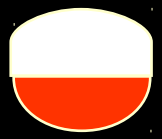
Zawansowane Modelowanie i Analiza Systemów Informatycznych

(l-3)



Polsko-Japońska Wyższa Szkoła Technik Komputerowych
Katedra Systemów Informacyjnych
2013

Overview

- **Steps 1-4 of CSDP**

ORM : Seven Steps of the Conceptual Schema Design Procedure (CSDP)

1. **Transform information examples into elementary facts, and apply quality checks.**
2. **Draw fact types, and apply population checks.**
3. **Check for arithmetic derivations of fact types, and superfluous entity types.**
4. **Add uniqueness constraints, and check the arity of fact types.**
5. **Add mandatory role constraints, and check logical derivations.**
6. **Add value, set comparisons, and subtyping constraints.**
7. **Add other constraints and perform final checks.**

Step 1:Transform familiar information examples into elementary facts.

Student	Subject	Lecturer
02244556	ITB220	P. Brown
02244557	ITB220	P. Brown
02244557	ITB106	J. Reye
02244557	ITB225	J. Reye

The STUDENT identified by STUDENT# 02244556 ENROLLED IN the SUBJECT identified by SUBJECT CODE ITB220.

The LECTURER identified by NAME P. Brown delivers the SUBJECT identified by SUBJECT CODE ITB220.

Step 1 (con't)

For each fact type prepare a sample population of this fact type. For example:

TEACHING

(Entity type)	SUBJECT	LECTURER
(Label type)	SUBJECT CODE	NAME
(Role type)	LECTURED-BY	DRLIVERS
(Fact instance)	ITB220	P. Brown
…….	ITB106	J. Reye
	ITB225	J. Reye

2. Perform Quality Design Checks

Is the following fact type structure better than the 2 fact types structures on slide 4?

The STUDENT with STUDENT# 02244556 ENROLLED IN SUBJECT with SUBJECT CODE itb220, and was LECTURED BY the LECTURER with NAME P. Brown.

Is the student enrolment information independent of lecturing information? (yes)

If the answer is "yes", the former fact type structure is a 'better' model of the reality.

Elementarty Fact Type

Definition:

The adjective „elementary” means that the fact cannot be ‘split” into smaller units of information that collectively provide the same information as the original fact type.

1) Use output reports or similar sources of information for the familiar examples

Example output report:

Car	*Make*	*Colour*
OAJ468	*Toyota*	*White*
OCR606	*Ford*	*Green*
OKH575	*Mitsubishi*	*Red*
OYX592	*GMH*	*Yellow*
OZY227	*Ford*	*Red*

Example sentences:

The Car with REG# OAJ468
was manufactured by
the Company with COMPANY_NAME Toyota
and
Is finished in
the Colour with COLOUR_NAME White.

WRONG !

not an elementary fact type; it is a 'conjunction' of two facts.

It may be seen from the output report that Colour is related to the Car and, independently, the Manufacturer and Car are related.

Correct deep structure sentences

The Car with REG# OAJ468
was manufactured by
the Company with COMPANY_NAME Toyota.

The Car with REG# OAJ468
is finished in the
Colour with COLOUR_NAME White.

2) for each sentence type

Prepare a fact type - instance table showing the entity types, label types and roles.

Populate each table with a set of significant instances of label types.

Car	*Company*
REG#	*COMPANY_NAME*
was_ manufactured_by	*manufactured*
OAJ468 *OCR606* *OKH575* *OYX592* *OZY227*	*Toyota* *Ford* *Mitsubishi* *GMH* *Ford*

Manufacturer

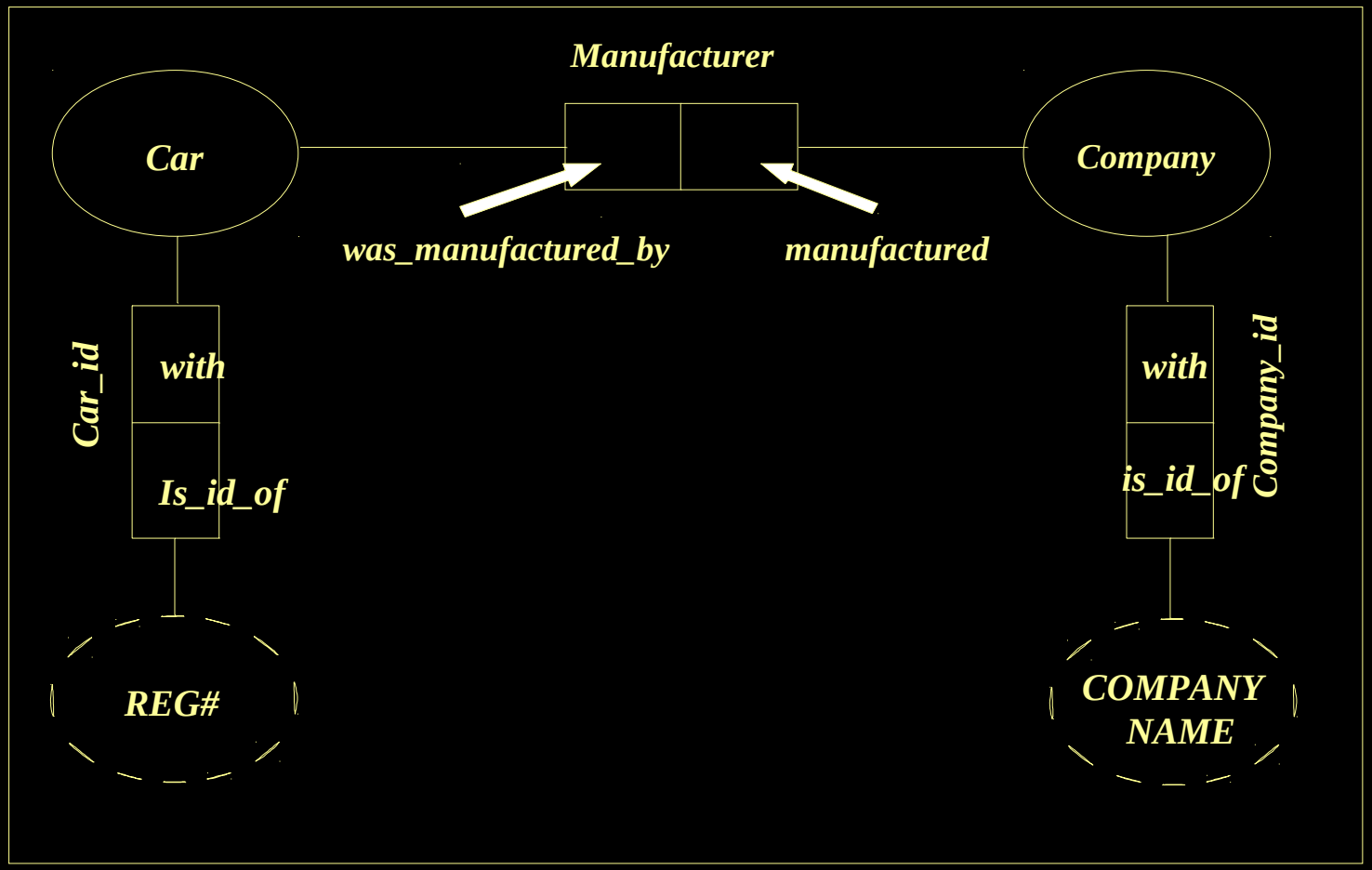
Car	*Colour*
REG#	*COLOUR_NAME*
is_finished_in	*is_finish_of*
OAJ468 *OCR606* *OKH575* *OYX592* *OZY227*	*White* *Green* *Red* *Yellow* *Red*

