SQL test Name and Surname: Klaudia Kiedrowska

APPS					
app_id	cust_id	applied_date	issued_date	amount	period
1	101	2018.06.10	2018.06.12	5 000,00	6
2	102	2018.06.15			
3	103	2018.06.25	2018.06.25	7 500,00	12
4	104	2018.07.01	2018.07.02	5 000,00	12
5	105	2018.07.05	2018.07.05	2 500,00	3
6	106	2018.07.15			
7	101	2018.07.20	2018.07.20	5 000,00	6

CUSTOMERS			
cust_id	name	surname	income
101	Jan	Kowalski	5 000,00
102	Adam	Nowak	2 000,00
103	Anna	Wiśniewska	2 500,00
104	Przemek	Kamiński	5 000,00
105	Joanna	Zielińska	4 500,00
106	Michał	Śliwa	3 000,00

INSTALLMENTS					
app_id	ins_num	due_date	capital	fee	total
1	1	2018.07.13	833,33	166,67	1 000,00
1	2	2018.08.12	833,33	166,67	1 000,00
1	3	2018.09.12	833,33	166,67	1 000,00
1	4	2018.10.12	833,33	166,67	1 000,00
1	5	2018.11.12	833,33	166,67	1 000,00
1	6	2018.12.13	833,33	166,67	1 000,00
3	1	2018.07.26	625,00	93,75	718,75
3	2	2018.08.25	625,00	93,75	718,75
3	3	2018.09.25	625,00	93,75	718,75
3	4	2018.10.25	625,00	93,75	718,75
3	5	2018.11.25	625,00	93,75	718,75
3	6	2018.12.26	625,00	93,75	718,75
3	7	2019.01.23	625,00	93,75	718,75
3	8	2019.02.23	625,00	93,75	718,75
3	9	2019.03.25	625,00	93,75	718,75
3	10	2019.04.25	625,00	93,75	718,75
3	11	2019.05.25	625,00	93,75	718,75
3	12	2019.06.25	625,00	93,75	718,75
4	1	2018.08.01	416,67	104,17	520,83
4	2	2018.09.01	416,67	104,17	520,83
4	3	2018.10.01	416,67	104,17	520,83
4	4	2018.11.01	416,67	104,17	520,83
4	5	2018.12.02	416,67	104,17	520,83
4	6	2018.12.30	416,67	104,17	520,83
4	7	2019.01.30	416,67	104,17	520,83
4	8	2019.03.01	416,67	104,17	520,83
4	9	2019.04.01	416,67	104,17	520,83
4	10	2019.05.01	416,67	104,17	520,83
4	11	2019.06.01	416,67	104,17	520,83
4	12	2019.07.02	416,67	104,17	520,83
5	1	2018.08.19	833,33	166,67	1 000,00
5	2	2018.09.19	833,33	166,67	1 000,00
5	3	2018.10.19	833,33	166,67	1 000,00
7	1	2018.08.19	833,33	250,00	1 083,33
7	2	2018.09.19	833,33	250,00	1 083,33
7	3	2018.10.19	833,33	250,00	1 083,33
7	4	2018.11.19	833,33	250,00	1 083,33
7	5	2018.12.20	833,33	250,00	1 083,33
7	6	2019.01.17	833,33	250,00	1 083,33

PAYMENTS		
app_id	paid_date	paid_amount
1	2018.07.12	1 000,00
1	2018.08.17	1 000,00
1	2018.08.18	500,00
4	2018.07.27	520,83
4	2018.09.01	520,83
7	2018.08.21	1 200,00
7	2018.09.21	1 200,00
7	2018.09.24	2 000,00

