Multilevel Security

Lecture 10

Computer Security DD2395

Roberto Guanciale
robertog@kth.se

2016-11-29

Interactive lecture

https://twitter.com/roberto_kth
@roberto_kth

http://www.quizsocket.com/
WYMYY7

Question 1: Does quizsocket work?

- A - Yes!
- B,C,D - No

Computer Security Models

- all complex systems have eventually revealed (design) flaws
- extraordinary difficult to implement (hw/sw) the design without introducing bugs
- methods to prove that a design satisfies a set of security requirements
- methods to prove that the implementation conforms the design

Computer Security Models

"A design without specification cannot be right or wrong, it can only be surprising!"

Young

Use formal methods to state properties, describe specifications and analyze designs

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$
- Discretionary AC: “owner” sets permissions
  - users make mistakes

Access controls

Formalize restrictions of accesses to resources

M	file1	file2	directory
user1	r,w	r	r,w,x
user2	r,w	-	r
user3	r	-	r,w

- s can do op on o if $op \in M[s, o]$
- Discretionary AC: "owner" sets permissions
    - users make mistakes
- Mandatory AC: system-wide policies
    - DAC can not give more access than MAC

Multi-Level Security

- MLS uses ordered security classes, e.g.
  - hardware: restricted/unrestricted CPU modes
  - software: superuser/user Linux/Windows
  - military: top secret, secret, confidential, restricted, unclassified
  - business: strategic, sensitive, confidential, public

Bell-La Padula (BLP) Model

- developed in 1970s
- formal access control model
- subjects and objects have a security class
  - subject has a security clearance level
  - object has a security classification level
  - classes control how subject may access an object

Bell-La Padula (BLP) Model

- security levels (partially) ordered
  - $L0 < L1 < L2 < L3$
  - $L0 < L1$, $L0 < L2$, $L1 < L3$, $L2 < L3$
- captures confidentiality
  - information can not flow from more secure to less secure levels
- access modes:
  - r: read
  - a: append
  - w: write
  - x: execute

BLP State

- state is a tuple (b, M, f)

BLP State

- state is a tuple (b, M, f)
- current access set $b = \{(s_1, o_1, a_1), \ldots, (s_n, o_n, a_n)\}$
  - (s_i, o_i, a_i): the subject s_i is exercising the access a_i to the object o_i

BLP State

- state is a tuple (b, M, f)
- current access set $b = \{(s_1, o_1, a_1), \ldots, (s_n, o_n, a_n)\}$
  - (s_i, o_i, a_i): the subject s_i is exercising the access a_i to the object o_i
- current access matrix M
  - s can do op on o if $op \in M[s, o]$

BLP State

- state is a tuple (b, M, f)
- current access set $b = \{(s_1, o_1, a_1), \ldots, (s_n, o_n, a_n)\}$
  - (s_i, o_i, a_i): the subject s_i is exercising the access a_i to the object o_i
- current access matrix M
  - s can do op on o if $op \in M[s, o]$
- level functions $f = (f_o, f_s, f_c)$
  - $f_O(o)$: classification level of object o
  - $f_S(s)$: security clearance (max sec.level) of subject s
  - $f_C(s)$: current sec.level of subject s $(f_C(s) \leq f_S(s))$

BLP: Simple Security

- ss-property: no read up
- a subject may read only if it has at least as high security clearance as the object
- $(s, o, read) \in b$ then $f_C(s) \geq f_O(o)$

BLP: Simple Security

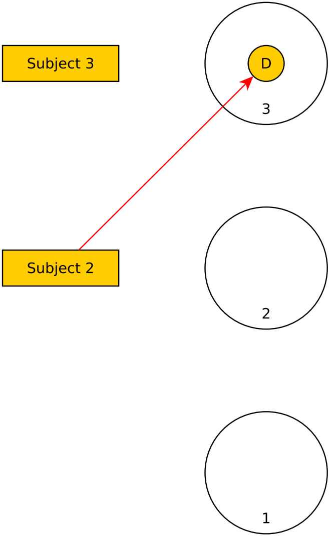
- ss-property: no read up
- a subject may read only if it has at least as high security clearance as the object
- $(s, o, read) \in b$ then $f_C(s) \geq f_O(o)$
- confidentiality: information can not flow from more secure to less secure levels

