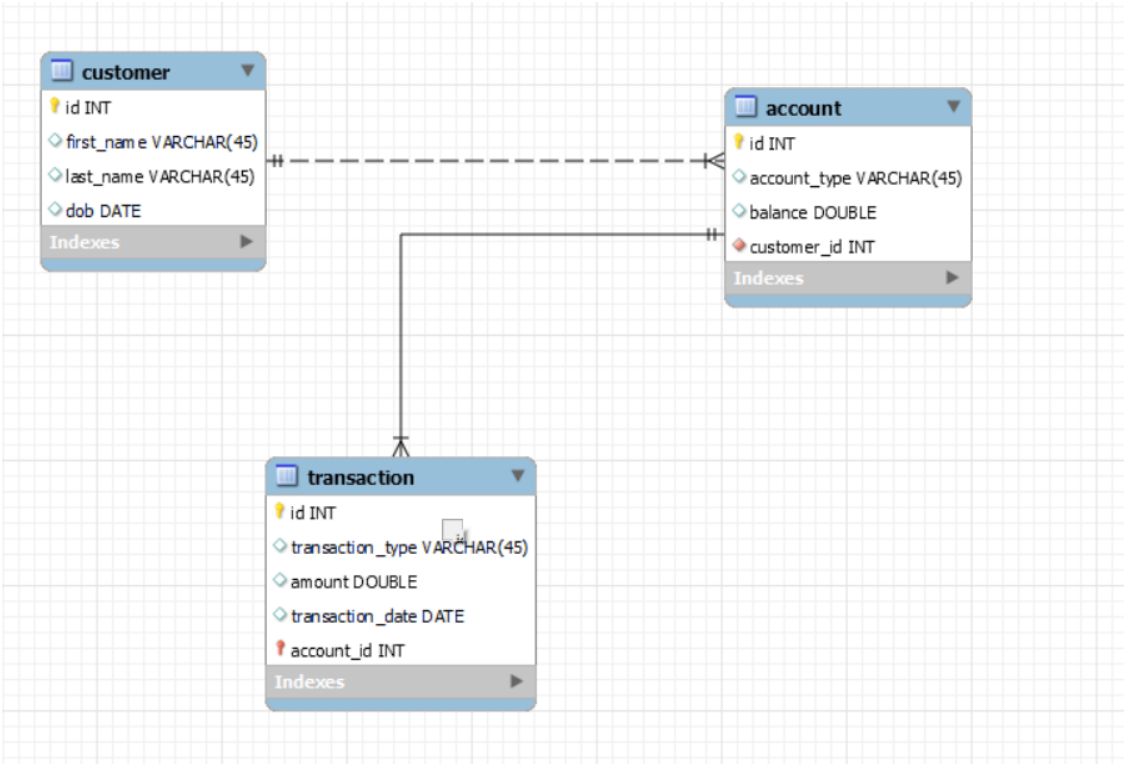
ASSIGNMENT – 3 Banking System

ER DIAGRAM:



CODE:

```
use bank_hex_feb_24;
show tables;
#insertions in customer tables
insert into customer(first_name,last_name,dob) values
('harry','potter','2002-03-21'),
('ronald','weasley','2001-02-10'),
('hermione','granger','2002-11-15');
#insertion into account table
insert into account(account_type,balance,customer_id) values
```

```
('savings',50000,1) ,
('current',120000,2) ,
('zero_balance',100000,3),
('current',150000,1) ,
('savings',30000,3);
#insertion into transaction table

insert into transaction(transaction_type,amount,transaction_date,account_id)
values
('deposit', 10000, '2024-02-01',1),
('withdrawal', 5000, '2024-02-02',1),
('deposit', 20000, '2024-02-02',2),
('withdrawal', 8000, '2024-02-02',3),
('transfer', 20000, '2024-02-01',4),
('transfer', 7000, '2024-02-05',5);

select * from transaction;
/* 1.Write a SQL query to retrieve the name, account type and email of all customers. */
select DISTINCT c.first_name,a.account_type from customer c JOIN account a ON c.id=a.customer_id group by c.first_name;

/*2.Write a SQL query to list all transaction corresponding customer.*/
SELECT distinct
   c.first_name,
   c.last_name,
   t.transaction_type,
```

```
    t.amount,
    t.transaction_date
FROM transaction t
JOIN account a ON t.account_id = a.customer_id
JOIN customer c ON a.customer_id = c.id;

/* 3.Write a SQL query to increase the balance of a specific account by a certain amount*/
UPDATE account
SET balance = balance + 10000
WHERE id=1;
select * from account;

/*4.Write a SQL query to Combine first and last names of customers as a full_name.*/
SELECT distinct CONCAT(c.first_name, ' ', c.last_name) as full_name
FROM customer c;

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

SET SQL_SAFE_UPDATES = 0;
DELETE FROM account
WHERE balance = 0 AND account_type = 'savings';
SET SQL_SAFE_UPDATES = 1;
```

