STOCK

MANAGER

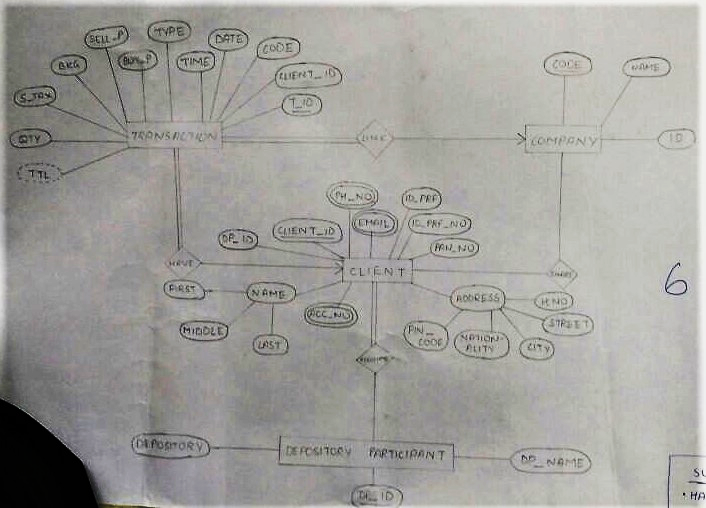
v1.0

By: Harguna Sood

(101510028)

(CML-2)

**E-R DIAGRAM**

****

The design of database system is as such that it enables a **stock broker** to make transactions for his/her **client**. The **stock broker** can have zero or more **clients**. Each **client** can have one or more **transactions** but each **transaction** can be made by only one **client**. Each **client** is further associated with one or more **depository**(ies) and each **depository** can have zero or more **clients**. **Transactions** are made of shares which belong to a **company**. Each **company** has its own share. Each **company** can give its share to zero or more **clients** and each **client** can have shares of zero or more **companies**. Also, each **transaction** can have only one **company**’s share but each **company** can be in zero or more **transactions**.

**E-R DIAGRAM 🡪 TABLE**

**CONVERSION**

**Rule:1**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T\_id | Client\_id | Code | Date | Time | Type | Buy\_p | Sell\_p | Bkg | S\_tax | Qty | Ttl |
|  |  |  |  |  |  |  |  |  |  |  |  |

Create separate table for each entity set

**TRANSACTION:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Client\_id | Name | Ph\_no | Email | Address | Id\_prf | Id\_prf\_no | Pan\_no | Acc\_no | Dp\_id |
|  |  |  |  |  |  |  |  |  |  |

**CLIENT:**

**COMPANY:**

|  |  |  |
| --- | --- | --- |
| Code | C\_name | Id |
|  |  |  |

**DP:**

|  |  |  |
| --- | --- | --- |
| Dp\_id | Dp\_name | Depository |
|  |  |  |

**Rule:2**

In case of composite attribute, create column for branches and ignore the root.

**CLIENT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Client\_id | First\_name | Middle\_name | Last\_name | Pin\_code | Nationality | City |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Street | H\_no | Ph\_no | Email | Id\_prf | Id\_prf\_no | Pan\_no | Acc\_no | Dp\_id |
|  |  |  |  |  |  |  |  |  |

**Rule:3**

No column for derived attribute

**TRANSACTION**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T\_id | Client\_id | Code | Date | Time | Type | Buy\_p | Sell\_p | Bkg | S\_tax | Qty |
|  |  |  |  |  |  |  |  |  |  |  |

**Rule:4**

For every multi-valued attribute, create new table

**CLIENT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Client\_id | First\_name | Middle\_name | Last\_name | Pin\_code | Nationality | City |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Street | H\_no | Id\_prf | Id\_prf\_no | Pan\_no | Dp\_id |
|  |  |  |  |  |  |

**ACCOUNT:**

|  |  |
| --- | --- |
| Client\_id | Acc\_no |
|  |  |

**PHONE:**

|  |  |
| --- | --- |
| Client\_id | Ph\_no |
|  |  |

**EMAIL\_ID:**

|  |  |
| --- | --- |
| Client\_id | Email |
|  |  |

Now, for **many to one relations**, (i.e. Transacion-Client, Transaction-Company) copy the pk of ‘one’ to ‘many’ side i.e. ‘client\_id’ and ‘code’ to Transaction, which is already there.

