

COMMIT;

До виконання запиту

```

6 •  SELECT order_id,
7      customer_id,
8      date_invoice,
9      date_returned
10     FROM orders
11     WHERE order_id = 10007;

```

Result Grid				
Filter Rows: <input type="text"/>				
	order_id	customer_id	date_invoice	date_returned
*	10007	1	2024-11-01	NULL
*	NULL	NULL	NULL	NULL

```

1 •  SELECT order_id,
2      goods_id,
3      quantity,
4      is_shipped
5      FROM orders_goods
6      WHERE order_id = 10007;

```

Result Grid				
Filter Rows: <input type="text"/>				
	order_id	goods_id	quantity	is_shipped
▶	10007	1	1.000	0
▶	10007	2	1.000	0
▶	10007	3	2.000	0
▶	10007	5	1.000	1
*	NULL	NULL	NULL	NULL

Після виконання запиту

```

6 •   SELECT order_id,
7       customer_id,
8       date_invoice,
9       date_returned
10  FROM orders
11 WHERE order_id = 10007;
1 •   SELECT order_id,
2       goods_id,
3       quantity,
4       is_shipped
5   FROM orders_goods
6 WHERE order_id = 10007;

```

Result Grid | Filter Rows: [] | Edit: []

order_id	customer_id	date_invoice	date_returned
NULL	NULL	NULL	NULL

Result Grid | Filter Rows: [] | Edit: []

order_id	goods_id	quantity	is_shipped
NULL	NULL	NULL	NULL

```

1   -- видалення резерву запчастин, якщо пройшло не менш 7 днів, як створено замовлення
2
3 • START TRANSACTION;
4
5   -- зберігаємо order_id в змінну
6 • SET @order_id = 10007;
7
8   -- Перевірка умов для видалення
9 • DELETE FROM orders
10 WHERE order_id = @order_id
11   AND date_returned IS NULL -- замовлення ще не повернули замовнику
12   AND DATEDIFF(NOW(), date_invoice) > 7 -- минуло більше 7 днів з моменту створення замовлення
13   AND NOT EXISTS (
14     SELECT 1
15     FROM orders_received
16     WHERE order_id = @order_id
17     AND employee_id IS NOT NULL -- техніку не було прийнято в ремонт
18   )
19   AND NOT EXISTS (
20     SELECT 1
21     FROM orders_goods
22     WHERE order_id = @order_id
23     AND is_shipped = 1 -- нічого не було відвантажено за цим замовленням
24   )
25   AND NOT EXISTS (
26     SELECT 1
27     FROM pay
28     WHERE order_id = @order_id -- немає оплат по замовленню з order_id:
29   );
30
31 • COMMIT;
32

```

Output

Action Output

#	Time	Action	Message
1	13:08:56	START TRANSACTION	0 row(s) affected
2	13:08:56	SET @order_id = 10007	0 row(s) affected
3	13:08:56	DELETE FROM orders WHERE order_id = @order_id AND date_returned IS NULL -- замовлення ще не повернули замовнику	1 row(s) affected
4	13:08:56	COMMIT	0 row(s) affected

-- #2 видалити всіх "порожніх" замовників з таблиці customers

DELETE FROM customers

WHERE balance = 0

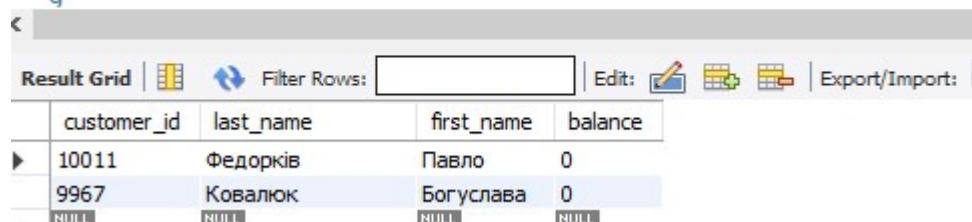
AND customer_id NOT IN (SELECT customer_id FROM orders)

AND customer_id NOT IN (SELECT customer_id FROM arrivals)

AND customer_id NOT IN (SELECT customer_id FROM pay);

До виконання запиту

```
1 •  SELECT customer_id,
2           last_name, first_name, balance
3   FROM customers
4   WHERE balance = 0
5       AND customer_id NOT IN (SELECT customer_id FROM orders)
6       AND customer_id NOT IN (SELECT customer_id FROM arrivals)
7       AND customer_id NOT IN (SELECT customer_id FROM pay)
8   ORDER BY customer_id DESC;
9
```



	customer_id	last_name	first_name	balance
▶	10011	Федорків	Павло	0
	9967	Ковалюк	Богуслава	0

```
1      -- #2 видалити всіх "порожніх" замовників з таблиці customers
2 •  START TRANSACTION;
3 •  SET SQL_SAFE_UPDATES = 0;  -- Вимкнути безпечний режим
4
5 •  DELETE FROM customers
6   WHERE balance = 0
7       AND customer_id NOT IN (SELECT customer_id FROM orders)
8       AND customer_id NOT IN (SELECT customer_id FROM arrivals)
9       AND customer_id NOT IN (SELECT customer_id FROM pay);
10
11 •  SET SQL_SAFE_UPDATES = 1;  -- Увімкнути безпечний режим назад
12 •  COMMIT;
13
```

29	22:28:40	START TRANSACTION	0 row(s) affected
30	22:28:40	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
31	22:28:40	DELETE FROM customers WHERE balance = 0 AND customer_id NOT IN (SELECT cus...	2 row(s) affected
32	22:28:40	SET SQL_SAFE_UPDATES = 1	0 row(s) affected
33	22:28:40	COMMIT	0 row(s) affected

Після виконання запиту

```

1 •  SELECT customer_id,
2                 last_name, first_name, balance
3  FROM customers
4  WHERE balance = 0
5      AND customer_id NOT IN (SELECT customer_id FROM orders)
6      AND customer_id NOT IN (SELECT customer_id FROM arrivals)
7      AND customer_id NOT IN (SELECT customer_id FROM pay)
8  ORDER BY customer_id DESC;
9

```

Result Grid				Filter Rows:	Edit:	Export/Import:
customer_id	last_name	first_name	balance			
*	NULL	NULL	NULL			

-- #3 видалити "неактивні" товари з таблиці goods_dscr по:

