Aaryan Sharma
Vanshika Dhingra
Prisha Arora
10 November 2022

Project Phase 3

ER Mapping & Relational Model Conversion

Explanation

ER to relational model Mapping

The mapping from ER to the relation model was a 7-step process. It consisted of -

1. Conversion of strong entity types to relations/tables
   1.1. Decomposing composite attributes into simple attributes
   1.2. Mapping simple attributes to columns in the new table
2. Conversion of weak entity types to relations/tables
   2.1. Taking the Primary Key (PK) from the identifying strong entity as Foreign Key (FK) in the new table formed and making the Foreign Key-Partial Key pair the primary key in the new relation formed
3. Conversion of one-one relationship types to relations/tables
   3.1. Taking the Primary Key from one of the entity types and appending it to the other entity type's relation as Foreign Key
   3.2. The old Primary Key and the Foreign Key together both make for the new Primary Key of the relation/table
4. Conversion of one-many relationship types to relations/tables
   4.1. Taking the Primary Key from the entity type having cardinality constraint as N and appending it to the table formed by the other entity type and having it refer as the Foreign Key
   4.2. The old Primary Key and the Foreign Key together both make for the new Primary Key of the relation/table
5. Conversion of many-many relationship types to relations/tables
   5.1. Taking both table's Primary Key and creating the new table with the two of them referred to as a Foreign Key

5.2. Both the Foreign Key together make for the newly created table's Primary Key

6. Handling multivalued attributes

6.1. Creating a new table with columns

6.1.1. Primary Key from the relation the multivalued attribute originates from and making it the Foreign Key

6.1.2. Multivalued attribute itself as the second column

6.2. The Foreign Key-Multivalued attribute pair make the Primary Key of the newly created table

7. Conversion of N-ary relationship types to relations/tables

7.1. Taking the Primary Key from all the participating relations & creating a new relation with all of them as the column, and referring to all of them as the Foreign Key

7.2. All of the Foreign Keys put together make the Primary Key of the newly created table

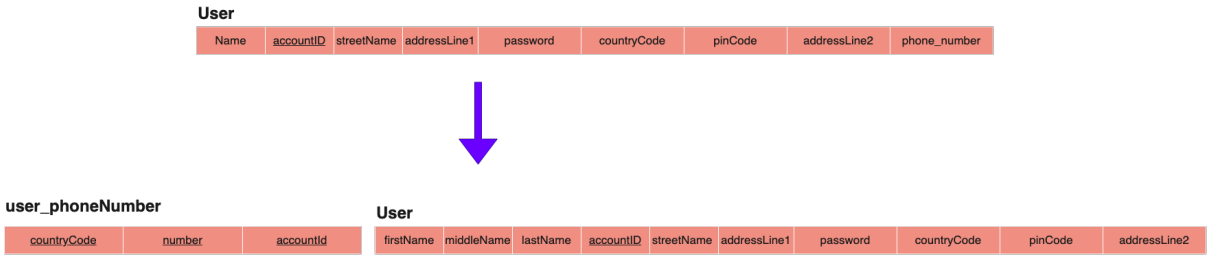
Relational model after conversion to 1NF, 2NF & 3NF

Definitions & Explanations

1NF: A relation is in 1NF if it contains an atomic value/attribute.

After following the 7-step process, we found the newly formed relations were already in 1NF as we dealt with all the composite and multivalued attributes while following the steps.

This step was done as follows:



2NF: A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key.

We found that all the columns in our relations were fully functionally dependent on the Primary Key of the relation, and hence they were already in 2NF

3NF: A relation will be in 3NF if it is in 2NF and no transition dependency exists.

We found that all the columns in our relations were independent of each other i.e. there was no functional dependency between columns of any relation.

As a consequence, all the relations were already in 3NF.

The relational model is attached on the next page

Stock/ Comprises of

currentPrice	quantity	bookValue	stockPE	dividendYield	faceValue	marketCap	high/low	securityCode	prevClose	ROE	ROCE	transactionId

Transaction

TransactionId	currentPrice	quantity	purchasePrice	comment	purchaseTIme

User/Supervision/Performs

firstName	middleName	lastName	accountID	streetName	addressLine1	password	countryCode	pinCode	addressLine2	transactionId	parent_accountId

Admin

AccountID	emailId	password

Feedback/Gives

title	accountId	message

portfolioList

indexTag	priceChange	percentageChange	securityCode	instrumentName	currentPrice	serialNo	totalQuantity	dayChange	marketValue	accountId

watchList

indexTag	priceChange	percentageChange	securityCode	instrumentName	currentPrice	serialNo	accountId

contains

securityCode	accountId	serialNo

user_phoneNumber

countryCode	number	accountId

has

accountId	portfolioList.serialNo	watchlist.serialNo

views

admin.accountId	portfolioList.serialNo	watchlist.serialNo	title	user.accountId