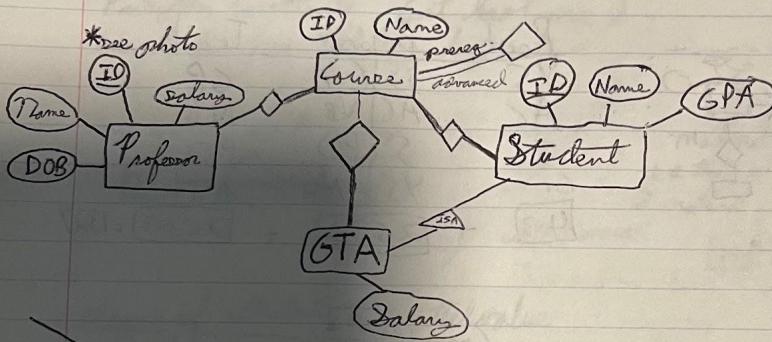
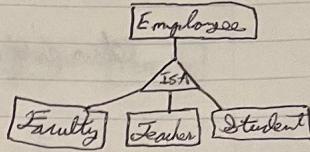


## ISA Hierarchies

points to parent entity

add descriptive attributes to subclasses, identify entities that participate in relationships

person belongs only in one row - overlay  
employees have to be hourly or contract - covering



HW1: 5% - Given a SSN, it only associates w/one person. Person has any # of SSN.



assume any table does not have repeating entry

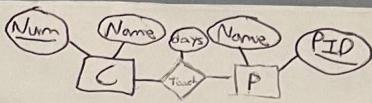
We have multiple unique identifiers - SSN, name - subset of attributes is also a key

keys used to design DB → primary keys - which is underlined

relation consists of relational schema - list of attribute/column names w/o data type  
means instance - tuples (row, records)

schema specifies relation name, name of field and its domain, primary key underlined  
- column, attr

relations are unordered, tuple is unique, demanded by relation definition



$\text{Tech}(\underline{\text{Num}}: \text{number}, \underline{\text{PID}}: \text{number}) \Rightarrow \text{Tech}(\underline{\text{num}}: \text{number}, \underline{\text{PID}}: \text{number}, \underline{\text{days}}: \text{number})$   
 number of tuples = cardinality (not row of attribute names)  
 degree = number of columns

→ bold arrow means one and only one  
 $\bowtie$   $\text{UniProf}(\underline{\text{salary}}, \underline{\text{id}}, \underline{\text{days}}, \underline{\text{Name}}, \underline{\text{PID}})$  underline id or PID

`CREATE TABLE UniProf (`

<code>salary REAL,</code>	* 1) write underline AS 2) translate to SQL 3) check candidate key 4) participation
<code>id CHAR(10),</code>	
<code>days INTEGER,</code>	
<code>Name CHAR(10),</code>	
<code>PID CHAR(10) UNIQUE,</code>	

`PRIMARY KEY (id)`

→ not bold  $\text{own}(\underline{\text{id}}, \underline{\text{PID}}, \underline{\text{days}})$  underline id or PID

`CREATE TABLE UniAcct (`

<code>salary REAL,</code>	* 1) write underline AS 2) translate to SQL 3) check candidate key 4) participation
<code>id CHAR(10)</code>	

`PRIMARY KEY (id)`

`CREATE TABLE`

`Prof (`

<code>Name CHAR(10),</code>	* 1) write underline AS 2) translate to SQL 3) check candidate key 4) participation
<code>PID CHAR(10),</code>	
<code>PRIMARY KEY (PID);</code>	

<code>PRIMARY KEY (PID);</code>	# entity set
---------------------------------	--------------

\* 1) `CREATE TABLE own (`

2) `id CHAR(10),`

3) `PID CHAR(10) UNIQUE,`

4) `days INTEGER,`

`PRIMARY KEY(id),`

`FOREIGN KEY(id)`

`REFERENCE UniAcct,`

`FOREIGN KEY(PID)`

`REFERENCE Professor);`

} Attr

5) participation

→ one bold arrow

CourseTeach(Name, CRN, PID) Professor(Name, PID)  
CREATE TABLE CourseTeach  
Name CHAR(10),  
CRN CHAR(10),  
PID CHAR(10), NOT NULL *# every course needs prof.*  
PRIMARY KEY(CRN)  
FOREIGN KEY(PID)  
REFERENCE Professor *# candidate key needs to be unique*

example 4)

CourseTeach(Number, Name, PID)  
CREATE TABLE CourseTeach  
Number CHAR(10),  
Name CHAR(10),  
PID CHAR(10), NOT NULL  
PRIMARY KEY(Number, PID)  
FOREIGN KEY(PID)  
REFERENCE Professor  
ON DELETE  
(CASCADE);

→ example 5)

