(Description) Logics

for Information Modelling and Access

Enrico Franconi

`franconi@cs.man.ac.uk`

`http://www.cs.man.ac.uk/~franconi`

Department of Computer Science, University of Manchester

Univ. of Liverpool – 22 January 2002

Summary

- Description Logics
- The role of logics in Information Systems
- Conceptual Modelling and Query Management

Description Logics – the standard view

- Expressive decidable fragments of (first-order) classical logic
- Close correspondence with modal logics (e.g., $\mathcal{ALC}$ vs. $\mathbf{K}$)
- Sound and complete algorithms implemented in efficient reasoners
- Knowledge representation formalism derived by semantic networks and frames in Artificial Intelligence
- Close correspondence with well known database conceptual data models

Knowledge representation is about objects

Description logics describe classes of objects (concepts) and their inter-relationships (roles).

The $\mathcal{ALC}$ concept expression

$$\mathtt{Professor} \sqcap \exists \mathtt{TEACHES}.\, \mathtt{UG\text{-}Course} \sqcap \forall \mathtt{TEACHES}.\, \mathtt{CS\text{-}Course}$$

corresponds to the $\mathbf{K}$ formula

$$Professor \wedge \Diamond UG\text{-}Course \wedge \Box CS\text{-}Course$$

where the accessibility relation is interpreted as the *TEACHES* relation

Description Logics are multi-modal

The $\mathcal{ALC}$ concept expression

$\texttt{Professor} \sqcap \exists\texttt{TEACHES}.\,\texttt{UG-Course} \sqcap \exists\texttt{DEGREE}.\,\texttt{Bs}$

corresponds to the $\mathbf{K_m}$ formula (over the same object domain)

$Professor \wedge \Diamond_{TEACHES}\, UG\text{-}Course \wedge \Diamond_{DEGREE}\, Bs$

Modalities (as roles) may have different properties

The $\mathcal{ALC}$ concept expression

$$\texttt{Professor} \sqcap \exists \texttt{TEACHES}.\, \texttt{UG-Course} \sqcap$$
$$\exists \texttt{IS-PART}.\, (\texttt{Staff} \sqcap \exists \texttt{IS-LOCATED}.\, \texttt{Department})$$

corresponds to the $\mathbf{K_m} \cup \mathbf{K4_m}$ formula (over the same object domain)

$$Professor \wedge \Diamond_{TEACHES} UG\text{-}Course \wedge$$
$$\Diamond_{IS\text{-}PART}(Staff \wedge \Diamond_{IS\text{-}LOCATED} Department)$$

where *TEACHES* is a $\mathbf{K_m}$ modality and *IS-PART*, *IS-LOCATED* are $\mathbf{K4_m}$ modalities

Relational structures

- “Modal logics are not appropriate as a representational tool since they do not always capture the details of the models”
- Do we care?

Relational structures

- "Modal logics are not appropriate as a representational tool since they do not always capture the details of the models"
- Do we care?
- No: this is not a bug, it is a feature!

  Why?

Relational structures

- “Modal logics are not appropriate as a representational tool since they do not always capture the details of the models”
- Do we care?
- No: this is not a bug, it is a feature!
  Why?
- If a formula is satisfiable in a model, it is also satisfiable in a model with the indistinguishable property. Hence, reasoning is not affected.

Additional genuine modalities

- Time, space, belief, etc: combination of modal logics over *distinct* domains (i.e., the object and the modal domains)
- Example of $\mathcal{ALC} \cup \mathsf{LTL}$ concept expression:

  $\texttt{Professor} \sqcap \Diamond(\exists \texttt{TEACHES}.\, \texttt{UG-Course}) \sqcap \forall \Box \texttt{TEACHES}.\, \texttt{CS-Course}$
- Asymmetric extension

Global axioms

$$
\begin{aligned}
\exists \mathtt{TEACHES}.\,\mathtt{Course} &\sqsubseteq (\mathtt{Student} \sqcap \exists \mathtt{DEGREE}.\,\mathtt{Bs}) \sqcup \mathtt{Prof} \\
\mathtt{Prof} &\sqsubseteq \exists \mathtt{DEGREE}.\,\mathtt{Ms} \\
\exists \mathtt{DEGREE}.\,\mathtt{Ms} &\sqsubseteq \exists \mathtt{DEGREE}.\,\mathtt{Bs} \\
\mathtt{Ms} \sqcap \mathtt{Bs} &\sqsubseteq \bot
\end{aligned}
$$

- Axioms should be satisfied by each object in the domain
- Satisfiability and logical implication in $\mathcal{ALC}$ ($\mathbf{K_m}$) become EXPTIME-complete

Global axioms, II

- $\mathbf{K}_\mathbf{m}^{\mathcal{H}}$ extends $\mathbf{K}_\mathbf{m}$ with statements on inclusions between modalities
- Decision problems for $\mathbf{K}_\mathbf{m}^{\mathcal{H}}$ and $\mathbf{K4}_\mathbf{m}$ are in PSPACE
- The *universal modality* can be encoded in $\mathbf{K}_\mathbf{m}^{\mathcal{H}} \cup \mathbf{K4}_\mathbf{m}$, and axioms can be internalised:
  - Define new *transitive* modality U that *includes* all other modalities
  - Satisfiability of ϕ w.r.t. $\psi_1 \rightarrow \varphi_1, \ldots, \psi_n \rightarrow \varphi_n$ is equivalent to satisfiability of $\phi \wedge \Box_U((\psi_1 \rightarrow \varphi_1) \wedge \ldots \wedge (\psi_n \rightarrow \varphi_n))$
- Satisfiability and logical implication in $\mathbf{K}_\mathbf{m}^{\mathcal{H}} \cup \mathbf{K4}_\mathbf{m}$ are EXPTIME-complete
- FaCT implements $\mathbf{K}_\mathbf{m}^{\mathcal{H}} \cup \mathbf{K4}_\mathbf{m}$

n-ary Relations

- Relations between objects in the world may necessarily involve more than just two objects
- Full fledged relational structures are needed, beyond Kripke structures
- We want to maintain the modal logic flavour
- $\mathcal{DLR}$ properly extends $\mathcal{ALC}$ with n-ary relations

$\mathcal{DLR}$

$$R \rightarrow \top_n \mid RN \mid \neg R \mid R_1 \sqcap R_2 \mid R_1 \sqcup R_2 \mid U_i/n : C$$

$$C \rightarrow \top \mid CN \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists[U_i]R \mid \exists^{\leqslant k}[U_i]R$$

$\texttt{Works-for} \sqsubseteq \texttt{subj}/2 : \texttt{Employee} \sqcap \texttt{obj}/2 : \texttt{Project}$

$\texttt{Manager} \sqsubseteq \texttt{Employee} \sqcap \neg\exists[\texttt{subj}]\texttt{Works-for}$

$\mathcal{DLR}$ includes $\mathcal{ALCQI}$: if R is a binary relation (i.e., a *role*) with named attributes *first* and *second* then

$\exists R.\, C \equiv \exists[\textit{first}](R \sqcap (\textit{second}/2 : C))$

Reasoning in $\mathcal{DLR}$ is EXPTIME-complete

$\mathcal{DLR}$ syntax	$\mathcal{DLR}$ semantics	RD encoding
$\top_n$	$\top_n^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$	$\top_n$
RN	$\mathit{RN}^{\mathcal{I}} \subseteq \top_n^{\mathcal{I}}$	C_{RN}
$\neg R$	$\top_n^{\mathcal{I}} \setminus R^{\mathcal{I}}$	$\neg C_R$
$R_1 \sqcap R_2$	$R_1{}^{\mathcal{I}} \cap R_2{}^{\mathcal{I}}$	$C_{R_1} \sqcap C_{R_2}$
$U_i/n : C$	$\{\langle d_1, \ldots, d_n \rangle \in \top_n^{\mathcal{I}} \mid d_i \in C^{\mathcal{I}}\}$	$\top_n \sqcap \forall U_i .\, C$
$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$	$\top$
CN	$\mathit{CN}^{\mathcal{I}} \subseteq \top^{\mathcal{I}}$	CN
$\neg C$	$\top^{\mathcal{I}} \setminus C^{\mathcal{I}}$	$\neg C$
$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$	$C_1 \sqcap C_2$
$\exists[U_i]R$	$\{d \in \top^{\mathcal{I}} \mid \exists \langle d_1, \ldots, d_n \rangle \in R.\ d_i = d\}$	$\exists U_i^- .\, C_R$
$\exists^{\leqslant k}[U_i]R$	$\{d \in \top^{\mathcal{I}} \mid \sharp\{\langle d_1, \ldots, d_n \rangle \in R \mid d_i = d\} \leqslant k\}$	$\leqslant k U_i^- .\, C_R$

Encoding conceptual data models in $\mathcal{DLR}$

- Object-oriented data models (e.g., UML and ODMG)
- Semantic data models (e.g., EER and ORM)
- Frame-based ontology languages (e.g., OIL)

Encoding conceptual data models in $\mathcal{DLR}$

- Object-oriented data models (e.g., UML and ODMG)
- Semantic data models (e.g., EER and ORM)
- Frame-based ontology languages (e.g., OIL)

- Theorems **prove** that a conceptual schema and its encoding as $\mathcal{DLR}$ inclusion dependencies constrain every database state in the same way – i.e., the models of the $\mathcal{DLR}$ theory correspond to the legal database states of the conceptual schema, and vice-versa.

