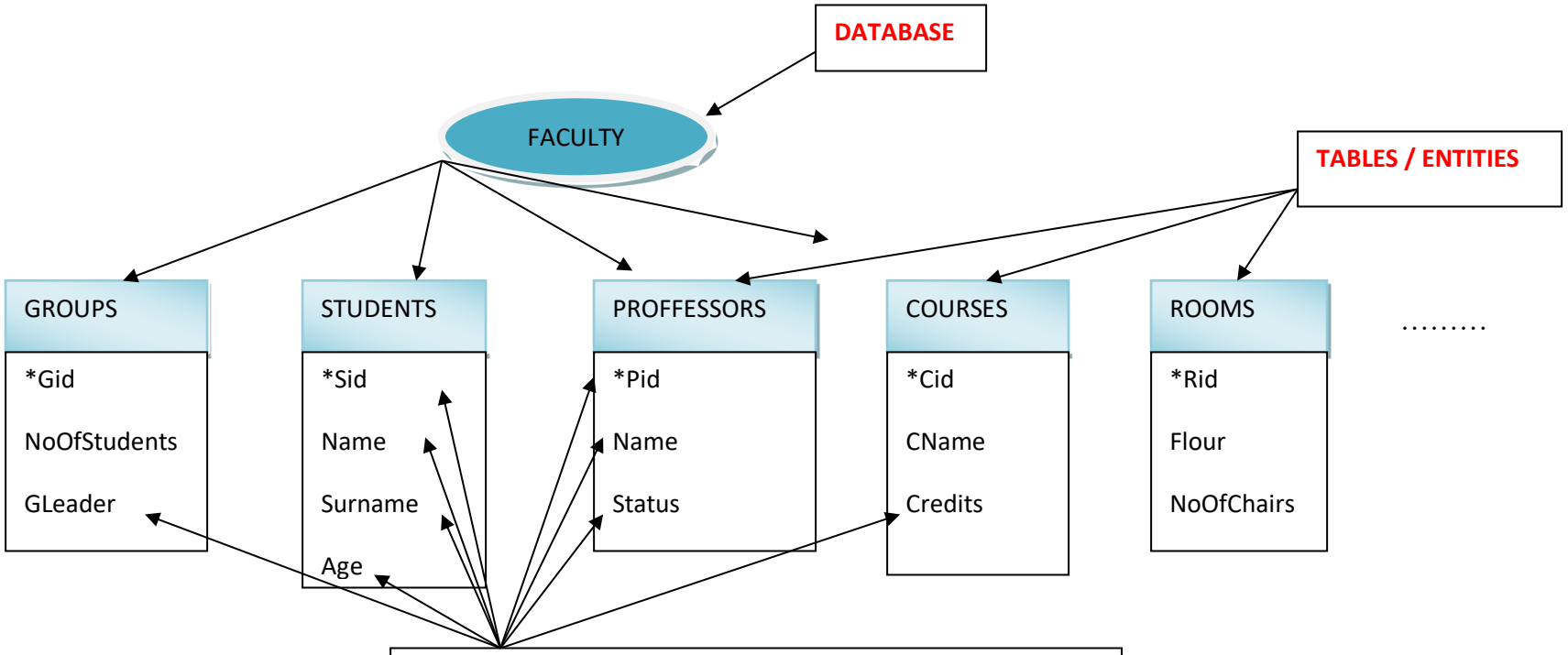
Emilia Pop

DATABASES

A database example



Each table has one or more **ATTRIBUTES / FIELDS / COLUMNS (in the table …)** (Gid, NoOfSTudents, GLeader, Sid, Name, …)

PRIMARY KEY = the field from a table that has UNIQUE values for each RECORD and it is NOT NULL. (Gid, Sid, …)
The primary key is denoted by * in front (in this document). Each table must have ONE primary key.

