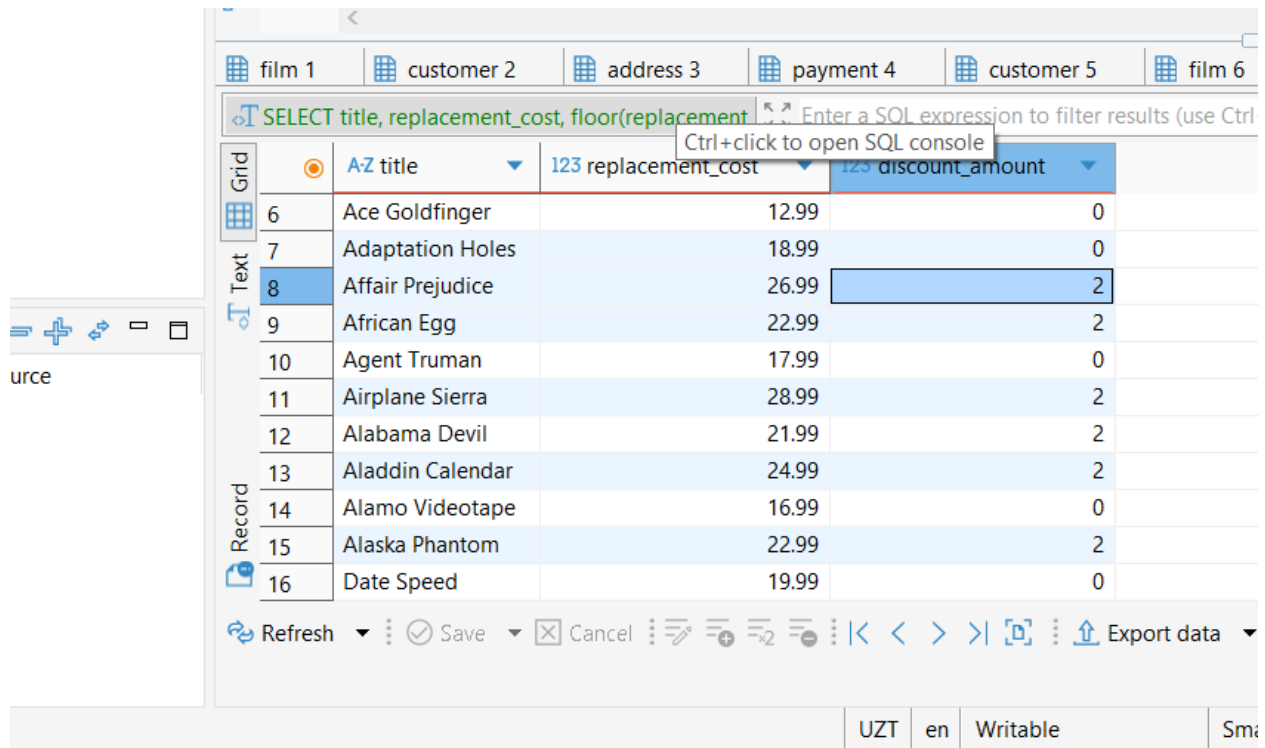


Built - in SQL functions

1. SELECT

```
title,  
replacement_cost,  
floor(replacement_cost / 20 ) * 20 * 0.1 AS discount_amount  
FROM film
```

Izoh: har bir 20 dan oshgan haridalarga 10 fozilik chegirma berilishi uchun kod



The screenshot shows a database application interface. At the top, there are tabs for different tables: film 1, customer 2, address 3, payment 4, customer 5, and film 6. Below the tabs is a SQL query editor with the following query: `SELECT title, replacement_cost, floor(replacement_cost / 20) * 20 * 0.1 AS discount_amount`. A tooltip above the query says "Enter a SQL expression to filter results (use Ctrl+click to open SQL console)". Below the query editor is a table with 4 columns: an index column, a title column, a replacement_cost column, and a discount_amount column. The table contains 11 rows of data. The first row is highlighted in blue. The second row is highlighted in blue. The third row is highlighted in blue. The fourth row is highlighted in blue. The fifth row is highlighted in blue. The sixth row is highlighted in blue. The seventh row is highlighted in blue. The eighth row is highlighted in blue. The ninth row is highlighted in blue. The tenth row is highlighted in blue. The eleventh row is highlighted in blue. The table is displayed in a grid view. On the left side of the table, there are icons for "Grid", "Text", and "Record". At the bottom of the table, there are buttons for "Refresh", "Save", "Cancel", and "Export data".

