

Baza danych

System Finansowy

System transakcyjny składa się z 4 tabel powiązanych ze sobą relacjami na schemacie powyżej. Tabela „Customers” zawiera prywatne dane klientów, tabela „Accounts” dane dla kont klientów, tabela „Money_transfer” gromadzi transakcje wykonane przez klientów, tabela „Types” to dostępne typy transakcji w systemie. Tabela „Accounts” zawiera pole „ACC_STATUS”, przechowujące informację o statusie danego konta. Tłumaczenie: *1 – OPENED, 2 – CLOSED, 3 – BLOCKED*. Pole „ACC_TYPE” przechowuje informację o typie danego konta. Tłumaczenie: *1 – BUY, 2 – SELL*. Aby klient mógł zrealizować transakcję ze swojego konta, wymagane jest, aby konto było otwarte w momencie wykonania transakcji, czyli w dniu TRN_DATE.

1. Stworzyć bazę danych. Utworzyć wszystkie tabele i relacje zgodnie ze schematem. Załadować dane do 3 tabel.
2. Napisać procedurę „insert_transaction”, w taki sposób, aby poniższe wywołania były możliwe. Załadować dane do tabeli „Money_transfer”.

```
CALL insert_transaction(1000, STR_TO_DATE('05-05-2020', '%d-%m-%Y'), 10, 100, 'EUR',  
'2021-05-06 11:00:00', 14);  
  
CALL insert_transaction(1200, STR_TO_DATE('29-10-2021', '%d-%m-%Y'), 10, 120000,  
'EUR', '2021-10-30 09:00:00', 12);  
  
CALL insert_transaction(1400, STR_TO_DATE('30-01-2022', '%d-%m-%Y'), 10, 1000, 'EUR',  
'2022-01-31 15:00:00', 18);  
  
CALL insert_transaction(1600, STR_TO_DATE('26-02-2022', '%d-%m-%Y'), 15, 500, 'EUR',  
'2022-02-27 10:00:00', 15);
```

Nazwa procedury

Typ

	Kierunek	Nazwa	Typ	Długość/Wartości	Opcje	
↑	IN	id	II	11		Usuń
↑	IN	date1	D	---	---	Usuń
↑	IN	type_id	II	11		Usuń
↑	IN	value1	II	11		Usuń
↑	IN	currenc	V	30	Ko	Usuń
↑	IN	process	T		---	Usuń
↑	IN	acc_id	II	11		Usuń

Parametry

Dodaj parametr

```

1 BEGIN
2 INSERT INTO money_transfer VALUES (id, date1, type_id,
3 value1, currency, processing_data, acc_id);
4 END

```

	TRN_ID	TRN_DATE	TRN_TYPE_ID	TRN_VALUE	TRN_CURRENCY	TRN_PROCESSING_DATE	TRN_ACC_ID
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	1000	2020-05-05	10	100	EUR	2021-05-06 11:00:00	14
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	1100	2021-10-01	15	3000	PLN	2021-10-02 12:00:00	11
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	1200	2021-10-29	10	120000	EUR	2021-10-30 09:00:00	12
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	1400	2022-01-30	10	1000	EUR	2022-01-31 15:00:00	18
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	1600	2022-02-26	15	500	EUR	2022-02-27 10:00:00	15
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	1800	2022-03-18	10	250	EUR	2022-03-19 11:00:00	18
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	2000	2022-04-29	10	300	EUR	2022-04-30 12:00:00	14
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	2500	2022-05-01	15	500	USD	2022-05-02 06:00:00	19
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	2900	2022-05-05	10	10000	EUR	2022-05-06 15:00:00	14
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	3000	2022-05-06	10	2000	PLN	2022-05-07 07:00:00	10
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	4000	2023-04-03	15	1500	PLN	2022-04-04 13:00:00	11
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	4600	2023-04-10	10	1000	USD	2023-04-11 15:00:00	18
<input type="checkbox"/> Edytuj <input type="checkbox"/> Kopiuj <input type="checkbox"/> Usuń	5000	2023-04-16	15	2000	EUR	2023-04-17 12:00:00	15

3. Napisać zapytanie tworzące prosty raport złożony z 2 kolumn dla wszystkich tabel w bazie. Raport powinien zawierać liczbę wierszy dla każdej z tabel oraz największe id.

