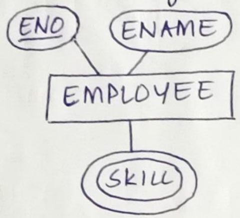
20/01/25

FIRST NORMAL FORM:

- When encountered with a database which is not in 1NF, perform reasoning and analyse if the multivalued / composite attribute can be substituted as a strong entity.
- Initially we discussed that we create a new relation that contains the atomic values of multivalued attribute and the primary key of initial entity as the foreign keys.

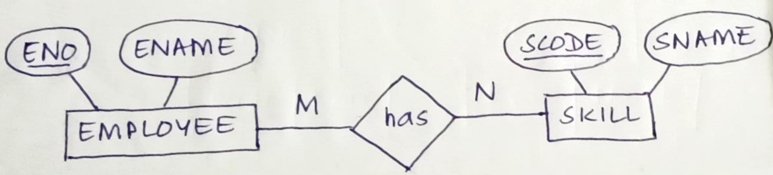
(ie)

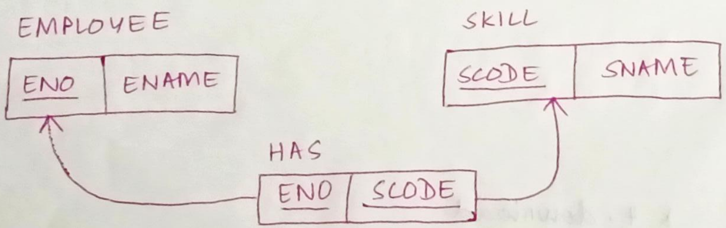


→

ENO	SKILL
101	Python
102	C++
101	Java

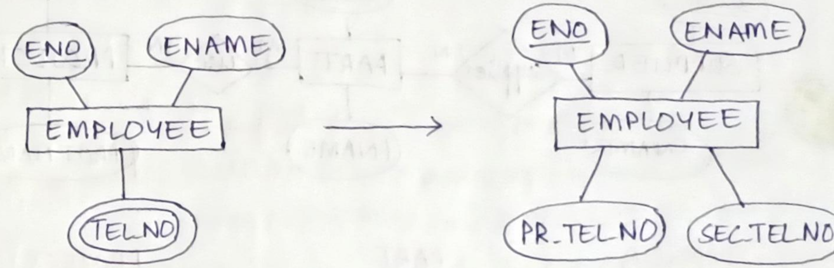
Now, we can model skill as a strong entity as well!





Whereas consider the example where the employee entity having telephone number as the multivalued attribute.

TIP: Instead of modeling attributes such as mail ids, telephone numbers as a multivalued attribute we can add two columns, (ie) primary no and secondary no where primary no will have a not-null constraint.



(Observe that methods such as expansion of primary key, boolean values are not feasible here)

Certain methods of normalizations might not be practically feasible (Eg: telephone number). In such cases, reasoning design is crucial to model a logical data design.