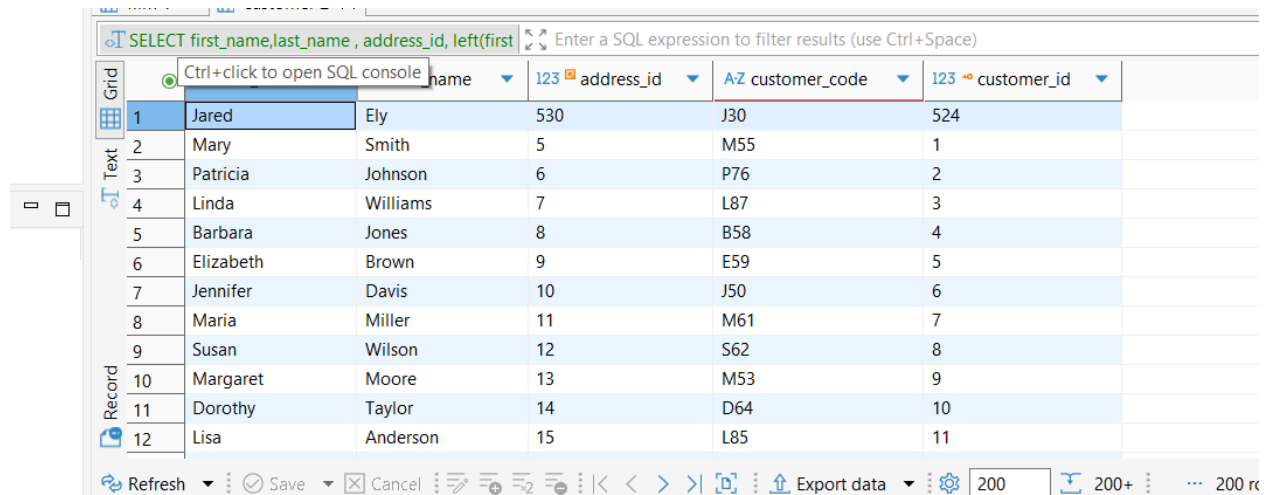
	A-Z title	123 replacement_cost	123 discount_amount
6	Ace Goldfinger	12.99	0
7	Adaptation Holes	18.99	0
8	Affair Prejudice	26.99	2
9	African Egg	22.99	2
10	Agent Truman	17.99	0
11	Airplane Sierra	28.99	2
12	Alabama Devil	21.99	2
13	Aladdin Calendar	24.99	2
14	Alamo Videotape	16.99	0
15	Alaska Phantom	22.99	2
16	Date Speed	19.99	0

```

2. SELECT first_name,last_name , address_id,
    left(first_name , 1) || " " || length(last_name) || " " || right(address_id::text , 1) AS customer_code,
    customer_id
FROM customer

