Uncertain Databases - Schema Design

Databases and Information Systems

Fabian Panse

panse@informatik.uni-hamburg.de

University of Hamburg

Referential Integrity

Conventional (certain) databases:

- Foreign keys for modeling 1:1, 1:n, and n:m relationships
- Foreign Key: Set of attributes that references to a unique set of attributes of another table
- Referential integrity is satisfied if every combination of referenced values exists in the referenced table

Person

	WK	***name***	***age***
t_1	***p1***	*J.Doe*	*27*
t_2	***p2***	*A.Miller*	*30*
t_3	***p3***	*K.Smith*	*45*

Attend

	FK	***lect***	***year***
t_4	***p2***	*Databases*	*2001*
t_5	***p3***	*Robotics*	*2002*
t_6	***p3***	*SE-1*	*2001*

FK → Person.WK

Referential Integrity

Conventional (certain) databases:

- Foreign keys for modeling 1:1, 1:n, and n:m relationships
- Foreign Key: Set of attributes that references to a unique set of attributes of another table
- Referential integrity is satisfied if every combination of referenced values exists in the referenced table

Person

	WK	***name***	***age***
t_1	***p1***	*J.Doe*	*27*
t_2	***p2***	*A.Miller*	*30*
t_3	***p3***	*K.Smith*	*45*

Attend

	FK	***lect***	***year***
t_4	***p2***	*Databases*	*2001*
t_5	***p3***	*Robotics*	*2002*
t_6	***p3***	*SE-1*	*2001*

FK → Person.WK

referential integrity is satisfied

Referential Integrity

Conventional (certain) databases:

- Foreign keys for modeling 1:1, 1:n, and n:m relationships
- Foreign Key: Set of attributes that references to a unique set of attributes of another table
- Referential integrity is satisfied if every combination of referenced values exists in the referenced table

Person

	WK	***name***	***age***
t_1	***p1***	*J.Doe*	*27*
t_2	***p2***	*A.Miller*	*30*
t_3	***p3***	*K.Smith*	*45*

Attend

	FK	***lect***	***year***
t_4	***p2***	*Databases*	*2001*
t_5	***p3***	*Robotics*	*2002*
t_6	***p5***	*SE-1*	*2001*

FK → Person.WK

Referential Integrity

Conventional (certain) databases:

- Foreign keys for modeling 1:1, 1:n, and n:m relationships
- Foreign Key: Set of attributes that references to a unique set of attributes of another table
- Referential integrity is satisfied if every combination of referenced values exists in the referenced table

Person

	WK	***name***	***age***
t_1	***p1***	*J.Doe*	*27*
t_2	***p2***	*A.Miller*	*30*
t_3	***p3***	*K.Smith*	*45*

Attend

	FK	***lect***	***year***
t_4	***p2***	*Databases*	*2001*
t_5	***p3***	*Robotics*	*2002*
t_6	***p5***	*SE-1*	*2001*

FK → Person.WK

referential integrity is NOT satisfied

Referential Integrity

Uncertain databases:

- Referential integrity is satisfied if it is satisfied in every possible world
- Foreign keys defined on world keys or representation keys
  - World key: Relationship is independent to tuple instance
  - Rep. key: Relationship is correlated with tuple instance
- Limitations depend on the used representation system

AOR-databases:

- World key is equal to representation key
- All A-tuples are certain

$\Rightarrow$ Referential integrity as in certain databases

Referential Integrity - TI-databases & AOR?-databases

TI-databases & AOR?-databases:

- World key is equal to representation key
- (A-)tuples can be maybe and all (A-)tuples are independent

⇒ References only to certain (A-)tuples allowed (otherwise there exists a world in which referential integrity is violated)

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p2	A.Miller	30	0.2
t_3	3	p3	K.Smith	45	1.0

Attend

	RK	FK	lect	year	p
t_4	1	p3	Databases	2001	0.5
t_5	2	p1	Robotics	2002	0.6
t_6	3	p2	SE-1	2001	0.4

FK → Person.WK

Referential Integrity - TI-databases & AOR?-databases

TI-databases & AOR?-databases:

- World key is equal to representation key
- (A-)tuples can be maybe and all (A-)tuples are independent

$\Rightarrow$ References only to certain (A-)tuples allowed (otherwise there exists a world in which referential integrity is violated)

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p2	A.Miller	30	0.2
t_3	3	p3	K.Smith	45	1.0

Attend

	RK	FK	lect	year	p
t_4	1	p3	Databases	2001	0.5
~~t_5~~	~~2~~	~~p1~~	~~Robotics~~	~~2002~~	~~0.6~~
~~t_6~~	~~3~~	~~p2~~	~~SE-1~~	~~2001~~	~~0.4~~

violates referential integrity

FK → Person.WK

Referential Integrity - BID-databases

- Reference to the representation key (reference to a tuple)
- Reference to the world key
  (collective reference to all tuples of the same block)
- Blocks are independent

$\Rightarrow$ References only to certain tuples/blocks allowed

Reference to representation key:

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p1	J.Doe	30	0.2
t_3	3	p2	K.Smith	45	1.0
t_4	4	p3	B.Miller	21	0.7
t_5	5	p3	B.Milla	22	0.3

Attend

	RK	FK	lect	year	p
t_7	1	3	Databases	2001	0.5
t_8	2	3	Robotics	2002	0.6
t_9	3	3	SE-1	2001	0.4