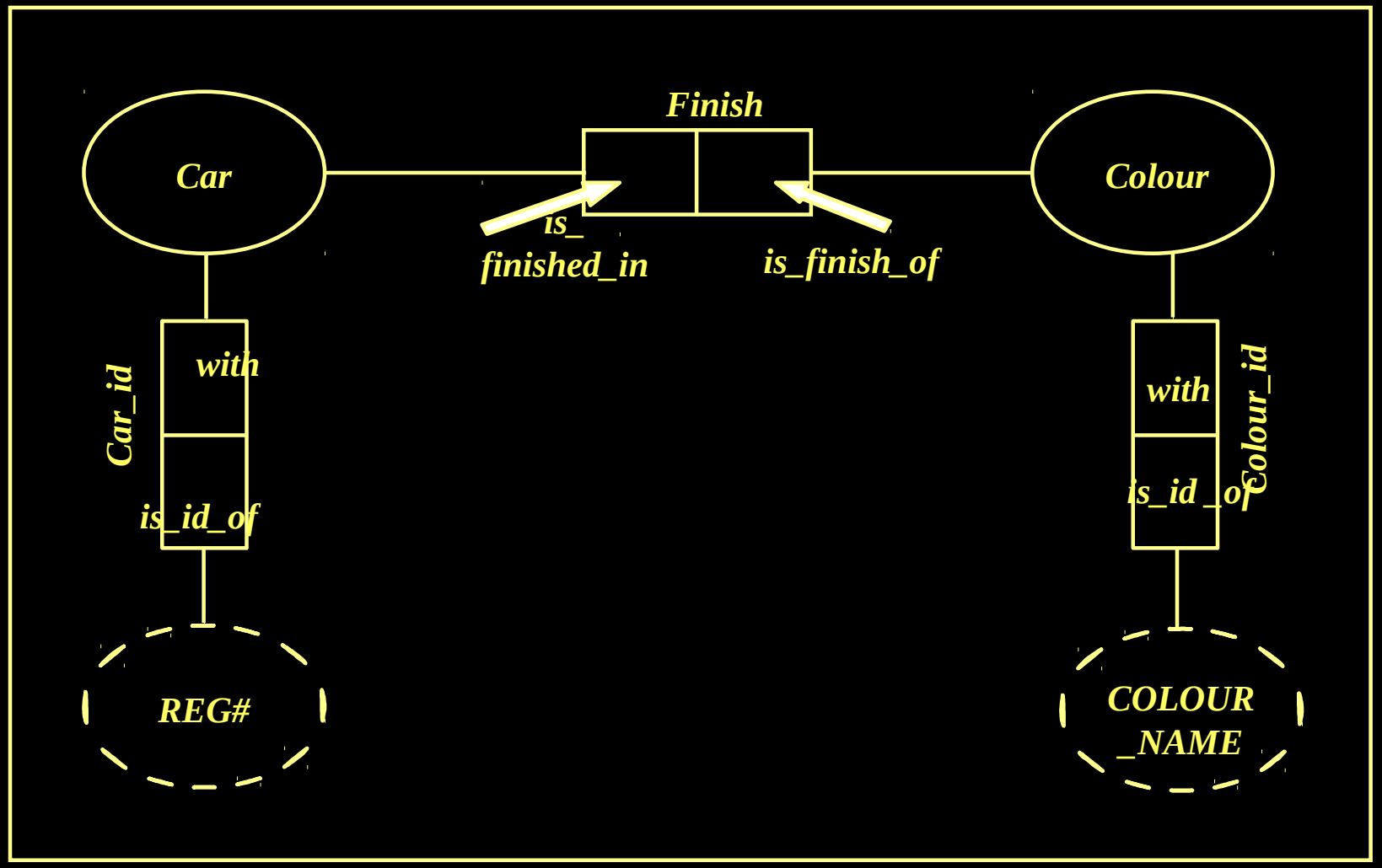
Finish

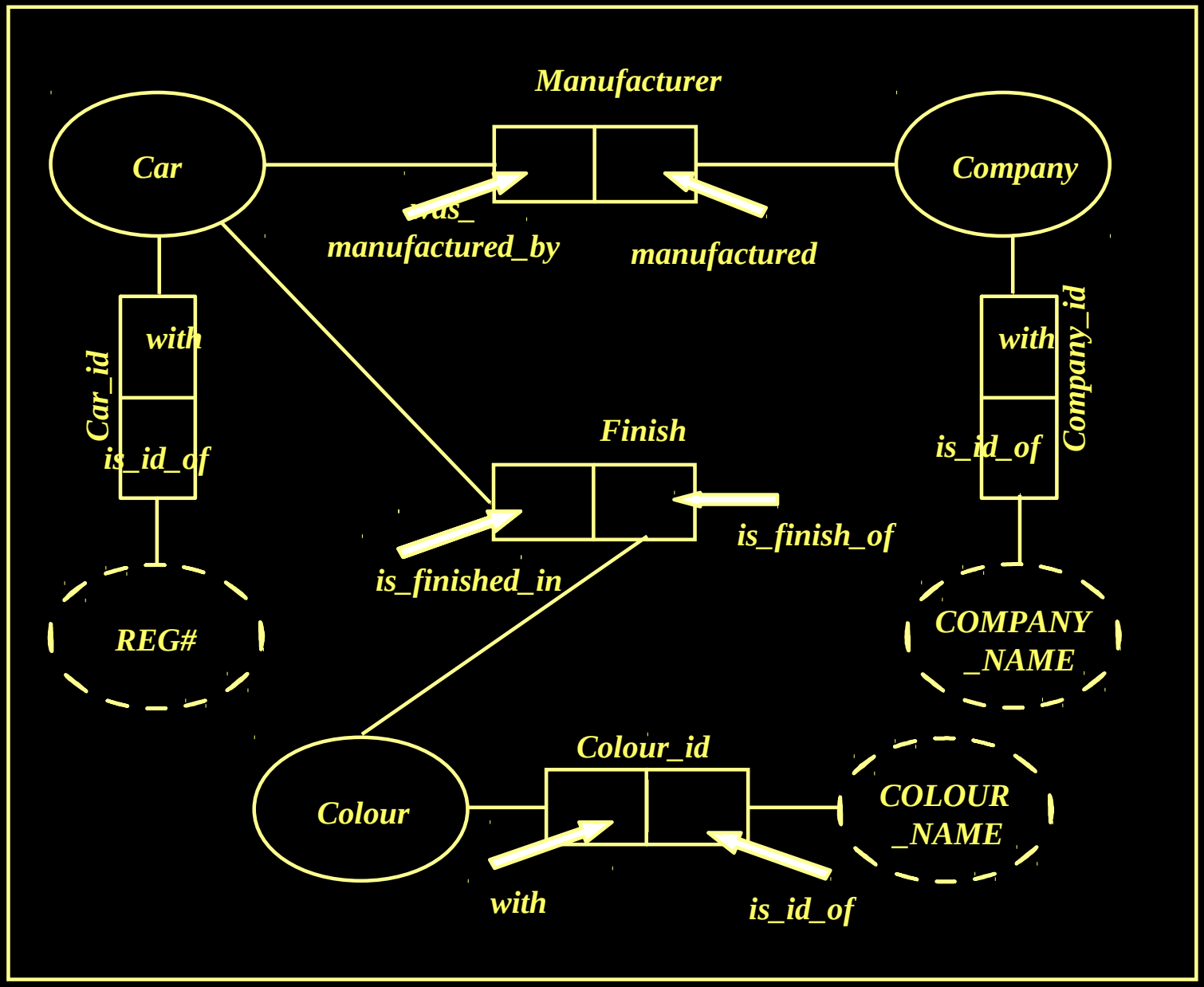
***Step 1* is critcal for the correctness of the final design!!!!**

For complex cases – formed on the bases of many reports – watch for ‘connectivity’ of the developed structures. Some information may be missing!!!

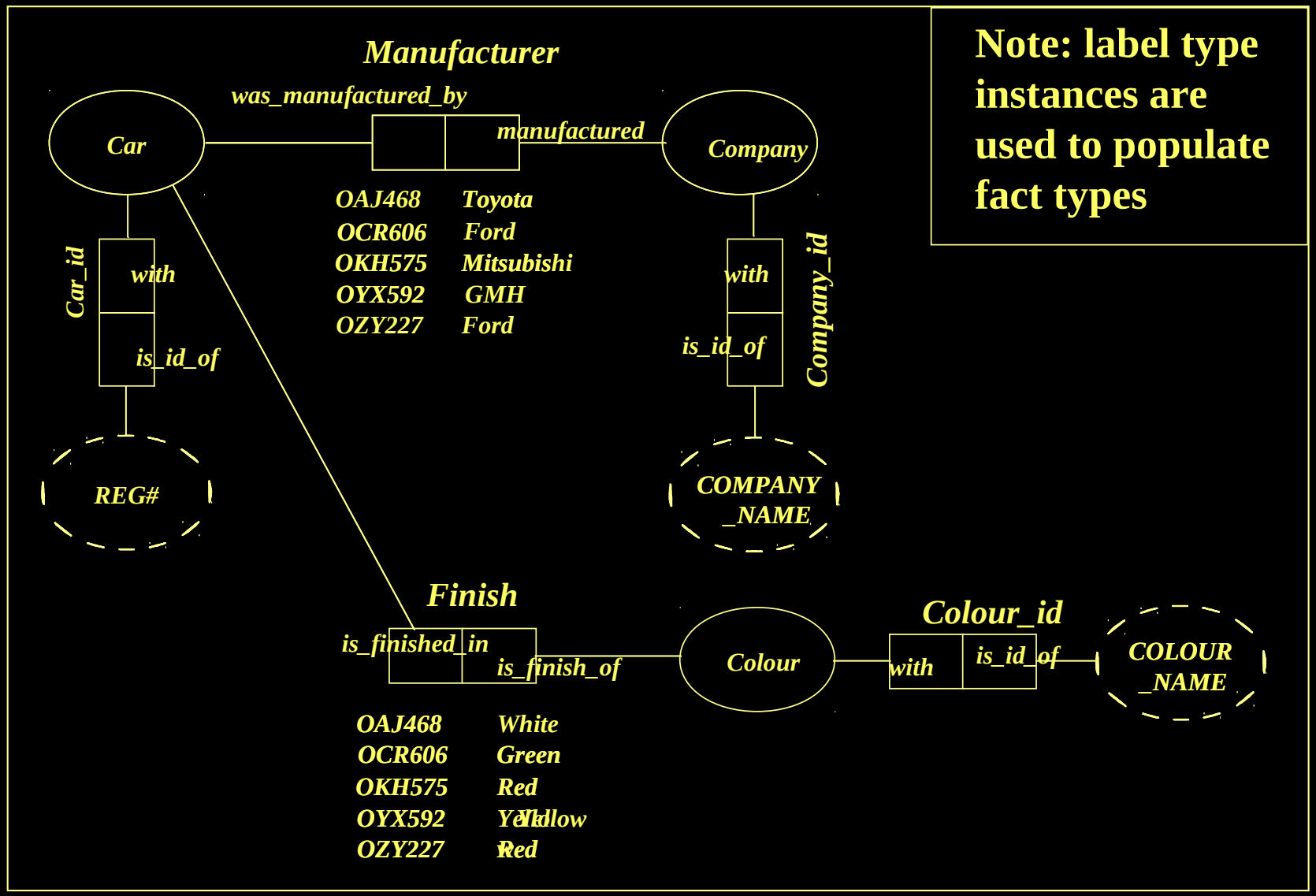
Step 2: Draw a first draft of the conceptual schema diagram and apply a population check.



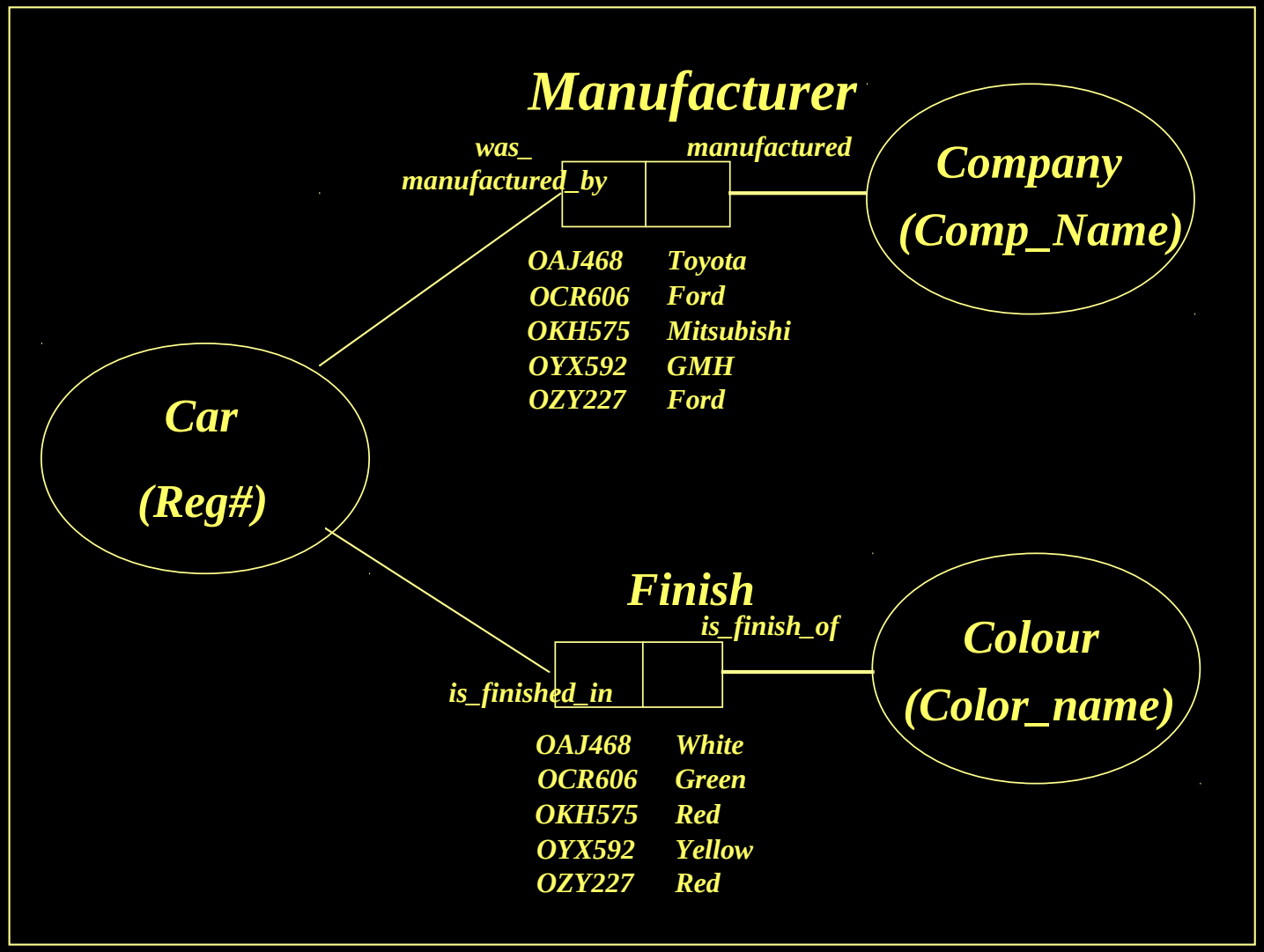

Finish
Car
Colour
is_
finished_in
is_finish_of
with
Car_id
is_id_of
with
Colour_id
is_id _of
REG#
COLOUR
_NAME