Manager(SSN, name, lot) Dept(Did, Dname, budget) Manages(SSN, Did, since)  
CREATE TABLE Manager(  
Did CHAR(10),  
Dname CHAR(10),  
budget INTEGER,  
PRIMARY KEY(Did))  
CREATE TABLE Manages(  
SSN CHAR(10),  
name CHAR(10),  
lot INTEGER,  
PRIMARY KEY(SSN))  
CREATE TABLE Dept(  
Did CHAR(10),  
Dname CHAR(10),  
since DATE,  
PRIMARY KEY(Did))  
FOREIGN KEY(SSN)  
REFERENCE Manager  
FOREIGN KEY(Did)  
REFERENCE Dept );

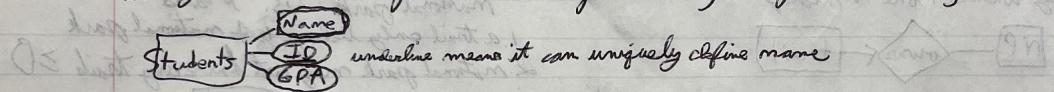
## ER Model - entity relationship ~ mind map

entity - object that exists & distinguishable from other obj., represented by set of attributes  
concrete like person/book ; abstract like holiday/disease

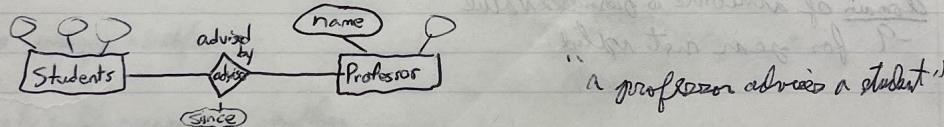
entity set (entities w/same type), not always disjoint (different entity sets, 1 entity)  
all people have account at bank

express logical structure graphically with ER diagram

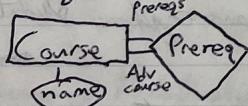
entity set has 3 components: rectangles = name, ellipses = attributes, lines link attributes



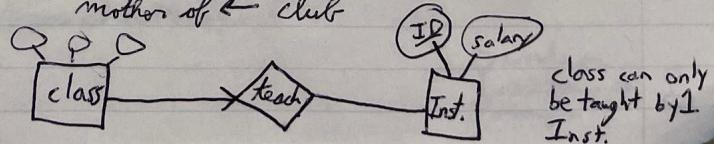
relationship : diamonds = relationships sets, lines linking 2+ entity sets to relationship sets



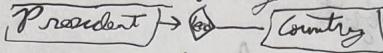
Prerequisite for CSCI 4333 are CSCI 3333 & CMPE 3333



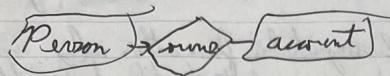
- Key Constraints - number of entity to number of other entity
- ex) student advised by only 1 professor, professor advises ≥ 1 student
- many-to-one constraint (Bowling Club → was born in)
- one-to-one constraint (1 man to 1 woman in marriage, or unmarried)
- one-to-many constraint (1 woman, 0 ≤ children, each child has only 1 mom)
- many-to-many (mother of ← club)



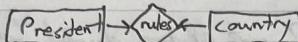
my answers

Current President vs Country → one to one constraint  


Bank Account vs Person - many-to-many

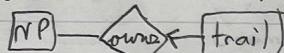


correct answers



Participation Constraints (at least one)  
cannot be described by key constraint

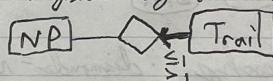
my answer: one to many



national park vs. trails

1. a trail only belongs to 1 national park
2. national park may not have any trails  $\geq 0$

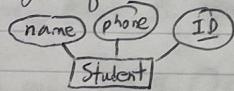
correct answer:



domain of attribute is permitted value  
-9 for year not valid

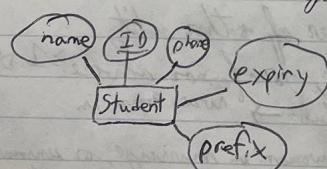
Dating App facts index better than just index, millions of calculations needed

ER Diagram for Tim Database 1) each student reports name & phone 2) student has ID



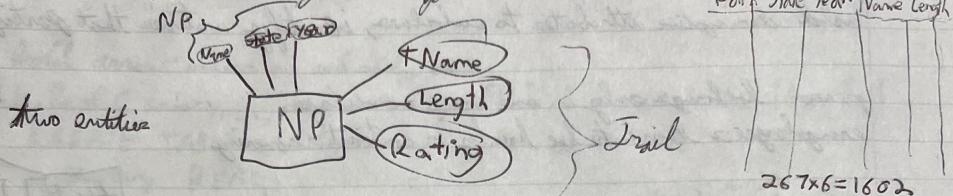
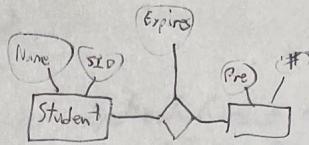
3) report expired date for phone 4) store pre-fix & number

my answer:



