COMP 3311
DATABASE MANAGEMENT SYSTEMS

TUTORIAL 2
RELATIONAL MODEL AND RELATIONAL DATABASE DESIGN

REVIEW: RELATIONAL MODEL

- A set of relation schemas define a relational database.

  Employee(empId, name, address, hkid, projectNo)

  Project(projectNo, name, budget)

- A table can be used to show the instances of a relation schema.

Employee

empId	name	address	hkid	projectNo
1	Holmes D.	86 Queen	A450361	3
5	Chan B.	21 Minto	C461378	2
35	Hui J.	16 Peak	F562916	1
8	Bell G.	53 Water	A417394	2
15	Wing R.	58 Aster	C538294	3

Project

projectNo	name	budget
1	E-commerce	200,000
2	Stock control	100,000
3	Web store	500,000

Relational Model		**Representation**	**Notation**
Relation	⇔	table	$R(A_1, A_2, \ldots, A_n)$
Attribute	⇔	column	A_i
Domain	⇔	type and range of attribute values	$\mathrm{dom}(A_i)$
Tuple / Record	⇔	row	
Attribute value	⇔	value in table cell	

REVIEW: E-R TO RELATION SCHEMA REDUCTION

We need to reduce:

generalizations / specializations ⇒ inheritance, coverage

attributes ⇒ composite, multivalued

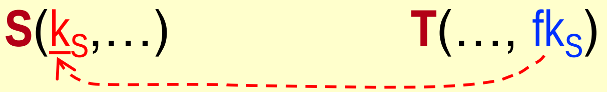
entities ⇒ strong, weak

relationships ⇒ degree (unary, binary)

⇒ constraints (cardinality, participation, inclusion)

> **Cardinality/participation constraints in the E-R model reduce to referential integrity constraints in the relational model.**

REVIEW: REFERENTIAL INTEGRITY ACTIONS



If relation **T** contains the primary key k_S of relation **S** as a foreign key fk_S, which can be specified as the foreign key constraint

foreign key (fk_S) references **S**(k_S)

then the value of fk_S in a tuple of **T** must either be equal to the value of the primary key k_S of a tuple in **S** or be entirely null.

To enforce this constraint, the following actions are required.

For E-R model: total participation

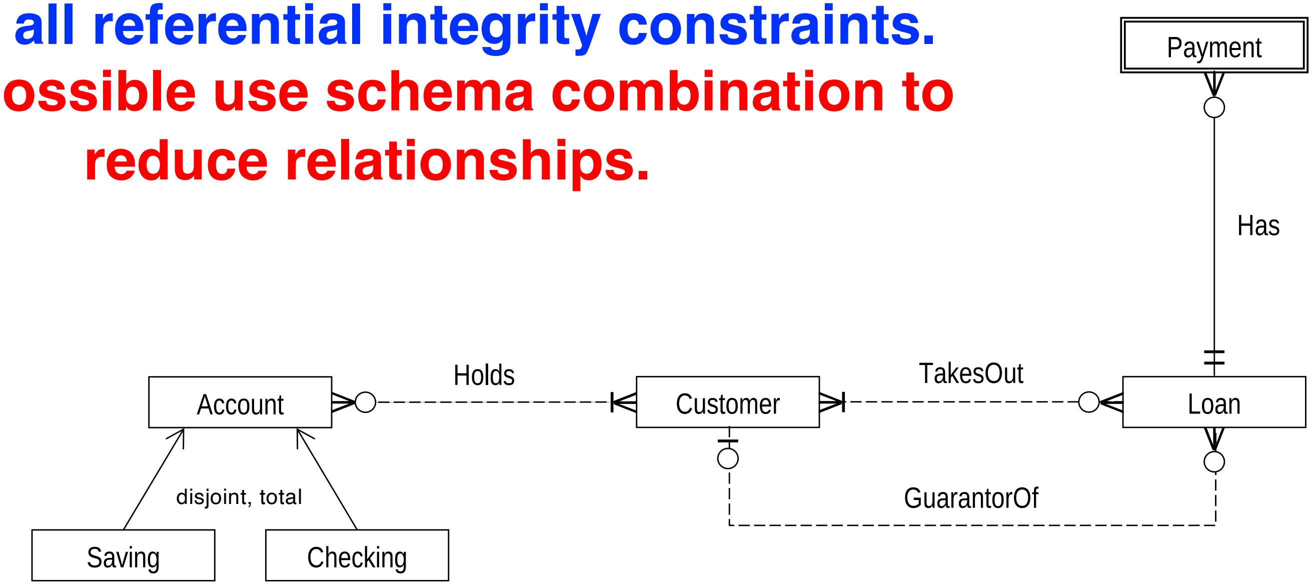
on delete cascade - Delete all tuples with foreign key values in **T** that match the primary key value of the deleted tuple in **S**.