- які is_active = 0
- яких немає в таблиці orders_goods
- залишок goods.stock = 0

```

SET SQL_SAFE_UPDATES = 0;
DELETE FROM goods_dscr
WHERE is_active = 0
AND goods_dscr_id NOT IN (
    SELECT DISTINCT g.goods_dscr_id
    FROM goods g
    JOIN orders_goods og ON og.goods_id = g.goods_id
    WHERE g.stock > 0
);

```

До виконання запиту

```

15 •     SELECT *
16     FROM goods_dscr
17     WHERE is_active = 0
18     AND goods_dscr_id NOT IN(
19         SELECT DISTINCT g.goods_dscr_id
20         FROM goods g
21         JOIN orders_goods og ON og.goods_id = g.goods_id
22         WHERE g.stock > 0);
23

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

goods_dscr_id	unit_id	group_id	part_number	name	margin	price	description	is_active
5	6	9	OYUYDLR13H	Товар YXOCV	30	9146	Сміяться порада зро...	0

Після виконання запиту

	goods_dscr_id	unit_id	group_id	part_number	name	margin	price	description	is_active
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- #4 видалити помилково списану запчастину з orders_goods, якщо замовлення ще не повернуто замовнику

DELETE FROM orders_goods

WHERE order_id = 10011

AND goods_id = 6

AND order_id IN (

SELECT o.order_id FROM orders o

WHERE o.order_id = 10011 AND o.date_returned IS NULL);

До виконання запиту

```

6 •     SELECT order_id,
7         customer_id,
8         date_invoice,
9         date_returned
10    FROM orders
11    WHERE order_id = 10011;

```

```

1 •     SELECT order_id,
2         goods_id,
3         quantity,
4         is_shipped
5     FROM orders_goods
6     WHERE order_id = 10011;

```

Result Grid | Filter Rows: | Edit: |

order_id	customer_id	date_invoice	date_returned
10011	6638	2024-09-11	NULL

Result Grid | Filter Rows: | Edit: |

order_id	goods_id	quantity	is_shipped
10011	1	2.000	0
10011	6	1.000	0

```

1      -- #4 видалити помилково списану запчастину з orders_goods, якщо замовлення ще не повернуто замовнику
2 •  DELETE FROM orders_goods
3     WHERE order_id = 10011
4     AND goods_id = 6
5     AND order_id IN (
6         SELECT o.order_id
7         FROM orders o
8         WHERE o.order_id = 10011
9         AND o.date_returned IS NULL
10    );
11

```

Output:

Action Output

#	Time	Action	Message
1	23:17:18	DELETE FROM orders_goods WHERE order_id = 10011 AND goods_id = 6 AND order_id ...	1 row(s) affected

Після виконання запиту

```

1 •  SELECT order_id,
2                 goods_id,
3                 quantity,
4                 is_shipped
5     FROM orders_goods
6     WHERE order_id = 10011;

```

Result Grid | Filter Rows:

	order_id	goods_id	quantity	is_shipped
▶	10011	1	2.000	0
*	NULL	NULL	NULL	NULL

-- #5 видалити з таблиці orders_repair запис про ремонт, що робив конкретний співробітник, якщо date_ready, date_returned is null

```

DELETE FROM orders_repair
WHERE order_id = 10012
AND employee_id = 1
AND date_ready IS NULL
AND order_id IN (
    SELECT order_id
    FROM orders
    WHERE date_returned IS NULL
);

```

До виконання запиту

```

6 •  SELECT order_id,
7      customer_id,
8      date_invoice,
9      date_returned
10     FROM orders
11    WHERE order_id = 10012;

```

```

1 •  SELECT *
2   FROM orders_repair
3  WHERE order_id = 10012;

```

order_id	customer_id	date_invoice	date_returned	
10012	1	2024-11-06	NULL	
order_id	employee_id	date_ready	price_repair	description
10012	1	NULL	0	NULL

```

2 •  DELETE FROM orders_repair
3    WHERE order_id = 10012
4      AND employee_id = 1
5      AND date_ready IS NULL
6      AND order_id IN (
7          SELECT order_id
8            FROM orders
9           WHERE date_returned IS NULL
10        );
11

```

Output

#	Time	Action	Message
1	23:34:58	DELETE FROM orders_repair WHERE order_id = 10012 AND employee_id = 1 AND date_r... 1 row(s) affected	

Після виконання запиту

```

1 •  SELECT *
2   FROM orders_repair
3  WHERE order_id = 10012;

```

order_id	employee_id	date_ready	price_repair	description
NULL	NULL	NULL	NULL	NULL

-- #6 видалити з таблиці city невірний UNIQUE запис, якщо city_id не використан в таблиці customers

```

DELETE FROM city
WHERE name = 'Харків'
AND city_id NOT IN (
    SELECT DISTINCT city_id

```

```
FROM customers
);
```

До виконання запиту

```
1 •  SELECT *
2   FROM city
3   WHERE name = "Хар'ков";
```

Result Grid		
	city_id	name
▶	11	Хар'ков
*	NULL	NULL

Після виконання запиту

```
1 •  SELECT *
2   FROM city
3   WHERE name = "Хар'ков";
```

Result Grid		
	city_id	name
1	NULL	NULL

```
12   -- #6 видалити з таблиці city невірний UNIQUE запис, якщо city_id не використан в таблиці customers
13 •  DELETE FROM city
14   WHERE name = 'Хар'ков'
15   AND city_id NOT IN (
16     SELECT DISTINCT city_id
17     FROM customers
18   );
:
Output:
Action Output
# Time Action
1 23:41:21 DELETE FROM city WHERE name = 'Хар'ков' AND city_id NOT IN ( SELECT DISTINCT city_id FROM customers ) 1 row(s) affected
```

-- #7 Видалення застарілих даних (більш як 3 роки, крім робітників).

START TRANSACTION;

SET SQL_SAFE_UPDATES = 0; -- Вимкнути безпечний режим

DELETE FROM orders

WHERE date_returned between '2020-01-01' and '2020-12-31';

DELETE FROM arrivals

WHERE date_arrival between '2020-01-01' and '2020-12-31';

-- тільки якщо неактивний більш як 3 троки

DELETE FROM customers

```

WHERE is_active = 0
AND balance = 0
AND DATEDIFF(NOW(), update_time) > 3*366;

```

```

SET SQL_SAFE_UPDATES = 1; -- Включити безпечний режим назад
COMMIT;

```

До виконання запиту