```

Izoh: bunda mijozlarga maxsuss kod yaratiladi



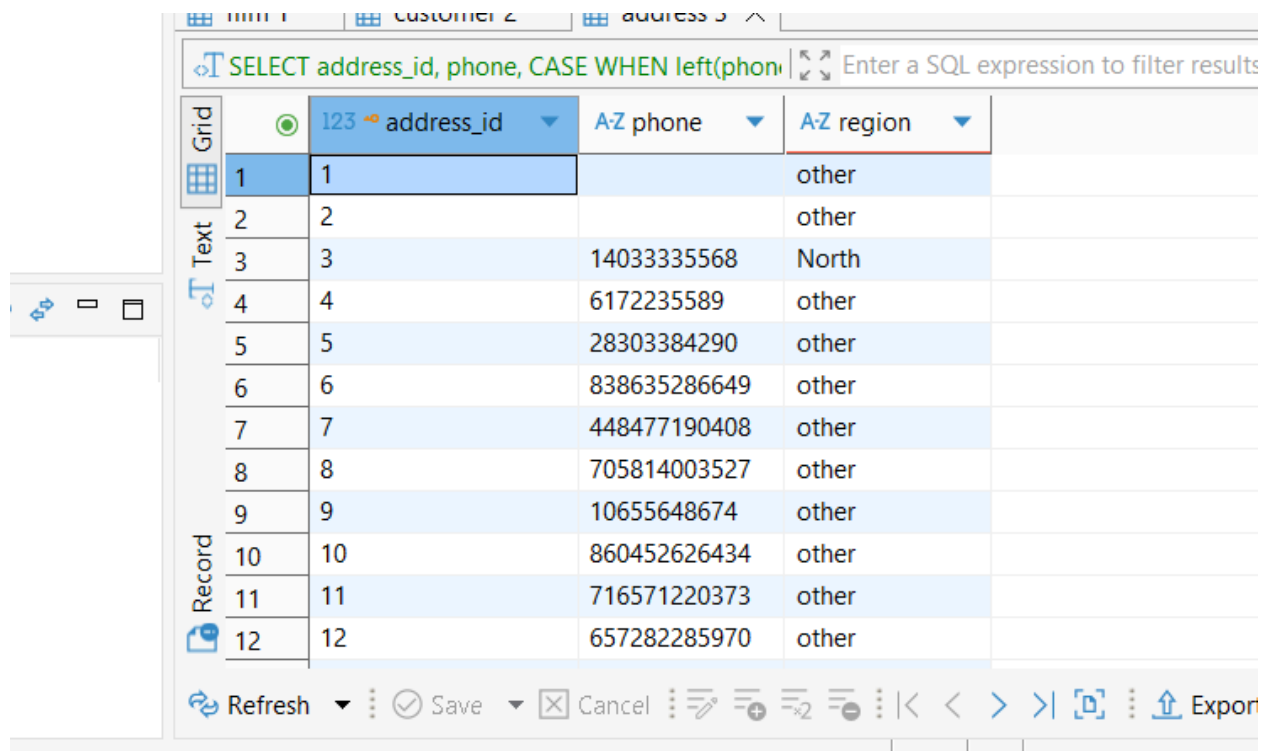
The screenshot shows a database application interface. At the top, there is a SQL query editor with the following query: `SELECT first_name,last_name , address_id, left(first_name , 1) || " " || length(last_name) || " " || right(address_id::text , 1) AS customer_code, customer_id FROM customer`. Below the query editor is a table with 6 columns: an index column, first_name, last_name, address_id, customer_code, and customer_id. The table contains 12 rows of data. The first row is highlighted in blue. On the left side of the table, there are tabs for 'Grid' and 'Text', and a 'Record' icon. At the bottom of the table, there is a toolbar with buttons for 'Refresh', 'Save', 'Cancel', and 'Export data', along with a pagination control showing '200' records.

	name	address_id	customer_code	customer_id
1	Jared	Ely	J30	524
2	Mary	Smith	M55	1
3	Patricia	Johnson	P76	2
4	Linda	Williams	L87	3
5	Barbara	Jones	B58	4
6	Elizabeth	Brown	E59	5
7	Jennifer	Davis	J50	6
8	Maria	Miller	M61	7
9	Susan	Wilson	S62	8
10	Margaret	Moore	M53	9
11	Dorothy	Taylor	D64	10
12	Lisa	Anderson	L85	11

3.SELECT

```
address_id,  
phone,  
CASE  
    WHEN left(phone::text , 3) IN ('140' , '150') THEN 'North'  
    WHEN left(phone::text , 3) IN ('160') THEN 'East'  
    ELSE 'other'  
END AS region  
FROM address
```

Izoh : bu yerda mijozlarni telefon raqamlari boyicha ularni 3 guruhga bolib chiqilayapti