For E-R model: partial participation

on delete set null - Set to null the foreign key value of all tuples in **T** whose foreign key value matches the primary key value of the deleted tuple in **S**.

EXERCISE 1: BANK APPLICATION

Reduce the bank E-R schema to relation schemas.
Specify all referential integrity constraints.
Where possible use schema combination to reduce relationships.



Customer
id
name
address
street
city
state
{phoneNo}

Account
accountNo
balance

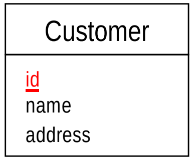
Saving
interestRate

Checking
overdraft

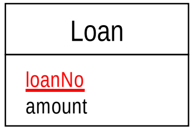
Loan
loanNo
amount

Payment
paymentNo
date
amount

EXERCISE 1: REDUCE STRONG ENTITIES



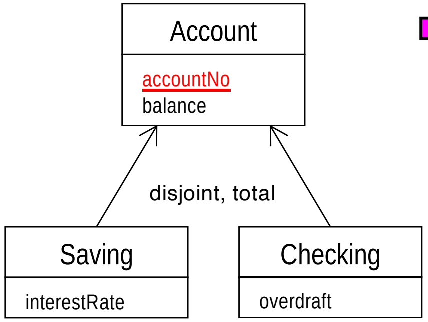
Customer(id, name, address)



Loan(loanNo, amount)

EXERCISE 1: REDUCE GENERALIZATIONS

Option 1: Reduce *all entities* to relation schemas.



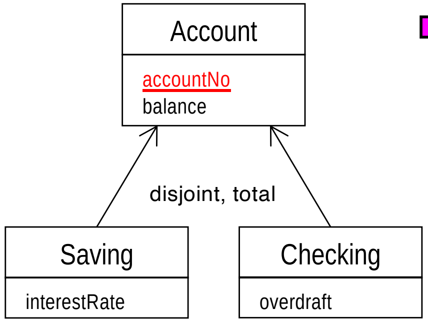
Which option to select?

Account(accountNo, balance)

Saving(accountNo, interestRate)
foreign key (accountNo) references Account(accountNo)
on delete cascade

Checking(accountNo, overdraft)
foreign key (accountNo) references Account(accountNo)
on delete cascade

Option 2: Reduce *only subclass entities* to relation schemas.



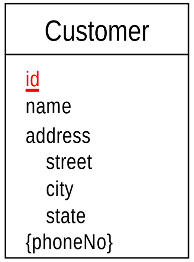
Saving(accountNo, balance, interestRate)

Checking(accountNo, balance, overdraft)

Select Option 1 since Account has a relationship to other entities and all the subclass entities have their own attributes.

EXERCISE 1: REDUCE COMPOSITE/MULTIVALUED ATTRIBUTES

Composite attributes: address


Customer
id
name
address
street
city
state
{phoneNo}

Option 1: single attribute

Customer(id, name, address)

Option 2: separate attributes

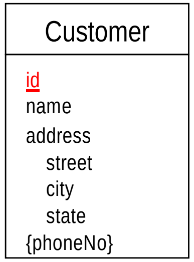
Customer(id, name, street, city, state)

Which option to select?

Which option to select will depend on the requirements of the application.
Here we subsequently use option 1.

Multivalued attributes: phoneNo

Customer(id, name, address)
(*previously reduced*)

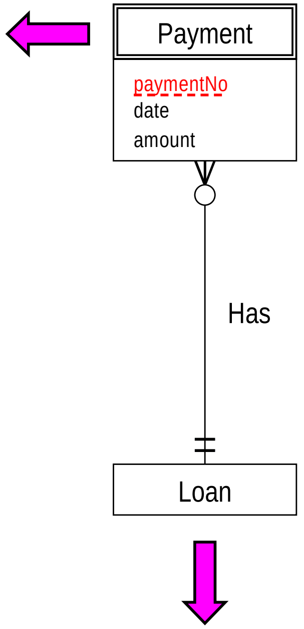

Customer
id
name
address
street
city
state
{phoneNo}

CustomerPhone(id, phoneNo)
foreign key (id) references Customer(id)
on delete cascade

EXERCISE 1: REDUCE WEAK ENTITIES

Payment entity

Payment(loanNo, paymentNo, date, amount)
foreign key (loanNo) references Loan(loanNo)
on delete cascade



Loan(loanNo, amount)
(*previously reduced*)

How do we reduce this entity?