FK $\rightarrow$ ***Person.RK***

Referential Integrity - BID-databases

- Reference to the representation key (reference to a tuple)
- Reference to the world key
  (collective reference to all tuples of the same block)
- Blocks are independent

$\Rightarrow$ References only to certain tuples/blocks allowed

Reference to representation key:

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p1	J.Doe	30	0.2
t_3	3	p2	K.Smith	45	1.0
t_4	4	p3	B.Miller	21	0.7
t_5	5	p3	B.Milla	22	0.3

Attend

	RK	FK	lect	year	p
t_7	1	3	Databases	2001	0.5
t_8	2	3	Robotics	2002	0.6
t_9	3	3	SE-1	2001	0.4

FK → Person.RK

tuple t_3 is the only tuple that is allowed to be referenced

Referential Integrity - BID-databases

- Reference to the representation key (reference to a tuple)
- Reference to the world key
  (collective reference to all tuples of the same block)
- Blocks are independent

$\Rightarrow$ References only to certain tuples/blocks allowed

Reference to world key:

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p1	J.Doe	30	0.2
t_3	3	p2	K.Smith	45	1.0
t_4	4	p3	B.Miller	21	0.7
t_5	5	p3	B.Milla	22	0.3

Attend

	RK	FK	lect	year	p
t_7	1	p2	Databases	2001	0.5
t_8	2	p3	Robotics	2002	0.6
t_9	3	p3	SE-1	2001	0.4

FK → Person.WK

Referential Integrity - BID-databases

- Reference to the representation key (reference to a tuple)
- Reference to the world key
  (collective reference to all tuples of the same block)
- Blocks are independent

$\Rightarrow$ References only to certain tuples/blocks allowed

Reference to world key:

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p1	J.Doe	30	0.2
t_3	3	p2	K.Smith	45	1.0
t_4	4	p3	B.Miller	21	0.7
t_5	5	p3	B.Milla	22	0.3

Attend

	RK	FK	lect	year	p
t_7	1	p2	Databases	2001	0.5
t_8	2	p3	Robotics	2002	0.6
t_9	3	p3	SE-1	2001	0.4

FK → Person.WK

block p1 is not allowed to be referenced

Referential Integrity - BID-databases

- Representation key is not part of the world schema

$\Rightarrow$ If we reference to the representation key, queries that are defined on the world schema need to be rewritten

SELECT a.lect, a.year
FROM Attend a
WHERE a.FK = 'p2'

Query Q

$\xrightarrow{rewrite}$

SELECT a.lect, a.year
FROM Attend a, Person p
WHERE a.FK = p.RK
AND p.WK = 'p1'

Query Q^*

Referential Integrity - BID-databases

- Representation key is not part of the world schema

⇒ If we reference to the representation key, queries that are defined on the world schema need to be rewritten

⇒ To reduce the amount of rewriting, we can additionally store the world key in the foreign key

Person

	RK	WK	name	age	p
t_1	1	p1	J.Doe	27	0.6
t_2	2	p1	J.Doe	30	0.2
t_3	3	p2	K.Smith	45	1.0
t_4	4	p3	B.Miller	21	0.7
t_5	5	p3	B.Milla	22	0.3

Attend

	RK	FK1	FK	lect	year	p
t_7	1	3	p2	Databases	2001	0.5
t_8	2	3	p2	Robotics	2002	0.6
t_9	3	3	p2	SE-1	2001	0.4

(*FK1,FK*) → (*Person.RK,Person.WK*)

Coverage of conditions

Formally:

- A condition covers another condition if a fulfillment of the latter includes a fulfillment of the first
- Let $\mathscr{u}(\Phi)$ be the set of variable assignments that satisfy condition Φ
- Condition Φ_1 covers Φ_2 if $\mathscr{u}(\Phi_1) \supseteq \mathscr{u}(\Phi_2)$

Examples:

- $X = 1$ covers $X = 1 \wedge Y = 2$
- $(X = 1 \wedge Y = 1) \vee (X = 2 \wedge Y = 2)$ covers $(X = 1 \wedge Y = 1)$
- $X = 1 \wedge (Y = 1 \vee Y = 2 \vee Y = 3)$ covers $X = 1 \wedge Y = 1$

Referential Integrity - pc-databases

- Modeling of any correlation possible

⇒ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to representation key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	3	Databases	2001	Y=1 v Y=3
t_7	2	4	Robotics	2002	X=1 v X=3
t_8	3	5	SE-1	2001	X=2 v X=4

FK → Person.RK

Referential Integrity - pc-databases

- Modeling of any correlation possible

⇒ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to representation key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	3	Databases	2001	Y=1 v Y=3
t_7	2	4	Robotics	2002	X=1 v X=3
t_8	3	5	SE-1	2001	X=2 v X=4

FK → Person.RK

referential integrity is satisfied

Referential Integrity - pc-databases

- Modeling of any correlation possible

$\Rightarrow$ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to representation key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	3	Databases	2001	Y=1 v Y=3
t_7	2	4	Robotics	2002	X=1 v X=3
t_8	3	5	SE-1	2001	X=1 v X=4

FK → Person.RK

Referential Integrity - pc-databases

- Modeling of any correlation possible

$\Rightarrow$ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to representation key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	3	Databases	2001	Y=1 v Y=3
t_7	2	4	Robotics	2002	X=1 v X=3
t_8	3	5	SE-1	2001	X=1 v X=4

