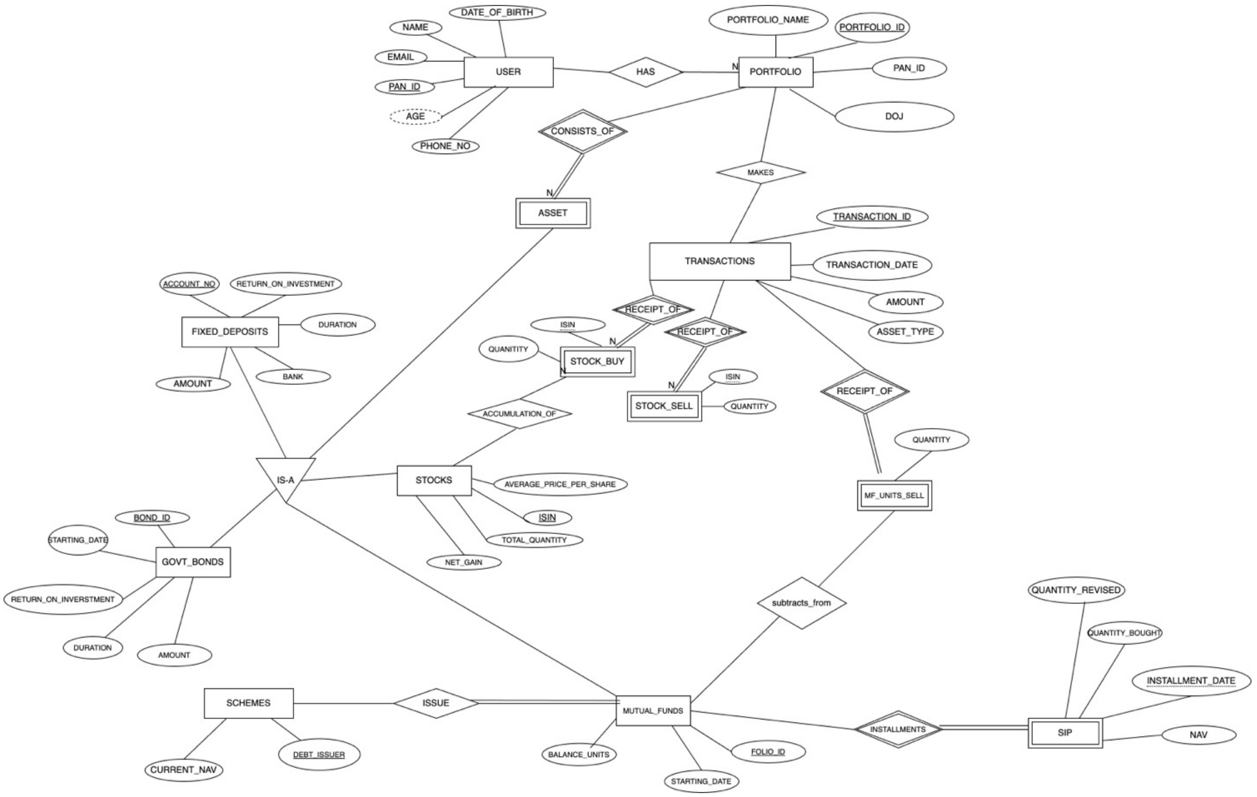
DBMS PROJECT
FINANCIAL PORTFOLIO MANAGEMENT SYSTEM

ARYAN PRASHANT PATIL - 22CSB0F35

SHRUTI VANKA - 22CSB0F33



Extended Entity Relationship Model

TABLE DESCRIPTIONS – ER TO RELATIONAL MODEL

NOTE: THE FUNCTIONAL DEPENDENCIES MENTIONED DO NOT INVOLVE THE PRIMARY KEY DEPENDENCIES AS THOSE DEPENDENCIES ARE COVERED IN THE DEFINITION OF PRIMARY KEY.

ENTITIES:

1. **USER**

   ATTRIBUTES:
   - PAN_ID
   - NAME
   - DOB: **AGE DERIVED FROM DOB (EXCLUDED FROM RELATIONAL SCHEMA)**
   - ADDRESS
   - EMAIL
   - PHONE_NO

   PRIMARY KEY: PAN_ID
2. **PORTFOLIO**

   ATTRIBUTES:
   - PORTFOLIO_ID
   - PAN_ID
   - PORTFOLIO_NAME
   - DATE_OF_JOINING
   - NUMBER_OF_INVESTMENTS

   PRIMARY KEY: PORTFOLIO_ID
3. **TRANSACTIONS**

   ATTRIBUTES:
   - TRANSACTION_ID
   - PORTFOLIO_ID
   - AMOUNT
   - ASSET_TYPE
   - TRANSACTION_DATE

   PRIMARY KEY: TRANSACTION_ID
4. **STOCK_BUY (WEAK)**

   ATTRIBUTES:
   - TRANSACTION_ID
   - ISIN
   - QUANTITY
   - DUMMY_QUANITITY

   PRIMARY KEY: TRANSACTION_ID, ISIN
5. **STOCK_SELL (WEAK)**

   ATTRIBUTES:
   - T_ID
   - ISIN
   - QUANTITY

   PRIMARY KEY: TRANSACTION_ID, ISIN
6. **STOCKS**

ATTRIBUTES:

- ISIN
- TOTAL_QUANTITY
- AVERAGE_PRICE_PER_SHARE
- NET_GAIN
- PORTFOLIO_ID

PRIMARY KEY: PORTFOLIO_ID, ISIN

7. **STOCK_INFO**

ATTRIBUTES:

- ISIN
- NAME
- CURRENT_VALUE
- P/E RATIO

PRIMARY KEY: ISIN

8. **SCHEMES**

ATTRIBUTES:

- CURRENT_NAV
- DEBT_ISSUER

PRIMARY KEY: DEBT_ISSUER

9. **MUTUAL_FUNDS**

NAV STANDS FOR NET ASSET VALUE

ATTRIBUTES:

- FOLIO_ID
- PORTFOLIO_ID
- BALANCE_UNITS
- STARTING_DATE
- DEBT_ISSUER

PRIMARY KEY: FOLIO_ID,PORTFOLIO_ID

10. **SIP**

WEAK ENTITY WITH RELATIONSHIP DERIVED FROM MUTUAL FUNDS

ATTRIBUTES:

- QUANTITY_BOUGHT
- QUANTITY_REVISED
- NAV
- PORTFOLIO_ID
- FOLIO_ID
- INSTALLMENT_DATE

PRIMARY KEY:INSTALLMENT_DATE,FOLIO_ID,PORTFOLIO_ID

11. **MF_UNITS_SELL**

ATTRIBUTES:

- QUANTITY
- TRANSACTION_ID
- FOLIO_ID
- PORTFOLIO_ID

PRIMARY KEY: TRANSACTION_ID,PORTFOLIO_ID,FOLIO_ID

12. **FD**

    ATTRIBUTES:

    - STARTING_DATE
    - ACC_NO
    - DURATION
    - BANK
    - AMOUNT
    - RETURN_ON_INVESTMENT
    - PORTFOLIO_ID

    PRIMARY KEY: PORTFOLIO_ID, ACC_NO

13. **GOVT_BONDS**

    ATTRIBUTES:

    - BOND_ID
    - STARTING_DATE
    - DURATION
    - RETURN_ON_INVESTMENT
    - PORTFOLIO_ID
    - AMOUNT

    PRIMARY KEY: BOND_ID, PORTFOLIO_ID

Normalisation:

USER:

Functional Dependencies:

- PAN_ID -> NAME
- PAN_ID -> DOB
- PAN_ID -> ADDRESS
- PAN_ID -> EMAIL
- PAN_ID -> PHONE_NO

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as PAN_ID is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because PAN_ID is the primary key.

USER is in BCNF and 3 NF.

PORTFOLIO:

Functional Dependencies:

- PORTFOLIO_ID -> PORTFOLIO_NAME
- PORTFOLIO_ID -> DATE_OF_JOINING
- PORTFOLIO_ID -> NUMBER_OF_INVESTMENTS
- PORTFOLIO_ID -> PAN_ID

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as PORTFOLIO_ID is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because PORTFOLIO_ID is the primary key.

PORTFOLIO is in BCNF and 3 NF.

TRANSACTIONS:

Functional Dependencies:

- TRANSACTION_ID -> AMOUNT
- TRANSACTION_ID -> ASSET_TYPE
- TRANSACTION_ID -> PORTFOLIO_ID
- TRANSACTION_ID -> TRANSACTION_DATE

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as TRANSACTION_ID is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because TRANSACTION_ID is the primary key.

TRANSACTIONS is in BCNF and 3 NF.

STOCK_BUY:

- TRANSACTION_ID, ISIN -> QUANTITY

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as TRANSACTION_ID, ISIN is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because TRANSACTION_ID, ISIN is the primary key.

STOCK_BUY is in BCNF and 3 NF.

STOCK_SELL:

- TRANSACTION_ID, ISIN -> QUANTITY

1 NF - all values are atomic, so it is in 1 NF

2 NF - there are no partial dependencies so it is in 2 NF as TRANSACTION_ID, ISIN is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because TRANSACTION_ID, ISIN is the primary key.

STOCK_SELL is in BCNF and 3 NF.

STOCKS:

Functional Dependencies:

- ISIN,PORTFOLIO_ID -> TOTAL_QUANTITY
- ISIN,PORTFOLIO_ID -> AVERAGE_PRICE_PER_SHARE
- ISIN,PORTFOLIO_ID -> NET_GAIN

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as ISIN,PORTFOLIO_ID is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because ISIN,PORTFOLIO_ID is the primary key.

STOCKS is in BCNF and 3 NF.

STOCK_INFO:

Functional Dependencies:

- ISIN -> NAME
- ISIN -> CURRENT_VALUE
- ISIN -> P/E RATIO

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as ISIN is a primary key, and all functional dependencies are primary key -> non-prime form

3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because ISIN is the primary key.

STOCK_INFO is in BCNF and 3 NF.

SCHEMES:

Functional Dependencies:

- DEBT_ISSUER->CURRENT_NAV

  1 NF - all values are atomic, so it is in 1 NF
  2 NF - there are no partial dependencies so it is in 2 NF as DEBT_ISSUER is a primary key, and all functional dependencies are primary key -> non-prime form
  3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
  BCNF - left hand side of each functional dependency is the primary key so BCNF as well because DEBT_ISSUER is the primary key.

SCHEMES is in BCNF and 3 NF.

MUTUAL_FUNDS:

Functional Dependencies:

- FOLIO_ID->DEBT_ISSUER
- FOLIO_ID->BALANCE_UNITS
- FOLIO_ID->STARTING_DATE
- FOLIO_ID->PORTFOLIO_ID

  1 NF - all values are atomic, so it is in 1 NF
  2 NF - there are no partial dependencies so it is in 2 NF as FOLIO_ID is a primary key, and all functional dependencies are primary key -> non-prime form
  3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.

MUTUAL_FUNDS is in 3 NF.

MF_UNITS_SELL:

Functional Dependencies:

- TRANSACTION_ID->QUANTITY
- TRANSACTION_ID->PORTFOLIO_ID

- FOLIO_ID->PORTFOLIO_ID
- TRANSACTION_ID->FOLIO_ID

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as TRANSACTION_ID, PORTFOLIO_ID,FOLIO_ID is a primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
MF_UNITS_SELL is in 3 NF.

SIP:
SIP is a weak entity deriving its primary key from the strong entity MUTUAL_FUNDS. It has INSTALLMENT_DATE as the discriminator which combines with FOLIO_ID(parent table primary key) to give the primary key.
Functional Dependencies:

- INSTALLMENT_DATE,FOLIO_ID->NAV
- INSTALLMENT_DATE,FOLIO_ID->QUANTITY_BOUGHT
- INSTALLMENT_DATE,FOLIO_ID->QUANYITY_REVISED
- INSTALLMENT_DATE,FOLIO_ID->PORTFOLIO_ID

1 NF - all values are atomic, so it is in 1 NF
2 NF - there are no partial dependencies so it is in 2 NF as INSTALLMENT_DATE and FOLIO_ID forms the primary key, and all functional dependencies are primary key -> non-prime form
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.
BCNF - left hand side of each functional dependency is the primary key so BCNF as well because INSTALLMENT_DATE and FOLIO_ID forms the is the primary key.
SIP is in BCNF and 3 NF.

FIXED_DEPOSITS:

Functional Dependencies:

- ACC_NO,PORTFOLIO_ID -> STARTING_DATE
- ACC_NO,PORTFOLIO_ID -> AMOUNT
- ACC_NO,PORTFOLIO_ID -> RETURN_ON_INVESTMENT
- ACC_NO,PORTFOLIO_ID -> DURATION
- ACC_NO,PORTFOLIO_ID -> BANK

1 NF - all values are atomic, so it is in 1 NF

2 NF - there are no partial dependencies so it is in 2 NF as ACC_NO,PORTFOLIO_ID is a primary key, and all functional dependencies are primary key -> non-prime form

3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.

BCNF - left hand side of each functional dependency is the primary key so BCNF as well because ACC_NO,PORTFOLIO_ID is the primary key.

FIXED_DEPOSITS is in BCNF and 3 NF.