Classical Integrity Constraints in $\mathcal{DLR}$

- arbitrary *boolean constructs*
- *unary inclusion* dependencies (e.g., referential integrity)
- special forms of *typed inclusion* dependencies
- *existence* and *exclusion* dependencies
- *unary functional* dependencies
- *view definitions*

Extensions of $\mathcal{DLR}$

$\mathcal{DLR}_{reg}$: regular expressions and recursive views (beyond FOL)

$\mathcal{DLR}_{\mathcal{US}}$: combination with temporal constructs to model temporal databases

$\mathcal{DLR}_{key}$: general key constraints

Queries under $\mathcal{DLR}$ constraints

- A query is an open FOL formula, whose predicates may be constrained by a $\mathcal{DLR}$ theory
- We consider only the conjunctive existential fragment (the conjunctive queries, or non-recursive datalog queries)
- Example:

$$\begin{aligned}\mathtt{Q}_1(\mathtt{x},\mathtt{y}) \text{ :- } & (\neg\mathtt{Professor})(\mathtt{x}) \wedge \mathtt{TEACHES}(\mathtt{x},\mathtt{y}) \wedge \\ & (\mathtt{UG\text{-}Course} \sqcup \mathtt{CS\text{-}Course})(\mathtt{x})\end{aligned}$$

Semantics of Evaluation and Containment

The *evaluation* of a query $\mathcal{Q}$ of arity n given a $\mathcal{DLR}$ theory Σ over a model $\mathcal{I}$ satisfying Σ

Semantics of Evaluation and Containment

The *evaluation* of a query $\mathcal{Q}$ of arity n given a $\mathcal{DLR}$ theory Σ over a model $\mathcal{I}$ satisfying Σ

is the set $\text{ANS}(\mathcal{Q}, \mathcal{I})$ composed by n-tuples $\overrightarrow{o}$ such that

Semantics of Evaluation and Containment

The *evaluation* of a query $\mathcal{Q}$ of arity n given a $\mathcal{DLR}$ theory Σ over a model $\mathcal{I}$ satisfying Σ

is the set $\textsc{ans}(\mathcal{Q}, \mathcal{I})$ composed by n-tuples $\overrightarrow{o}$ such that

$$\textsc{ans}(\mathcal{Q}, \mathcal{I}) = \{\overrightarrow{o} \mid \mathcal{I} \models \bigvee_j \exists \overrightarrow{y_j}.\, \mathcal{Q}_j(\overrightarrow{o}, \overrightarrow{y_j}, \overrightarrow{c_j})\}$$

Semantics of Evaluation and Containment

The *evaluation* of a query $\mathcal{Q}$ of arity n given a $\mathcal{DLR}$ theory Σ over a model $\mathcal{I}$ satisfying Σ

is the set ANS$(\mathcal{Q}, \mathcal{I})$ composed by n-tuples $\overrightarrow{o}$ such that

$$\text{ANS}(\mathcal{Q}, \mathcal{I}) = \{\overrightarrow{o} \mid \mathcal{I} \models \bigvee_j \exists \overrightarrow{y_j}.\, \mathcal{Q}_j(\overrightarrow{o}, \overrightarrow{y_j}, \overrightarrow{c_j})\}$$

$\mathcal{Q}_1$ *is contained* in $\mathcal{Q}_2$ with respect to a $\mathcal{DLR}$ theory Σ

Semantics of Evaluation and Containment

The *evaluation* of a query $\mathcal{Q}$ of arity n given a $\mathcal{DLR}$ theory Σ

over a model $\mathcal{I}$ satisfying Σ

is the set $\text{ANS}(\mathcal{Q}, \mathcal{I})$ composed by n-tuples $\overrightarrow{o}$ such that

$$\text{ANS}(\mathcal{Q}, \mathcal{I}) = \{\overrightarrow{o} \mid \mathcal{I} \models \bigvee_j \exists \overrightarrow{y_j}. \mathcal{Q}_j(\overrightarrow{o}, \overrightarrow{y_j}, \overrightarrow{c_j})\}$$

$\mathcal{Q}_1$ *is contained* in $\mathcal{Q}_2$ with respect to a $\mathcal{DLR}$ theory Σ

if and only if for every model $\mathcal{I}$ satisfying Σ

$$\text{ANS}(\mathcal{Q}_1, \mathcal{I}) \subseteq \text{ANS}(\mathcal{Q}_2, \mathcal{I})$$

Semantics of Evaluation and Containment

The *evaluation* of a query $\mathcal{Q}$ of arity n given a $\mathcal{DLR}$ theory Σ

over a model $\mathcal{I}$ satisfying Σ

is the set $\text{ANS}(\mathcal{Q},\mathcal{I})$ composed by n-tuples $\overrightarrow{o}$ such that

$$\text{ANS}(\mathcal{Q},\mathcal{I}) = \{\overrightarrow{o} \mid \mathcal{I} \models \bigvee_j \exists \overrightarrow{y_j}.\, \mathcal{Q}_j(\overrightarrow{o}, \overrightarrow{y_j}, \overrightarrow{c_j})\}$$

$\mathcal{Q}_1$ *is contained* in $\mathcal{Q}_2$ with respect to a $\mathcal{DLR}$ theory Σ

if and only if for every model $\mathcal{I}$ satisfying Σ

$$\text{ANS}(\mathcal{Q}_1,\mathcal{I}) \subseteq \text{ANS}(\mathcal{Q}_2,\mathcal{I})$$

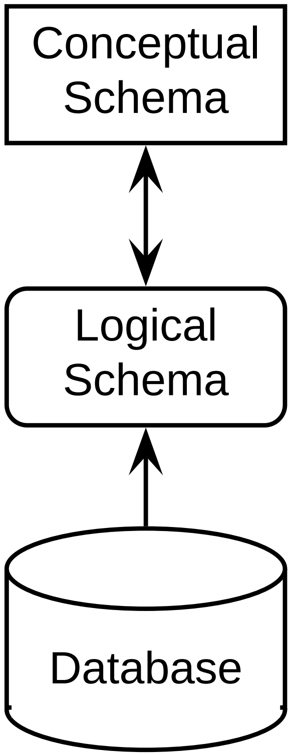
Containment of disjunctions of conjunctive queries under $\mathcal{DLR}$ ($\mathcal{DLR}_{\mathcal{US}}$) constraints is decidable in 2EXPTIME

The i•com tool for Intelligent Conceptual Modelling

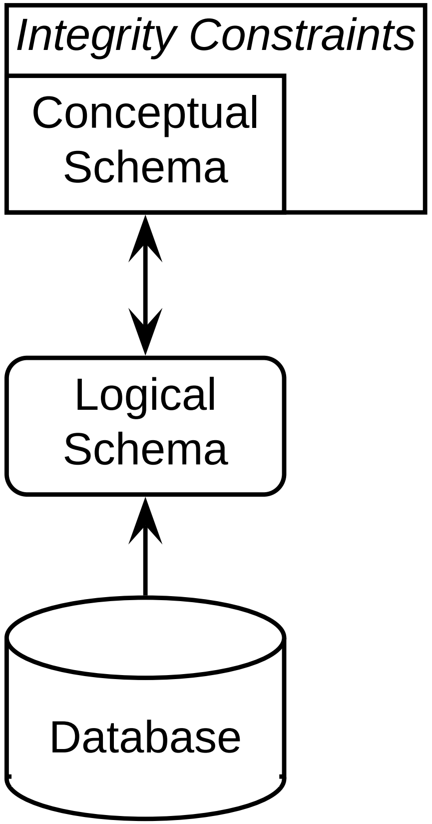
- i•com is an advanced CASE tool which allows the user to design *multiple* extended Entity-Relationship schemas or UML class diagrams with inter- and intra-schema *constraints*.
- Complete logical reasoning is employed by the tool to:
  - verify the specification,
  - infer implicit facts,
  - devise stricter constraints,
  - and manifest any local inconsistency.

`http://www.cs.man.ac.uk/~franconi/icom/`

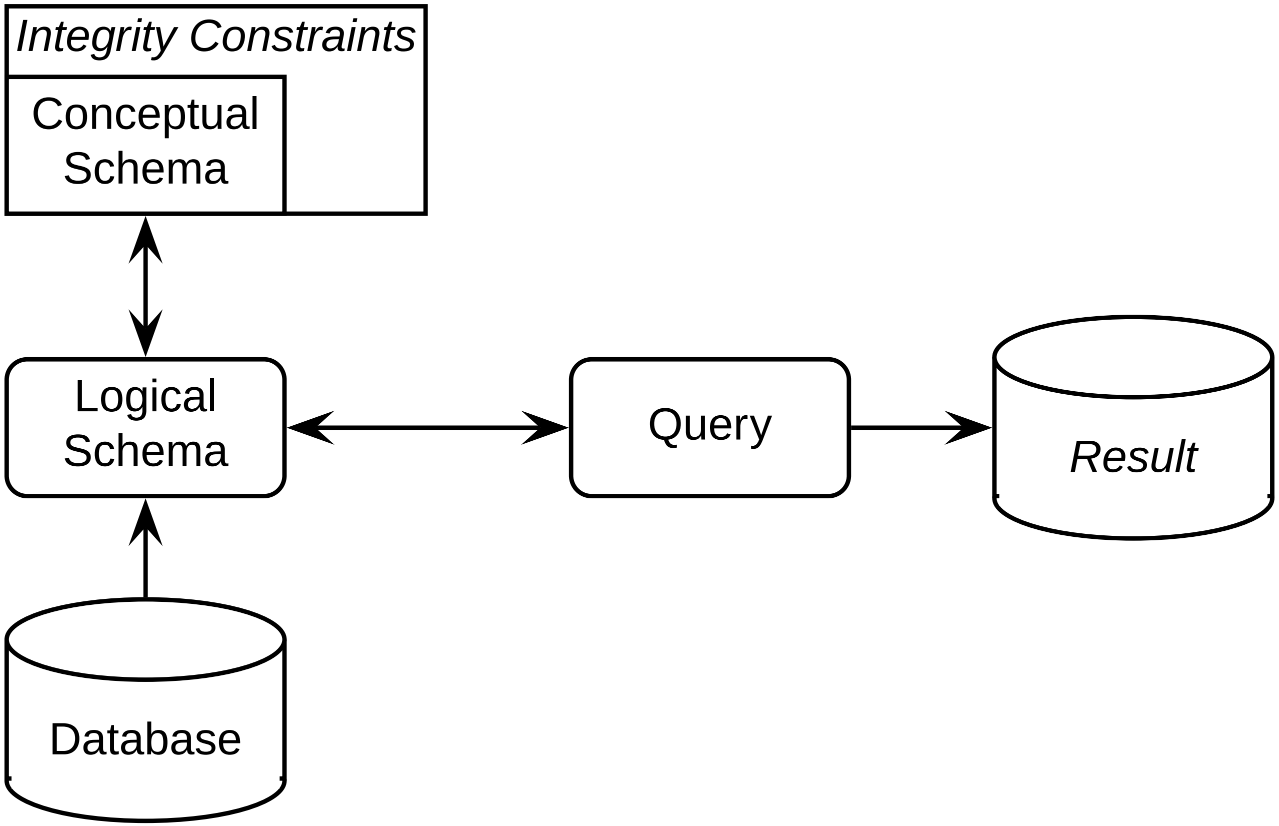
The role of logics in Information Systems



