## Portfolio positions database design

Database: portfoliosdb

A securities and trades table were created using the DDL below to handle these 2 entities in that comprise portfolios.

CREATE TABLE securities (

isin STRING PRIMARY KEY,

multiplier DOUBLE,

price DOUBLE

);

isin: the unique identifier for the security

multiplier: as described in the problem set, this is divided by to get the market value

price: the price of the security

CREATE TABLE trades (

tradeid INTEGER PRIMARY KEY,

isin STRING REFERENCES securities (isin) ON DELETE CASCADE,

buyer BOOLEAN,

amount DOUBLE,

portfolioname STRING

);

tradeid: a unique id to identify the trade

isin: the unique identifier for the security

buyer: true or false flag to indicate whether the trades is a buy or sell

amount: the quantity of the trade

portfolioname: the name of the portfolio that the trade is under

#### Indices:

There are indices on the following:

1. The isin in the securities and trades tables in order to make the join condition when computing market value more efficient
2. The portfolioname in order to be able to filter by the portfolio efficiently

##### Keys:

There are primary keys on the following:

1. The isin in the securities table since it uniquely identifies each record
2. The tradeid in the trades table since it uniquely identifies each trade

There is a foreign key on isin in the trades table as well that references isin in the securities table. This is to maintain referential integrity so that if a trade is being inserted for a security that does not exist yet in the securities table, then it fails. Also the “on delete cascade” constraint is used so that if a security is removed from the securities table, then there won’t be any trades that reference those as well.

### Portfolio positions market value view

The following view was defined to represent the market value across the different positions in each portfolio

CREATE VIEW marketvalue\_positions\_view AS

SELECT T.portfolioname,

T.isin,

(sum(T.amount) \* S.price / S.multiplier) AS marketvalue

FROM trades T,

securities S

WHERE T.isin = S.isin

GROUP BY T.portfolioname,

T.isin;

To use the marketvalue filter, we can use the following query

Example: filter by positions with market value greater than 900.0

Select T.portfolioname, T.isin, (sum(T.amount) \* S.price / S.multiplier) as marketvalue

From trades T, securities S

Where T.isin = S.isin

Group By T.portfolioname, T.isin

Having marketvalue > 900.0;

However with the marketvalue\_positions\_view view defined as explained above, you can simply query the view with:

select \*

from marketvalue\_positions\_view

where marketvalue > 900.0;

#### Example:

securities

|  |  |  |
| --- | --- | --- |
| isin | multiplier | price |
| US1234567891 | 30 | 50 |
| US1111111111 | 20 | 40 |
| US2222222222 | 10 | 20 |
| US3333333333 | 10 | 30 |
| US4444444444 | 40 | 70 |
| US5555555555 | 20 | 90 |

trades

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tradeid | isin | buyer | amount | portfolioname |
| 1 | US1234567891 | 1 | 100 | ABC |
| 2 | US1234567891 | 1 | 200 | ABC |
| 3 | US1234567891 | 0 | 300 | ABC |
| 4 | US1111111111 | 1 | 200 | DEF |
| 5 | US1111111111 | 1 | 100 | DEF |
| 6 | US2222222222 | 0 | 600 | GHI |
| 7 | US3333333333 | 1 | 300 | DEF |

The marketvalue\_positions\_view shows

|  |  |  |
| --- | --- | --- |
| portfolioname | isin | marketvalue |
| ABC | US1234567891 | 1000 |
| DEF | US1111111111 | 600 |
| DEF | US3333333333 | 900 |
| GHI | US2222222222 | 1200 |

And if we filter by marketvalue > 900.0 as in the query example it gives

|  |  |  |
| --- | --- | --- |
| portfolioname | isin | marketvalue |
| ABC | US1234567891 | 1000 |
| GHI | US2222222222 | 1200 |