Now, for **many to many relations,** i.e.(Client-Company, Client-DP), create a new table:

|  |  |
| --- | --- |
| Client\_id | Code |
|  |  |

|  |  |
| --- | --- |
| Client\_id | DP\_id |
|  |  |

So, after conversion, we have following tables:

TRANSACTION, CLIENT, ACCOUNT, PHONE, EMAIL, DP, COMPANY, (Client\_id, Code), (Client\_id, DP\_id)

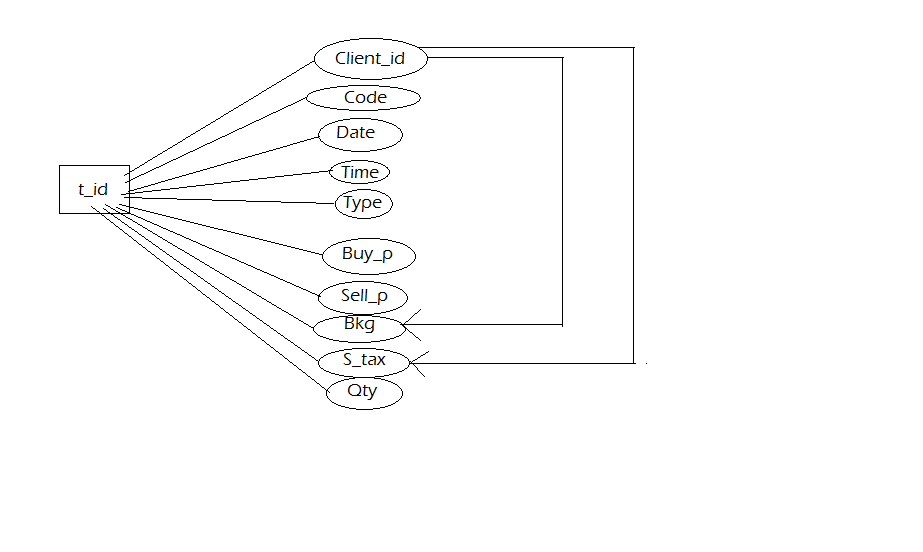
**NORMALIZATION**

**TRANSACTION**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T\_id | Client\_id | Code | Date | Time | Type | Buy\_p | Sell\_p | Bkg | S\_tax | Qty |
|  |  |  |  |  |  |  |  |  |  |  |

First N/F: It is already in first n/f as each row and column combination will have at least one value.