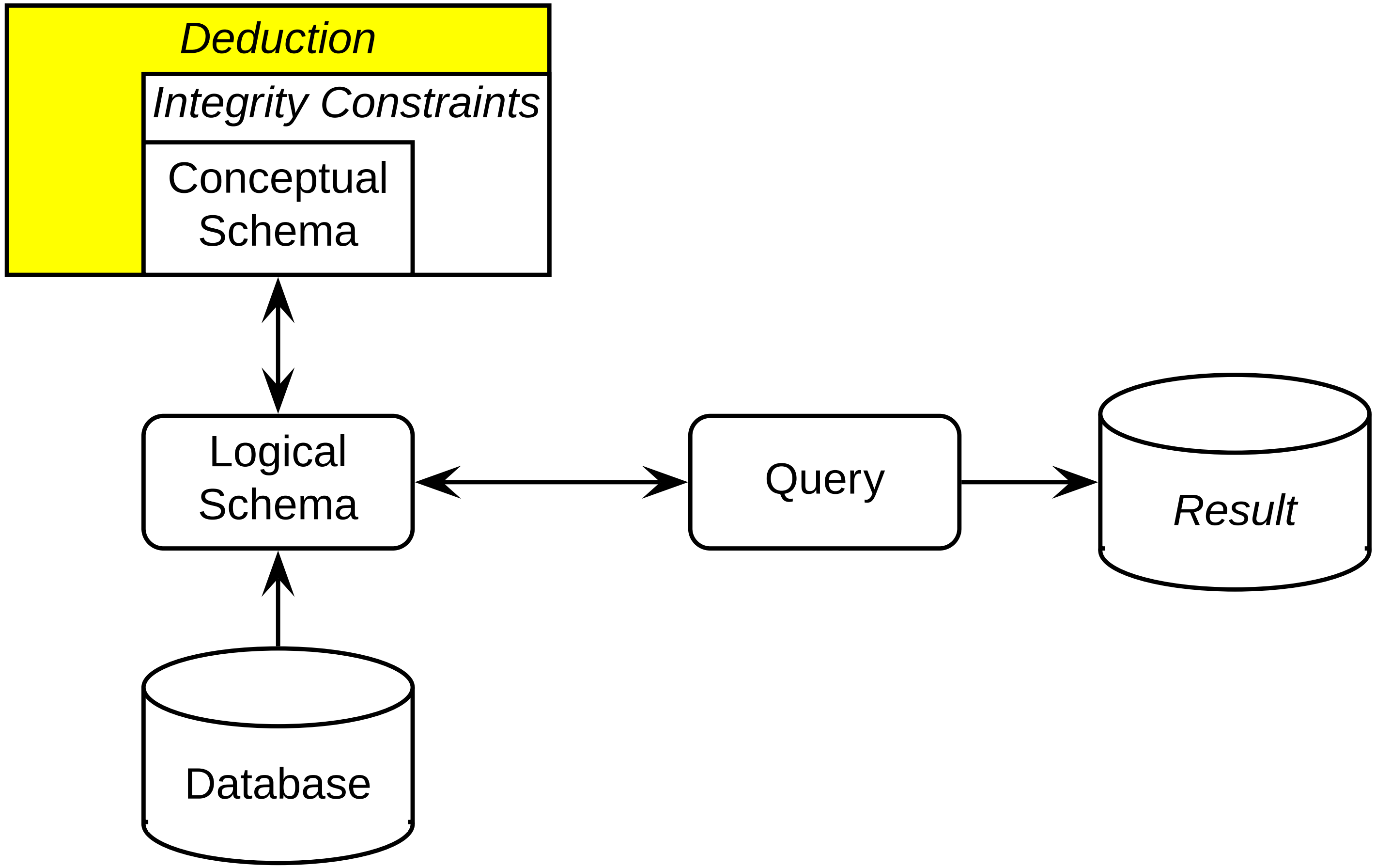
The role of logics in Information Systems



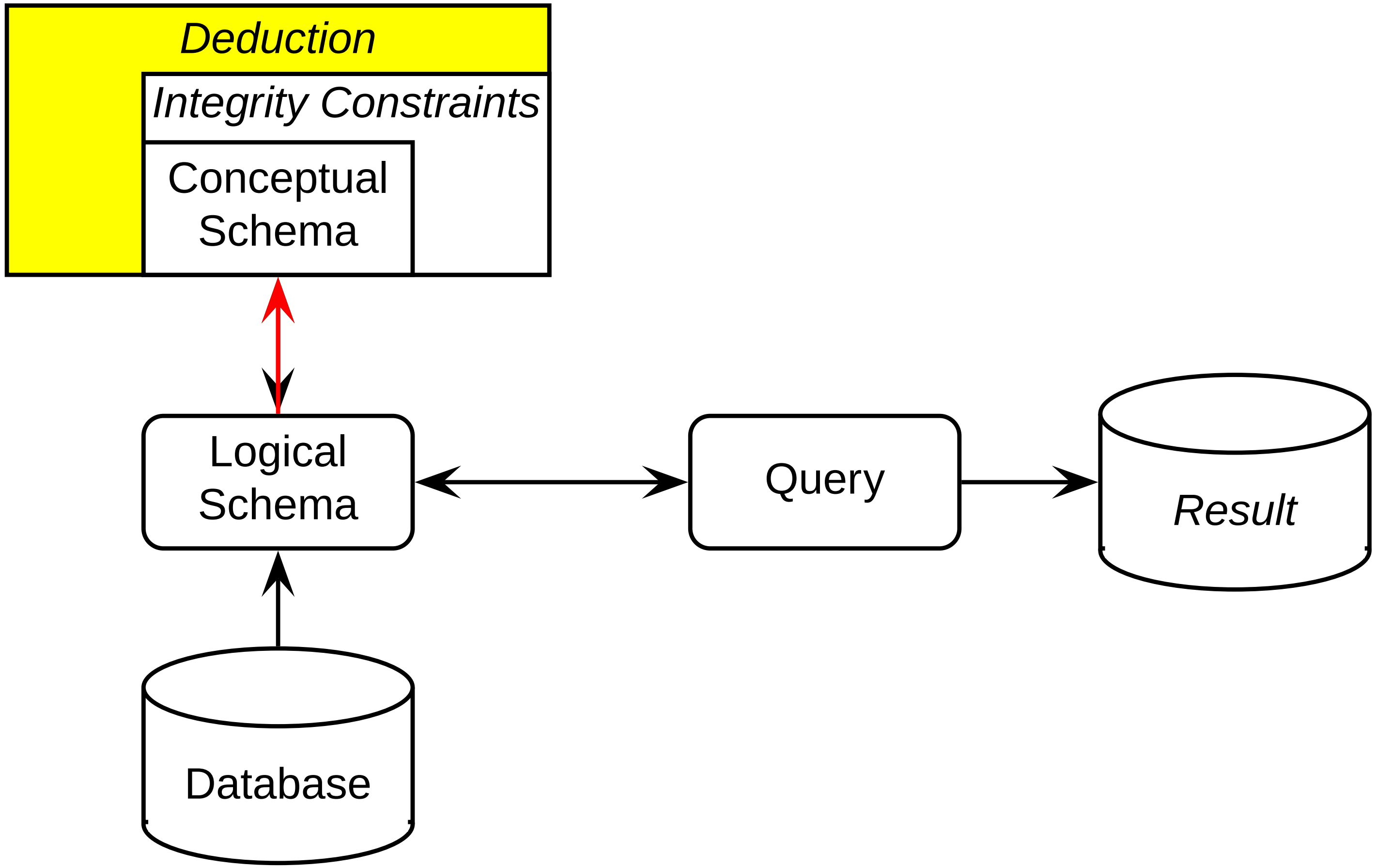
The role of logics in Information Systems



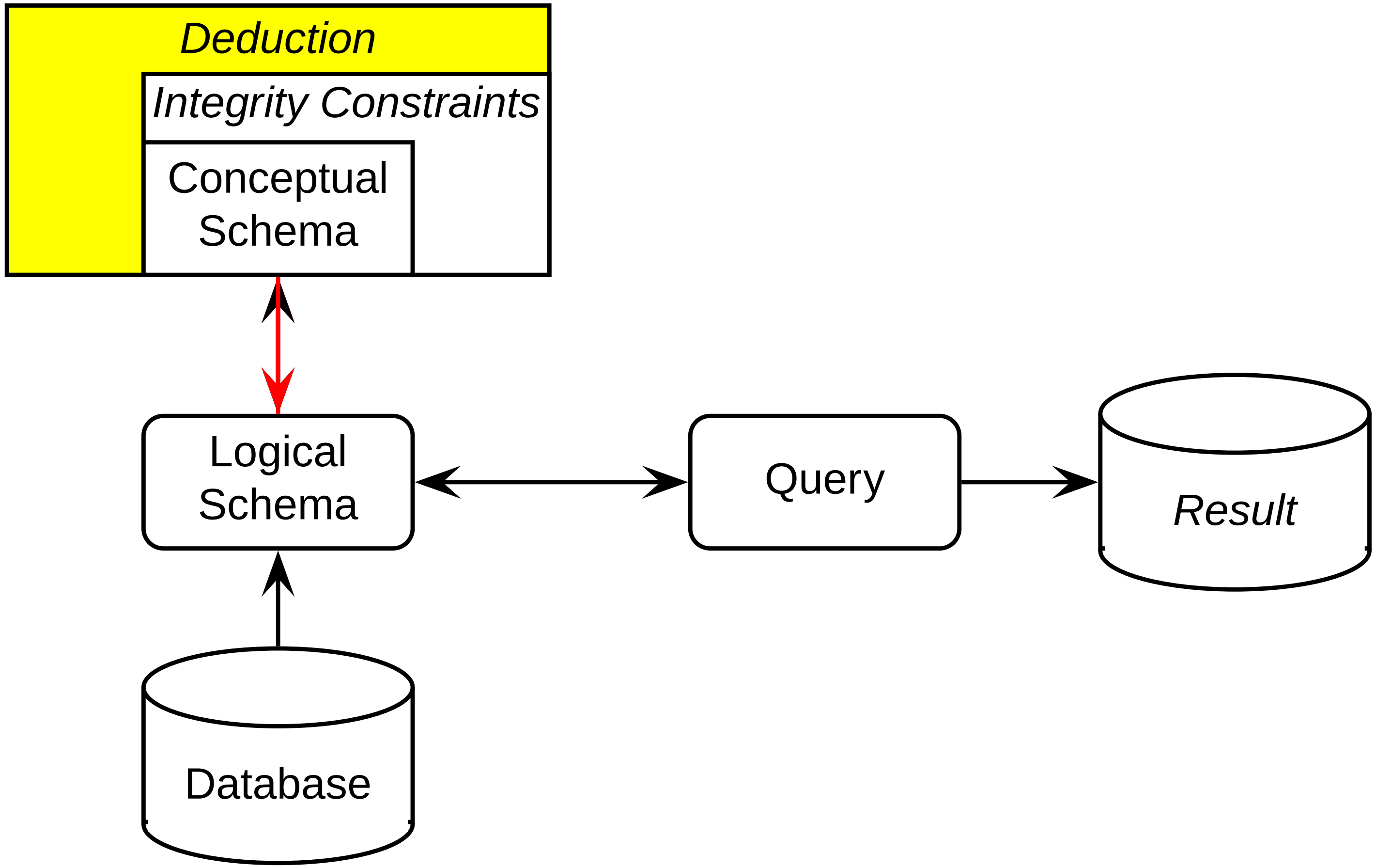
The role of logics in Information Systems