FK → Person.RK

referential integrity is NOT satisfied

Referential Integrity - pc-databases

- Modeling of any correlation possible

$\Rightarrow$ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to world key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	p2	Databases	2001	Y=1 v Y=3
t_7	2	p3	Robotics	2002	X=1 v X=3
t_8	3	p3	SE-1	2001	X=1 v X=4

FK → Person.WK

Referential Integrity - pc-databases

- Modeling of any correlation possible

⇒ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to world key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	p2	Databases	2001	Y=1 v Y=3
t_7	2	p3	Robotics	2002	X=1 v X=3
t_8	3	p3	SE-1	2001	X=1 v X=4

FK → Person.WK

referential integrity is satisfied

Referential Integrity - pc-databases

- Modeling of any correlation possible

⇒ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to world key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	p2	Databases	2001	Y=4
t_7	2	p3	Robotics	2002	X=1 v X=3
t_8	3	p3	SE-1	2001	X=1 v X=4

FK → Person.WK

Referential Integrity - pc-databases

- Modeling of any correlation possible

⇒ Reference to every tuple is theoretically allowed

- Condition of referencing tuple has to be covered by the condition of the referenced tuple

Reference to world key:

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	p2	Databases	2001	Y=4
t_7	2	p3	Robotics	2002	X=1 v X=3
t_8	3	p3	SE-1	2001	X=1 v X=4

FK → Person.WK

referential integrity is NOT satisfied

Referential Integrity - pc-databases

References to world keys:

- Cannot model correlations between possible instances of persons and the fact of attendance
- No redundant data storage

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

Attend

	RK	FK	lect	year	cond.
t_6	1	p2	Databases	2001	Y=1 v Y=3
t_7	2	p3	Robotics	2002	X=1 v X=3
t_8	3	p3	SE-1	2001	X=1 v X=4

FK → Person.WK

Referential Integrity - pc-databases

References to representation keys:

- Can model correlations between possible instances of persons and the fact of attendance
- Redundant data storage in the case of independence

Person

	RK	WK	name	age	cond.
t_1	1	p1	J.Doe	27	X=1
t_2	2	p1	J.Doe	30	X=2
t_3	3	p2	K.Smith	45	Y=1 v Y=3
t_4	4	p3	B.Miller	21	X=1 v X=3
t_5	5	p3	B.Milla	22	X=2 v X=4

