```
CREATE DATABASE BANK;
USE BANK;


CREATE OR REPLACE TABLE DISTRICT(
District_Code INT PRIMARY KEY ,
District_Name VARCHAR(100)     ,
Region VARCHAR(100)    ,
No_of_inhabitants          INT,
No_of_municipalities_with_inhabitants_less_499 INT,
No_of_municipalities_with_inhabitants_500_btw_1999  INT,
No_of_municipalities_with_inhabitants_2000_btw_9999             INT,
No_of_municipalities_with_inhabitants_less_10000 INT,
No_of_cities      INT,
Ratio_of_urban_inhabitants       FLOAT,
Average_salary  INT,
No_of_entrepreneurs_per_1000_inhabitants INT,
No_committed_crime_2017       INT,
No_committed_crime_2018 INT
) ;

SELECT UPPER(REGION) FROM DISTRICT ;

-- DELETE FROM DISTRICT

SELECT FORMAT(DATE,'mm-dd-yyyy') AS CURRENT_DATE_TIME FROM transactions;

CREATE OR REPLACE TABLE ACCOUNT(
account_id INT PRIMARY KEY,
district_id      INT,
Date DATE ,
Account_Type VARCHAR(100) ,
frequency        VARCHAR(40),
Card_Assigned VARCHAR(20),
FOREIGN KEY (district_id) references DISTRICT(District_Code)
);

USE bank

SELECT district_id, COUNT(DISTINCT account_id) FROM ACCOUNT GROUP BY 1 ORDER BY 2 DESC ;

SELECT COUNT(distinct district_code) FROM district

CREATE OR REPLACE TABLE ORDER_LIST (
order_id         INT PRIMARY KEY,
account_id       INT,
bank_to          VARCHAR(45),
account_to       INT,
amount FLOAT,
FOREIGN KEY (account_id) references ACCOUNT(account_id)
);

SELECT bank_to FROM order_list WHERE lower(SUBSTRING(bank_to,1,1)) in('a','b')

select bank_to FROM order_list WHERE SUBSTRING(bank_to,-1,1) IN ('k','s')

SELECT COUNT(DISTINCT *) FROM order_list;
```

```
CREATE OR REPLACE TABLE LOAN(
loan_id INT ,
account_id      INT,
Date    DATE,
amount  INT,
duration        INT,
payments        INT,
status VARCHAR(35),
FOREIGN KEY (account_id) references ACCOUNT(account_id)
);

SELECT * FROM LOAN



CREATE OR REPLACE TABLE TRANSACTIONS(
trans_id INT,
account_id      INT,
Date    DATE,
Type    VARCHAR(30),
operation       VARCHAR(40),
amount  INT,
balance FLOAT,
Purpose         VARCHAR(40),
bank    VARCHAR(45),
account_partner_id INT,
FOREIGN KEY (account_id) references ACCOUNT(account_id));
```

```
SELECT YEAR(DATE), COUNT(*) FROM transactions

GROUP BY 1

ORDER BY 2 DESC;


SELECT DISTINCT YEAR(DATE) FROM TRANSACTIONS



CREATE OR REPLACE TABLE CLIENT(

client_id       INT PRIMARY KEY,

Birth_date      DATE,

district_id INT,

Sex     CHAR(10),

FOREIGN KEY (district_id) references DISTRICT(District_Code)

);



CREATE OR REPLACE TABLE DISPOSITION(

disp_id INT PRIMARY KEY,

client_id INT,

account_id      INT,

type CHAR(15),

FOREIGN KEY (account_id) references ACCOUNT(account_id),

FOREIGN KEY (client_id) references CLIENT(client_id)

);



CREATE OR REPLACE TABLE CARD(

card_id INT PRIMARY KEY,
disp_id  INT,
type CHAR(10)   ,
issued DATE,
FOREIGN KEY (disp_id) references DISPOSITION(disp_id)
);

SELECT * FROM card;
------------------------------------------------------------------------------------------