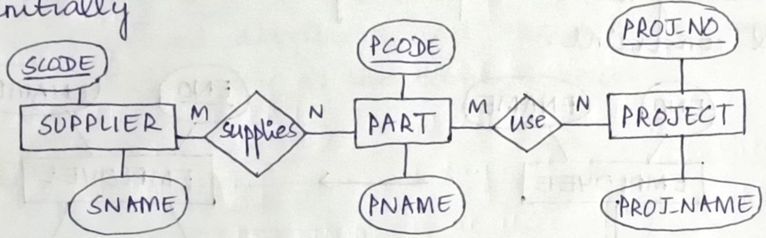
EMPLOYEE

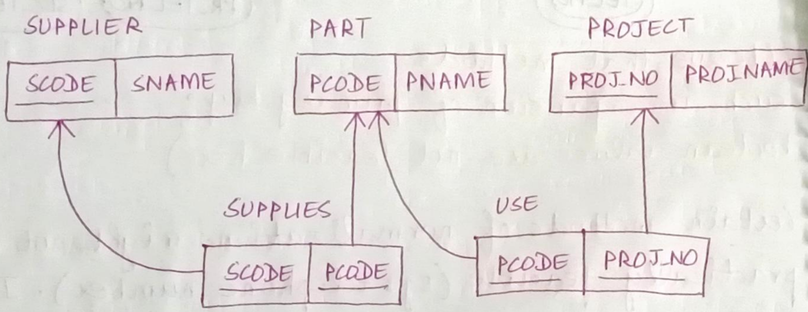
ENO	ENAME	PR_TELNO	SEC_TELNO

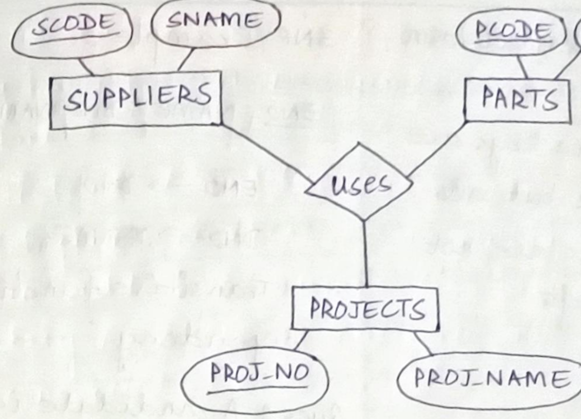
NOT NULL CONSTRAINT

Consider the functional requirement where a supplier can supply many products, a product is supplied by many suppliers and products are used for projects.

Initially







Homeworks:

Representing cardinality ratios in Graph databases: cardinality ratios between entities are represented implicitly / explicitly by the no. of connections (edges) a node can have to other nodes. (one edge between the nodes)

① One to one (1:1)

Each node is connected to exactly one other node by the relationship

(Person {name: "Alice"})
-[:HAS]→(Passport {number: "12345"}) ~~(Multiple outg~~

② One to Many (1:N)

A node connects to multiple nodes through outgoing edges of the same relationship type (multiple outgoing edges from one node)

(Teacher {name: "Anne"})
-[:TEACHES]→(Student {name: "Alice"})
-[:TEACHES]→(Student {name: "Bob"})

③ Many-to-Many (N:M)

Multiple instances of one entity are related to multiple instances of another entity.

(Student {name: "Alice"})
-[:ENROLLED-IN]→(Course {name: "Math"})
-[:ENROLLED-IN]→(Course {name: "Science"})

Hello!

Boyce Codd Normal Form (BCNF):

All candidate keys are determinant but all determinants are not candidate keys.

1NF Example:

EMPLOYEE

<u>ENO</u>	NAME	DOB	BASIC

<u>DNO</u>	DESIGNATION

DEPARTMENT

<u>DNO</u>	DNAME

DNO → DNAME

Full functional dependency

2NF Example:

EMP_WORKS

<u>ENO</u>	NAME	DESIGNATION

<u>PCODE</u>	PNAME	<u>DW</u>	IT	OT

ENO → NAME, DESIGNATION

PCODE → PNAME

Partial functional

ENO, PCODE, DW are determinants. But (ENO, PCODE, DW) candidate

DNO is a determinant but not a candidate key

<u>ENO</u>	NAME	DNO	DNAME

ENO → DNO

DNO → DNAME

Transitive functional dependeny

Ques: A candidate is interviewed only once on a day. If a candidate is unsuccessful, he/she can appear for interview on another day. An interviewee is assigned a specific room on a day. The same room can be assigned to more than one interviewee on the same day but at different times.

Identify the determinants in table INTERVIEW:

CANDIDATE

<u>CID</u>	CNAME

INTERVIEW

<u>CID</u>	<u>INTID</u>	<u>DOI</u>	TOI	ROOMNO

INTERVIEWER

<u>INTID</u>	INTNAME

Dependencies:

① CID, DOI → INTID, TOI, ROOMNO (c) (as a candidate is interviewed only once a day)

② DOI, TOI, ROOMNO → CID, INTID (c)

③ ~~CID, INTID~~ INTID, DOI → ROOMNO (does not define all the attributes, so not CK)

④ INTID, DOI, TOI → CID, ROOMNO (c)

(For a value of x, there must be exactly only one value of y, when x → y)

Prime attributes - All the attributes of relation INTERVIEW

For a relation to be in BCNF, every determinant must be a candidate key. Dependency ③ is not a candidate key, hence INTERVIEW is not in BCNF

How to normalize it to BCNF:

Decompose the relation in such a way that:

- Create a relation with all attributes except the attribute that are dependent by a determinant which is not a candidate key
- For ~~a~~ each determinant that is not a candidate key, create a seperate relation with the determinant and the attributes it determines. Mark as candidate key the determinant of the attributes

CANDIDATE

<u>CID</u>	CNAME

INTERVIEWER

<u>INTID</u>	INTNAME

INTERVIEW

<u>CID</u>	<u>INTID</u>	<u>DOI</u>	TOI

INTER_ROOM

<u>INTID</u>	<u>DOI</u>	ROOM NO

EXE 1: Normalize the given relation to BCNF :

$R(\underline{A}, B, C, D, E, F, G, H, I)$

$A \rightarrow B, C$

$D \rightarrow E, F$

$A, G \rightarrow D, H, I$ (c)

$D, G, H \rightarrow A, I$ (c)

$G, H, I \rightarrow A, D$ (c)

$G, D \rightarrow I$

Candidate key :

AG, DGH, GHI

By Armstrong axioms,

$AG \rightarrow D, H I$

$A \rightarrow B, C$

So, $AG \rightarrow B, C, D, H, I$

$D \rightarrow E, F$

$D, G, H \rightarrow A, I, E, F$

So, $D, G, H \rightarrow B, C, I, E, F$

~~So~~ Also,

$GHI \rightarrow A, D$

$GHI \rightarrow B, C, E, F$

Hence, AG, DGH and GHI are candidate keys.

Hello
Hello

So, we consider the dependencies where the determinants are also the candidate keys first:

$R_1(\underline{A}, B, C)$ both are in BCNF

$R_2(\underline{D}, E, F)$

Amongst the 3 candidate keys, we consider 1 and form a relation as it includes all the attributes

$R_{12}(\underline{G, H, I}, A, D)$

In this relation, there exists a determinant $GD \rightarrow I$, Hence it is not in BCNF.

Normalizing :

$R_3(G, H, A, D)$ $R_4(G, D, I)$

EXE 2: Normalize the given relation to BCNF :

$R(A\ B\ C\ D\ E\ F\ G\ H\ I)$

$A \rightarrow B$

$C \rightarrow DEF$

$C \rightarrow E$

$E \rightarrow F$

(c) $AG \rightarrow CHI$

(c) ~~GH~~ CGH $\rightarrow AI$

$CG \rightarrow I$

(c) $GHI \rightarrow AC$

Candidate keys :

① $AG \rightarrow CHI$

$\Rightarrow AG \rightarrow EHI$ ($C \rightarrow E$)

$\Rightarrow AG \rightarrow FHI$ ($E \rightarrow F$)

$\Rightarrow AG \rightarrow DEFHI$ ($C \rightarrow DEF$)

How to bring B in ?

② $CGH \rightarrow AI$

$\Rightarrow CGH \rightarrow BI$ ($A \rightarrow B$)

How to bring in the rest ?

③ $GHI \rightarrow AC$

$\Rightarrow GHI \rightarrow BDEF$

($A \rightarrow B$, $C \rightarrow DEF$)

Relations :

$R_1(\underline{A}, B)$

$R_2(\underline{C}, D, E, F)$

~~$R_3(C, E)$~~

$R_4(\underline{E}, F)$

All the above relations are in 3NF & BCNF as all the determinants are candidate keys.

Now consider,

$R_5(\underline{G, H, I}, A, C)$

where all the attributes are prime attributes & GHI is the candidate key. Now, $CG \rightarrow I$ disallows BCNF. So,

$R_5(G, H, A, C)$

$R_6(\underline{C, G}, I)$

CANDIDATE

<u>CID</u>	CNAME

INTERVIEWER

<u>INTID</u>	INTNAME

From ques no.1

INTERVIEWER

CID	INTID	DOI	TOI	ROOMNO

Dependencies :

1. CID, DOI → INTID, TOI, ROOMNO
2. ROOMNO, ~~INTID~~ DOI, TOI → INTID, CID
3. INTID, DOI, TOI → CID, ROOMNO
4. INTID, DOI → ROOMNO.

Now, BCNF Normalized relation looks like :

CANDIDATE

<u>CID</u>	CNAME

INTERVIEWER

<u>INTID</u>	INTNAME

INTERVIEW

<u>CID</u>	<u>INTID</u>	<u>DOI</u>	TOI

INTERVIEW_ROOM

<u>INTID</u>	<u>DOI</u>	ROOMNO

Assume that 25 people are interviewed on a single day in room no.5. And a change has been induced to change the room no. to 10.

