B.Sc (Computer Science)
Database Management Systems
UNIT- II

1. What is Normalization?

Normalization is a process of evaluating and correcting table structures to minimize data redundancies, thereby reducing the likelihood of data anomalies. The normalization process involves assigning attributes to the tables based on the concept of Relational Data Model.

The objective of normalization is to ensure that each table conforms to the concept of well-formed relations. Normalized tables have the following characteristics.

- Each table represents a single subject. For example, a course table will contain only data that directly pertains to courses. Similarly, a student table will contain only student data.
- No data item will be unnecessarily stored in more than one table. The reason for this requirement is to ensure that the data are updated in only one place.
- All non-prime attribute in a table are dependent on the primary key. The reason for this requirement is to ensure that the data are uniquely identifiable by a primary key value.
- Each table is void of insertion, update or deletion anomalies. This is to ensure the integrity and consistency of the data.

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

The above table is not normalized. Consider the following deficiencies:

- PROJ_NUM is apparently intended to be a primary key or atleast a part of PK, but it contains nulls.
- The table entries invite data inconsistencies. For example, the JOB_CLASS value "Database Designer" or "DB Designer".
- The table displays data redundancies.
    1. update anomalies: For example modifying JOB_CLASS for employee number 101 requires many alterations.

2. Insertion anomalies: Just to complete a row definition, a new employee must be assigned to a project. If the employee is not yet assigned, a phantom project must be created to complete the employee data entry.
3. Deletion anomalies: Suppose that only one employee is associated with a given project. If that employee leaves the company and employee data are deleted, the project information will also be deleted. To prevent the loss of the project information, a fictitious employee must be created just to save the project information.

2. When is a table in 1NF?

A relational table doesn't contain repeating groups. The existence of repeating groups do exist, they must be eliminated by making sure that each row defines a single entity. In addition, the dependencies must be identified to diagnose the normal form.

1NF starts with a simple three-step procedure:

Step1: Eliminate the repeating groups

Start by presenting the data in a tabular format, where each cell has a single value and there are no repeating groups. To eliminate repeating groups, eliminate the nulls by making sure that each repeating group attribute contains an appropriate data value.

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
15	Evergreen	101	John G. News	Database Designer	$105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	$105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	$35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	$18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	$26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
25	Starflight	101	John G. News *	Database Designer	$105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	$48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	$96.75	23.6
25	Starflight	118	James J. Frommer	General Support	$18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

Step2: Identify the Primary Key

PROJ_NUM is not an adequate primary key because the project number doesn't uniquely identify the rows. To maintain a proper primary key that will uniquely identify any attribute value, the new key must be composed of a combination or PROJ_NUM and EMP_NUM.

Step3: Identify all dependencies

The identification of PK is already identified the following dependency:

PROJ_NUM, EMP_NUM → PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOUR, HOURS

The project number identifies the project name.

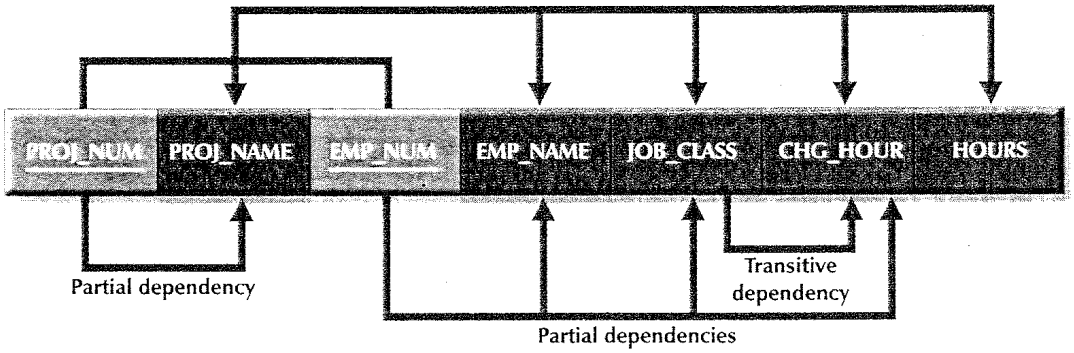
PROJ_NUM → PROJ_NAME

Employee number identifies the details of employee

EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR

Job classification identifies the employee's charge per hour.
JOB_CLASS → CHG_HOUR

The following diagram shows the various dependencies:



Partial dependencies: A dependency based on only part of a composite key is called as partial dependency. PROJ_NUM determines the PROJ_NAME; that is PROJ_NAME is dependent on only part of the primary key. And EMP_NUM is required to find EMP_NAME, JOB_CLASS,CHG_HOUR.