⟹ Create a relation from Payment and include loanNo, the key of Loan, as a foreign key.

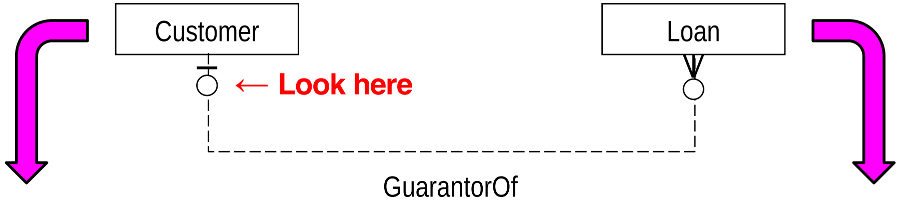
What is the key of this relation?

What is the foreign key constraint?

What is the referential integrity action?

EXERCISE 1: REDUCE 1:N RELATIONSHIPS

GuarantorOf between **Customer** and **Loan** (using schema combination)



Customer(id, name, address) *(previously reduced)*

Loan(loanNo, amount, id)
foreign key (id) references Customer(id)
on delete set null

Which relation do we use?

⟹ Loan (Add id, the key of the Customer relation, as a foreign key.)

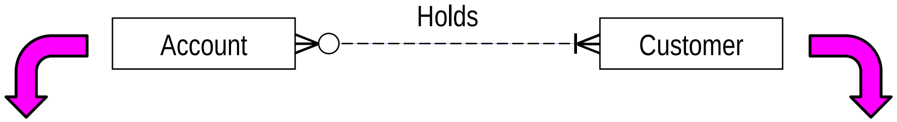
What is the referential integrity action?

The referential integrity action is determined by the participation constraint of the entity into which the foreign key is placed.

- partial: on delete set null
- total: on delete cascade

EXERCISE 1: REDUCE N:M RELATIONSHIPS

Holds relationship between Account and Customer



Account(accountNo, balance) *(previously reduced)*

Customer(id, name, address) *(previously reduced)*

Holds(accountNo, id)

foreign key (accountNo) references Account(accountNo)
on delete cascade
foreign key (id) references Customer(id)
on delete cascade

How do we reduce this relationship?

⟹ Create a relation, Holds, with the key, accountNo, of the Account relation and the key, id, of the Customer relation.

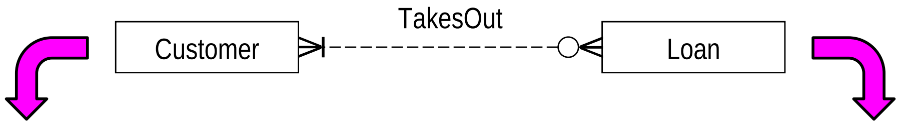
What is the key of the relation?

What are the referential integrity actions?

For a relationship that represents a relationship, the referential integrity action is always on delete cascade.

EXERCISE 1: REDUCE N:M RELATIONSHIPS

TakesOut relationship between **Customer** and **Loan**



Customer(id, name, address) (*previously reduced*)

Loan(loanNo, amount) (*previously reduced*)

TakesOut(id, loanNo)
foreign key (id) references Customer(id)
on delete cascade
foreign key (loanNo) references Loan(loanNo)
on delete cascade

How do we reduce this relationship?

⟹ Create a relation, TakesOut, with the key, id, of the Customer relation and the key, loanNo, of the Loan relation.

EXERCISE 1: BANK APPLICATION REDUCTION

Account(accountNo, balance)

Saving(accountNo, interestRate)
foreign key (accountNo) references Account(accountNo)
on delete cascade

Checking(accountNo, overdraft)
foreign key (accountNo) references Account(accountNo)
on delete cascade

Customer(id, name, address)[1]

CustomerPhone(id, phoneNo)
foreign key (id) references Customer(id)
on delete cascade

Payment(loanNo, paymentNo, date, amount)
foreign key (loanNo) references Loan(loanNo)
on delete cascade

Loan(loanNo, amount, id)
foreign key (id) references Customer(id)
on delete set null