GROUPS

Gid	NoOfStudents	GLeader
921	28	Paul
923	27	Alin
223	29	Cristina
…		

RECORDS / LINES / ROWS (in the table …)

FOREIGN KEY = the PRIMARY KEY from the table used in the RELATIONSHIP with which establish the properties:

- have the SAME TYPE
- have the SAME VALUES (not necessarily unique, a value can appear for zero, one or more times)
- NOT NULL

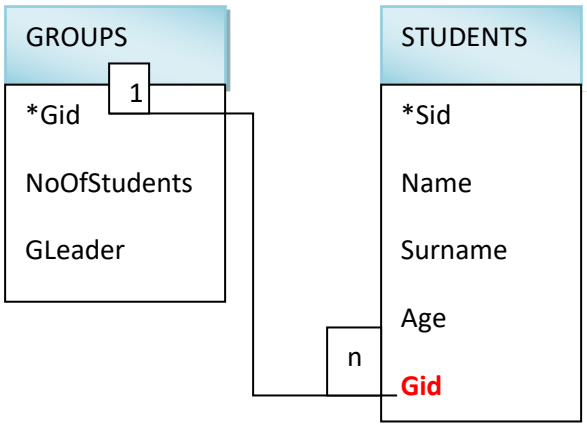
All – database, table, attribute, primary key, foreign key – SHOULD have only ONE WORD (even if are like NumberOfStudents), to access them easy.

RELATIONSHIP (2 tables involved)

- 1 – 1 : Students - DrivingLicences, Students - IdentityCards, Groups - LeaderGroups, Groups - Tutors, …
- 1 – n : Groups – Students, …
- m – n : Students – Courses, Professors – Students, …

Relationship 1-n (one to many)

- The ORDER of the tables is IMPORTANT
- 1 – n - Groups – Students - **In a Group** there are **one or more Students and a Student** is part of **only one Group**
- n – 1 – Students – Groups – **A Student** is from **only one Group and in a Group** there are **one or more Students**
- The PRIMARY KEY is taken from the table that is in the part 1 of the relationship and becomes FOREIGN KEY in the table that is in the part n of the relationship
- The first table that will be created is the table with the Primary key (the one from the part 1 of the relationship), so that one can extract the values from this one and use them as values for the foreign key



- Gid – can have any name that one wants in the table Students
- Gid from Groups and Gid from Students must have the same type (INT, VARCHAR(30), DATE, …)
- Gid from Students must have the same values as Gid from Groups, otherwise, error
- NULL cannot be inserted in Gid from Groups and also in Students

Gid	NoOfStudents	GLeader
921	28	Paul
923	27	Alin
223	29	Cristina
…		
…		

GROUPS (table)

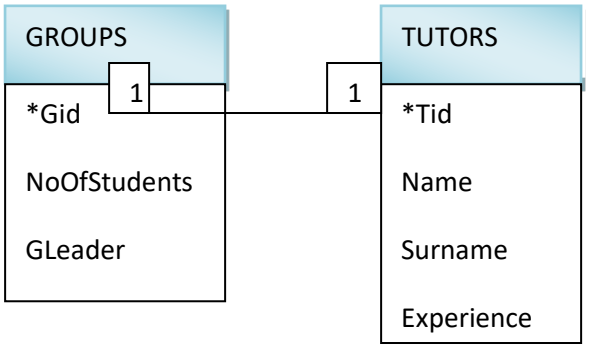
Sid	Name	Surname	Age	Gid
1	A	G	20	**921**
2	B	T	20	**921**
3	C	H	21	**921**
4	D	D	22	**223**
5	E	H	19	**223**
6	F	Y	20	**225**

Students (table)

No possible, because in Groups table the Gid value 225 does not exist.
An error message will be received.

Relationship 1 – 1 (one to one)

- A particular case of the relationship 1-n established between 2 primary keys – one it is also foreign key
- The first table created will have only primary key
- The second table created will have the primary key set it also as foreign key
- The order is not important (Groups – Tutors or Tutors-Groups)



- if the Groups is the first created, then the primary key is Gid in table Groups and

- Tid from Tutors will be set as Primary key with the same type as Gid from Groups and

- also Tid will be Foreign key in Tutors for Gid from Groups

Gid	NoOfStudents	GLeader
921	28	Paul
923	27	Alin
223	29	Cristina
…		
…		

GROUPS (table)

