Лабораторна робота № 3. Побудова простих запитів.

Виконавець: ІП-11 Прищепа Владислав Станіславович

Варіант: 25

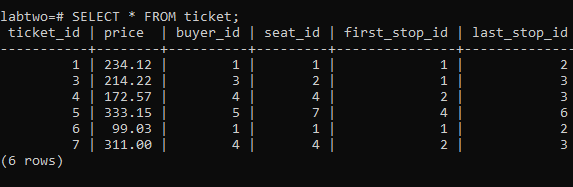
Мета: − Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць

− Вивчити команди SQL для створення запитів з використанням підзапитів та з’єднань

1) Створити запити для вибірки даних з використанням (разом 12 запитів):

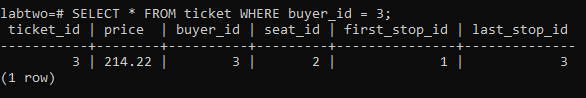
a. Найпростіших умов

Початкова таблиця ticket:



Виведемо рядки з таблиці ticket, де buyer\_id рівен 3:

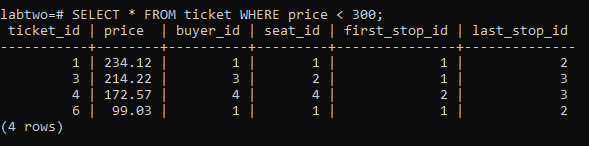
SELECT \* FROM ticket WHERE buyer\_id = 3;



b. Операторів порівняння

Виведемо рядки з таблиці ticket, де price < 300:

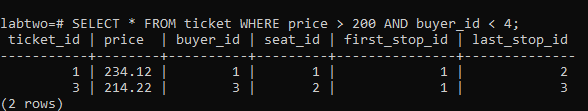
SELECT \* FROM ticket WHERE price < 300;



c. Умов з використанням логічних операторів AND, OR та NOT.

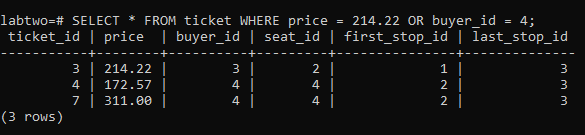
Виведемо рядки з таблиці ticket, де price > 200 та buyer\_id < 4:

SELECT \* FROM ticket WHERE price > 200 AND buyer\_id < 4;



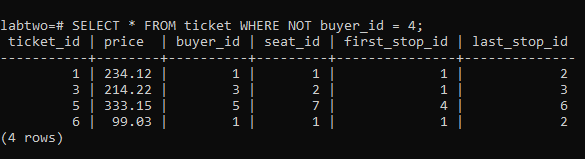
Виведемо рядки з таблиці ticket, де price = 214.22 чи buyer\_id = 4:

SELECT \* FROM ticket WHERE price = 214.22 OR buyer\_id = 4;



Виведемо рядки з таблиці ticket, де buyer\_id не рівен 4:

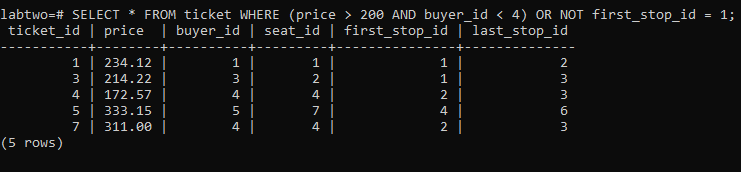
SELECT \* FROM ticket WHERE NOT buyer\_id = 4;



d. Умов з використанням комбінацій логічних операторів

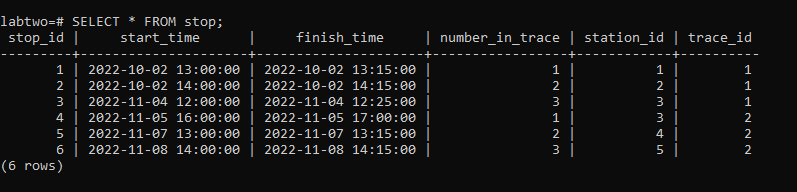
Виведемо рядки з таблиці ticket, де price > 200 та buyer\_id < 4, або first\_stop\_id не рівен 1:

SELECT \* FROM ticket WHERE (price > 200 AND buyer\_id < 4) OR NOT first\_stop\_id = 1;



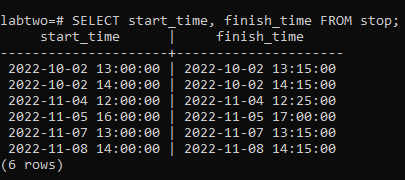
e. З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах

Початкова таблиця stop:



