# **LECTURE 5 - CANONICAL COVERS**

Canonincal Cover Minimal Cover
Schema Refinement = Can **FD**'s **F** be minimized/more compact?

**G** is a *canonical/mnimal cover* of **F** when:

- 1. **G** is *equivalent* to **F** (**G F**)
- 2. **G** is *minimal* (if any part of G is removed G F)
- 3. Every **FD** in **G** has a single attribute on the RHS.

A canonical cover **G** is minimal in two respects:

- 1. Every FD in **G** is "required" in order for G to be equivalent to F.
- 2. Every FD in **G** is as "small" as possible (only one attribute RHS)

# **Computing Canonical Cover**

A FD in the set is *redundant* if it can be derived from the other FD's in the set. ORDER MATTERS

- 1. Decompose all FDs in standard form
  - i.e. only one attribute on the RHS
- 2. Check LHS for **Redundant Attributes**:
  - o Check FD's with attributes → on the LHS fore redun
  - for each FD AB
     C in G, check if A or B on the LHS is redundant
  - Can A be removed from ABC?
    - Check A
    - if C A then A is Redundant
    - then A can be removed from AB

<sup>\*</sup> multiple canonical covers can exists



- Check B<sup>r</sup>
- if C B then A is Redundant
- then B can be removed from AB

### 3. Remove Redundant FD's:

• Remove each FD one at a time, and check fro the closure of F with it removed. If the result can be acheived without the FD it is redundant.



- For every FD X A in G
  - Remove {X A} from G; call the result G'
  - Compute X under G'
  - If A X under G', then X A is redundant and hence remove X A from G.

### **Example:**

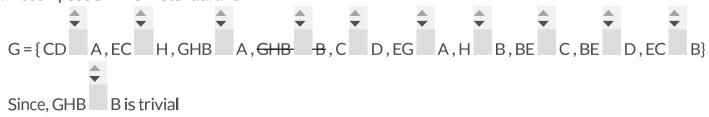
This example explains computing canonical covers well, just dont move to the next line until you know exactly whats going on first.

 $R = \{A, B, C, D, E, F, G, H\}$ 



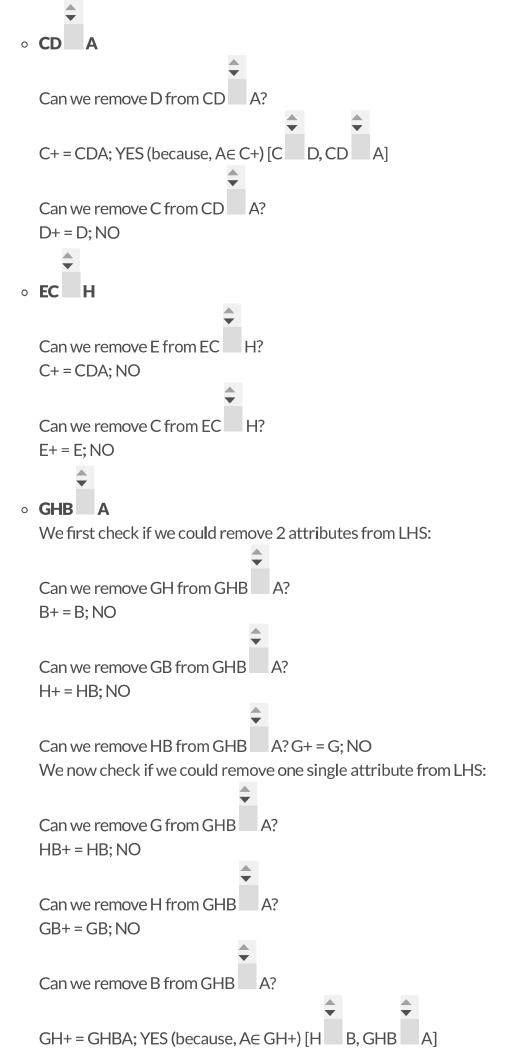
Find a canonical cover for F?