BLP: Simple Security

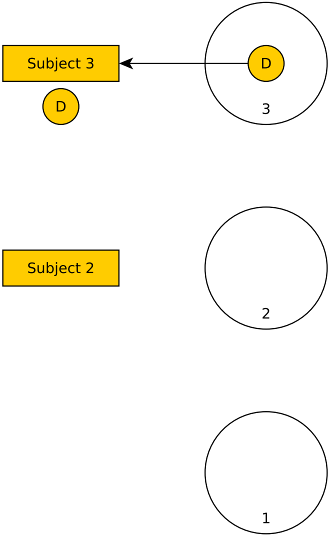
- ss-property: no read up
- a subject may read only if it has at least as high security clearance as the object
- $(s, o, read) \in b$ then $f_C(s) \geq f_O(o)$
- confidentiality: information can not flow from more secure to less secure levels

- Question 2: is ss-property sufficient to guarantee confidentiality?
  - A Yes
  - B No

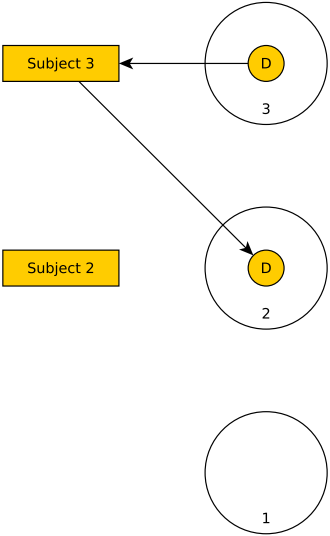
BLP: Simple Security



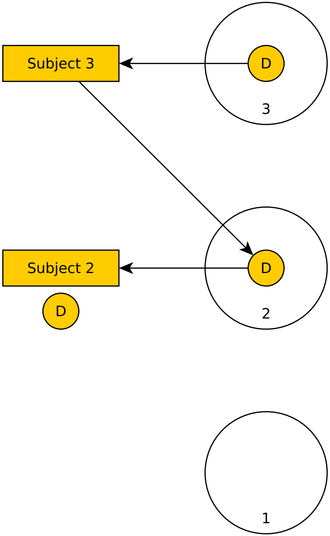
BLP: Simple Security



BLP: Simple Security



BLP: Simple Security



BLP: Star Security

- $*$-property: no write down
- a subject can write (append) only if it has equal (at most as) security clearance as the object
- $(s, o, write) \in b$ then $f_C(s) = f_O(o)$
- $(s, o, append) \in b$ then $f_C(s) \leq f_O(o)$

BLP: Star Security

- $*$-property: no write down
- a subject can write (append) only if it has equal (at most as) security clearance as the object
- $(s, o, write) \in b$ then $f_C(s) = f_O(o)$
- $(s, o, append) \in b$ then $f_C(s) \leq f_O(o)$
- with the ss-property implies that:
    - can't read a high-level object while writing a lower-level object
    - $(s, o, read) \in b$ and $(s, o', write) \in b$ then $f_O(o) \leq f_O(o')$

BLP: Discretionary Security

- ds-property: discretionary access control
- only (owner) permitted accesses are allowed
- $(s, o, a) \in b$ then $a \in M[s, o]$

BLP Rules

- get access: add a triple (s, o, a) to b

BLP Rules

- get access: add a triple (s, o, a) to b
- release access: remove triple from b

BLP Rules

- get access: add a triple (s, o, a) to b
- release access: remove triple from b
- change object level (f_O)

BLP Rules

- get access: add a triple (s, o, a) to b
- release access: remove triple from b
- change object level (f_O)
- change current level of subject (f_C)

BLP Rules

- get access: add a triple (s, o, a) to b
- release access: remove triple from b
- change object level (f_O)
- change current level of subject (f_C)
- give access permission (M)

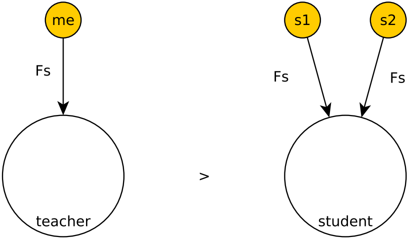
BLP Rules

- get access: add a triple (s, o, a) to b
- release access: remove triple from b
- change object level (f_O)
- change current level of subject (f_C)
- give access permission (M)
- rescind access permission (M)