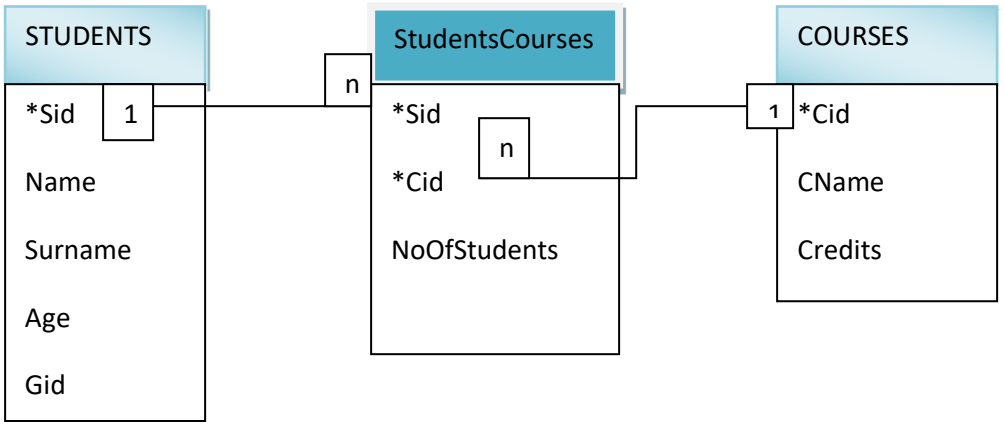
Tid	Name	Surname	Experience
921	N1	Paul	3
923	N2	Alin	4
223	N3	Cristina	5
921	Name 4	Andy	1

Tutors (table)

No possible, because Tid in table Tutors is primary key, so each value can appear only once. An error message will be received.

Relationship m-n (many to many)

- This relationship must contain an INTERMEDIATE Table – this table will contain both primary keys from the 2 tables involved set as FOREIGN KEY (and preferable also as Primary keys)
- Students - Courses – **one or more Students** can take part **in one or more Courses / A Student** can participate in **one or more Courses and in a Course** can participate **one or more Students**



- The intermediate table can be called as one wants (for example the name of both of the tables), or something specific (for example, Participations, or, StudentsCourses, or, …) (for example for tables Clients and Products can be called Orders/Commands)
- The intermediate table can contain other attributes, too
- This implementation with the primary keys set it as foreign key also, WON'T ALLOW DUPLICATE PAIRS

Sid	Name	Surname	Age	Gid
1	A	G	20	921
2	B	T	20	921

Students (table)

Cid	CName	Credits
11	Databases	6
22	MAP	6

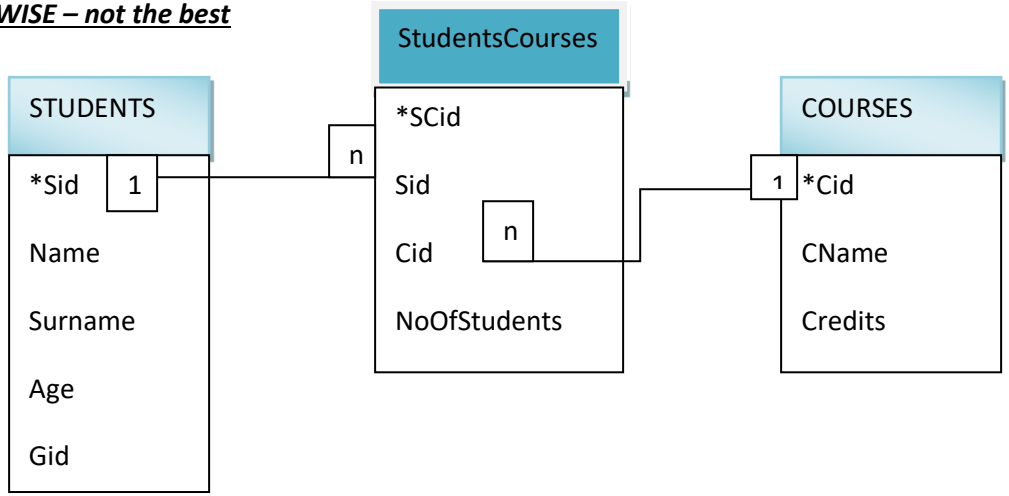
Courses (table)

All possible values will be displayed (because duplicates in pairs will not be allowed – the primary key pair is (Sid, Cid)):

Sid	Cid	NoOfStudents
1	11	21
1	22	21
2	11	21
2	22	23

StudentsCourses (table)

OTHERWISE – not the best



- The intermediate table can contain the same pairs for the foreign keys as many times as one wants (so, we can have duplicates, but with different values). The primary key is only SCid.