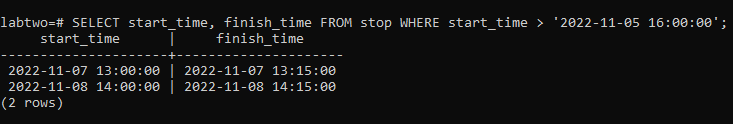
Виведемо стовпці start\_time, finish\_time з таблиці stop:

SELECT start\_time, finish\_time FROM stop;



Виведемо рядки з стовпців start\_time, finish\_time з таблиці stop, де start\_time > '2022-11-05 16:00:00':

SELECT start\_time, finish\_time FROM stop WHERE start\_time > '2022-11-05 16:00:00';

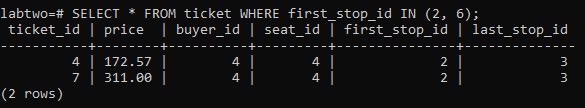


f. Використання операторів:

i. Приналежності множині

Виведемо рядки з таблиці ticket, де first\_stop\_id належить множині (2, 6):

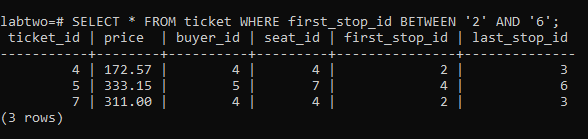
SELECT \* FROM ticket WHERE first\_stop\_id IN (2, 6);



ii. Приналежності діапазону

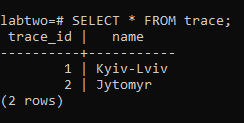
Виведемо рядки з таблиці ticket, де first\_stop\_id належить діапазону [2, 6]:

SELECT \* FROM ticket WHERE first\_stop\_id BETWEEN '2' AND '6';



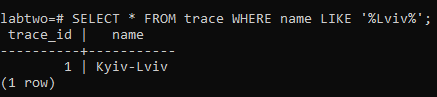
iii. Відповідності шаблону

Початкова таблиця trace:



Виведемо рядки з таблиці trace, де name містить Lviv:

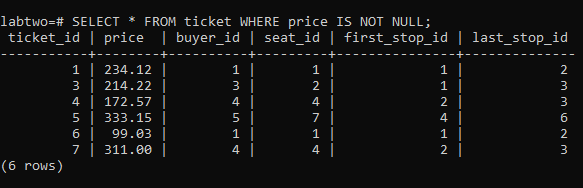
SELECT \* FROM trace WHERE name LIKE '%Lviv%';



iv. Перевірка на невизначене значення

Виведемо рядки з таблиці ticket, де price заповнене:

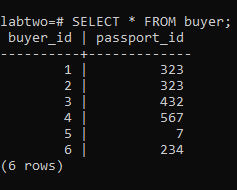
SELECT \* FROM ticket WHERE price IS NOT NULL;



2) Створити запити з використанням підзапитів та з’єднань (разом 11 запитів) (в запитах повинні використовуватись 2 та більше таблиць):

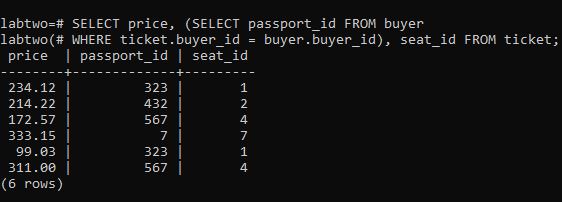
a. Використання підзапитів в рядку вибірки полів (у секції select) та вибірки з таблиць (у секції from)

Початкова таблиця buyer:



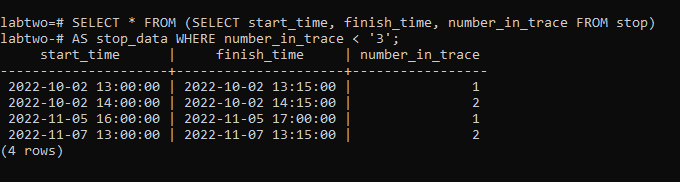
Виведемо стовпці price, seat\_id з таблиці ticket та відповідні значення passport\_id з таблиці buyer до кожного рядка (мають співпадати ticket.buyer\_id та buyer.buyer\_id в початкових таблицях):

SELECT price, (SELECT passport\_id FROM buyer WHERE ticket.buyer\_id = buyer.buyer\_id), seat\_id FROM ticket;



Виведемо рядки з таблиці, яка складається з стовпців start\_time, finish\_time, number\_in\_trace з таблиці stop, де number\_in\_trace < 3:

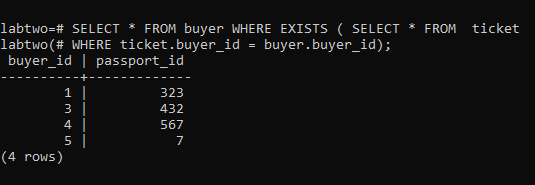
SELECT \* FROM (SELECT start\_time, finish\_time, number\_in\_trace FROM stop) AS stop\_data WHERE number\_in\_trace < '3';



b. Використання підзапитів в умовах з конструкціями EXISTS, IN

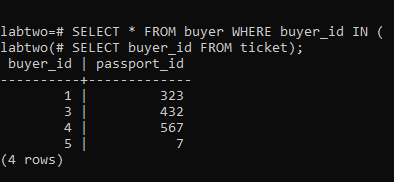
Виведемо рядки з таблиці buyer, де buyer\_id є серед рядків у таблиці ticket:

SELECT \* FROM buyer WHERE EXISTS ( SELECT \* FROM ticket WHERE ticket.buyer\_id = buyer.buyer\_id);



Виведемо рядки з таблиці buyer, де buyer\_id є серед множини buyer\_id із таблиці ticket:

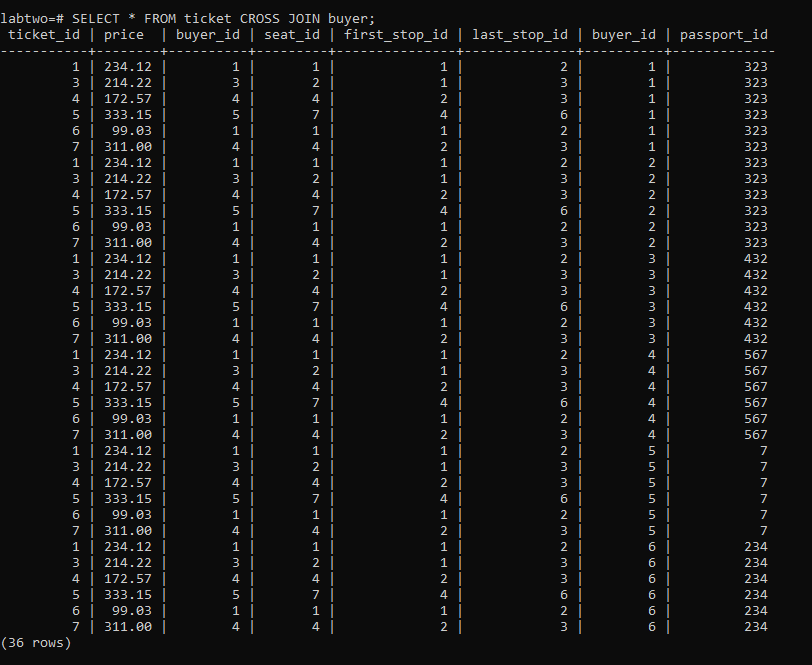
SELECT \* FROM buyer WHERE buyer\_id IN (SELECT buyer\_id FROM ticket);



c. Декартовий добуток

Виведемо декартів добуток рядків з таблиць ticket та buyer:

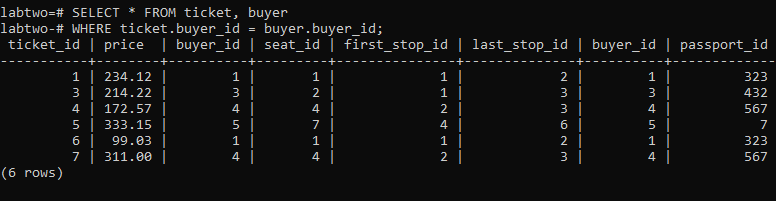
SELECT \* FROM ticket CROSS JOIN buyer;



d. З’єднання декількох таблиць за рівністю

Виведемо рядки зі з’єднаних таблиць ticket, buyer за рівністю buyer\_id у кожній з них:

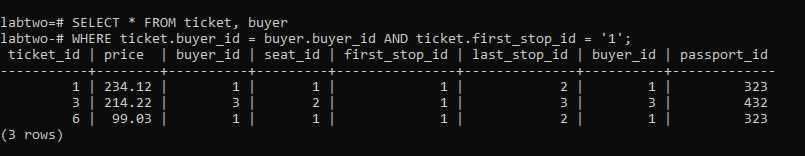
SELECT \* FROM ticket, buyer WHERE ticket.buyer\_id = buyer.buyer\_id;



e. З’єднання декількох таблиць за рівністю та умовою відбору

Виведемо рядки зі з’єднаних таблиць ticket, buyer за рівністю buyer\_id у кожній з них, де ticket.first\_stop\_id де 1:

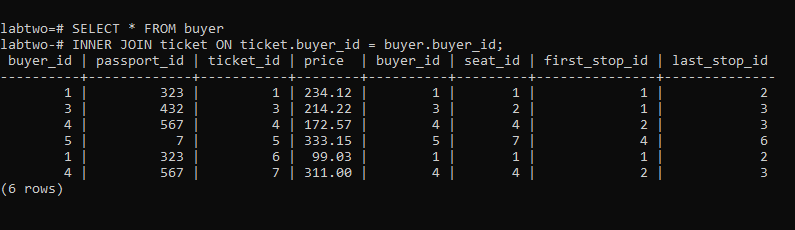
SELECT \* FROM ticket, buyer WHERE ticket.buyer\_id = buyer.buyer\_id AND ticket.first\_stop\_id = '1';



f. Внутрішнього з’єднання

Виведемо рядки зі внутрішньо з’єднаних таблиць buyer та ticket за рівністю buyer\_id у кожній з них:

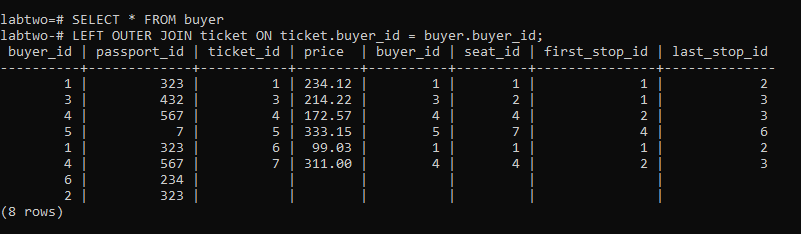
SELECT \* FROM buyer INNER JOIN ticket ON ticket.buyer\_id = buyer.buyer\_id;



g. Лівого зовнішнього з’єднання

Виведемо рядки зі зовнішньо ліво з’єднаних таблиць buyer та ticket за рівністю buyer\_id у кожній з них:

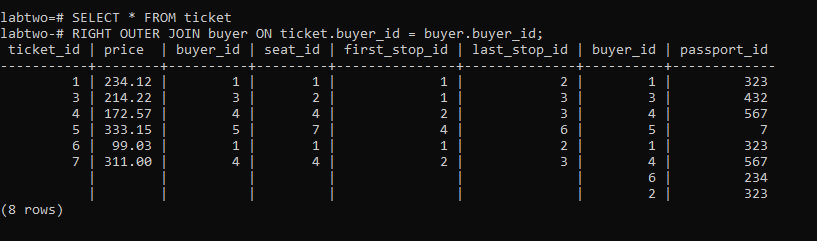
SELECT \* FROM buyer LEFT OUTER JOIN ticket ON ticket.buyer\_id = buyer.buyer\_id;



h. Правого зовнішнього з’єднання

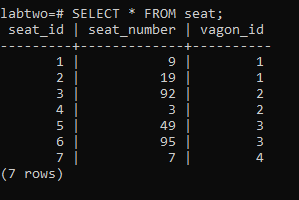
Виведемо рядки зі зовнішньо право з’єднаних таблиць ticket та buyer за рівністю buyer\_id у кожній з них:

SELECT \* FROM ticket RIGHT OUTER JOIN buyer ON ticket.buyer\_id = buyer.buyer\_id;



i. Об’єднання запитів

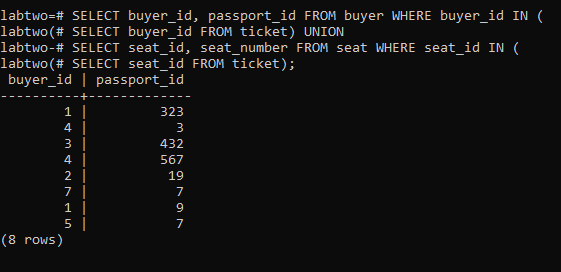
Початкова таблиця seat:



Виведемо рядки стовпців buyer\_id, passport\_id з таблиці buyer, де buyer\_id є серед записів у стовпці buyer\_id в таблиці ticket, та допишемо у ці стовпці дані з стовпців seat\_id, seat\_number рядків таблиці seat, seat\_id яких є серед записів стовбця seat\_id таблиці ticket:

SELECT buyer\_id, passport\_id FROM buyer WHERE buyer\_id IN (SELECT buyer\_id FROM ticket) UNION

SELECT seat\_id, seat\_number FROM seat WHERE seat\_id IN (SELECT seat\_id FROM ticket);



Висновок: під час виконання лабораторної роботи я навчився створювати запити для вибірки даних з таблиць, об’єднувати таблиці.