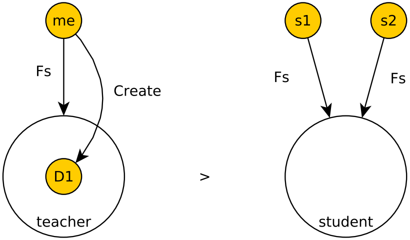
BLP: system security

- a state $S = (b, M, f)$ is secure if and only if
    - $ss - property(S)$
    - $* - property(S)$
    - $ds - property(S)$
- a transition $S \rightarrow S'$ is secure if both S and S' are secure
- a system is secure if the initial state(s) is secure and all transitions are secure

BLP: Example

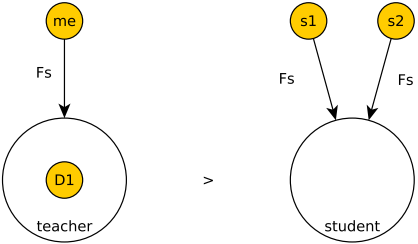


BLP: Example, D1 is the exam template



1. Create an object

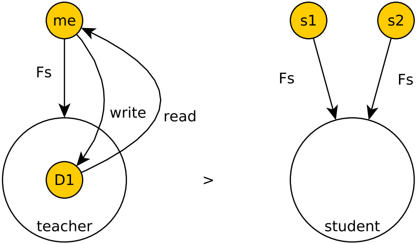
BLP: Example, D1 is the exam template



Subject	D1
me	r,w,a
s1	-
s2	-

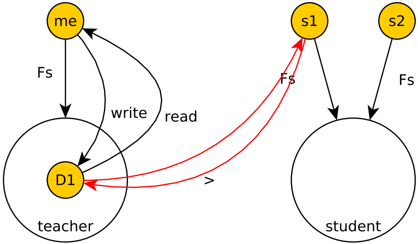
1. Give access permission

BLP: Example, D1 is the exam template



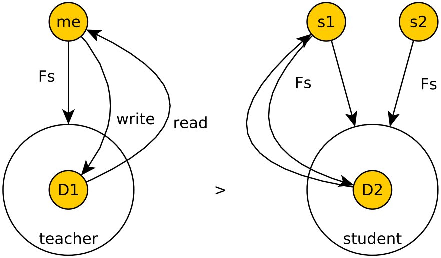
Subject	D1
me	r,w,a
s1	-
s2	-

BLP: Example, D1 is the exam template



Subject	D1
me	r,w,a
s1	-
s2	-

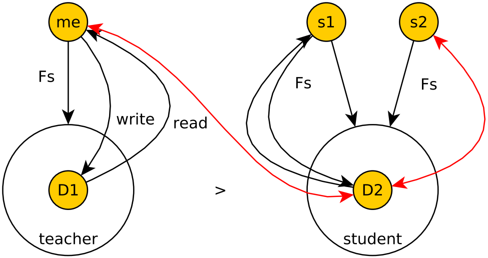
BLP: Example, D2 is the lab S report



Subject	D1	D1
me	r,w,a	-
s1	-	r,w,a
s2	-	-

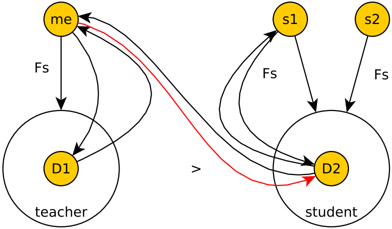
1. Create an object
2. Give access permission

BLP: Example, D2 is the lab S report



Subject	D1	D2
me	r,w,a	-
s1	-	r,w,a
s2	-	-

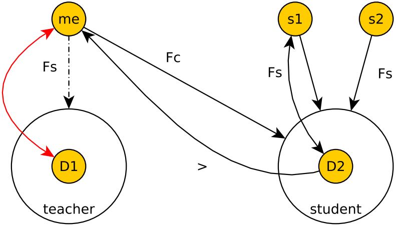
BLP: Example, D2 is the lab S report



Subject	D1	D2
me	r,w,a	r,w,a
s1	-	r,w,a
s2	-	-

1. Give access permission

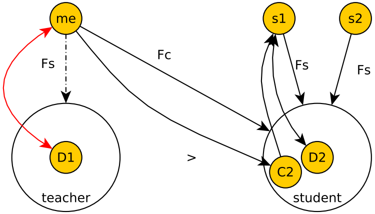
BLP: Example, C2 contains the comments to the report



