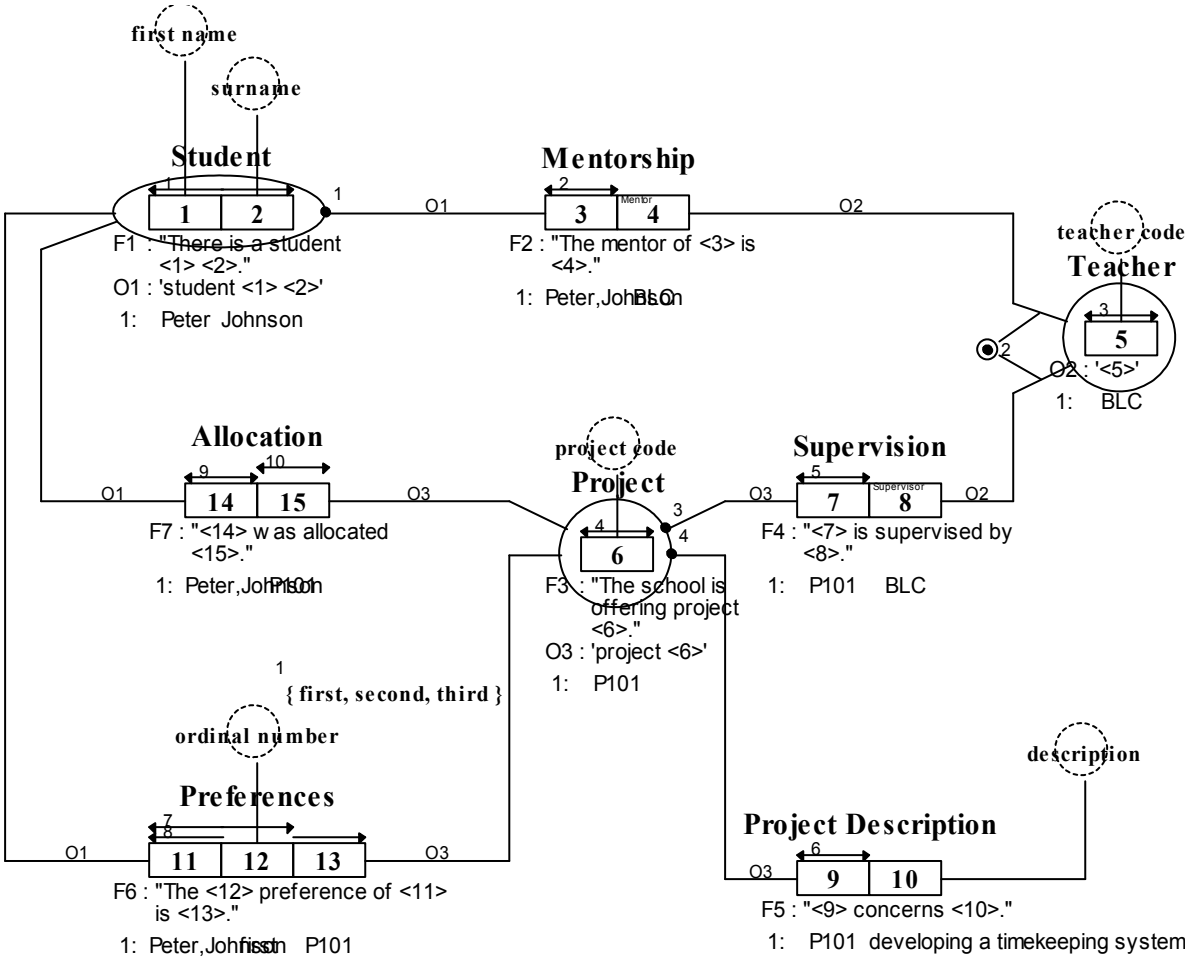
Repositories and transformations - FCO-IM to ERM

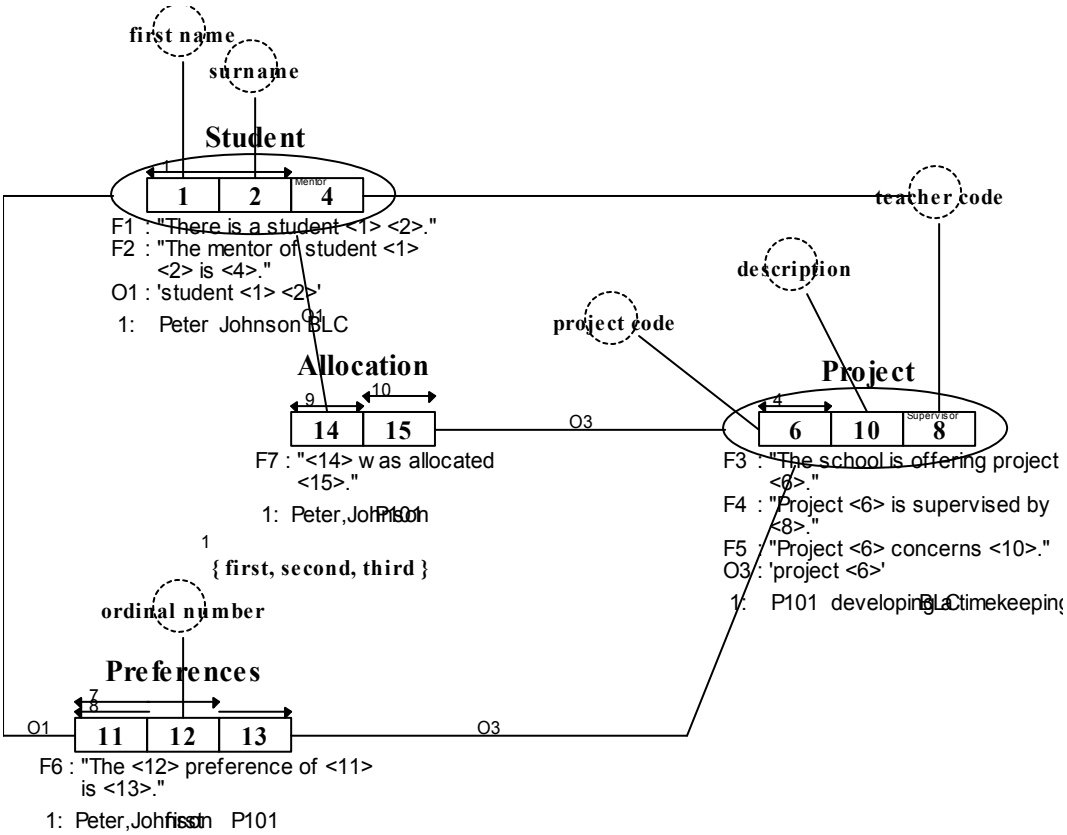
In this material are described:

- The exported repository of FCO-IM grammar
- The repository used to keep data about an Entity Relationship Model
- Queries/Views against the exported repository of the FCO-IM grammar to retrieve the Entity Relationship Model

As an example, is taken the student case, which is found in FCO-IM book.