Manufacturer
Car
Company
was_
manufactured_by
manufactured
with
Car_id
is_id_of
with
Company_id
is_id_of
Finish
is_finish_of
is_finished_in
REG#
COMPANY
_NAME
Colour_id
Colour
COLOUR
_NAME
with
is_id_of

Add Population of fact types to the conceptual schema diagram.



If a label type instance uniquely identifies each entity type instance abbreviate the diagram as follows:



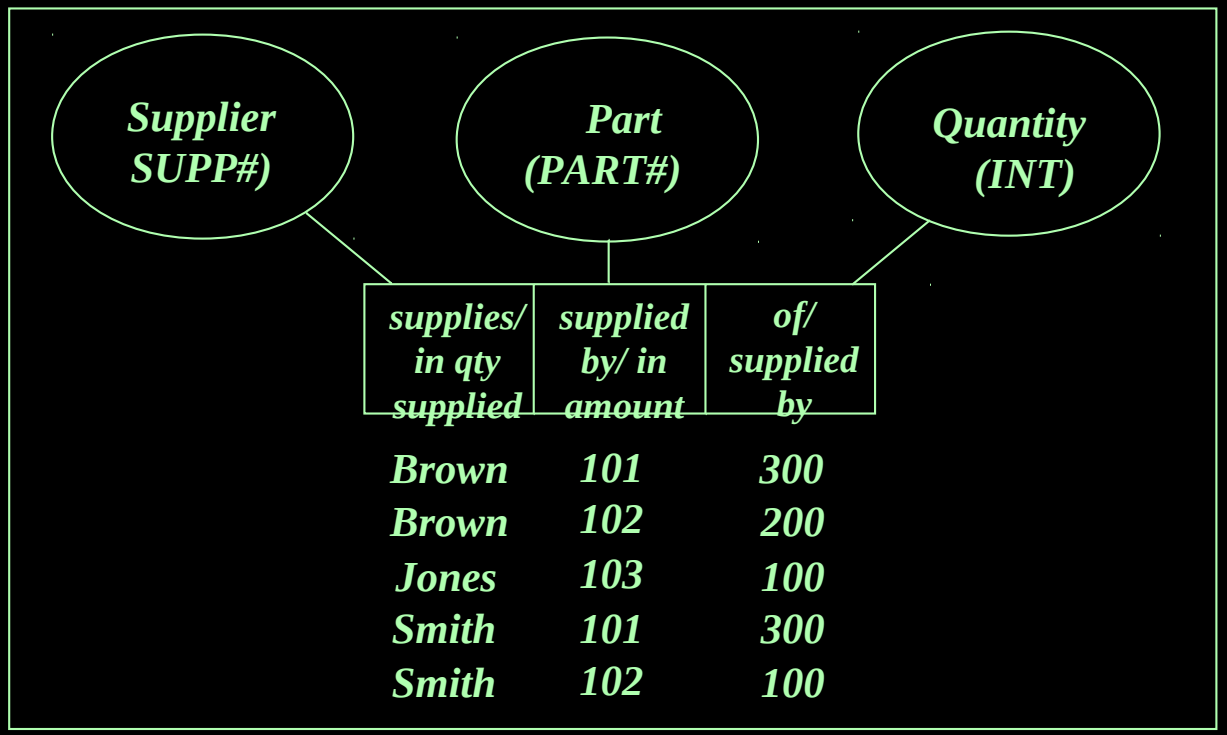
Example output report: Supply

Supplier	*Part*	*Quantity*
Brown	*101*	*300*
Brown	*102*	*200*
Jones	*103*	*100*
Smith	*101*	*300*
Smith	*102*	*100*

Various approaches to a solution will be discussed.

Example sentence:

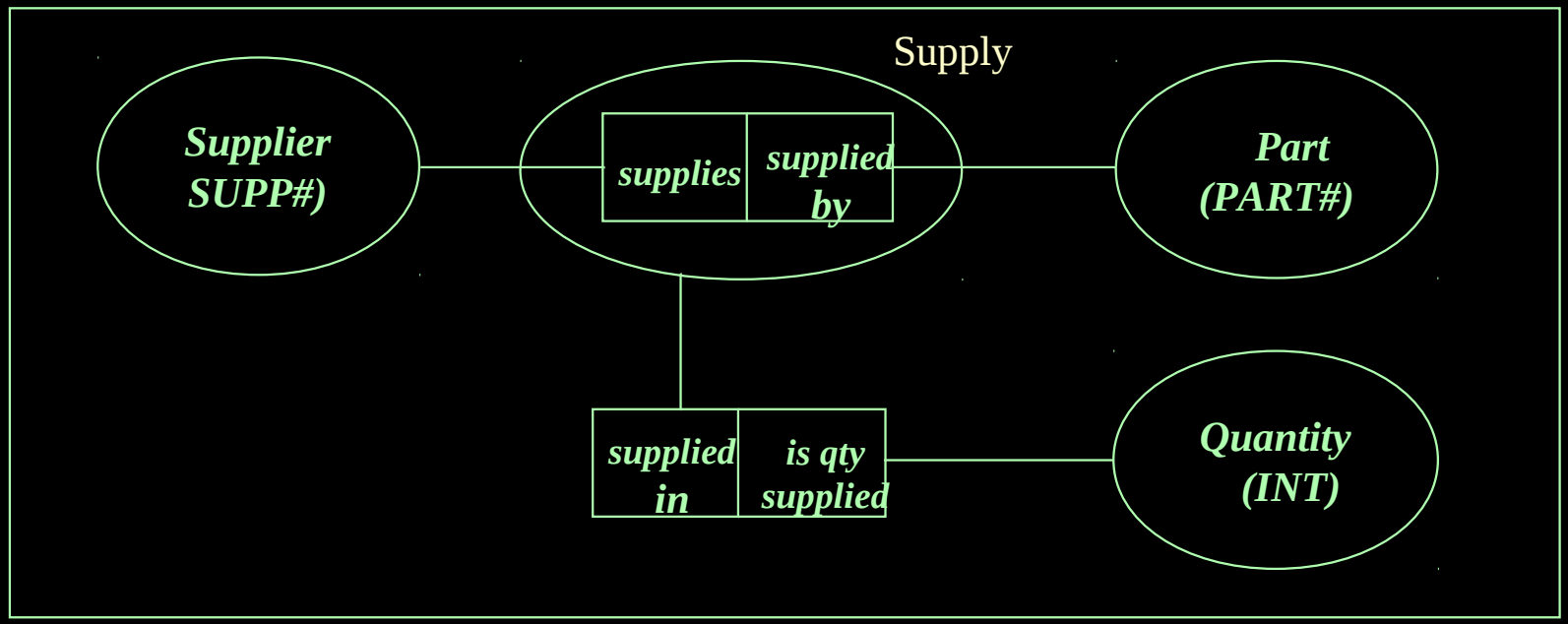
The Supplier with SUPP# Brown supplies the Part with PART# 101 in qty supplied Quantity of INT 300



Alternative example sentence:
The Supplier with SUPPLIER_NAME Brown supplies
the Part with PART# 101.

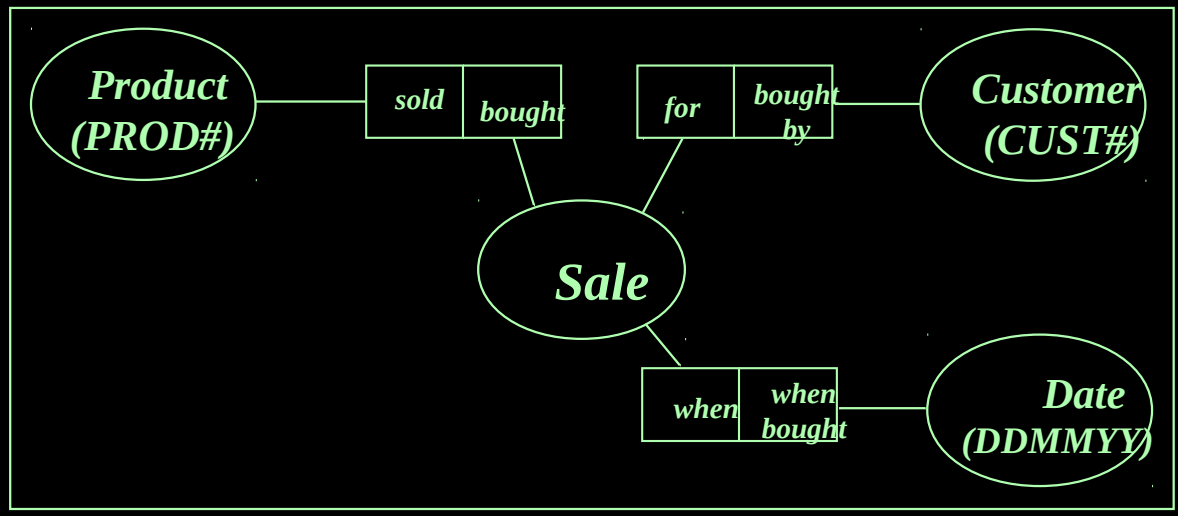
This Supply is supplied in Quantity of INT 300.