Subject	D1	D2
me	r,w,a	r,w,a
s1	-	r,w,a
s2	-	-

1. Change current level

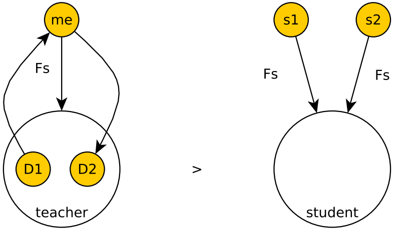
BLP: Example, C2 contains the comments to the report



Subject	D1	D2	C2
me	r,w,a	r,w,a	r,w,a
s1	-	r,w,a	r,w,a
s2	-	-	-

1. Create an object
2. Give access permission

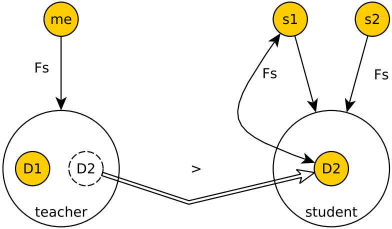
BLP: Example, D2 contains the exam for the student



Subject	D1	D1
me	r,w,a	r,w,a
s1	-	-
s2	-	-

1. Create an object
2. Give access permission

BLP: Example, D2 contains the exam for the student



Subject	D1	D1
me	r,w,a	r,w,a
s1	-	r,w,a
s2	-	-

1. Give access permission
2. Change object level (declassification)

BLP Limitation

- No internal provision for downgrading
- Classification creep by consolidation of documents from different sources and levels
- trusted subjects: set of subjects which are allowed to break *-property (assuming they always “clean” the information)
- “trusted” means “can hurt you”

Biba Integrity Model

- deals with integrity
- uses integrity levels
- reverses permitted flows: no “dirty” low-integrity info may flow to “clean” high-level info, but other way OK

Biba Policy

- simple integrity
  - no write up
  - $(s, o, write) \in b$ then $i_C(s) \geq i_O(o)$

Biba Policy

- simple integrity
  - no write up
  - $(s, o, write) \in b$ then $i_C(s) \geq i_O(o)$
- integrity confinement
  - no read down
  - $(s, o, read) \in b$ then $i_C(s) \leq i_O(o)$

Biba Policy

- simple integrity
  - no write up
  - $(s, o, \mathit{write}) \in b$ then $i_C(s) \geq i_O(o)$
- integrity confinement
  - no read down
  - $(s, o, \mathit{read}) \in b$ then $i_C(s) \leq i_O(o)$
- invocation property
  - invocation property
  - $(s, s', \mathit{invoke}) \in b$ then $i_C(s) \geq i_C(s')$

Chinese Wall model

- inspired by commercial applications
- conflict of interest
- hierarchical
  - objects ($O \in DS$): individual item of information
  - dataset ($DS \in CI$): all objects that concern the same corporation
  - conflict of interest class (CI): corporations in competition
- information can not flow between two corporations in competition

Chinese Wall policy

- keep access list H

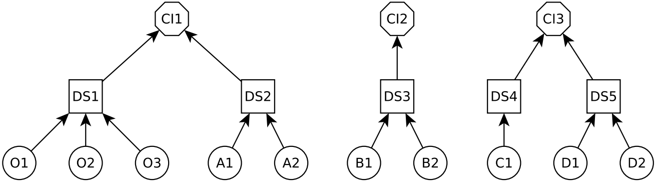
Chinese Wall policy

- keep access list H
- simple security rule
  if $(s, o, read) \in b$ then
  - $\exists o \in DS(o).(s, o', read) \in H$ or
  - $\not\exists o' \in CI(o).(s, o', read) \in H$
  - read allowed if the subject already accessed the dataset or he has not accessed any information from the CI