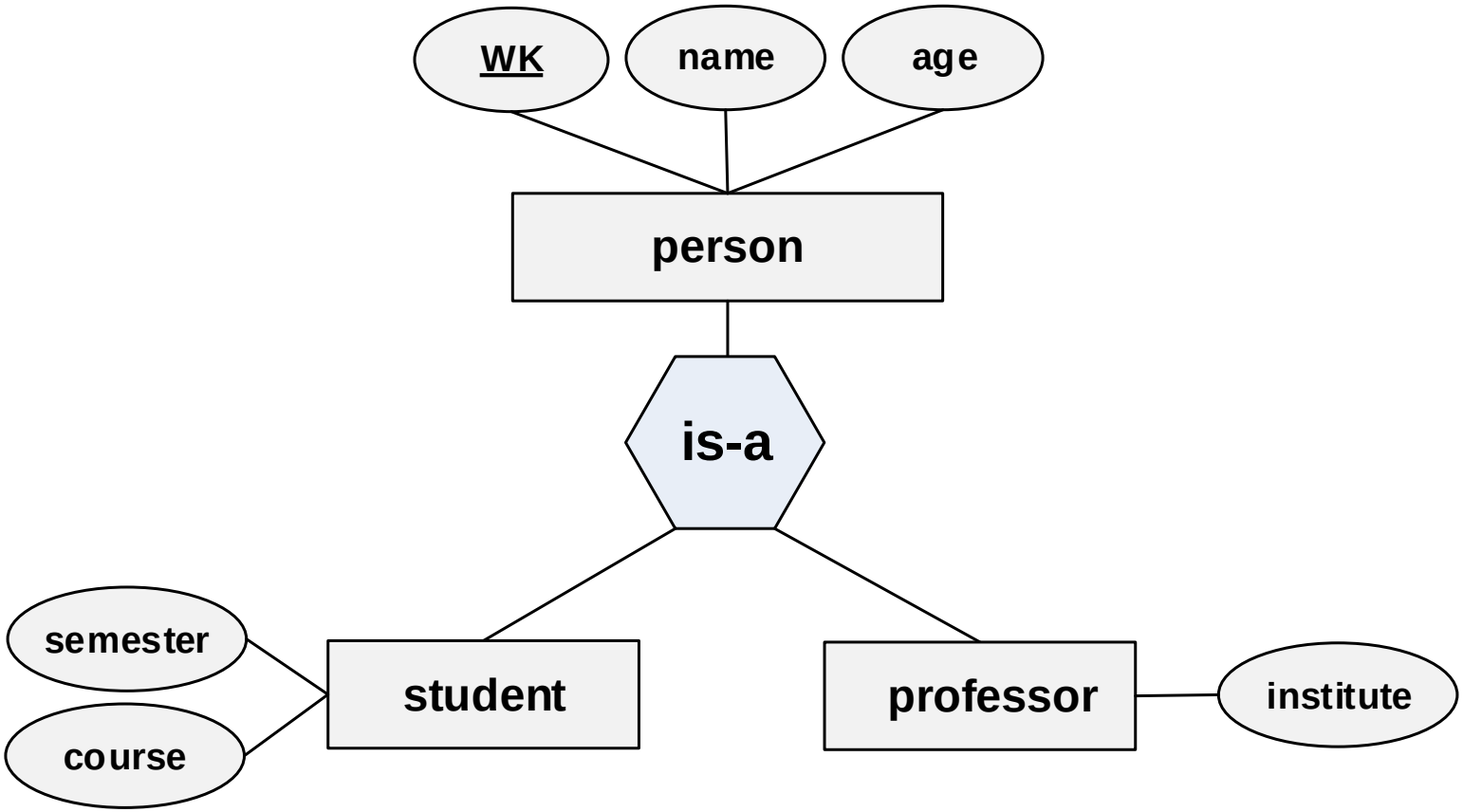
Attend

	RK	FK	lect	year	cond.
t_6	1	3	Databases	2001	Y=1 v Y=3
t_7	2	4	Robotics	2002	X=1 v X=3
t_8	3	5	SE-1	2001	X=2 v X=4

FK → Person.RK

Inheritance hierarchies

Example - ER-schema:



Inheritance hierarchies - Single table approach

- All attributes in a single table
- Extra Boolean attributes to model membership to subtypes (called *membership attributes*)

Person

	WK	***name***	***age***	***isStud***	***sem.***	***course***	***isProf***	***institute***
t_1	**p1**	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL
t_2	**p2**	*I.Miller*	*32*	*false*	NULL	NULL	*true*	*Databases*

- Efficient: 'Select all persons'
- Inefficient: 'Select all students'
- Many null values $\Rightarrow$ unnecessary storage requirements

Inheritance hierarchies - Vertical partitioning approach

- Separate table per entity type
- Connections per foreign keys

Person

	WK	name	age
t_1	p1	J.Doe	27
	p2	I.Miller	32

Student

	WK	sem.	course
t_2	p1	4	Math

WK→Person.WK

Professor

	WK	institute
t_3	p2	Databases

WK→Person.WK

- Efficient: 'Select all students'
- Inefficient: 'Select all persons'
- Requires joins in many cases

Inheritance hierarchies - Vertical partitioning approach

Conventional (certain) databases:

- No differences in modeling power
- Differences only in efficiency of queries and storage requirements

$\Rightarrow$ Design depends on used queries (and available storage)

Uncertain databases:

- Modeling power depends on representation system

$\Rightarrow$ Design rather depends on modeling power than on used queries

Inheritance hierarchies

TI-databases, AOR-databases, and AOR?-databases:

- Cannot model correlations between attribute values or tuples

$\Rightarrow$ Modeling of uncertainty in inheritance hierarchies not possible

pc-databases:

- Can model correlations between attribute values and tuples

$\Rightarrow$ No difference in modeling power between both approaches

BID-databases:

- Can model correlations between attribute values and tuples of the same block, but not between different blocks (or tables)

$\Rightarrow$ Single table approach much more powerful than vertical partitioning approach

Inheritance hierarchies - BID-tables

Scenario 1:

- Hierarchy contains a set of disjoint specializations

⇒ An entity can belong to only one of the subtypes (e.g. a person can either be a student or a professor, but not both)

- Vertical partitioning approach: Cannot be modeled because requires a mutual exclusion of tuples of different tables

W_1, Pr=0.6

Person

	WK	name	age
t_1	p1	J.Doe	27

Student

	WK	sem.	course
t_2	p1	4	Math

WK → Person.WK

Professor

WK	institute

WK → Person.WK

W_2, Pr=0.4

Person

	WK	name	age
t_1	p1	J.Doe	27

Student

WK	sem.	course

WK → Person.WK

Professor

	WK	institute
t_3	p1	Databases

WK → Person.WK

Inheritance hierarchies - BID-tables

Scenario 1:

- Hierarchy contains a set of disjoint specializations

$\Rightarrow$ An entity can belong to only one of the subtypes (e.g. a person can either be a student or a professor, but not both)

- Single table approach: Can be modeled because exclusion is modeled by the exclusion of tuples of the same block

W'_1, Pr=0.6 *Person*

	WK	*name*	*age*	*isStud*	*sem.*	*course*	*isProf*	*institute*
t'_1	*p1*	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL

W'_2, Pr=0.4 *Person*

	WK	*name*	*age*	*isStud*	*sem.*	*course*	*isProf*	*institute*
t'_2	*p1*	*J.Doe*	*27*	*false*	NULL	NULL	*true*	*Databases*

Inheritance hierarchies - BID-tables

Scenario 1:

- Hierarchy contains a set of disjoint specializations

⇒ An entity can belong to only one of the subtypes (e.g. a person can either be a student or a professor, but not both)

- Single table approach: Can be modeled because exclusion is modeled by the exclusion of tuples of the same block

Person

	RK	***WK***	***name***	***age***	***isStud***	***sem.***	***course***	***isProf***	***institute***	***p***
t'_1	**1**	**p1**	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL	0.6
t'_2	**2**	**p1**	*J.Doe*	*27*	*false*	NULL	NULL	*true*	*Databases*	0.4

Inheritance hierarchies - BID-tables

Scenario 2:

- Hierarchy contains a total specialization

$\Rightarrow$ An entity must belong to one of the subtypes (e.g. each person is a student or a professor)

