

ASSIGNMENT-BANKING SYSTEM

TASK 1

1. Create the database named "HMBank"

```
CREATE DATABASE HMBANK;
```

#	Time	Action
1	22:28:36	CREATE DATABASE HMBANK

2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

```
create table Customdet(customer_id int primary key, first_name text, last_name text,
DOB date, email varchar(50), phone_number varchar(10), address varchar(150));
```

```
create table Accounts ( account_id int primary key, customer_id int, foreign
key(customer_id) references Customers(customer_id), account_type
varchar(20),balance bigint);
```

```
create table Transactions( transaction_id int primary key, account_id int,
transaction_type varchar(25), amount bigint,transaction_date date, foreign
key(account_id) references Accounts(account_id));
```

accounts
customers
transactions

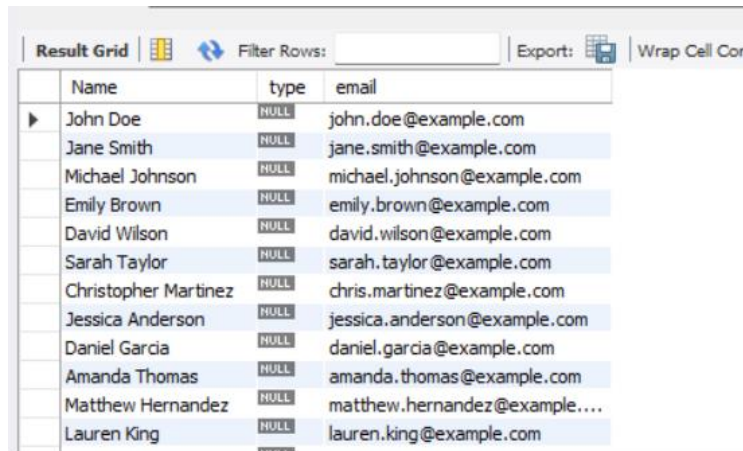
Field	Type	Null	Key	Default	Extra
account_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
account_type	varchar(20)	YES		NULL	
balance	bigint	YES		NULL	

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
first_name	text	YES		NULL	
last_name	text	YES		NULL	
DOB	date	YES		NULL	
email	varchar(50)	YES		NULL	
phone_number	varchar(10)	YES		NULL	
address	varchar(150)	YES		NULL	

TASK 2

1. Write a SQL query to retrieve the name, account type and email of all customers.

```
select concat(first_name," ",last_name) as Name,(select account_type from Accounts where Accounts.customer_id=Custdet.customer_id)as type ,email from Custdet;
```

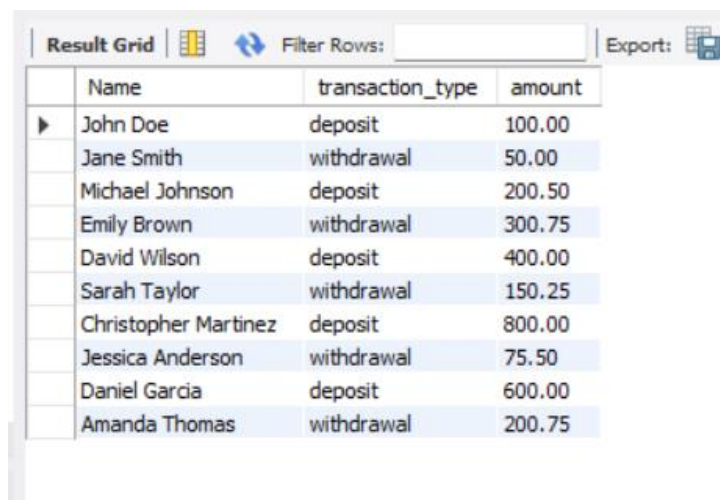


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with columns: Name, type, and email. The 'type' column has a small 'NULL' icon next to each value. The table lists 12 customers with their names and email addresses.

Name	type	email
John Doe	NULL	john.doe@example.com
Jane Smith	NULL	jane.smith@example.com
Michael Johnson	NULL	michael.johnson@example.com
Emily Brown	NULL	emily.brown@example.com
David Wilson	NULL	david.wilson@example.com
Sarah Taylor	NULL	sarah.taylor@example.com
Christopher Martinez	NULL	chris.martinez@example.com
Jessica Anderson	NULL	jessica.anderson@example.com
Daniel Garcia	NULL	daniel.garcia@example.com
Amanda Thomas	NULL	amanda.thomas@example.com
Matthew Hernandez	NULL	matthew.hernandez@example....
Lauren King	NULL	lauren.king@example.com

2. Write a SQL query to list all transaction corresponding customer.

```
select concat(c.first_name," ",c.last_name) as Name, t.transaction_type,t.amount from Custdet c join Accounts a on c.customer_id=a.customer_id join Transactions t on a.account_id=t.account_id;
```

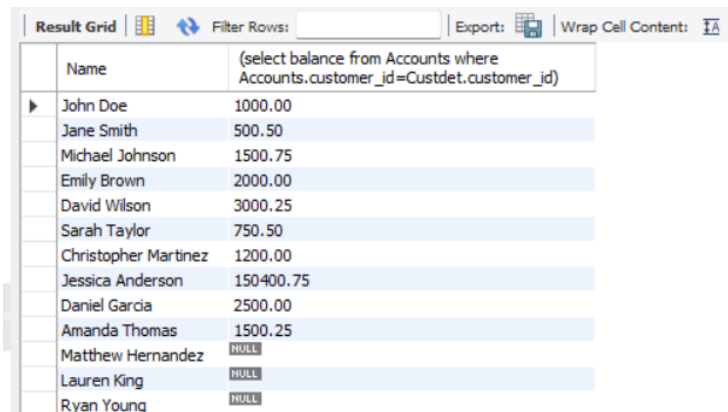


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with columns: Name, transaction_type, and amount. The table lists 10 transactions for different customers, showing the transaction type and the amount.

