Midterm Review

Finding Candidate keys

- R = { A, B, C, D, E, F, G, H}
- F={CH G,

 A BC,

 B CFH,

 E A,

 F EG}
- 1. Start with all the attributes not present on the RHS = { D }
- 2. Calculate closure of { D } and all { DX }
 - ∘ {D} = D
 - {DA} = ADBCFHEG = candidate key
 - {DB} = DBCFHEGA = candidate key
 - {DC} = DC
 - {DE} = DEA stop, DA already a candiadte key thus we know DE = candidate key
 - {DF} = DFE stop, DE already a candiadte key thus we know DF = candidate key
- 3. Candidate Keys = { DA, DB, DE, DF } = 4 possibilites

Finding Canonical(Minimal) Cover

- 1. Decompose all FDs in standard form
 - o i.e. only one attribute on the RHS
- 2. Check LHS for **Redundant Attributes**:
 - 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 **AB C**?
 - Check A
 - if C A then A is Redundant
 - then A can be removed from AB
- Can **B** be removed from **AB C**?
 - Check B^r
 - 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.

1NF

1. No attribute is allowed to be composite or multi valued

Example:

The following relation is **not** in 1NF: Student (SID, SName, {(Courseld, CouseName, Grade)})

2NF

2. Every non-prime attribute of relation is fully functionally dependent on the primary key

For each non-key attribute, ask:

If I knew the value for part of the Primary-Key, could I tell what the value for a non-key attribute would be?

Example:

Inventory (Item, Supplier, Cost, SupplierAddress)

If I know just Item, can I find out Supplier Address? NO

If I know just Supplier, can I find out SupplierAddress? YES

SupplierAddress is NOT fully functionally dependent upon the ENTIRE Primary-Key NOT 2NF



3NF

One of the 3 must be met for every FD X A:

Need to compute cnadidate keys in order to check!



- 2. LHS is a superkey (i.e. a key is contained in LHS)
- 3. **RHS** is part of any key of **R**

BCNF

One of the 3 must be met for every FD X



- 2. X is a superkey
- 3. RHS is part of any key of R

BCNF can always obtain lossless-join decomposition

BCNF is not always dependency-preserving

Synthetic 3NF Decompostion

- 1. Compute the canonical cover **F**
- 2. Create relations
- 3. Check if at least one of the keys exists in the above relations
- 4. Add an extra relation containing those attributes that form any key of **R**

Example:

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

2. Create the relations:

R	F
$R^{\prime} = \{A, C, D\}$	F = {C AD}
$R^{\prime} = \{E, C, H\}$	F = {EC H}
$R' = \{A, G, H\}$	F = {GH A}
$R^r = \{A, E, G\}$	F = {EG A}
$R^r = \{B, H\}$	F = {H B}
$R^r = \{B, C, E\}$	F' = {BE C}

- 3. Check if at least one of the keys {BEFG, CEFG, EFGH} exists in the above relations. Since none of these keys is in the relations, this decomposition is **not lossless**.
- 4. add an extra relation containing those attributes that form any key of R: $\,$

R	F
$R^{-} = \{B, E, F, G\}$	F\$\$-7 = {}

The relation schema is now lossles as well as dependency preserving.

Is R in 3NF?

R is **NOT** in **3NF**, because CD A violates the 3NF requirements

- 1. CD A is not trivial FD
- 2. CD is not a superkey & CD is not a key
- 3. A is not part of any key of R either

Is Dependency is Preserved?

Given a decomposed set of relations and FD's R, R, and F, F, ..., R, and F, F, ..., F, ..., F, ...

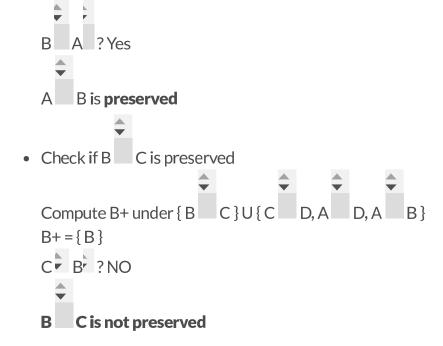
Compute {X} for the original X A originals FD's under the decomposed FD's and check if A {X}

Example:

Decomposed into:

R	F
$R^{r} = \{A, B\}$	F = A B
$R^{r} = \{A, C, D\}$	F = C D, A D, A C

Is the decomposition R = {R1, R2} dependency-preserving?



Lossy Decomposition?

The decompositio of relation R into R1 and R2 is lossy when the join of R1 and R2 does not yield the same relation as in R.