Before BCNF normalization → 25 tuples has to be updated

After BCNF normalization → only 1 tuple of relation INTERVIEW ROOM to be updated

→ Hence, Updation anomaly is overriden

...m ques no.2,

consider, AG → CHI, A → B, C → DEF

By armstrong's axioms,

AG → B DEF CHI.

Now, A → B denies 2NF, so,

R_1

<u>A</u>	<u>G</u>	C	H	I	D	E	F

R_2

<u>A</u>	B

✓

Now, AG → C, C → DEF denies 3NF, so,

R_{11}

<u>A</u>	<u>G</u>	C	H	I

GC → I

R_{12}

<u>C</u>	D	E	F

Now C → DEF and E → F denies 3NF, so,

R_{121}

<u>C</u>	D	E

✓

R_{122}

<u>E</u>	F

✓

Now, all the attributes are prime and there exists CG → I which denies BCNF,

R_{111}

<u>A</u>	<u>G</u>	C	H

✓

R_{112}

<u>G</u>	<u>C</u>	I

✓

So, the relations are :

<u>A</u>	B

<u>C</u>	D	E

<u>A</u>	<u>G</u>	C	H

<u>E</u>	F

<u>G</u>	<u>C</u>	I

4th NORMAL FORM:

Disallows Non-trivial multivalued dependency

Eg: R(A, B, C)

$A \twoheadrightarrow B$ (A multidetermines B)

$A \twoheadrightarrow C$ (A multidetermines C)

[i.e., for a value of A there are multiple values of C and B]

Example 1:

EMP_SKILL_HOBBY

ENO SKILL HOBBY

ENO $\twoheadrightarrow$ SKILL

ENO $\twoheadrightarrow$ HOBBY

Solution - has to be decomposed in such a way that

R_1 A B R_2 A C

Has to be decomposed as:

ENO SKILL ENO HOBBY

EMP_SKILL EMP_HOBBY

5th NORMAL FORM (PROJECT-JOIN NORMAL FORM)

Not a practical Normal form. Disallows non-trivial join dependencies.

Example 2:

SPJ

SNO	PNO	JNO
S_1	P_1	J_1
S_1	P_2	J_1
S_2	P_1	J_2
S_2	P_3	J_2
S_3	P_1	J_3
S_3	P_2	J_3

SP(SNO, PNO)

SNO	PNO
S_1	P_1
S_1	P_2
S_2	P_1
S_2	P_3
S_3	P_1
S_3	P_2

PJ(PNO, JNO)

PNO	JNO
P_1	J_1
P_2	J_1
P_3	J_2
P_1	J_2
P_2	J_3
P_1	J_3

SJ(SNO, JNO)

SNO	JNO
S_1	J_1
S_2	J_2
S_3	J_3

Natural Join:

i) Between SP, PJ.

SP $\bowtie$ PJ

SNO	PNO	PNO	JNO
S_1	P_1	P_1	J_1
S_1	P_1	P_1	J_2
S_1	P_1	P_1	J_3
S_1	P_2	P_2	J_1
S_1	P_2	P_2	J_3
S_2	P_1	P_1	J_1
S_2	P_1	P_1	J_2
S_2	P_1	P_1	J_3
S_2	P_3	P_3	J_2
S_3	P_1	P_1	J_1
S_3	P_1	P_1	J_2
S_3	P_1	P_1	J_3
S_3	P_2	P_2	J_1
S_3	P_2	P_2	J_3

ii) Between PJ & SJ.

PJ $\bowtie$ SJ.

PNO	JNO	SNO
P_1	J_1	S_1
P_2	J_1	S_1
P_3	J_2	S_2
P_1	J_2	S_2
P_2	J_3	S_3
P_1	J_3	S_3

iii) Between SP & SJ

SP $\bowtie$ SJ.