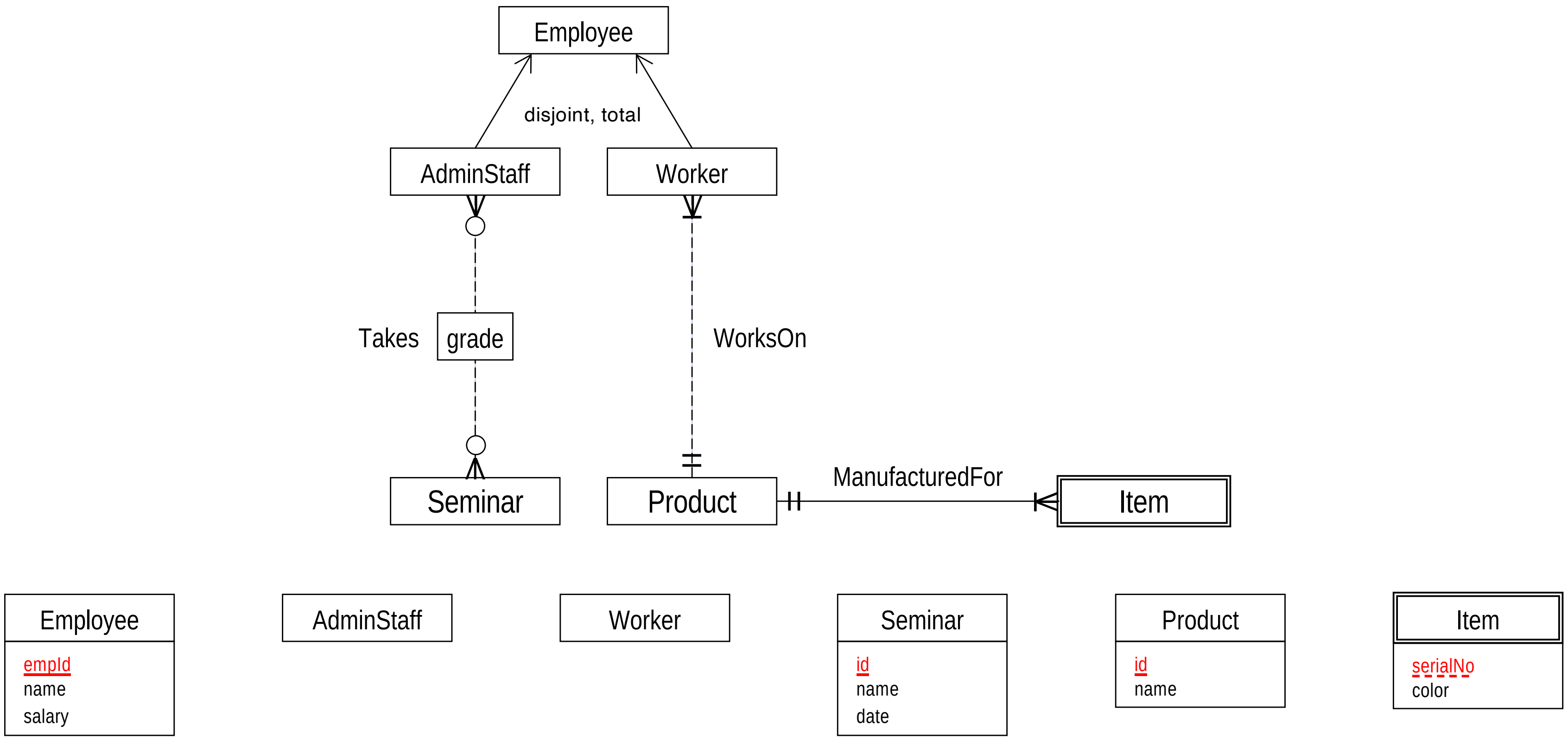
Holds(accountNo, id)
foreign key (accountNo) references
Account(accountNo)
on delete cascade
foreign key (id) references Customer(id)
on delete cascade

TakesOut(id, loanNo)
foreign key (id) references Customer(id)
on delete cascade
foreign key (loanNo) references Loan(loanNo)
on delete cascade

[1] Using option 1 for address composite attribute.

EXERCISE 2: FACTORY APPLICATION

Reduce the factory E-R schema to relation schemas.
Specify all referential integrity constraints.
Where possible, use schema combination to reduce relationships.



Employee

disjoint, total

AdminStaff

Worker

Takes

grade

WorksOn

Seminar

Product

ManufacturedFor

Item

Employee
empId
name
salary

AdminStaff

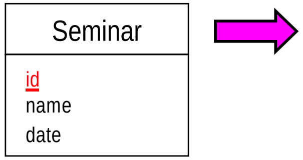
Worker

Seminar
id
name
date

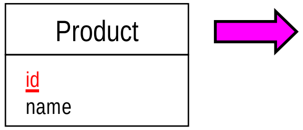
Product
id
name

Item
serialNo
color

EXERCISE 2: REDUCE STRONG ENTITIES



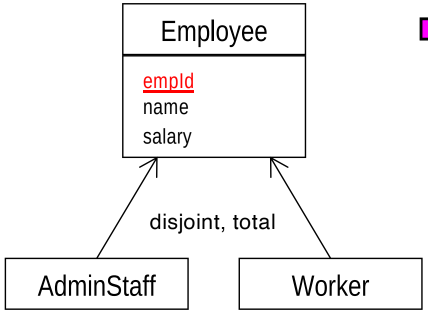
Seminar(id, name, date)



Product(id, name)

EXERCISE 2: REDUCE GENERALIZATION

Option 1: Reduce all entities to relation schemas.