✓ Pokazano wiersze 0 - 3 (4 ogółem, Wykonanie zapytania trwało 0,0045 sekund(y).)

```
SELECT CONCAT("Liczba wierszy ", COUNT(*)) AS Liczba_Wierszy, CONCAT("Max id ", MAX(money_transfer.TRN_ID)) AS Max_id FROM money_transfer UNION SELECT CONCAT("Liczba wierszy ", COUNT(*)) AS Liczba_Wierszy, CONCAT("Max id ", MAX(customers.CUST_ID)) AS Max_id FROM customers UNION SELECT CONCAT("Liczba wierszy ", COUNT(*)) AS Liczba_Wierszy, CONCAT("Max id ", MAX(accounts.ACC_ID)) AS Max_id FROM accounts UNION SELECT CONCAT("Liczba wierszy ", COUNT(*)) AS Liczba_Wierszy, CONCAT("Max id ", MAX(types.TYPE_ID)) AS Max_id FROM types;
```

☐ Profilowanie [Edytuj w linii] [Edytuj] [Wyjaśnij SQL] [Utwórz kod PHP] [Odśwież]

☐ Pokaż wszystko | Liczba wierszy: 25 | Filtrowanie wierszy: Przeszukaj tę tabelę | Sortuj wg klucza: Żaden

+ Opcje

Liczba_Wierszy	Max_id
Liczba wierszy 13	Max id 5000
Liczba wierszy 5	Max id 5
Liczba wierszy 10	Max id 19
Liczba wierszy 2	Max id 15

4. Napisać zapytanie wyświetlające numery, statusy i typy kont oraz średnią wartość transakcji dla tych kont. Wykorzystać polecenia CASE WHEN oraz IF tak, aby statusy i typy kont nie były liczbami.

✓ Pokazano wiersze 0 - 12 (13 ogółem, Wykonanie zapytania trwało 0,0029 sekund(y).)

```
SELECT accounts.ACC_NUMBER, customers.CUST_STATUS, types.TYPE_NAME FROM accounts INNER JOIN customers ON accounts.ACC_CUST_ID=customers.CUST_ID INNER JOIN money_transfer ON accounts.ACC_ID=money_transfer.TRN_ACC_ID INNER JOIN types ON money_transfer.TRN_TYPE_ID=types.TYPE_ID;
```

☐ Profilowanie [Edytuj w linii] [Edytuj] [Wyjaśnij SQL] [Utwórz kod PHP] [Odśwież]

☐ Pokaż wszystko | Liczba wierszy: 25 | Filtrowanie wierszy: Przeszukaj tę tabelę | Sortuj wg klucza: Żaden

+ Opcje

ACC_NUMBER	CUST_STATUS	TYPE_NAME
121212	Active	BUY
232323	Blocked	BUY
343434	Active	BUY
343434	Active	BUY
343434	Active	BUY
565656	Active	BUY
565656	Active	BUY
565656	Active	BUY
212121	Active	SELL
212121	Active	SELL
434343	Active	SELL
434343	Active	SELL
656565	Active	SELL

5. Napisać 2 zapytania wyświetlające imię, nazwisko klienta, liczbę wykonanych transakcji (min. 2 wykonane transakcje) oraz sumę ich wartości, tylko dla typu „BUY”. Wymagania:
- 1 query – powiązanie tabeli „Types” instrukcją JOIN,
 - 2 query - powiązanie tabeli „Types” zapytaniem zagnieżdżonym,
- W sprawozdaniu załączyć czasy wykonania obu zapytań.

