1NF, 2NF, 3NF,
Compute closure, BCNF
Find all keys from FDs

Concept review

- K is a superkey for relation R if K functionally determines all attributes of R
- K is a key for R if K is a superkey, but no proper subset of K is a superkey (key is minimal superkey)
- Key might not be unique for relation R
- An attribute is prime if it is a member of any key
- X --> A violates 3NF iff X is not a superkey and A is not prime

1NF

For a table to be in the First Normal Form, it should follow the following 4 rules:

(i) It should only have single(atomic) valued attributes/columns

(ii) Values stored in a column should be of the same domain

(iii) All the columns in a table should have unique names

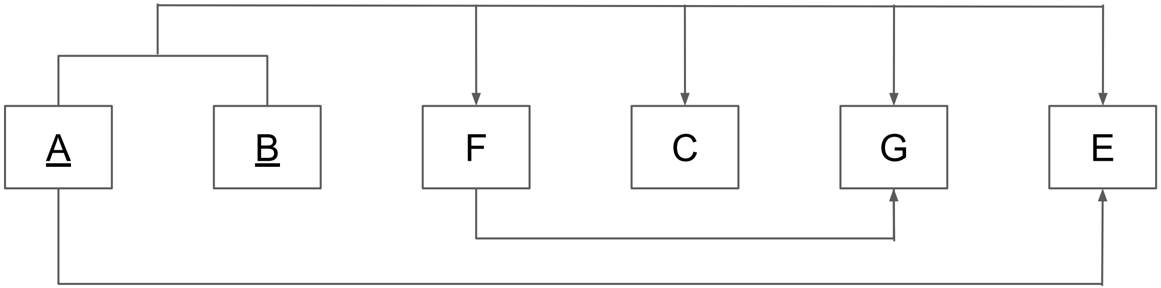
(iv) And the order in which data is stored, does not matter

Partial dependency & Transitive dependency

- X --> Y is a partial dependency if for some A ∈ X, X − A --> Y
  (A is redundancy in X, in terms of determining Y)

- X --> Y is a transitive dependency if there exists Z such that X --> Z and Z --> Y, and that Z is neither a key nor a subset (prime) of any key

E.g., Label partial dependency and transitive dependency for the following relation.



(composite) key: A,B

Fig.1

partial dependency: A,B --> E (since B is redundancy)

transitive dependency: A,B --> G (since A,B --> F, and F --> G, and F is neither a key nor prime)

2NF

A relation schema R is in second normal form (2NF) if (i) it is in 1NF and (ii) there is no partial dependency from any (composite) key to any non-key (non-prime) attribute in R

Decompose the example schema (Fig.1) so that they are in 2NF, and link them back with foreign key

Idea is to remove partial dependency in Fig.1 from the key A,B to the non-prime E

{A, B} is composite primary key,
A itself is a foreign key

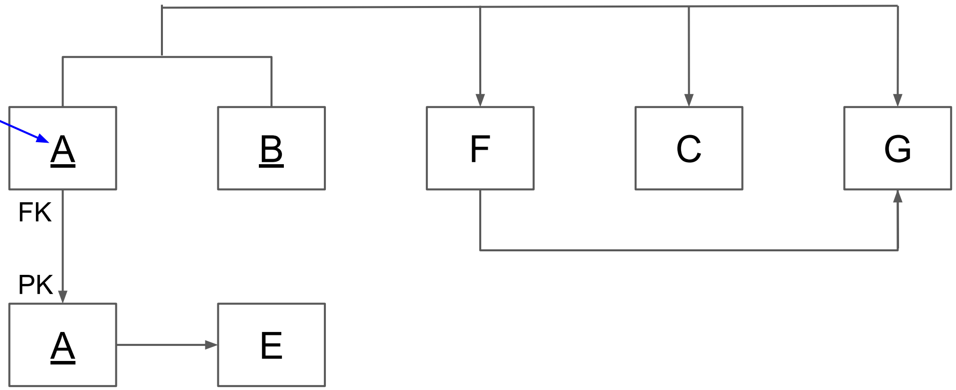


Fig.2: in 2NF, but not in 3NF (F --> G violates 3NF since F is not a super key and G is not prime)

3NF

A relation schema R is in third normal form (3NF) if (i) it is in 2NF and (ii) there is no transitive dependency from any key to any non-key (non-prime) attribute in R

Decompose the relation schema in Fig.2 so that they are in 3NF, and link them back with foreign key.

Idea is to remove the transitive dependency in Fig.2 from key {A,B} to the non-prime G

{A, B} is composite primary key,
A itself is a foreign key

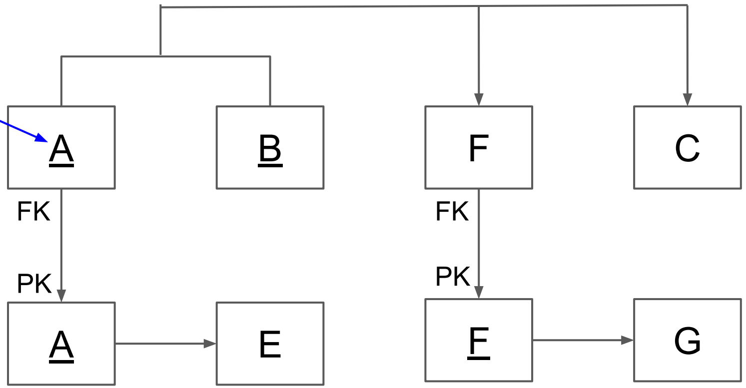


Fig.3, in 3NF

Compute Closure

$$\begin{aligned}
&Y^+ = Y \\
&\text{if } X \subseteq Y \text{ and } X \to A \\
&\text{then } Y^+ \leftarrow Y^+ \cup A
\end{aligned}$$

In 3NF ⇔ $\forall$ nontrivial X→Y (nontrivial means Y is not contained in X), X is a superkey, **or** Y is prime

In BCNF ⇔ $\forall$ nontrivial X→Y (nontrivial means Y is not contained in X), X is a superkey

Find all keys from FDs

Find all keys in R(A,B,C,D,E) with FDs: A→B, BC→E, ED→A

Step1: find all attributes that show up in Left hand side (Left): ABCED

Step2: find all attributes that show up in both side (Both): ABE

Step3: find all attributes that only show up in Left hand side (LeftOnly = Left \Both): CD

Step4: add Both to LeftOnly one attribute at a time, from empty to all attributes:

LeftOnly CD	Both ABE
CD	$(CD)^+ = CD$
★CDA	$(CDA)^+ = CDA \xrightarrow{A\to B} CDAB \xrightarrow{BC\to E} CDABE$
★CDB	$(CDB)^+ = CDB \xrightarrow{BC\to E} CDBE \xrightarrow{ED\to A} CDBEA$
★CDE	$(CDE)^+ = CDE \xrightarrow{ED\to A} CDEA \xrightarrow{A\to B} CDEAB$
x CDAB	
x CDAE	
x CDBE	
x CDABE	