GOVT_BONDS:

Functional Dependencies:

- BOND_ID,PORTFOLIO_ID -> STARTING_DATE
- BOND_ID,PORTFOLIO_ID -> AMOUNT
- BOND_ID,PORTFOLIO_ID -> RETURN_ON_INVESTMENT
- BOND_ID,PORTFOLIO_ID -> DURATION
- BOND_ID PORTFOLIO_ID -> BANK

1 NF - all values are atomic, so it is in 1 NF

2 NF - there are no partial dependencies so it is in 2 NF as BOND_ID,PORTFOLIO_ID is a primary key, and all functional dependencies are primary key -> non-prime form

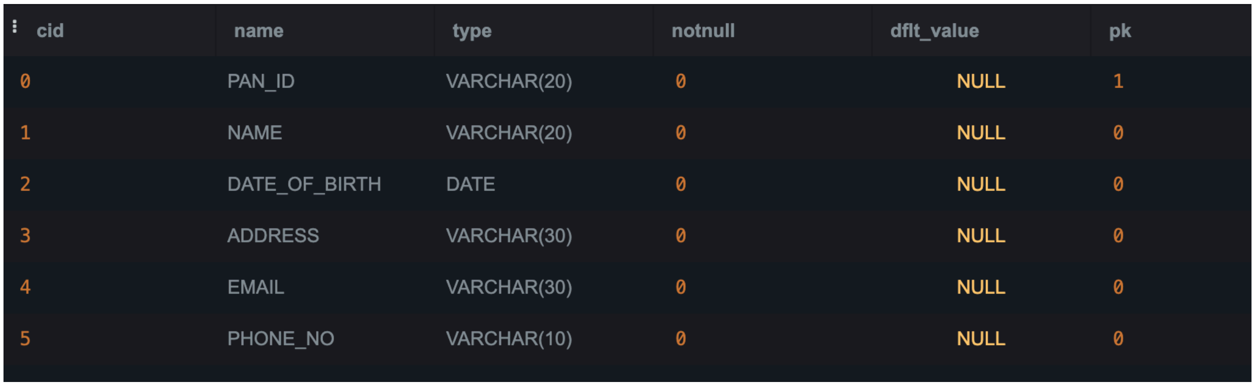
3 NF - there are no transitive dependencies so it is in 3 NF as well as there are no prime -> non - prime dependencies.

BCNF - left hand side of each functional dependency is the primary key so BCNF as well because BOND_ID,PORTFOLIO_ID is the primary key.

GOVT_BONDS is in BCNF and 3 NF.

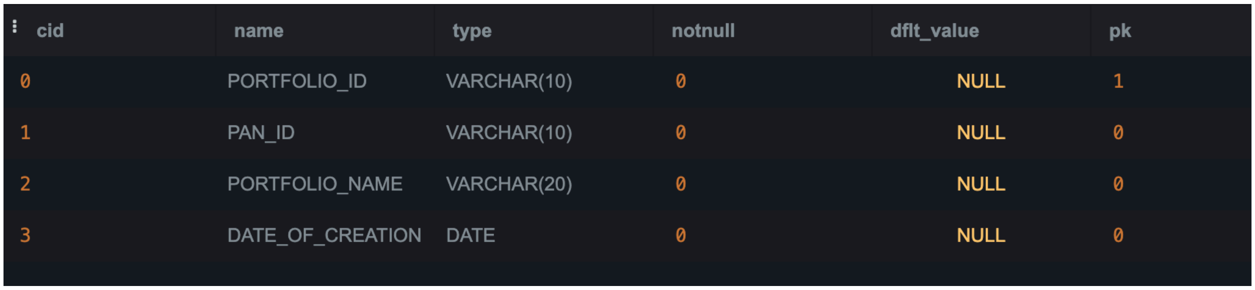
Table creations:

```
--USER:

CREATE TABLE USER(
PAN_ID VARCHAR(20),
NAME VARCHAR(20),
DATE_OF_BIRTH DATE,
ADDRESS VARCHAR(30),
EMAIL VARCHAR(30),
PHONE_NO VARCHAR(10),
PRIMARY KEY(PAN_ID)
);

pragma table_info('user');
```



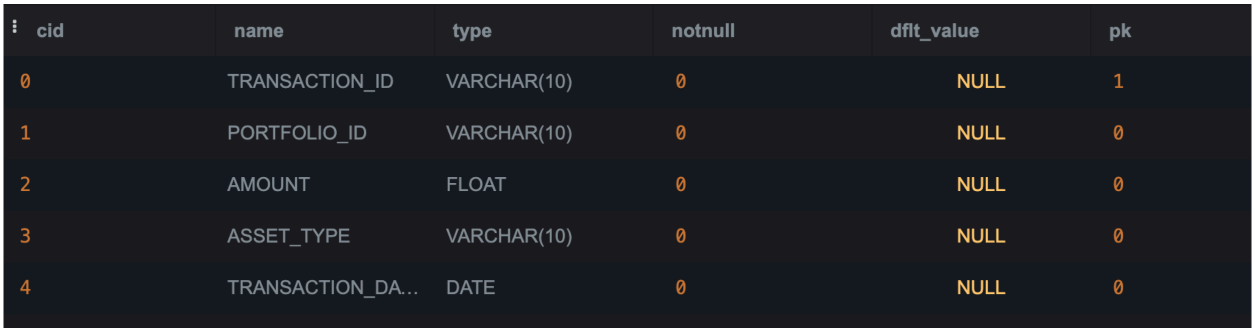
cid	name	type	notnull	dflt_value	pk
0	PAN_ID	VARCHAR(20)	0	NULL	1
1	NAME	VARCHAR(20)	0	NULL	0
2	DATE_OF_BIRTH	DATE	0	NULL	0
3	ADDRESS	VARCHAR(30)	0	NULL	0
4	EMAIL	VARCHAR(30)	0	NULL	0
5	PHONE_NO	VARCHAR(10)	0	NULL	0

```
--PORTFOLIO:

CREATE TABLE PORTFOLIO(
PORTFOLIO_ID VARCHAR(10),
PAN_ID VARCHAR(10),
PORTFOLIO_NAME VARCHAR(20),
DATE_OF_CREATION DATE,
PRIMARY KEY(PORTFOLIO_ID),
FOREIGN KEY (PAN_ID) REFERENCES USER(PAN_ID)
);

pragma table_info(‘portfolio’);
```



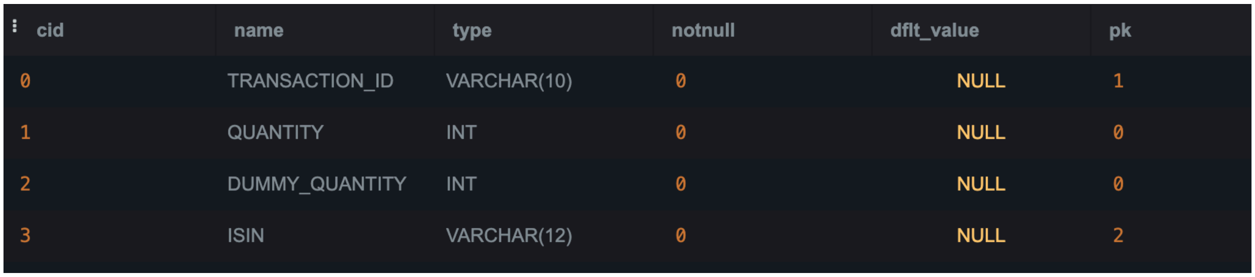
cid	name	type	notnull	dflt_value	pk
0	PORTFOLIO_ID	VARCHAR(10)	0	NULL	1
1	PAN_ID	VARCHAR(10)	0	NULL	0
2	PORTFOLIO_NAME	VARCHAR(20)	0	NULL	0
3	DATE_OF_CREATION	DATE	0	NULL	0

```
--TRANSACTIONS:

CREATE TABLE TRANSACTIONS(
TRANSACTION_ID VARCHAR(10),
PORTFOLIO_ID VARCHAR(10),
AMOUNT FLOAT,
ASSET_TYPE VARCHAR(10),
TRANSACTION_DATE DATE,
PRIMARY KEY (TRANSACTION_ID),
FOREIGN KEY (PORTFOLIO_ID) REFERENCES PORTFOLIO(PORTFOLIO_ID)
);

pragma table_info('transactions');
```



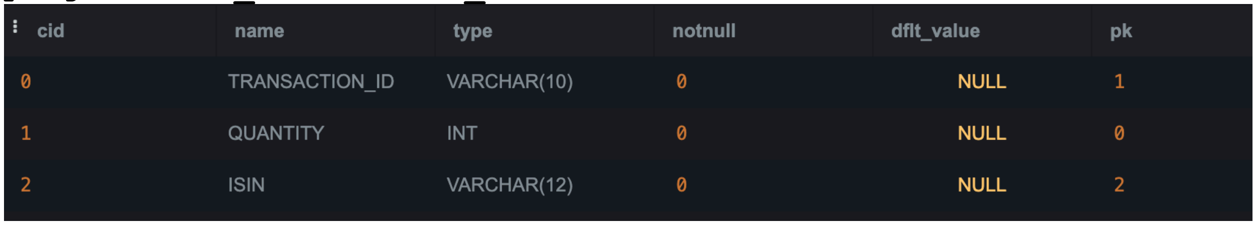
cid	name	type	notnull	dflt_value	pk
0	TRANSACTION_ID	VARCHAR(10)	0	NULL	1
1	PORTFOLIO_ID	VARCHAR(10)	0	NULL	0
2	AMOUNT	FLOAT	0	NULL	0
3	ASSET_TYPE	VARCHAR(10)	0	NULL	0
4	TRANSACTION_DA...	DATE	0	NULL	0

```
--STOCK_BUY:

CREATE TABLE STOCK_BUY(
TRANSACTION_ID VARCHAR(10),
QUANTITY INT,
DUMMY_QUANTITY INT,
ISIN VARCHAR(12),
PRIMARY KEY(TRANSACTION_ID, ISIN),
FOREIGN KEY (TRANSACTION_ID) REFERENCES TRANSACTIONS
(TRANSACTION_ID)
);

pragma table_info('stock_buy');
```



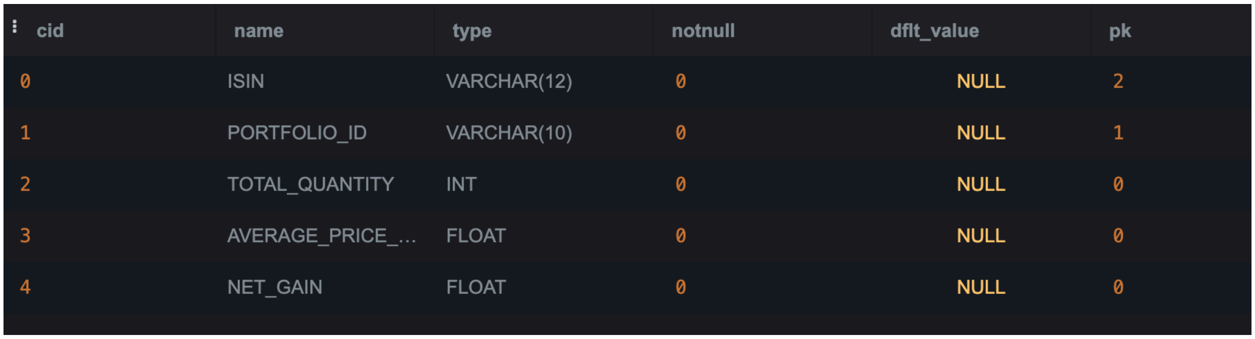
cid	name	type	notnull	dflt_value	pk
0	TRANSACTION_ID	VARCHAR(10)	0	NULL	1
1	QUANTITY	INT	0	NULL	0
2	DUMMY_QUANTITY	INT	0	NULL	0
3	ISIN	VARCHAR(12)	0	NULL	2

```
--STOCK_SELL:

CREATE TABLE STOCK_SELL(
TRANSACTION_ID VARCHAR(10),
QUANTITY INT,
ISIN VARCHAR(12),
PRIMARY KEY(TRANSACTION_ID, ISIN),
FOREIGN KEY (TRANSACTION_ID) REFERENCES TRANSACTIONS
(TRANSACTION_ID)
);

pragma table_info('stock_sell');
```



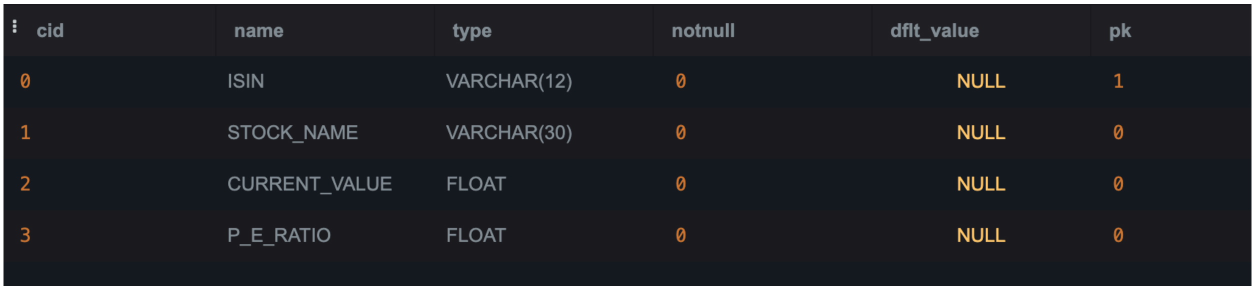
cid	name	type	notnull	dflt_value	pk
0	TRANSACTION_ID	VARCHAR(10)	0	NULL	1
1	QUANTITY	INT	0	NULL	0
2	ISIN	VARCHAR(12)	0	NULL	2

```
--STOCKS:

CREATE TABLE STOCKS(
ISIN VARCHAR(12),
PORTFOLIO_ID VARCHAR(10),
TOTAL_QUANTITY INT,
AVERAGE_PRICE_PER_SHARE FLOAT,
NET_GAIN FLOAT,
PRIMARY KEY(PORTFOLIO_ID, ISIN),
FOREIGN KEY (PORTFOLIO_ID) REFERENCES PORTFOLIO(PORTFOLIO_ID)
);

pragma table_info('stocks');
```