✓ Pokazano wiersze 0 - 1 (2 total, Wykonanie zapytania trwało 0,0006 sekund(y).)

```
SELECT customers.CUST_FIRST_NAME AS "Imie", customers.CUST_LAST_NAME AS "Nazwisko",  
COUNT(money_transfer.TRN_ID) AS "Liczba transakcji", SUM(money_transfer.TRN_VALUE) AS "Suma" FROM  
customers INNER JOIN accounts ON customers.CUST_ID=accounts.ACC_CUST_ID INNER JOIN money_transfer ON  
accounts.ACC_ID=money_transfer.TRN_ACC_ID INNER JOIN types ON money_transfer.TRN_TYPE_ID=types.TYPE_ID  
WHERE types.TYPE_NAME="BUY" GROUP BY customers.CUST_FIRST_NAME HAVING COUNT(money_transfer.TRN_ID)>1;
```

☐ Profilowanie [Edytuj w linii] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Liczba wierszy: 25 ▼ Filter rows: Przeszukaj tę tabelę

Extra options

Imie	Nazwisko	Liczba transakcji	Suma
Bridget	Von Hammersmark	3	2250
Shosanna	Dreyfus	3	10400

✓ Pokazano wiersze 0 - 1 (2 total, Wykonanie zapytania trwało 0,0009 sekund(y).)

```
SELECT customers.CUST_FIRST_NAME AS "Imie", customers.CUST_LAST_NAME AS "Nazwisko",  
COUNT(money_transfer.TRN_ID) AS "Liczba transakcji", SUM(money_transfer.TRN_VALUE) AS "Sum" FROM customers  
INNER JOIN accounts ON customers.CUST_ID=accounts.ACC_CUST_ID INNER JOIN money_transfer ON  
accounts.ACC_ID=money_transfer.TRN_ACC_ID WHERE money_transfer.TRN_TYPE_ID IN (SELECT types.TYPE_ID FROM  
types WHERE TYPE_NAME="BUY") GROUP BY customers.CUST_FIRST_NAME HAVING COUNT(money_transfer.TRN_ID)>1;
```

☐ Profilowanie [Edytuj w linii] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Liczba wierszy: 25 ▼ Filter rows: Przeszukaj tę tabelę

Extra options

Imie	Nazwisko	Liczba transakcji	Sum
Bridget	Von Hammersmark	3	2250
Shosanna	Dreyfus	3	10400

Zapytanie wykorzystujące JOIN było szybsze, lecz różnica jest niewielka.

6. Utworzyć raport przy pomocy SQL, w którym będzie wyświetlana liczba wszystkich otwartych kont oraz ostatnia data transakcji dla otwartego konta, gdzie kod typu transakcji to „222”. Wymagane instrukcje: WITH, CONCAT, UNION, filtrowanie pola JSON.

```
Uruchom zapytanie SQL/zapytania w tabeli finansowa_1.money_transfer: ⓘ

1 WITH otwarte_konta AS (
2     SELECT *
3     FROM accounts
4     WHERE ACC_STATUS=1
5 ),
6 ostatnie_transakcje AS (
7     SELECT
8         money_transfer.TRN_ACC_ID AS id_konta,
9         MAX(money_transfer.TRN_DATE) AS ostatnia_transakcja,
10        money_transfer.TRN_TYPE_ID AS typ
11     FROM money_transfer
12     GROUP BY money_transfer.TRN_ACC_ID
13 )
14 SELECT
15     CONCAT("Liczba otwartych kont: ",COUNT(ACC_ID)) AS ""
16     FROM otwarte_konta
17     UNION
18     SELECT CONCAT("Ostatnia transakcja konta nr: ", ostatnie_transakcje.id_konta, " - ",
19        ostatnie_transakcje.ostatnia_transakcja)
20     FROM ostatnie_transakcje
21     INNER JOIN otwarte_konta ON ostatnie_transakcje.id_konta=otwarte_konta.ACC_ID
22     INNER JOIN types ON ostatnie_transakcje.typ=types.TYPE_ID
23     WHERE JSON_EXTRACT(types.TYPE_DETAILS,"$.code") = '222'
```

Liczba otwartych kont: 5

Ostatnia transakcja konta nr: 15 - 2023-04-16

Ostatnia transakcja konta nr: 11 - 2023-04-03

Baza danych

Sprzedaż Wysyłkowa

1. Utworzyć widok dla złożonych zamówień: numer zamówienia, imię, nazwisko, data zamówienia, zamówiony towar, liczba sztuk, cena sprzedaży, wartość zamówionego towaru (nie uwzględniać kosztów wysyłki)