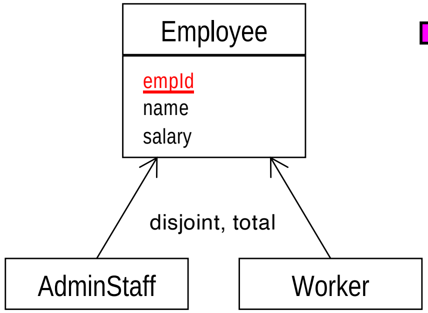
Employee(empId, name, salary)

Which option to select?

AdminStaff(empId)
  foreign key (empId) references Employee(empId)
    on delete cascade

Worker(empId)
  foreign key (empId) references Employee(empId)
    on delete cascade

Option 2: Reduce only subclass entities to relation schemas.

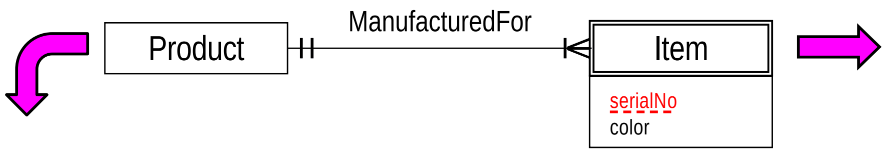


AdminStaff(empId, name, salary)

Worker(empId, name, salary)

Select Option 2 since Employee has no relationships to other entities, the subclasses have no attributes, and the generalization is disjoint and total.

EXERCISE 2: REDUCE WEAK ENTITIES



Item(id, serialNo, color)
foreign key (id) references Product(id)
on delete cascade

Product(id, name) (*previously reduced*)

How do we reduce this entity?

⟹ Create a relation from Item that includes the key, id, of the Product relation.

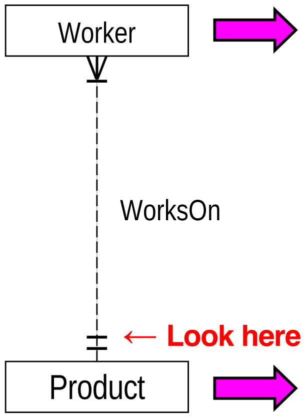
What is the key of this relation?

What is the foreign key constraint?

What is the referential integrity action?

EXERCISE 2: REDUCE 1:N RELATIONSHIPS

WorksOn relationship between **Worker** and **Product**



← **Look here**

Worker(empId, name, salary, id)
foreign key (id) references Product(id)
on delete cascade

Product(id, name) (*previously reduced*)

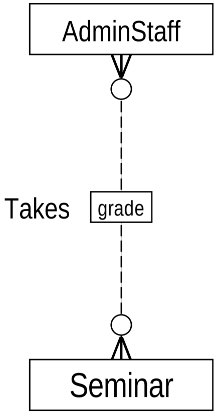
Which relation do we use?

⟹ Worker (Add the key, id, of the Product relation as a foreign key.)

What is the referential integrity action?

EXERCISE 2: REDUCE N:M RELATIONSHIPS

Takes relationship between **AdminStaff** and **Seminar**



AdminStaff(empId) *(previously reduced)*

Takes(empId, id, grade)
    foreign key (empId) references AdminStaff(empId)
        on delete cascade
    foreign key (id) references Seminar(id)
        on delete cascade

Seminar(id, name, date) *(previously reduced)*

How do we reduce this relationship?

⟹ Create a relation Takes with the key of AdminStaff and Seminar.

Anything else?

⟹ Add the attribute grade.

EXERCISE 2: FACTORY APPLICATION REDUCTION

```
AdminStaff(empId, name, salary)

Worker(empId, name, salary, id)
        foreign key (id) references Product(id)
            on delete cascade

Seminar(id, name, date)

Product(id, name)

Item(id, serialNo, color)
        foreign key (id) references Product(id)
            on delete cascade

Takes(empId, id, grade)
        foreign key (empId) references AdminStaff(empId)
            on delete cascade
        foreign key (id) references Seminar(id)
            on delete cascade
```