Sid	Name	Surname	Age	Gid
1	A	G	20	921
2	B	T	20	921

Students (table)

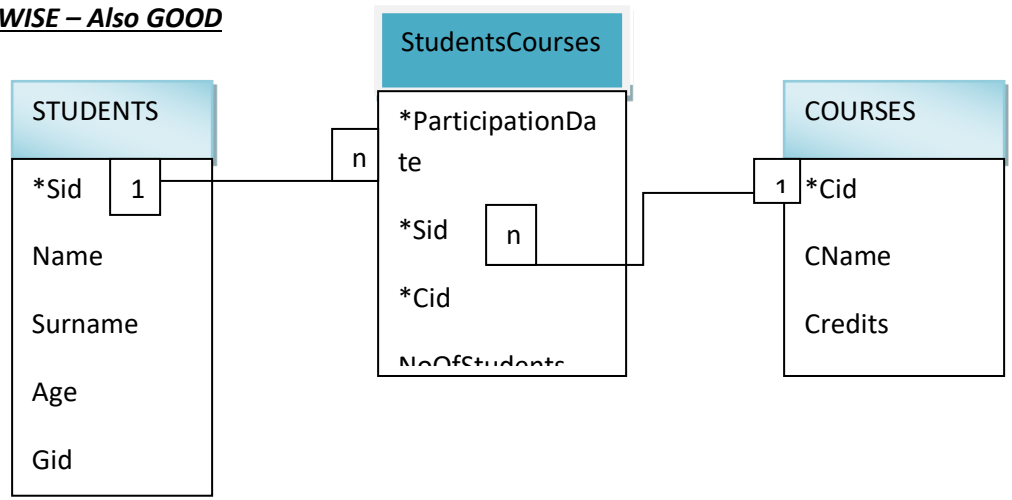
Cid	CName	Credits
11	Databases	6
22	MAP	6

Courses (table)

SCid	Sid	Cid	NoOfStudents
1	**1**	**11**	21
2	1	22	21
3	**1**	**11**	21
4	**1**	**11**	23

StudentsCourses (table)

OTHERWISE – Also GOOD



- The intermediate table will contain different pairs of records for the primary key (ParticipationDate, Sid, Cid)

Sid	Name	Surname	Age	Gid
1	A	G	20	921
2	B	T	20	921

Students (table)

Cid	CName	Credits
11	Databases	6
22	MAP	6

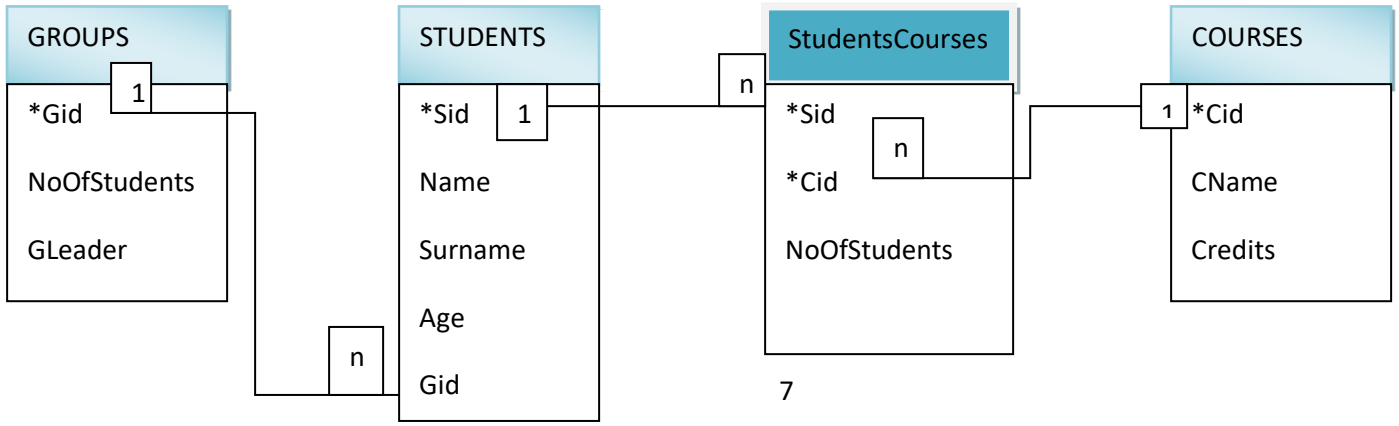
Courses (table)