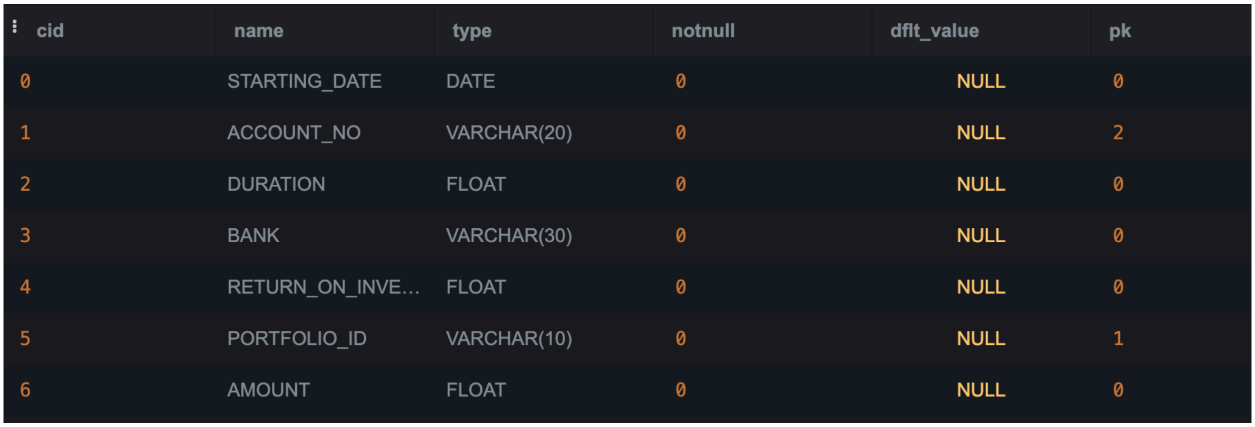
cid	name	type	notnull	dflt_value	pk
0	ISIN	VARCHAR(12)	0	NULL	2
1	PORTFOLIO_ID	VARCHAR(10)	0	NULL	1
2	TOTAL_QUANTITY	INT	0	NULL	0
3	AVERAGE_PRICE_…	FLOAT	0	NULL	0
4	NET_GAIN	FLOAT	0	NULL	0

```
--STOCK_INFO:

CREATE TABLE STOCK_INFO(
ISIN VARCHAR(12),
STOCK_NAME VARCHAR(30),
CURRENT_VALUE FLOAT,
P_E_RATIO FLOAT,
PRIMARY KEY(ISIN)
);
```



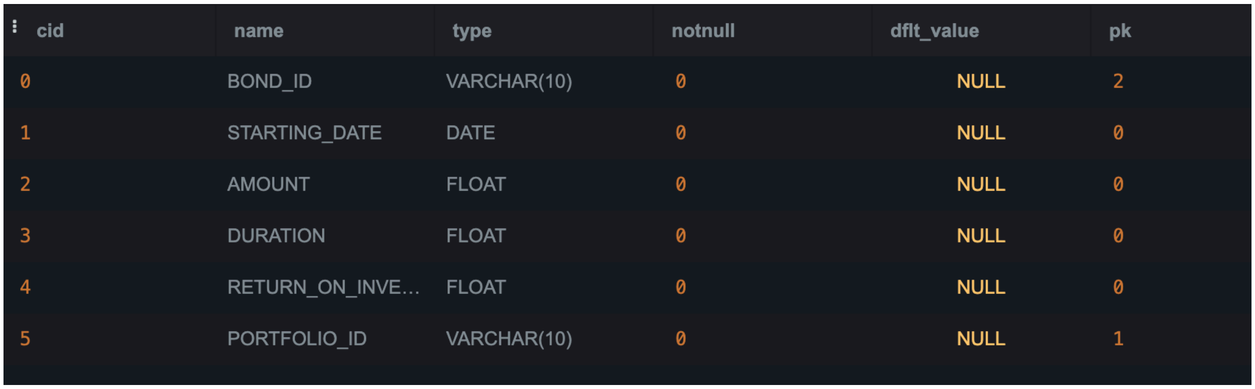
cid	name	type	notnull	dflt_value	pk
0	ISIN	VARCHAR(12)	0	NULL	1
1	STOCK_NAME	VARCHAR(30)	0	NULL	0
2	CURRENT_VALUE	FLOAT	0	NULL	0
3	P_E_RATIO	FLOAT	0	NULL	0

```
--FIXED_DEPOSITS:

CREATE TABLE FIXED_DEPOSITS(
STARTING_DATE DATE,
ACCOUNT_NO VARCHAR(20),
DURATION FLOAT,
BANK VARCHAR(30),
RETURN_ON_INVESTMENT FLOAT,
PORTFOLIO_ID VARCHAR(10),
AMOUNT FLOAT,
PRIMARY KEY (PORTFOLIO_ID, ACCOUNT_NO),
FOREIGN KEY (PORTFOLIO_ID) REFERENCES PORTFOLIO (PORTFOLIO_ID)
);


pragma table_info('fixed_deposits');
```



cid	name	type	notnull	dflt_value	pk
0	STARTING_DATE	DATE	0	NULL	0
1	ACCOUNT_NO	VARCHAR(20)	0	NULL	2
2	DURATION	FLOAT	0	NULL	0
3	BANK	VARCHAR(30)	0	NULL	0
4	RETURN_ON_INVE...	FLOAT	0	NULL	0
5	PORTFOLIO_ID	VARCHAR(10)	0	NULL	1
6	AMOUNT	FLOAT	0	NULL	0

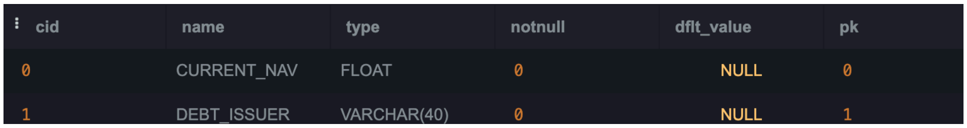
```
--GOVT_BONDS:

CREATE TABLE GOVT_BONDS(
BOND_ID VARCHAR(10),
STARTING_DATE DATE,
AMOUNT FLOAT,
DURATION FLOAT,
RETURN_ON_INVESTMENT FLOAT,
PORTFOLIO_ID VARCHAR(10),
PRIMARY KEY (PORTFOLIO_ID, BOND_ID),
FOREIGN KEY (PORTFOLIO_ID) REFERENCES PORTFOLIO (PORTFOLIO_ID)
);

pragma table_info('govt_bonds');
```



cid	name	type	notnull	dflt_value	pk
0	BOND_ID	VARCHAR(10)	0	NULL	2
1	STARTING_DATE	DATE	0	NULL	0
2	AMOUNT	FLOAT	0	NULL	0
3	DURATION	FLOAT	0	NULL	0
4	RETURN_ON_INVE...	FLOAT	0	NULL	0
5	PORTFOLIO_ID	VARCHAR(10)	0	NULL	1

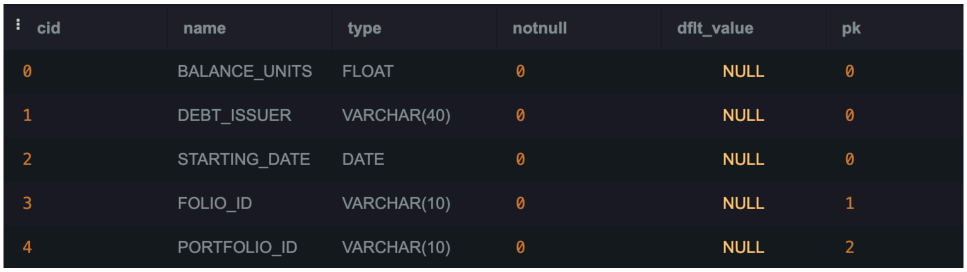
```
--SCHEMES
CREATE TABLE SCHEMES(
CURRENT_NAV FLOAT,
DEBT_ISSUER VARCHAR(40),
PRIMARY KEY(DEBT_ISSUER)
);
```

```
pragma table_info('schemes');
```



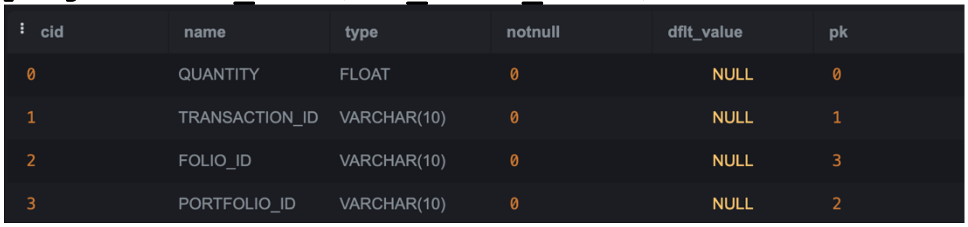
cid	name	type	notnull	dflt_value	pk
0	CURRENT_NAV	FLOAT	0	NULL	0
1	DEBT_ISSUER	VARCHAR(40)	0	NULL	1

```
--MUTUAL_FUNDS
CREATE TABLE MUTUAL_FUNDS(
BALANCE_UNITS FLOAT,
DEBT_ISSUER VARCHAR(40),
STARTING_DATE DATE,
FOLIO_ID VARCHAR(10),
PORTFOLIO_ID VARCHAR(10),
FOREIGN KEY (PORTFOLIO_ID) REFERENCES PORTFOLIO(PORTFOLIO_ID),
FOREIGN KEY(DEBT_ISSUER) REFERENCES SCHEMES(DEBT_ISSUER),
PRIMARY KEY(FOLIO_ID,PORTFOLIO_ID)
);
pragma table_info('mutual_funds');
```



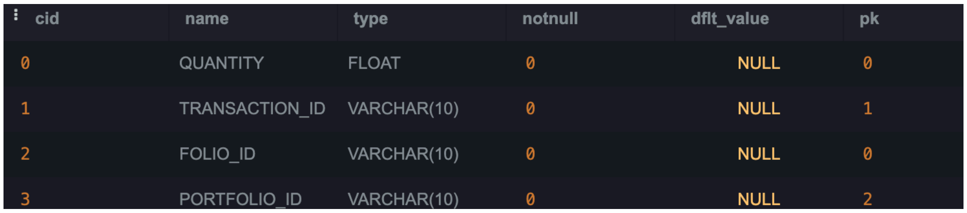
cid	name	type	notnull	dflt_value	pk
0	BALANCE_UNITS	FLOAT	0	NULL	0
1	DEBT_ISSUER	VARCHAR(40)	0	NULL	0
2	STARTING_DATE	DATE	0	NULL	0
3	FOLIO_ID	VARCHAR(10)	0	NULL	1
4	PORTFOLIO_ID	VARCHAR(10)	0	NULL	2

```
--MF_UNITS_SELL
CREATE TABLE MF_UNITS_SELL    (
QUANTITY FLOAT,
TRANSACTION_ID VARCHAR(10),
FOLIO_ID VARCHAR(10),
  PORTFOLIO_ID VARCHAR(10),
FOREIGN KEY (PORTFOLIO_ID) REFERENCES PORTFOLIO(PORTFOLIO_ID),
PRIMARY KEY(TRANSACTION_ID,PORTFOLIO_ID,FOLIO_ID)
);
pragma table_info('mf_units_sell');
```



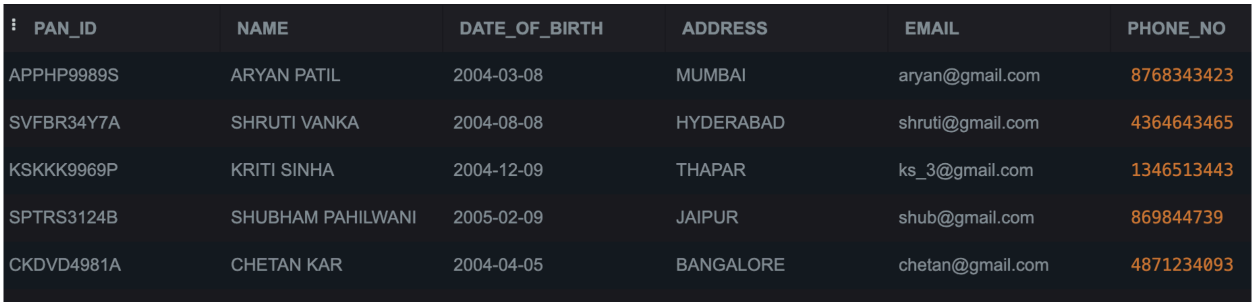
cid	name	type	notnull	dflt_value	pk
0	QUANTITY	FLOAT	0	NULL	0
1	TRANSACTION_ID	VARCHAR(10)	0	NULL	1
2	FOLIO_ID	VARCHAR(10)	0	NULL	3
3	PORTFOLIO_ID	VARCHAR(10)	0	NULL	2

```
--SIP
CREATE TABLE SIP(
INSTALLMENT_DATE DATE,
QUANTITY_BOUGHT FLOAT,
     QUANTITY_REVISED FLOAT,
NAV FLOAT,
FOLIO_ID VARCHAR(10),
  PORTFOLIO_ID VARCHAR(10),
FOREIGN KEY(FOLIO_ID,PORTFOLIO_ID) REFERENCES
MUTUAL_FUNDS(FOLIO_ID,PORTFOLIO_ID),

PRIMARY KEY(FOLIO_ID,PORTFOLIO_ID,INSTALLMENT_DATE)
);
pragma table_info('sip');
```