```

6 •  SELECT order_id,
7      customer_id,
8      date_invoice,
9      date_returned
10     FROM orders
11    WHERE order_id = 10013;

```

Result Grid			
order_id	customer_id	date_invoice	date_returned
10013	1	2020-01-05	2020-01-10
NULL	NULL	NULL	NULL

```

1 •  SELECT order_id,
2      goods_id,
3      quantity,
4      is_shipped
5     FROM orders_goods
6    WHERE order_id = 10013;

```

Result Grid			
order_id	goods_id	quantity	is_shipped
10013	6	1.000	1
NULL	NULL	NULL	NULL

```

4 •  SELECT * FROM ss_innodb.pay WHERE order_id = 10013;

```

Result Grid								
pay_id	customer_id	employee_id	arrival_id	order_id	date_pay	pay_sum	pay_form	de
11004	1	1	NULL	10013	2020-01-20	30000	оплата на р.р.	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

1 •  SELECT *
2     FROM arrivals
3    WHERE arrival_id = 1001
4   ORDER BY arrival_id DESC;

```

Result Grid				
arrival_id	customer_id	employee_id	date_arrival	total
1001	3964	9	2020-02-24	312000
NULL	NULL	NULL	NULL	NULL

```

1 •  SELECT *
2     FROM arrivals_goods
3    WHERE arrival_id = 1001;

```

Result Grid		
arrival_id	goods_id	quantity
1001	4228	20.000
1001	6219	15.000

```

1 •  SELECT * FROM pay
2   WHERE arrival_id = 1001;
3
4
5

```

Result Grid								
pay_id	customer_id	employee_id	arrival_id	order_id	date_pay	pay_sum	pay_type	pay_status
11004	1	1	NULL	10013	2020-01-20	30000	0	0

```

10 •  SELECT customer_id, last_name, first_name, balance, update_time, is_active
11   FROM customers
12   WHERE customer_id = 10012
13   ORDER BY customer_id DESC;
14

```

Result Grid						
customer_id	last_name	first_name	balance	update_time	is_active	
10012	Баклан	Наталія	0	2020-02-11 00:00:00	0	

```

1   -- #7 Видалення застарілих даних (більш як 3 роки, крім робітників).
2 •  START TRANSACTION;
3 •  SET SQL_SAFE_UPDATES = 0;  -- Вимкнути безпечний режим
4
5 •  DELETE FROM orders
6   WHERE date_returned between '2020-01-01' and '2020-12-31';
7
8 •  DELETE FROM arrivals
9   WHERE date_arrival between '2020-01-01' and '2020-12-31';
10
11  -- тільки якщо неактивний більш як 3 троки
12 •  DELETE FROM customers
13   WHERE is_active = 0
14     AND balance = 0
15     AND DATEDIFF(NOW(), update_time) > 3*366;
16
17 •  SET SQL_SAFE_UPDATES = 1;  -- Включити безпечний режим назад
18 •  COMMIT;

```

Output ::::::::::::

Action Output

#	Time	Action	Message
1	00:30:59	START TRANSACTION	0 row(s) affected
2	00:30:59	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
3	00:30:59	DELETE FROM orders WHERE date_returned between '2020-01-01' and '2020-12-31'	1 row(s) affected
4	00:30:59	DELETE FROM arrivals WHERE date_arrival between '2020-01-01' and '2020-12-31'	1 row(s) affected
5	00:30:59	DELETE FROM customers WHERE is_active = 0 AND balance = 0 AND DATEDIFF(NOW(), update_time) > 3*366;	1 row(s) affected
6	00:30:59	SET SQL_SAFE_UPDATES = 1	0 row(s) affected
7	00:30:59	COMMIT	0 row(s) affected

Після виконання запиту

```
6 •   SELECT order_id,  
7           customer_id,  
8           date_invoice,  
9           date_returned  
10      FROM orders  
11      WHERE order_id = 10013;
```

```
1 •   SELECT order_id,  
2                 goods_id,  
3                 quantity,  
4                 is_shipped  
5   FROM orders_goods  
6   WHERE order_id = 10013;
```

Result Grid				
	order_id	customer_id	date_invoice	date_returned
	NULL	NULL	NULL	NULL

result Grid |   Filter Rows:

order_id	goods_id	quantity	is_shipped
NULL	NULL	NULL	NULL

```
7 •  SELECT * FROM ss_innodb.pay WHERE order_id = 10013;
```

```
1 •   SELECT *
2     FROM arrivals
3    WHERE arrival_id = 1001
4 ORDER BY arrival_id DESC
```

```
1 •  SELECT *
2      FROM arrivals_goods
3      WHERE arrival_id = 1001;
```

Result Grid					
	arrival_id	customer_id	employee_id	date_arrival	total
*	NULL	NULL	NULL	NULL	NULL

Result Grid			  Filter Rows: <input type="text"/>
arrival_id	goods_id	quantity	
NULL	NULL	NULL	

```
1 •  SELECT *
2      FROM pay
3      WHERE arrival_id = 1001;
4
5
```

```

10 •  SELECT customer_id, last_name, first_name, balance, update_time, is_active
11   FROM customers
12   WHERE customer_id = 10012
13   ORDER BY customer_id DESC;
**
```

The screenshot shows the MySQL Workbench interface with a 'Result Grid' tab. The grid displays a single row of data with the following columns and values:

customer_id	last_name	first_name	balance	update_time	is_active
NULL	NULL	NULL	NULL	NULL	NULL

2.3.11. 5 SQL-запитів (у тому числі складених), що відповідають специфіці високонавантажених систем.

#1 Додавання нового приходу запчастин

START TRANSACTION;

-- 1. Додаємо новий приход в таблицю arrivals

```
INSERT INTO arrivals (customer_id, employee_id, date_arrival, total)
VALUES (1, 1, '2024-11-06', 260000);
```

-- 2. Отримуємо arrival_id останнього вставленого приходу

```
SET @arrival_id = LAST_INSERT_ID();
```

-- 3. Додаємо записи у таблицю arrivals_goods, використовуючи отриманий arrival_id

```
INSERT INTO arrivals_goods (arrival_id, goods_id, quantity)
VALUES (@arrival_id, 1, 5),
       (@arrival_id, 2, 10),
       (@arrival_id, 3, 20),
       (@arrival_id, 4, 20),
       (@arrival_id, 5, 20);
```

COMMIT;

До виконання запиту

```

1 •  SELECT arrival_id
2   FROM arrivals
3   ORDER BY arrival_id DESC;
4
5

```

Result Grid	
	arrival_id
▶	1002
▶	1000
▶	999

```

1 •  SELECT *
2   FROM arrivals_goods
3   ORDER BY arrival_id DESC;
4
5

```

	arrival_id	goods_id	quantity
▶	1000	9262	24.000
▶	1000	6456	23.000
▶	1000	5985	23.000
▶	1000	4694	29.000
▶	1000	3697	21.000
...