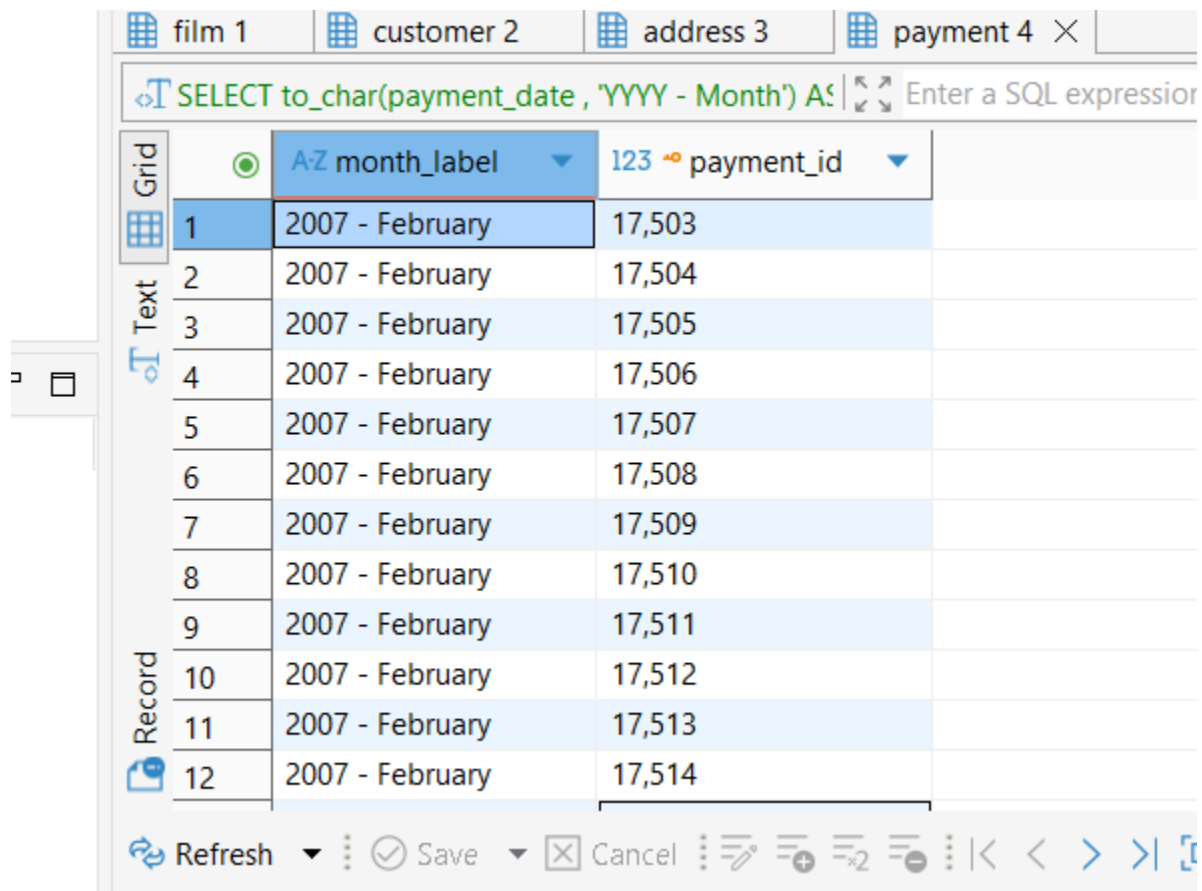
The screenshot shows a database application interface. At the top, a SQL query is entered in a text box: `SELECT address_id, phone, CASE WHEN left(phone::text , 3) IN ('140' , '150') THEN 'North' WHEN left(phone::text , 3) IN ('160') THEN 'East' ELSE 'other' END AS region FROM address`. Below the query, a table grid displays the results. The table has four columns: `address_id`, `phone`, and `region`. The `region` column is calculated based on the first three digits of the `phone` number. The results are as follows:

address_id	phone	region
1		other
2		other
3	14033335568	North
4	6172235589	other
5	28303384290	other
6	838635286649	other
7	448477190408	other
8	705814003527	other
9	10655648674	other
10	860452626434	other
11	716571220373	other
12	657282285970	other

The interface includes a sidebar with icons for Grid, Text, and Record views. The bottom of the window features a toolbar with buttons for Refresh, Save, Cancel, and Export.

4. **SELECT** **to_char**(payment_date , 'YYYY - Month') **AS** month_label,
payment_id
FROM payment

izoh : bu yerda datani berilgan tartibda chiqarish uchun dastlan uni text holatiga ornatib keyin aytilgan tartibda joylashtiriladi



The screenshot shows a database application window with a tabbed interface. The active tab is 'payment 4'. The SQL editor at the top contains the query: `SELECT to_char(payment_date , 'YYYY - Month') AS month_label, payment_id FROM payment`. Below the editor, a table displays the results. The table has two columns: 'month_label' and 'payment_id'. The 'month_label' column contains the string '2007 - February' for all 12 rows. The 'payment_id' column contains values from 17,503 to 17,514. The table is currently in 'Text' view, as indicated by the 'Text' button being selected in the left sidebar. The bottom of the window features a toolbar with buttons for 'Refresh', 'Save', 'Cancel', and various navigation icons.