✓ Pokazano wiersze 0 - 11 (12 ogółem, Wykonanie zapytania trwało 0,0039 sekund(y).)

```
SELECT orderline.orderinfo_id, customer.fname, customer.lname,
orderinfo.date_placed, item.description, orderline.quantity, item.sell_price,
item.sell_price*orderline.quantity FROM orderline INNER JOIN orderinfo ON
orderline.orderinfo_id=orderinfo.orderinfo_id INNER JOIN customer ON
orderinfo.customer_id=customer.customer_id INNER JOIN item ON
orderline.item_id=item.item_id;
```

☐ Profilowanie [Edytuj w linii] [Edytuj] [Wyjaśnij SQL] [Utwórz kod PHP] [Odśwież]

☐ Pokaż wszystko | Liczba wierszy: 25 ▼ Filtrowanie wierszy: Przeszukaj tę tabelę

+ Opcje

orderinfo_id	fname	lname	date_placed	description	quantity	sell_price	item.sell_price*ord
1	Alex	Matthew	2000-03-13	Tissues	1	3.99	3.99
1	Alex	Matthew	2000-03-13	Fan Large	1	19.95	19.95
1	Alex	Matthew	2000-03-13	Roman Coin	1	2.45	2.45
2	Ann	Stones	2000-06-23	Wood Puzzle	1	21.95	21.95
2	Ann	Stones	2000-06-23	Tissues	2	3.99	7.98
2	Ann	Stones	2000-06-23	Fan Large	2	19.95	39.90
2	Ann	Stones	2000-06-23	Carrier Bag	1	0.00	0.00
3	David	Hudson	2000-09-02	Wood Puzzle	1	21.95	21.95
3	David	Hudson	2000-09-02	Rubic Cube	1	11.49	11.49
4	Laura	Hendy	2000-09-03	Picture Frame	2	9.95	19.90
5	Ann	Stones	2000-07-21	Wood Puzzle	1	21.95	21.95
5	Ann	Stones	2000-07-21	Linux CD	1	2.49	2.49

2. Utworzyć procedurę składowaną, która po uruchomieniu wyświetla następujące informacje: numer zamówienia, imię, nazwisko, data zamówienia, zamówiony towar, liczba sztuk, cena sprzedaży, wartość zamówionego towaru (nie uwzględniać kosztów wysyłki) ale tylko dla wybranego klienta, którego dane są przekazywane jako parametry procedury.

Szczegóły

Nazwa procedury

procedura2

Typ

PROCEDURE

Parametry

	Kierunek	Nazwa	Typ	Długość/Wartości	Opcje
↑	IN	fname	V	50	Ko Usun
↑	IN	lname	V	50	Ko Usun

Dodaj parametr

Określenie

```

1 BEGIN
2 SELECT orderline.orderinfo_id, customer.fname,
   customer.lname, orderinfo.date_placed, item.description,
   orderline.quantity, item.sell_price,
   item.sell_price*orderline.quantity FROM orderline INNER
   JOIN orderinfo ON
   orderline.orderinfo_id=orderinfo.orderinfo_id INNER JOIN
   customer ON orderinfo.customer_id=customer.customer_id
   INNER JOIN item ON orderline.item_id=item.item_id WHERE
   customer.fname=fname AND customer.lname=lname;
3 END

```

```
SET @p0='Ann'; SET @p1='Stones'; CALL `procedura2`(@p0, @p1);
```

Wynik wykonanych procedur `procedura2`

orderinfo_id	fname	lname	date_placed	description	quantity	sell_price	item.sell_price*orderline.que
2	Ann	Stones	2000-06-23	Wood Puzzle	1	21.95	21.95
2	Ann	Stones	2000-06-23	Tissues	2	3.99	7.98
2	Ann	Stones	2000-06-23	Fan Large	2	19.95	39.90
2	Ann	Stones	2000-06-23	Carrier Bag	1	0.00	0.00
5	Ann	Stones	2000-07-21	Wood Puzzle	1	21.95	21.95
5	Ann	Stones	2000-07-21	Linux CD	1	2.49	2.49

3. Utworzyć procedurę składowaną tworzącą tabelę z informacjami o towarach, których w tabeli stock jest mniej niż wartość zadanej liczby (będącej parametrem procedury). Działanie procedury zademonstrować dla towarów, których jest mniej niż 4 sztuki.