Після виконання запиту

```

1 •  SELECT arrival_id
2   FROM arrivals
3   ORDER BY arrival_id DESC;
4
5

```

Result Grid	
	arrival_id
▶	1003
▶	1002
▶	1000
▶	999

```

1 •  SELECT *
2   FROM arrivals_goods
3   ORDER BY arrival_id DESC;
4
5

```

	arrival_id	goods_id	quantity
▶	1003	6	20.000
▶	1003	4	20.000
▶	1003	3	20.000
▶	1003	2	10.000
▶	1003	1	5.000
▶	1000	9262	24.000
...

Запчастини на складі до та після запиту

```

8 •  SELECT goods_id, stock, reserve
9   FROM ss_innodb.goods
10  WHERE goods_id IN (1,2,3,4,6);

```

Result Grid		
goods_id	stock	reserve
1	9.000	5.000
2	2.000	2.000
3	1.000	1.000
4	5.000	0.000
6	2.000	0.000
NULL	NULL	NULL

```

8 •  SELECT goods_id, stock, reserve
9   FROM ss_innodb.goods
10  WHERE goods_id IN (1,2,3,4,6);

```

Result Grid		
goods_id	stock	reserve
1	14.000	5.000
2	12.000	2.000
3	21.000	1.000
4	25.000	0.000
6	22.000	0.000
NULL	NULL	NULL

```

2 •  START TRANSACTION;
3
4      -- 1. Додаємо новий приход в таблицю arrivals
5 •  INSERT INTO arrivals (customer_id, employee_id, date_arrival, total)
6      VALUES (1, 1, '2024-11-06', 260000);
7
8      -- 2. Отримуємо arrival_id останнього вставленого приходу
9 •  SET @arrival_id = LAST_INSERT_ID();
10
11     -- 3. Додаємо записи у таблицю arrivals_goods, використовуючи отриманий arrival_id
12 •  INSERT INTO arrivals_goods (arrival_id, goods_id, quantity)
13      VALUES (@arrival_id, 1, 5),
14          (@arrival_id, 2, 10),
15          (@arrival_id, 3, 20),
16          (@arrival_id, 4, 20),
17          (@arrival_id, 6, 20);
18
19 •  COMMIT;

```

Action Output			Message
#	Time	Action	
1	17:27:00	START TRANSACTION	0 row(s) affected
2	17:27:00	INSERT INTO arrivals (customer_id, employee_id, date_arrival, total) VALUES (1, 1, '2024-11-06', 260...	1 row(s) affected
3	17:27:00	SET @arrival_id = LAST_INSERT_ID()	0 row(s) affected
4	17:27:00	INSERT INTO arrivals_goods (arrival_id, goods_id, quantity) VALUES (@arrival_id, 1, 5), (@arrival_id,...	5 row(s) affected Records: 5
5	17:27:00	COMMIT	0 row(s) affected

#2 Редагування приходу запчастин, якщо не readonly, у розрізі приходування на склад

```

UPDATE arrivals_goods ag
JOIN arrivals a ON ag.arrival_id = a.arrival_id
SET ag.goods_id = 1,
    ag.quantity = 10
WHERE ag.arrival_id = 1
    AND ag.goods_id = 139
    AND a.read_only = 0;

```

До виконання запиту

```

1 •  SELECT *
2   FROM arrivals_goods
3   WHERE arrival_id = 1;

```

	arrival_id	goods_id	quantity
▶	1	139	16.000
	1	2005	23.000
	1	2686	27.000
	1	3457	18.000
	1	4729	19.000
	1	4940	17.000
	1	6461	11.000
*	NULL	NULL	NULL

Після виконання запиту

```

1 •  SELECT *
2   FROM arrivals_goods
3   WHERE arrival_id = 1;

```

	arrival_id	goods_id	quantity
	1	1	10.000
	1	2005	23.000
	1	2686	27.000
	1	3457	18.000
	1	4729	19.000
	1	4940	17.000
	1	6461	11.000
	NULL	NULL	NULL

```

3 •  UPDATE arrivals_goods ag
4   JOIN arrivals a ON ag.arrival_id = a.arrival_id
5   SET ag.goods_id = 1,                                -- новое значение goods_id
6       ag.quantity = 10,                                -- новое значение quantity
7   WHERE ag.arrival_id = 1,                            -- arrival_id, по которому производится замена
8       AND ag.goods_id = 139,                          -- текущее значение goods_id, которое меняется
9       AND a.read_only = 0;
10

```

Output		
#	Time	Action
1	12:51:08	UPDATE arrivals_goods ag JOIN arrivals a ON ag.arrival_id = a.arrival_id SET ag.goods_id = 1, -- но... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Використання EXPLAIN для аналізу цього запиту

```

2 •  EXPLAIN
3   UPDATE arrivals_goods ag
4   JOIN arrivals a ON ag.arrival_id = a.arrival_id
5   SET ag.goods_id = 1,
6       ag.quantity = 11
7   WHERE ag.arrival_id = 1
8       AND ag.goods_id = 1
9       AND a.read_only = 0;

```

Result Grid Filter Rows: [] Export: [] Wrap Cell Content: []										rows	filtered	Extra
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	UPDATE	ag	NULL	const	PRIMARY, fk_arrival_goods_arrivals1_idx, fk_arr...	PRIMARY	8	const,const	1	100.00	NULL
1	1	SIMPLE	a	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL

Бачимо, що запит працює оптимальним образом, считуючи по одному рядку з кожної таблиці.