- Vertical partitioning approach: Cannot be modeled because requires correlations between tuples of different tables
- Single table approach: Modeling by a simple integrity constraint

$\Rightarrow$ No tuple is allowed to have the value 'false' in all membership attributes (e.g. 'isStud' and 'isProf')

Inheritance hierarchies - BID-tables

Scenario 2:

- Hierarchy contains a total specialization

⇒ An entity must belong to one of the subtypes (e.g. each person is a student or a professor)

- Vertical partitioning approach: Cannot be modeled because requires correlations between tuples of different tables
- Single table approach: Modeling by a simple integrity constraint

⇒ No tuple is allowed to have the value 'false' in all membership attributes (e.g. 'isStud' and 'isProf')

Person

	RK	***WK***	***name***	***age***	***isStud***	***sem.***	***course***	***isProf***	***institute***	***p***
t'_1	**1**	**p1**	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL	0.6
t'_2	**2**	**p1**	*J.Doe*	*27*	*false*	NULL	NULL	*true*	*Databases*	0.4

Inheritance hierarchies - BID-tables

Scenario 2:

- Hierarchy contains a total specialization

⇒ An entity must belong to one of the subtypes (e.g. each person is a student or a professor)

- Vertical partitioning approach: Cannot be modeled because requires correlations between tuples of different tables
- Single table approach: Modeling by a simple integrity constraint

⇒ No tuple is allowed to have the value 'false' in all membership attributes (e.g. 'isStud' and 'isProf')

Person

	RK	***WK***	***name***	***age***	***isStud***	***sem.***	***course***	***isProf***	***institute***	***p***
t'_1	**1**	**p1**	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL	0.6
t'_2	**2**	**p1**	*J.Doe*	*27*	*false*	NULL	NULL	*true*	*Databases*	0.3
t'_3	**3**	**p1**	*J.Doe*	*27*	*false*	NULL	NULL	*false*	NULL	0.1

Inheritance hierarchies - BID-tables

Scenario 2:

- Hierarchy contains a total specialization

⇒ An entity must belong to one of the subtypes (e.g. each person is a student or a professor)

- Vertical partitioning approach: Cannot be modeled because requires correlations between tuples of different tables
- Single table approach: Modeling by a simple integrity constraint

⇒ No tuple is allowed to have the value 'false' in all membership attributes (e.g. 'isStud' and 'isProf')

Person

	RK	***WK***	***name***	***age***	***i***			***of***	***institute***	***p***
t'_1	**1**	**p1**	*J.Doe*	*27*	*t*				NULL	0.6
t'_2	**2**	**p1**	*J.Doe*	*27*	*false*	NULL	NULL	*true*	*Databases*	0.3
t'_3	**3**	**p1**	*J.Doe*	*27*	*false*	NULL	NULL	*false*	NULL	0.1

NOT allowed if specialization is total

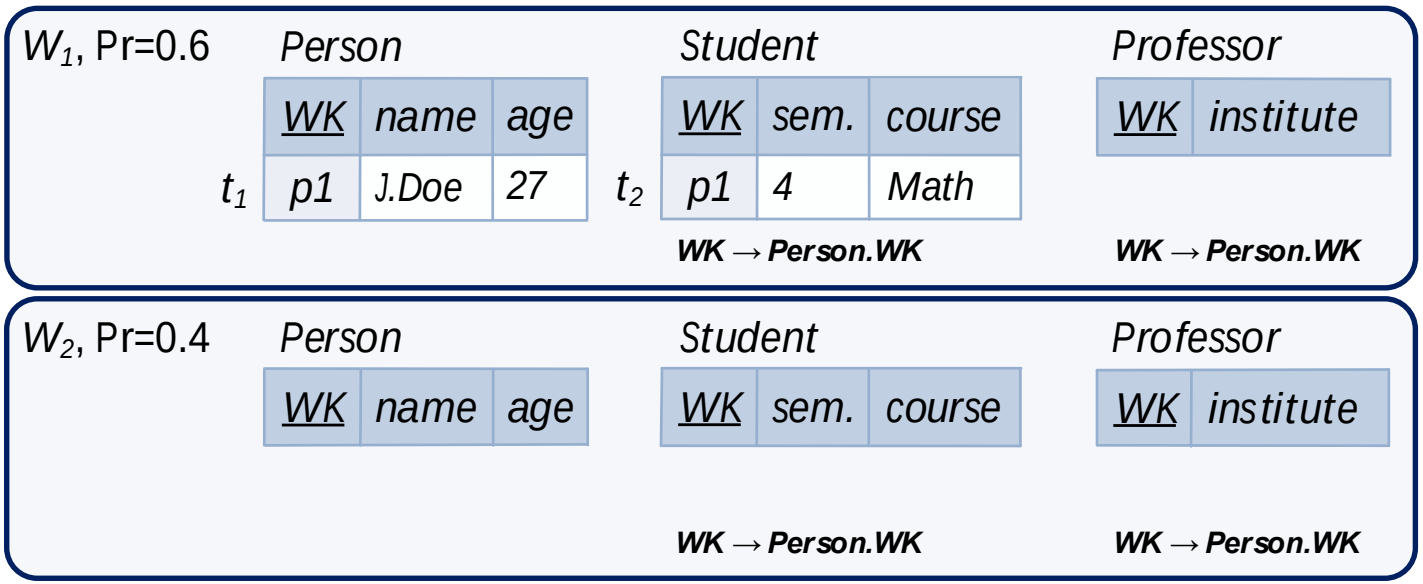
Inheritance hierarchies - BID-tables

Scenario 3:

- The membership to the supertype can be uncertain

⇒ An entity maybe does not belong to the supertype (e.g. an entity is maybe not a person)

- Vertical partitioning approach: Cannot be modeled because requires an implication of tuples of different tables



Inheritance hierarchies - BID-tables

Scenario 3:

- The membership to the supertype can be uncertain

⇒ An entity maybe does not belong to the supertype (e.g. an entity is maybe not a person)

- Single table approach: Can be modeled because a block can be maybe

W_1', Pr=0.6 *Person*

	WK	*name*	*age*	*isStud*	*sem.*	*course*	*isProf*	*institute*
t_1'	*p1*	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL

W_2', Pr=0.4 *Person*

WK	*name*	*age*	*isStud*	*sem.*	*course*	*isProf*	*institute*

Inheritance hierarchies - BID-tables

Scenario 3:

- The membership to the supertype can be uncertain

⇒ An entity maybe does not belong to the supertype (e.g. an entity is maybe not a person)

- Single table approach: Can be modeled because a block can be maybe

Person

	RK	***WK***	***name***	***age***	***isStud***	***sem.***	***course***	***isProf***	***institute***	***p***
t'_1	**1**	**p1**	*J.Doe*	*27*	*true*	*4*	*Math*	*false*	NULL	0.6

Inheritance hierarchies - BID-tables

Scenario 4:

- Memberships to different subtypes (or values in different subtypes) are correlated for some entities (e.g. the course and the library account of a particular person are correlated)
- Vertical partitioning approach: Cannot be modeled because requires correlations between tuples of different tables

W_1, Pr=0.6

Person

	WK	name	age
t_1	p1	J.Doe	27

Student

	WK	sem.	course
t_2	p1	4	Math

WK → Person.WK

Library User

	WK	account
t_3	p1	DoeM12

WK → Person.WK

W_2, Pr=0.4

Person

	WK	name	age
t_1	p1	J.Doe	27

Student

	WK	sem.	course
t_2	p1	4	Bio

WK → Person.WK

Library User

	WK	account
t_3	p1	DoeB12

WK → Person.WK

Inheritance hierarchies - BID-tables

Scenario 4:

- Memberships to different subtypes (or values in different subtypes) are correlated for some entities (e.g. the course and the library account of a particular person are correlated)
- Single table approach: Modeling by using correlations between attribute values

W'_1, Pr=0.6 *Person*

	WK	name	age	isStud	sem.	course	isLU	account
t'_1	p1	J.Doe	27	true	4	Math	true	DoeM12

W'_2, Pr=0.4 *Person*

	WK	name	age	isStud	sem.	course	isLU	accoun
t'_2	p1	J.Doe	27	true	4	Bio	true	DoeB12

Inheritance hierarchies - BID-tables

Scenario 4:

- Memberships to different subtypes (or values in different subtypes) are correlated for some entities (e.g. the course and the library account of a particular person are correlated)
- Single table approach: Modeling by using correlations between attribute values

Person

	RK	WK	name	age	isStud	sem.	course	isLU	institute	p
t'_1	1	p1	J.Doe	27	true	4	Math	true	DoeM12	0.6
t'_2	2	p1	J.Doe	27	true	4	Bio	true	DoeB12	0.4

Inheritance hierarchies - Conclusion

- Uncertainty in inheritance hierarchies cannot be modeled with TI-databases, AOR-databases and AOR?-databases
- Uncertainty in inheritance hierarchies can be modeled with BID-databases and pc-databases
- For pc-databases, both modeling approaches are equally powerful
- For BID-databases, the single table approach is more powerful than the vertical partitioning approach

$\Rightarrow$ This difference needs to be considered in designing an uncertain database

Decomposition of BID-databases

Transformation from AOR?-database to BID-database:

- One block per A-tuple
- One tuple per possible instance

$\Rightarrow$ Plain presentation of possible instances

$\Rightarrow$ Loss in Compactness

Example:

- A-tuple with 3 alternative values in each of 5 attributes

$\Rightarrow$ $3^5 = 243$ possible instances

$\Rightarrow$ $243 \times 5 = 1215$ attribute values instead of $3 \times 5 = 15$

Decomposition of BID-databases

- If attributes are independent, we can achieve the same compactness as in AOR?-databases by using decomposition
- One table per non-key attribute
- One view that connects all tables

Person

	RK	***WK***	***name***	***age***	***p***
t_1	**1**	**p1**	*J.Doe*	*20*	0.30
t_2	**2**	**p1**	*J.Doe*	*30*	0.18
t_3	**3**	**p1**	*J.Doe*	*40*	0.12
t_4	**4**	**p1**	*J.Ho*	*20*	0.20
t_5	**5**	**p1**	*J.Ho*	*30*	0.12
t_6	**6**	**p1**	*J.Ho*	*40*	0.08

Decomposition of BID-databases

- If attributes are independent, we can achieve the same compactness as in AOR?-databases by using decomposition
- One table per non-key attribute
- One view that connects all tables