/*

            DATA TRANSFORMATION

*/




/*                                    ADDING AN AGE COLUMN TO CLIENTS TABLE BY SUBTRACTING THE CURRENT
YEAR WITH BIRTH YEAR.                         */


ALTER TABLE CLIENT
ADD COLUMN AGE INT;


UPDATE CLIENT
SET AGE = YEAR('2023-01-01') - YEAR(BIRTH_DATE);


SELECT * FROM CLIENT ;
```

```
/*                                   CHANGING THE YEARS */


UPDATE TRANSACTIONS

SET DATE = DATE_ADD(DATE, INTERVAL 1 YEAR);


/* TOTAL TRANSACTIONS DONE PER YEAR AFTR CHANGING THE YEARS*/


SELECT YEAR(DATE), COUNT(*) FROM transactions GROUP BY 1 ORDER BY 2 DESC;


/* CHECKING THE NULL VALUES IN BANK COLUMN */


SELECT YEAR(DATE),COUNT(*)

FROM TRANSACTIONS

WHERE BANK IS NULL

GROUP BY 1 ;


/* UPDATING NULL VALUES IN THE BANK COLUMN*/


UPDATE TRANSACTIONS

SET BANK = 'Sky Bank' WHERE BANK IS NULL AND YEAR(DATE) = 2022;


UPDATE TRANSACTIONS

SET BANK = 'DBS Bank' WHERE BANK IS NULL AND YEAR(DATE) = 2021;
```

```
UPDATE TRANSACTIONS

SET BANK = 'Northern Bank' WHERE BANK IS NULL AND YEAR(DATE) = 2019;


UPDATE TRANSACTIONS

SET BANK = 'Southern Bank' WHERE BANK IS NULL AND YEAR(DATE) = 2018;


UPDATE TRANSACTIONS

SET BANK = 'ADB Bank' WHERE BANK IS NULL AND YEAR(DATE) = 2017;




/*   What is the demographic profile of the bank's clients and how does it vary across districts?
                        */


CREATE OR REPLACE TABLE DEMOGRAPHIC_DATA_KPI as

SELECT  C.DISTRICT_ID,D.DISTRICT_NAME,D.AVERAGE_SALARY,

ROUND(AVG(C.AGE),0) AS AVG_AGE,

SUM(CASE WHEN C.Sex = 'Male' THEN 1 ELSE 0 END) AS MALE_CLIENT ,

SUM(CASE WHEN C.Sex = 'Female' THEN 1 ELSE 0 END) AS FEMALE_CLIENT ,

ROUND( SUM(CASE WHEN C.Sex = 'Female' THEN 1 ELSE 0 END)/(SUM(CASE WHEN C.Sex = 'Male' THEN
1 ELSE 0 END))*100,2) AS FEMALE_MALE_RATIO_PERC,

COUNT(*)AS TOTAL_CLIENT

FROM CLIENT C

INNER JOIN DISTRICT D ON C.DISTRICT_ID = D.DISTRICT_CODE

GROUP BY 1,2,3
```

```
ORDER BY 1;


SELECT * FROM DEMOGRAPHIC_DATA_KPI


/*                              How the banks have performed obver the years.Give their detailed analysis
month wise?                  */



SELECT * FROM ACC_LATEST_TXNS_WITH_BALANCE ;


SELECT LATEST_TXN_DATE,COUNT(*) AS TOT_TXNS
FROM ACC_LATEST_TXNS_WITH_BALANCE
GROUP BY 1
ORDER BY 2 DESC;



/*   ASSUMING EVERY LAST MONTH CUSTOMER ACCOUNT IS GETTING TXNCTED */


CREATE OR REPLACE TABLE ACC_LATEST_TXNS_WITH_BALANCE
AS
SELECT LTD.*,TXN.BALANCE
FROM TRANSACTIONS AS TXN
INNER JOIN
(
  SELECT ACCOUNT_ID,YEAR(DATE) AS TXN_YEAR,
  MONTH(DATE) AS TXN_MONTH,
  MAX(DATE) AS LATEST_TXN_DATE,  COUNT(*) AS TOTAL_TXNS_PER_MONTH
  FROM TRANSACTIONS
  GROUP BY 1,2,3
```

```
   ORDER BY 1,2,3

) AS LTD ON TXN.ACCOUNT_ID = LTD.ACCOUNT_ID AND TXN.DATE = LTD.LATEST_TXN_DATE
WHERE TXN.TYPE = 'Credit' -- this is the assumptions am having : month end txn data is credit
ORDER BY TXN.ACCOUNT_ID,LTD.TXN_YEAR,LTD.TXN_MONTH;