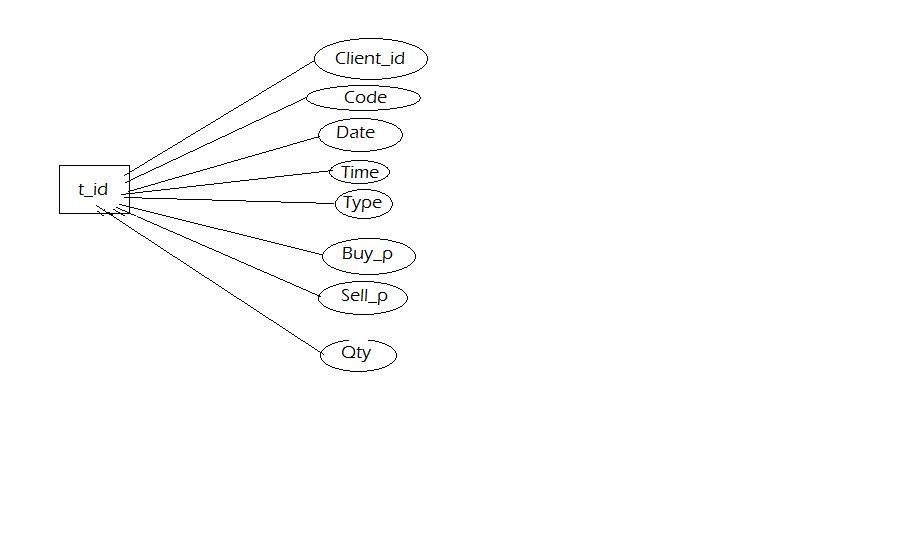
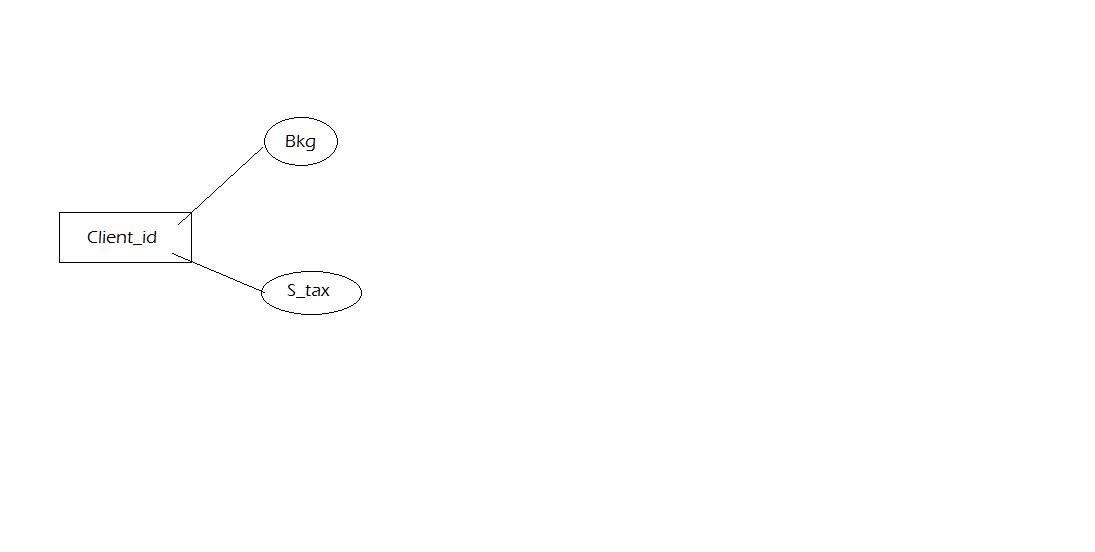
Second N/F:



It is already in first n/f and ‘t\_id’ functionally determines all other non-key attributes. Thus, it is in second n/f.

Third N/F:

It is already in second n/f but there is a transitive edge. Thus, it is not in third n/f

Now, there is not transitive edge and in new table client\_id functionally determines bkg and s\_tax, thus, it is in third n/f.

**Tax:**

|  |  |  |
| --- | --- | --- |
| Client\_id | Bkg | S\_tax |
|  |  |  |

BC N/F: It satisfies bc n/f as no non-key attribute functionally determines ‘t\_id’.

Fourth N/F:

It is already in third n/f and there is no multi value dependency, thus, it is in fourth n/f.

Fifth N/F:

It is in fourth n/f abd can’t be decomposed any further into smaller tables, thus, is in fifth n/f.

**CLIENT:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Client\_id | First\_name | Middle\_name | Last\_name | Pin\_code | Nationality | City |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Street | H\_no | Id\_prf | Id\_prf\_no | Pan\_no | Dp\_id |
|  |  |  |  |  |  |

First N/F: It is already in first n/f as each row and column combination will have at least one value.

Second N/F:



It is already in first n/f and ‘client\_id’ functionally determines all other non-key attributes. Thus, it is in second n/f.