PersonName

	RK	WK	name	p
t_1	1	p1	J.Doe	0.60
t_2	2	p1	J.Ho	0.40

PersonAge

	RK	WK	age	p
t_3	1	p1	20	0.50
t_4	2	p1	30	0.30
t_5	3	p1	40	0.20

```
CREATE VIEW Person AS
SELECT p1.WK, p1.name, p2.age
FROM PersonName p1, PersonAge p2
WHERE p1.WK=p2.WK
```

Decomposition of BID-databases

- Desirable: A combination of attribute-ORs and tuple-ORs

$\Rightarrow$ Increases compactness

- Can be modeled by the use of decomposition

Representation System that mixes BID- and AOR?-databases

Person

	RK	WK	name	age	p	
t_1	1	p1	*J.Doe*	{*20*:0.8, 30:0.2}	0.50	2 + 2 + 4 = 8 possible instances
t_2	2	p1	{*J.Doe*:0.9, *J.Do*:0.1}	*35*	0.20	
t_3	3	p1	{*J.Ho*:0.6, *J.Do*:0.4}	{*40*:0.5, 50:0.5}	0.10	
t_4	4	p2	*K.Smith*	*32*	1.00	
t_5	5	p3	*J.Doe*	{*29*:0.7, 30:0.3}	0.80	2 + 1 = 3 possible instances
t_6	6	p3	*J.Ho*	*40*	0.20	

Decomposition of BID-databases

```
CREATE VIEW Person AS
SELECT  p1.WK, p2.name, p3.age
FROM    Person p1, PersonName p2, PersonAge p3
WHERE p1.name=p2.WK
AND     p1.age=p3.WK
```

In PersonName and PersonAge maybe blocks are not allowed (referential integrity)

Person

	RK	WK	name	age	p
t_1	1	p1	1	1	0.50
t_2	2	p1	2	2	0.20
t_3	3	p1	3	3	0.10
t_4	4	p2	4	4	1.00
t_5	5	p3	5	5	0.80
t_6	6	p3	6	6	0.20

name → PersonName.WK,
age → PersonAge.WK

PersonName

	RK	WK	name	p
t_7	1	1	J.Doe	1.00
t_8	2	2	J.Doe	0.90
t_9	3	2	J.Do	0.10
t_{10}	4	3	J.Ho	0.60
t_{11}	5	3	J.Do	0.40
t_{12}	6	4	K.Smith	1.00
t_{13}	7	5	J.Doe	1.00
t_{14}	8	6	J.Ho	1.00

PersonAge

	RK	WK	age	p
t_{15}	1	1	20	0.80
t_{16}	2	1	30	0.20
t_{17}	3	2	35	1.00
t_{18}	4	3	40	0.50
t_{19}	5	3	50	0.50
t_{20}	6	4	32	1.00
t_{21}	7	5	29	0.70
t_{22}	8	5	30	0.30
t_{23}	9	6	40	0.20

Probabilistic Database Systems

Trio:

- Project of the Stanford University (Stanford, California)
- Project page: `http://infolab.stanford.edu/trio/`
- TriQL Manual:
  `http://cs.stanford.edu/people/widom/triql.html`

MayBMS:

- Project of the Cornell University (Ithaca, New York)
- Project page: `http://maybms.sourceforge.net/`
- Manual:
  `http://maybms.sourceforge.net/manual/maybms_manual.pdf`

Trio

Uncertain Lineage DataBases (ULDB):

- Representation system based on pc-databases
- ULDB is a set of so-called *x-relations*
- Each x-relation is a set of so-called *x-tuples*
- An x-tuple corresponds to a tuple block where each tuple is called an *alternative*
- Mutual exclusion between x-tuple alternatives implicitly modeled by the system (no variables are required)
- Alternatives of base x-tuples are associated with probabilities
- Alternatives of derived x-tuples (e.g. query results) are associated with lineage formulas (denoted by λ)
- Lineage can be used to model tuple correlations

Trio - Example

Person

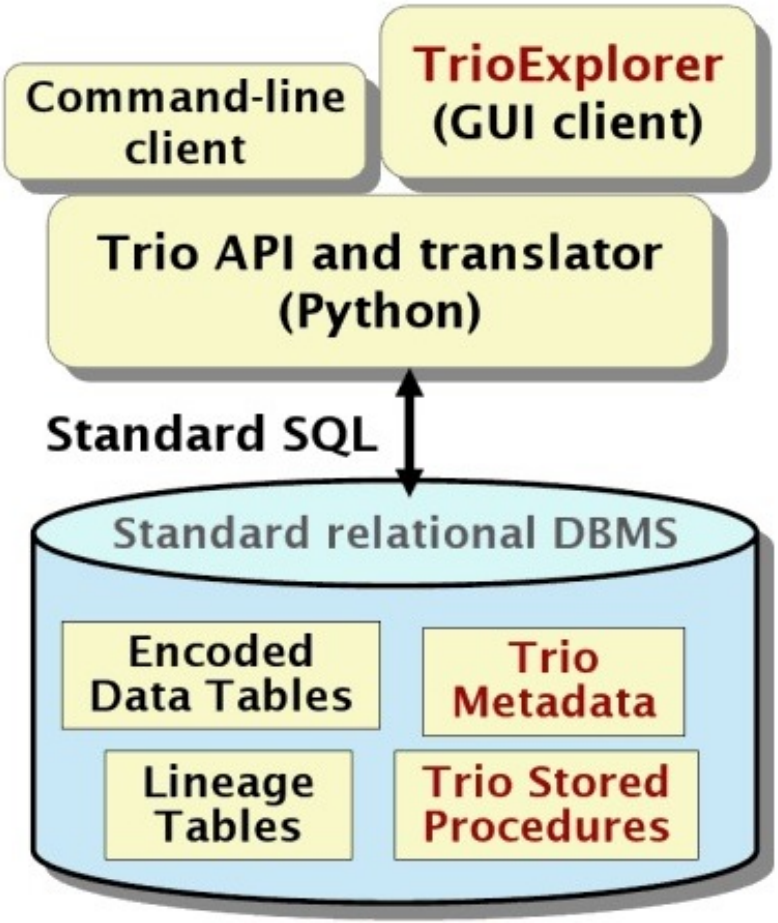
	XID	AID	WK	name	age	lineage	p
t_1	1	1	p1	J.Doe	27	λ(1,1)=(4,1)	(0.6)
t_2	1	2	p1	J.Doe	28	λ(1,2)=(4,2)	(0.2)
t_3	2	1	p2	K.Smith	32	-	0.8
t_4	2	2	p2	S.Kmith	32	-	0.2
t_5	3	1	p3	J.Doe	28	λ(3,1)=(4,1) v (4,3)	(0.8)
t_6	3	2	p3	J.Ho	29	λ(3,2)=(4,2)	(0.2)