ParticipationDate	Sid	Cid	NoOfStudents
12/12/2019	**1**	**11**	21
4/6/2020	1	22	21
1/1/2020	**1**	**11**	21
5/11/2020	**1**	**11**	23

StudentsCourses (table)

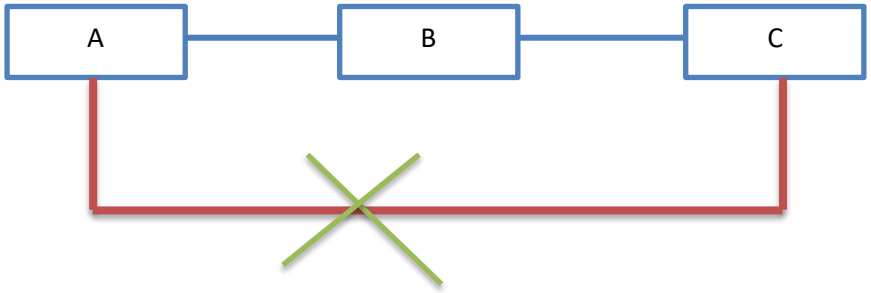
Relatioships – one should not use CICLES / CIRCLES.

- Do you need another relationship between Groups – Courses?
- NO, because there exist one – from Courses – to Students – to StudentsCourses – to Courses.
- Choose the 'principal relationships', the one you consider the most important and directly (for example Students are participating to Courses and Students are organized in Groups, and definitely Groups will be also related to Coursed, due to the Students) (other view can be Groups participate to Courses and the Students are organized in Groups, and so Students are related to Courses, due to Groups)



For example let A, B, C be 3 tables. Between them can be established relationships like:

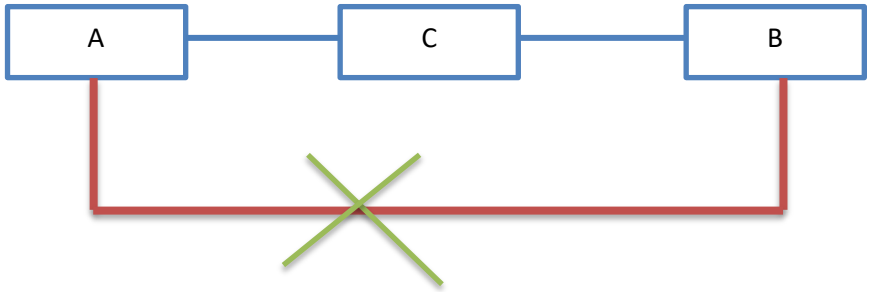
- A and B are in relationship; B and C are in relationship; so, A and C are in relationship through B



e.g. Department – Employers; Employers – Projects; and through Employers there is a relationship between Department – (Employers) – Projects.

The red line is not need it.

- A and C are in relationship; B and C are in relationship; so, A and B are in relationship through C