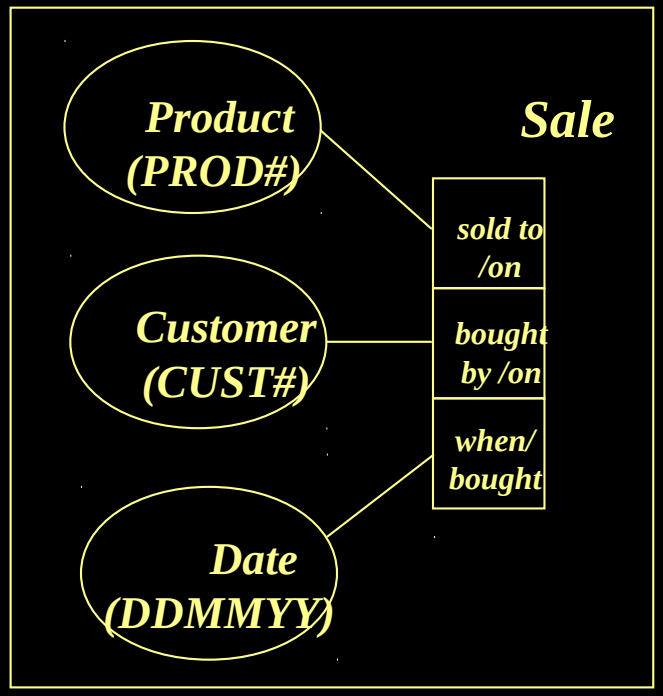
The fact type 'Supply' is called an OBJECTIFIED fact type or a NESTED fact type



Step 3: Eliminate any surplus entity types and common roles and identify any derived fact types.

Unnecessary entity types can be removed:



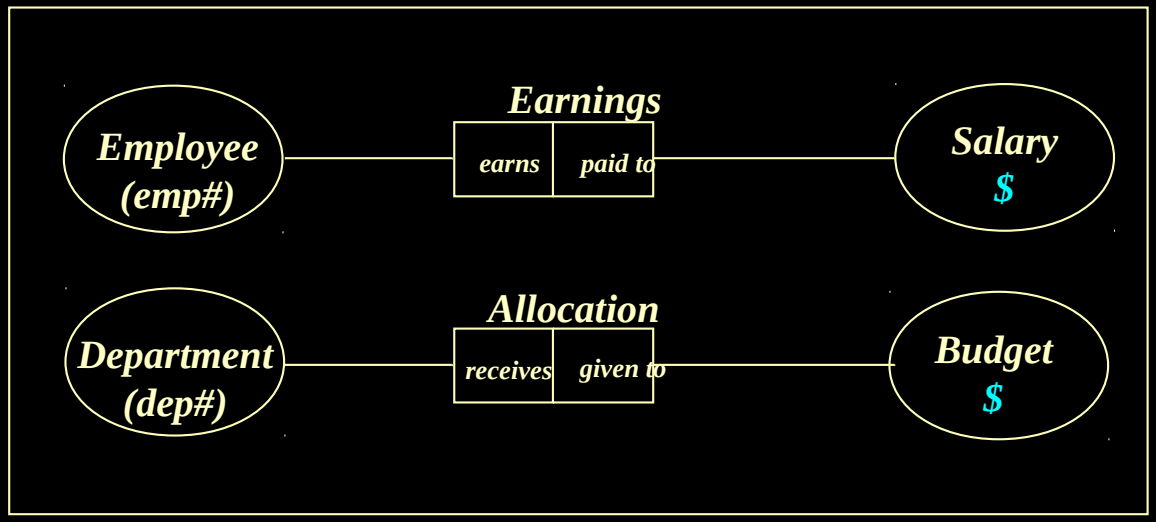


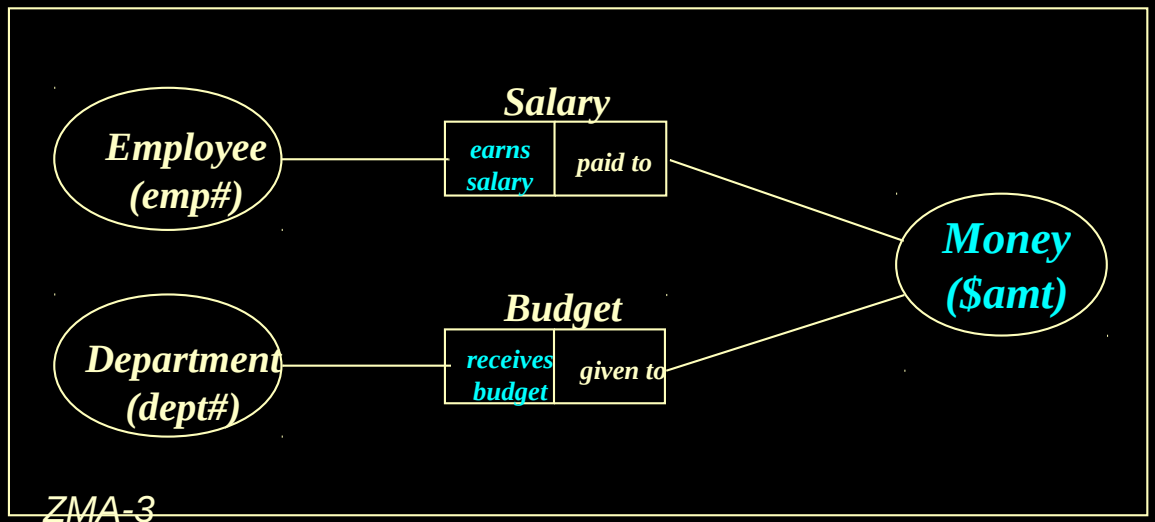
The redundant entity type Sale is not needed:

Examine all entity types represented in the draft schema(s) and combine those which:

-share at least one common role

-can be meaningfully compared ie have the same dimensions



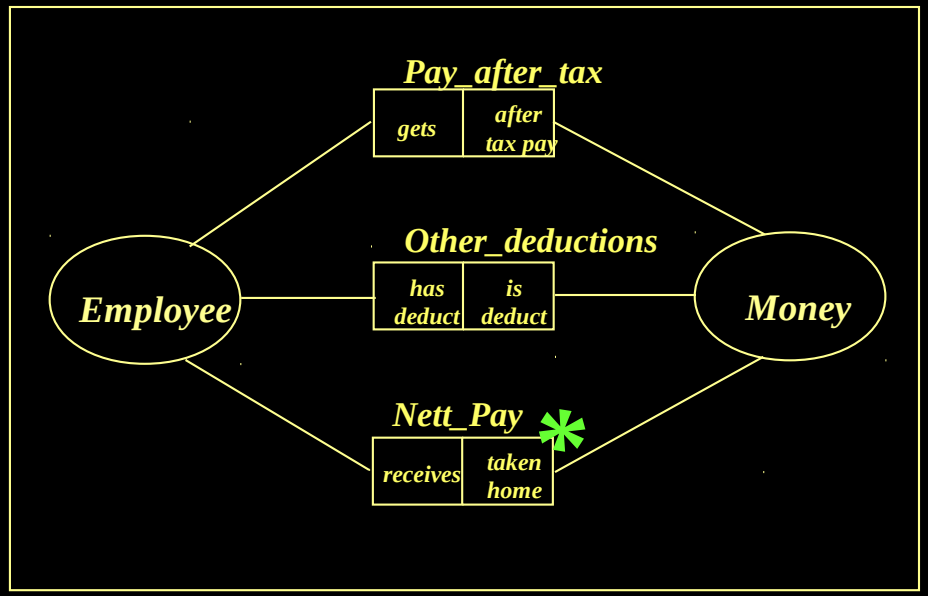


Money becomes the single entity type

Step 3 Cont. - Eliminating derivable Fact types.

At the conceptual level all redundancy should be removed.

Consider the following example:

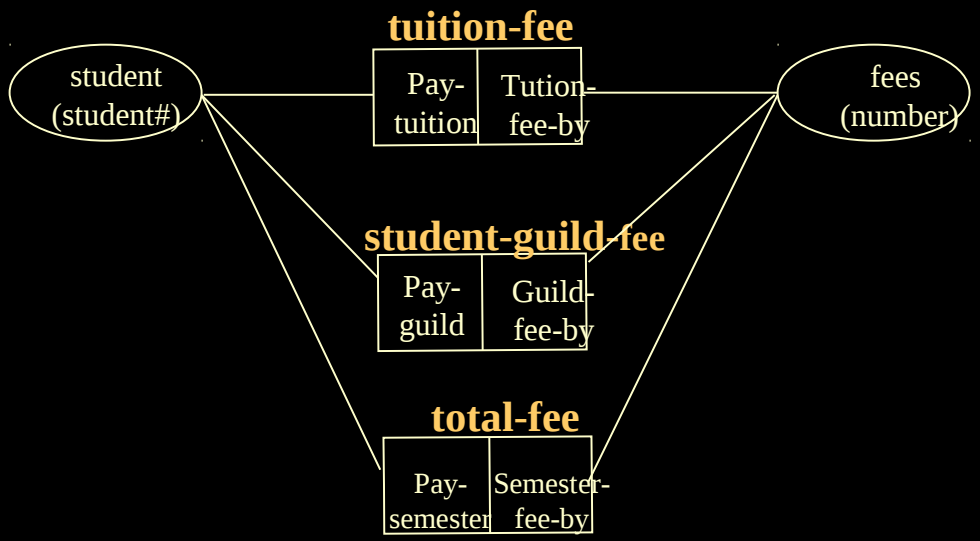


One of the three fact types is redundant and should be removed or marked as redundant by an asterisk. Assuming Nett_Pay is derivable then mark it with an asterisk, and write a derivation routine:

Nett_pay := Pay_after_tax - other_deductions

Example

2. Check for arithmetic derivations.



Arithmetic derivation:

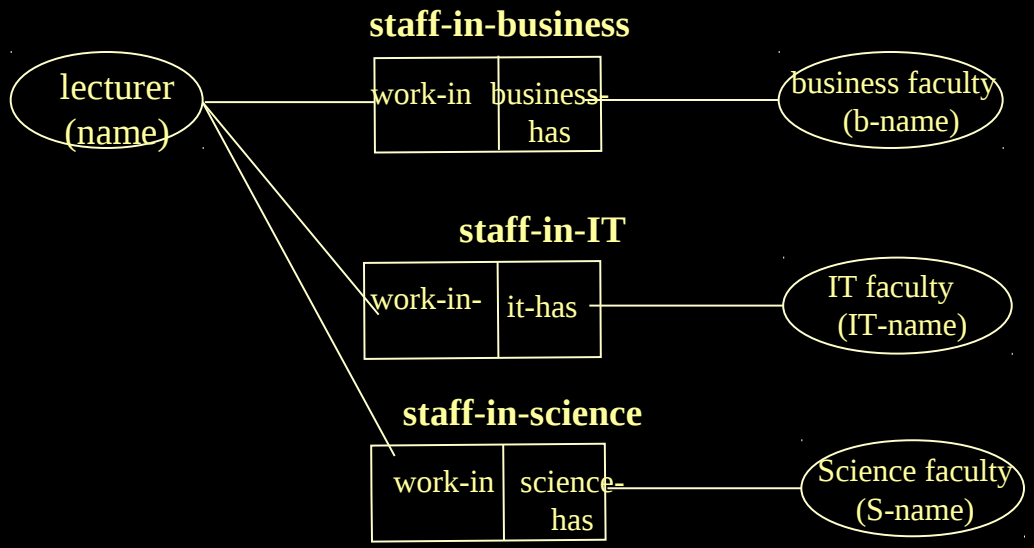
total-fee = tuition-fee + student-guild-fee

Therefore, the fact type “total-fee” is redundant!