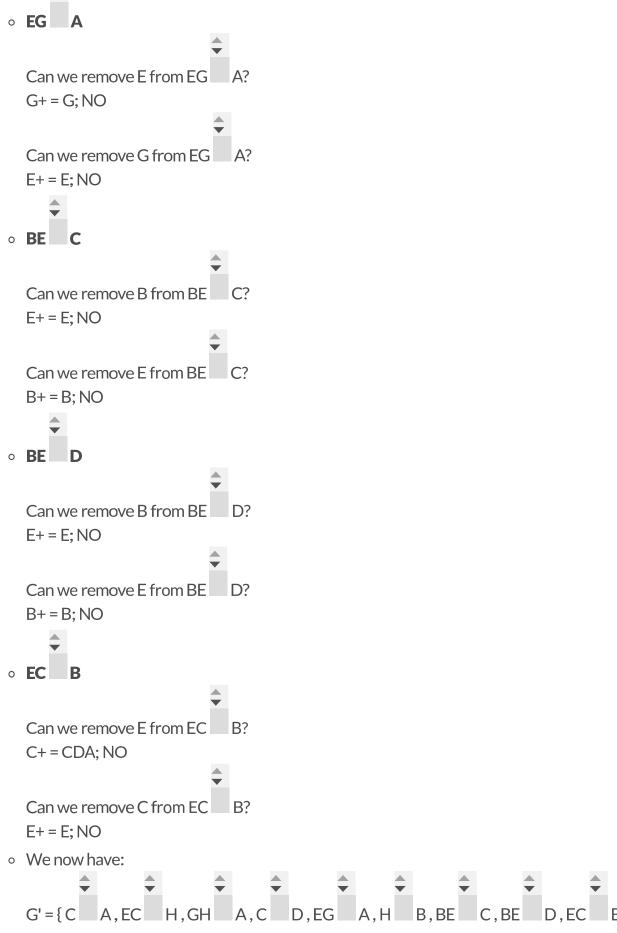
1. Decompose all FDs in standard form



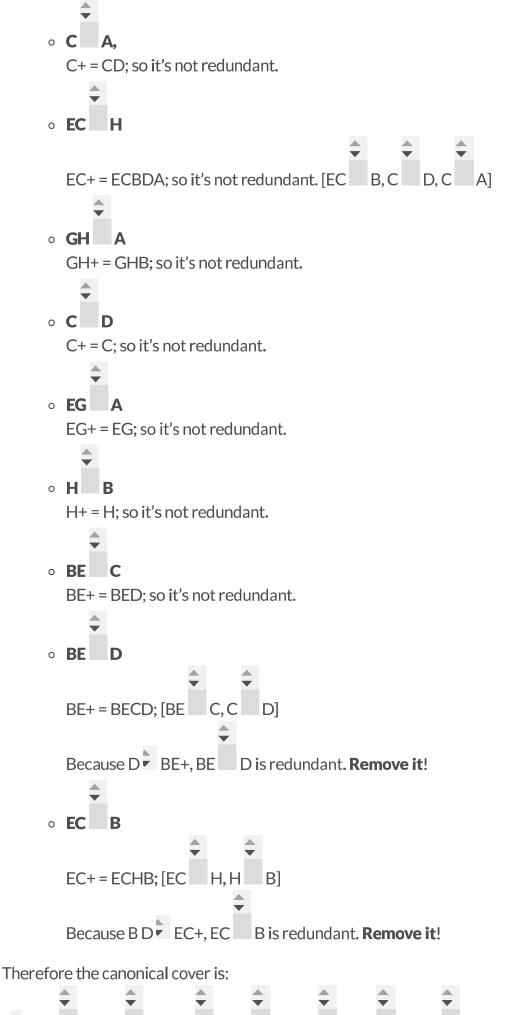
2. Eliminate unnecessary attributes from LHS

We need to check all the FDs that have more than one attribute on the LHS:





3. Remove redundant FD(s)



# **How To Deal With Redundancy**

Decompose the relation into seperate relations:

### Before:

Name	Address	RepresentingFirm	Spokesperson
Carrie Fisher	123 Maple	Eone	Joe Smith
Harrison Ford	789 Palmer	Eone	Joe Smith
Mark Hamil	456 Oak	LoserRus	Mary Johns

### After:

Name	Address	RepresentingFirm
Carrie Fisher	123 Maple	Eone
Harrison Ford	789 Palmer	Eone
Mark Hamil	456 Oak	LoserRus

RepresentingFirm	Spokesperson
Eone	Joe Smith
LoserRus	Mary Johns

# **Properties of a decomposition?**

- 1. Lossless-join (a must)
- 2. **Dependency-preserving** (desirable)

Α	В	С
1	2	3
4	2	5

### Decomposed into:

A	В
1	2
4	2

В	С
2	3
2	5

# When joined:

Α	В	С
1	2	3
4	2	5
4	2	3
1	2	5

## Lossless-join:

A	В	С
1	2	3
4	2	5

<sup>\*</sup> no new tuples 🙌

# **Dependency-preservation**

Α	В	С	D
1	2	5	7
4	3	6	8

## Decomposed into:

Α	В
1	2
4	3

|--|

В	С	D
2	5	7
3	6	8

\* Dependency-preservation would preserve A D