Name	transaction_type	amount
John Doe	deposit	100.00
Jane Smith	withdrawal	50.00
Michael Johnson	deposit	200.50
Emily Brown	withdrawal	300.75
David Wilson	deposit	400.00
Sarah Taylor	withdrawal	150.25
Christopher Martinez	deposit	800.00
Jessica Anderson	withdrawal	75.50
Daniel Garcia	deposit	600.00
Amanda Thomas	withdrawal	200.75

3. Write a SQL query to increase the balance of a specific account by a certain amount.

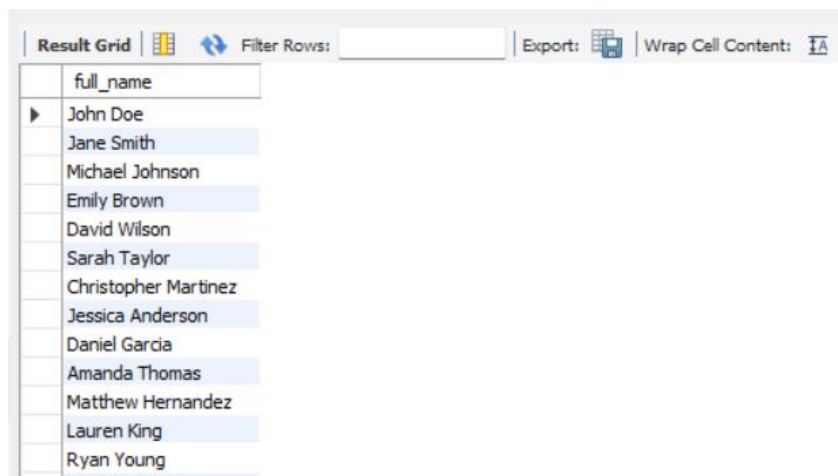
```
update Accounts set balance=(balance+150000) where customer_id=8;  
select concat(first_name," ",last_name) as Name,(select balance from Accounts where  
Accounts.customer_id=Custdet.customer_id) from Custdet;
```



Name	(select balance from Accounts where Accounts.customer_id=Custdet.customer_id)
John Doe	1000.00
Jane Smith	500.50
Michael Johnson	1500.75
Emily Brown	2000.00
David Wilson	3000.25
Sarah Taylor	750.50
Christopher Martinez	1200.00
Jessica Anderson	150400.75
Daniel Garcia	2500.00
Amanda Thomas	1500.25
Matthew Hernandez	NULL
Lauren King	NULL
Ryan Young	NULL

4. Write a SQL query to Combine first and last names of customers as a full name.



```
select concat(first_name," ", last_name ) as full_name from Custdet;
```



full_name
John Doe
Jane Smith
Michael Johnson
Emily Brown
David Wilson
Sarah Taylor
Christopher Martinez
Jessica Anderson
Daniel Garcia
Amanda Thomas
Matthew Hernandez
Lauren King
Ryan Young

5. Write a SQL query to remove accounts with a balance of zero where the account type is savings.


```
delete from Transactions where account_id in( Select account_id from Accounts where Balance=0
and account_type="Savings");
delete from Accounts where balance=0 and account_type="Savings";
```


Result Grid			 Filter Rows:	
	full_name	account_type	balance	
▶	John Doe	Savings	50000	
	Jane Smith	John Doe	10000	
	Michael Johnson	Savings	75000	
	Emily Brown	Checking	25000	
	David Wilson	Savings	100000	
	Sarah Taylor	Savings	60000	
	Christopher Martinez	Checking	20000	
	Jessica Anderson	Savings	840000	
	Daniel Garcia	Checking	30000	
	Amanda Thomas	Savings	80000	
	Matthew Hernandez	Checking	40000	
	Lauren King	Savings	95000	
	Ryan Young	Checking	35000	
	Stephanie Scott	Savings	70000	
	Justin Ramirez	Checking	50000	

6. Write a SQL query to Find customers living in a specific city.

```
select concat(first_name," ",last_name) as Name from Custdet where address like
"%%%%%%%%Banana%%%%%%%%" or address like "%%%%%%%%Cherry%%%%%%%%" ;
```

Result Grid

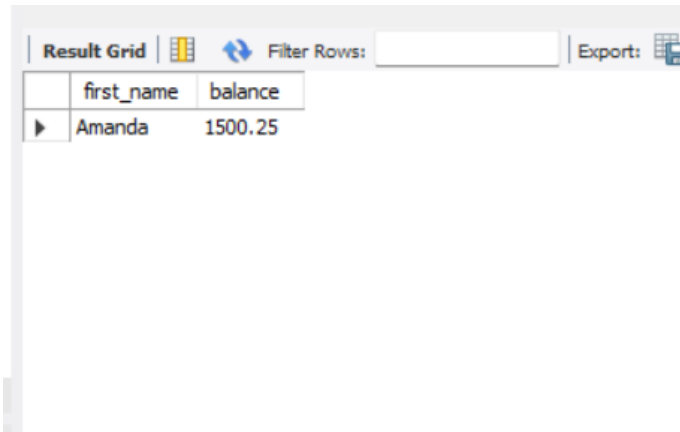



Filter Rows:

	Name
▶	Daniel Garcia
	Ryan Young

7. Write a SQL query to Get the account balance for a specific account.

```
select first_name , balance from Custdet join Accounts on  
Custdet.customer_id=Accounts.customer_id where account_id=10;
```

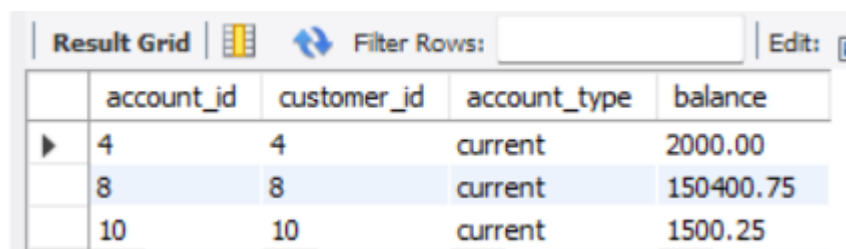


The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' search bar and an 'Export' button. The grid contains one row with the columns 'first_name' and 'balance'.

	first_name	balance
▶	Amanda	1500.25

8. Write a SQL query to List all current accounts with a balance greater than \$1,000.

```
SELECT * FROM accounts WHERE account_type = 'current' AND balance > 1000;
```

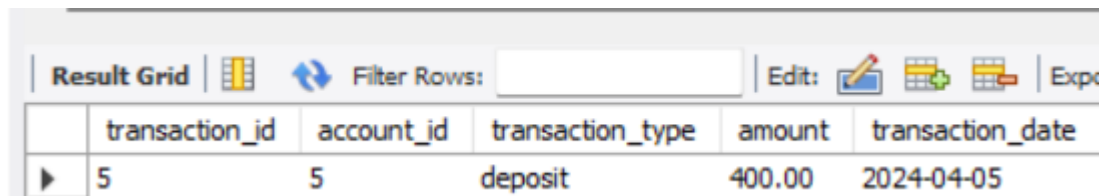


The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' search bar and an 'Edit' button. The grid contains three rows with the columns 'account_id', 'customer_id', 'account_type', and 'balance'.

	account_id	customer_id	account_type	balance
▶	4	4	current	2000.00
	8	8	current	150400.75
	10	10	current	1500.25

9. Write a SQL query to Retrieve all transactions for a specific account.

```
SELECT * FROM transactions WHERE account_id = '5';
```

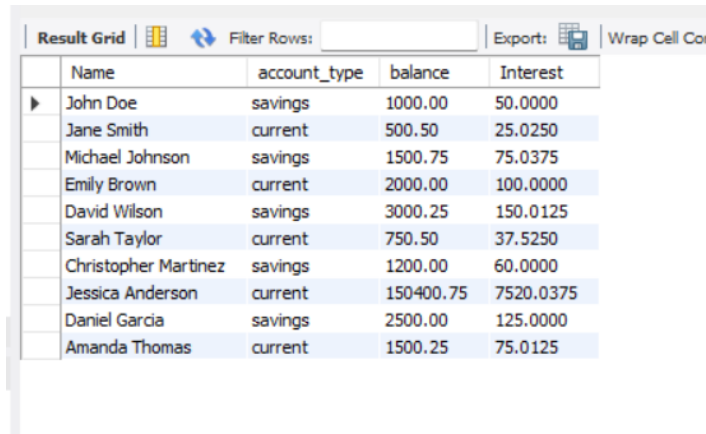


The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' search bar and an 'Edit' button with icons for insert, update, and delete. The grid contains one row with the columns 'transaction_id', 'account_id', 'transaction_type', 'amount', and 'transaction_date'.

	transaction_id	account_id	transaction_type	amount	transaction_date
▶	5	5	deposit	400.00	2024-04-05

10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.

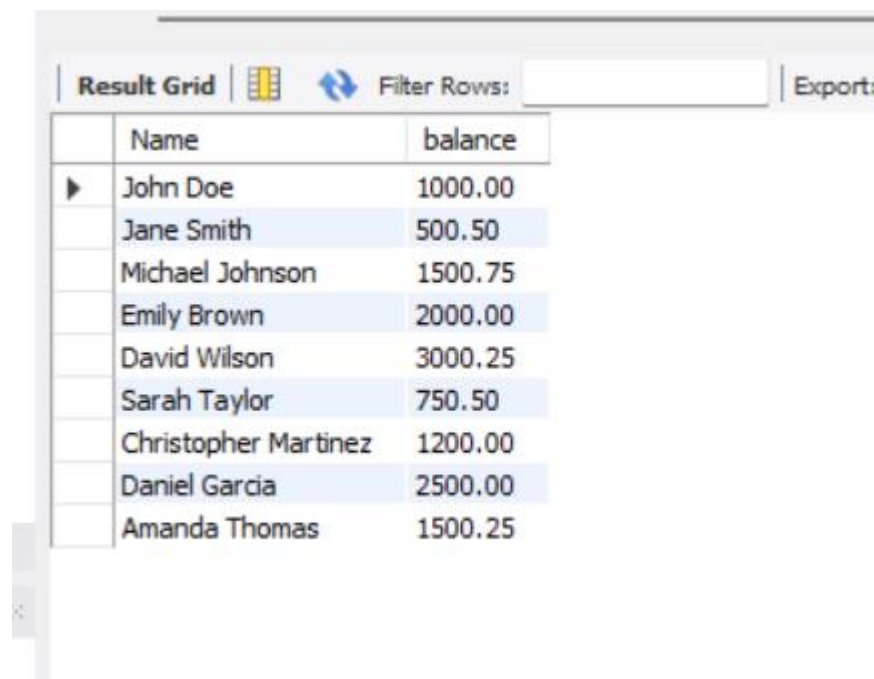
```
select concat(first_name," ",last_name) as Name, account_type, balance, balance*0.05 as Interest
from Custdet join Accounts on Custdet.customer_id=Accounts.customer_id;
```