	AZ month_label	123 payment_id
1	2007 - February	17,503
2	2007 - February	17,504
3	2007 - February	17,505
4	2007 - February	17,506
5	2007 - February	17,507
6	2007 - February	17,508
7	2007 - February	17,509
8	2007 - February	17,510
9	2007 - February	17,511
10	2007 - February	17,512
11	2007 - February	17,513
12	2007 - February	17,514

5. **SELECT** customer_id, email
FROM customer
WHERE email **NOT LIKE** '%@%';

Izoh: email @ belgisi yoqlarini chiqarish uhcun yozilgan kod

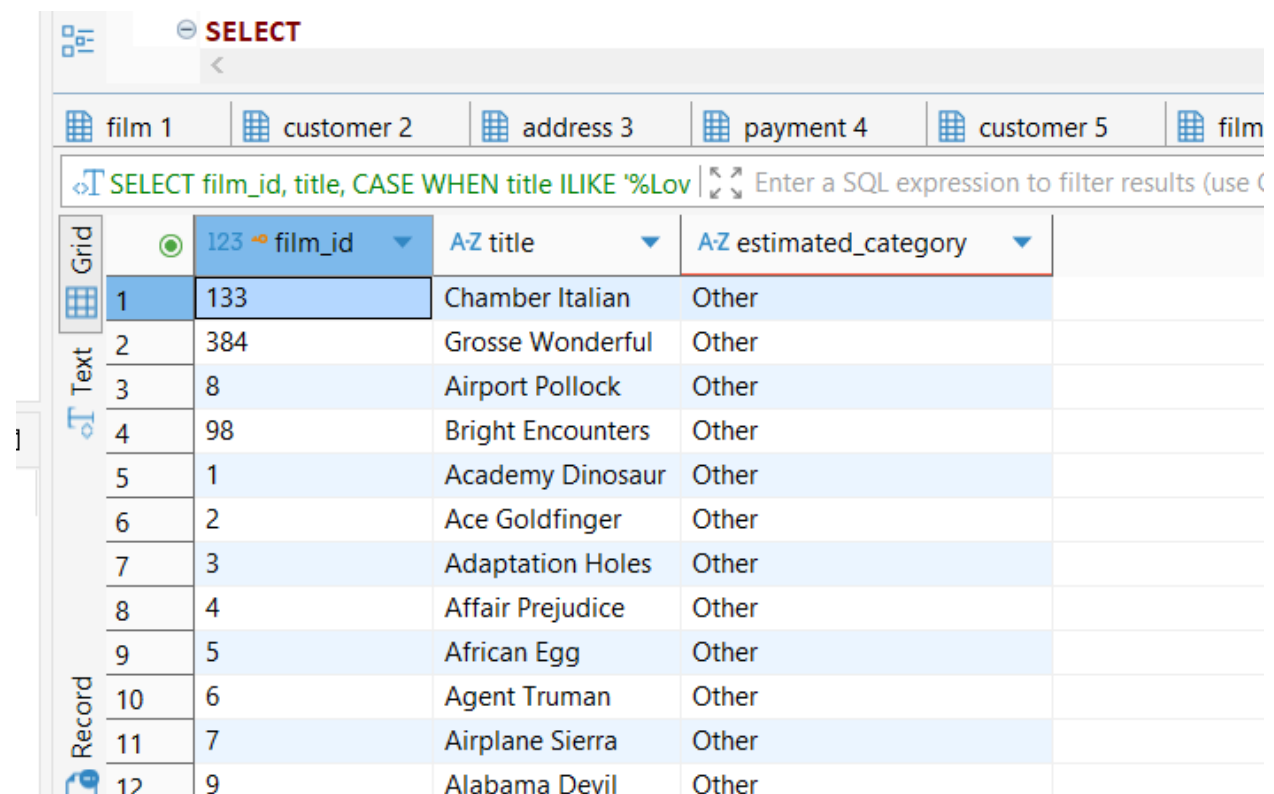
The screenshot shows a database application window with multiple tabs: 'film 1', 'customer 2', 'address 3', 'payment 4', and 'customer 5'. The 'customer 2' tab is active. The SQL query editor displays the query: `SELECT customer_id, email FROM customer WHERE`. Below the query editor, there is a table with columns 'customer_id' and 'email'. The first row shows the value '123' for 'customer_id' and 'A-Z email' for 'email'. The table is currently empty of data rows.

customer_id	email
-------------	-------

6.SELECT

```
film_id,  
title,  
CASE  
    WHEN title ILIKE '%Love%' OR title ILIKE '%Wedding%' THEN 'Romance'  
    WHEN title ILIKE '%Fight%' OR title ILIKE '%War%' THEN 'Action'  
    ELSE 'Other'  
END AS estimated_category  
FROM film
```