Transitive dependencies: A transitive dependency is a dependency of one non-prime attribute on another non-prime attribute. CHG_HOUR is dependent on JOB_CLASS.

3. When is a table in 2NF?

Converting to 2NF is done only when the 1NF has a composite primary key. If the 1NF has a single attribute primary key, then the table is automatically in 2NF.

Step1: Write each key component on a separate line

Write each component on a separate line; then write the original(Composite key) on the last line.

PROJ_NUM
EMP_NUM
PROJ_NUM, EMP_NUM

Each component will become the key in a new table. In other words, the original table is now divided into three tables (PROJECT, EMPLOYEE and ASSIGNMENT).

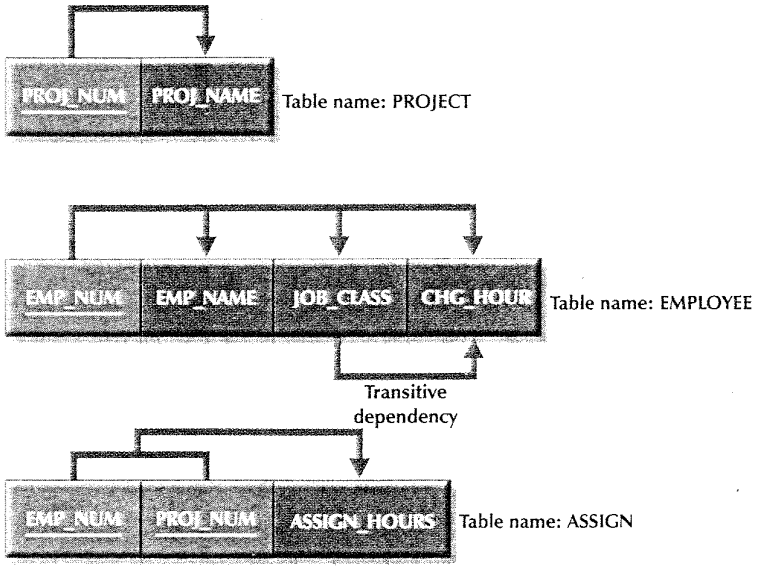
Step2: Assign corresponding dependent attributes

The three tables are described as follows:
PROJECT (PROJ_NUM, PROJ_NAME)

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

The number of hours spent on each project by each employee is dependent on both PROJ_NUM and EMP_NUM in the ASSIGNMENT table; place HOURS in the ASSIGNMENT table as ASSIGN_HOURS.



4. When is a table in 3NF?

A relation is said to be in 3NF if
(1). It is in 2NF
(2). There are no transitive dependencies

Step1: Identify each new determinant.

For every transitive dependency, write its determinant as a PK for a new table. A determinant is any attribute whose value determines other values within a row. If there are three dependencies, then there are three different determinants.

Write the determinant for this transitive dependency as:
JOB_CLASS

Step2: Identify the dependent attributes.

Identify the attributes that are dependent on each determinant identified in Step1 and identify the dependency.

JOB_CLASS - > CHG_HOUR

Step3: Remove the Dependent attributes from Transitive dependency.

Eliminate all dependent attributes in the transitive dependencies from each table that have such transitive relationship.

Eliminate CHG_HOUR from the employee table to leave the EMPLOYEE table dependency definition as:
EMP_NUM - > EMP_NAME, JOB_CLASS
Note that the JOB_CLASS remains in the EMPLOYEE table to serve as the FK.

The new tables are as follows:

PROJECT (**PROJ_NUM**, PROJ_NAME)

ASSIGN (**PROJ_NUM**, **EMP_NUM**, ASSIGN_HOURS)

EMPLOYEE (**EMP_NUM**, EMP_NAME, JOB_CLASS)

JOB (**JOB_CLASS**, CHG_HOUR)