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

*/

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

```
select distinct balance from account where account_type='savings';
```

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

```
select distinct account_type,balance from account where
account_type='current'and balance>82846;
```

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

```
select  distinct t.account_id, t.transaction_type, t.amount, t.transaction_date
from transaction t
where t.account_id =1;
```

/* 10.Write a SQL query to Calculate the interest accrued on savings accounts based on a

given interest rate.*/

```
select CONCAT(c.first_name,' ' ,c.last_name) as customer_name,
    a.balance,(a.balance * 0.05) as estimated_interest
from customer c
inner join account a on c.id = a.customer_id
where a.account_type = 'savings';
```

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

```
overdraft limit.*/
select CONCAT(c.first_name,' ',c.last_name) as customer_name,
a.account_type, a.balance
from customer c
inner join account a on c.id = a.customer_id
where a.balance < -50000;

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

SHOW COLUMNS FROM customer;

/*----------------------------------TASK-3--------------------------*/

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

SELECT AVG(balance) AS average_balance
FROM account;

/*2. Write a SQL query to Retrieve the top 10 highest account balances.
 */
 SELECT CONCAT(c.first_name,' ',c.last_name) as customer_name,
a.account_type, a.balance
```

```
FROM customer c
INNER JOIN account a ON c.id = a.customer_id
ORDER BY a.balance DESC
LIMIT 10;
```

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

*/

```
SELECT CONCAT(c.first_name,' ',c.last_name) as customer_name,
SUM(t.amount) AS total_deposits
FROM customer c
INNER JOIN account a ON c.id = a.customer_id
INNER JOIN transaction t ON a.id = t.account_id
WHERE t.transaction_type = 'deposit' AND t.transaction_date = 'specific_date'
GROUP BY c.id;
```

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

*/

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

```
select distinct a.id,
    t.transaction_type,
    t.amount,
    t.transaction_date,
```

```
    a.account_type
from transaction t inner join account a ON t.account_id = a.id;


/*6. Write a SQL query to Get a list of customers along with their account details*/
select concat(c.first_name,' ',c.last_name) as full_name,
    a.account_type, a.balance
from customer c
inner join account a on c.id = a.customer_id;


/* 7.Write a SQL query to Retrieve transaction details along with customer information for a
specific account.*/
select distinct t.account_id, t.transaction_type, t.amount, t.transaction_date,
    c.first_name, c.last_name, a.account_type, a.balance
from transaction t
inner join account a on t.account_id = a.id
inner join customer c on a.customer_id = c.id
where a.id=1;


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


SELECT CONCAT(c.first_name,' ',c.last_name) as customer_name, COUNT(a.id) as num_accounts
FROM customer c
INNER JOIN account a ON c.id = a.customer_id
```

```
GROUP BY c.id

HAVING num_accounts > 1;


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

SELECT t.transaction_type, SUM(CASE WHEN t.transaction_type = 'deposit'
THEN t.amount ELSE -t.amount END) AS transaction_difference

FROM transaction t

GROUP BY t.transaction_type;


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

period */


SELECT a.id, AVG(a.balance) AS average_daily_balance

FROM account a

INNER JOIN transaction t ON a.id = t.account_id

WHERE t.transaction_date BETWEEN 'start_date' AND 'end_date'

GROUP BY a.id;


/*11.calculate total balance of each account type*/

select account_type, SUM(balance) as total_balance

from account

group by account_type;


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