Generating tables:

```
CREATE TABLE apps (
    app_id INT PRIMARY KEY,
    cust_id INT,
    applied_date DATE,
    issued_date DATE,
    amount INT,
    period INT
);
```

```
INSERT INTO apps (app_id, cust_id, applied_date, issued_date, amount,
period)
VALUES
    (1, 101, '2018-06-10', '2018-06-12', 5000, 6),
    (2, 102, '2018-06-15', null, null, null),
    (3, 103, '2018-06-25', '2018-06-25', 7500, 12),
    (4, 104, '2018-07-01', '2018-07-02', 5000, 12),
    (5, 105, '2018-07-05', '2018-07-05', 2500, 3),
    (6, 106, '2018-07-15', null, null, null),
    (7, 101, '2018-07-20', '2018-07-20', 5000, 6);

--  'customers' table
CREATE TABLE customers (
    cust_id INT PRIMARY KEY,
    name VARCHAR(50),
    surname VARCHAR(50),
    income INT
);

-- Filling in  'customers' table
INSERT INTO customers (cust_id, name, surname, income)
VALUES
    (101, 'Jan', 'Kowalski', 5000),
    (102, 'Adam', 'Nowak', 2000),
    (103, 'Anna', 'Wisniewska', 2500),
    (104, 'Przemek', 'Kaminski', 5000),
    (105, 'Joanna', 'Zielinska', 4500),
    (106, 'Michał', 'Sliwa', 3000);

--  'payments' table
CREATE TABLE payments (
    app_id INT,
    paid_date DATE,
    paid_amount DECIMAL(8,2),
    FOREIGN KEY (app_id) REFERENCES apps(app_id)
);

-- Filling in 'payments' table
INSERT INTO payments (app_id, paid_date, paid_amount)
VALUES
    (1, '2018-07-12', 1000),
    (1, '2018-08-17', 1000),
    (1, '2018-08-18', 500),
    (4, '2018-07-27', 520),
    (4, '2018-09-01', 520),
    (7, '2018-08-21', 1200),
    (7, '2018-09-21', 1200),
```

```
    (7, '2018-09-24', 2000);

-- 'installments' table
CREATE TABLE installments (
    app_id INT,
    ins_num INT,
    due_date DATE,
    capital DECIMAL(8,2),
    fee DECIMAL(8,2),
    total DECIMAL(8,2),
    FOREIGN KEY (app_id) REFERENCES apps(app_id)
);

-- Filling in 'installments' table
INSERT INTO installments (app_id, ins_num, due_date, capital, fee, total)
VALUES
(1, 1, '2018-07-13', 833.33, 166.67, 1000.00),
(1, 2, '2018-08-12', 833.33, 166.67, 1000.00),
 (1, 3, '2018-09-12', 833.33, 166.67, 1000.00),
 (1, 4, '2018-10-12', 833.33, 166.67, 1000.00),
 (1, 5, '2018-11-12', 833.33, 166.67, 1000.00),
 (1, 6, '2018-12-13', 833.33, 166.67, 1000.00),
 (3, 1, '2018-07-26', 625.00, 93.75, 718.75),
 (3, 2, '2018-08-25', 625.00, 93.75, 718.75),
 (3, 3, '2018-09-24', 625.00, 93.75, 718.75),
 (3, 4, '2018-10-24', 625.00, 93.75, 718.75),
 (3, 5, '2018-11-23', 625.00, 93.75, 718.75),
 (3, 6, '2018-12-23', 625.00, 93.75, 718.75),
 (3, 7, '2019-01-22', 625.00, 93.75, 718.75),
 (3, 8, '2019-02-21', 625.00, 93.75, 718.75),
 (3, 9, '2019-03-23', 625.00, 93.75, 718.75),
 (3, 10, '2019-04-22', 625.00, 93.75, 718.75),
 (3, 11, '2019-05-22', 625.00, 93.75, 718.75),
 (3, 12, '2019-06-21', 625.00, 93.75, 718.75),
 (4, 1, '2018-08-01', 416.67, 104.17, 520.83),
 (4, 2, '2018-09-01', 416.67, 104.17, 520.83),
 (4, 3, '2018-10-02', 416.67, 104.17, 520.83),
 (4, 4, '2018-11-02', 416.67, 104.17, 520.83),
 (4, 5, '2018-12-03', 416.67, 104.17, 520.83),
 (4, 6, '2019-01-03', 416.67, 104.17, 520.83),
 (4, 7, '2019-02-03', 416.67, 104.17, 520.83),
 (4, 8, '2019-03-06', 416.67, 104.17, 520.83),
 (4, 9, '2019-04-06', 416.67, 104.17, 520.83),
 (4, 10, '2019-05-07', 416.67, 104.17, 520.83),
 (4, 11, '2019-06-07', 416.67, 104.17, 520.83),
 (4, 12, '2019-07-08', 416.67, 104.17, 520.83),
 (5, 1, '2018-08-19', 833.33, 166.67, 1000.00),
```

(5, 2, '2018-09-18', 833.33, 166.67, 1000.00),
(5, 3, '2018-10-18', 833.33, 166.67, 1000.00),
(7, 1, '2018-08-19', 833.33, 250.00, 1083.33),
(7, 2, '2018-09-18', 833.33, 250.00, 1083.33),
(7, 3, '2018-10-18', 833.33, 250.00, 1083.33),
(7, 4, '2018-11-17', 833.33, 250.00, 1083.33),
(7, 5, '2018-12-17', 833.33, 250.00, 1083.33),
(7, 6, '2019-01-16', 833.33, 250.00, 1083.33);