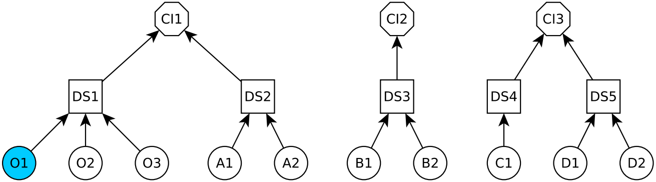
Chinese Wall policy

- keep access list H
- simple security rule
  if $(s, o, read) \in b$ then
  - $\exists o \in DS(o).(s, o', read) \in H$ or
  - $\nexists o' \in CI(o).(s, o', read) \in H$
  - read allowed if the subject already accessed the dataset or he has not accessed any information from the CI
- $*$-property rule
  if $(s, o, write) \in b$ then
  - simple security rule $ss(s, o)$ and
  - $\forall o'.ss(s, o') \Rightarrow DS(o') = DS(o)$
  - write allowed if the subject can read the object and can not read outside the DS

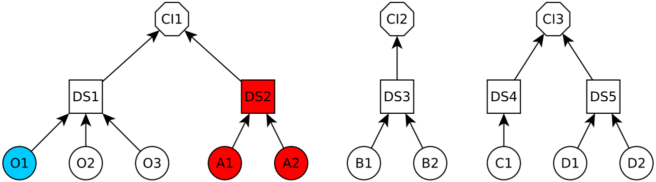
Chinese Wall: Example



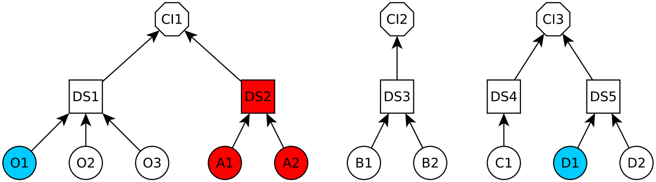
Chinese Wall: Example



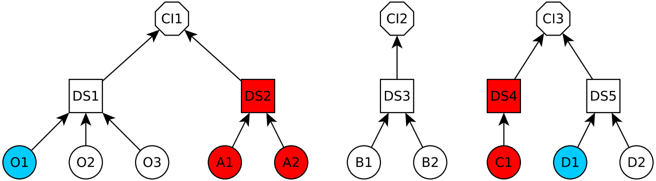
Chinese Wall: Example



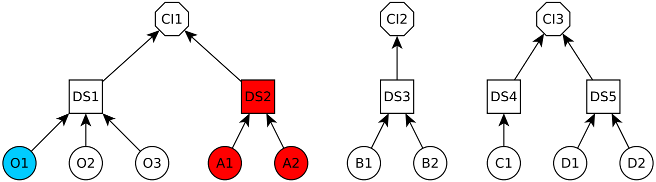
Chinese Wall: Example



Chinese Wall: Example



Chinese Wall: Writing

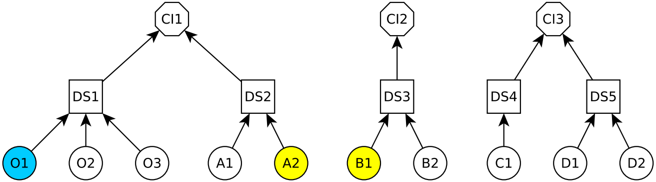


Question 3: Can I write into B1?

A No

B Yes

Chinese Wall: Writing



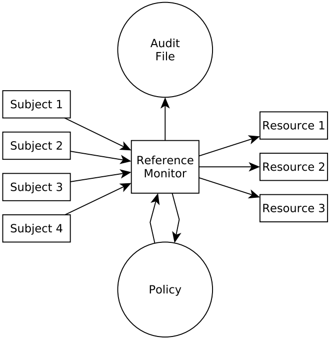
Question 3: Can I write into B1?