#3 Додавання запчастин зі складу в існуюче замовлення

START TRANSACTION;

-- отримуємо дані для транзакції

SET @order_id = 1;

SET @goods_id = 1;

SET @quantity = 1;

-- Перевірка: якщо запчастина вже в замовленні, то завершити транзакцію

SELECT 1

FROM orders_goods

WHERE order_id = @order_id AND goods_id = @goods_id

LIMIT 1;

-- Додавання запчастини, якщо її немає в замовленні

INSERT INTO orders_goods (order_id, goods_id, quantity)

SELECT @order_id, @goods_id, @quantity

WHERE NOT EXISTS (

SELECT 1

FROM orders_goods

WHERE order_id = @order_id

AND goods_id = @goods_id

);

COMMIT;

До виконання запиту

Після виконання запиту

```

1 •  SELECT order_id,
2          goods_id,
3          quantity,
4          is_shipped
5  FROM orders_goods
6 WHERE order_id = 1;

```

Result Grid			
order_id	goods_id	quantity	is_shipped
1	886	1.834	1
1	5216	0.389	1
1	8568	0.500	1
NULL	NULL	NULL	NULL

```

1 •  SELECT order_id,
2          goods_id,
3          quantity,
4          is_shipped
5  FROM orders_goods
6 WHERE order_id = 1;

```

Result Grid			
order_id	goods_id	quantity	is_shipped
1	1	1.000	1
1	886	1.834	1
1	5216	0.389	1
1	8568	0.500	1
NULL	NULL	NULL	NULL

```

1      -- Добавлення запчастини (по goods_id) зі складу в існуюче замовлення:
2 •  START TRANSACTION;
3
4      -- отримуємо дані для транзакції
5 •  SET @order_id = 1;
6 •  SET @goods_id = 1;
7 •  SET @quantity = 1;
8
9      -- Перевірка: якщо запчастина вже в замовленні, то завершити транзакцію
10 •  SELECT 1
11     FROM orders_goods
12     WHERE order_id = @order_id AND goods_id = @goods_id
13     LIMIT 1;
14
15      -- Добавлення запчастини, якщо її немає в замовленні
16 •  INSERT INTO orders_goods (order_id, goods_id, quantity)
17     SELECT @order_id, @goods_id, @quantity
18     WHERE NOT EXISTS (
19         SELECT 1
20         FROM orders_goods
21         WHERE order_id = @order_id
22             AND goods_id = @goods_id
23     );
24
25 •  COMMIT;
26

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:																																
	1																																				
Result 1 x																																					
Output ::::::::::::::::::::																																					
Action Output																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Time</th> <th>Action</th> <th>Message</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>11:50:42</td> <td>START TRANSACTION</td> <td>0 row(s) affected</td> </tr> <tr> <td>2</td> <td>11:50:42</td> <td>SET @order_id = 1</td> <td>0 row(s) affected</td> </tr> <tr> <td>3</td> <td>11:50:42</td> <td>SET @goods_id = 1</td> <td>0 row(s) affected</td> </tr> <tr> <td>4</td> <td>11:50:42</td> <td>SET @quantity = 1</td> <td>0 row(s) affected</td> </tr> <tr> <td>5</td> <td>11:50:42</td> <td>SELECT 1 FROM orders_goods WHERE order_id = @order_id AND goods_id = @goods_id LIMIT 1</td> <td>0 row(s) returned</td> </tr> <tr> <td>6</td> <td>11:50:42</td> <td>INSERT INTO orders_goods (order_id, goods_id, quantity) SELECT @order_id, @goods_id, @quant...</td> <td>1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0</td> </tr> <tr> <td>7</td> <td>11:50:42</td> <td>COMMIT</td> <td>0 row(s) affected</td> </tr> </tbody> </table>						#	Time	Action	Message	1	11:50:42	START TRANSACTION	0 row(s) affected	2	11:50:42	SET @order_id = 1	0 row(s) affected	3	11:50:42	SET @goods_id = 1	0 row(s) affected	4	11:50:42	SET @quantity = 1	0 row(s) affected	5	11:50:42	SELECT 1 FROM orders_goods WHERE order_id = @order_id AND goods_id = @goods_id LIMIT 1	0 row(s) returned	6	11:50:42	INSERT INTO orders_goods (order_id, goods_id, quantity) SELECT @order_id, @goods_id, @quant...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0	7	11:50:42	COMMIT	0 row(s) affected
#	Time	Action	Message																																		
1	11:50:42	START TRANSACTION	0 row(s) affected																																		
2	11:50:42	SET @order_id = 1	0 row(s) affected																																		
3	11:50:42	SET @goods_id = 1	0 row(s) affected																																		
4	11:50:42	SET @quantity = 1	0 row(s) affected																																		
5	11:50:42	SELECT 1 FROM orders_goods WHERE order_id = @order_id AND goods_id = @goods_id LIMIT 1	0 row(s) returned																																		
6	11:50:42	INSERT INTO orders_goods (order_id, goods_id, quantity) SELECT @order_id, @goods_id, @quant...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0																																		
7	11:50:42	COMMIT	0 row(s) affected																																		

#4 Додавання нового замовлення с резервуванням запчастин

START TRANSACTION;

-- 1. Додаємо нове замовлення в таблицю orders

```
INSERT INTO orders (customer_id, date_invoice)
VALUES ('1', '2024-11-06');
```

-- 2. Отримуємо order_id останнього вставленого замовлення

```
SET @order_id = LAST_INSERT_ID();
```

-- 3. Додаємо записи у таблицю orders_goods, резервуємо запчастини (is_shipped=0)

```
INSERT INTO orders_goods (order_id, goods_id, quantity, is_shipped)
VALUES (@order_id, 1, 1, 0),
       (@order_id, 2, 1, 0),
       (@order_id, 3, 1, 0);
```

COMMIT;

До виконання запиту

```
1 •  SELECT order_id, customer_id,
2      date_invoice, date_returned
3      FROM orders
4      ORDER BY order_id desc
5      LIMIT 2;
```

Result Grid				
	order_id	customer_id	date_invoice	date_returned
▶	10012	1	2024-11-06	NULL
▶	10011	6638	2024-09-11	NULL
*	NULL	NULL	NULL	NULL

```
1 •  SELECT order_id,
2      goods_id,
3      quantity,
4      is_shipped
5      FROM orders_goods
6      ORDER BY order_id DESC;
```

Result Grid				
	order_id	goods_id	quantity	is_shipped
▶	10011	1	2.000	0
▶	10009	3	1.000	1

Після виконання запиту