Nazwa procedury

Typ

Parametry

Kierunek	Nazwa	Typ	Długość/Wartości	Opcje
IN	q	II		<input type="button" value="Usuń"/>

```
1 BEGIN
2 CREATE TABLE quantityTable (description varchar(255),
  quantity int) SELECT item.description AS description,
  stock.quantity AS quantity FROM stock INNER JOIN item ON
  stock.item_id=item.item_id WHERE stock.quantity < q;
3 END
```

Wykonaj procedurę `procedura4`

Parametry procedury

Nazwa	Typ	Funkcja	Wartość
q	INT		4

✓ Pokazano wiersze 0 - 2 (3 ogółem, Wykonanie zapytania trwało 0,0004 sekund(y).)

[SELECT * FROM `quantitytable`](#)

☐ Profilowanie [\[Edytuj w linii \]](#) [\[Edytuj \]](#) [\[Wyjaśnij SQL \]](#) [\[Utwórz kod PHP \]](#) [\[Odśwież \]](#)

☐ Pokaż wszystko | Liczba wierszy: | Filtrowanie wierszy:

+ Opcje

description	quantity
Rubic Cube	2
Picture Frame	3
Carrier Bag	1

4. Utworzyć trigger (typu AFTER) dla INSERT dla tabeli item. Trigger powinien wypełniać rekordy w tabeli item_history. Standardowo, po wstawieniu nowego towaru przy użyciu INSERT powinna pojawiać się nazwa tego towaru oraz OK. Ale, gdy ceny zawierają NULL, zamiast OK, powinna pojawić się o tym informacja.

Details

Nazwa wyzwalacza

item_insert_trigger

Table

item

Czas

AFTER

Event

INSERT

Definition

```

1 BEGIN
2   DECLARE price_status VARCHAR(20);
3   IF (new.cost_price IS NULL) OR (new.sell_price IS
NULL) THEN
4     SET price_status = 'NULL price';
5   ELSE
6     SET price_status = 'OK';
7   END IF;
8
9   INSERT INTO item_history (item_name, price_status)
10  VALUES (new.description, price_status);
11 END

```

<input type="checkbox"/>	Edit	Copy	Delete	16 TV	10.00	15.00
<input type="checkbox"/>	Edit	Copy	Delete	17 Computer	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	18 Domofon	1.00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	19 Domek	10.00	NULL
<input type="checkbox"/>	Edit	Copy	Delete	20 Phone	1.00	1.00

			history_id	item_name	price_status
<input type="checkbox"/>	Edit	Copy	Delete	1 TV	OK
<input type="checkbox"/>	Edit	Copy	Delete	2 Computer	NULL price
<input type="checkbox"/>	Edit	Copy	Delete	3 Domofon	NULL price
<input type="checkbox"/>	Edit	Copy	Delete	4 Domek	NULL price
<input type="checkbox"/>	Edit	Copy	Delete	5 Phone	OK

5. Utworzyć procedurę przechowywaną wypisującą przekazywany tekst trzy razy (w jednej linii): użyć konstrukcji LOOP, działanie procedury zademonstrować dla nazwy ulubionego miasta.

Details

Nazwa procedury

procedura1

Typ

PROCEDURE ▾

Parametry

Direction	Nazwa	Typ	Długość/Wartości
IN ▾	text_to_print	VARCHAR ▾	255

Add parameter

Definition

```
1 BEGIN
2   DECLARE i INT DEFAULT 1;
3   DECLARE text_with_newline VARCHAR(255);
4
5   SET text_with_newline = CONCAT(' ',text_to_print,' ');
6
7   WHILE i < 3 DO
8     SET i = i + 1;
9     SET text_to_print = CONCAT(text_to_print, text_with_newline);
10  END WHILE;
11
12  SELECT text_to_print;
13 END
```

Wykonanie zapytania/zapytań SQL do bazy danych jo: ?

```
1 CALL procedura1('Kraków');
```

✓ Pokazano wiersze 0 - 0 (1 total, Wykonanie zapytania trwało 0,0004 sekund(y).)

```
CALL procedura1('Kraków');
```

[Edytuj w linii] [Edit] [Create PHP code]