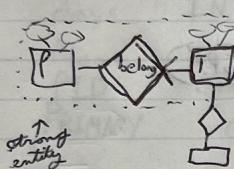
vs attribute of entity

entity set related to attribute by relationship set



Ranking	Point	State	Year	Name	Length

$267 \times 6 = 1602$



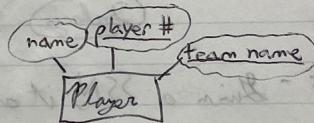
Part	Belong	Trail
YS	YS	GP
AC	AC	NB
S	S	PO
Gr	YS	MH
$4 \times 3$	$263 \times 2$	$263 \times 3$
		$= 1327$

eliminate redundancy

Key - unique identifier

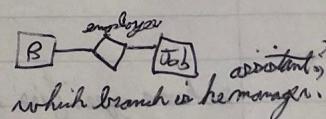
weak entity - no unique identifier

- associate with strong entity to identify  
... is considered a weak entity



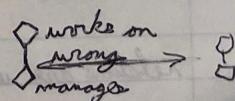
### Ternary Relationships (higher order)

- can be converted into set of binary relationships
- employee at  $\leq$  branches job descriptions
- bad if subsets of entity don't use that connection need constraints



### Aggregation

- manager see tasks of employee at another branch



example 6) similar to ex5 but Manager < PRIMARY KEY (DID) >  
 no more SSN

NP × State

Np.state = State.state

name	Np.state	State.state	capital
YS	WY	WY	Ch
YS	WY	TX	An
BB	TX	WY	Ch
BB	TX	TX	An

is easier

S.I.D of sailors who receive both red & green boat? Table A: S.I.D of sailors who receive red boat  
 (SELECT S.I.D)

FROM Reserve R, Boat B

WHERE R.b.id = B.b.id

AND B.color = 'red'

INTERSECT

(SELECT S.I.D FROM Reserve R2, Boat B2 WHERE R2.b.id = B2.b.id  
 AND B2.color = 'green')

two different workers  
 or green, all that changes is INTERSECT to UNION

red boat but never receive green boat  
 change INTERSECT to EXCEPT

SELECT R.s.id, R.name  
 FROM Reserve R, Boat B, Sailor S  
 Reserve R2, Boat B2, Sailor S2

WHERE

R.s.id = R2.s.id AND R.b.id = B.b.id AND R2.b.id = B2.b.id  
 AND B.color = 'red' AND B2.color = 'green' AND  
 B.b.id < > R2.b.id) AND S.s.id = R.s.id AND S2.s.id = R2.s.id

another solution

SELECT name FROM Sailor  
WHERE rating < 7 AND age > 34

r > 4, reserves all boats:

SELECT name FROM Sailor S

WHERE rating > 4 AND

NOT EXISTS

SELECT B.b\_id FROM Boat B

EXCEPT

SELECT B2.b\_id FROM Buy B2

WHERE B2.s\_id = S.s\_id

)

SELECT AVG(rating) FROM Sailor

SELECT company, MIN(rating) AS min\_rating

FROM Sailor

GROUP BY company

For each yellow boat reserved by sailors from different companies, find total # of reservations

SELECT bid, COUNT(\*) FROM

Buy B NATURAL JOIN Boat B2 NATURAL JOIN Sailor S

WHERE B2.color = "yellow"

GROUP BY bid

HAVING COUNT(DISTINCT S.company) > 1

$R(A, B, C)$   $F = \{A \rightarrow C, C \rightarrow A\}$  fill according to  $F^+$   
 $A \rightarrow C$        $B^+ = B$        $AC^+ = ABC$        $AC^+ = AC$   
 $C \rightarrow A$        $C^+ = AC$        $BC^+ = ABC$   
 $AB^+ = ABC$        $ABC = ABC$

$R(A, B, C, D, E)$   $F = \{A \rightarrow DE, B \rightarrow CD, A \rightarrow B\}$   
✓  $A \rightarrow DE$ ,  $A \rightarrow ABCDE$   
✓  $A \rightarrow B$ ,  $A \rightarrow ABCDE$   
✗  $B \rightarrow CD$ ,  $B \not\rightarrow ABCDE$

Decomposition:  $ABCDE$   
 $\boxed{BCD}$        $\boxed{ABE}$       BCNF

A	B	C
1	1	2
2	1	3
3	1	2
4	1	2
5	1	3

$C \rightarrow B$  ✓, every value is the same  
 $B \rightarrow AX$

we need AC, AB decomposition  
 A is X, C is Y.  $XY = AC$

$R$  is  $ABC$ ,  $Y$  is  $C$        $R - Y = ABC - C = AB$

$\begin{array}{ccccc} A & \leftarrow & 1 & \rightarrow & A \\ 1 & \leftarrow & 2 & \rightarrow & 1 \\ 2 & \leftarrow & 3 & \rightarrow & 2 \\ 3 & \leftarrow & 1 & \rightarrow & 3 \\ 4 & \leftarrow & 2 & \rightarrow & 1 \\ 5 & \leftarrow & 1 & \rightarrow & 2 \end{array}$

$\rightarrow A \leftarrow A$

$\rightarrow A \leftarrow A$