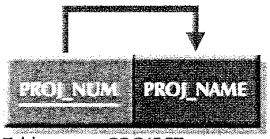


Table name: PROJECT

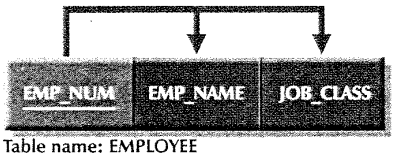


Table name: EMPLOYEE

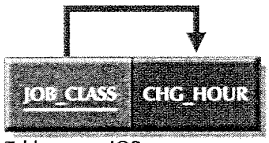


Table name: JOB

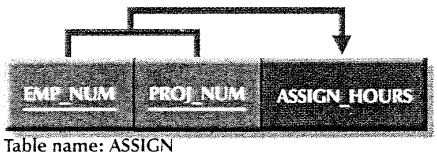


Table name: ASSIGN

5. What is a surrogate key and when should we use SK?

A composite key might become too cumbersome to use as the number of attributes grow. It becomes difficult to create a suitable foreign key when the related tables use composite primary key. For this reason, the primary key is considered to be unsuitable, designers use surrogate key.

At the implementation level, a surrogate key is a system-defined attribute generally created and managed via the DBMS. Usually, a system-defined surrogate key is numeric and its value is automatically incremented in each row. For example, MS-Access uses an AutoNumber, MS SQL Server uses an identity column and Oracle uses a sequence object.

6. Explain the types of Attributes

Composite & Simple Attributes: Attributes are primarily classified as Simple and Composite. A composite attribute is an attribute which can be divided into sub-attributes. For example ADDRESS can be divided into HNO, STREET,CITY,STATE and PINCODE. A PHONE_NO can be divided into AREA_CODE and EXCHANGE_CODE. A simple attribute is an attribute which can't be sub-divided. For example GENDER.

Single-Valued Attribute: It is an attribute that can have a single value. A person can have a single VOTERID. A single attribute is not necessarily a simple attribute. VOTERID can be classified according to STATE, DISTRICT.

Multi-Valued Attributes: An attribute that can have multiple values is called Multi-Valued Attribute. For example, a person can have several college degrees. A person can have multiple phone numbers.

Derived Attributes: It is an attribute whose value may be calculated from other attributes. The derived attribute need not be physically stored in the database; It can be derived by

using an algorithm. For example, an EMP_AGE is calculate as the difference of CURRENT_DATE and EMP_DOB.

Required and Optional Attribute: A required is an attribute that must have a value; in other words, it cannot be left empty. An optional attribute is an attribute is an attribute that doesn't require a value; therefore it can be left empty.

Domains: Attributes have a domain (A set of possible values for a given attribute). For example, the domain for gender attribute consists of only two possible values: M or F.

Identifier or Key Attribute: It is one or more attributes that uniquely identifies each entity instance. For example,STUD_NO and EMP_NO are Key Attributes.

8. What is Generalization and Specialization?

Generalization: An object set that is a superset of another object set.

Specialization: An object set that is a subset of another object set.

9. Explain Entity-Relationship diagram.

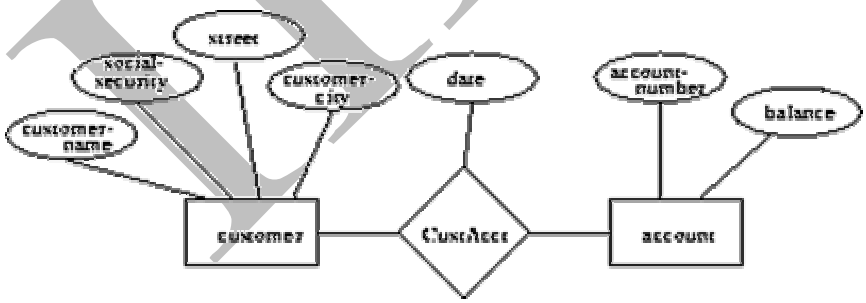
E-R Diagram: An entity-relationship diagram (ERD) is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. Any ER diagram has an equivalent relational table, and any relational table has an equivalent ER diagram.

Entity: The entity is a person, object, place or event for which data is collected. It is equivalent to a database table. An entity can be defined by means of its properties, called attributes. For example, the CUSTOMER entity may have attributes for such things as name, address and telephone number.

Relationship: The relationship is the interaction between the entities.

E-R diagram components are:

- Rectangles representing entity sets.
- Ellipses representing attributes.
- Diamonds representing relationship sets.
- Lines' linking attributes to entity sets and entity sets to relationship sets.



7. Explain about Relationships in E-R Model

A Relationship is an association between the entities. The entities that participate in a relationship are also known as participants. The entity-relationship is classifies as one-one, one-many or many-many. Cardinality expresses the minimum and maximum number of entity occurrences associated with one occurrence of the related entity. An entity is said to be existence dependent if it can exist in the database only when it is associated with another related entity occurrence.