Another example

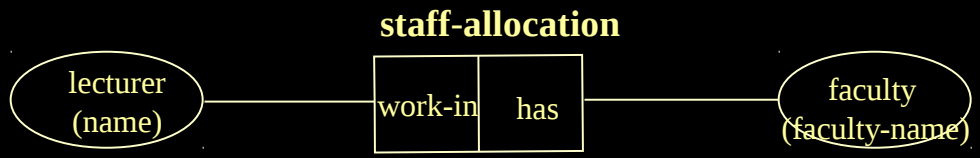
1. Check for superfluous entity types.

Conceptual schema 1



Business, IT, Science are only instances of the Faculty entity type!! Schema 1 has superfluous entity types.

Conceptual schema 2



Entity or Label?

Should I use an entity type or a label type to model a column (attribute, domain of a set of values) appearing in a sample report?

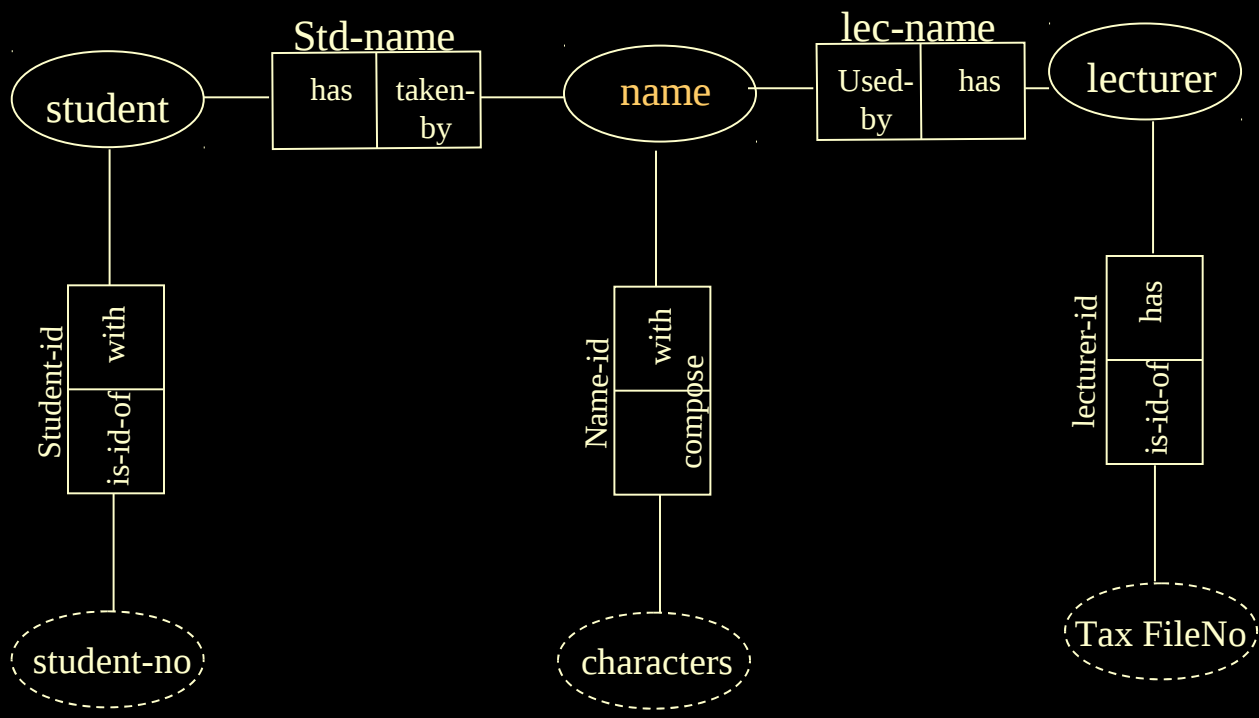
In E-R model or UML, attributes are always used to describe an entity (object). In ORM, if the DOMAIN of a type of data is related to only ONE entity type, the associated attribute can be modelled as a label type; otherwise it should be modelled as an entity type that is associated with an elementary label type.

Remember the syntactic rule in ORM:

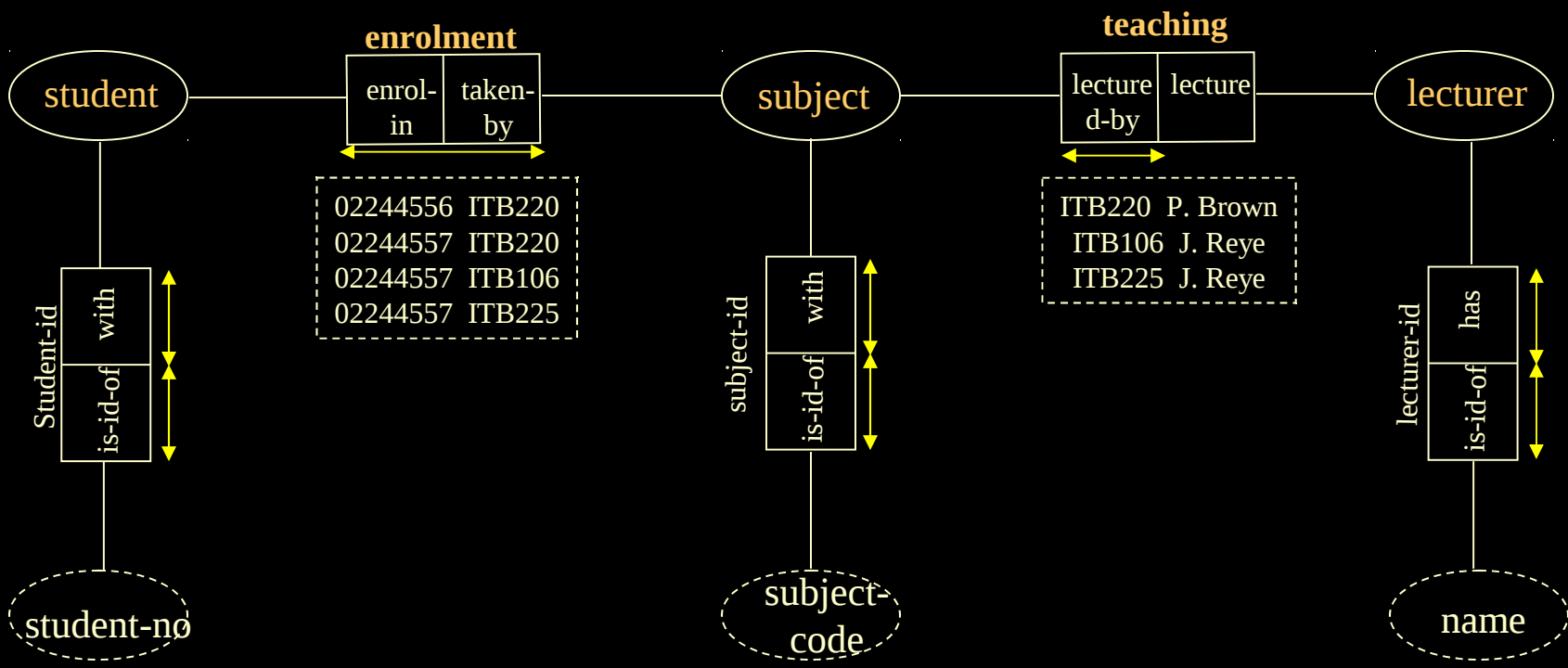
a label type can only be associated with ONE entity type!

Entity or Label?

Example: If NAME is an attribute that can be shared by the student entity and the lecturer entity, it must be modelled as an entity type rather than a label type!



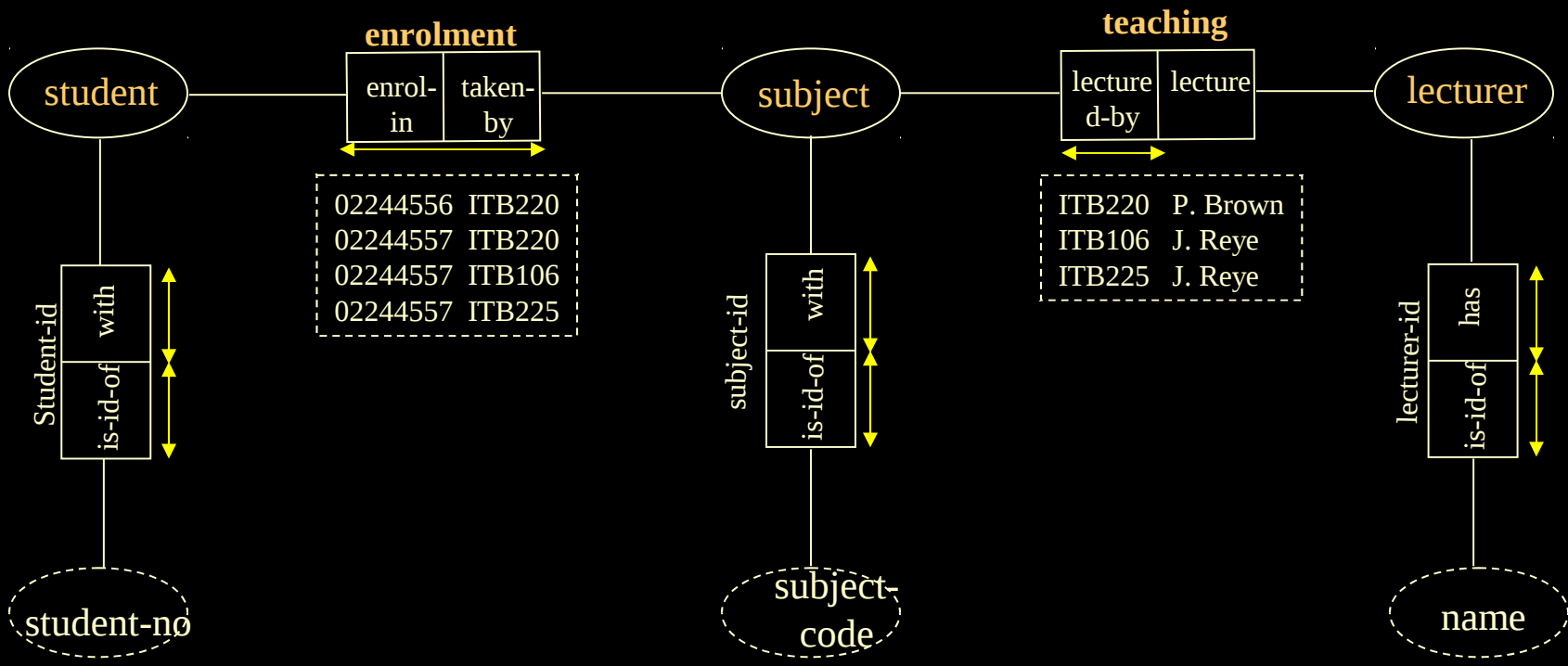
Step 4 - Add uniqueness constraints



Is there anything wrong in this conceptual schema diagram?