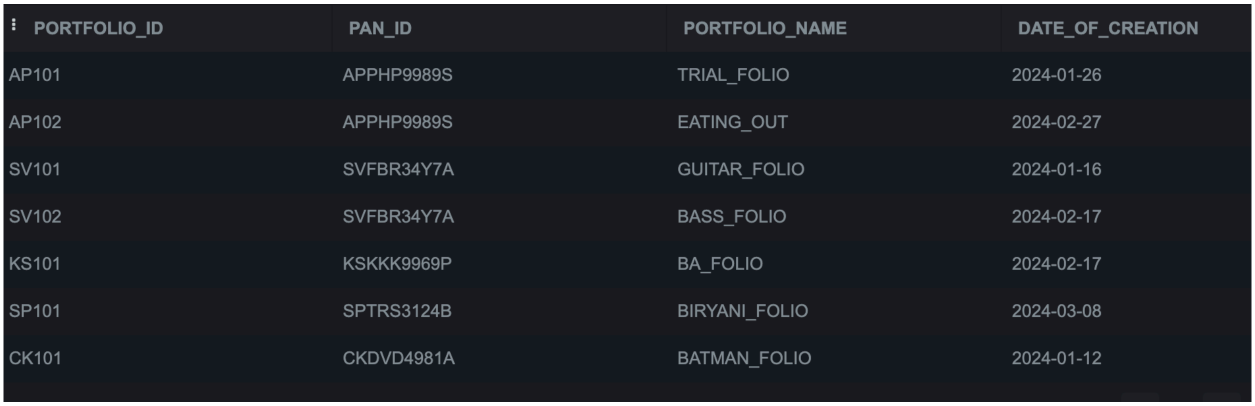
cid	name	type	notnull	dflt_value	pk
0	QUANTITY	FLOAT	0	NULL	0
1	TRANSACTION_ID	VARCHAR(10)	0	NULL	1
2	FOLIO_ID	VARCHAR(10)	0	NULL	0
3	PORTFOLIO_ID	VARCHAR(10)	0	NULL	2

```
--SQL QUERIES

--ALL USERS HERE
insert into USER VALUES('APPHP9989S','ARYAN PATIL','2004-03-08','MUMBAI','aryan@gmail.com',8768343423);
insert into USER VALUES('SVFBR34Y7A','SHRUTI VANKA','2004-08-08','HYDERABAD','shruti@gmail.com',4364643465);
insert into USER VALUES('KSKKK9969P','KRITI SINHA','2004-12-09','THAPAR','ks_3@gmail.com',1346513443);
insert into USER VALUES('SPTRS3124B','SHUBHAM PAHILWANI','2005-02-09','JAIPUR','shub@gmail.com',869844739);
insert into USER VALUES('CKDVD4981A','CHETAN KAR','2004-04-05','BANGALORE','chetan@gmail.com',4871234093);

SELECT * FROM USER;
```



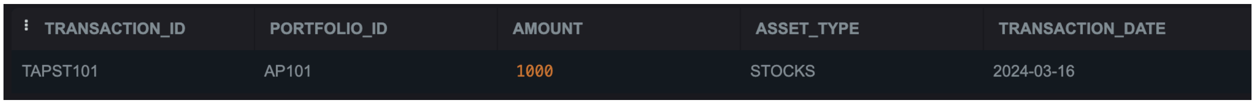
PAN_ID	NAME	DATE_OF_BIRTH	ADDRESS	EMAIL	PHONE_NO
APPHP9989S	ARYAN PATIL	2004-03-08	MUMBAI	aryan@gmail.com	8768343423
SVFBR34Y7A	SHRUTI VANKA	2004-08-08	HYDERABAD	shruti@gmail.com	4364643465
KSKKK9969P	KRITI SINHA	2004-12-09	THAPAR	ks_3@gmail.com	1346513443
SPTRS3124B	SHUBHAM PAHILWANI	2005-02-09	JAIPUR	shub@gmail.com	869844739
CKDVD4981A	CHETAN KAR	2004-04-05	BANGALORE	chetan@gmail.com	4871234093

```
--ALL PORTFOLIOS HERE
insert into PORTFOLIO
VALUES('AP101','APPHP9989S','TRIAL_FOLIO','2024-01-26');
insert into PORTFOLIO
VALUES('AP102','APPHP9989S','EATING_OUT','2024-02-27');
insert into PORTFOLIO
VALUES('SV101','SVFBR34Y7A','GUITAR_FOLIO','2024-01-16');
insert into PORTFOLIO
VALUES('SV102','SVFBR34Y7A','BASS_FOLIO','2024-02-17');
INSERT INTO PORTFOLIO
VALUES('KS101','KSKKK9969P','BA_FOLIO','2024-02-17');
insert into PORTFOLIO
VALUES('SP101','SPTRS3124B','BIRYANI_FOLIO','2024-03-08');
insert into PORTFOLIO
VALUES('CK101','CKDVD4981A','BATMAN_FOLIO','2024-01-12');

SELECT * FROM PORTFOLIO;
```



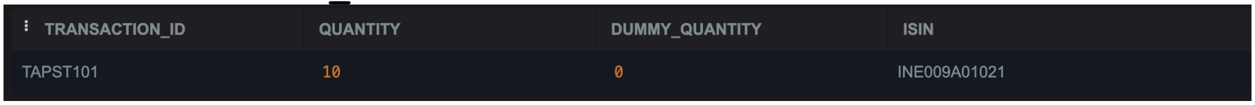
PORTFOLIO_ID	PAN_ID	PORTFOLIO_NAME	DATE_OF_CREATION
AP101	APPHP9989S	TRIAL_FOLIO	2024-01-26
AP102	APPHP9989S	EATING_OUT	2024-02-27
SV101	SVFBR34Y7A	GUITAR_FOLIO	2024-01-16
SV102	SVFBR34Y7A	BASS_FOLIO	2024-02-17
KS101	KSKKK9969P	BA_FOLIO	2024-02-17
SP101	SPTRS3124B	BIRYANI_FOLIO	2024-03-08
CK101	CKDVD4981A	BATMAN_FOLIO	2024-01-12

```
--STEP-WISE FOR EVERY OPERATION NOW

-- BUYING 10 STOCKS OF INFOSYS FOR THE 1ST TIME
INSERT INTO TRANSACTIONS
VALUES('TAPST101','AP101',1000,'STOCKS','2024-03-16');
```



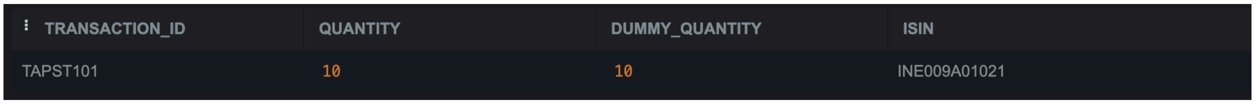
TRANSACTION_ID	PORTFOLIO_ID	AMOUNT	ASSET_TYPE	TRANSACTION_DATE
TAPST101	AP101	1000	STOCKS	2024-03-16

```
INSERT INTO STOCK_BUY VALUES('TAPST101',10,0,'INE009A01021');
```



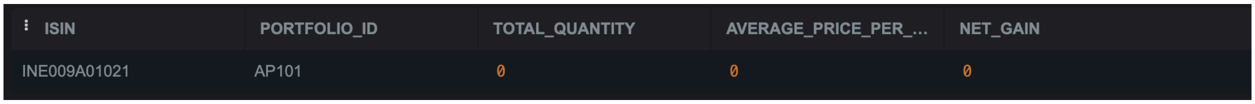
TRANSACTION_ID	QUANTITY	DUMMY_QUANTITY	ISIN
TAPST101	10	0	INE009A01021

```
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TAPST101';
```



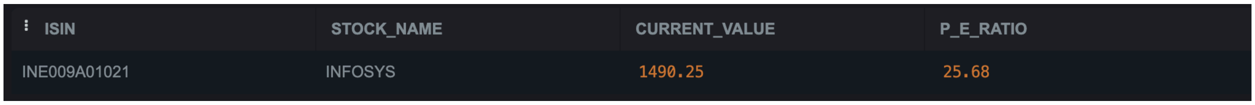
TRANSACTION_ID	QUANTITY	DUMMY_QUANTITY	ISIN
TAPST101	10	10	INE009A01021

```
INSERT INTO STOCKS VALUES('INE009A01021','AP101',0,0,0);
```



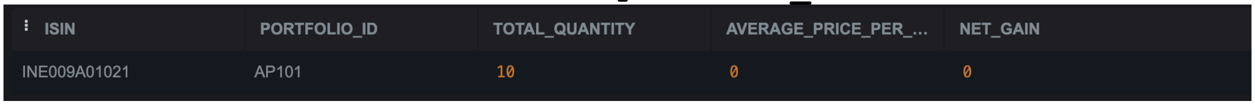
ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	0	0	0

```
INSERT INTO STOCK_INFO
VALUES('INE009A01021','INFOSYS',1490.25,25.68);
```



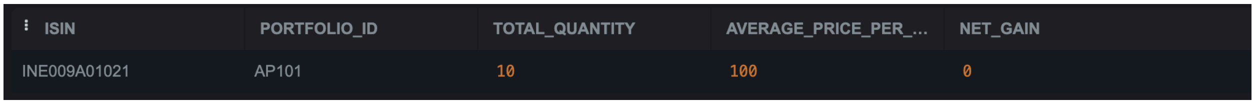
ISIN	STOCK_NAME	CURRENT_VALUE	P_E_RATIO
INE009A01021	INFOSYS	1490.25	25.68

```
UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101';
```



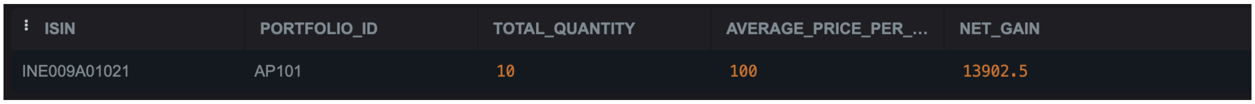
ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	10	0	0

```
UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
) WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101';
```



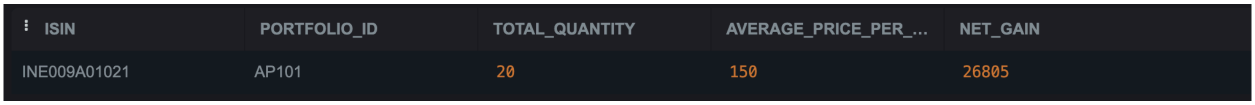
ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	10	100	0

```
UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE009A01021' AND PORTFOLIO_ID = 'AP101')
WHERE STOCKS.ISIN = 'INE009A01021' AND PORTFOLIO_ID = 'AP101';
```



ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	10	100	13902.5

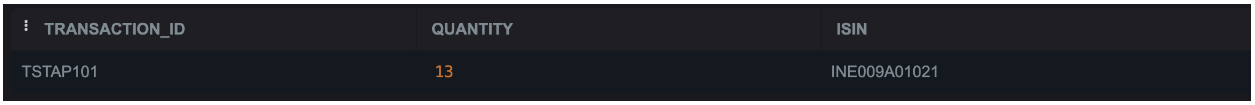
```
-- BUYING ANOTHER 10 STOCKS OF INFOSYS AT A DIFFERENT PRICE
INSERT INTO TRANSACTIONS
VALUES('TAPST102','AP101',2000,'STOCKS','2024-03-17');

INSERT INTO STOCK_BUY VALUES('TAPST102',10,0,'INE009A01021');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TAPST102';

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
) WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE009A01021' AND PORTFOLIO_ID = 'AP101')
WHERE STOCKS.ISIN = 'INE009A01021' AND PORTFOLIO_ID = 'AP101';
```



ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	20	150	26805

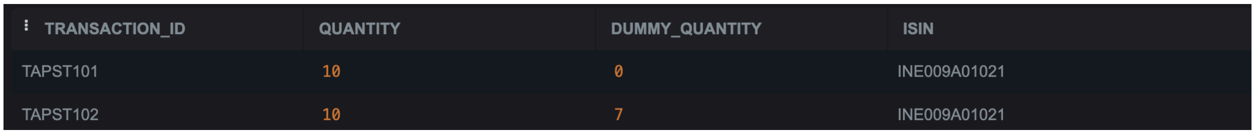
```
-- SELLING 13 STOCKS OF INFOSYS USING FIFO MECHANISM

INSERT INTO TRANSACTIONS VALUES('TSTAP101','AP101',-
13000,'STOCKS','2024-03-17');
```

```
INSERT INTO STOCK_SELL VALUES('TSTAP101',13,'INE009A01021');
```



TRANSACTION_ID	QUANTITY	ISIN
TSTAP101	13	INE009A01021

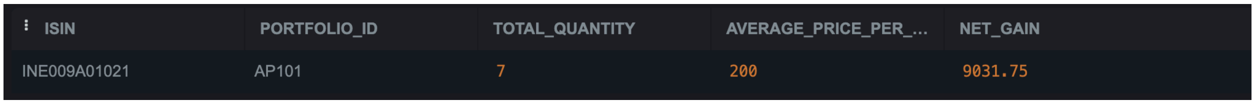
```
UPDATE STOCK_BUY SET dummy_quantity = 0 WHERE TRANSACTION_ID =
'TAPST101';
UPDATE STOCK_BUY SET dummy_quantity = 7 WHERE TRANSACTION_ID =
'TAPST102';
```



TRANSACTION_ID	QUANTITY	DUMMY_QUANTITY	ISIN
TAPST101	10	0	INE009A01021
TAPST102	10	7	INE009A01021

--As it can be seen changes made according to FIFO mechanism in stocks

```
UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101')
) WHERE ISIN = 'INE009A01021' AND portfolio_id = 'AP101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE009A01021' AND PORTFOLIO_ID = 'AP101')
WHERE STOCKS.ISIN = 'INE009A01021' AND PORTFOLIO_ID = 'AP101';
```