BC N/F:

It is in second n/f but id\_prf\_no and pan\_no are also key attributes i.e. candidate key. Thus, we will make two new table: (Client\_id, Pan\_no) and (Client\_id, id\_prf\_no)

**Pan: id\_no:**

|  |  |
| --- | --- |
| Client\_id | Pan\_no |
|  |  |

|  |  |
| --- | --- |
| Client\_id | Id\_prf\_no |
|  |  |



Third N/F:

It is already in second n/f and there is no transitive edge. Thus, it is in third n/f

Fourth N/F:

It is already in third n/f and there is no multi value dependency, thus, it is in fourth n/f.

Fifth N/F:

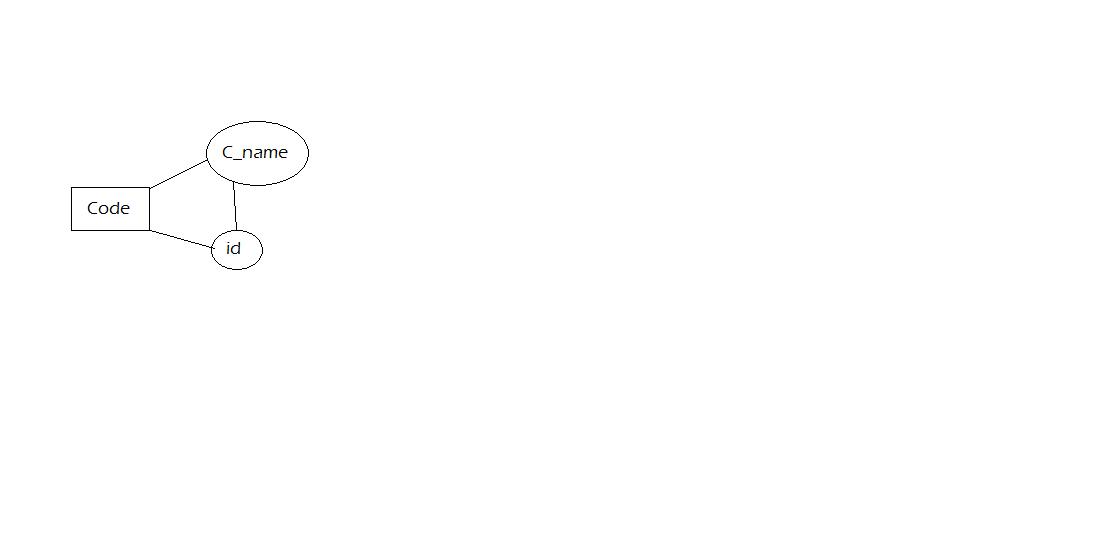
It is in fourth n/f abd can’t be decomposed any further into smaller tables, thus, is in fifth n/f.

**COMPANY:**

|  |  |  |
| --- | --- | --- |
| Code | C\_name | Id |
|  |  |  |

First N/F: It is already in first n/f as each row and column combination will have at least one value.

Second N/F:



It is already in first n/f and in second n/f as ‘code’ functionally determines ‘c\_name’ and ‘id’.

Third N/F: It is in second n/f but not in third n/f as there is a transitive edge. So, we will make new tables: (code, c\_name) and (id, c\_name). Now, it is in third n/f.

**Company\_code: Company\_id:**

|  |  |
| --- | --- |
| Code | C\_name |
|  |  |

|  |  |
| --- | --- |
| id | C\_name |
|  |  |

BC N/F: It satisfies bc n/f as the non-key attribute which functionally determines key attribute is already in separate table

Fourth N/F: It is already in third n/f and there is no multi value dependency, thus, it is in fourth n/f.