Step 4 (con't)

Answer: No, the conceptual schema satisfies the given fact instances.
However, real life situations may be more complicated!



Consider, that a subject can be delivered by more than one lecturer and the same name may be used by many lecturers. Then the uniquenes constraint on fact type teaching should be revised.

Step 4 (Con't)

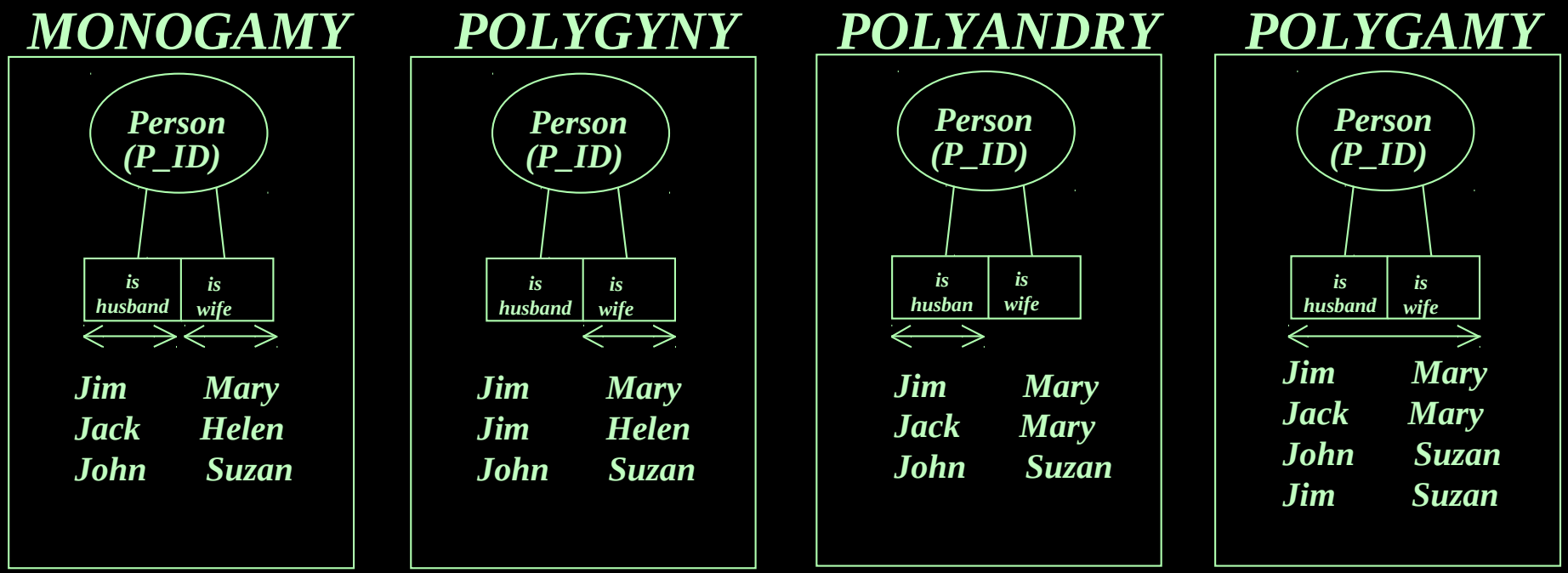
In reality, you can obtain the semantics of information (e.g. uniqueness constraints) from system owners/users.

Develop your conceptual schema diagram solely based on the given information (e.g. requirements of a particular organization) rather than your personal knowledge or experience!!!

Step 4 -more examples for binary fact types

Example:
The marriage contract

(covering current marriages)



1. The husband or the wife provide unique identification to the Marriage_Contract fact.

2. In this situation the wife's name provides identification of a single Marriage_Contract.

3. As a wife has many husbands (is involved in many Marriage_ Contracts), the husband's name uniquely identifies the fact.

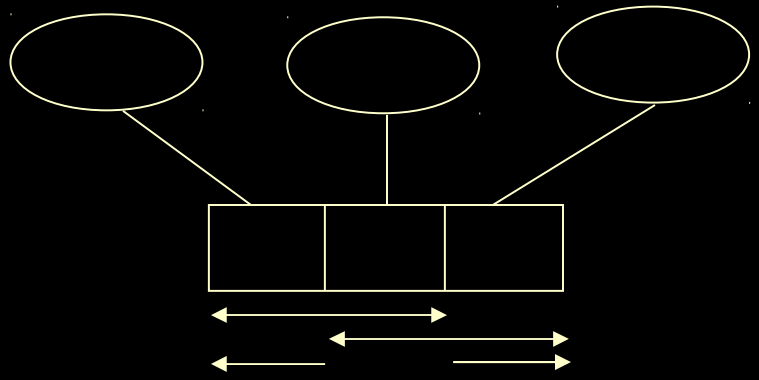
4. Both wives and husbands are involved in many Marriage_Contracts, therefore the combination identifies a fact.

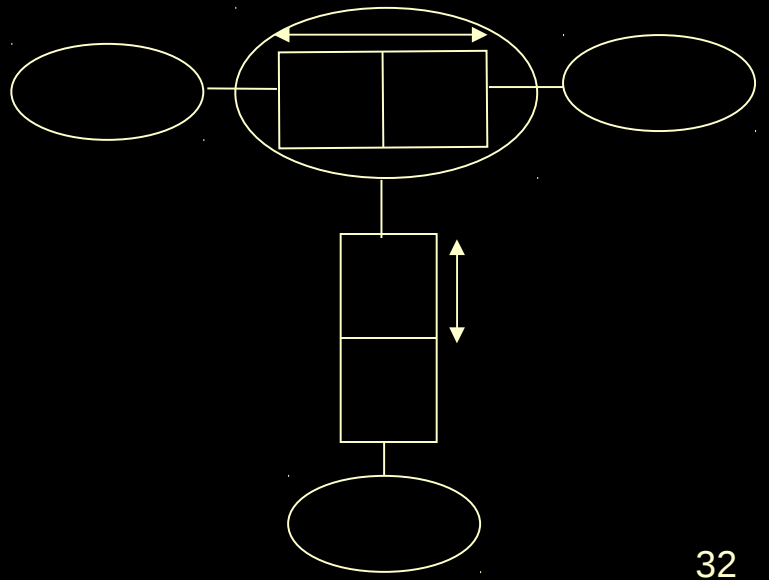
Special important cases:

A given n-ary fact type may have at most n uniqueness constrains with each one spanning (n – 1) roles.

Each n-ary fact type has at least one uniqueness constraint which spans at least (n – 1) roles.

In a nested fact type, the uniqueness constraint involves every role of the nested fact type.

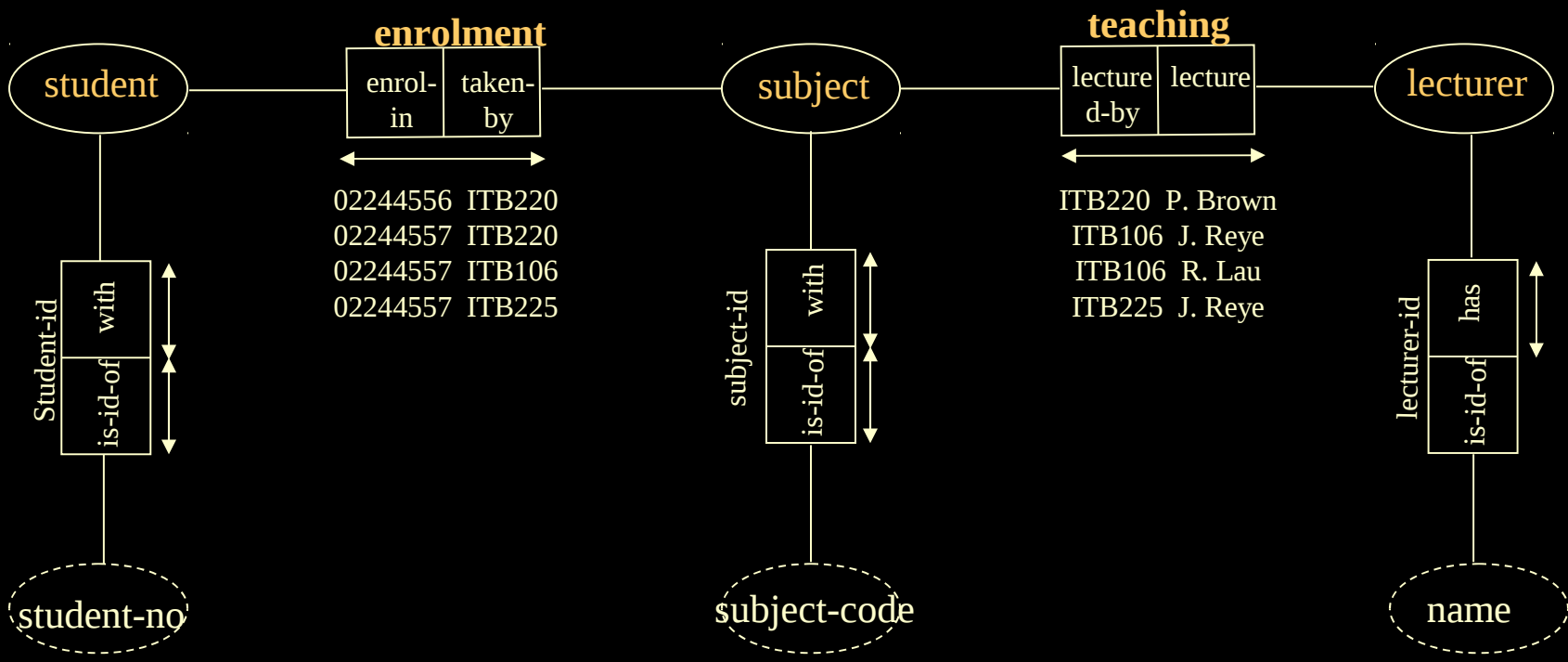




Inter-fact type uniqueness constraint:

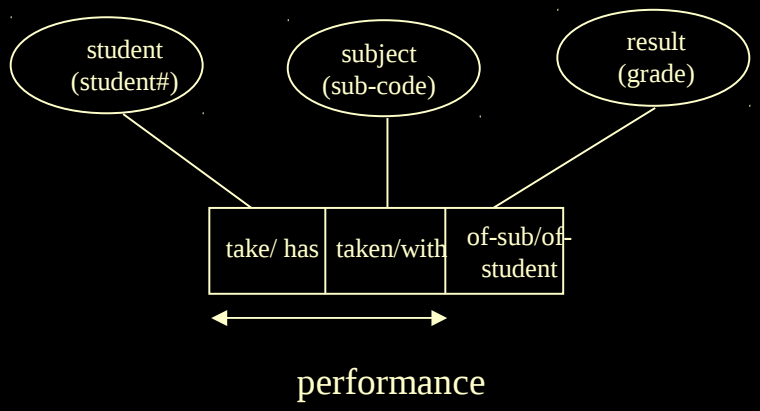
Enrolment : A student can enrol in many subjects. A subject can be taken by many students.