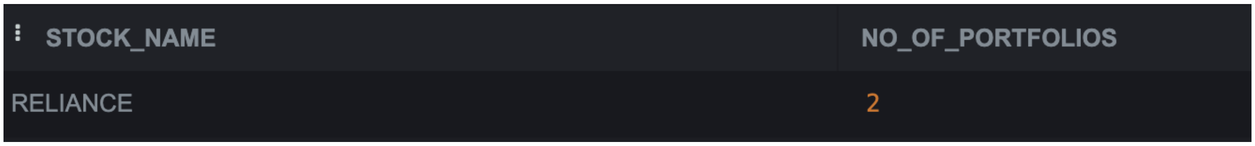


ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	7	200	9031.75

```
-- BUYING 10 STOCKS OF RELIANCE FOR ANOTHER USER

INSERT INTO TRANSACTIONS
VALUES('TSVST101','SV101',30000,'STOCKS','2024-02-03');

INSERT INTO STOCK_BUY VALUES('TSVST101',10,0,'INE002A01018');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TSVST101';

INSERT INTO STOCKS VALUES('INE002A01018','SV101',0,0,0);

INSERT INTO STOCK_INFO
VALUES('INE002A01018','RELIANCE',2983.65,39.19);


UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101')
) WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE002A01018' AND PORTFOLIO_ID = 'SV101')
WHERE STOCKS.ISIN = 'INE002A01018' AND PORTFOLIO_ID = 'SV101';
```

```
-- BUYING 5 MORE STOCKS OF RELIANCE FOR THIS USER

INSERT INTO TRANSACTIONS
VALUES('TSVST102','SV101',12000,'STOCKS','2024-02-04');

INSERT INTO STOCK_BUY VALUES('TSVST102',5,0,'INE002A01018');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TSVST102';

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101')
) WHERE ISIN = 'INE002A01018' AND portfolio_id = 'SV101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE002A01018' AND PORTFOLIO_ID = 'SV101')
WHERE STOCKS.ISIN = 'INE002A01018' AND PORTFOLIO_ID = 'SV101';


-- 1st user is buying a few stocks of reliance now

INSERT INTO TRANSACTIONS
VALUES('TAPST104','AP101',13000,'STOCKS','2024-02-07');

INSERT INTO STOCK_BUY VALUES('TAPST104',6,0,'INE002A01018');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TAPST104';
```

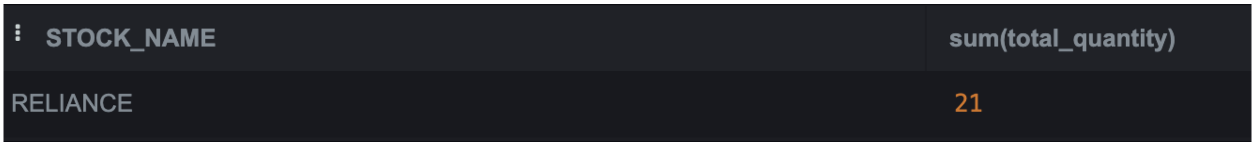
```
select * from stock_buy;

INSERT INTO STOCKS VALUES('INE002A01018','AP101',0,0,0);

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'AP101';

select * from stocks;

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'AP101')
WHERE ISIN = 'INE002A01018' AND portfolio_id = 'AP101')
) WHERE ISIN = 'INE002A01018' AND portfolio_id = 'AP101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE002A01018' AND PORTFOLIO_ID = 'AP101')
WHERE STOCKS.ISIN = 'INE002A01018' AND PORTFOLIO_ID = 'AP101';

-- lets get count of how many total portfolios have this stock
select STOCK_NAME, count(PORTFOLIO_ID) AS NO_OF_PORTFOLIOS from
STOCK_INFO JOIN
stocks ON STOCKS.ISIN = STOCK_INFO.ISIN group by STOCKS.isin
having STOCKS.ISIN = 'INE002A01018';
```



STOCK_NAME	NO_OF_PORTFOLIOS
RELIANCE	2

```
-- total stocks bought by multiple users for reliance
```

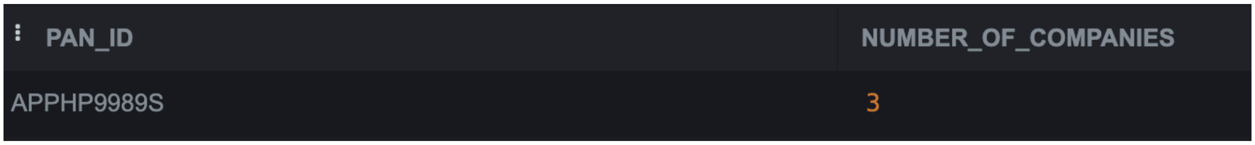
```
select STOCK_NAME, sum(total_quantity) from
STOCK_INFO JOIN
stocks ON STOCKS.ISIN = STOCK_INFO.ISIN group by STOCKS.isin
having STOCKS.isin = 'INE002A01018';
```



STOCK_NAME	sum(total_quantity)
RELIANCE	21

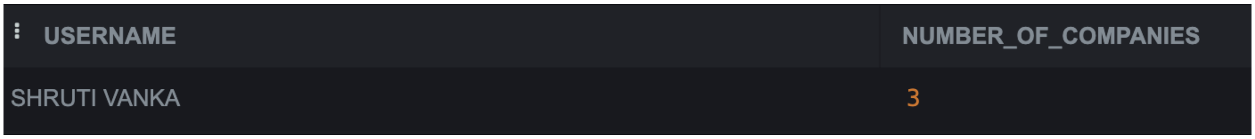
```
-- buying a few stocks of hdfc bank by 1st user but different
portfolio

INSERT INTO TRANSACTIONS
VALUES('TAPST105','AP102',12000,'STOCKS','2024-02-07');

insert into STOCK_BUY VALUES('TAPST105',8,0,'INE0040A01034');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TAPST105';

insert into stock_info values('INE0040A01034', 'HDFC
BANK',1445.70,23.92);

insert into STOCKS values('INE0040A01034','AP102',0,0,0);


UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE0040A01034' AND portfolio_id = 'AP102')
WHERE ISIN = 'INE0040A01034' AND portfolio_id = 'AP102';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE0040A01034' AND portfolio_id = 'AP102')
WHERE ISIN = 'INE0040A01034' AND portfolio_id = 'AP102')
) WHERE ISIN = 'INE0040A01034' AND portfolio_id = 'AP102';

UPDATE STOCKS
```

```
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE0040A01034' AND PORTFOLIO_ID = 'AP102')
WHERE STOCKS.ISIN = 'INE0040A01034' AND PORTFOLIO_ID = 'AP102';

SELECT * FROM STOCKS;

-- LETS GET TOTAL NUMBER OF STOCKS OWNED BY THE SAME USER IN ALL
PORTFOLIOS, PAN_ID WISE

SELECT PORTFOLIO.PAN_ID, COUNT(ISIN) AS NUMBER_OF_COMPANIES FROM
STOCKS JOIN PORTFOLIO ON STOCKS.PORTFOLIO_ID =
PORTFOLIO.PORTFOLIO_ID
GROUP BY PORTFOLIO.PAN_ID
HAVING PORTFOLIO.PAN_ID = 'APPHP9989S';
```



PAN_ID	NUMBER_OF_COMPANIES
APPHP9989S	3

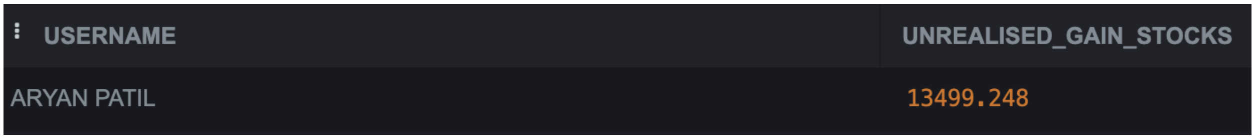
```
-- USER'S NAME WISE

SELECT USER.NAME AS USERNAME, COUNT(ISIN) AS NUMBER_OF_COMPANIES
FROM
STOCKS JOIN PORTFOLIO ON STOCKS.PORTFOLIO_ID =
PORTFOLIO.PORTFOLIO_ID
JOIN USER ON PORTFOLIO.PAN_ID = USER.PAN_ID
GROUP BY PORTFOLIO.PAN_ID
HAVING USER.NAME = 'SHRUTI VANKA';
```



USERNAME	NUMBER_OF_COMPANIES
SHRUTI VANKA	3

```
-- 2ND USER BUYS STOCKS OF ZOMATO

INSERT INTO TRANSACTIONS
VALUES('TSVST201','SV101',3400,'STOCKS','2023-11-05');

insert into STOCK_BUY VALUES('TSVST201',8,0,'INE758T01015');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TSVST201';
```

```
insert into stock_info values('INE758T01015',
'ZOMATO',179.65,847.98);

insert into STOCKS values('INE758T01015','SV101',0,0,0);

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101')
) WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE758T01015' AND PORTFOLIO_ID = 'SV101')
WHERE STOCKS.ISIN = 'INE758T01015' AND PORTFOLIO_ID = 'SV101';

-- TOTAL UNREALISED GAIN FOR ARYAN PATIL

SELECT USER.NAME AS USERNAME,SUM(NET_GAIN) AS
UNREALISED_GAIN_STOCKS FROM
STOCKS JOIN PORTFOLIO ON STOCKS.PORTFOLIO_ID =
PORTFOLIO.PORTFOLIO_ID
JOIN USER ON PORTFOLIO.PAN_ID = USER.PAN_ID
GROUP BY PORTFOLIO.PAN_ID
HAVING USER.NAME = 'ARYAN PATIL';
```



USERNAME	UNREALISED_GAIN_STOCKS
ARYAN PATIL	13499.248

```
-- 2ND USER SELLING SOME STOCKS

INSERT INTO TRANSACTIONS VALUES('TSTSV201','SV101',-
2000,'STOCKS','2023-11-09');

INSERT INTO STOCK_SELL VALUES('TSTSV201',4,'INE758T01015');

UPDATE STOCK_BUY SET dummy_quantity = 4 WHERE transaction_id =
'TSVST201';

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101')
WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101')
) WHERE ISIN = 'INE758T01015' AND portfolio_id = 'SV101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE758T01015' AND PORTFOLIO_ID = 'SV101')
WHERE STOCKS.ISIN = 'INE758T01015' AND PORTFOLIO_ID = 'SV101';


-- BUYING A FEW STOCKS OF MARICO

INSERT INTO TRANSACTIONS
VALUES('TKSST101','KS101',11200,'STOCKS','2024-02-05');
```

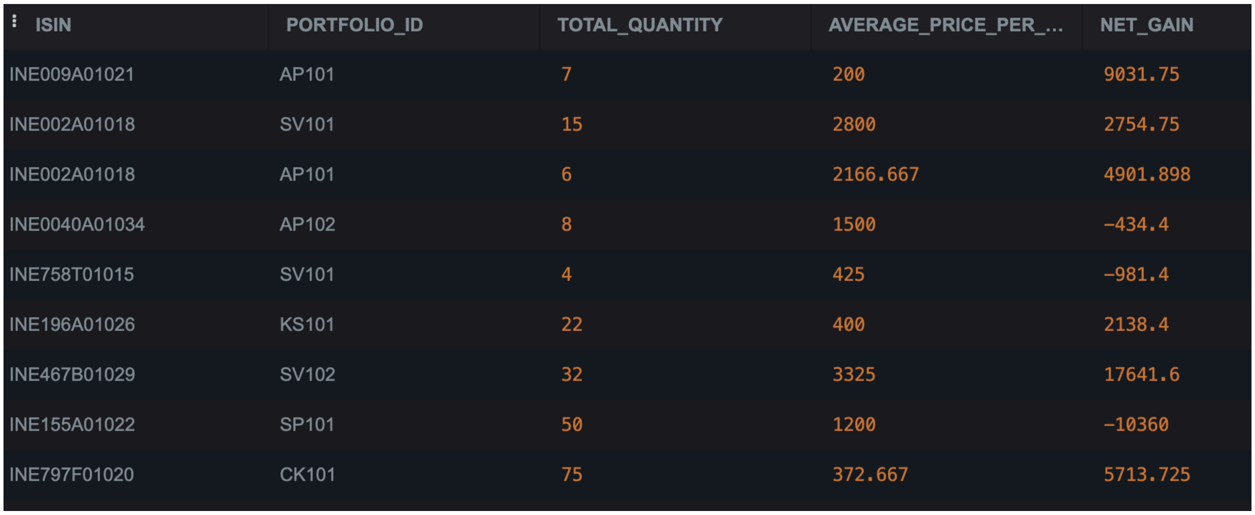
```
insert into STOCK_BUY VALUES('TKSST101',28,0,'INE196A01026');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TKSST101';

insert into stock_info values('INE196A01026',
'MARICO',497.20,51.21);

insert into STOCKS values('INE196A01026','KS101',0,0,0);

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
) WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE196A01026' AND PORTFOLIO_ID = 'KS101')
WHERE STOCKS.ISIN = 'INE196A01026' AND PORTFOLIO_ID = 'KS101';

-- SELLING A FEW STOCKS OF MARICO

INSERT INTO TRANSACTIONS VALUES('TSTKS101','KS101',-
1200,'STOCKS','2024-02-05');

INSERT INTO STOCK_SELL VALUES('TSTKS101',2,'INE196A01026');
UPDATE STOCK_BUY SET dummy_quantity = 26 WHERE transaction_id =
'TKSST101';
```

```
UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
) WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE196A01026' AND PORTFOLIO_ID = 'KS101')
WHERE STOCKS.ISIN = 'INE196A01026' AND PORTFOLIO_ID = 'KS101';

-- SELLING A FEW MORE STOCKS OF THE SAME

INSERT INTO TRANSACTIONS VALUES('TSTKS102','KS101',-
3200,'STOCKS','2024-02-05');

INSERT INTO STOCK_SELL VALUES('TSTKS102',4,'INE196A01026');

UPDATE STOCK_BUY SET dummy_quantity = 22 WHERE transaction_id =
'TKSST101';

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
```

```
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101')
) WHERE ISIN = 'INE196A01026' AND portfolio_id = 'KS101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE196A01026' AND PORTFOLIO_ID = 'KS101')
WHERE STOCKS.ISIN = 'INE196A01026' AND PORTFOLIO_ID = 'KS101';

-- BUYING STOCKS OF TCS FOR SV102

INSERT INTO TRANSACTIONS
VALUES('TSVST301','SV102',63000,'STOCKS','2024-02-15');

insert into STOCK_BUY VALUES('TSVST301',18,0,'INE467B01029');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TSVST301';

insert into stock_info values('INE467B01029',
'TCS',3876.30,34.87);

insert into STOCKS values('INE467B01029','SV102',0,0,0);

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102')
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102')
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102')
) WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE467B01029' AND PORTFOLIO_ID = 'SV102')
WHERE STOCKS.ISIN = 'INE467B01029' AND PORTFOLIO_ID = 'SV102';

-- BUYING MORE STOCKS OF TCS

INSERT INTO TRANSACTIONS
VALUES('TSVST302','SV102',43400,'STOCKS','2024-02-15');

insert into STOCK_BUY VALUES('TSVST302',14,0,'INE467B01029');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TSVST302';


UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102')
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102')
WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102')
) WHERE ISIN = 'INE467B01029' AND portfolio_id = 'SV102';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE467B01029' AND PORTFOLIO_ID = 'SV102')
WHERE STOCKS.ISIN = 'INE467B01029' AND PORTFOLIO_ID = 'SV102';

-- BUYING STOCKS OF TATA MOTORS

INSERT INTO TRANSACTIONS
VALUES('TSPST101','SP101',90000,'STOCKS','2024-01-29');

insert into STOCK_BUY VALUES('TSPST101',75,0,'INE155A01022');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TSPST101';

insert into stock_info values('INE155A01022', 'TATA
MOTORS',992.80,92.81);

insert into STOCKS values('INE155A01022','SP101',0,0,0);

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101')
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101')
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101')
) WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE155A01022' AND PORTFOLIO_ID = 'SP101')
WHERE STOCKS.ISIN = 'INE155A01022' AND PORTFOLIO_ID = 'SP101';
```