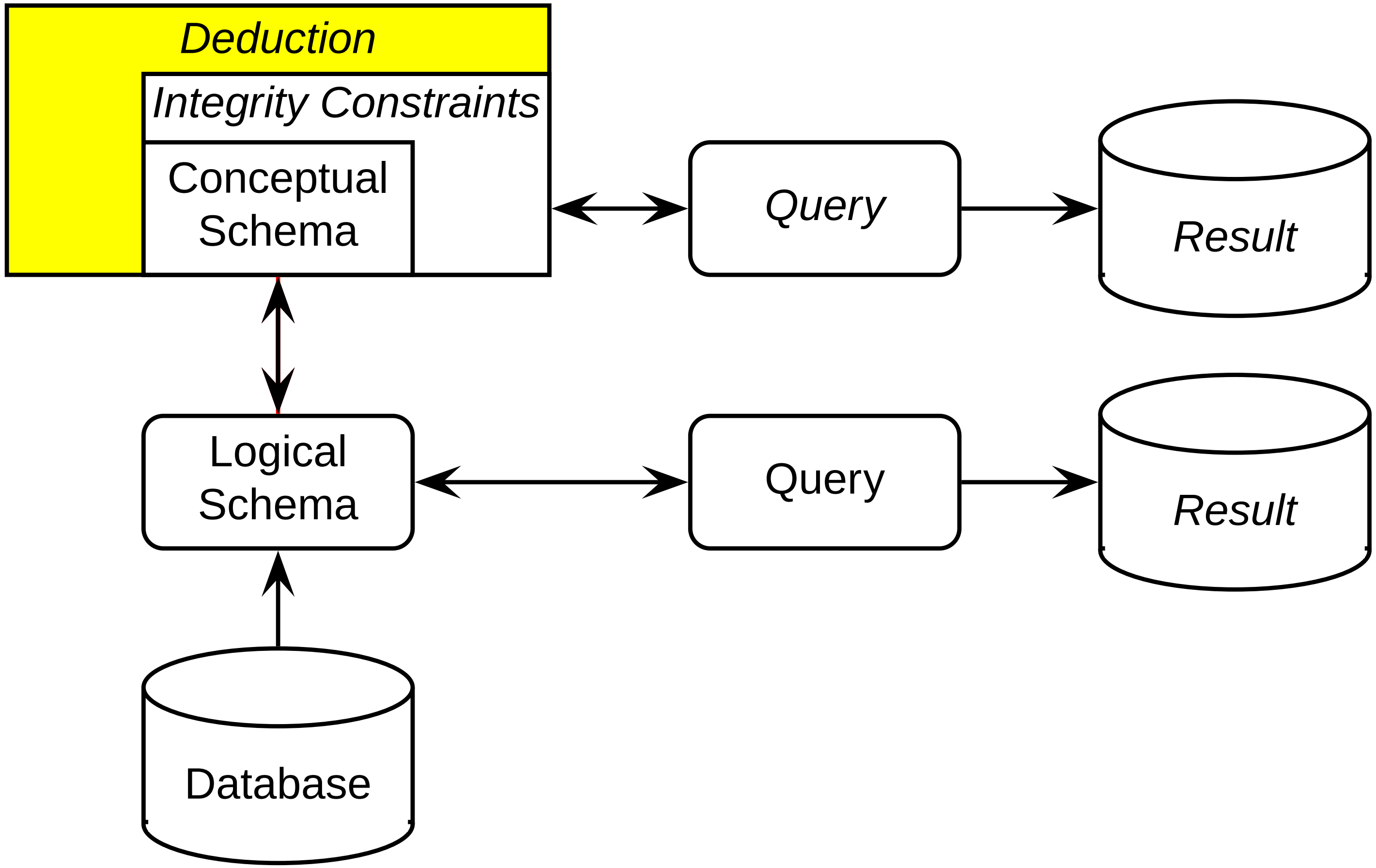
The role of logics in Information Systems



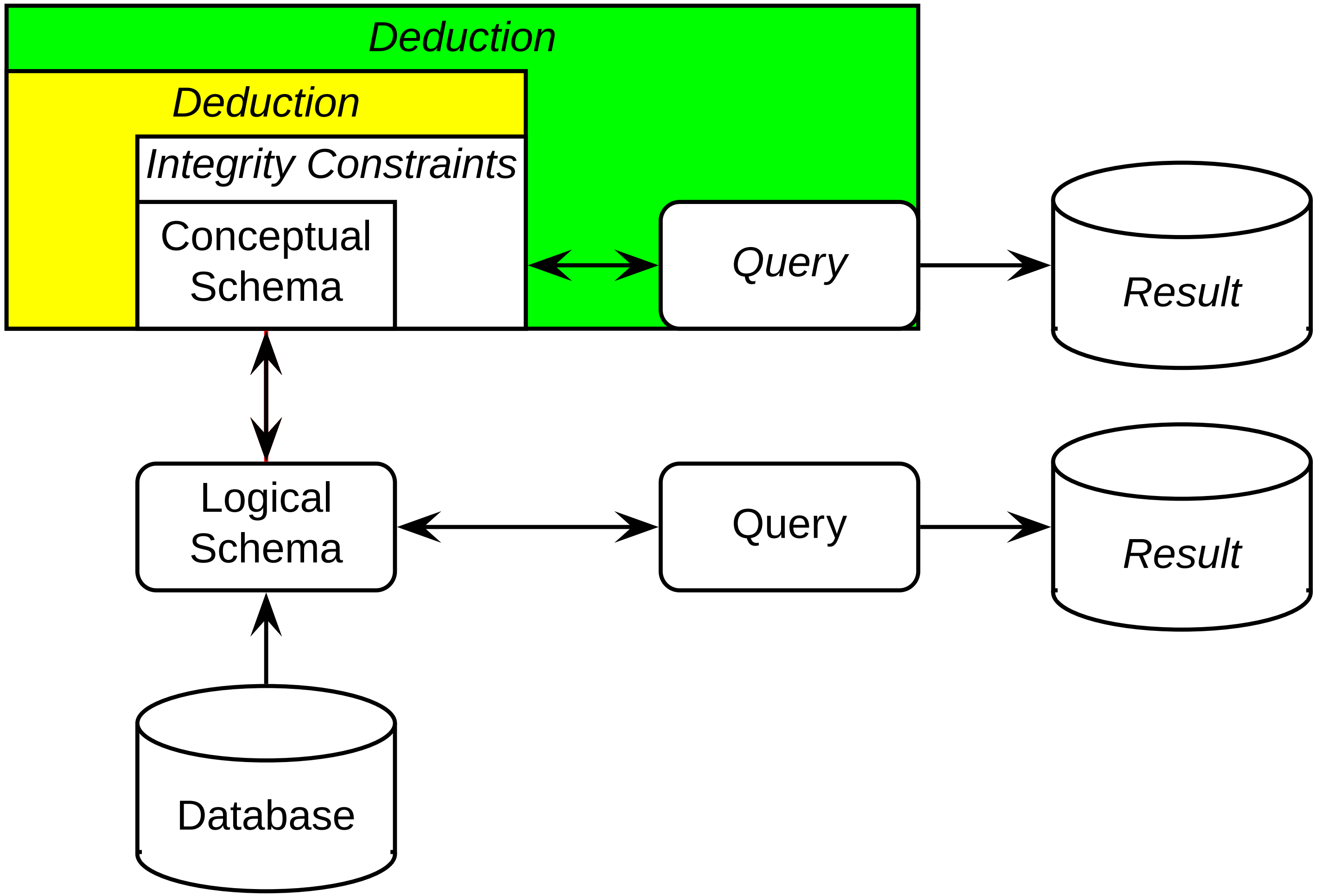
The role of logics in Information Systems



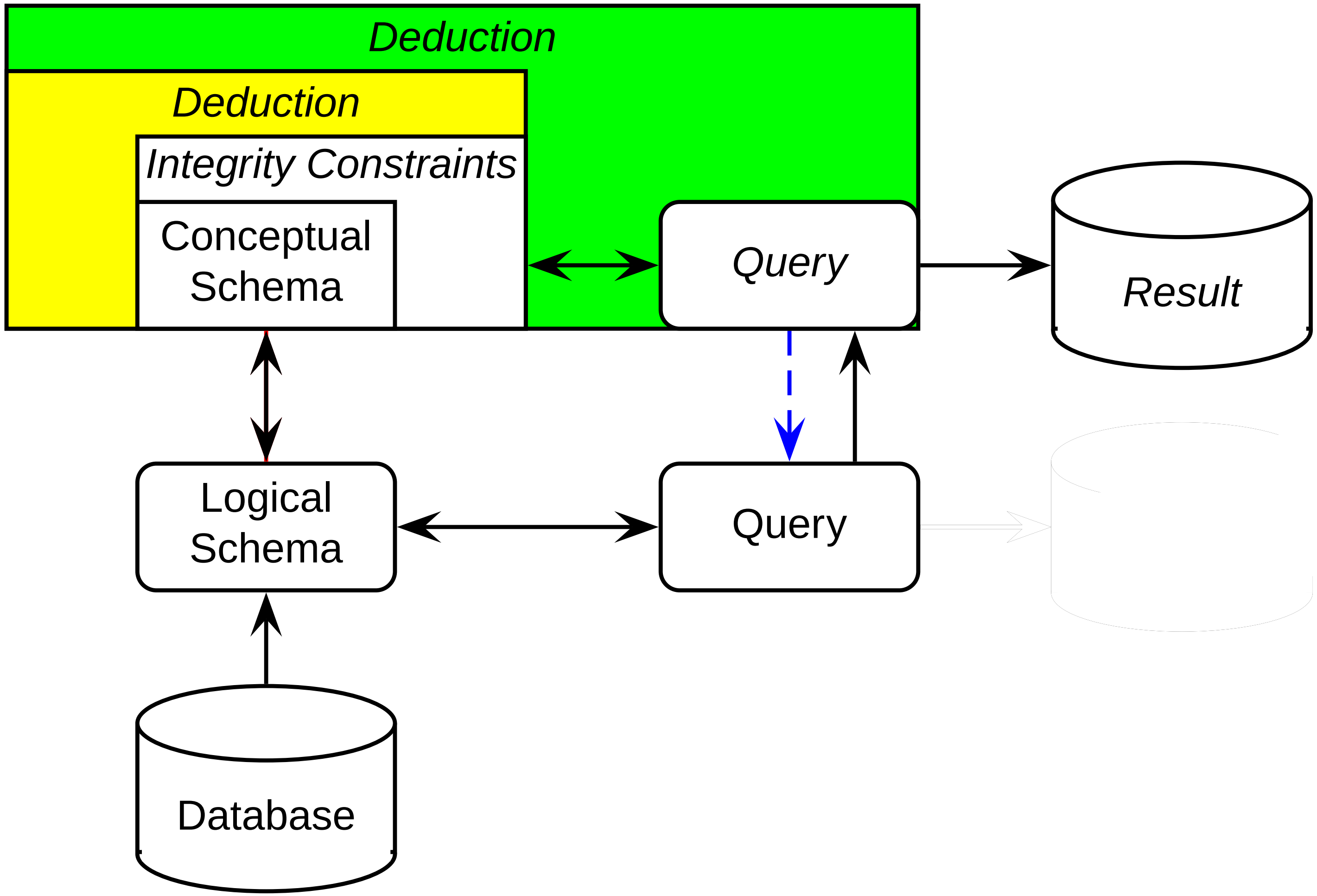
The role of logics in Information Systems