select * from ACC_LATEST_TXNS_WITH_BALANCE;




/* ----------------------------------------------------------------------------*/



CREATE OR REPLACE TABLE BANKING_KPI AS
SELECT  ALWB.TXN_YEAR , ALWB.TXN_MONTH,T.BANK,A.ACCOUNT_TYPE,

COUNT(DISTINCT ALWB.ACCOUNT_ID) AS TOT_ACCOUNT,
COUNT(DISTINCT T.TRANS_ID) AS TOT_TXNS,
COUNT(CASE WHEN T.TYPE = 'Credit' THEN 1 END) AS DEPOSIT_COUNT ,
COUNT(CASE WHEN T.TYPE = 'Withdrawal' THEN 1 END) AS WITHDRAWAL_COUNT,

SUM(ALWB.BALANCE) AS TOT_BALANCE,

ROUND((COUNT(CASE WHEN T.TYPE = 'Credit' THEN 1 END) / COUNT(DISTINCT T.TRANS_ID) ) * 100,2) AS DEPOSIT_PERC ,
ROUND((COUNT(CASE WHEN T.TYPE = 'Withdrawal' THEN 1 END) / COUNT(DISTINCT T.TRANS_ID)) * 100,2) AS WITHDRAWAL_PERC ,
NVL(SUM(ALWB.BALANCE) / COUNT(DISTINCT ALWB.ACCOUNT_ID),0) AS AVG_BALANCE,
ROUND(COUNT(DISTINCT T.TRANS_ID)/COUNT(DISTINCT ALWB.ACCOUNT_ID),0) AS TPA


FROM TRANSACTIONS AS T

INNER JOIN  ACC_LATEST_TXNS_WITH_BALANCE AS ALWB ON T.ACCOUNT_ID = ALWB.ACCOUNT_ID

LEFT OUTER JOIN  ACCOUNT AS A ON T.ACCOUNT_ID = A.ACCOUNT_ID

GROUP BY 1,2,3,4

ORDER BY 1,2,3,4;




SELECT bank, account_type,sum(withdrawal_count), sum(deposit_count), sum(withdrawal_count+
deposit_count) AS tot FROM  banking_kpi  GROUP BY 1,2 order BY 1


SELECT SUM(WITHDRAWAL_COUNT) FROM BANKING_KPI;


SELECT SUM(WITH_COUNT)AS WITH_C FROM

(SELECT SUM(CASE WHEN TYPE = 'Withdrawal' THEN 1 ELSE 0 END )AS WITH_COUNT, SUM(CASE WHEN
TYPE = 'Credit' THEN 1 ELSE 0 END )AS Cred_COUNT

FROM TRANSACTIONS);


/* -------------------------------------------------------------------------*/


select TXN_YEAR,COUNT(*) AS TOTAL

FROM BANKING_KPI

GROUP BY 1

ORDER BY 2 DESC;


SELECT * FROM BANKING_KPI

ORDER BY txn_year,BANK;
```

```sql
SELECT * FROM TRANSACTIONS

WHERE ACCOUNT_ID = 1

ORDER BY DATE;
```

```sql
SELECT * FROM BANKING_KPI

where txn_year =2019;
```

```sql
select TXN_YEAR AS TXN_YEAR,BANK,

SUM(AVG_BALANCE) AS TOT_AVG_BALANCE

from BANKING_KPI

GROUP BY 1,2

ORDER BY TOT_AVG_BALANCE DESC;
```

```sql
SELECT * FROM TRANSACTIONS

WHERE BANK = 'Sky Bank' AND ACCOUNT_ID = 7745

ORDER BY DATE ,BANK;
```

```sql
SELECT * FROM TRANSACTIONS

WHERE ACCOUNT_ID = 1 AND YEAR(DATE) = 2019 AND MONTH(DATE) = 7;
```

```sql
SELECT * FROM ACC_LATEST_TXNS_WITH_BALANCE ;
```

```sql
SELECT DISTINCT STATUS,SUM(AMOUNT)

FROM LOAN
```

```
GROUP BY 1

ORDER BY 1;



SELECT * from disposition WHERE disp_id  in (SELECT disp_id FROM disposition WHERE account_id BETWEEN 3 AND 6);



CREATE OR REPLACE TABLE loan_joins AS

SELECT d.disp_id AS disp_id, d.type AS type, NT.* FROM disposition AS d JOIN

(

SELECT a.account_id AS account_id , a.account_type AS account_type, a.card_assigned AS card_assigned, l.loan_id AS loan_id, l.amount AS amount,

l.duration AS duration, l.status AS STATUS, l.Date AS loan_date

FROM account AS a JOIN loan AS l USING(account_id) ORDER BY 1) AS NT

USING(account_id);


SELECT * FROM loan_joins
```