```
1 •  SELECT order_id, customer_id,
2      date_invoice, date_returned
3      FROM orders
4      ORDER BY order_id desc
5      LIMIT 2;
```

Result Grid				
	order_id	customer_id	date_invoice	date_ref
▶	10014	1	2024-11-06	NULL
▶	10012	1	2024-11-06	NULL
*	NULL	NULL	NULL	NULL

```
1 •  SELECT order_id,
2      goods_id,
3      quantity,
4      is_shipped
5      FROM orders_goods
```

Result Grid				
	order_id	goods_id	quantity	is_shipped
▶	10014	3	1.000	0
▶	10014	2	1.000	0
▶	10014	1	1.000	0
▶	10011	1	2.000	0

Запчастини на складі до та після запиту

```
8 •  SELECT goods_id, stock, reserve
9      FROM ss_innodb.goods
10     WHERE goods_id IN (1,2,3);
```

Result Grid			
	goods_id	stock	reserve
▶	1	14.000	5.000
▶	2	12.000	2.000
▶	3	21.000	1.000
*	NULL	NULL	NULL

```
8 •  SELECT goods_id, stock, reserve
9      FROM ss_innodb.goods
10     WHERE goods_id IN (1,2,3);
```

Result Grid			
	goods_id	stock	reserve
▶	1	14.000	6.000
▶	2	12.000	3.000
▶	3	21.000	2.000
*	NULL	NULL	NULL

```

1      -- #4 Добавлення нового замовлення з резервуванням запчастин
2 •  START TRANSACTION;
3
4      -- 1. Додаємо нове замовлення в таблицю orders
5 •  INSERT INTO orders (customer_id, date_invoice)
6      VALUES ('1', '2024-11-06');
7
8      -- 2. Отримуємо order_id останнього вставленого замовлення
9 •  SET @order_id = LAST_INSERT_ID();
10
11     -- 3. Додаємо записи у таблицю orders_goods, резервуємо запчастини (is_shipped=0)
12 •  INSERT INTO orders_goods (order_id, goods_id, quantity, is_shipped)
13      VALUES (@order_id, 1, 1, 0),
14          (@order_id, 2, 1, 0),
15          (@order_id, 3, 1, 0);
16
17 •  COMMIT;
18

```

Action Output			Message
#	Time	Action	
1	17:47:09	START TRANSACTION	0 row(s) affected
2	17:47:09	INSERT INTO orders (customer_id, date_invoice) VALUES ('1', '2024-11-06')	1 row(s) affected
3	17:47:09	SET @order_id = LAST_INSERT_ID()	0 row(s) affected
4	17:47:09	INSERT INTO orders_goods (order_id, goods_id, quantity, is_shipped) VALUES (@order_id, 1, 1, 0), (@order_id, 2, 1, 0), (@order_id, 3, 1, 0)	3 row(s) affected Records: 3
5	17:47:09	COMMIT	0 row(s) affected

Використання EXPLAIN для аналізу цього запиту немає сенсу.

result Grid											
Filter Rows: <input type="text"/> Export: Wrap Cell Content:											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	INSERT	orders	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#5 Редагування замовлень, якщо не readonly, у розрізі списання зі складу

START TRANSACTION;

```

UPDATE orders_goods og
JOIN orders a ON og.order_id = a.order_id
SET og.goods_id = 4,

```

```

og.quantity = 2
WHERE og.order_id = 10014
AND og.goods_id = 3
AND o.read_only = 0
AND date_returned is null;

```

COMMIT;

До виконання запиту

```

1 •  SELECT order_id,
2          goods_id,
3          quantity,
4          is_shipped
5  FROM orders_goods
6 WHERE order_id = 10014;

```

result Grid | Filter Rows:

order_id	goods_id	quantity	is_shipped
10014	1	1.000	0
10014	2	1.000	0
10014	3	1.000	0
NONE	NONE	NONE	NONE

result Grid | Filter Rows:

order_id	goods_id	quantity	is_shipped
10014	1	1.000	0
10014	2	1.000	0
10014	3	1.000	0
NONE	NONE	NONE	NONE

Після виконання запиту

```

1 •  SELECT order_id,
2          goods_id,
3          quantity,
4          is_shipped
5  FROM orders_goods
6 WHERE order_id = 10014;

```

result Grid | Filter Rows:

order_id	goods_id	quantity	is_shipped
10014	1	1.000	0
10014	2	1.000	0
10014	4	2.000	0
NONE	NONE	NONE	NONE

result Grid | Filter Rows:

order_id	goods_id	quantity	is_shipped
10014	1	1.000	0
10014	2	1.000	0
10014	4	2.000	0
NONE	NONE	NONE	NONE

Запчастини на складі до та після запиту

```

8 •  SELECT goods_id, stock, reserve
9   FROM ss_innodb.goods
10  WHERE goods_id IN (1,2,3,4);

```

Result Grid		
goods_id	stock	reserve
1	14.000	6.000
2	12.000	3.000
3	21.000	2.000
4	25.000	0.000
NULL	NULL	NULL

```

8 •  SELECT goods_id, stock, reserve
9   FROM ss_innodb.goods
10  WHERE goods_id IN (1,2,3,4);

```

Result Grid		
goods_id	stock	reserve
1	14.000	6.000
2	12.000	3.000
3	21.000	1.000
4	25.000	2.000
NULL	NULL	NULL

```

4 •  START TRANSACTION;
5
6 •  UPDATE orders_goods og
7   JOIN orders o ON og.order_id = o.order_id
8   SET og.goods_id = 4,          -- нове значення goods_id
9      og.quantity = 2          -- нове значення quantity
10  WHERE og.order_id = 10014    -- order_id, за яким проводиться заміна
11  AND og.goods_id = 3          -- поточне значення goods_id, яке змінюється
12  AND read_only = 0
13  AND date_returned is null;
14
15 •  COMMIT;

```

Output		
Action Output		
#	Time	Action
1	18:38:23	START TRANSACTION
2	18:38:23	UPDATE orders_goods og JOIN orders o ON og.order_id = o.order_id SET og.goods_id = 4, -- нове ... 1 row(s) affected Rows matched: 1 Changed: 1
3	18:38:23	COMMIT

Використаємо EXPLAIN для аналізу цього запиту