Teaching : A lecturer can be responsible for many subjects. A subject can be shared by many lecturers.



2. Check the fact type for correctness of arity.

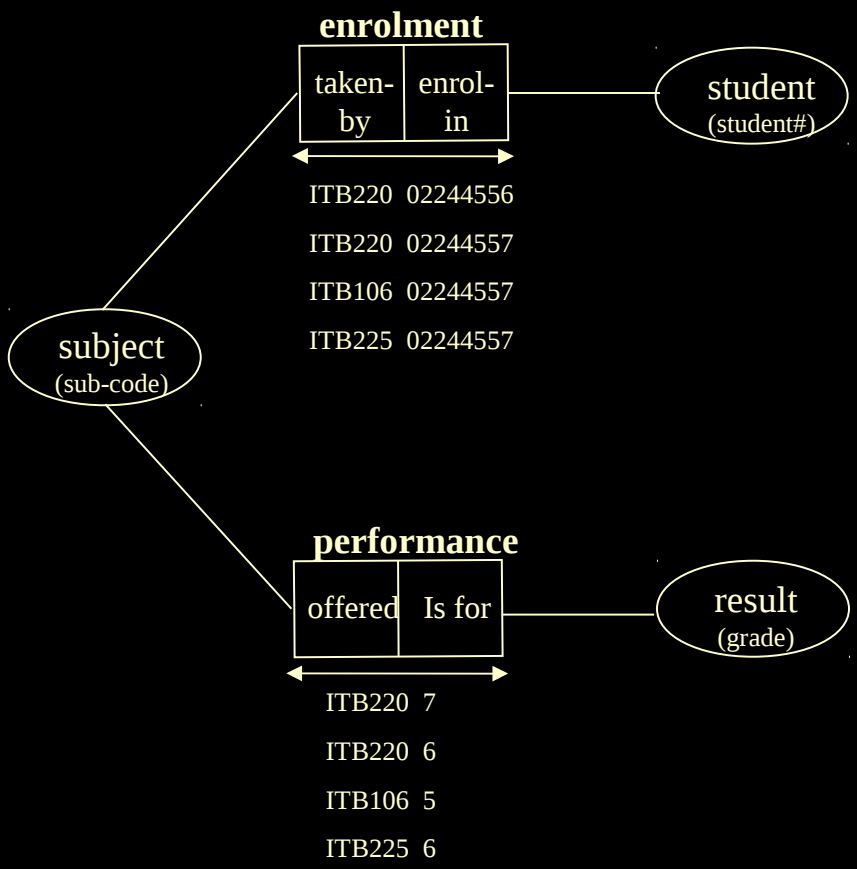
(It can be a nontrivial process)



STD-NO	SUBJECT	RESULT
02244556	ITB220	7
02244557	ITB220	6
02244557	ITB106	5
02244557	ITB225	6

Step 4 (con't)

Can we use two binary fact types to represent the UoD?



STD-NO	SUBJECT	RESULT
02244556	ITB220	7
02244557	ITB220	6
02244557	ITB106	5
02244557	ITB225	6

Step 4 (con't)

Check the fact type for correctness of arity.

ITB220 02244556		ITB220 7		ITB220 02244556 7
ITB220 02244557	⋈	ITB220 6	=	ITB220 02244556 6
ITB106 02244557		ITB106 5		ITB220 02244557 7
ITB225 02244557		ITB225 6		ITB220 02244557 6
				……
				……..

It produces spurious tuples! So, the conceptual schema in the previous slide has wrong fact type arity!

To ensure the correct fact type arity, one should try to go through each possible natural join on the decomposed fact types before one can draw a conclusion that the fact type in question is non-splittable.

Step 4 (con't)

The general procedure :

1. For an n-ary fact type, we need to check each of the n possible joins among the smaller 'parts' of that fact type

2. If all the possible joins fail, it means that you can't split the original fact type, so most likely it is an elementary fact type.

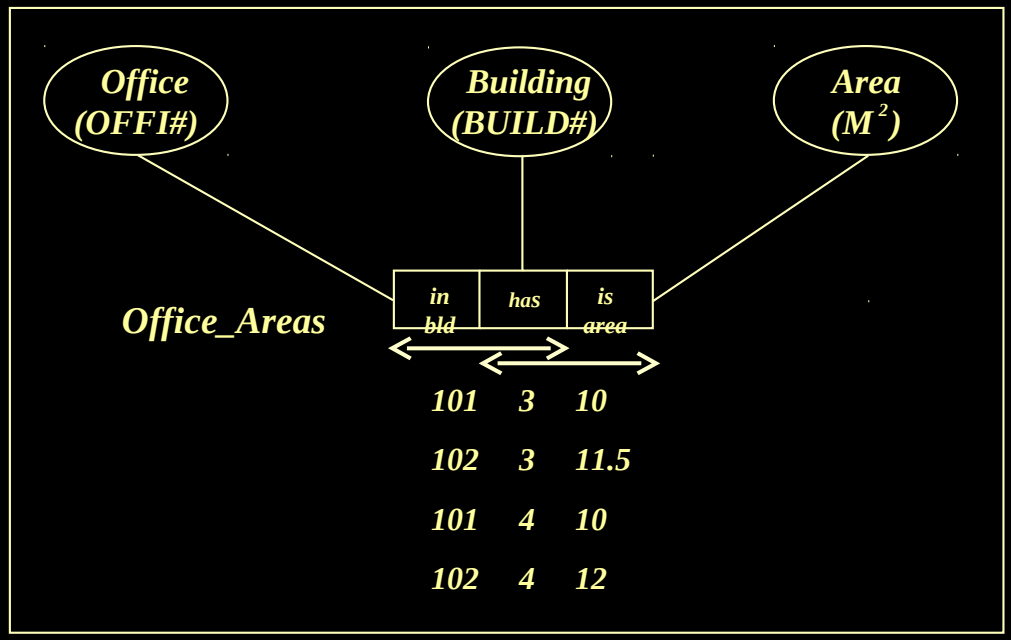
Example:

The previous student-result fact type consists of 3 roles. So, we should check if the 3 possible combinations of the pairs of binary fact types can recover the original population. If the answer is no, we need to check if joining all 3 binary fact types produces the original population.

The presented split into 'enrollment' and 'performance' fact types the 'performance' fact type associates the RESULT with the SUBJECT . It does not make any semantic sense within the University UoD unless all students would receive the same result for a subject.

Experienced analyst would reject that split upfront without even trying to compute the join of respective projections

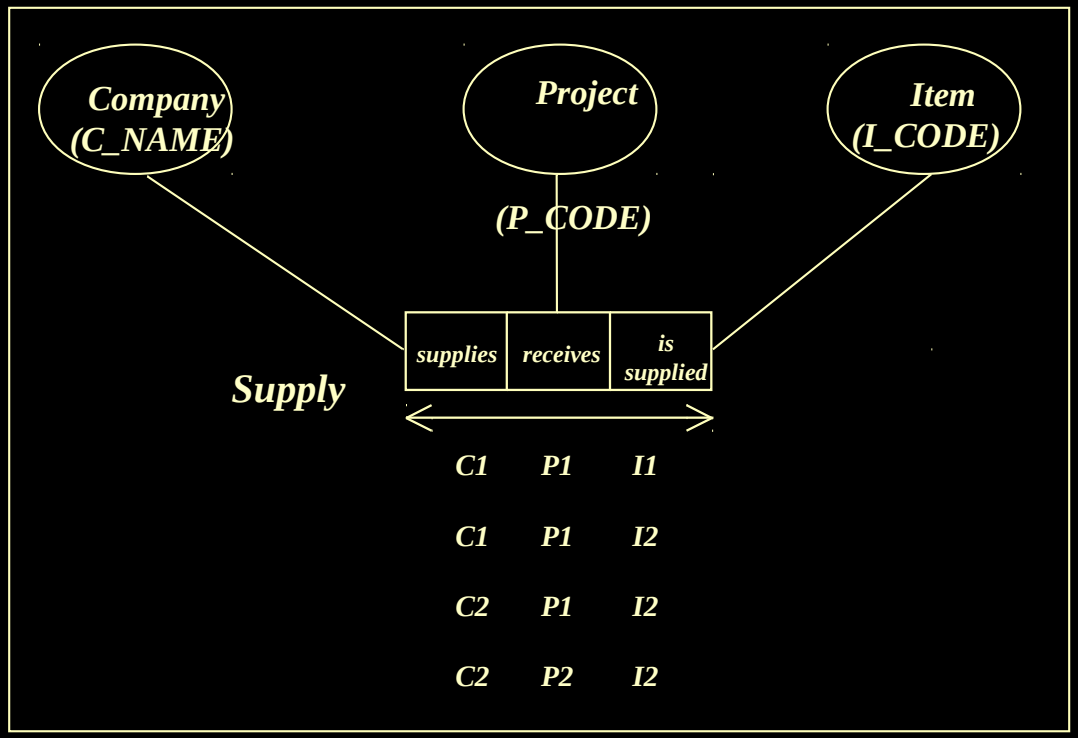
Uniqueness constraints for a ternary fact type



It is the combination of the first two roles which provides uniqueness.

Building No and Office Area are also unique with the data given, but they would be a poor choice of uniqueness constraint. Why?