1) Write an SQL querry and choose name, surname of a client, how many applications did client have and how many contracts were disbursed. Napisz zapytanie SQL wybierające imię, nazwisko Klienta, ile Klient złożył wniosków i ile zostało mu wypłaconych umów:

```
SELECT
C.name,
C.surname,
COUNT(A.applied_date) AS 'No_of_applications',
COUNT(A.issued_date) AS 'No_of_loans_disbursed'
FROM customers C
LEFT JOIN apps A ON C.cust_id = A.cust_id
GROUP BY C.surname, C.name;
```

name	surname	No_of_applications	No_of_loans_disbursed
Przemek	Kaminski	1	1
Jan	Kowalski	2	2
Adam	Nowak	1	0
Michał	Sliwa	1	0
Anna	Wisniewska	1	1
Joanna	Zielinska	1	1

2) Write an SQL query selecting only those customers to whom we have granted loans, and display their surname, average loan period, total loan amount disbursed, and the sum of all installments. Each customer should appear only once.

   Napisz zapytanie SQL wybierające tylko tych Klientów którym wypłaciliśmy pożyczki i wyświetl dla nich nazwisko, średni okres kredytowania, kwotę wypłaconą pożyczek i sumę wszystkich rat. Klient może wystąpić tylko raz:

```
SELECT
sub1.surname,
sub1.sum_total,
sub2.sum_amount,
sub2.avg_period
FROM
(SELECT C.surname, SUM(I.total) AS 'sum_total'
FROM installments I
```

```
LEFT JOIN apps A ON I.app_id = A.app_id
LEFT JOIN customers C ON C.cust_id=A.cust_id
GROUP BY C.surname) AS sub1
JOIN
(SELECT SUM(A.amount) AS 'sum_amount', AVG(A.period) AS 'avg_period',
C.surname
FROM apps A
JOIN customers C
on A.cust_id=C.cust_id
GROUP BY C.surname) AS sub2
ON sub1.surname = sub2.surname;
```

surname	sum_total	sum_amount	avg_period
Kaminski	6249.96	5000	12
Kowalski	12529.98	10000	6
Wisniewska	8625	7500	12
Zielinska	3000	2500	3

3) Write an SQL query displaying the sum of installments to be paid by customers per month (yyyy-mm)
Napisz zapytanie SQL wyświetlające po miesiącach (yyyy-mm) sumę rat które będą spłacane przez Klientów:

```
SELECT
strftime('%Y-%m', I.due_date) AS month,
SUM(I.total) AS total_installments
FROM installments I
JOIN apps A ON I.app_id = A.app_id
GROUP BY month;
```

month	total_installments
2018-07	9625
2018-08	16779.94
2018-09	1000
2018-10	1000
2018-11	1000
2018-12	1000

4) Write an SQL query selecting customers who have been granted loans. Display the loan amount, the already paid amount and the amount they still have to pay.
Napisz zapytanie SQL wybierające Klientów tylko tych którym wypłaciliśmy pożyczki. Wyświetl dla nich nazwisko, kwotę udzielonej pożyczki, kwotę spłaconej pożyczki oraz kwotę do zapłaty.

```
SELECT
sub1.surname, sub1.sum_total,
sub2.sum_paid_amount,
(sub1.sum_total-COALESCE(sub2.sum_paid_amount,0)) AS
'amount_to_pay'
FROM
(SELECT
C.surname,
A.cust_id AS 'cust_idA',
SUM(I.total) AS 'sum_total'
FROM installments I
JOIN apps A ON I.app_id = A.app_id
JOIN customers C ON A.cust_id=C.cust_id
Group BY C.surname) AS sub1
LEFT JOIN
(SELECT
P.app_id,
SUM(P.paid_amount) AS 'sum_paid_amount',
A.cust_id AS 'cust_idA'
FROM payments P
JOIN apps A ON P.app_id = A.app_id
GROUP BY A.cust_id) AS sub2
ON sub1.cust_idA = sub2.cust_idA;
```

surname	sum_total	sum_paid_amount	amount_to_pay
Kaminski	6249.96	1040	5209.96
Kowalski	12529.98	6900	5629.98
Wisniewska	8625	NULL	8625
Zielinska	3000	NULL	3000