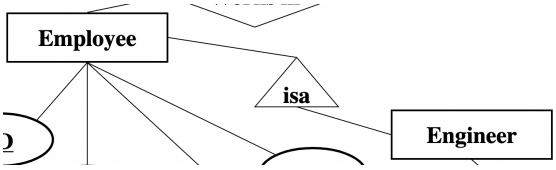
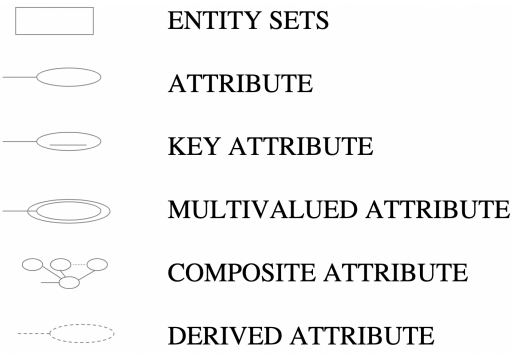
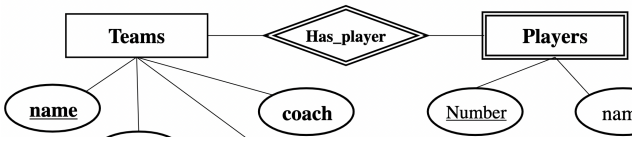
ER

Relationships can have attributes
Relationship end without arrows: many. With arrows: one
Can label relationship lines (“edges”)



Only root ISA has a key attribute





Logical Design

Informal Terms	Formal Terms
Table	Relation
Column	Attribute
Row	Tuple
Possible values in a column	Domain
Table Definition	Schema of a Relation
Populated Table	Instances

Entity	Relation
Weak-entity	Relation, including the primary key from the parent entity as an attribute
1:1 relationship	Merged relation, Foreign key, Relationship relation
1:N relationship	Foreign key, Relationship relation
M:N relationship	Relationship relation
Atomic attribute	Attribute
Key attribute	Primary key
Composite attributes	A set of attributes
Multi-valued attributes	Relation and foreign key

Super key # no two distinct tuples can have the same values in all fields
Candidate key # Super key where the above condition is not true for any subset
If |Candidate key|>1, one of them is chosen to be a Primary key
Referential integrity for foreign keys

Normalization

$X \subset Y$ # Trivial functional dependency
$X \cap Y = \emptyset$ # Completely non-trivial
1NF: table
2NF: non-prime (not members of candidate key) attributes are dependent on candidate key
3NF: 2NF + non-transitively dependent
BCNF: for all $X \rightarrow Y$, X is a super key (key->other attrs)

Decomposition to BCNF

1. Identify the candidate keys
2. Find a BCNF violation
   - That is, a non-trivial FD $X \rightarrow Y$ in R where X is not a super key of R
3. “Grow” Y to include all the attributes determined by X.
4. Decompose R into R_1 and R_2, where
   - R_1 has attributes $X \cup Y$
   - R_2 has attributes $X \cup Z$, where Z contains all attributes of R that are in neither X nor Y (i.e. $Z = attr(R) - X - Y$)
5. Repeat until all relations are in BCNF

- R(A, B, C, D):
  - AB→C
  - C→D
  - D→A

1. Identify the candidate keys

AB → C and C→D: AB→D
Hence, AB→ABCD. AB is a key
D→A: DB→AB. DB is a key
C→D: CB→DB→AB. CB is a key

2. Find a BCNF violation X→Y
   - C→D is a violation since C is not a key
3. “Grow” Y to include all the attributes determined by X
   - C→D, D→A, hence, C→DA
4. Decompose R into R_1 and R_2: R_1(C, D, A), R_2(B, C)
5. Repeat this
   - D→A is a violation since D is not a key
   - Eventually, we have: R_2(B, C), R_3(C, D), R_4(D, A)