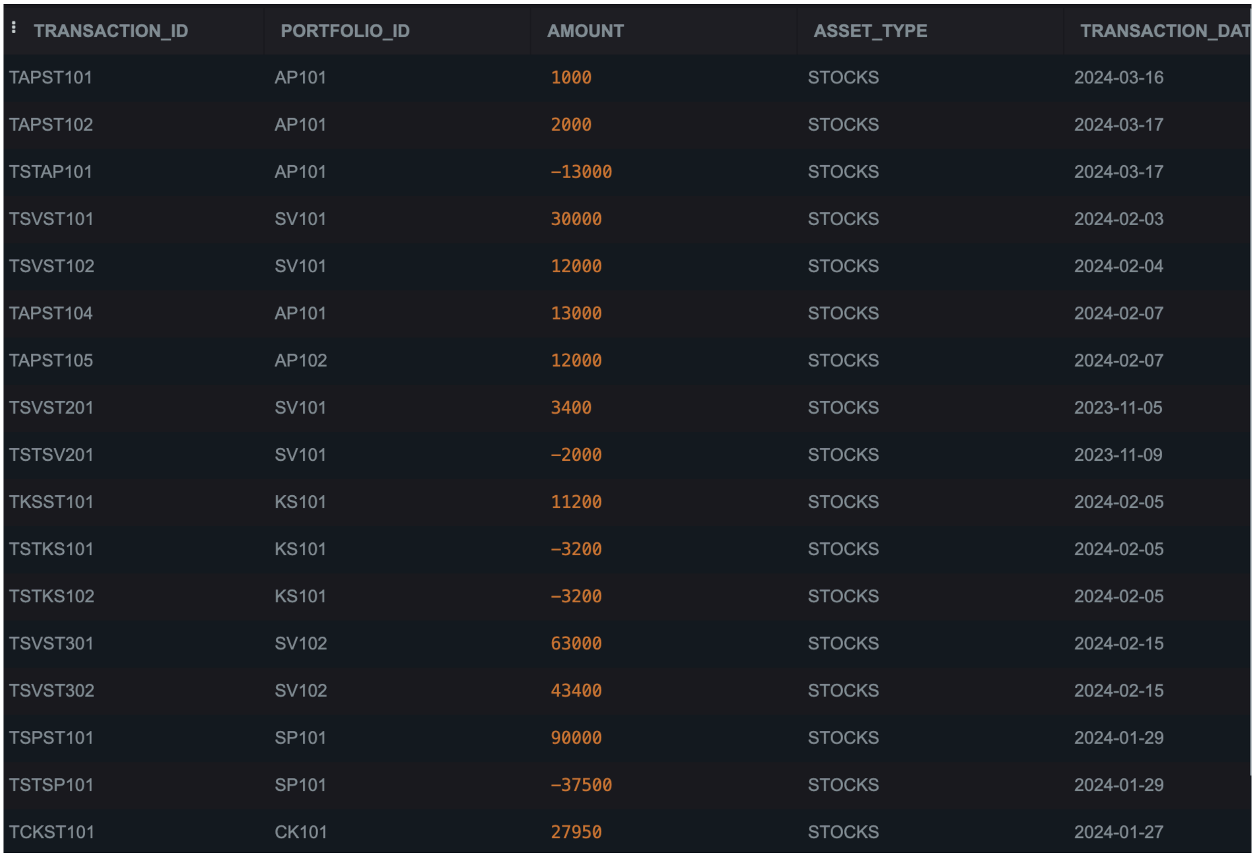
```
-- SELLING STOCKS OF TATA FOR SP101

INSERT INTO TRANSACTIONS VALUES('TSTSP101','SP101',-
37500,'STOCKS','2024-01-29');

INSERT INTO STOCK_SELL VALUES('TSTSP101',25,'INE155A01022');

UPDATE STOCK_BUY SET dummy_quantity = 50 WHERE transaction_id =
'TSPST101';

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101')
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101')
WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101')
) WHERE ISIN = 'INE155A01022' AND portfolio_id = 'SP101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE155A01022' AND PORTFOLIO_ID = 'SP101')
WHERE STOCKS.ISIN = 'INE155A01022' AND PORTFOLIO_ID = 'SP101';

-- BUYING STOCKS OF JUBILANT FOOD

INSERT INTO TRANSACTIONS
VALUES('TCKST101','CK101',27950,'STOCKS','2024-01-27');

insert into STOCK_BUY VALUES('TCKST101',75,0,'INE797F01020');
UPDATE STOCK_BUY SET DUMMY_QUANTITY = QUANTITY WHERE
TRANSACTION_ID = 'TCKST101';
```

```
insert into stock_info values('INE797F01020', 'JUBILANT
FOOD',448.85,66.54);

insert into STOCKS values('INE797F01020','CK101',0,0,0);

UPDATE STOCKS SET total_quantity =
(SELECT SUM(DUMMY_QUANTITY) FROM STOCK_BUY JOIN TRANSACTIONS ON
STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE797F01020' AND portfolio_id = 'CK101')
WHERE ISIN = 'INE797F01020' AND portfolio_id = 'CK101';

UPDATE STOCKS SET average_price_per_share = (
  SELECT
ROUND(SUM(PRICE_PER_SHARE*dummy_quantity)/total_quantity,2) FROM
(
  SELECT AMOUNT/QUANTITY AS PRICE_PER_SHARE, dummy_quantity FROM
(
  SELECT * FROM
TRANSACTIONS
JOIN STOCK_BUY
ON STOCK_BUY.TRANSACTION_ID = TRANSACTIONS.TRANSACTION_ID
WHERE ISIN = 'INE797F01020' AND portfolio_id = 'CK101')
WHERE ISIN = 'INE797F01020' AND portfolio_id = 'CK101')
) WHERE ISIN = 'INE797F01020' AND portfolio_id = 'CK101';

UPDATE STOCKS
SET net_gain =
(SELECT ROUND((current_value-
average_price_per_share)*total_quantity,2) FROM STOCKS JOIN
STOCK_INFO ON STOCKS.ISIN = STOCK_INFO.ISIN
WHERE STOCKS.ISIN = 'INE797F01020' AND PORTFOLIO_ID = 'CK101')
WHERE STOCKS.ISIN = 'INE797F01020' AND PORTFOLIO_ID = 'CK101';

-- STOCKS TABLE FINALLY
```



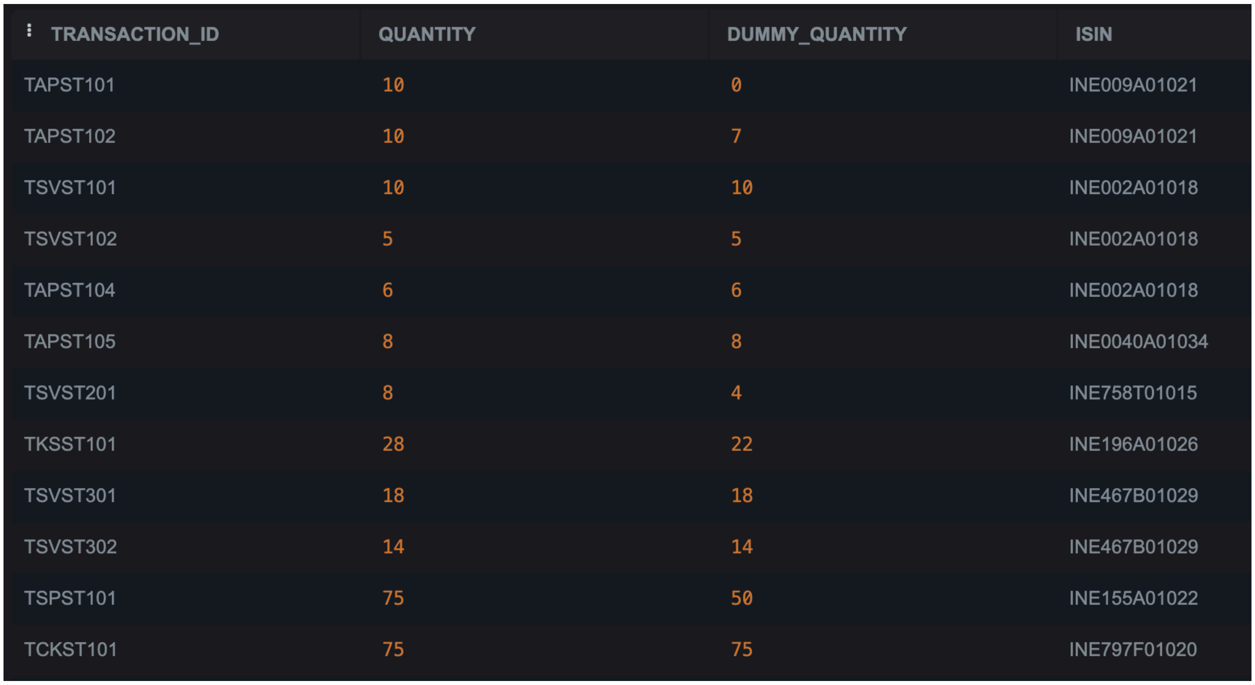
ISIN	PORTFOLIO_ID	TOTAL_QUANTITY	AVERAGE_PRICE_PER_...	NET_GAIN
INE009A01021	AP101	7	200	9031.75
INE002A01018	SV101	15	2800	2754.75
INE002A01018	AP101	6	2166.667	4901.898
INE0040A01034	AP102	8	1500	-434.4
INE758T01015	SV101	4	425	-981.4
INE196A01026	KS101	22	400	2138.4
INE467B01029	SV102	32	3325	17641.6
INE155A01022	SP101	50	1200	-10360
INE797F01020	CK101	75	372.667	5713.725

```
-- TRANSACTIONS AFTER STOCKS
```



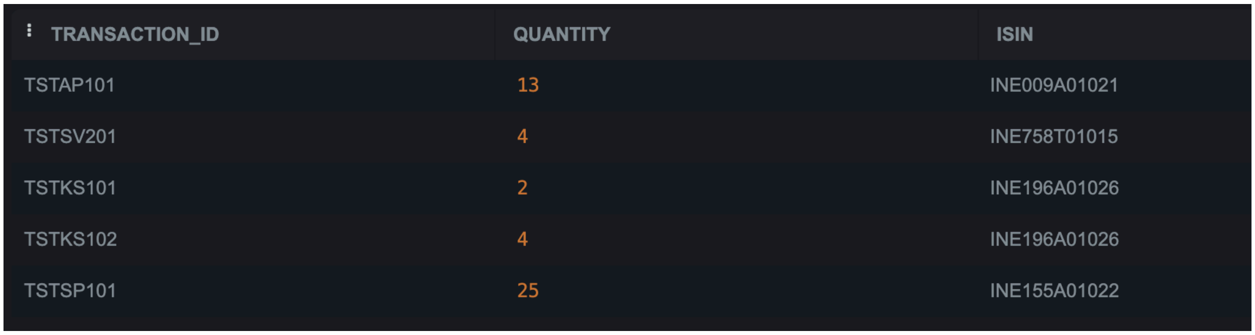
TRANSACTION_ID	PORTFOLIO_ID	AMOUNT	ASSET_TYPE	TRANSACTION_DAT
TAPST101	AP101	1000	STOCKS	2024-03-16
TAPST102	AP101	2000	STOCKS	2024-03-17
TSTAP101	AP101	-13000	STOCKS	2024-03-17
TSVST101	SV101	30000	STOCKS	2024-02-03
TSVST102	SV101	12000	STOCKS	2024-02-04
TAPST104	AP101	13000	STOCKS	2024-02-07
TAPST105	AP102	12000	STOCKS	2024-02-07
TSVST201	SV101	3400	STOCKS	2023-11-05
TSTSV201	SV101	-2000	STOCKS	2023-11-09
TKSST101	KS101	11200	STOCKS	2024-02-05
TSTKS101	KS101	-3200	STOCKS	2024-02-05
TSTKS102	KS101	-3200	STOCKS	2024-02-05
TSVST301	SV102	63000	STOCKS	2024-02-15
TSVST302	SV102	43400	STOCKS	2024-02-15
TSPST101	SP101	90000	STOCKS	2024-01-29
TSTSP101	SP101	-37500	STOCKS	2024-01-29
TCKST101	CK101	27950	STOCKS	2024-01-27

```
-- STOCK_BUY AFTER STOCKS
```