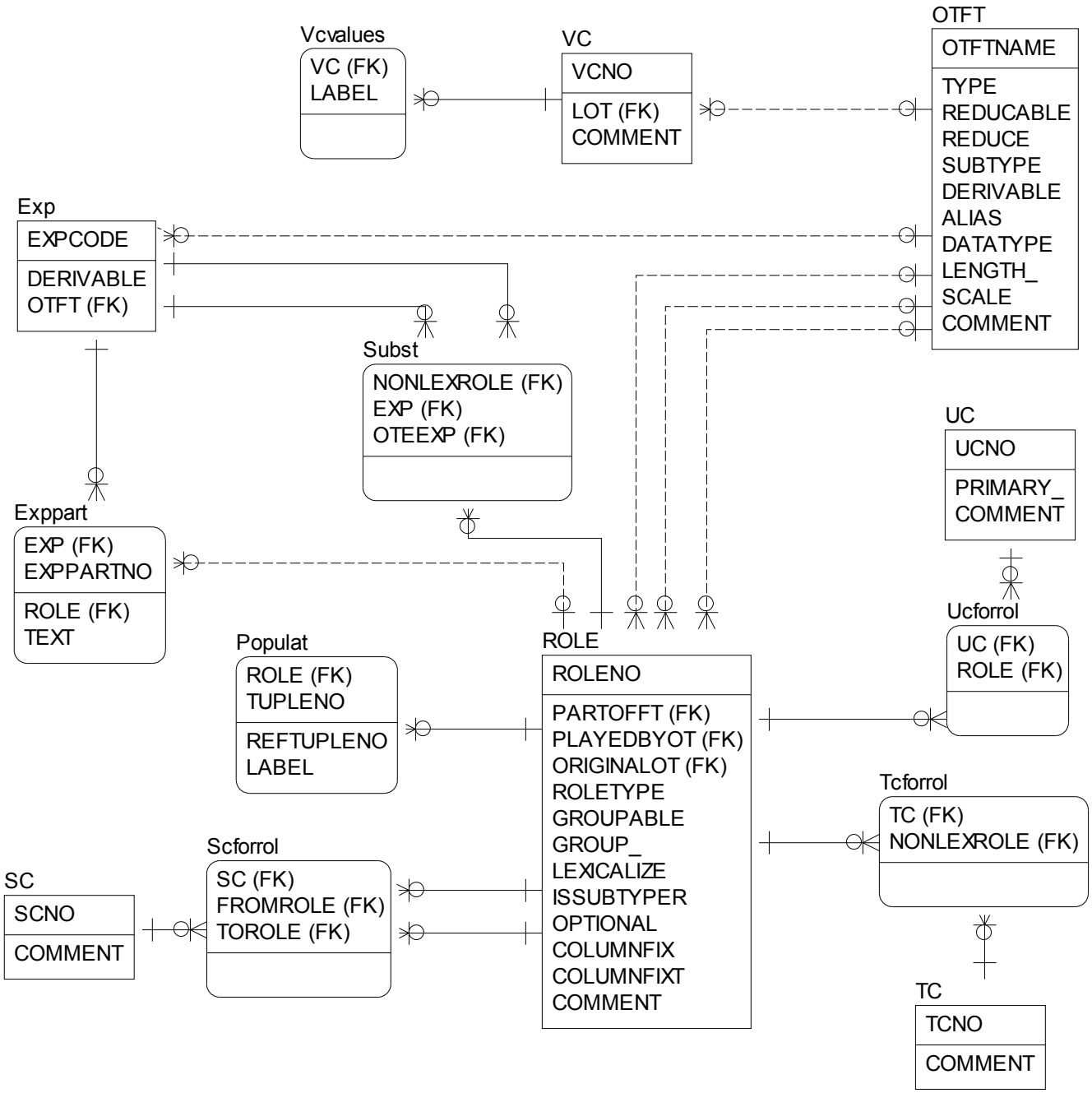
Elementary Information Grammar Diagram



Grouped and Reduced Information Grammar Diagram

1 Tables of the exported FCO-IM repository

In this chapter, the tables of the exported FCO-IM repository are described. There are in total 14 tables. For each table is explained what kind of information they contain, the description for each attribute and a population. In addition, the primary key on the table (written below the attributes that are involved in the primary key) and alternative keys (if any) are given. In the description column is explained the foreign key for the mentioned column if it exists one.



Relational Schema of Exported FCO-IM Repository

1.1 Table OTFT

This table contains information on object type / fact type and there properties.

Name	*Optional?*	*Type*	*Size*	*Description*
OTFTNAME (Primary Key)	No	Text	40	Name of the object type / fact type
ALIAS (Alternate Key)	Yes	Text	40	The alias of fact type/object type < OTFTNAME >
TYPE	No	Text	1	The type of fact type / object type < OTFTNAME > It can be: F = Fact Type, L = Label Type, N = Nominalized Fact Type
DERIVABLE	Yes	Text	3	Specifies if <OTFTNAME> is derivable {YES/NO}
SUBTYPE	Yes	Text	3	Specifies if the fact type <OTFTNAME> is a subtype {YES/NO}
DATATYPE	Yes	Text	40	The data type of lexical object type <OTFTNAME>
LENGTH_	Yes	Text	3	The total length of lexical object type <OTFTNAME>
SCALE	Yes	Text	3	The decimal length of lexical object type <OTFTNAME>
REDUCABLE	Yes	Text	3	Specifies if nominalized fact type < OTFTNAME > could be reduced {YES/NO}
REDUCE	Yes	Text	3	Specifies if reducable nominalized fact type object type <OTFTNAME> must be reduced {YES/NO}
COMMENT	Yes	Text	254	

Population Sample

OTFTNAME	ALIAS	TYPE	DERIVABLE	SUBTYPE	DATATYPE	LENGTH_	SCALE	REDUCABLE	REDUCE	COMMENT
Description		L			char	44	0			
first name		L			char	5	0			
Ordinal number		L			char	6	0			
Preference		F	No	No		0	0	No		
Project		N	No	No		0	0	No		
Project code		L			char	4	0			
Student		N	No	No		0	0	No		
Surname		L			char	7	0			
Teacher code		L			char	3	0			

Note: The attribute COMMENT is not shown by using the Repository Rollup of the FCO-IM Casetool, but it can be populated by using the Edit option of the IG Tasks button of the FCO-IM Casetool and the stored comment is then exported as well. The same holds with respect to the COMMENT attribute of the repository tables ROLE, UC, TC, SC and VC.

1.2 Table ROLE

This table contains information about roles and their properties.

Name	*Optional?*	*Type*	*Size*	*Description*
ROLENO (Primary Key)	No	Text	11	Number of the role
PARTOFFT	No	Text	40	The fact type to which the role <ROLENO> belongs. ***Foreign key referencing the Object Type/Fact type <OTFTNAME>in the table OTFT***
PLAYEDBYOT	No	Text	40	Object type which plays the role <ROLENO>. ***Foreign key referencing the Object Type <OTFTNAME> in the table OTFT***
COLUMNFIX	Yes	Text	40	The column fix which is added to role <ROLENO>
COLUMNFIXT	Yes	Text	7	The type of column fix added to role <ROLENO>. {Postfix/Prefix/Replacefix}
ORIGINALOT	Yes	Text	40	Non-lexical role <ROLENO> was originally played by object type <ORIGINALOT>. ***Foreign key referencing the Object Type <OTFTNAME> in the table OTFT***
OPTIONAL	No	Text	3	Specifies if role <ROLENO> is optional
GROUPABLE	Yes	Text	3	Specifies if the non-lexical role <ROLENO> could be grouped
GROUP_	Yes	Text	3	Specifies if groupable non-lexical role <ROLENO> must be grouped {YES/NO}
LEXICALIZE	Yes	Text	3	Specifies if non-lexical role <ROLENO> must be lexicalized
COMMENT	Yes	Text	254	
ISSUBTYPER	Yes	Text	3	Specifies if non-lexical role <ROLENO> is a subtype role {YES/NO}