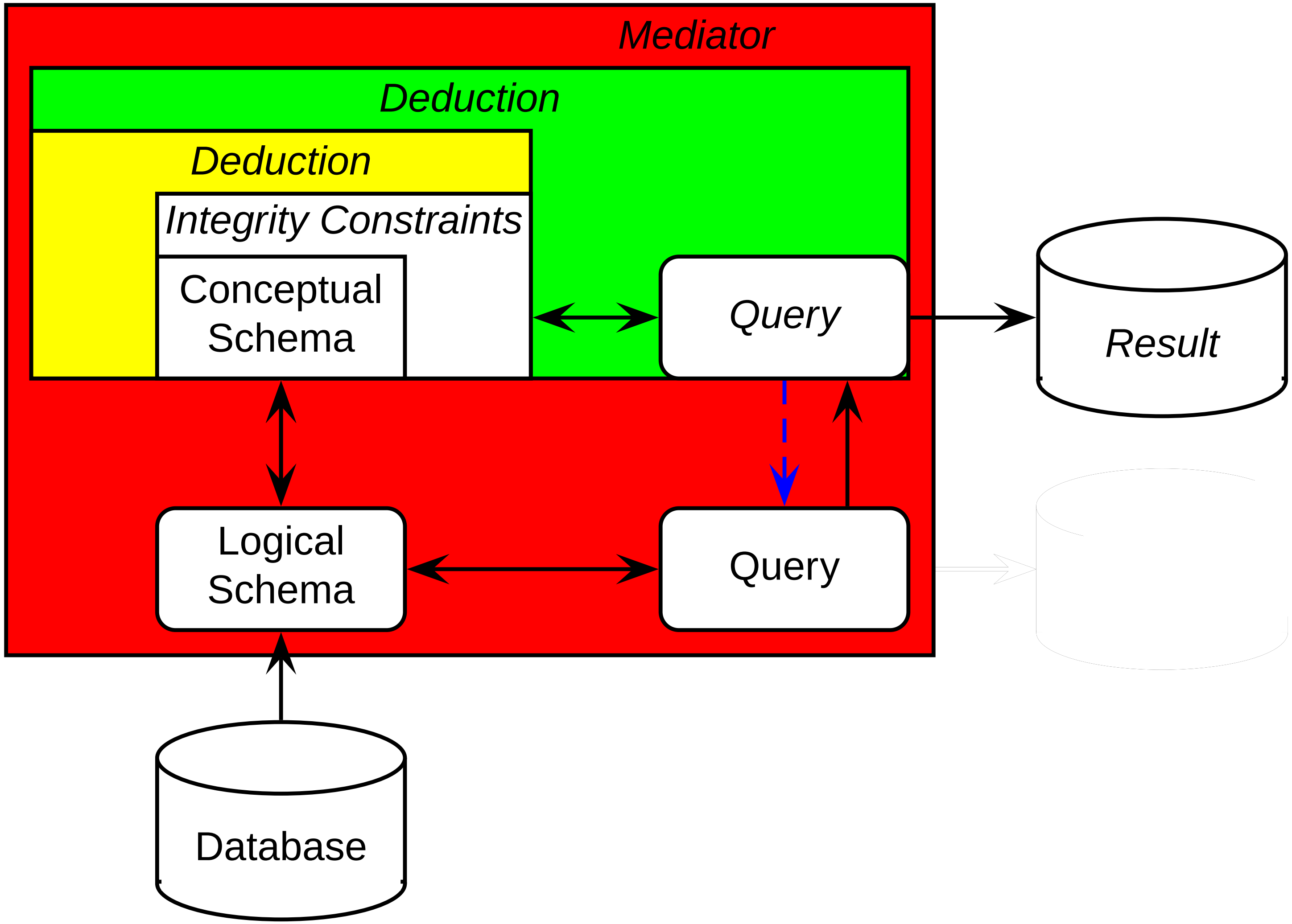
The role of logics in Information Systems



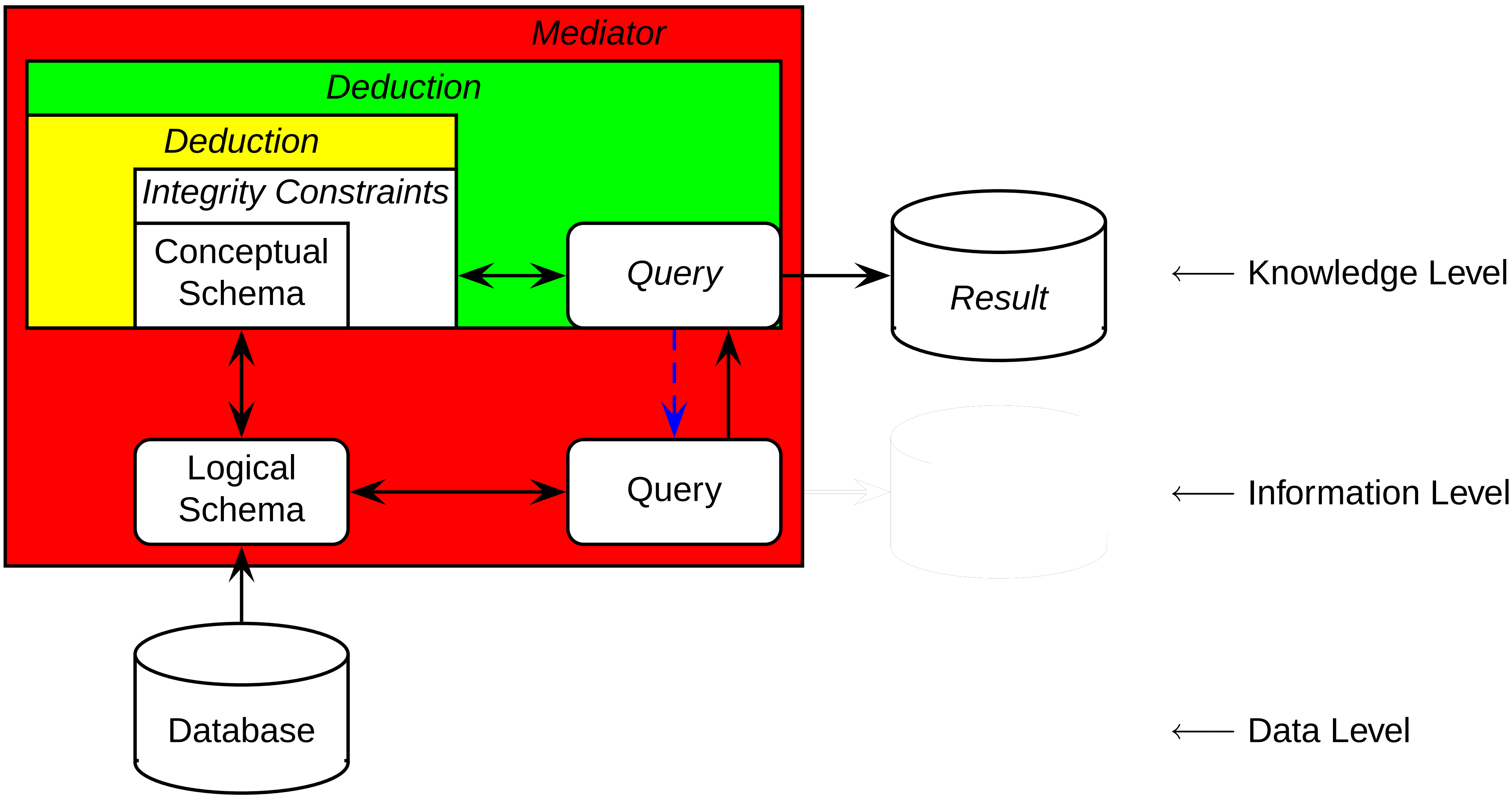
The role of logics in Information Systems



The role of logics in ~~Information Systems~~ a Mediator



The role of logics in ~~Information Systems~~ a Mediator



A Relational Database

$\texttt{CompanyEmployee}/2; \texttt{CompanyProject}/3$

CompanyEmployee	
name	project
john	esprit-dwq
...	...

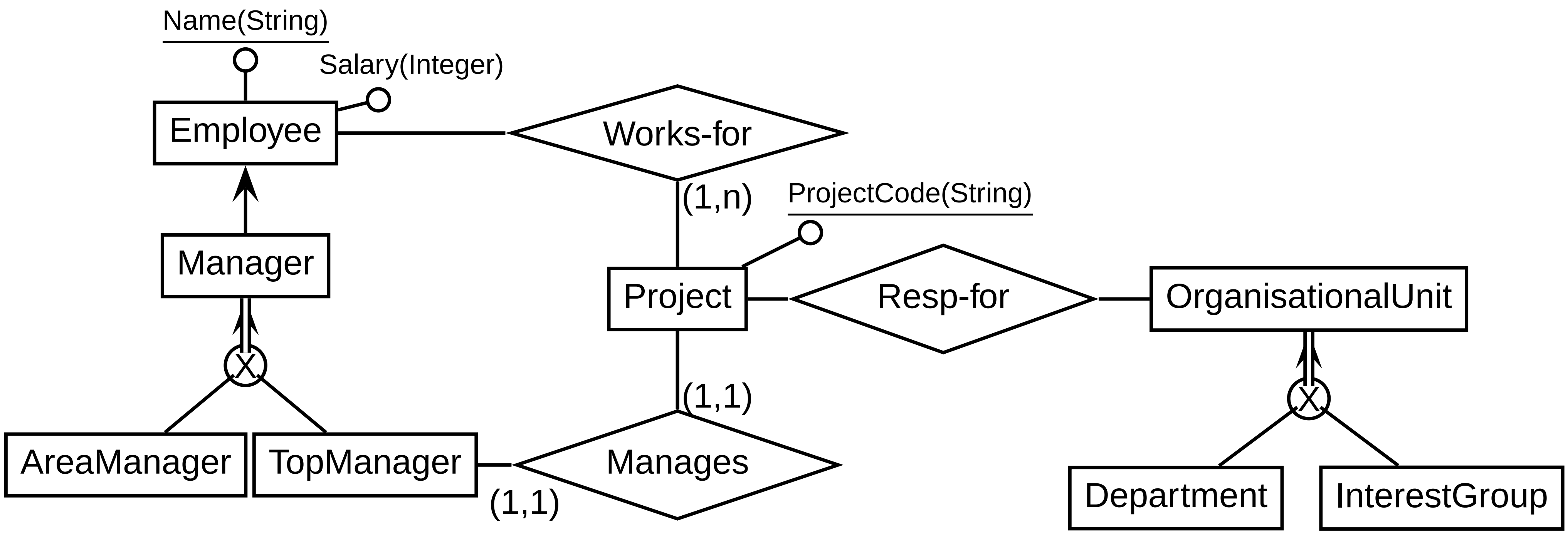
CompanyProject		
project	manager	department
esprit-dwq	enrico	cs-uman
...	...	...