A No

B Yes

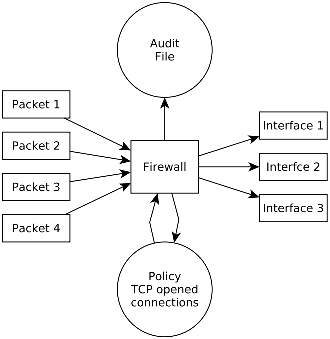
Reference Monitor

- complete mediation
- isolation
- verifiability



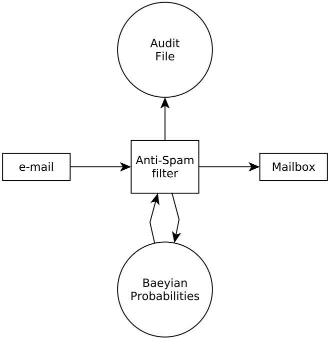
Reference Monitor

- complete mediation
- isolation
- verifiability



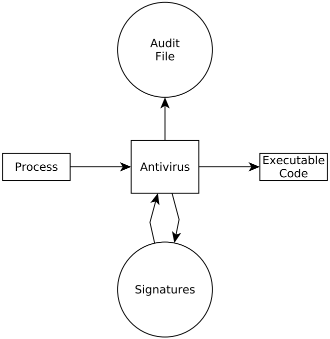
Reference Monitor

- complete mediation
- isolation
- verifiability

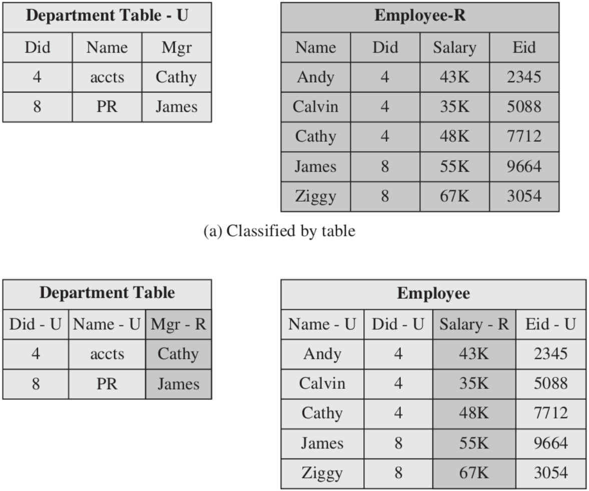


Reference Monitor

- complete mediation
- isolation
- verifiability



MLS and (relational)-databases



Department Table - U		
Did	Name	Mgr
4	accts	Cathy
8	PR	James

Employee-R			
Name	Did	Salary	Eid
Andy	4	43K	2345
Calvin	4	35K	5088
Cathy	4	48K	7712
James	8	55K	9664
Ziggy	8	67K	3054

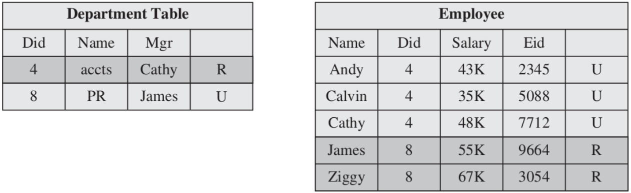
(a) Classified by table

Department Table		
Did - U	Name - U	Mgr - R
4	accts	Cathy
8	PR	James

Employee			
Name - U	Did - U	Salary - R	Eid - U
Andy	4	43K	2345
Calvin	4	35K	5088
Cathy	4	48K	7712
James	8	55K	9664
Ziggy	8	67K	3054

(b) Classified by column (attribute)

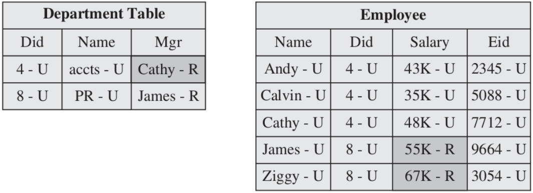
MLS and (relational)-databases



Department Table			
Did	Name	Mgr	
4	accts	Cathy	R
8	PR	James	U

Employee				
Name	Did	Salary	Eid	
Andy	4	43K	2345	U
Calvin	4	35K	5088	U
Cathy	4	48K	7712	U
James	8	55K	9664	R
Ziggy	8	67K	3054	R

(c) Classified by row (tuple)



Department Table		
Did	Name	Mgr
4 - U	accts - U	Cathy - R
8 - U	PR - U	James - R

Employee			
Name	Did	Salary	Eid
Andy - U	4 - U	43K - U	2345 - U
Calvin - U	4 - U	35K - U	5088 - U
Cathy - U	4 - U	48K - U	7712 - U
James - U	8 - U	55K - R	9664 - U
Ziggy - U	8 - U	67K - R	3054 - U

(b) Classified by element

Questions?