Correlation Table

	XID	AID	value	p
t_7	4	1	1	0.6
t_8	4	2	2	0.2
t_9	4	3	3	0.2

- Correlation table serves as world-table
- Lineage $\lambda(3,1) = (4,1) \vee (4,3)$ means that t_5 exists if either the 1st alternative of the 4th x-tuple (tuple $t_7 = t_{4,1}$) or the 3rd alternative of the 4th x-tuple (tuple $t_9 = t_{4,3}$) exists

The Trio System



- Build on a conventional database system
- Extra tables that store lineage formulas
- Rewriting of TriQL queries in conventional queries
- Stored procedures (e.g. probability computation)

MayBMS

U-relational databases:

- Representation system based on pc-databases
- Conditions are limited to conjunctions of k atomic conditions $X = c$ where X is a variable and c a constant
- Relational concept to store conditions: Two *condition columns* **V** (variable) and **D** (constant) per atomic condition
- Modeling of disjunctions of conjunctions by storing duplicate tuples with different conditions
- Representation key is the combination of all condition columns and the world key
- Increased compactness by decomposition
  (note, because of the existence of conditions, the attributes do not need to be independent)

MayBMS - Example

$U_{Person[name]}$

	V	D	RK	WK	name
t_1	X	1	1	p1	J.Doe
t_2	X	2	2	p1	J.Doe
t_3	Y	1	3	p2	K.Smith
t_4	Y	2	4	p2	S.Kmith
t_5	X	1	5	p3	J.Doe
t_6	X	3	6	p3	J.Doe
t_7	X	2	7	p3	J.Ho

$U_{Person[age]}$

	V	D	RK	WK	age
t_8	X	1	1	p1	27
t_9	X	2	2	p1	28
t_{10}	$\perp$	$\perp$	3	p2	32
t_{11}	X	1	4	p3	28
t_{12}	X	3	5	p3	28
t_{13}	X	2	6	p3	29

World-Table

V	D	Pr
X	1	0.6
X	2	0.2
X	3	0.2
Y	1	0.8
Y	2	0.2

- The tuples t_1 and t_2 have the same values, but different conditions

$\Rightarrow$ Together they form a disjunction of two atomic conditions

- In this example only one pair of condition columns

$\Rightarrow$ Modeling of conjunctions is not possible without adding another pair of condition columns

MayBMS - Example

$U_{Person[name]}$

	V1	D1	V2	D2	RK	WK	name
t_1	X	1	Y	1	1	p1	J.Doe
t_2	X	2	⊥	⊥	2	p1	J.Doe
t_3	Y	1	⊥	⊥	3	p2	K.Smith
t_4	Y	2	X	3	4	p2	S.Kmith

World-Table

V	D	Pr
X	1	0.6
X	2	0.2
X	3	0.2
Y	1	0.8
Y	2	0.2

- Two pairs of condition columns

⇒ Each tuple can be associated with a conjunction of two atomic conditions

- Examples: $\Phi_{t_1} = (X = 1) \wedge (Y = 1)$ and $\Phi_{t_2} = (X = 2)$
- Tuples t_1 and t_2 have the same values, i.e. ('p1','J.Doe')

⇒ Together they represent a logical tuple t_{12} with the condition

$$\Phi_{t_{12}} = ((X = 1) \wedge (Y = 1)) \vee (X = 2)$$

Further Probabilistic DBMS (Research Prototypes)

- MystiQ: University of Washington
  `https://homes.cs.washington.edu/~suciu/project-mystiq.html`
- Sprout: University of Oxford
  `http://www.cs.ox.ac.uk/projects/SPROUT/`
- BayesStore: University of California, Berkeley (graphical)
  `http://www.eecs.berkeley.edu/Research/Projects/Data/102060.html/`
- PrDB: University of Maryland (graphical)
  `http://www.cs.umd.edu/~amol/PrDB/`
- Orion: Purdue University (continuous)
  `http://orion.cs.purdue.edu/`
- Pip: Cornell University (continuous)
  `http://maybms.sourceforge.net/pip/index.html`
- JudgeD: University of Twente (datalog)
  `https://github.com/utdb/judged`