Relationship Strength: The concept of relationship is based on how the primary key of a related entity is defined. To implement a relationship, the primary key of one entity appears as a foreign key in the related entity.

(b) Weak Relationship: A weak relationship is also called as Non-Identifying Relationship, exists if the PK of the related entity does not contain a PK component in the parent entity.

(c) Strong Relationship: A strong relationship is also known as Identifying Relationship, exists when the PK of the related entity contains a PK Component of the parent entity.

Relationship Degree: A Relationship degree indicates the number of entities associated with a relationship. The following are the types of relationships:

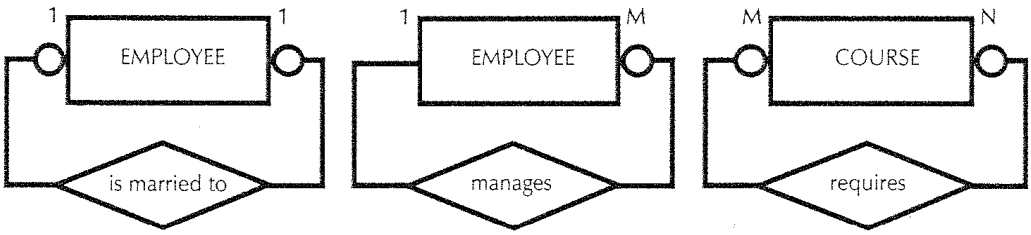
(a) Unary Relationship: It is a relationship exists when an association is maintained within a single entity.

(b) Binary Relationship: A binary relationship exists when two entities are associated in a relationship.

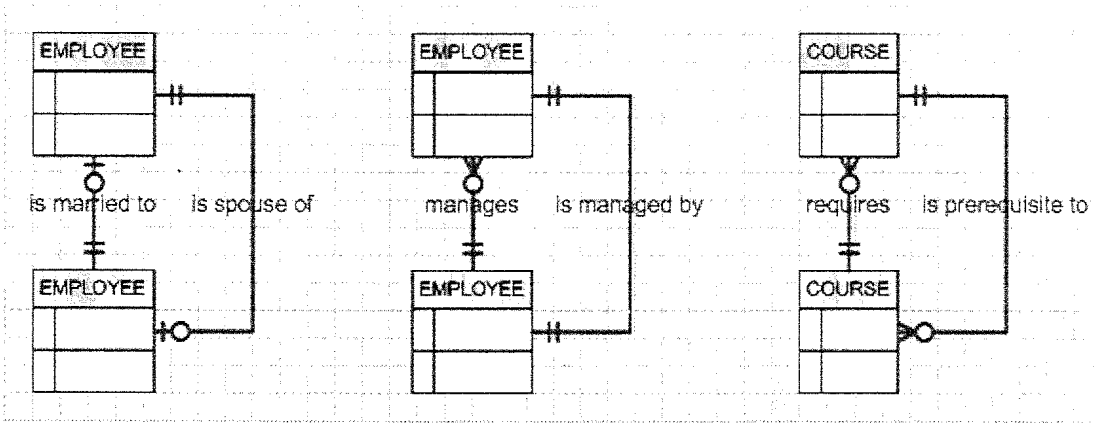
(c) Ternary Relationship: A ternary relationship exists when three entities are associated.

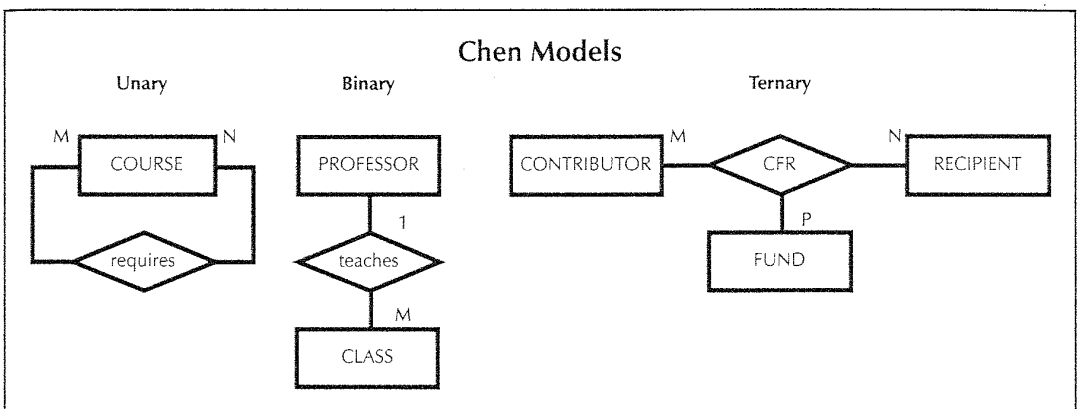
Recursive Relationship: A recursive relationship is one in which relationship can exist between occurrences of the same entity. Generally, such condition is found within unary relationship.