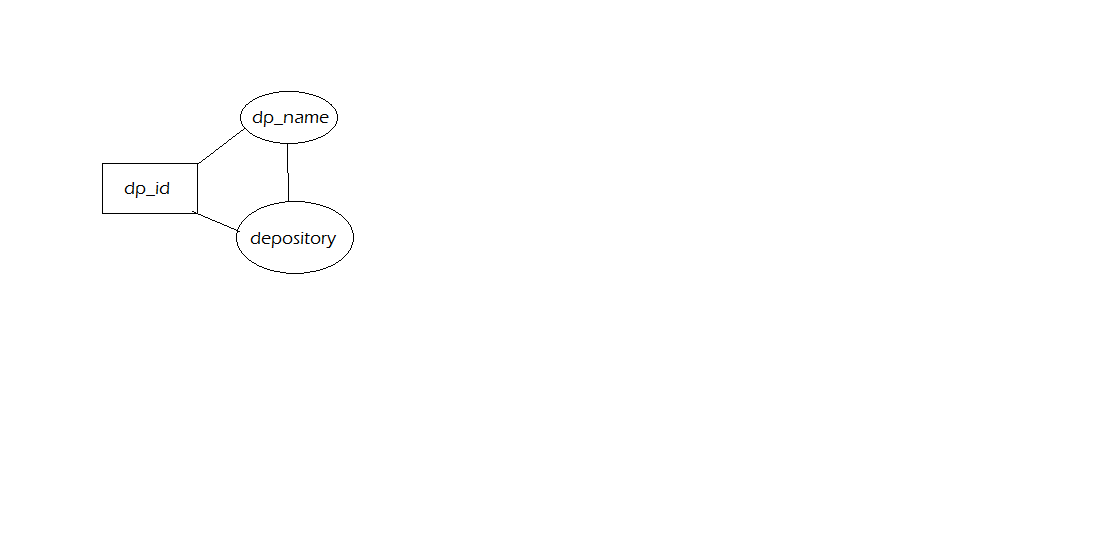
Fifth N/F: It is in fourth n/f abd can’t be decomposed any further into smaller tables, thus, is in fifth n/f.

**DP:**

|  |  |  |
| --- | --- | --- |
| Dp\_id | Dp\_name | Depository |
|  |  |  |

First N/F: It is already in first n/f as each row and column combination will have at least one value.

Second N/F:



It is already in first n/f and in second n/f as ‘dp\_id’ functionally determines ‘dp\_name’ and ‘depository’.

Third N/F: It is in second n/f but not in third n/f as there is a transitive edge. So, we will make new table: (dp\_name, depository). Now, it is in third n/f.

**Depository: DP:**

|  |  |
| --- | --- |
| Dp\_name | Depository |
|  |  |

|  |  |
| --- | --- |
| Dp\_name | DP\_id |
|  |  |

These tables are selected as it is convenient to remember depository name than its id and find them.

BC N/F: It satisfies bc n/f as attribute which functionally determines pk is already in separate table.

Fourth N/F: It is already in third n/f and there is no multi value dependency, thus, it is in fourth n/f.

Fifth N/F: It is in fourth n/f abd can’t be decomposed any further into smaller tables, thus, is in fifth n/f.

After normalization, we have following tables:

TRANSACTION(t\_id, client\_id, code, date, time, type, buy\_p, sell\_p, qty);

TAX(client\_id, bkg, s\_tax);

CLIENT(client\_id, dp\_id, id\_prf, first\_name, middle\_name, last\_name, pin\_code, nationality, city, street, h\_no);

PAN(client\_id, pan\_no);

ID\_NO(client\_id, id\_prf\_no);

COMPANY\_CODE(code, c\_name);

COMPANY\_ID(id, c\_name);

DEPOSITORY(dp\_name, depository);

DP(dp\_name, dp\_id);

PHONE(client\_id, ph\_no);

ACCOUNT(client\_id, acc\_no);

EMAIL\_ID(client\_id, email);

The tables: (client\_id, code) and (client\_id, dp\_id) are dropped as information in these is already in TRANSACTION and CLIENT respectively.