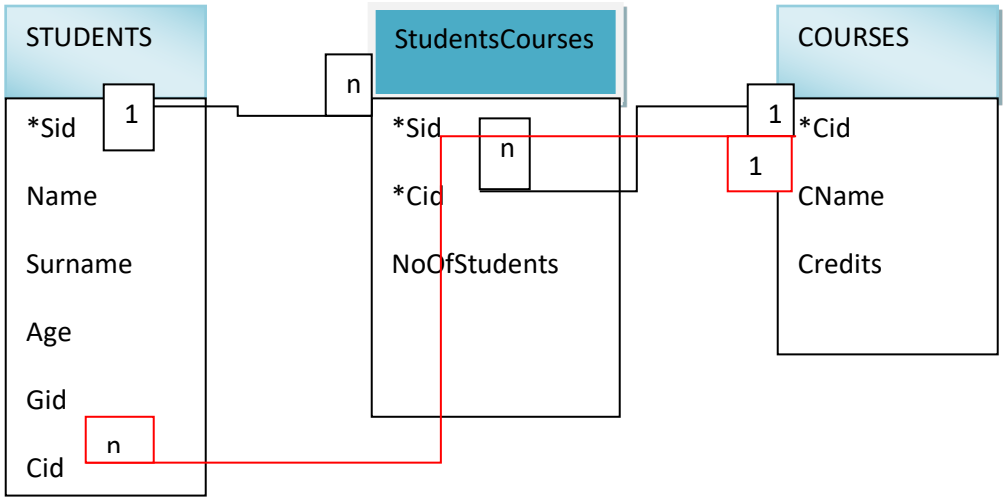
e.g. Department - Projects; Projects – Employers; and through Projects there is a relationship between Department – (Projects) - Employers.

The red line is not need it.

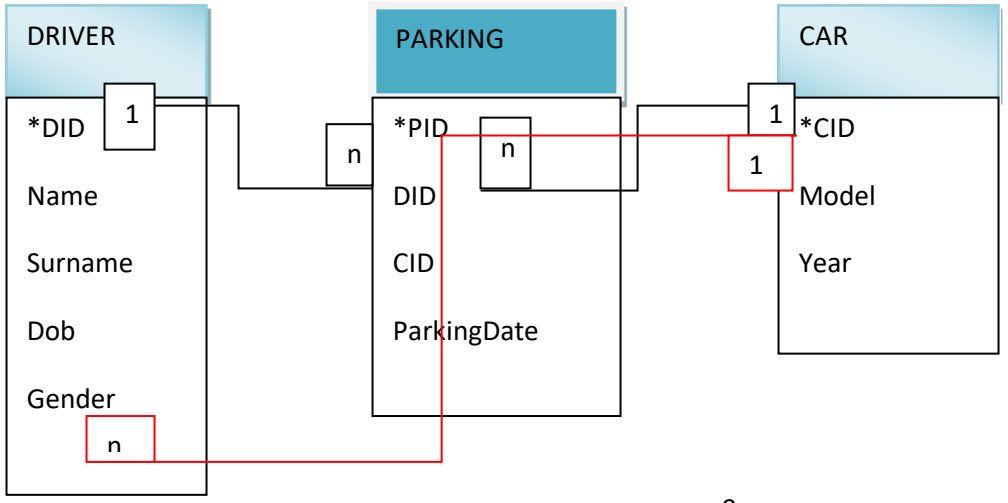
Observations

- In every table MUST EXIST at least ONE PRIMARY KEY
- In a database all tables must be related = each table must have at least one relationship with another table, such that one can extract data from each table using the relationships (please avoid cicles)
- A relationship can be of ONLY ONE TYPE (e.g. 1-n OR m-n) – Courses - Students has to be m-n OR 1-n, but not both of them! (ONLY black line OR red line) – otherwise it is a cicle

Example 1: STUDENTS – COURSES (m-n through StudentsCourses) OR STUDENTS – COURSES (1-m without table StudentsCourses)



Example 2: DRIVER – CAR (m-n through PARKING) OR DRIVER – CAR (1-m without table PARKING)



Address and Telephone are usually tables!!! NOT fields/attributes/columns in a table.

The Address table has attributes, like: City, Street, Number, Zip Code, County, Country, …

The Telephone table has also attributes, like: Phone Number, Mobile Number, Fax, … If someone wants to refer exclusively to only one of these fields (like, Mobile Number), then, the Telephone can be used as a field/column/attribute.

- Also, the isolated tables (=the tables that have no relation with any other table from the database) should be avoided (because we won't be able to access the information that is inside of them).
- The relationships are established always between 2 tables.
- Each table should be connected with at least one another, such that won't exist groups of tables or isolated tables inside of the database.

Possible 1-n relationships that can be applied for any database:

- Address – Person (1-n) – at a specified address can live one or multiple persons and a person lives at one address.
- TypeS – S (1-n) (e.g. BookType – Book, TeaType – Tea, ProductType – Product, …) – a type characterize multiple instances and an instance is of one type.