Query = "Tell me the projects in which John works, and their managers and departments."

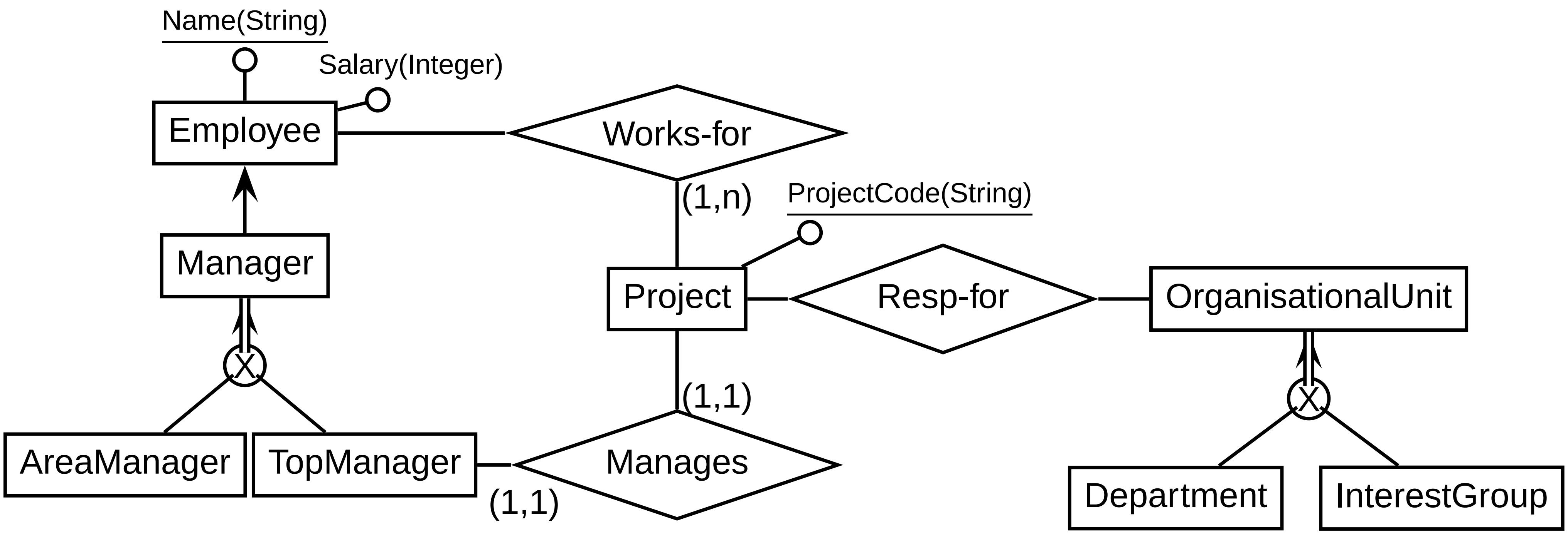
$$\texttt{Query} \equiv$$
$$\pi_{\texttt{project},\texttt{manager},\texttt{dept}.}\,\sigma_{\texttt{name}=\texttt{john}}\,(\texttt{CompanyEmployee} \bowtie_{\texttt{project}} \texttt{CompanyProject})$$

$$\texttt{Query}(\texttt{x},\texttt{y},\texttt{z}) \Leftrightarrow \texttt{CompanyEmployee}(\texttt{john},\texttt{x}) \wedge \texttt{CompanyProject}(\texttt{x},\texttt{y},\texttt{z})$$

Constraints from the Conceptual Schema



Constraints from the Conceptual Schema



$\mathtt{Works\text{-}for} \sqsubseteq \mathtt{emp/2} : \mathtt{Employee} \sqcap \mathtt{act/2} : \mathtt{Project}$

$\mathtt{Manages} \sqsubseteq \mathtt{man/2} : \mathtt{TopManager} \sqcap \mathtt{prj/2} : \mathtt{Project}$

$\mathtt{Employee} \sqsubseteq \exists^{=1}[\mathtt{worker}](\mathtt{Name} \sqcap \mathtt{thename/2} : \mathtt{String}) \sqcap \exists^{=1}[\mathtt{payee}](\mathtt{Salary} \sqcap \mathtt{amount/2} : \mathtt{Integer})$

$\top \sqsubseteq \exists^{\leq 1}[\mathtt{thename}](\mathtt{Name} \sqcap \mathtt{worker/2} : \mathtt{Employee})$

$\mathtt{Manager} \sqsubseteq \mathtt{Employee} \sqcap (\mathtt{AreaManager} \sqcup \mathtt{TopManager})$

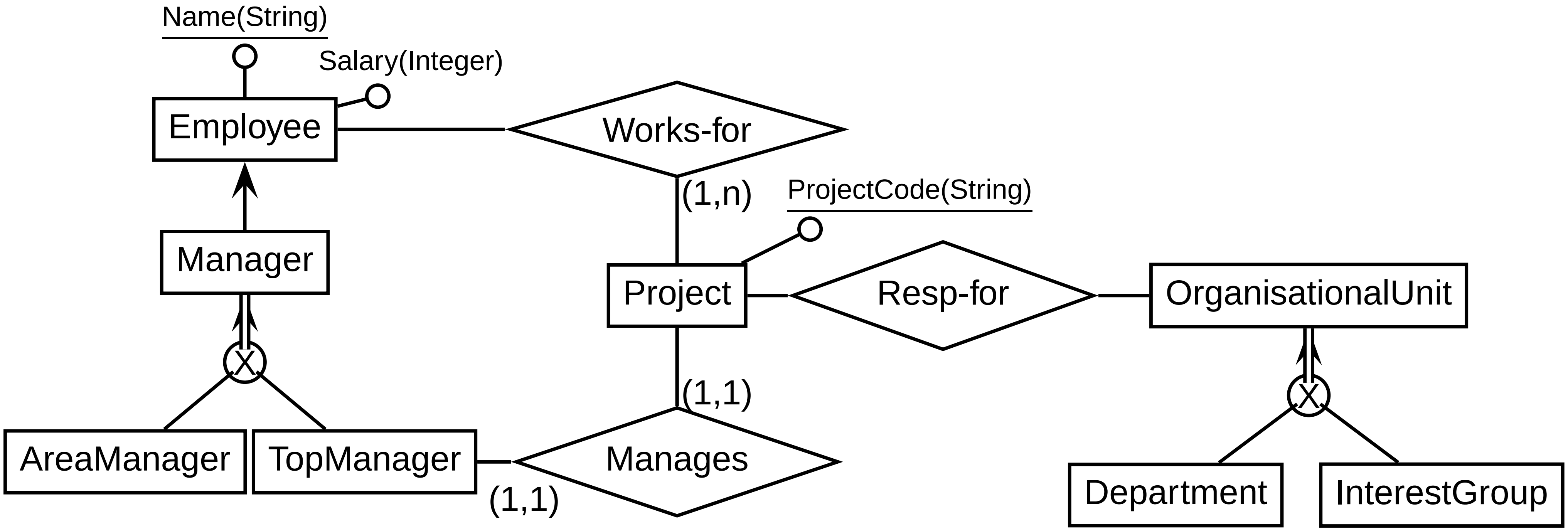
$\mathtt{AreaManager} \sqsubseteq \mathtt{Manager} \sqcap \neg\mathtt{TopManager}$

$\mathtt{TopManager} \sqsubseteq \mathtt{Manager} \sqcap \exists^{=1}[\mathtt{man}]\mathtt{Manages}$

$\mathtt{Project} \sqsubseteq \exists^{\geq 1}[\mathtt{act}]\mathtt{Works\text{-}for} \sqcap \exists^{=1}[\mathtt{prj}]\mathtt{Manages}$

...

Integrity Constraints and Logical Implication

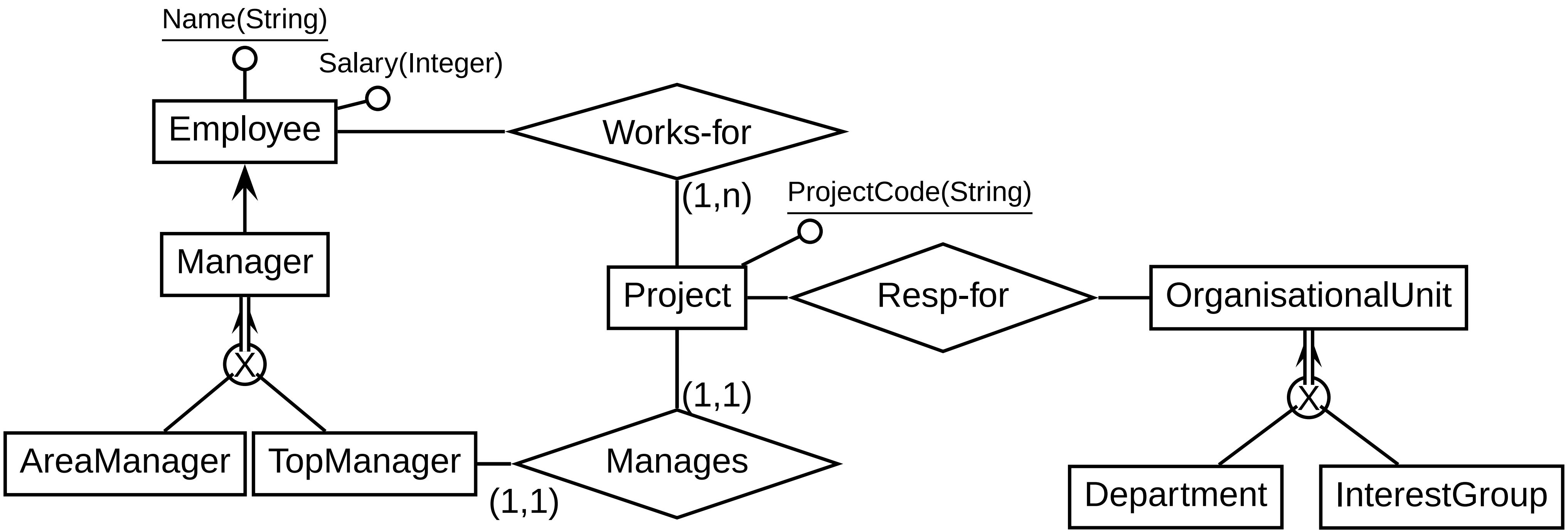


Managers are employees who do not work for a project (she/he just manages it):

$$\mathtt{Employee} \sqcap \neg(\exists^{\geq 1}[\mathtt{emp}]\mathtt{Works\text{-}for}) \sqsubseteq \mathtt{Manager}$$

$$\mathtt{Manager} \sqsubseteq \neg(\exists^{\geq 1}[\mathtt{emp}]\mathtt{Works\text{-}for})$$