Population sample

ROLENO	PARTOFFT	PLAYEDBYOT	COLUMNFIX	COLUMNFIXT	ORIGINALOT	OPTIONAL	GROUPABLE	GROUP_	LEXICALIZE	COMMENT	ISSUBTYPER
1	Student	First name			first name	No	No				
2	Student	Surname			surname	No	No				
5	Student	Teacher code	Mentor	Replace	Teacher	No	No				No
6	Project	Project code			project code	No	No				
8	Project	Teacher code			Teacher	No	No				No
10	Project	Description			description	No	No				No
11	Preference	Student	Student	Prefix	Student	No	No				
12	Prefer	Ordinal			ordinal number	No	No				

ROLE NO	PARTOFFT	PLAYEDBYOT	COLUMNFIX	COLUMNFIXT	ORIGINALOT	OPTIONAL	GROUPABLE	GROUP_	LEXICALIZE	COMMENT	ISSUBTYPER
	ence	number									
13	Preference	Project			Project	No	No				
15	Student	Project			Project	Yes	No				No

1.3 Table EXP

This table contains information about fact type / object type expressions

Name	*Optional?*	*Type*	*Size*	*Description*
EXPCODE (Primary Key)	No	Text	6	The code of the type expression The format is {F,O}+<NUMBER>
OTFT	Yes	Text	40	Fact type / Non-lexical object type to which fact type expression belongs. ***Foreign key referencing the Non-Lexical Object Type/Fact type <OTFTNAME>in the table OTFT***
DERIVABLE	Yes	Text	3	Specifies if type expression <EXPCODE> is derivable {YES/NO}

Population sample

EXPCODE	OTFT	DERIVABLE
F1	Student	
F2	Student	
F3	Project	
F4	Project	
F5	Project	
F6	Preference	
F7	Student	
O1	Student	
O3	Project	

1.4 Table EXPPART

This table contains information about expression parts and their properties

Name	*Optional?*	*Type*	*Size*	*Description*
EXP (Primary Key)	No	Text	6	The expression to which the expression part <EXPPARTNO> belongs. The format is {F,O}+<NUMBER> ***Foreign key referencing the expression <EXPCODE>***

				in the table EXP.
EXPPARTNO (Primary Key)	No	Text	3	Number of expression part
ROLE	Yes	Text	11	Role <ROLE> is part of expression <EXP> as expression part <EXPPARTNO> ***Foreign key referencing the Role <ROLENO> in the table ROLE.***
TEXT	Yes	Text	80	The <TEXT> is part of type expression <EXP> as expression part < EXPPARTNO >

Population example

EXP	EXPPARTNO	ROLE	TEXT
O1	1		Student
O1	2	1	
O1	3		
O1	4	2	
O3	1		Project
O3	2	6	
F1	1		There is a student
F1	2	1	
F1	3		
F1	4	2	
F2	1		The mentor of student
F2	2	1	
F2	3		
F2	4	2	
F2	5		Is
F2	6	5	
F3	1		The school is offering project
F3	2	6	
F4	1		Project
F4	2	6	
F4	3		Is supervised by
F4	4	8	
F5	1		Project
F5	2	6	
F5	3		Concerns
F5	4	10	
F6	1		The
F6	2	12	
F6	3		Preference of
F6	4	11	
F6	5		Is
F6	6	13	
F7	1		Student
F7	2	1	
F7	3		
F7	4	2	
F7	5		Was allocated

F7	6	15	

1.5 Table SUBST

This table contains information on substitutions and their properties

Name	*Optional?*	*Type*	*Size*	*Description*
EXP (Primary Key)	No	Text	6	The identifier for expression ***Foreign key referencing the expression <EXPCODE> in the table EXP***
NONLEXROLE (Primary Key)	No	Text	11	The role for which in expression <EXP> , object type expression <OTEEXP> must be filled in ***Foreign key referencing role <ROLENO> in the table ROLE***
OTEEXP (Primary Key)	No	Text	6	The identifier for object type expression ***Foreign key referencing the object type expression <EXPCODE> in the table EXP***

Population example

EXP	NONLEXROLE	OTEEXP
F6	11	O1
F6	13	O3
F7	15	O3

1.6 Table UC

This table contains information about uniqueness constraints and their properties.

Name	*Optional?*	*Type*	*Size*	*Description*
UCNO (Primary Key)	No	Text	5	The identifier of uniqueness constraint
PRIMARY_	Yes	Text	3	Specifies if uniqueness constraint is a primary key {YES/NO}
COMMENT	Yes	Text	254	

Note: In the repository table UC that is implemented in the FCO-IM Casetool there is an attribute named IntraUC that doesn't exist in the exported table UC.

Population example

UCNO	PRIMARY_	COMMENT
1	No	
4	No	
7	Yes	
8	No	
10	No	

1.7 Table UCFORROL

This table contains information on roles to which uniqueness constraints belong

Name	*Optional?*	*Type*	*Size*	*Description*
UC (Primary Key)	No	Text	5	Identifier of uniqueness constraint ***Foreign key referencing the uniqueness constraint <UCNO> in the table UC***
ROLE (Primary Key)	No	Text	11	Role covered from uniqueness constraint <UC> ***Foreign key referencing the role <ROLENO> in the table ROLE***

Population example

UC	ROLE
1	1
1	2
4	6
7	11
7	12
8	11
8	13
10	15

1.8 Table TC

This table contains information on totally constraints

Name	*Optional?*	*Type*	*Size*	*description*
TCNO (Primary Key)	No	Text	5	The identifier of the totally constraint
COMMENT	Yes	Text	254	

Population example

TCNO	COMMENT

1.9 Table TCFORROL

This table contains information on relation between totally constraints and non-lexical roles

Name	*Optional?*	*Type*	*Size*	*Description*
TC (Primary Key)	No	Text	5	The identifier of totally constraint ***Foreign key referencing the totally constraint <TCNO> in the table TC***
NONLEXROLE (Primary Key)	No	Text	11	Role concerned by totally constraint <TC> ***Foreign key referencing the role <ROLENO> in the table ROLE***

Population example

TC	NONLEXROL

1.10 Table SC

This table contains information on subset constraints

Name	*Optional?*	*Type*	*Size*	*description*
SCNO (Primary Key)	No	Text	5	The identifier of the subset constraint
COMMENT	Yes	Text	254	

Population example

SCNO	COMMENT

1.11 Table SCFORROL

This table contains information on relation between totally constraints and non-lexical roles

Name	*Optional?*	*Type*	*Size*	*Description*
SC (Primary Key)	No	Text	5	The identifier of subset constraint ***Foreign key referencing the subset constraint <SCNO> in the table TC***
FROMROLE (Primary Key)	No	Text	11	The role from where subset constraint <SC> comes ***Foreign key referencing the role <ROLENO> in the table ROLE***
TOROLE (Primary Key)	No	Text	11	The role where subset constraint <SC> goes ***Foreign key referencing the role <ROLENO> in***

				the table ROLE

Population example

SC	FROMROLE	TOROLE

1.12 Table VC

This table contains information on value constraints and their properties

Name	*Optional?*	*Type*	*Size*	*Description*
VCNO (Primary Key)	No	Text	5	The identifier of value constraint
LOT (Alternate key)	Yes	Text	40	The lexical object type to which the value constraint <VCNO> restricts the labels ***Foreign key referencing the lexical object type <OTFTNAME> in the table OTFT***
COMMENT	Yes	Text	254	

Population example

VCNO	LOT	COMMENT
1	ordinal	

1.13 Table VCVALUES

This table contains information on relation value constraint and permitted labels

Name	*Optional?*	*Type*	*Size*	*Description*
VC (Primary Key)	No	Text	5	The identifier of the value constraint ***Foreign key referencing the value constraint <VCNO> in the table VC***
LABEL (Primary Key)	No	Text	80	Permitted labels in the value constraint <VC>

Population example

VC	LABEL
1	first
1	second
1	third

1.14 Table POPULAT

This table contains information about the population (tuples or rows)