	Name	account_type	balance	Interest
▶	John Doe	savings	1000.00	50.0000
	Jane Smith	current	500.50	25.0250
	Michael Johnson	savings	1500.75	75.0375
	Emily Brown	current	2000.00	100.0000
	David Wilson	savings	3000.25	150.0125
	Sarah Taylor	current	750.50	37.5250
	Christopher Martinez	savings	1200.00	60.0000
	Jessica Anderson	current	150400.75	7520.0375
	Daniel Garcia	savings	2500.00	125.0000
	Amanda Thomas	current	1500.25	75.0125

11. Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.

```
select concat(first_name," ",last_name) as Name, balance from Custdet join Accounts on
Custdet.customer_id=Accounts.customer_id where balance<50000;
```



	Name	balance
▶	John Doe	1000.00
	Jane Smith	500.50
	Michael Johnson	1500.75
	Emily Brown	2000.00
	David Wilson	3000.25
	Sarah Taylor	750.50
	Christopher Martinez	1200.00
	Daniel Garcia	2500.00
	Amanda Thomas	1500.25

12. Write a SQL query to Find customers not living in a specific city.

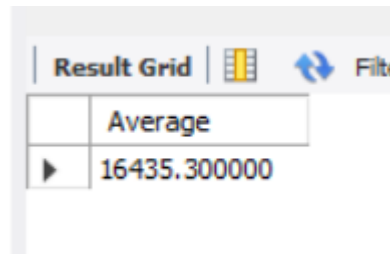
```
select concat(first_name," ",last_name) as Name, address from custdet where address not like "%Banana%";
```

Result Grid			Filter Rows:	Export:
	Name	address		
▶	John Doe	123 Main St, City, Country		
	Jane Smith	456 Elm St, City, Country		
	Michael Johnson	789 Oak St, City, Country		
	Emily Brown	321 Maple St, City, Country		
	David Wilson	654 Pine St, City, Country		
	Sarah Taylor	876 Cedar St, City, Country		
	Christopher Martinez	432 Birch St, City, Country		
	Jessica Anderson	567 Walnut St, City, Country		
	Daniel Garcia	890 Cherry St, City, Country		
	Amanda Thomas	789 Pineapple St, City, Country		
	Matthew Hernandez	345 Mango St, City, Country		
	Lauren King	234 Peach St, City, Country		
	Stephanie Scott	543 Grape St, City, Country		
	Justin Ramirez	876 Orange St, City, Country		

TASK 3

1. Write a SQL query to Find the average account balance for all customers.

```
select avg(balance) As Average from Accounts;
```

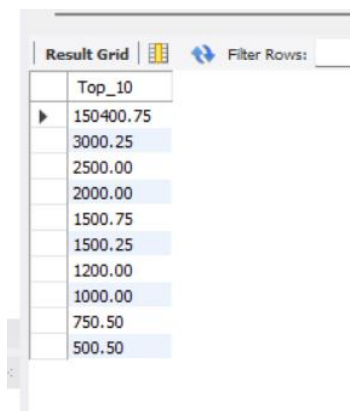


A screenshot of a database application's 'Result Grid'. The grid has two columns. The first column is empty. The second column has a header 'Average' and a single data row with the value '16435.300000'. Above the grid, there are icons for 'Result Grid', a grid icon, a refresh icon, and a 'Filter' button.

	Average
	16435.300000

2. Write a SQL query to Retrieve the top 10 highest account balances.

```
select balance as Top_10 from Accounts order by(balance) desc limit 10 ;
```

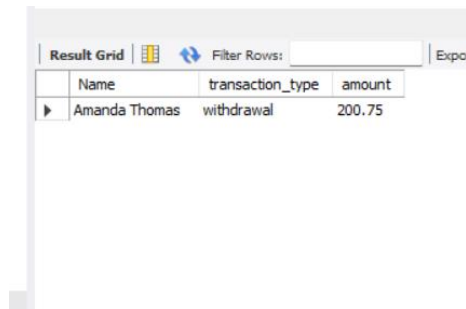


A screenshot of a database application's 'Result Grid'. The grid has two columns. The first column is labeled 'Top_10' and contains a small triangle icon. The second column contains the top 10 highest account balances in descending order. Above the grid, there are icons for 'Result Grid', a grid icon, a refresh icon, and a 'Filter Rows:' button.