Integrity Constraints and Logical Implication



Managers are employees who do not work for a project (she/he just manages it):

$$\texttt{Employee} \sqcap \neg(\exists^{\geq 1}[\texttt{emp}]\texttt{Works-for}) \sqsubseteq \texttt{Manager}$$

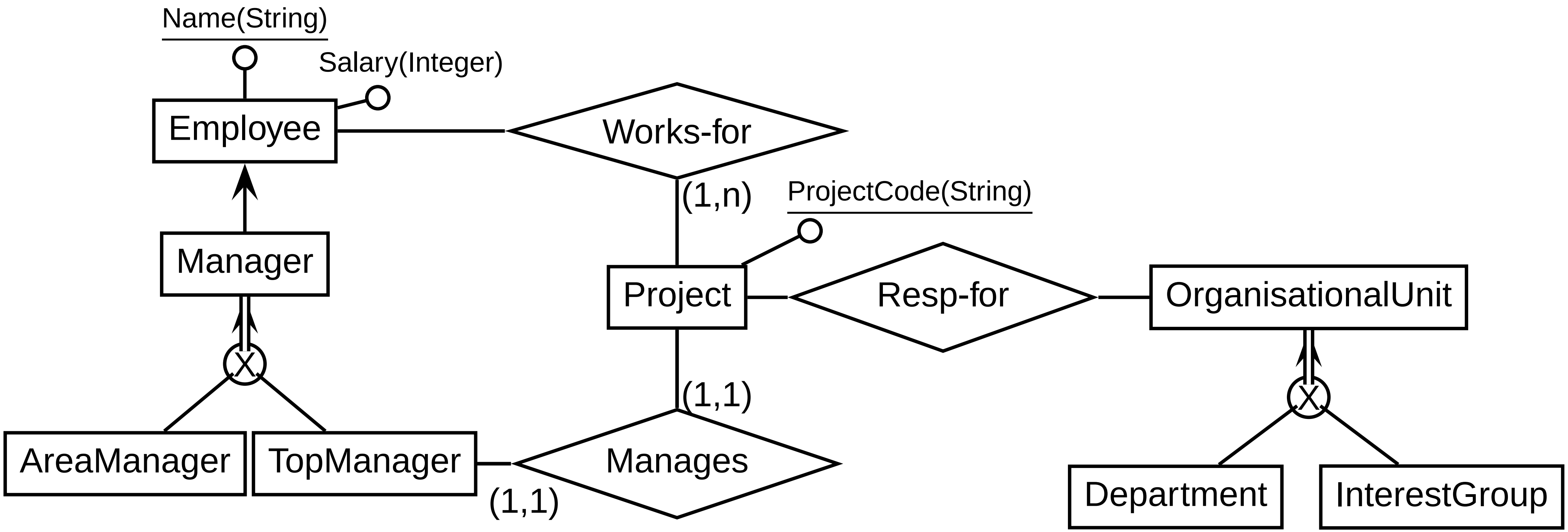
$$\texttt{Manager} \sqsubseteq \neg(\exists^{\geq 1}[\texttt{emp}]\texttt{Works-for})$$

$\leadsto$ For every project, there is at least one employee who is not a manager:

$$\Sigma \models \texttt{Project} \sqsubseteq \exists^{\geq 1}[\texttt{act}](\texttt{Works-for} \sqcap \texttt{emp} : \neg\texttt{Manager})$$

Querying the Virtual Database (local-as-view)

$$\mathtt{Q}(x,y,z) \Leftarrow \mathtt{Project}(x) \wedge \mathtt{Works\text{-}for}(john,x) \wedge \mathtt{TopManager}(y) \wedge \mathtt{Manages}(y,x) \wedge \neg\mathtt{InterestGroup}(z) \wedge \mathtt{Resp\text{-}for}(z,x).$$

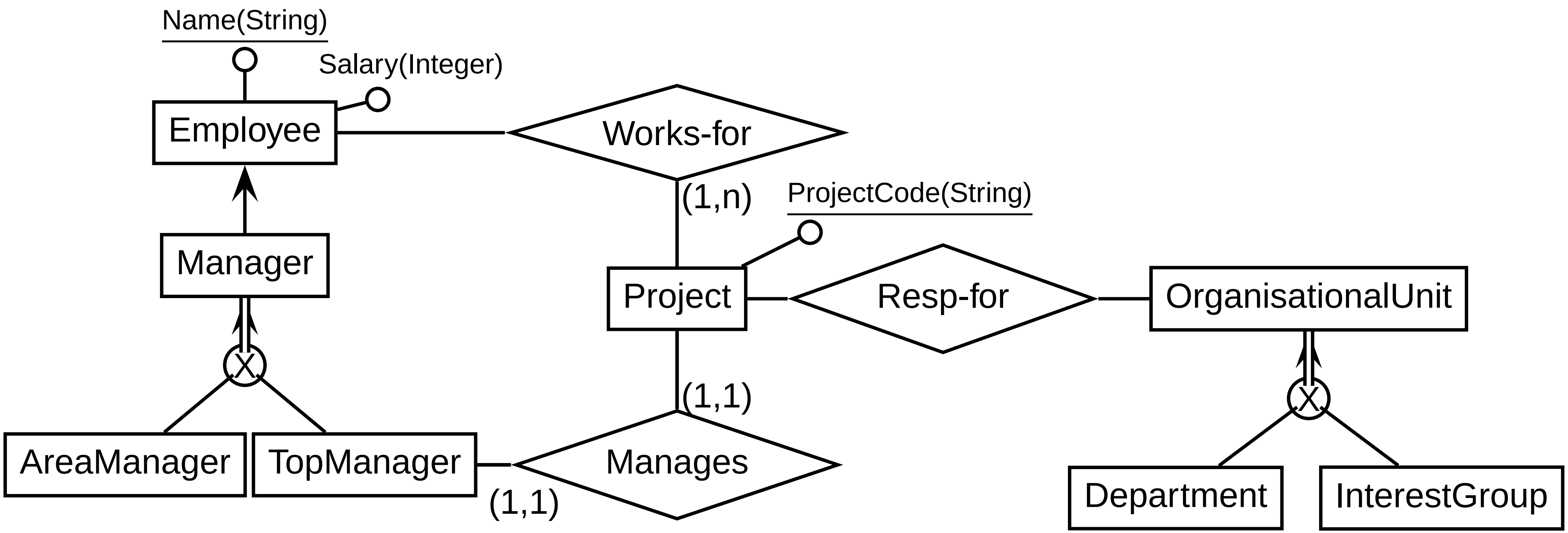


$$\mathtt{CompanyEmployee}(x,y) \Leftarrow \mathtt{Employee}(x) \wedge \mathtt{Project}(y) \wedge \mathtt{Works\text{-}for}(x,y).$$

$$\mathtt{CompanyProject}(x,y,z) \Leftarrow \mathtt{Project}(x) \wedge \mathtt{Manager}(y) \wedge \mathtt{Department}(z) \wedge \mathtt{Manages}(y,x) \wedge \mathtt{Resp\text{-}for}(z,x).$$

Querying the Virtual Database (local-as-view)

$$\mathtt{Q}(x,y,z) \Leftarrow \mathtt{Project}(x) \wedge \mathtt{Works\text{-}for}(john,x) \wedge \mathtt{TopManager}(y) \wedge \mathtt{Manages}(y,x) \wedge \neg\mathtt{InterestGroup}(z) \wedge \mathtt{Resp\text{-}for}(z,x).$$



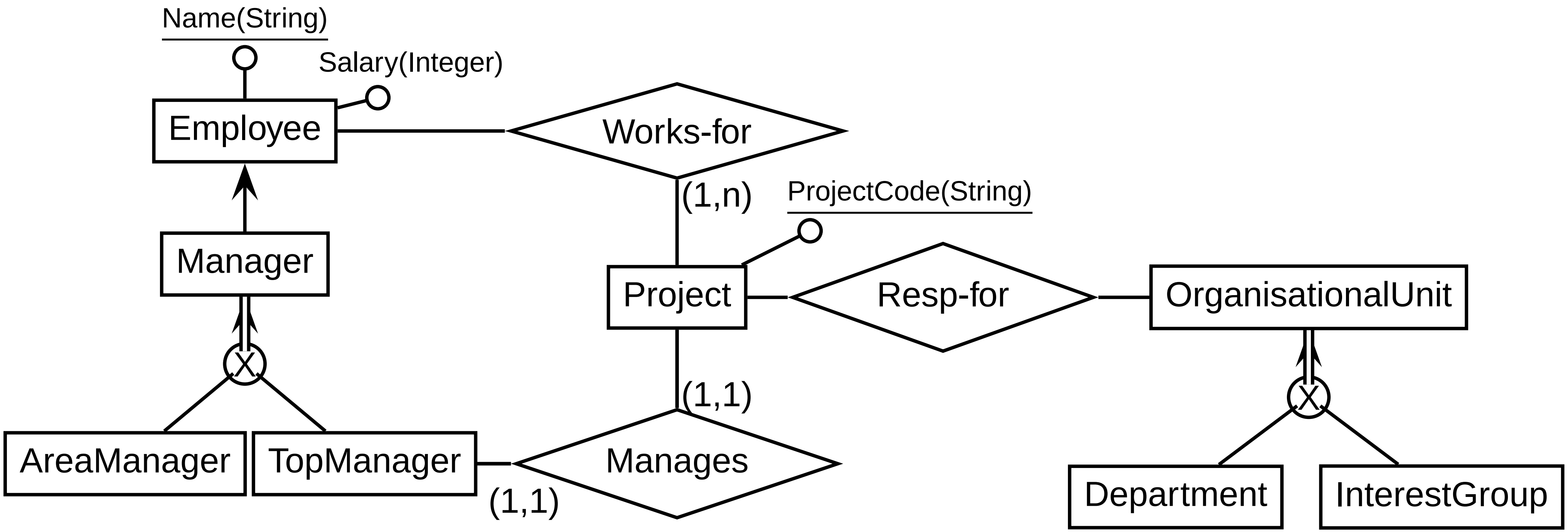
$$\mathtt{CompanyEmployee}(x,y) \Leftarrow \mathtt{Employee}(x) \wedge \mathtt{Project}(y) \wedge \mathtt{Works\text{-}for}(x,y).$$

$$\mathtt{CompanyProject}(x,y,z) \Leftarrow \mathtt{Project}(x) \wedge \mathtt{Manager}(y) \wedge \mathtt{Department}(z) \wedge \mathtt{Manages}(y,x) \wedge \mathtt{Resp\text{-}for}(z,x).$$

$$\leadsto \mathtt{Q}(x,y,z) \Leftarrow \mathtt{CompanyEmployee}(john,x) \wedge \mathtt{CompanyProject}(x,y,z)$$

Querying the Virtual Database (global-as-view)

$$\texttt{Q}(x, y, z) \Leftarrow \texttt{Project}(x) \land \texttt{Works-for}(\texttt{john}, x) \land \texttt{TopManager}(y) \land \texttt{Manages}(y, x) \land$$
$$\neg\texttt{InterestGroup}(z) \land \texttt{Resp-for}(z, x).$$