Izoh: filmlarni berilgan tartibda 3 guruhga bo'lish uchun case foydalanamiz



	123 film_id	A-Z title	A-Z estimated_category
1	133	Chamber Italian	Other
2	384	Grosse Wonderful	Other
3	8	Airport Pollock	Other
4	98	Bright Encounters	Other
5	1	Academy Dinosaur	Other
6	2	Ace Goldfinger	Other
7	3	Adaptation Holes	Other
8	4	Affair Prejudice	Other
9	5	African Egg	Other
10	6	Agent Truman	Other
11	7	Airplane Sierra	Other
12	9	Alabama Devil	Other

7.SELECT

```
film_id,  
rental_duration,  
rental_duration * 24 * 60 AS minutes
```

FROM film

Izoh: filmlarning ijaraga berilish kunlarni soatlarda chiqariladi

film 1

customer 2

address 3

payment 4

customer 5

film 6

SELECT film_id, rental_duration, rental_duration * ;

Enter a SQL expression to filter results (use Ctrl+S)

Grid

123 film_id

123 rental_duration

123 minutes

1

133

7

10,080

2

384

5

7,200

3

8

6

8,640

4

98

4

5,760

5

1

6

8,640

6

2

3

4,320

7

3

7

10,080

8

4

5

7,200

9

5

6

8,640

10

6

3

4,320

11

7

6

8,640

12

9

3

4,320

Refresh

Save