Top_10	
	150400.75
	3000.25
	2500.00
	2000.00
	1500.75
	1500.25
	1200.00
	1000.00
	750.50
	500.50

3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

```
select concat(first_name," ",last_name) as Name,transaction_type, amount from Custdet  
join Accounts on Custdet.customer_id=Accounts.customer_id join Transactions on  
Accounts.account_id=Transactions.account_id where transaction_date="2024-04-10";
```

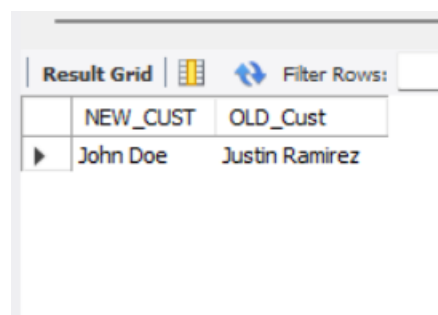


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The columns are labeled 'Name', 'transaction_type', and 'amount'. The data in the row is 'Amanda Thomas', 'withdrawal', and '200.75'.

Name	transaction_type	amount
Amanda Thomas	withdrawal	200.75

4. Write a SQL query to Find the Oldest and Newest Customers.

```
SELECT MAX(CASE WHEN customer_id = min_id THEN CONCAT(first_name, ' ', last_name)  
END) AS NEW_CUST, MAX(CASE WHEN customer_id = max_id THEN CONCAT(first_name, ' ',  
last_name) END) AS OLD_Cust FROM Custdet  
CROSS JOIN  
(SELECT MIN(customer_id) AS min_id, MAX(customer_id) AS max_id FROM Custdet) AS  
subquery;
```

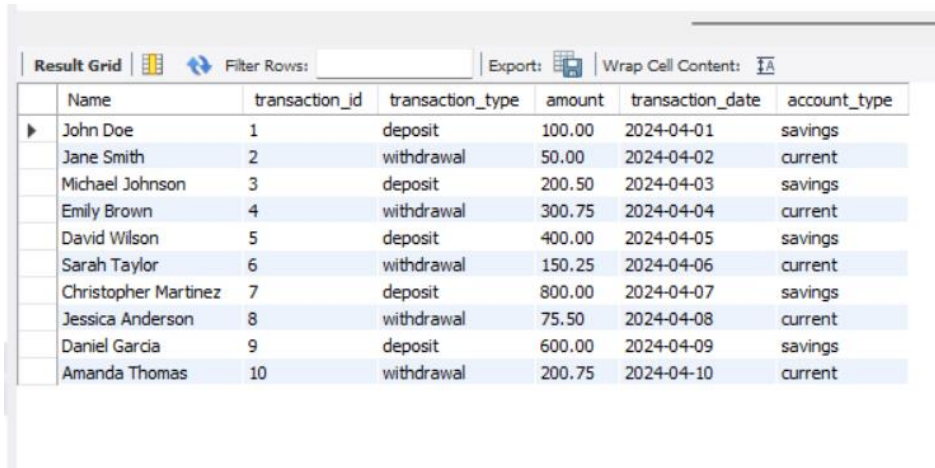


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The columns are labeled 'NEW_CUST' and 'OLD_Cust'. The data in the row is 'John Doe' and 'Justin Ramirez'.

NEW_CUST	OLD_Cust
John Doe	Justin Ramirez

5. Write a SQL query to Retrieve transaction details along with the account type.

```
select concat(first_name," ",last_name) as Name,  
transaction_id,transaction_type,amount,transaction_date, account_type from Custdet join  
Accounts on Custdet.customer_id=Accounts.customer_id join Transactions on  
Accounts.account_id=Transactions.account_id;
```

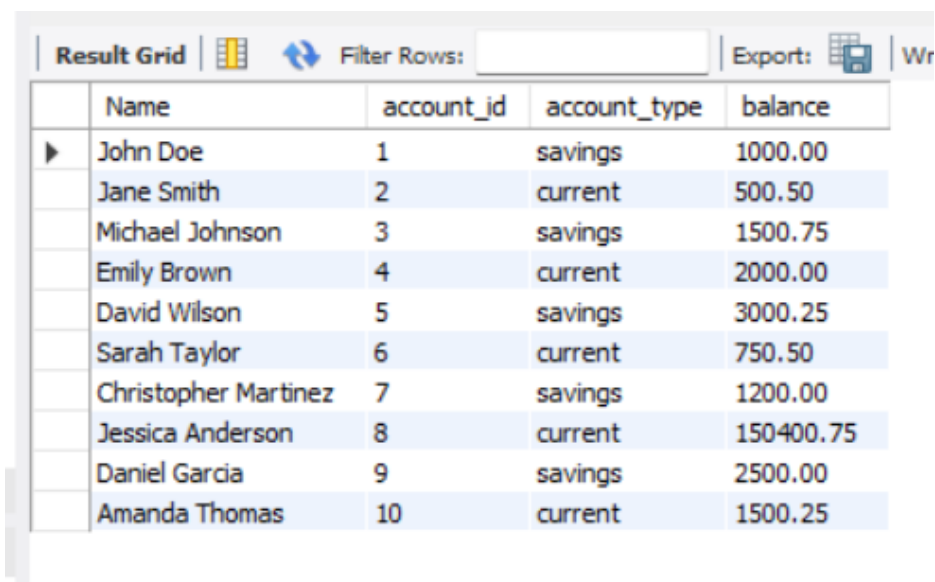


The screenshot shows a database query result grid with the following data:

	Name	transaction_id	transaction_type	amount	transaction_date	account_type
▶	John Doe	1	deposit	100.00	2024-04-01	savings
	Jane Smith	2	withdrawal	50.00	2024-04-02	current
	Michael Johnson	3	deposit	200.50	2024-04-03	savings
	Emily Brown	4	withdrawal	300.75	2024-04-04	current
	David Wilson	5	deposit	400.00	2024-04-05	savings
	Sarah Taylor	6	withdrawal	150.25	2024-04-06	current
	Christopher Martinez	7	deposit	800.00	2024-04-07	savings
	Jessica Anderson	8	withdrawal	75.50	2024-04-08	current
	Daniel Garcia	9	deposit	600.00	2024-04-09	savings
	Amanda Thomas	10	withdrawal	200.75	2024-04-10	current

6. Write a SQL query to Get a list of customers along with their account details.

```
select concat(first_name," ",last_name) as Name, account_id,account_type,balance from  
Custdet join Accounts on Custdet.customer_id=Accounts.customer_id;
```

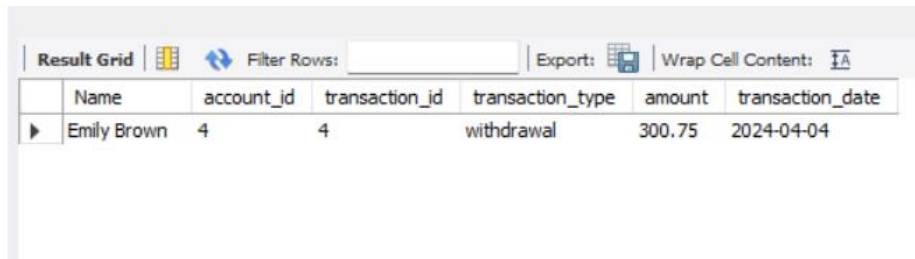


The screenshot shows a database query result grid with the following data:

	Name	account_id	account_type	balance
▶	John Doe	1	savings	1000.00
	Jane Smith	2	current	500.50
	Michael Johnson	3	savings	1500.75
	Emily Brown	4	current	2000.00
	David Wilson	5	savings	3000.25
	Sarah Taylor	6	current	750.50
	Christopher Martinez	7	savings	1200.00
	Jessica Anderson	8	current	150400.75
	Daniel Garcia	9	savings	2500.00
	Amanda Thomas	10	current	1500.25

7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

```
select distinct concat(first_name," ",last_name) as Name, Accounts.account_id,  
transaction_id,transaction_type,amount,transaction_date from Custdet join Accounts on  
Custdet.customer_id=Accounts.customer_id join Transactions on  
Accounts.account_id=Transactions.account_id where Accounts.account_id="4";
```

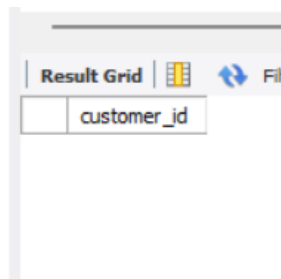


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The columns are: Name, account_id, transaction_id, transaction_type, amount, and transaction_date. The data row shows 'Emily Brown' for Name, '4' for account_id, '4' for transaction_id, 'withdrawal' for transaction_type, '300.75' for amount, and '2024-04-04' for transaction_date.

Name	account_id	transaction_id	transaction_type	amount	transaction_date
Emily Brown	4	4	withdrawal	300.75	2024-04-04

8. Write a SQL query to Identify customers who have more than one account.

```
select customer_id from Accounts group by customer_id having count(*)>1;
```

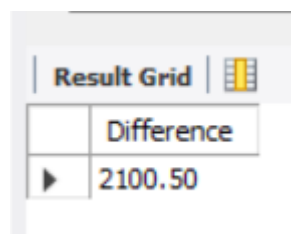


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The column is: customer_id. The data row shows '4' for customer_id.

customer_id
4

9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

```
SELECT SUM(CASE WHEN transaction_type = 'Deposit' THEN amount  
WHEN transaction_type = 'Withdrawal' THEN -amount  
ELSE 0 END) AS Difference  
FROM Transactions;
```

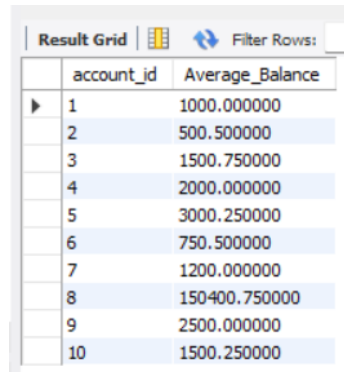


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The column is: Difference. The data row shows '2100.50' for Difference.