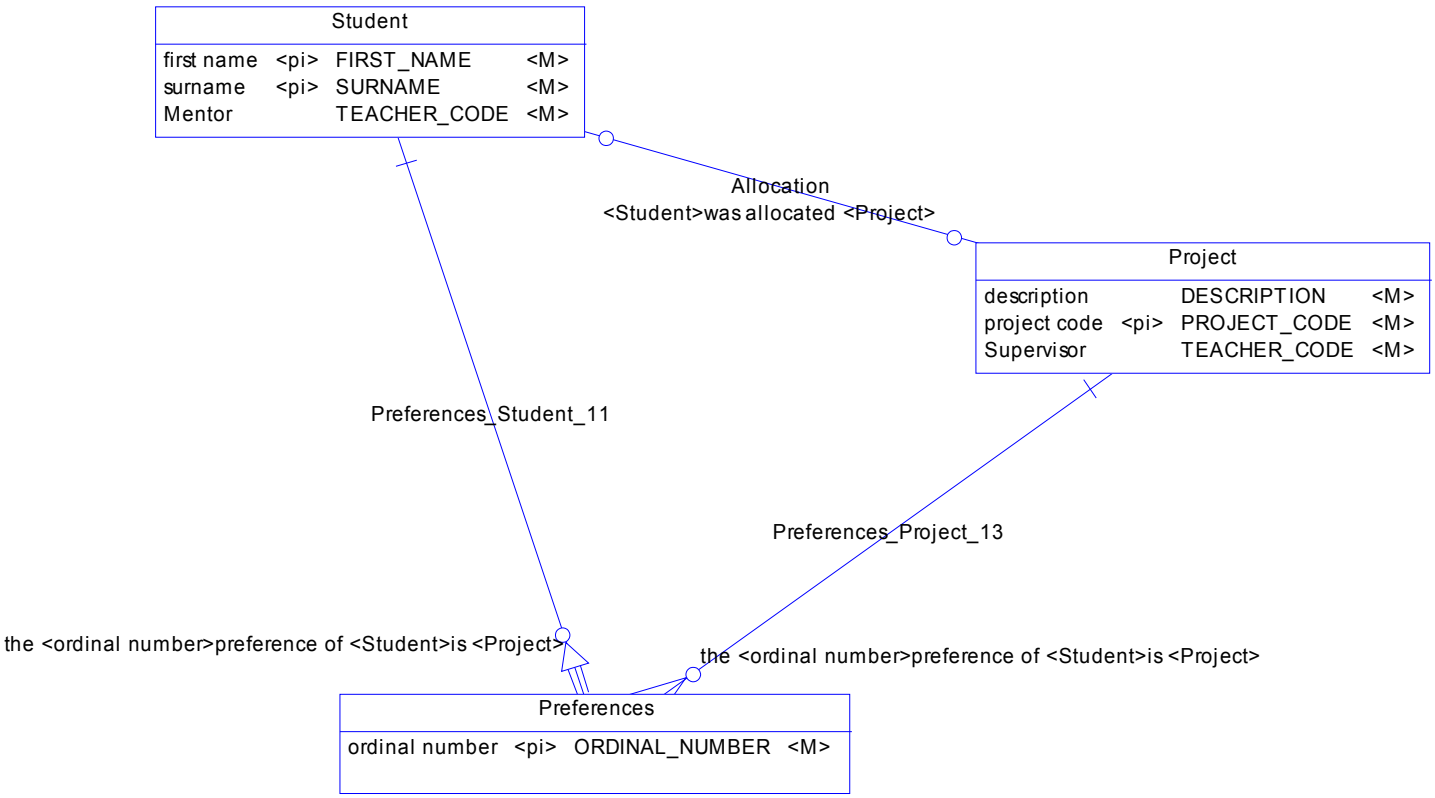
Name	*Optional?*	*Type*	*Size*	*Description*

ROLE (Primary Key)	No	Text	11	Identifier for the populated role ***Foreign key referencing the role <ROLENO> in the table ROLE***
TUPLENO (Primary Key)	No	Text	5	Position where role <ROLE> is populated
REFTUPLENO	Yes	Text	5	The tuple where non-lexical role population of role <ROLE> at position <TUPLENO> is referenced
LABEL	Yes	Text	80	The label of the lexical role population of role <ROLE> at position <TUPLENO>

Population example

ROLE	TUPLENO	REFTUPLENO	LABEL
1	1		Peter
1	2		Fred
1	3		John
1	4		Elsa
1	5		Maria
2	1		Johnson
2	2		Smith
2	3		Hartman
2	4		Doyle
2	5		Jones
5	1		BLC
5	2		GPB
5	3		BLC
5	4		BAK
5	5		VRM
6	1		P101
6	2		P204
6	3		P110
6	4		P120
6	5		P200
6	6		P203
8	1		BLC
8	2		FEL
8	3		LEK
8	4		FEL
8	5		BAK
8	6		BAK
10	1		developing a timekeeping system
10	2		implementing a design for a database
10	3		introducing an RDBMS into a business company
10	4		converting a dBASE system to Foxpro
10	5		developing a technical information system
10	6		writing course material on FCO-IM
11	1	1	
11	2	3	
11	3	1	
11	4	1	

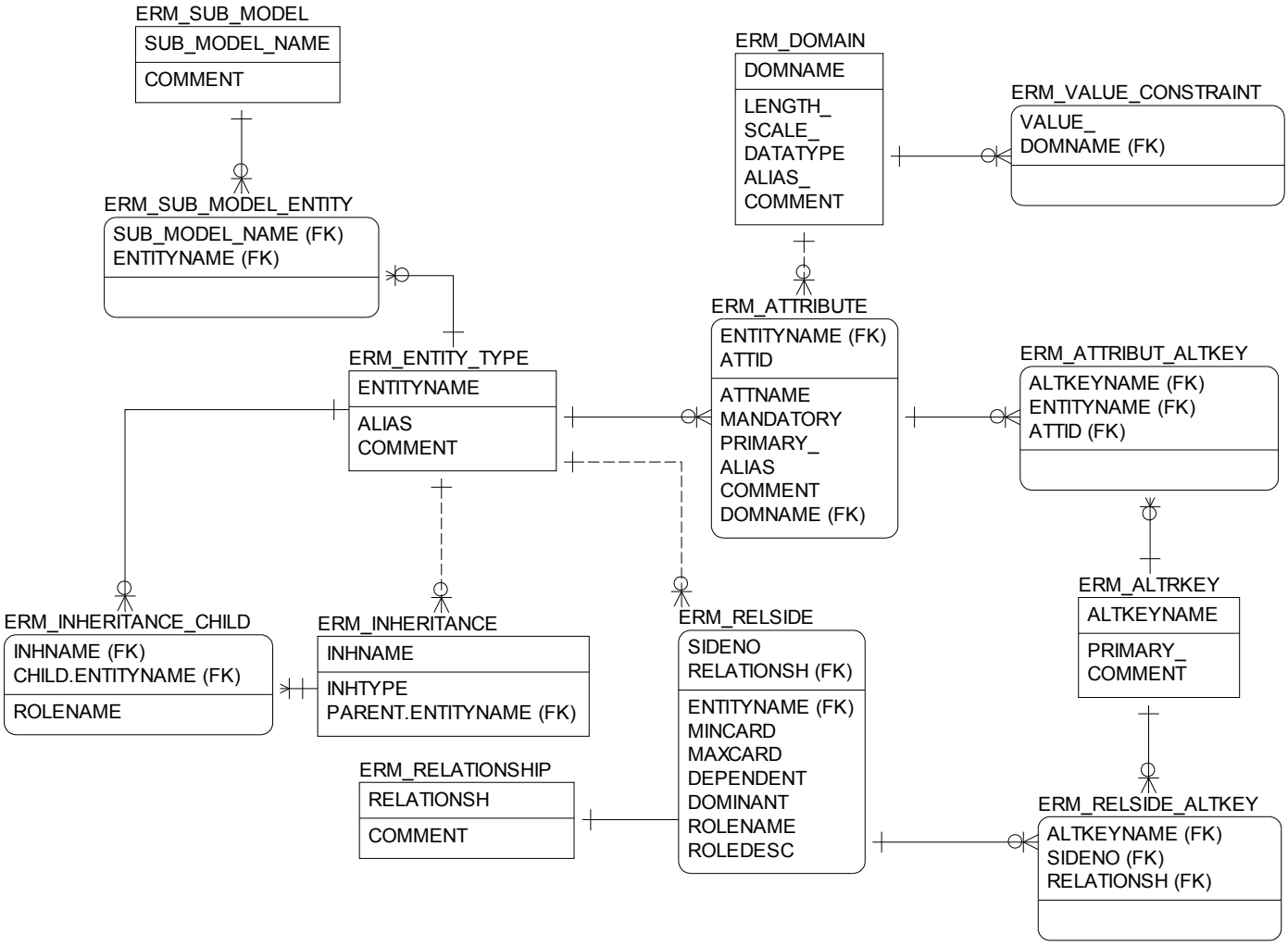
11	5	3	
11	6	3	
12	1		first
12	2		third
12	3		second
12	4		third
12	5		first
12	6		second
13	1	1	
13	2	5	
13	3	3	
13	4	4	
13	5	6	
13	6	1	
15	1	1	
15	3	6	