☐ Show all | Liczba wierszy: 25 ▾ | Filter rows: Przeszukaj tę tabelę

Extra options

text_to_print

Kraków Kraków Kraków

6. Załóżmy, że dane są dwie tabele dwukolumnowe. Pierwsze kolumny są identyczne i stanowią dla obu tabel klucz podstawowy. Drugie kolumny zawierają dowolne liczby całkowite. Przyjmijmy, że mamy po pięć rekordów. Należy napisać procedurę przechowywaną z kursorem tworzącą trzecią dwukolumnową tabelę. W trzeciej tabeli pierwsza kolumna powinna być taka sama, jak w tabelach wyjściowych. Do drugiej kolumny wstawiamy tekst zależny od tego, która liczba większa w drugich kolumnach tabel wyjściowych: $t1 > t2$, albo $t1 \leq t2$.

Details

Nazwa procedury

procedura_3_tabela

Typ

PROCEDURE ▼

Parametry

Direction	Nazwa	Typ	Długość/Wartości
-----------	-------	-----	------------------

Add parameter

Definition

```
1 BEGIN
2   DECLARE done INT DEFAULT FALSE;
3   DECLARE t1_value INT;
4   DECLARE t2_value INT;
5   DECLARE text_value VARCHAR(11);
6   DECLARE t1_cursor CURSOR FOR SELECT id, wartosc FROM table1;
7   DECLARE t2_cursor CURSOR FOR SELECT id, wartosc FROM table2;
8   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
9
10
11  CREATE TABLE IF NOT EXISTS table3 (
12    id INT PRIMARY KEY,
13    wartosc1 VARCHAR(11)
14  );
15
16  OPEN t1_cursor;
17  OPEN t2_cursor;
18
19  read_loop: LOOP
20    FETCH t1_cursor INTO t1_value, text_value;
21    FETCH t2_cursor INTO t2_value, text_value;
22    IF done THEN
23      LEAVE read_loop;
24    END IF;
25    IF t1_value > t2_value THEN
26      SET text_value = 't1>t2';
27    ELSE
28      SET text_value = 't1<=t2';
29    END IF;
30
31    INSERT INTO table3 (id, wartosc1) VALUES (t1_value, text_value);
32  END LOOP;
33
34  CLOSE t1_cursor;
35  CLOSE t2_cursor;
36 END
```

Definition

✓ Pokazano wiersze 0 - 4 (5 total, Wykonanie zapytania trwało 0,0007 sekund(y).)

`SELECT * FROM `table1``

☐ Profilowanie [Edytuj w linii] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Liczba wierszy: 25 | Filter rows: Przeszukaj tę tabelę

Extra options

	id	wartosc
<input type="checkbox"/> Edit Copy Delete	1	10
<input type="checkbox"/> Edit Copy Delete	2	20
<input type="checkbox"/> Edit Copy Delete	3	30
<input type="checkbox"/> Edit Copy Delete	4	40
<input type="checkbox"/> Edit Copy Delete	5	50

✓ Pokazano wiersze 0 - 4 (5 total, Wykonanie zapytania trwało 0,0007 sekund(y).)

`SELECT * FROM `table2``

☐ Profilowanie [Edytuj w linii] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Liczba wierszy: 25 | Filter rows: Przeszukaj tę tabelę

Extra options

	id	wartosc
<input type="checkbox"/> Edit Copy Delete	1	20
<input type="checkbox"/> Edit Copy Delete	2	10
<input type="checkbox"/> Edit Copy Delete	3	40
<input type="checkbox"/> Edit Copy Delete	4	30
<input type="checkbox"/> Edit Copy Delete	5	50

✓ Pokazano wiersze 0 - 4 (5 total, Wykonanie zapytania trwało 0,0003 sekund(y).)

`SELECT * FROM `table3``

☐ Profilowanie [Edytuj w linii] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Liczba wierszy: 25 | Filter rows: Przeszukaj tę tabelę

Extra options

	id	wartosc1
<input type="checkbox"/> Edit Copy Delete	1	t1<=t2
<input type="checkbox"/> Edit Copy Delete	2	t1<=t2
<input type="checkbox"/> Edit Copy Delete	3	t1<=t2
<input type="checkbox"/> Edit Copy Delete	4	t1<=t2
<input type="checkbox"/> Edit Copy Delete	5	t1<=t2