```
select a.id, a.account_type, a.balance, COUNT(t.id) as transaction_count
from account a
join transaction t on a.id = t.account_id
group by a.id
order by transaction_count DESC;

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

SELECT
   CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
   a.account_type,
   SUM(a.balance) AS aggregate_balance
FROM customer c
JOIN account a ON c.id = a.customer_id
GROUP BY c.id, a.account_type
HAVING aggregate_balance > 100000; -- Adjust the threshold as needed


/*14. Identify and list duplicate transactions based on transaction amount, date, and account.
 */
SELECT
   t.transaction_type,
   t.amount,
```

```
    t.transaction_date,
    t.account_id,
    COUNT(*) AS duplicate_count
FROM transaction t
GROUP BY t.transaction_type, t.amount, t.transaction_date, t.account_id
HAVING duplicate_count > 1;



/*---------------------------------TASK-4-------------------------*/



/*1. Retrieve the customer(s) with the highest account balance.
 */
SELECT
    CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
    MAX(a.balance) AS highest_balance
FROM customer c
JOIN account a ON c.id = a.customer_id;

/* 2. Calculate the average account balance for customers who have more than
one account.
*/
SELECT
    CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
    AVG(a.balance) AS average_balance
FROM customer c
JOIN account a ON c.id = a.customer_id
```

```
GROUP BY c.id
HAVING COUNT(a.id) > 1;

/*3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.
 */
SELECT
   a.account_type,
   t.amount,
   t.transaction_date
FROM account a
JOIN transaction t ON a.id = t.account_id
WHERE t.amount > (SELECT AVG(amount) FROM transaction);

/*4. Identify customers who have no recorded transactions.
 */
 SELECT
   CONCAT(c.first_name, ' ', c.last_name) AS customer_name
FROM customer c
WHERE NOT EXISTS (
   SELECT 1
   FROM transaction t
   WHERE t.account_id IN (SELECT a.id FROM account a WHERE a.customer_id = c.id)
);
```

```
/*5. Calculate the total balance of accounts with no recorded transactions.
 */
SELECT
   a.account_type,
   SUM(a.balance) AS total_balance
FROM account a
WHERE NOT EXISTS (
   SELECT 1
   FROM transaction t
   WHERE t.account_id = a.id
)
GROUP BY a.account_type;

/*6. Retrieve transactions for accounts with the lowest balance.
 */
SELECT
   t.transaction_type,
   t.amount,
   t.transaction_date
FROM transaction t
JOIN account a ON t.account_id = a.id
WHERE a.balance = (SELECT MIN(balance) FROM account);

/*7. Identify customers who have accounts of multiple types.
 */
SELECT
```

```
    CONCAT(c.first_name, ' ', c.last_name) AS customer_name
FROM customer c
WHERE (
    SELECT COUNT(DISTINCT a.account_type)
    FROM account a
    WHERE a.customer_id = c.id
) > 1;

/* 8. Calculate the percentage of each account type out of the total number of accounts.
*/
SELECT
    account_type,
    COUNT(id) AS account_count,
    (COUNT(id) / (SELECT COUNT(id) FROM account)) * 100 AS percentage
FROM account
GROUP BY account_type;


/* 9. Retrieve all transactions for a customer with a given customer_id.
*/
SELECT
    t.transaction_type,
    t.amount,
    t.transaction_date
FROM transaction t
JOIN account a ON t.account_id = a.id
```

```
WHERE a.customer_id = 1; -- Replace 1 with the desired customer_id


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

clause */


SELECT

   t.transaction_type,

   t.amount,

   t.transaction_date

FROM transaction t

JOIN account a ON t.account_id = a.id

WHERE a.customer_id = 1; -- Replace 1 with the desired customer_id
```