SNO	PNO	JNO
S_1	P_1	J_1
S_1	P_2	J_1
S_2	P_1	J_2
S_2	P_3	J_2
S_3	P_1	J_3
S_3	P_2	J_3

Observe lossy decomposition due to spurious tuples in the above natural Joins

Natual Join between SP, PJ and SJ

Condition : SP.P = PJ.P AND PJ.J = SJ.J AND SP.S = SJ.S

SP	PJ	SJ
S_1	P_1	J_1
S_1	P_2	J_1
S_2	P_1	J_2
S_2	P_3	J_2
S_3	P_1	J_3
S_3	P_2	J_3

Example 3 :

SPJ

SNO	PNO	JNO
S_1	P_1	J_1
S_1	P_2	J_1
S_1	P_1	J_2
S_1	P_3	J_2
S_2	P_1	J_3
S_2	P_2	J_3

Decomposition :

PJ (PNO, JNO)

PNO	JNO
P_1	J_1
P_2	J_1
P_1	J_2
P_3	J_2
P_1	J_3
P_2	J_3

SJ (SNO, JNO)

SNO	JNO
S_1	J_1
S_1	J_2
S_2	J_3

SP (SNO, PNO)

SNO	PNO
S_1	P_1
S_1	P_2
S_1	P_3
S_2	P_1
S_2	P_2

Natural Joins :

① Between PJ, SJ

PJ.J = SJ.J

PNO	JNO	SNO
P_1	J_1	S_1
P_2	J_1	S_1
P_1	J_2	S_1
P_3	J_2	S_1
P_1	J_3	S_2
P_2	J_3	S_2

② Between SJ, SP

SJ.S = SP.S

JNO	SNO	PNO
J_1	S_1	P_1
J_1	S_1	P_2
J_1	S_1	P_3
J_2	S_1	P_1
J_2	S_1	P_2
J_2	S_1	P_3
J_3	S_2	P_1
J_3	S_2	P_2

③ Between PJ, SP

PJ.P = SP.P

JNO	PNO	SNO
J_1	P_1	S_1
J_1	P_1	S_2
J_1	P_2	S_1
J_1	P_2	S_2
J_2	P_1	S_1
J_2	P_1	S_2
J_2	P_3	S_1
J_3	P_1	S_1
J_3	P_1	S_2
J_3	P_2	S_1
J_3	P_2	S_2

Natual Join between SP

PJ.J = SJ.J AND SJ.S = SP.S AND SP.P = PJ.P

PNO	JNO	SNO
P_1	J_1	S_1
P_2	J_1	S_1
P_1	J_2	S_1
P_3	J_2	S_1
P_1	J_3	S_2
P_2	J_3	S_2

Example 4 :

SPJ

SNO	PNO	JNO
S_1	P_1	J_1
S_2	P_1	J_1
S_1	P_2	J_1
S_2	P_2	J_1
S_1	P_1	J_2
S_1	P_3	J_2
S_2	P_1	J_3
S_2	P_2	J_3

PJ

PNO	JNO
P_1	J_1
P_2	J_1
P_1	J_2
P_3	J_2
P_1	J_3
P_2	J_3

Decomposition :

SP

SNO	PNO
S_1	P_1
S_2	P_1
S_1	P_2
S_2	P_2
S_1	P_3

SJ

SNO	JNO
S_1	J_1
S_2	J_1
S_1	J_2
S_2	J_3

Natural Join :

① Between SP & SJ.

PNO	SNO	JNO
P_3	S_1	J_2
P_1	S_1	J_1
P_1	S_1	J_2
P_1	S_2	J_1
P_1	S_2	J_3
P_2	S_1	J_1
P_2	S_1	J_2
P_2	S_2	J_1
P_2	S_2	J_3
P_3	S_1	J_1

② Between SP & PJ.

SNO	PNO	JNO
S_1	P_1	J_1
S_1	P_1	J_2
S_1	P_1	J_3
S_2	P_1	J_1
S_2	P_1	J_2
S_2	P_1	J_3
S_1	P_2	J_1
S_1	P_2	J_3
S_2	P_2	J_1
S_2	P_2	J_3
S_1	P_3	J_2

③ Between PJ and SJ.

PNO	JNO	SNO
P_1	J_1	S_1
P_1	J_1	S_2
P_2	J_1	S_1
P_2	J_1	S_2
P_1	J_2	S_1
P_3	J_2	S_1
P_1	J_3	S_2
P_2	J_3	S_2

Natural join between

SP.P = PJ.P AND PJ.J = SJ.J

AND