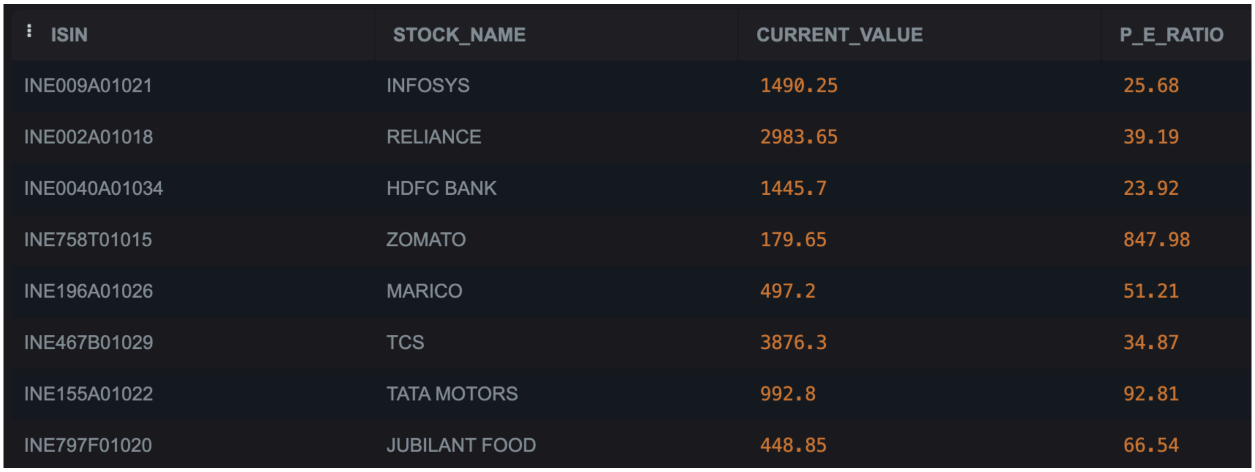
TRANSACTION_ID	QUANTITY	DUMMY_QUANTITY	ISIN
TAPST101	10	0	INE009A01021
TAPST102	10	7	INE009A01021
TSVST101	10	10	INE002A01018
TSVST102	5	5	INE002A01018
TAPST104	6	6	INE002A01018
TAPST105	8	8	INE0040A01034
TSVST201	8	4	INE758T01015
TKSST101	28	22	INE196A01026
TSVST301	18	18	INE467B01029
TSVST302	14	14	INE467B01029
TSPST101	75	50	INE155A01022
TCKST101	75	75	INE797F01020

-- STOCK_SELL AFTER STOCKS



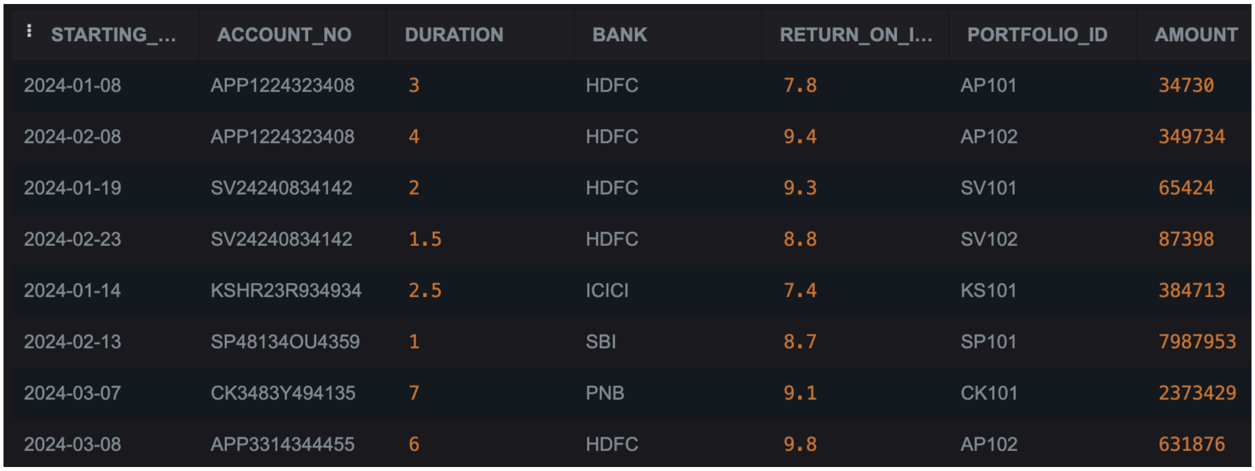
TRANSACTION_ID	QUANTITY	ISIN
TSTAP101	13	INE009A01021
TSTSV201	4	INE758T01015
TSTKS101	2	INE196A01026
TSTKS102	4	INE196A01026
TSTSP101	25	INE155A01022

-- STOCK_INFO AFTER STOCKS



ISIN	STOCK_NAME	CURRENT_VALUE	P_E_RATIO
INE009A01021	INFOSYS	1490.25	25.68
INE002A01018	RELIANCE	2983.65	39.19
INE0040A01034	HDFC BANK	1445.7	23.92
INE758T01015	ZOMATO	179.65	847.98
INE196A01026	MARICO	497.2	51.21
INE467B01029	TCS	3876.3	34.87
INE155A01022	TATA MOTORS	992.8	92.81
INE797F01020	JUBILANT FOOD	448.85	66.54

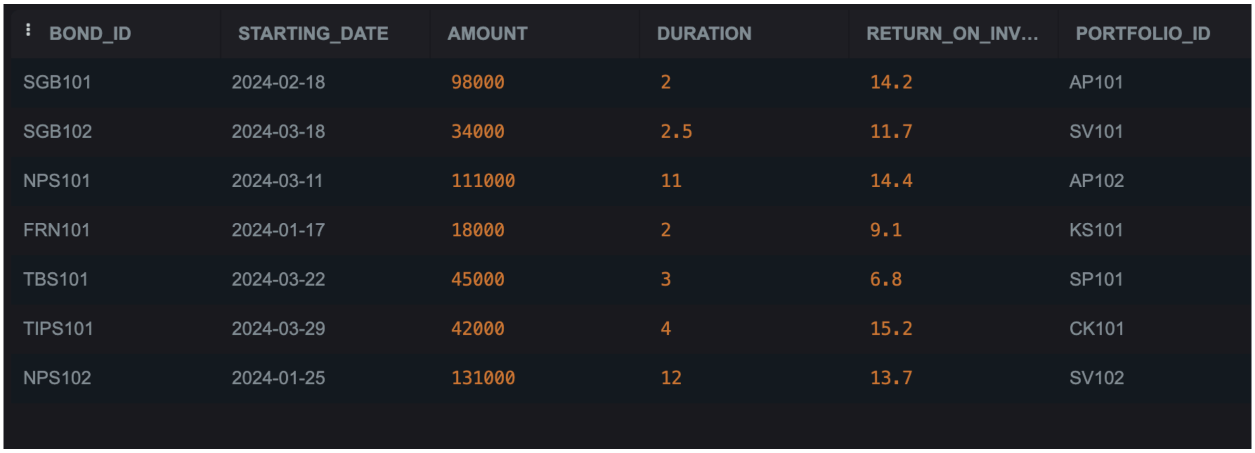
```
-- making some fixed deposits

insert into FIXED_DEPOSITS values('2024-01-
08','APP1224323408',3,'HDFC',7.8,'AP101',34730);
insert into FIXED_DEPOSITS values('2024-02-
08','APP1224323408',4,'HDFC',9.4,'AP102',349734);
insert into FIXED_DEPOSITS values('2024-01-
19','SV24240834142',2,'HDFC',9.3,'SV101',65424);
insert into FIXED_DEPOSITS values('2024-02-
23','SV24240834142',1.5,'HDFC',8.8,'SV102',87398);
insert into FIXED_DEPOSITS values('2024-01-
14','KSHR23R934934',2.5,'ICICI',7.4,'KS101',384713);
insert into FIXED_DEPOSITS values('2024-02-
13','SP48134OU4359',1,'SBI',8.7,'SP101',7987953);
insert into FIXED_DEPOSITS values('2024-03-
07','CK3483Y494135',7,'PNB',9.1,'CK101',2373429);
insert into FIXED_DEPOSITS values('2024-03-
08','APP3314344455',6,'HDFC',9.8,'AP102',631876);
-- SELECT * FROM FIXED_DEPOSITS
```



STARTING_...	ACCOUNT_NO	DURATION	BANK	RETURN_ON_I...	PORTFOLIO_ID	AMOUNT
2024-01-08	APP1224323408	3	HDFC	7.8	AP101	34730
2024-02-08	APP1224323408	4	HDFC	9.4	AP102	349734
2024-01-19	SV24240834142	2	HDFC	9.3	SV101	65424
2024-02-23	SV24240834142	1.5	HDFC	8.8	SV102	87398
2024-01-14	KSHR23R934934	2.5	ICICI	7.4	KS101	384713
2024-02-13	SP48134OU4359	1	SBI	8.7	SP101	7987953
2024-03-07	CK3483Y494135	7	PNB	9.1	CK101	2373429
2024-03-08	APP3314344455	6	HDFC	9.8	AP102	631876

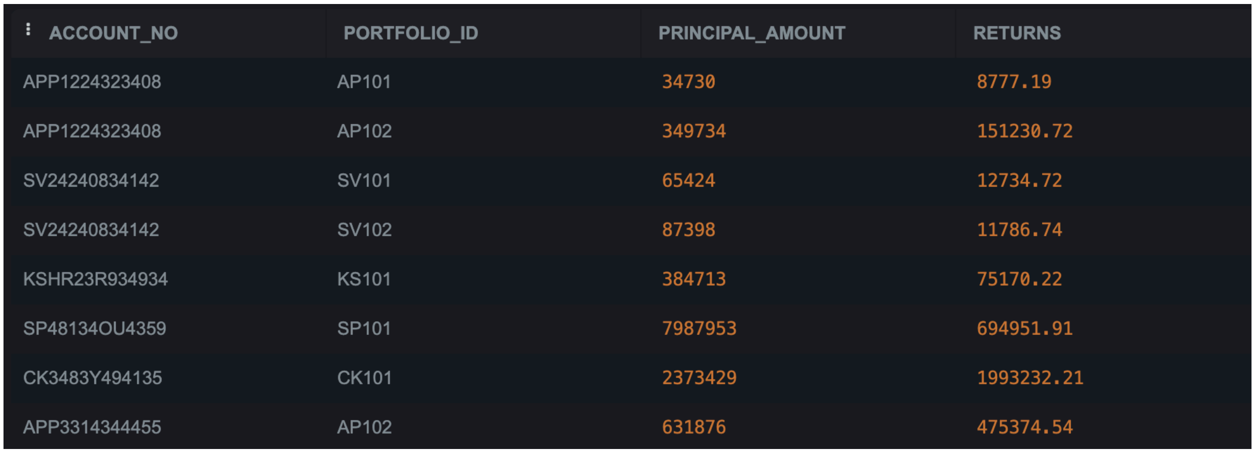
```
-- making some government bonds

insert into GOVT_BONDS VALUES('SGB101','2024-02-
18',98000,2,14.2,'AP101');
insert into GOVT_BONDS VALUES('SGB102','2024-03-
18',34000,2.5,11.7,'SV101');
insert into GOVT_BONDS VALUES('NPS101','2024-03-
11',111000,11,14.4,'AP102');
insert into GOVT_BONDS VALUES('FRN101','2024-01-
17',18000,2,9.1,'KS101');
insert into GOVT_BONDS VALUES('TBS101','2024-03-
22',45000,3,6.8,'SP101');
```

```
insert into GOVT_BONDS VALUES('TIPS101','2024-03-
29',42000,4,15.2,'CK101');
insert into GOVT_BONDS VALUES('NPS102','2024-01-
25',131000,12,13.7,'SV102');

-- SELECT * FROM GOVT_BONDS
```



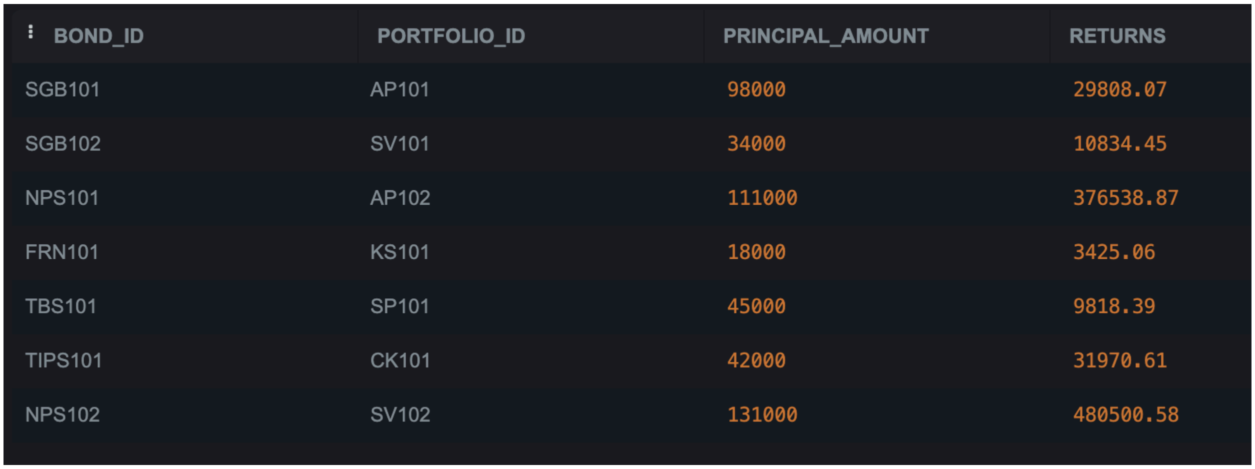
BOND_ID	STARTING_DATE	AMOUNT	DURATION	RETURN_ON_INV...	PORTFOLIO_ID
SGB101	2024-02-18	98000	2	14.2	AP101
SGB102	2024-03-18	34000	2.5	11.7	SV101
NPS101	2024-03-11	111000	11	14.4	AP102
FRN101	2024-01-17	18000	2	9.1	KS101
TBS101	2024-03-22	45000	3	6.8	SP101
TIPS101	2024-03-29	42000	4	15.2	CK101
NPS102	2024-01-25	131000	12	13.7	SV102

```
-- RETURNS ON FIXED_DEPOSITS FOR EACH FD AT THE END OF TENURE

SELECT account_no,
portfolio_id,
amount AS PRINCIPAL_AMOUNT,
ROUND((AMOUNT*POWER(1+return_on_investment*0.01,duration)-
AMOUNT),2) AS RETURNS
FROM FIXED_DEPOSITS;
```