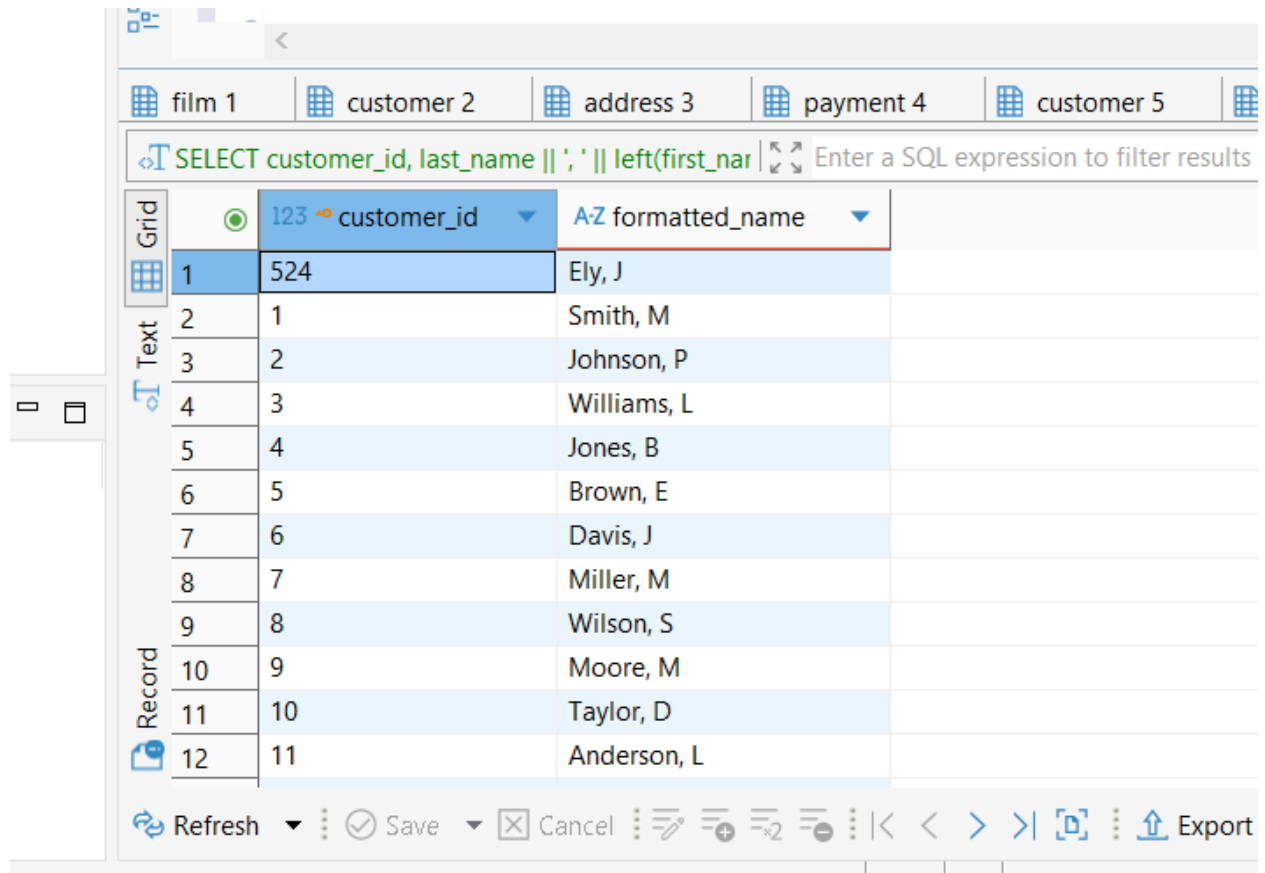
Cancel

Export data

8.SELECT

```
customer_id,  
last_name || ', ' || left(first_name, 1) AS formatted_name  
FROM customer
```

Izoh: berilgan tartibda familiya va ism ning 1 harfini chiqarish uchun yozilgan kod



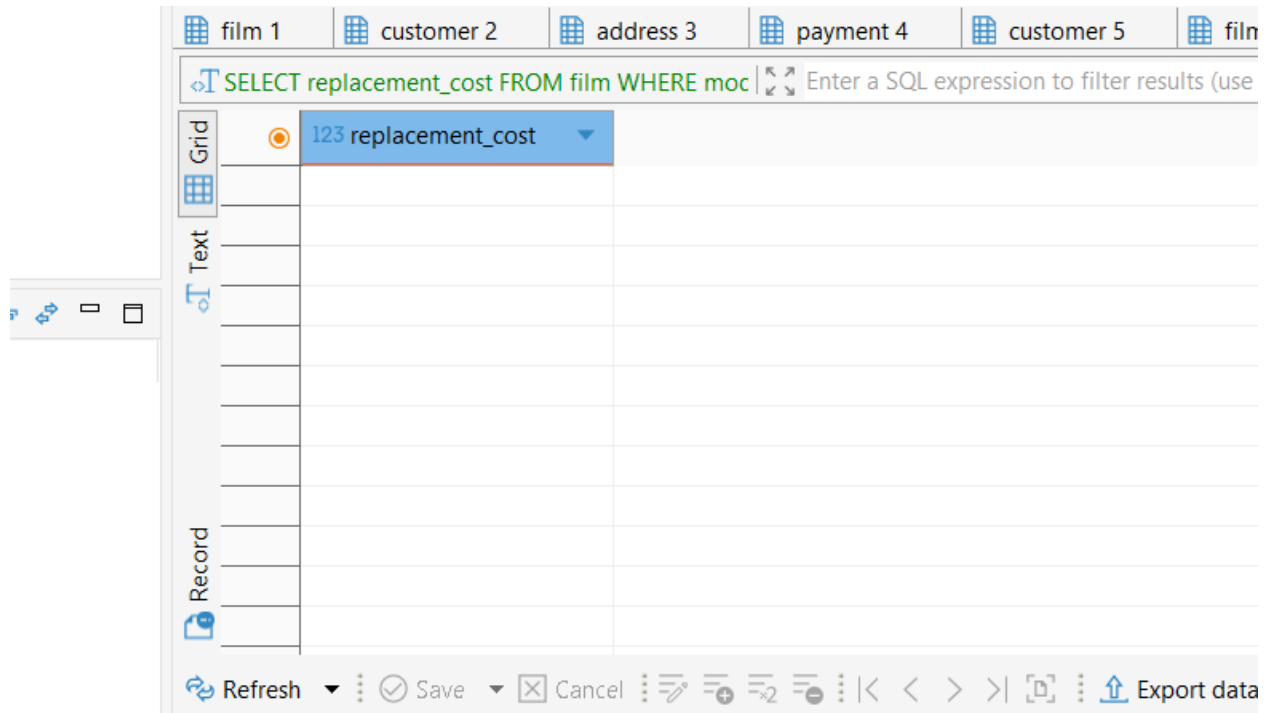
The screenshot shows a database application window with a toolbar at the top containing icons for film, customer, address, payment, and customer tables. Below the toolbar is a SQL query editor with the text: `SELECT customer_id, last_name || ', ' || left(first_name, 1) AS formatted_name`. Below the query editor is a grid view of the results. The grid has three columns: a row number column, a column for `customer_id`, and a column for `AZ formatted_name`. The grid contains 12 rows of data. On the left side of the grid, there are three vertical tabs labeled 'Grid', 'Text', and 'Record', with 'Grid' being the active tab. At the bottom of the window is a toolbar with buttons for 'Refresh', 'Save', 'Cancel', and 'Export', along with various navigation icons.