Difference
2100.50

10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

```
select account_id , avg(balance) as Average_Balance from Accounts group by account_id;
```

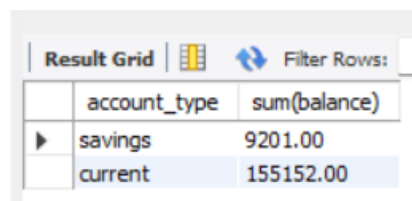


The screenshot shows a 'Result Grid' with two columns: 'account_id' and 'Average_Balance'. It contains 10 rows of data, with the first row selected. The values for 'Average_Balance' are: 1000.000000, 500.500000, 1500.750000, 2000.000000, 3000.250000, 750.500000, 1200.000000, 150400.750000, 2500.000000, and 1500.250000.

account_id	Average_Balance
1	1000.000000
2	500.500000
3	1500.750000
4	2000.000000
5	3000.250000
6	750.500000
7	1200.000000
8	150400.750000
9	2500.000000
10	1500.250000

11. Calculate the total balance for each account type.

```
select account_type , sum(balance) from Accounts group by account_type;
```

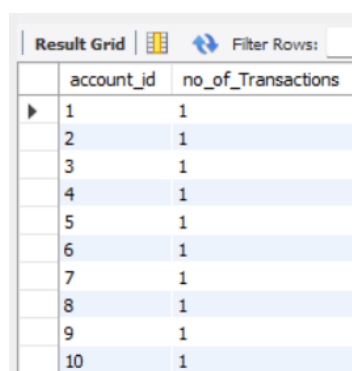


The screenshot shows a 'Result Grid' with two columns: 'account_type' and 'sum(balance)'. It contains two rows of data: 'savings' with a sum of 9201.00, and 'current' with a sum of 155152.00. The first row is selected.

account_type	sum(balance)
savings	9201.00
current	155152.00

12. Identify accounts with the highest number of transactions order by descending order.

```
select account_id, count(account_id) as no_of_Transactions from Transactions group by account_id order by no_of_Transactions desc;
```

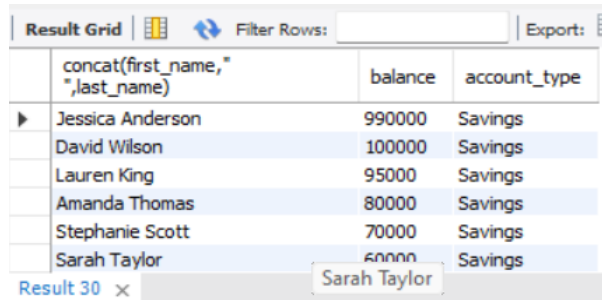


The screenshot shows a 'Result Grid' with two columns: 'account_id' and 'no_of_Transactions'. It contains 10 rows of data, all with a value of 1 for 'no_of_Transactions'. The first row is selected.

account_id	no_of_Transactions
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

13. List customers with high aggregate account balances, along with their account types.

```
select concat(first_name," ",last_name), balance , account_type from Custdet join Accounts  
on Custdet.customer_id=Accounts.customer_id group by first_name order by balance desc;
```

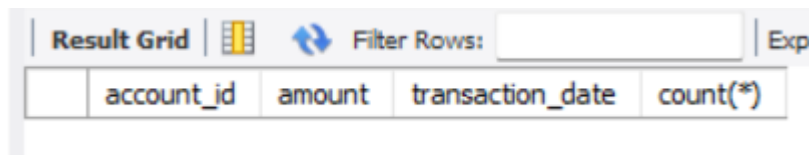


The screenshot shows a database query result grid. The header row contains the following columns: 'concat(first_name," ",last_name)', 'balance', and 'account_type'. The data rows are sorted by balance in descending order. The first row is for Jessica Anderson with a balance of 990,000 and a Savings account type. The second row is for David Wilson with a balance of 100,000 and a Savings account type. The third row is for Lauren King with a balance of 95,000 and a Savings account type. The fourth row is for Amanda Thomas with a balance of 80,000 and a Savings account type. The fifth row is for Stephanie Scott with a balance of 70,000 and a Savings account type. The sixth row is for Sarah Taylor with a balance of 60,000 and a Savings account type. The interface includes a 'Result Grid' tab, a 'Filter Rows' search bar, and an 'Export' button. A status bar at the bottom indicates 'Result 30' and a close button for the Sarah Taylor row.

concat(first_name," ",last_name)	balance	account_type
Jessica Anderson	990000	Savings
David Wilson	100000	Savings
Lauren King	95000	Savings
Amanda Thomas	80000	Savings
Stephanie Scott	70000	Savings
Sarah Taylor	60000	Savings

14. Identify and list duplicate transactions based on transaction amount, date, and account.

```
select account_id, amount, transaction_date, count(*) from Transactions group by  
account_id, amount, transaction_date having count(*)>1;
```



The screenshot shows a database query result grid. The header row contains the following columns: 'account_id', 'amount', 'transaction_date', and 'count(*)'. The interface includes a 'Result Grid' tab, a 'Filter Rows' search bar, and an 'Exp' button.

account_id	amount	transaction_date	count(*)
------------	--------	------------------	----------

TASK 4

1.Retrieve the customer(s) with the highest account balance.

```
select * from Custdet where customer_id=(select customer_id from Accounts where  
balance=(select max(balance) from Accounts));
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

customer_id	first_name	last_name	DOB	email	phone_number	address
8	Jessica	Anderson	1975-08-22	jessica.anderson@example.com	7778889999	567 Walnut St, City, Country
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.Retrieve accounts with transactions whose amounts exceed the average transaction amount.