Entity Relationship Model in IE Notation – PowerDesigner 9

2 Tables of ERM repository

In this chapter is given database structure of ERM repository. There are in total 9 stables. \
For each table is explained what kind of information they contain, the description for each attribute and a population. In addition, the primary key on the table (written below the attributes that are involved in the primary key) and alternative keys (if any) are given. In the description column is explained the foreign key for the mentioned column if it exists one.



Relational schema of ERM model

2.1 Table ERM_ENTITY_TYPE

This table contains information about entities

Name	*Optional?*	*Type*	*Size*	*Description*
ENTITYNAME (Primary Key)	No	Text	80	Name of the entity
ALIAS	Yes	Text	80	The alias of the entity <ENTITYNAME>
COMMENT	Yes	Memo		Note of the entity <ENTITYNAME>

Population sample

ENTITYNAME	ALIAS	COMMENT
Preferences		
Project		
Student		

2.2 Table ERM_DOMAIN

This table contains information about domains and their properties.

Name	*Optional?*	*Type*	*Size*	*Description*
DOMNAME (Primary Key)	No	Text	80	Name of the domain
LENGTH_	No	Text	3	Total length of the domain <DOMNAME>
SCALE_	No	Text	3	Decimal length of the domain <DOMNAME>
DATATYPE	No	Text	40	Data type of the domain <DOMNAME>
ALIAS_	Yes	Text	80	Alias of the domain <DOMNAME>
COMMENT	Yes	Memo		Note of the domain <DOMNAME>

Population sample

DOMNAME	LENGTH_	SCALE_	DATATYPE	ALIAS_	COMMENT
description	31	0	char		
first name	5	0	char		
ordinal number	5	0	char		
project code	4	0	char		
surname	7	0	char		
teacher code	3	0	char		

2.3 Table ERM_VALUE_CONSTRAINT

This table contains information on value constraints

Name	*Optional?*	*Type*	*Size*	*Description*
DOMNAME	No	Text	80	Domain name

(Primary Key)				***Foreign key referencing the domain <DOMNAME> in the table ERM_DOMAIN***
VALUE_ (Primary Key)	No	Text	40	Value constraint of the domain <DOMNAME>

Population example

DOMNAME	VALUE_
Ordinal number	first
Ordinal number	second
Ordinal number	third

2.4 Table ERM_ATTRIBUTE

This table contains information on attributes and their properties.

Name	*Optional?*	*Type*	*Size*	*Description*
ATTID (Primary Key)	No	Text	5	Attribute identifier
ENTITYNAME (Primary Key)	No	Text	80	Entity name of which the attribute <ATTNAME> is part ***Foreign key referencing the entity name <ENTITYNAME> in the table ERM_ENTITY_TYPE***
ATTNAME	No	Text	80	Name of the attribute
DOMNAME	No	Text	80	Domain of the attribute <ATTNAME> ***Foreign key referencing the domain <DOMNAME> in the table ERM_DOMAIN***
PRIMARY_	No	Text	3	Specifies if the attribute <ATTNAME> could be part of the primary identifier {YES/NO} This is a derived value from the fact that attribute is part of an alternative key, which is primary key
MANDATORY	No	Text	3	Attribute <ATTNAME> could be mandatory {YES/NO}
ALIAS	Yes	Text	80	The alias of attribute <ATTNAME>
COMMENT	Yes	Memo		Note about attribute <ATTNAME>

Population Sample

ATTID	ENTITYNAME	ATTNAME	DOMNAME	MANDATORY	PRIMARY_	ALIAS	COMMENT
12	Preferences	ordinal number	ordinal number	Yes	Yes		the <ordinal number>preference of <Student>is <Project>
10	Project	description	description	Yes	No		project <project code>concerns <description>
6	Project	project code	project code	Yes	Yes		the school is offering project <project code>
8	Project	Supervisor	teacher code	Yes	No		project <project code>is supervised by <teacher code>
1	Student	first name	first name	Yes	Yes		there is a student <first name> <surname>
2	Student	surname	surname	Yes	Yes		there is a student <first name> <surname>
4	Student	Mentor	teacher code	Yes	No		the mentor of student <first name> <surname>is <teacher code>

2.5 Table ERM_RELATIONSHIP

This table contains information on relationships.

Name	*Optional?*	*Type*	*Size*	*Description*
RELATIONSH (Primary Key)	No	Text	150	Name of the relationship
COMMENT	Yes	Memo		Note of the relationship

Population example

RELATIONSH	COMMENT
Allocation	
Preferences_Project_13	
Preferences_Student_11	

2.6 Table ERM_RELSIDE

This table contains information about the entities involved in a relationship or sides of a relationship. For one relationship there are two rows for each side of it. See population example.

Name	*Optional?*	*Type*	*Size*	*Description*
RELATIONSH (Primary Key)	No	Text	150	Relationship name ***Foreign key referencing the relationship <RELATIONSH> in the table ERM_RELATIONSHIP***
SIDENO (Primary Key)	No	Text	1	The side number. Identifies the side of relationship <RELATIONSH>. Value constraint for this column is {1,2}
ENTITYNAME	No	Text	80	Entity name connected in the side number <SIDENO> in the relationship <RELATIONSH> ***Foreign key referencing the entity <ENTITYNAME> in the table ERM_ENTITY_TYPE***
MINCARD	No	Text	1	Minimum of cardinality in the side <SIDENO> of relationship <RELATIONSH>
MAXCARD	No	Text	1	Maximum of cardinality in the side <SIDENO> of relationship <RELATIONSH>
DEPENDENT	No	Text	3	Specifies if relationship in side <SIDENO> could be dependent {YES/NO} This is a derived value from the fact that the relationship side is part of an alternative key, which is primary key
DOMINANT	No	Text	3	Specifies if relationship in side <SIDENO> could be dominant {YES/NO}
ROLEDESC	Yes	Text	211	Description on side <SIDENO> of relationship <RELATIONSH>