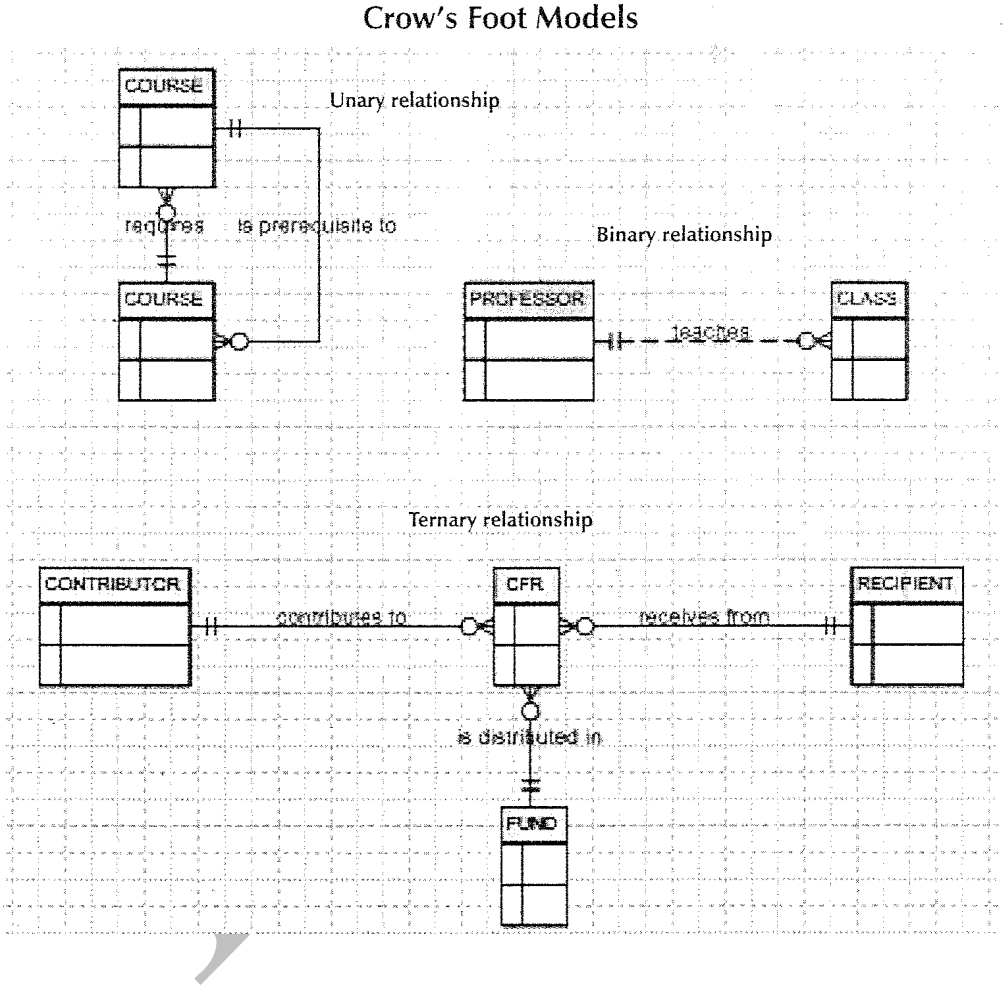
Chen Model



Crow's Foot Model




Chen Models
Unary
Binary
Ternary
M
N
COURSE
requires
PROFESSOR
1
teaches
M
CLASS
CONTRIBUTOR
M
CFR
N
RECIPIENT
P
FUND


Crow's Foot Models
COURSE
Unary relationship
requires
is prerequisite to
COURSE
Binary relationship
PROFESSOR
teaches
CLASS
Ternary relationship
CONTRIBUTOR
contributes to
CFR
receives from
RECIPIENT
is distributed in
FUND