```

5 •  EXPLAIN
6   UPDATE orders_goods og
7     JOIN orders o ON og.order_id = o.order_id
8     SET price_out = 15000,
9        og.goods_id = 6,          -- нове значення goods_id
10       og.quantity = 3          -- нове значення quantity
11  WHERE og.order_id = 10014    -- order_id, за яким проводиться заміна
12  AND og.goods_id = 2          -- поточне значення goods_id, яке змінюється
13  AND read_only = 0
14  AND date_returned is null;

```

Result Grid											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	UPDATE	og	NULL	const	PRIMARY, fk_goods_repair_goods1_idx	PRIMARY	8	const,const	1	100.00	NULL
1	SIMPLE	o	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL

Бачимо, що запит працює оптимальним образом, считуючи по одному рядку з кожної таблиці.

З точки зори високонавантажених систем має сенс розглянути також наступні запити (які вже були наведені раніше):

#6 Підрахунок суми замовлення - order_total

```
SELECT o.order_id,
       ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) +
COALESCE(repairs.total_repair, 0)) AS order_total
FROM orders o
LEFT JOIN orders_goods og ON o.order_id = og.order_id
LEFT JOIN (
    SELECT order_id, SUM(price_repair) AS total_repair
    FROM orders_repair
    GROUP BY order_id
) AS repairs ON o.order_id = repairs.order_id
GROUP BY o.order_id
ORDER BY o.order_id DESC;
```

Використаємо EXPLAIN для аналізу цього запиту

```
4 •  EXPLAIN
5   SELECT o.order_id,
6     ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair, 0)) AS order_total
7   FROM orders o
8   LEFT JOIN orders_goods og ON o.order_id = og.order_id
9   LEFT JOIN (
10     SELECT order_id, SUM(price_repair) AS total_repair
11     FROM orders_repair
12     GROUP BY order_id
13   ) AS repairs ON o.order_id = repairs.order_id
14   GROUP BY o.order_id
15   ORDER BY o.order_id DESC;
```

Result Grid												
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	PRIMARY	o	HULL	index	PRIMARY, fk_orders_customers_idx, fk_orders_delive...	PRIMARY	4	NULL	9632	100.00	Backward index scan; Using index
1	1	PRIMARY	og	HULL	ref	PRIMARY	PRIMARY	4	ss_innodb.o.order_id	1	100.00	NULL
1	1	PRIMARY	<derived2>	HULL	ref	<auto_key0>	<auto_key0>	4	ss_innodb.o.order_id	10	100.00	NULL
2	2	DERIVED	orders_repair	HULL	index	PRIMARY, fk_orders_repair_orders1_idx, fk_orders_r...	PRIMARY	5	NULL	7751	100.00	NULL

Бачимо, що запит не оптимальний, можливо його оптимізувати, додавши умову WHERE o.order_id = 1, якщо це влаштовує.

```

17 •  EXPLAIN
18   SELECT o.order_id,
19     ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair, 0)) AS order_total
20   FROM orders o
21   LEFT JOIN orders_goods og ON o.order_id = og.order_id
22   LEFT JOIN (
23     SELECT order_id, SUM(price_repair) AS total_repair
24     FROM orders_repair
25     WHERE order_id = 1
26     GROUP BY order_id
27   ) AS repairs ON o.order_id = repairs.order_id
28   WHERE o.order_id = 1
29   GROUP BY o.order_id
30   ORDER BY o.order_id DESC;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	o	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using index
1	PRIMARY	og	NULL	ref	PRIMARY	PRIMARY	4	const	4	100.00	NULL
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	1	100.00	Using where; Using join buffer (hash join)
2	DERIVED	orders_repair	NULL	ref	PRIMARY, fk_orders_repair_orders1_idx, fk_orders_r...	PRIMARY	4	const	2	100.00	NULL

#7 картка замовника з групуванням та балансом

```

SELECT c.last_name,
       c.first_name,
       o.order_id,
       o.date_invoice,
       o.date_returned,
       COALESCE(order_totals.order_total, 0) AS order_total,
       COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
       (COALESCE(order_totals.order_total, 0) -
       COALESCE(pay_totals.total_pay_sum, 0)) AS balance
  FROM orders o
 JOIN customers c USING(customer_id)
 LEFT JOIN (
   SELECT order_id, SUM(pay_sum) AS total_pay_sum
   FROM pay
   WHERE customer_id = 1
   GROUP BY order_id
 ) AS pay_totals ON o.order_id = pay_totals.order_id
 LEFT JOIN (
   SELECT o.order_id,

```

```

        ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) +
COALESCE(repairs.total_repair, 0)) AS order_total
FROM orders o
LEFT JOIN orders_goods og ON o.order_id = og.order_id
LEFT JOIN (
    SELECT order_id, SUM(price_repair) AS total_repair
    FROM orders_repair
    GROUP BY order_id
) AS repairs ON o.order_id = repairs.order_id
GROUP BY o.order_id
) AS order_totals ON o.order_id = order_totals.order_id
WHERE o.customer_id = 1
AND date_invoice BETWEEN '2024-01-01' AND '2024-09-30'
ORDER BY date_invoice;

```

Використаємо EXPLAIN для аналізу цього запиту

```

43 • EXPLAIN
44   SELECT c.last_name,
45       c.first_name,
46       o.order_id,
47       o.date_invoice,
48       o.date_returned,
49       COALESCE(order_totals.order_total, 0) AS order_total,
50       COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
51       (COALESCE(order_totals.order_total, 0) - COALESCE(pay_totals.total_pay_sum, 0)) AS balance
52   FROM orders o
53   JOIN customers c USING(customer_id)
54   LEFT JOIN (
55       SELECT order_id, SUM(pay_sum) AS total_pay_sum
56       FROM pay
57       WHERE customer_id = 1
58       GROUP BY order_id
59   ) AS pay_totals ON o.order_id = pay_totals.order_id
60   LEFT JOIN (
61       SELECT o.order_id,
62           ROUND(COALESCE(SUM(og.price_out * og.quantity), 0) + COALESCE(repairs.total_repair, 0)) AS order_total
63       FROM orders o
64       LEFT JOIN orders_goods og ON o.order_id = og.order_id
65       LEFT JOIN (
66           SELECT order_id, SUM(price_repair) AS total_repair
67           FROM orders_repair
68           GROUP BY order_id
69       ) AS repairs ON o.order_id = repairs.order_id
70       GROUP BY o.order_id
71   ) AS order_totals ON o.order_id = order_totals.order_id
72   WHERE o.customer_id = 1
73   AND date_invoice BETWEEN '2024-09-01' AND '2024-09-30'
74   ORDER BY date_invoice;