	customer_id	AZ formatted_name
1	524	Ely, J
2	1	Smith, M
3	2	Johnson, P
4	3	Williams, L
5	4	Jones, B
6	5	Brown, E
7	6	Davis, J
8	7	Miller, M
9	8	Wilson, S
10	9	Moore, M
11	10	Taylor, D
12	11	Anderson, L

9.SELECT

```
replacement_cost  
FROM film  
WHERE mod(replacement_cost , 1) = 0
```

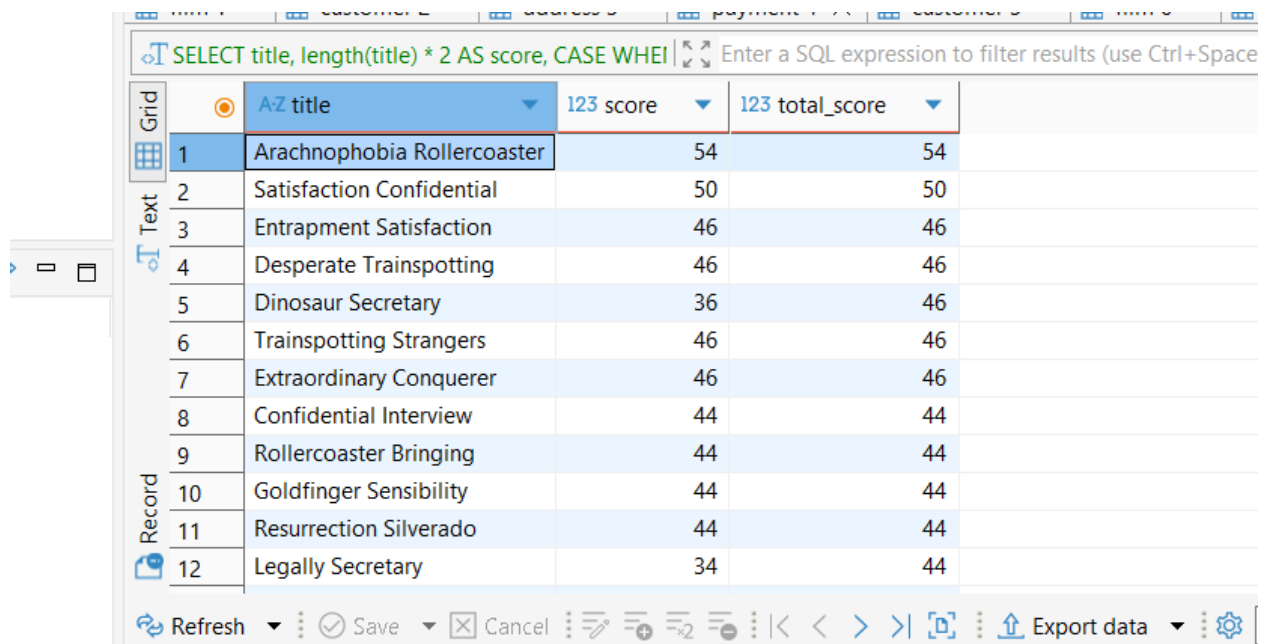
Izoh: bu yerda filmning ijaraga berilish narxining butun qiymatlari bormi yoqmi shuni aniqlanadi buni mod da qilingan



10.SELECT

```
title,  
length(title) * 2 AS score,  
CASE  
    WHEN title ILIKE '%Secret%' THEN length(title) * 2 + 10  
    ELSE length(title) * 2  
END AS total_score  
FROM film  
ORDER BY total_score DESC
```

Izoh: bu kodda filmning titlega kora unga ball beriladi va agar berilgan jumla bolsa film titleda unga qoshimcha ball beriladdi



	A-Z title	123 score	123 total_score
1	Arachnophobia Rollercoaster	54	54
2	Satisfaction Confidential	50	50
3	Entrapment Satisfaction	46	46
4	Desperate Trainspotting	46	46
5	Dinosaur Secretary	36	46
6	Trainspotting Strangers	46	46
7	Extraordinary Conquerer	46	46
8	Confidential Interview	44	44
9	Rollercoaster Bringing	44	44
10	Goldfinger Sensibility	44	44
11	Resurrection Silverado	44	44
12	Legally Secretary	34	44