$\texttt{Project}(x) \Leftarrow \texttt{CompanyEmployee}(y, \mathbf{x}) \cup \texttt{CompanyProject}(\mathbf{x}, y, z)$

$\texttt{Works-for}(\mathbf{x}, \mathbf{y}) \Leftarrow \texttt{CompanyEmployee}(\mathbf{x}, \mathbf{y})$

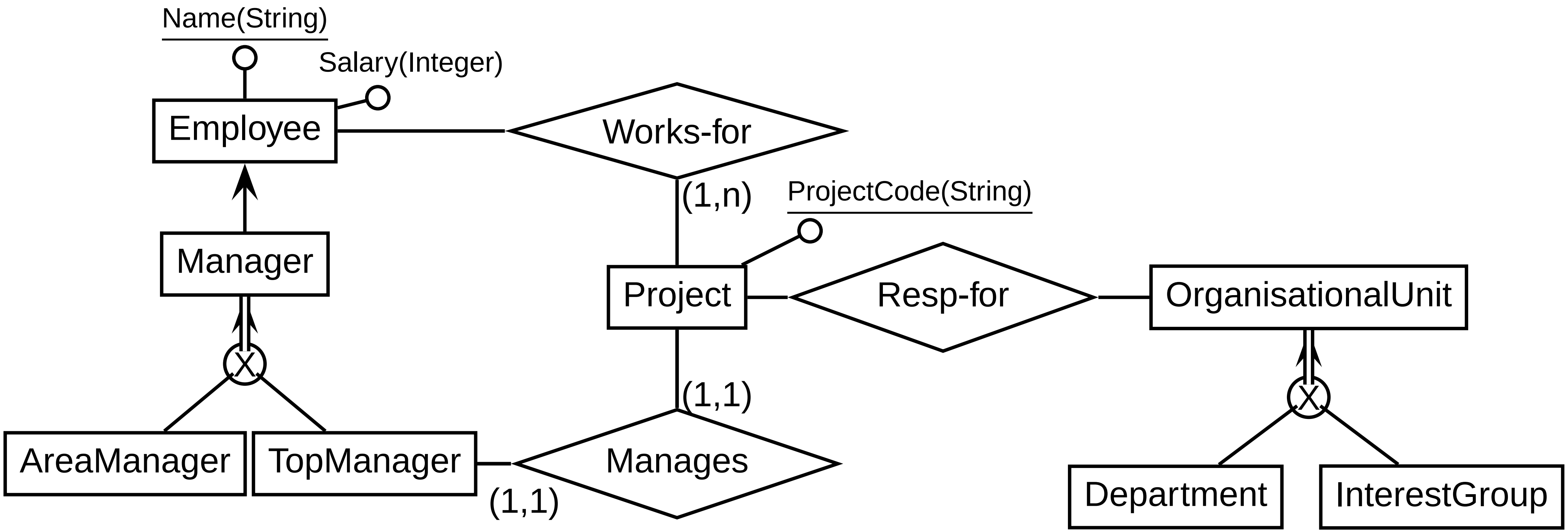
$\texttt{TopManager}(\mathbf{x}) \Leftarrow \texttt{CompanyProject}(y, \mathbf{x}, z)$

$\texttt{Manages}(\mathbf{x}, \mathbf{y}) \Leftarrow \texttt{CompanyProject}(\mathbf{y}, \mathbf{x}, z)$

...

Querying the Virtual Database (global-as-view)

$$Q(x, y, z) \Leftarrow \texttt{Project}(x) \wedge \texttt{Works-for}(\texttt{john}, x) \wedge \texttt{TopManager}(y) \wedge \texttt{Manages}(y, x) \wedge$$
$$\neg \texttt{InterestGroup}(z) \wedge \texttt{Resp-for}(z, x).$$

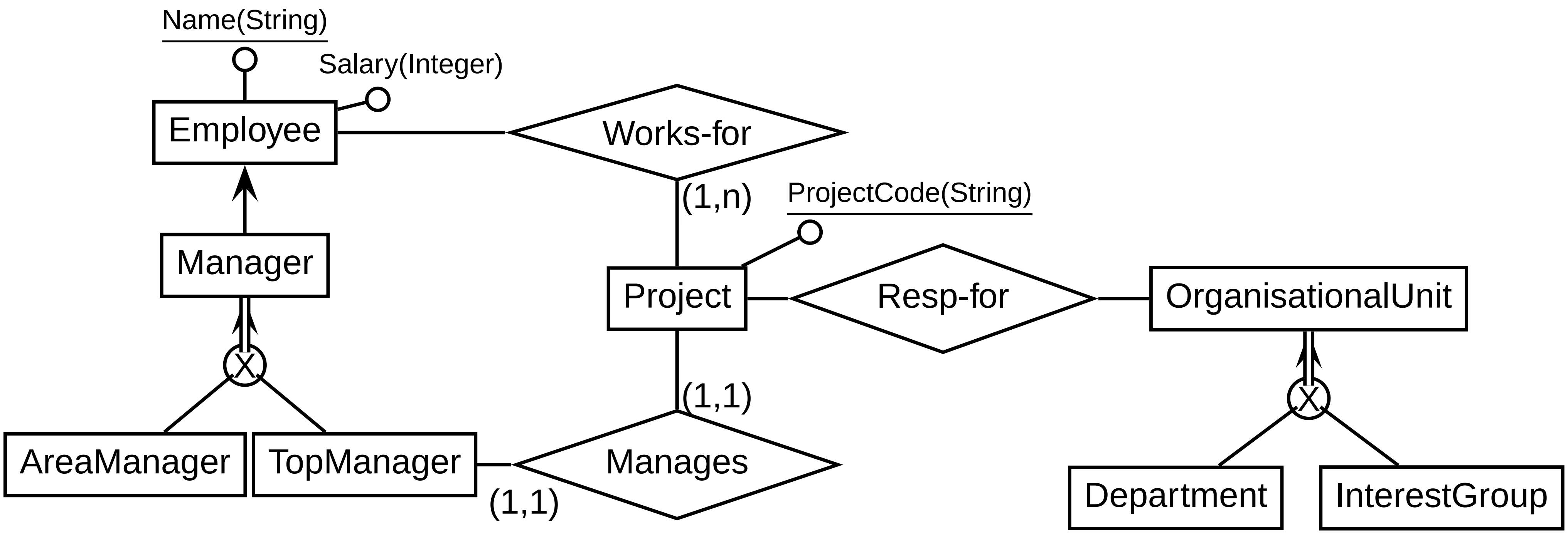


$$\texttt{Project}(\mathbf{x}) \Leftarrow \texttt{CompanyEmployee}(y, \mathbf{x}) \cup \texttt{CompanyProject}(\mathbf{x}, y, z)$$
$$\texttt{Works-for}(\mathbf{x}, \mathbf{y}) \Leftarrow \texttt{CompanyEmployee}(\mathbf{x}, \mathbf{y})$$
$$\texttt{TopManager}(\mathbf{x}) \Leftarrow \texttt{CompanyProject}(y, \mathbf{x}, z)$$
$$\texttt{Manages}(\mathbf{x}, \mathbf{y}) \Leftarrow \texttt{CompanyProject}(\mathbf{y}, \mathbf{x}, z)$$
$$\ldots$$

$$\rightsquigarrow \; Q(x, y, z) \Leftarrow \texttt{CompanyEmployee}(\texttt{john}, x) \wedge \texttt{CompanyProject}(x, y, z)$$

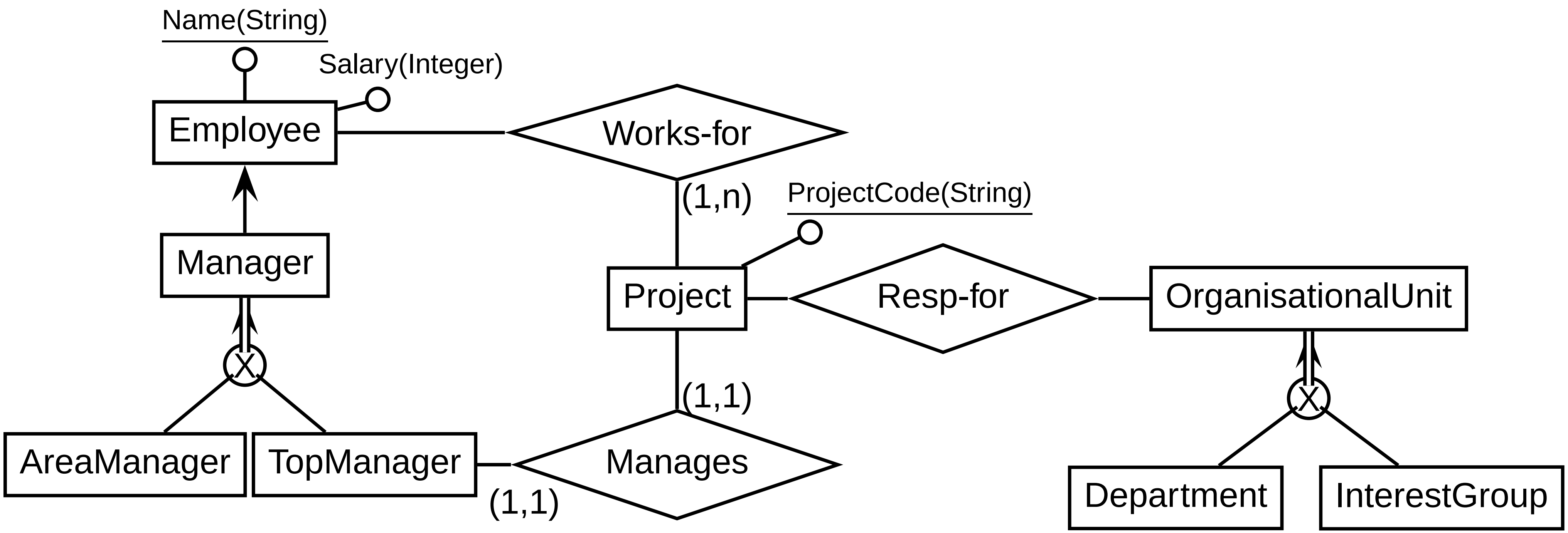
Reasoning on Queries

$$\mathtt{Q}(\mathtt{x},\mathtt{y}) \Leftarrow \mathtt{Employee}(\mathtt{x}) \wedge \mathtt{Works\text{-}for}(\mathtt{x},\mathtt{y}) \wedge \mathtt{Manages}(\mathtt{x},\mathtt{y}).$$



Reasoning on Queries

$$\mathtt{Q}(x, y) \Leftarrow \mathtt{Employee}(x) \wedge \mathtt{Works\text{-}for}(x, y) \wedge \mathtt{Manages}(x, y).$$



$\leadsto$ INCONSISTENT QUERY!