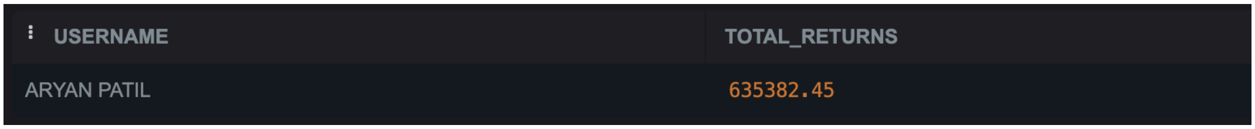
ACCOUNT_NO	PORTFOLIO_ID	PRINCIPAL_AMOUNT	RETURNS
APP1224323408	AP101	34730	8777.19
APP1224323408	AP102	349734	151230.72
SV24240834142	SV101	65424	12734.72
SV24240834142	SV102	87398	11786.74
KSHR23R934934	KS101	384713	75170.22
SP48134OU4359	SP101	7987953	694951.91
CK3483Y494135	CK101	2373429	1993232.21
APP3314344455	AP102	631876	475374.54

```
-- RETURNS ON GOVT_BONDS FOR EACH GB AT THE END OF TENURE

SELECT BOND_ID,
portfolio_id,
amount AS PRINCIPAL_AMOUNT,
ROUND((AMOUNT*POWER(1+return_on_investment*0.01,duration)-
AMOUNT),2) AS RETURNS
FROM GOVT_BONDS;
```



BOND_ID	PORTFOLIO_ID	PRINCIPAL_AMOUNT	RETURNS
SGB101	AP101	98000	29808.07
SGB102	SV101	34000	10834.45
NPS101	AP102	111000	376538.87
FRN101	KS101	18000	3425.06
TBS101	SP101	45000	9818.39
TIPS101	CK101	42000	31970.61
NPS102	SV102	131000	480500.58

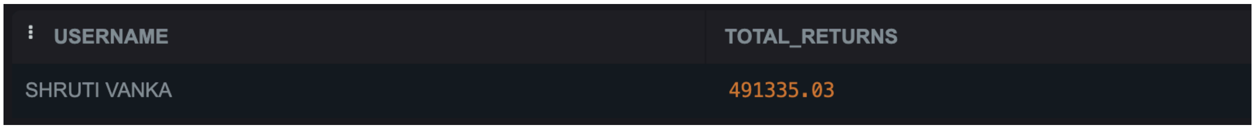
```
-- TOTAL RETURNS FOR A USER FROM ALL THE FIXED_DEPOSITS IN HIS
PORTFOLIOS

SELECT USER.NAME AS USERNAME,
ROUND(SUM(ROUND((AMOUNT*POWER(1+return_on_investment*0.01,durati
on)- AMOUNT),2)),2) AS TOTAL_RETURNS FROM
FIXED_DEPOSITS JOIN PORTFOLIO ON FIXED_DEPOSITS.PORTFOLIO_ID =
PORTFOLIO.PORTFOLIO_ID
JOIN USER ON PORTFOLIO.PAN_ID = USER.PAN_ID
WHERE USER.NAME = 'ARYAN PATIL';
```



USERNAME	TOTAL_RETURNS
ARYAN PATIL	635382.45

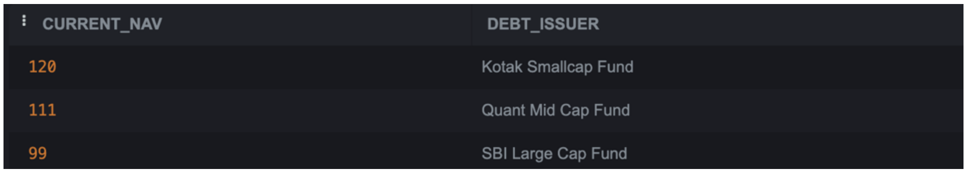
```
-- TOTAL RETURNS FOR A USER FROM ALL THE GOVT_BONDS IN HIS
PORTFOLIOS

SELECT USER.NAME AS USERNAME,
ROUND(SUM(ROUND((AMOUNT*POWER(1+return_on_investment*0.01,durati
on)- AMOUNT),2)),2) AS TOTAL_RETURNS FROM
GOVT_BONDS JOIN PORTFOLIO ON GOVT_BONDS.PORTFOLIO_ID =
PORTFOLIO.PORTFOLIO_ID
JOIN USER ON PORTFOLIO.PAN_ID = USER.PAN_ID
```

```
WHERE USER.NAME = 'SHRUTI VANKA';
```



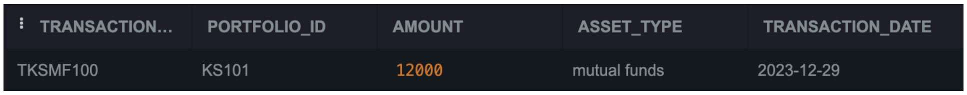
USERNAME	TOTAL_RETURNS
SHRUTI VANKA	491335.03

```
--INITIALIZE SCHEMES TABLE WITH MUTUAL FUNDS SCHEMES

INSERT INTO SCHEMES VALUES(120,'Kotak Smallcap Fund'),

  (111,'Quant Mid Cap Fund'),

                        (99,'SBI Large Cap Fund');
```



CURRENT_NAV	DEBT_ISSUER
120	Kotak Smallcap Fund
111	Quant Mid Cap Fund
99	SBI Large Cap Fund

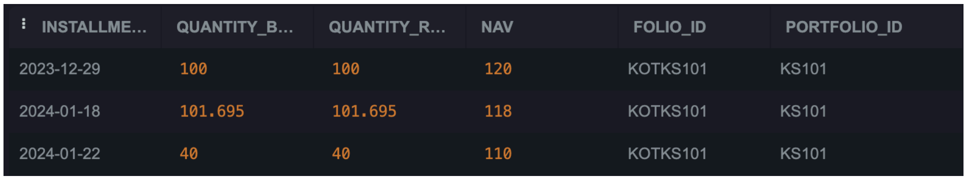
```
--1

--buy 100 units of Kotak Smallcap Fund

insert into TRANSACTIONS values('TKSMF100','KS101',12000,'mutual
funds','2023-12-29');
```



TRANSACTION...	PORTFOLIO_ID	AMOUNT	ASSET_TYPE	TRANSACTION_DATE
TKSMF100	KS101	12000	mutual funds	2023-12-29

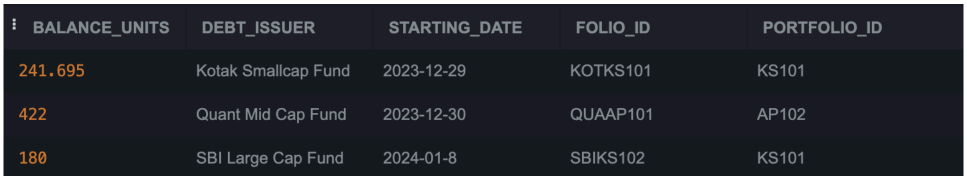
```
insert into MUTUAL_FUNDS values(100,'Kotak Smallcap Fund','2023-
12-29','KOTKS101','KS101');

insert into sip values('2023-12-
29',100,100,120,'KOTKS101','KS101');

update schemes set current_nav=118 where debt_issuer='Kotak
Smallcap Fund';

insert into sip values('2024-01-
18',101.695,101.695,118,'KOTKS101','KS101');

update MUTUAL_FUNDS set balance_units=balance_units+101.695
where folio_id='KOTKS101';

update schemes set current_nav=110 where debt_issuer='Kotak
Smallcap Fund';

--TOP-UPS FOR LUMPSUMS INCLUDED IN SIP TABLE
```

```
insert into sip values('2024-01-
22',40,40,110,'KOTKS101','KS101');

update MUTUAL_FUNDS set balance_units=balance_units+40 where
folio_id='KOTKS101';

--MUTUAL_FUNDS AFTER ALL TRANSACTIONS PERTAINING TO ABOVE
PORTFOLIO
```



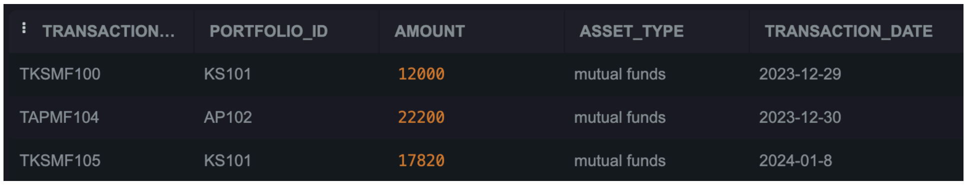
INSTALLME...	QUANTITY_B...	QUANTITY_R...	NAV	FOLIO_ID	PORTFOLIO_ID
2023-12-29	100	100	120	KOTKS101	KS101
2024-01-18	101.695	101.695	118	KOTKS101	KS101
2024-01-22	40	40	110	KOTKS101	KS101

```
--2

--buy 200 units of Quant Mid Cap Fund

insert into TRANSACTIONS values('TAPMF104','AP102',22200,'mutual
funds','2023-12-30');

insert into MUTUAL_FUNDS values(200,'Quant Mid Cap Fund','2023-
12-30','QUAAP101','AP102');

insert into sip values('2023-12-
30',200,200,111,'QUAAP101','AP102');

update schemes set current_nav=100 where debt_issuer='Quant Mid
Cap Fund';

insert into sip values('2024-01-
29',222,222,100,'QUAAP101','AP102');

update MUTUAL_FUNDS set balance_units=balance_units+222 where
folio_id='QUAAP101';

--3

--buy 180 of SBI Large Cap Fund this transaction is a lumpsum
will not have further installments

insert into TRANSACTIONS values('TKSMF105','KS101',17820,'mutual
funds','2024-01-8');

insert into MUTUAL_FUNDS values(180,'SBI Large Cap Fund','2024-
01-8','SBIKS102','KS101');

--MUTUAL_FUNDS FINAL TABLE
```



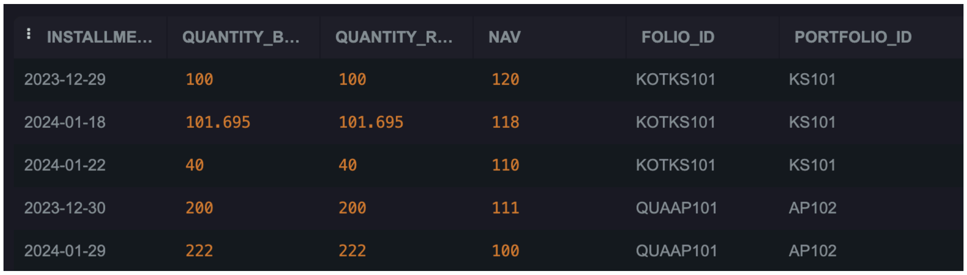
BALANCE_UNITS	DEBT_ISSUER	STARTING_DATE	FOLIO_ID	PORTFOLIO_ID
241.695	Kotak Smallcap Fund	2023-12-29	KOTKS101	KS101
422	Quant Mid Cap Fund	2023-12-30	QUAAP101	AP102
180	SBI Large Cap Fund	2024-01-8	SBIKS102	KS101

```
--TRANSACTIONS FINAL TABLE( MUTUAL FUND TRANSACTIONS)
```



TRANSACTION...	PORTFOLIO_ID	AMOUNT	ASSET_TYPE	TRANSACTION_DATE
TKSMF100	KS101	12000	mutual funds	2023-12-29
TAPMF104	AP102	22200	mutual funds	2023-12-30
TKSMF105	KS101	17820	mutual funds	2024-01-8

```
--SIP TABLE FINAL
```



INSTALLME...	QUANTITY_B...	QUANTITY_R...	NAV	FOLIO_ID	PORTFOLIO_ID
2023-12-29	100	100	120	KOTKS101	KS101
2024-01-18	101.695	101.695	118	KOTKS101	KS101
2024-01-22	40	40	110	KOTKS101	KS101
2023-12-30	200	200	111	QUAAP101	AP102
2024-01-29	222	222	100	QUAAP101	AP102

```
SELECT * FROM TRANSACTIONS;

select * from STOCKS;

SELECT * FROM STOCK_INFO;

SELECT * FROM STOCK_BUY;

SELECT * FROM STOCK_SELL;

select * from user;

SELECT * FROM PORTFOLIO;

SELECT * FROM FIXED_DEPOSITS;

SELECT * FROM GOVT_BONDS;
```