Population example

RELATIONSH	SIDENO	ENTITYNAME	MINCARD	MAXCARD	DEPENDENT	DOMINANT	ROLEDESC
Allocation	1	Project	0	1	No	No	<Student>was allocated <Project>
Allocation	2	Student	0	1	No	No	
Preferences_Project_13	1	Preferences	0	n	No	No	the <ordinal number>preference of

RELATIONSH	SIDENO	ENTITYNAME	MINCARD	MAXCARD	DEPENDENT	DOMINANT	ROLEDESC
							<Student>is <Project>
Preferences_Project_13	2	Project	1	1	No	No	
Preferences_Student_11	1	Preferences	0	n	Yes	No	the <ordinal number>preference of <Student>is <Project>
Preferences_Student_11	2	Student	1	1	No	No	

2.7 Table ERM_INHERITANCE

This table contains information about inheritances

Name	*Optional?*	*Type*	*Size*	*Description*
INHNAME (Primary Key)	No	Text	150	Name of the inheritance
PARENT	No	Text	80	Name of the entity parent ***Foreign key referencing the entity parent <ENTITYNAME> in table ERM_ENTITY_TYPE***
INHTYPE	No	Text	3	The type of inheritance. INC – Inclusive EXC – Exclusive

Population example

INHNAME	PARENT	INHTYPE

2.8 Table ERM_INHERITANCE_CHILD

This table contains information about relation parent-child in inheritances.

Name	*Optional?*	*Type*	*Size*	*Description*
CHILD (Primary Key)	No	Text	80	Name of the entity child ***Foreign key referencing the entity child < ENTITYNAME> in table ERM_ENTITY_TYPE***
INHNAME (Primary Key)	No	Text	150	The inheritance where the entity child <CHILD> and <PARENT> are involved ***Foreign key referencing the inheritance <INHNAME> in table ERM_INHERITANCE***

Population example

CHILD	INHNAME

2.9 Table ERM_ALTRKEY

This table contains information about Alternative keys that are not Primary Key.

Name	*Optional?*	*Type*	*Size*	*Description*
ALTKEYNAME (Primary Key)	No	Text	40	Identifier of alternative key
COMMENT	Yes	Memo		Comment about the alternative key
PRIMARY_	No	Text	3	Specifies if the alternative key <ALTKEYNAME> could be primary identifier {YES/NO}

Population example

ALTKEYNAME	PRIMARY_	COMMENT
1	Yes	
4	Yes	
7	Yes	
8	No	

2.10 Table ERM_ATTRIBUT_ALTKEY

This table contains information about the attributes that are part of an alternative key

Name	*Optional?*	*Type*	*Size*	*Description*
ALTKEYNAME (Primary Key)	No	Text	40	Identifier of Alternative Key
ENTITYNAME (Primary Key)	No	Text	80	Entity type where the attribute that is part of alternative key, is found
ATTID (Primary Key)	No	Text	80	Identifier of the attribute that is part of alternative key

Population example

ENTITYNAME	ATTID	ALTKEYNAME
Preferences	12	7
Project	6	4
Student	1	1
Student	2	1

2.11 Table ERM_RELSIDE_ALTKEY

This table contains information about the relationship sides that are part of an alternative key

Name	*Optional?*	*Type*	*Size*	*Description*
ALTKEYNAME (Primary Key)	No	Text	40	Identifier of Alternative Key
SIDENO (Primary Key)	No	Text	1	The side of relationship which is involved in alternative key
RELATIONSH (Primary Key)	No	Text	150	Relationship that is involved in the alternative key

Population example

RELATIONSH	SIDENO	ALTKEYNAME
Preferences_Project_13	1	8
Preferences_Student_11	1	7
Preferences_Student_11	1	8

2.12 Table ERM_SUB_MODEL

This table contains information about the sub-models

Name	*Optional?*	*Type*	*Size*	*Description*
SUB_MODEL_NAME (Primary Key)	No	Text	254	Sub-model name
COMMENT	Yes	Memo		Comment about the sub-model

Population example

SUB_MODEL_NAME	COMMENT
Preferences of student	
Projects in Preferences	

2.13 Table ERM_SUB_MODEL_ENTITY

This table contains information about the sub-models and their entity types

Name	*Optional?*	*Type*	*Size*	*Description*
SUB_MODEL_NAME (Primary Key)	No	Text	254	Sub-model name
ENTITYNAME	No	Text	80	Entity type, which is part of the sub-model

Population example

SUB_MODEL_NAME	ENTITYNAME
Preferences of student	Preferences
Preferences of student	Student
Projects in Preferences	Preferences

SUB_MODEL_NAME	ENTITYNAME
Projects in Preferences	Project

3 Queries used to transform the Objects from the FCO-IM Repository to ERM Repository

Transformations explained below are as the result of applying the algorithm on an FCOIM grammar to ERM. The queries for finding and storing of entities, attributes, domains, value constraints over domains, inheritances, relationships and relationship sides from FCO IM Repository to ERM Repository are shown with the necessary explanations. The algorithm has been split up in pieces in the front of the queries to better understand them.
Queries are built in MS Access 97. For this reason you will see very often, functions that are not SQL standard, but MS Access 97 standard.

3.1 Entities

Tables involved in this transformation are:

3.1.1 FCO-IM Repository

ROLE (to get information on the number of roles involved in fact types)
OTFT (to get the name of the entity)

3.1.2 ERM Repository

ERM_ENTITY_TYPE

3.1.3 Queries

As a name of entity type is taken the Alias or the Fact Type Name.
So entity type will become:

3.1.3.1 Non-nominalized Fact type with number of roles n *(number of roles)*<>2

```
SELECT
OTFT.OTFTNAME,
OTFT.ALIAS,
OTFT.COMMENT
FROM ROLE, OTFT
WHERE ROLE.PARTOFFT = OTFT.OTFTNAME
AND TYPE='F'
GROUP BY OTFT.OTFTNAME, OTFT.ALIAS, OTFT.COMMENT
HAVING Count(*)<>2
```

3.1.3.2 Every non-nominalized binary (number of roles n=2) fact type with at least one role played by a lexical object type (label type) or that is marked as a subtype

```
SELECT
OTFT_1.OTFTNAME AS NAME,
```

```
OTFT_1.ALIAS,
OTFT_1.COMMENT
FROM ROLE, OTFT AS OTFT_1
WHERE ROLE.PARTOFFT = OTFT_1.OTFTNAME
        AND TYPE='F' AND (SUBTYPE='Yes' OR
         (EXISTS (SELECT ROLE.PARTOFFT
                        FROM OTFT AS OTFT_2, ROLE
                        WHERE OTFT_2.OTFTNAME = ROLE.PLAYEDBYOT
                        AND ROLE.PARTOFFT=OTFT_1.OTFTNAME
                        AND OTFT_2.TYPE='L')))
GROUP BY OTFT_1.OTFTNAME, OTFT_1.ALIAS, OTFT_1.COMMENT
HAVING Count(*)=2
```

3.1.3.3 Every non lexical object type

```
SELECT
OTFTNAME AS NAME,
ALIAS,
COMMENT
FROM OTFT
WHERE TYPE="N"
```

3.2 Domain

3.2.1 FCO-IM Repository

OTFT (to get all the information about domains)

3.2.2 ERM Repository

ERM_DOMAIN

3.2.3 Queries

Every lexical object type (label type) is considered as a domain.

```
SELECT
OTFTNAME AS DOMNAME,
DATATYPE,
LENGTH_,
SCALE,
ALIAS
FROM OTFT
WHERE TYPE= 'L'
```