```

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	c		NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using filesort
1	PRIMARY	o		NULL	ref	fk_orders_customers_idx	fk_orders_customers_idx	4	const	8	11.11	Using where
1	PRIMARY	<derived2>		NULL	ref	<auto_key0>	<auto_key0>	4	ss_innodb.o.order_id	2	100.00	NULL
1	PRIMARY	<derived3>		NULL	ref	<auto_key0>	<auto_key0>	4	ss_innodb.o.order_id	19	100.00	NULL
3	DERIVED	o		NULL	index	PRIMARY, fk_orders_customers_idx, fk_orders_deliv...	PRIMARY	4	NULL	9632	100.00	Using index
3	DERIVED	og		NULL	ref	PRIMARY	PRIMARY	4	ss_innodb.o.order_id	1	100.00	NULL
3	DERIVED	<derived4>		NULL	ref	<auto_key0>	<auto_key0>	4	ss_innodb.o.order_id	10	100.00	NULL
4	DERIVED	orders_repair		NULL	index	PRIMARY, fk_orders_repair_orders1_idx, fk_orders_r...	PRIMARY	5	NULL	7751	100.00	NULL
2	DERIVED	pay		NULL	ref	fk_pay_customers1_idx, fk_pay_orders1_idx	fk_pay_customers1_idx	4	const	3	100.00	Using temporary

Бачимо, що запит не оптимальний, считає дві таблиці повністю, orders та orders_repair. Кардинальне рішення цього питання – додати поле order_total до таблиці orders, та заповнювати його за допомогою тригера.

#8 картка постачальника з групуванням по оплатам та балансом

```
SELECT a.arrival_id,
       a.customer_id,
       c.last_name,
       c.first_name,
       a.total as arrival_total,
       COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
       (a.total - COALESCE(pay_totals.total_pay_sum, 0)) AS balance
  FROM arrivals a
 JOIN customers c ON a.customer_id = c.customer_id
 LEFT JOIN (
    SELECT arrival_id, SUM(pay_sum) AS total_pay_sum
      FROM pay
     WHERE customer_id = 3964
     GROUP BY arrival_id
  ) AS pay_totals ON a.arrival_id = pay_totals.arrival_id
 WHERE (a.date_arrival BETWEEN '2024-09-01' AND '2024-09-30' OR
       a.date_arrival is null)
       AND a.customer_id = 3964;
```

Використаємо EXPLAIN для аналізу цього запиту

```

77 • EXPLAIN
78  SELECT a.arrival_id,
79      a.customer_id,
80      c.last_name,
81      c.first_name,
82      a.total AS arrival_total,
83      COALESCE(pay_totals.total_pay_sum, 0) AS pay_total,
84      (a.total - COALESCE(pay_totals.total_pay_sum, 0)) AS balance
85  FROM arrivals a
86  JOIN customers c ON a.customer_id = c.customer_id
87  LEFT JOIN (
88      SELECT arrival_id, SUM(pay_sum) AS total_pay_sum
89      FROM pay
90      WHERE customer_id = 3964
91      GROUP BY arrival_id
92  ) AS pay_totals ON a.arrival_id = pay_totals.arrival_id
93  WHERE (a.date_arrival BETWEEN '2024-09-01' AND '2024-09-30' OR a.date_arrival IS NULL)
94  AND a.customer_id = 3964;

```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	c	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL
1	PRIMARY	a	NULL	ref	fk_arrivals_customers1_idx	fk_arrivals_customers1_idx	4	const	2	20.00	Using where
1	PRIMARY	<derived2>	NULL	ref	<auto_key0>	<auto_key0>	5	ss_innodb.a.arrival_id	2	100.00	NULL
2	DERIVED	pay	NULL	ref	fk_pay_customers1_idx,fk_pay_arrivals1_idx	fk_pay_customers1_idx	4	const	1	100.00	Using temporary

На відміну від попереднього цей запит набагато оптимальніший завдяки тому, що не треба підраховувати arrival_total, це поле вже є в таблиці arrivals.

#9 Каталог товарів, із зазначенням кількості товару на складі, в резерві, ціна за одиницю, с сортировкою по групам

Використаємо EXPLAIN для аналізу цього запиту

```

30  -- каталог замовленого товару (catalog_name), кількості товару на складі (quantity + reserve),
31  -- ціна за одиницю (price), с сортировкою по групам
32 • EXPLAIN
33  SELECT gd.name,
34      gd.price,
35      g.stock,
36      (SELECT SUM(quantity)
37      FROM orders_goods og
38      WHERE is_shipped = 0 AND og.goods_id = g.goods_id
39      GROUP BY goods_id) AS goods_reserve,
40      gr.name
41  FROM goods g, goods_dscr gd, `group` gr
42  WHERE g.goods_dscr_id = gd.goods_dscr_id
43  AND gd.group_id = gr.group_id
44  ORDER BY gr.name;

```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	gr	NULL	ALL	PRIMARY	NULL	NULL	NULL	13	100.00	Using filesort
1	PRIMARY	gd	NULL	ref	PRIMARY,fk_goods_dscr_group1_idx	fk_goods_dscr_group1_idx	2	ss_innodb.gr.group_id	739	100.00	NULL
1	PRIMARY	g	NULL	ref	fk_goods_goods_dscr1_idx	fk_goods_goods_dscr1_idx	4	ss_innodb.gd.goods_dscr_id	1	100.00	NULL
2	DEPENDENT SUBQUERY	og	NULL	ref	PRIMARY,fk_goods_repair_goods1_idx	fk_goods_repair_goods1_idx	4	ss_innodb.g.goods_id	2	10.00	Using where

Я оптимізував цей запит, додавши поле reserve до таблиці goods

Висновок

Відносно запитів, що відповідають специфіці високонавантажених систем, на мою думку, тут треба розглядати запити, які одночасно можуть

звертатись на запис чи читання до БД, та впливати на вибір даних, створювати конфлікти, особливо це стосується склада запчастин, поля `stock` та `reserve`. Тому я приділив богато уваги цим запитам та спробував їх працю з тригерами, де використав блокування конкретної записі перед редагуванням даних. Більш подробно про це в наступній лабораторній роботі.

Якщо використовуємо (а я використовую) каскадне видалення даних, треба врахувати, що тригери на видалення не будуть працювати.