```
select account_id,amount from transactions  
where amount >(select avg(amount) from transactions);
```

Result Grid	Filter
account_id	amount
4	300.75
5	400.00
7	800.00
9	600.00

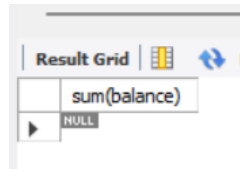
4.Identify customers who have no recorded transactions.

```
SELECT c.customer_id,CONCAT(c.first_name,' ', c.last_name) AS customer_name  
FROM Custdet c  
LEFT JOIN Accounts a ON c.customer_id = a.customer_id LEFT JOIN Transactions t ON  
a.account_id = t.account_id WHERE t.account_id IS NULL;
```

Result Grid	Filter Rows:
customer_id	customer_name
11	Matthew Hernandez
12	Lauren King
13	Ryan Young
14	Stephanie Scott
15	Justin Ramirez

5. Calculate the total balance of accounts with no recorded transactions.

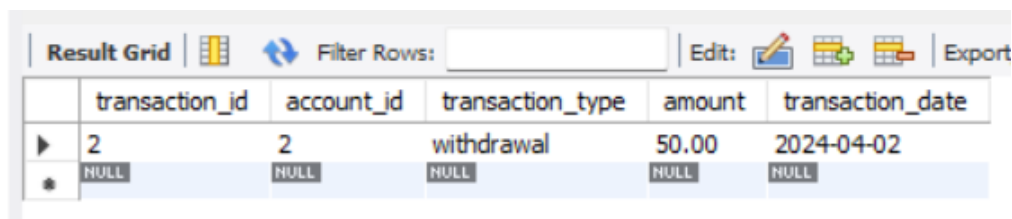
```
select sum(balance) from accounts
where account_id not in (select account_id from transactions);
```



A screenshot of a database query result grid. The grid has a single column labeled 'sum(balance)' and a single row with the value 'NULL'. The interface includes a 'Result Grid' tab and a refresh button.

6. Retrieve transactions for accounts with the lowest balance.

```
SELECT *
FROM transactions
WHERE account_id
IN (SELECT account_id FROM accounts WHERE balance = (SELECT MIN(balance)
FROM accounts));
```

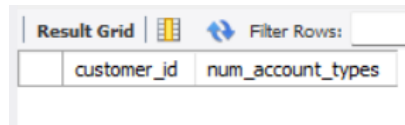


A screenshot of a database query result grid. The grid has columns: transaction_id, account_id, transaction_type, amount, and transaction_date. The first row shows transaction_id 2, account_id 2, transaction_type withdrawal, amount 50.00, and transaction_date 2024-04-02. The second row shows NULL values for all columns. The interface includes a 'Result Grid' tab, a 'Filter Rows' input, and buttons for 'Edit' and 'Export'.

	transaction_id	account_id	transaction_type	amount	transaction_date
▶	2	2	withdrawal	50.00	2024-04-02
★	NULL	NULL	NULL	NULL	NULL

7. Identify customers who have accounts of multiple types.

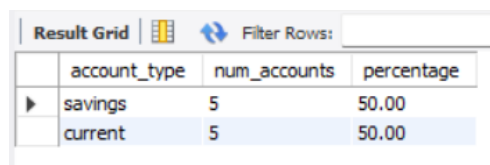
```
SELECT customer_id,COUNT(DISTINCT account_type) AS num_account_types
FROM Accounts
GROUP BY customer_id
HAVING COUNT(DISTINCT account_type) > 1;
```



customer_id	num_account_types
-------------	-------------------

8. Calculate the percentage of each account type out of the total number of accounts.

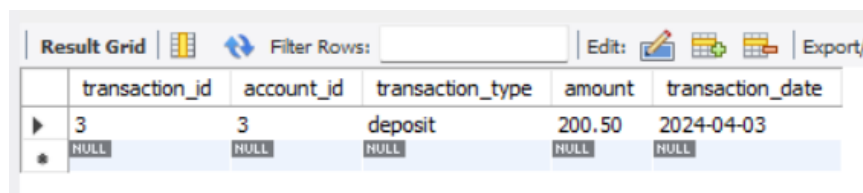
```
SELECT account_type,COUNT(*) AS num_accounts,
ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts), 2) AS percentage
FROM Accounts
GROUP BY account_type;
```



account_type	num_accounts	percentage
savings	5	50.00
current	5	50.00

9. Retrieve all transactions for a customer with a given customer_id.

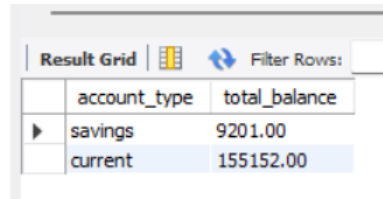
```
SELECT * FROM Transactions WHERE account_id IN (SELECT account_id FROM
Accounts WHERE customer_id = 3 );
```



transaction_id	account_id	transaction_type	amount	transaction_date
3	3	deposit	200.50	2024-04-03
*	NULL	NULL	NULL	NULL

10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

```
(SELECT SUM(balance) FROM Accounts WHERE account_type = a.account_type) AS  
total_balance  
FROM (SELECT DISTINCT account_type FROM Accounts) AS a;
```



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'account_type' and 'total_balance'. There are two rows of data: 'savings' with a total balance of 9201.00, and 'current' with a total balance of 155152.00. The 'current' row is highlighted in blue.

	account_type	total_balance
▶	savings	9201.00
	current	155152.00