3.3 Value constraints

3.3.1 FCO-IM Repository

OTFT (to get the name of the domain)
VC (to check if the lexical object type (Label type) or the domain has a value constraint)
VCVALUES (to get the value constraint values of the domain found)

3.3.2 ERM Repository

ERM_VALUE_CONSTRAINT

3.3.3 Queries

For the lexical object types (label types) taken as domains the domain value constraints are found in the VCVALUES table

SELECT
VC.LOT AS DOMNAME,
VCVALUES.LABEL AS VALUE_
FROM VC, VCVALUES
WHERE VC.VCNO= VCVALUES.VC

3.4 Attributes

Tables involved in this transformation are:

3.4.1 FCO-IM Repository

ROLE (to get information about the entity type, column fix type (prefix, replacefix, postfix), if is mandatory and which played it originally)
OTFT(to get information about the name of attribute (OTFTNAME), the name of the domain, ALIAS, COMMENT)

3.4.2 ERM Repository

ERM_ATTRIBUTE

3.4.3 Queries to get attributes

For the role that is part of OTFT loaded as entity type, the name of the Lexical Object Type (LOT) playing that role with prefix, replace fix or postfix if there is one, is used as an attribute name of the entity type which contains that role.
The name of the LOT playing that role is used as a domain of that attribute.
If the lexical role is optional then the attribute is Mandatory.

SELECT DISTINCT
ROLE.ROLENO AS ATTID,
ROLE.PARTOFFT AS ENTITY,
IIf(COLUMNFIXT='Replace',COLUMNFIX, IIf(COLUMNFIXT='Prefix',(COLUMNFIX + '_' +
ROLE.ORIGINALOT), IIf(COLUMNFIXT='Postfix', ROLE.ORIGINALOT + '_' + COLUMNFIX,
ROLE.ORIGINALOT))) AS ATTRIBUTE_NAMED_BY_ORIGINALOT,
OTFT.OTFTNAME AS [DOMAIN],
IIf(OPTIONAL='No', 'Yes', 'No') AS MANDATORY,
ROLE.COMMENT
FROM OTFT, ROLE
WHERE OTFT.OTFTNAME = ROLE.PLAYEDBYOT
AND OTFT.TYPE ='L'

Note: If original object type that plays the role, is not participating in the name of attribute then we must replace the part where the name of attribute is generated by

IIf(COLUMNFIXT='Replace',COLUMNFIX,IIf(COLUMNFIXT='Prefix',(COLUMNFIX & '_' &
OTFTNAME),IIf(COLUMNFIXT='Postfix',OTFTNAME & '_' & COLUMNFIX,OTFTNAME))) AS
ATTRIBUTE

3.4.4 Adding description for the attributes

Query below is used to retrieve information on role description for the sides of relationships. Role descriptions are fact type expressions where the role involved in relationship is part of them. This role is given as parameter ROLEPARAM. After, this record set must be navigated iteratively in order to get the sentence.

```
SELECT EXP,
EXPPARTNO,
ROLE,
PLAYEDBYOT,
[TEXT] AS TEXT_
FROM EXPPART LEFT JOIN ROLE ON EXPPART.ROLE= ROLE.ROLENO
WHERE EXP IN
        (SELECT MIN(EXP)
        FROM EXPPART
        WHERE ROLE=ROLEPARAM)
ORDER BY 2
```

3.5 Inheritance

3.5.1 FCO-IM Repository

ROLE (It is used to get the role number, OTFT that plays the role. This OTFT will be the future parent and OTFT where the role is will be the future child)
TCFORROL (It is used to identify the Inheritance)

3.5.2 ERM Repository

ERM_INHERITANCE (It is used to store the information about the name, parent entity and the type of inheritance)
ERM_INHERITANCE CHILD (It is used to store the information about the children for a certain inheritance)

3.5.3 Queries

The inheritances are taken out of subtypes. For each subtype an inheritance is added. The name of inheritance is <super type>_<totality constraint if exists which covers the role played by super type>. For each inheritance, the super type name is the name of entity type parent and the subtype name is the entity child.

3.5.3.1 This query is used to identify the name of inheritances and to put this information in the ERM_INHERITANCE table. The type of inheritance is by default EXCLUSIVE,

```
SELECT
(PLAYEDBYOT + '_' + IIF(ISNULL(TC),'',TC)) AS INHNAME,
'EXC' AS INHTYPE,
PLAYEDBYOT AS PARENT
FROM ROLE LEFT JOIN TCFORROL ON ROLE.ROLENO=TCFORROL.NONLEXROLE
WHERE ISSUBTYPER= 'Yes'
```

3.5.3.2 This query is used to identify the entity parent and entity child of a certain inheritance and to put this information in ERM_INHERITANCE_CHILD table.

SELECT
(PLAYEDBYOT + '_' + IIF(ISNULL(TC),'',TC)) AS INHNAME,
PARTOFFT AS CHILD
FROM ROLE LEFT JOIN TCFORROL ON ROLE.ROLENO=TCFORROL.NONLEXROLE
WHERE ISSUBTYPER= 'Yes'

3.6 Relationships

3.6.1 FCO-IM Repository

ROLE (in this table is taken the information on the role number and is used to check if the role is played by a supper type)
OTFT (it is used to get the information on the nominalized OTFT which plays the role)
UCFORROL (It is used to find out if the role is part of a UC and with how many other roles)
TCFORROL (It is used to find out if the role is part of a TC and with how many other roles)
UC (it is used to check if the role is covered by a primary UC)

3.6.2 ERM Repository

ERM_RELATIONSHIP
ERM_RELSIDE

3.6.3 Queries

There are two kind of relationships defined from the algorithm.

3.6.3.1 Relationships coming from non-lexical roles

The roles, which the relationships are coming from, must fulfill the conditions:

- Not to be played by a super type. These relationships are inheritance relationships.
- The role must be played by an non-lexical object type
- The role must not be part of a binary non-nominalized fact type that has only non-lexical role
- The name of this kind of relationship is <OTFTNAME of fact type which plays the role>_ <OTFTNAME of fact type where the role is found>_ROLENO

Side number 1 comes from the fact type where the role is found.

- Minimum cardinality is 1 if there is a single role TC, otherwise is 0
- Maximum cardinality is 1 if there is a one role UC for that role, otherwise is n

Side number 2 comes from the non-lexical object type which plays the role.

- Minimum cardinality is 0 if the role is optional, otherwise is 1
- Maximum cardinality of this side is 1

If the role is covered by a UC and no other role is covered by this UC, then this relationship side is marked as dominant, otherwise no

SELECT
ROLE.PARTOFFT + '_' + ROLE.PLAYEDBYOT + '_' + ROLE.ROLENO AS RELNAME,
ROLE.PARTOFFT AS ENTITY1,
IIF((**SELECT** COUNT(*) **FROM** TCFORROL AS TCF **WHERE** TCF.TC=TCFORROL.TC)=1, '1', '0') AS MINIMUMCARD1,
IIF((**SELECT** COUNT(*) **FROM** UCFORROL AS UCF **WHERE** UCF.UC=UCFORROL.UC)=1,'1','n') AS MAXIMUMCARD1,
'No' AS DOMINANT1,
ROLE.PLAYEDBYOT AS ENTITY2,
IIF(OPTIONAL='Yes','0','1') AS MINIMUMCARD2,
'1' AS MAXIMUMCARD2,

```
IIF((SELECT COUNT(*) FROM UCFORROL AS UCF WHERE UCF.UC=UCFORROL.UC)=1,'Yes','No')
AS DOMINANT2,
ROLE.COMMENT
FROM ((ROLE INNER JOIN OTFT ON ROLE.PLAYEDBYOT = OTFT.OTFTNAME) LEFT JOIN
TCFORROL ON TCFORROL.NONLEXROLE=ROLE.ROLENO) LEFT JOIN UCFORROL ON
UCFORROL.ROLE=ROLE.ROLENO
WHERE (ISSUBTYPER='No' OR ISSUBTYPER IS NULL)
AND OTFT.TYPE='N'
AND PARTOFFT NOT IN
      (SELECT PARTOFFT
       FROM ROLE, OTFT
       WHERE ROLE.PARTOFFT= OTFT.OTFTNAME
       AND OTFT.TYPE='F'
       AND OTFT.OTFTNAME NOT IN
            (SELECT PARTOFFT
             FROM ROLE, OTFT
             WHERE ROLE.PLAYEDBYOT= OTFT.OTFTNAME AND OTFT.TYPE='L')
       GROUP BY PARTOFFT
       HAVING COUNT(*)=2)
```

3.6.3.2 Relationships coming from fact types

These fact types couldn't become entity type. They are non-nominalized binary fact types with both non-lexical roles.
The name of this kind of relationship is OTFTNAME of the fact type itself.