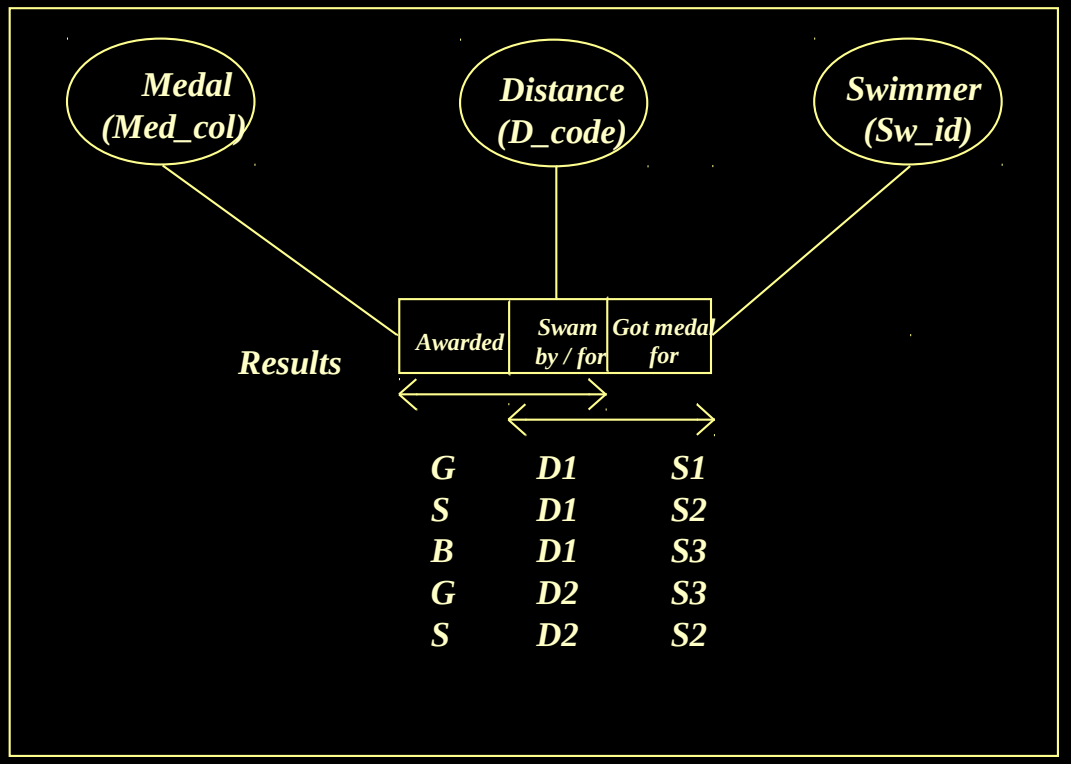
Ans. In one office building there could be many offices with the same floor area, so the combination of the second and third roles should be doubtfull as the uniqueness constraint choice. The analyst should check with UoD user in this regard if the sample data is significant



It is the combination of all three roles which provides uniqueness

An n-ary fact type may contain n uniqueness constraints, each spanning n-1 roles:

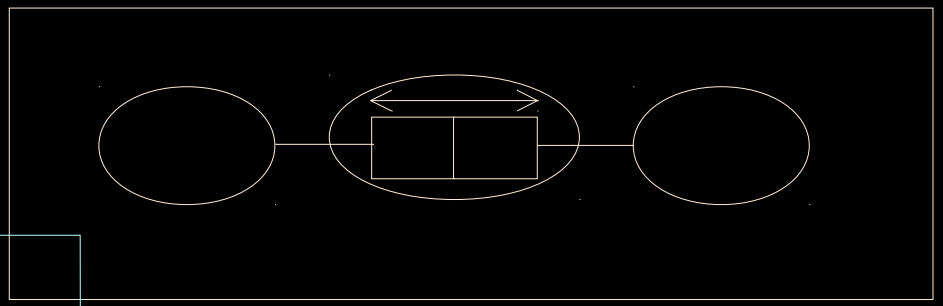
Example

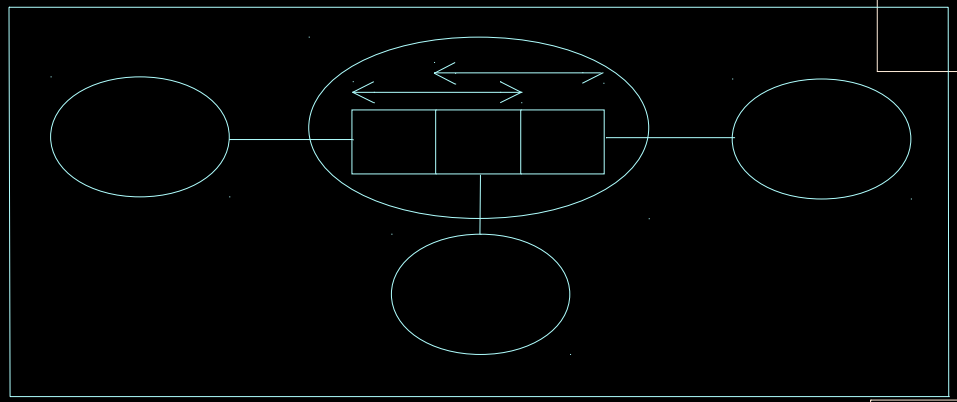


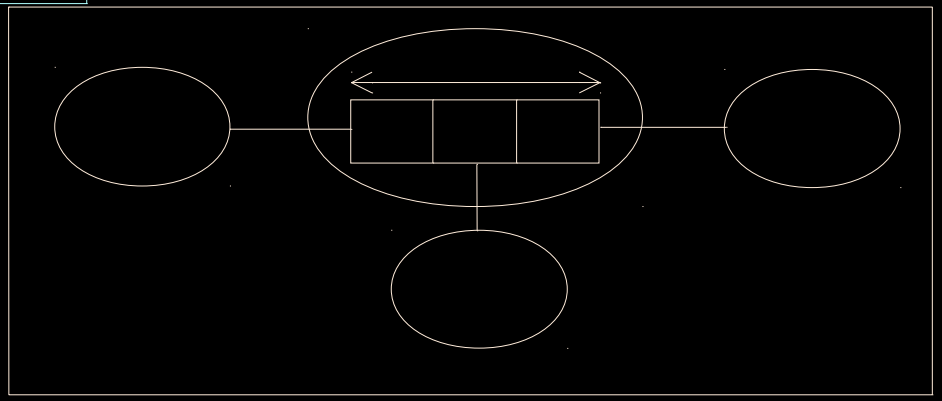
General rule:

An n-ary fact type should have at least one uniqueness constraint which spans at least (n-1) roles (if there is no uniqueness constraint spanning all n roles).

Note: In a nested fact type, the uniqueness constraints usually involve every role of the nested fact type, (all attributes are prime) and should be longer than one role.

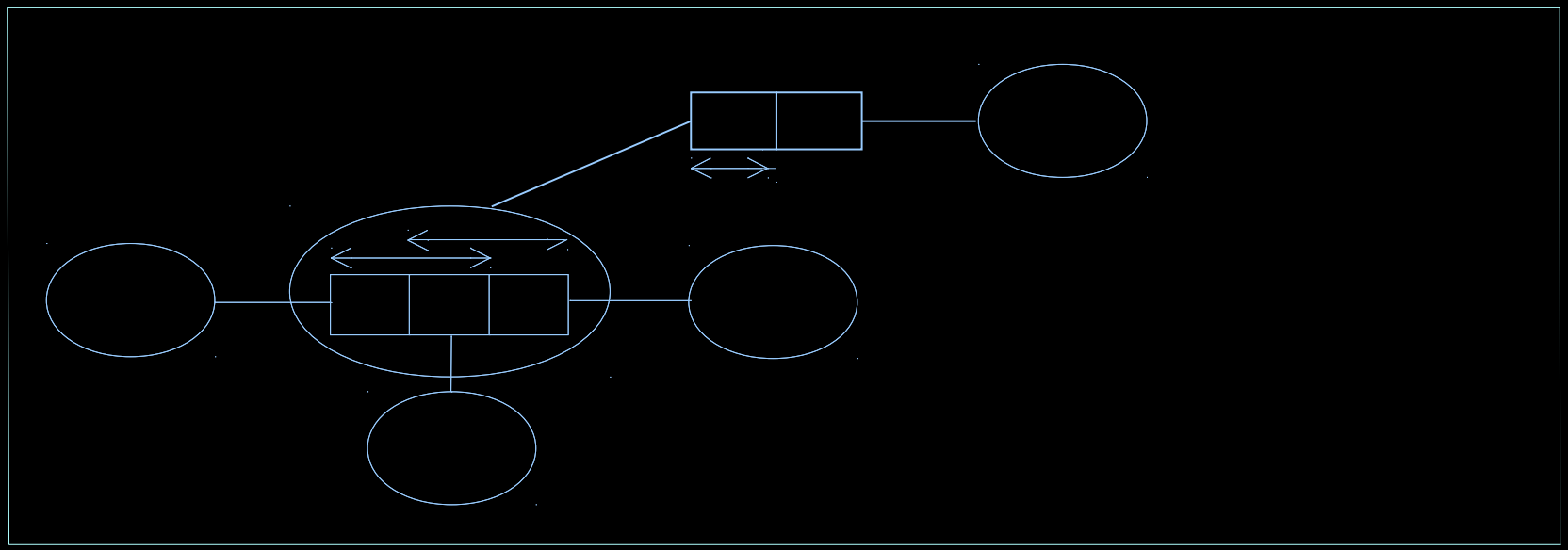






Important requirement

Nested fact type cannot exist without not nested fact type: It must play a role in a not nested fact type and that role should be covered with a uniqueness constraint.



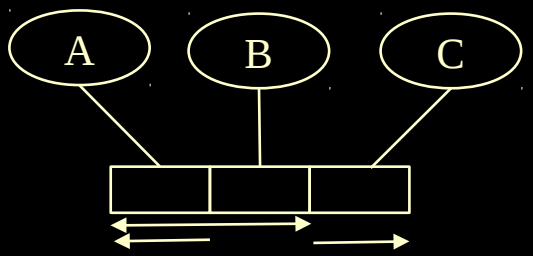
Summary

- **ORM is intended to provide desirable properties of a modelling language: expressibility, clarity, simplicity, semantic stability and relevance, validation mechanism and formal foundation**
- **ORM conceptual schema design procedure (CSDP) requires (steps1-3)**
  - Verbalisation of familiar information examples as facts (domain expert task)
  - Refine these into formal elementary facts (modeller task
  - Draw the fact types and apply population check
  - Check if entity types could be combined and note any (arithmetic) derivations
- **Uniqueness constraints (UC) are probably the most important constraints in the design of ORM schema representation of the UoD.**
- **Identification of UCs for fact types is based on the analysis of the data sample and should be confronted with the rules that are valid in the modelled UoD. Note: Sample data should be significant, however there is no mechanism to find out if the sample data satisfies that condition.**

Data sample is called significant if and only if there exixt one-to-one correspondence between set of constraints valid in the respective domain in the UoD and the set of constraints that can be deducted from the analysis of that sample.

- **Arity check is instrumental in finding correct model of the fact type. The result is not necessary one correct model. That could depend on the modeller approach (example):**
  - Student gets a result for a subject (ternary fact type)
  - Enrolment of a student in a subject (binary fact type nested) ends up in a result (binary fact type associating Enrolment and the result)
- **Splitting (based on projection-join check) is used to find out if detected fact type is too long.**
  - Understanding of semantics of the UoD can be used to eliminate the need for checking some splits however, some splits are beyond the intuition.
  - If fact type of longer arity has two or more roles not covered by the UC then it is definitely splitable. The modeller should actualy find out in earlier steps that in such case there is a conjunction of fact types with the same predicates on the roles covered by UC.
-

- **The ORM CSDP has background in the theory of normalisation. The term Functional Dependency (FD) is reflected in uniqueness constraints and the splitting in project-join procedure**
  - If the ORM conceptual schema is correct, then all FDs are implied by uniqueness constraints.
  - If a nonimplied FD X → Y exists then the predicate should be split on the source of X.
  - Each uniqueness constraint not covering all roles in a fact type implies an FD of which LHS has attributes generated by the roles covered by UC



FDs implied:

AB → C

AC → B