- Minimum cardinality in the side of the entity type coming from the non-lexical object type which plays the role is 1 if there is a single role TC on the other role, otherwise 0
- Maximum cardinality in the side of the entity type coming from the non-lexical object type which plays the role is 1 if there is a single role UC on the other role, otherwise n

```
SELECT ROLE.ROLENO,
ROLE.PARTOFFT AS RELNAME,
ROLE.PLAYEDBYOT AS ENTITY,
(SELECT COUNT(*) FROM TCFORROL AS TCF WHERE TCF.TC=TC_1.TC) AS TCNO,
(SELECT COUNT(*) FROM UCFORROL AS UCF WHERE UCF.UC=UCF_1.UC) AS UCNO,
(SELECT COMMENT FROM OTFT WHERE OTFT.OTFTNAME= ROLE.PARTOFFT) AS COMMENT
FROM  ((ROLE INNER JOIN OTFT ON ROLE.PLAYEDBYOT = OTFT.OTFTNAME)
LEFT JOIN UCFORROL AS UCF_1 ON UCF_1.ROLE=ROLE.ROLENO)
LEFT JOIN TCFORROL AS TC_1 ON TC_1.NONLEXROLE=ROLE.ROLENO
WHERE (ROLE.ISSUBTYPER='NO' OR ROLE.ISSUBTYPER IS NULL)
AND OTFT.TYPE='N'
AND ROLE.PARTOFFT IN
        (SELECT PARTOFFT
        FROM ROLE, OTFT
        WHERE ROLE.PARTOFFT= OTFT.OTFTNAME
        AND OTFT.TYPE='F'
        AND OTFT.OTFTNAME NOT IN
                (SELECT PARTOFFT
                FROM ROLE, OTFT
                WHERE ROLE.PLAYEDBYOT= OTFT.OTFTNAME AND OTFT.TYPE='L' )
        GROUP BY PARTOFFT
        HAVING COUNT(*)=2)
```

3.6.3.3 Adding description on the side of the relationship

Query below is used to retrieve information on role description for the sides of relationships. Role descriptions are fact type expressions where the role involved in relationship is part of them. This role is given as parameter ROLEPARAM. After, this record set must be navigated iteratively in order to get the sentence.

```
SELECT EXP,
EXPPARTNO,
ROLE,
PLAYEDBYOT,
[TEXT] AS TEXT_
FROM EXPPART LEFT JOIN ROLE ON EXPPART.ROLE= ROLE.ROLENO
WHERE EXP IN
        (SELECT MIN(EXP)
        FROM EXPPART
        WHERE ROLE=ROLEPARAM)
ORDER BY 2
```

3.7 Alternative Keys

An alternative key can be a combination of attributes and relationship sides. Are excluded primary keys. Primary keys are seen to be more part of the entity type.

3.7.1 FCO-IM Repository

UCFORROL (It is used to find out if the role is part of a UC)
ROLE (in this table is taken the information about the role)
OTFT (it is used to get the information on the nominalized OTFT which plays the role)
UC (it is used to get the information about uniqueness constraint like Primary attribute and comment)

3.7.2 ERM Repository

ALTRKEY
ATTRAKEY
RELSAKEY

3.7.3 Queries

3.7.3.1 Query for retrieving information about the alternative keys

```
SELECT
UC AS ALTKEYNAME,
IIF(ISNULL(UC.PRIMARY_), 'No', UC.PRIMARY_) AS PRIMARY_,
UC.COMMENT AS COMMENT
FROM UC, UCFORROL, ROLE
WHERE UC.UCNO= UCFORROL.UC
AND ROLE.ROLENO= UCFORROL.ROLE
AND (ISSUBTYPER='NO' OR ISSUBTYPER IS NULL)
AND ROLE.PARTOFFT NOT IN
        (SELECT RELNAME
        FROM ERM_TEMP_RELATIONSHIP)
```

Note: ERM_TEMP_RELATIONSHIP table is a temporary table where the information about relationships is kept.

3.7.3.2 Query for retrieving the information about the attributes that are part of an alternative key

```
SELECT
UCFORROL.UC AS ALTKEYNAME,
ROLE.PARTOFFT AS ENTITYNAME,
ROLE.ROLENO AS ATTID
FROM UCFORROL, ROLE, OTFT
WHERE UCFORROL.ROLE = ROLE.ROLENO
AND ROLE.PLAYEDBYOT = OTFT.OTFTNAME
AND OTFT.TYPE = 'L'
```

3.7.3.3 Query for retrieving the information about the relationship sides that are part of an alternative key

```
SELECT
UCFORROL.UC AS ALTKEYNAME,
'1' AS SIDENO,
ROLE.PARTOFFT + '_' + ROLE.PLAYEDBYOT + '_'  + ROLE.ROLENO AS RELATIONSH
FROM UCFORROL, ROLE, OTFT
WHERE UCFORROL.ROLE = ROLE.ROLENO
AND ROLE.PLAYEDBYOT = OTFT.OTFTNAME
AND (ISSUBTYPER='NO' OR ISSUBTYPER IS NULL)
AND OTFT.TYPE='N'
AND ROLE.PARTOFFT NOT IN
(SELECT RELNAME
FROM ERM_TEMP_RELATIONSHIP)
```

3.8 Primary Keys

Primary keys are defined in ERM_ALTRKEY by the property PRIMARY_. The property of being part of the primary key is represented for the attribute by PRIMARY_ in table ERM_ATTRIBUTE and for the relationship side by DEPENDENT in ERM_RELSIDE.

3.8.1 Queries

3.8.1.1 Query for assigning to the attribute the attribute of being part of primary key

```
UPDATE ERM_ATTRIBUTE
SET PRIMARY_= 'Yes'
WHERE ATTID IN
        (SELECT ATTID
        FROM ERM_ATTRIBUT_ALTKEY AS ATTA, ERM_ALTRKEY AS A
        WHERE ATTA.ALTKEYNAME= A.ALTKEYNAME
        AND ATTA.ENTITYNAME= ERM_ATTRIBUTE.ENTITYNAME
        AND A.PRIMARY_='Yes')
```

3.8.1.2 Query for assigning to the relationship side the attribute of being part of primary key

```
UPDATE ERM_RELSIDE
SET DEPENDENT = 'Yes'
WHERE RELATIONSH IN
        (SELECT RELATIONSH
        FROM ERM_RELSIDE_ALTKEY AS RSA, ERM_ALTRKEY AS A
        WHERE RSA.ALTKEYNAME= A.ALTKEYNAME
        AND RSA.SIDENO=ERM_RELSIDE.SIDENO
        AND A.PRIMARY_='Yes')
```