Select cond
Proj → attribute
R → rename

No. :
Date :

2102 Final Notes

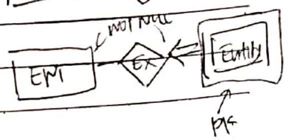
- Superkey : uniquely identifies relation
- Cand. key / key : Minimal superkey
- PK : chosen cand key. → unique & notnull
  - multi attrib PK → not null on each attrib, unique on < tuple >
- FK : tuple ∃ in ref table or (one of attrib is NULL)
  - Conditions (WHERE) → ACCEPT on TRUE
  - Constraints (CHECK) → REJECT on FALSE

- In a RS, all attrib must not be null — can merge or not
- Key Constraint → sgn : single PK
- Total Part Constraint → sgn : NOT NULL on FK. (usually in key+total setting) → NOT NULL on FK
  - unconstrained → dual PK.
- Weak Entity → Must have total participation in identifying relationship set
  - ↳ (identity) only have partial key.
  - ↳ FK w/ ON DELETE CASCADE. NOT NULL
  - ~~NOT N~~ PK for (parent, part_key).

card keys cannot be represented

- Inner Join is commutative → order doesn't matter, first full outer join is commutative.
  - ↳ = cartesian prod w/ A.attr1 = B.attr1 AND A.attr2 = B.attr2 AND ...

- Min cover – Remove redundant attrib on LHS
  - split RHS
  - Remove redundant FDs.
  - } draw HG

- Prime attrib : attrib of some cand. key.
  - identity dep
  - ex: book chapters 
  - ex insurance policy 

ISA → Overlap constraint : allow
→ covering constraint → φ

No. :
Date :

- Overlapping candidate keys → Cfm not BCNF.

Stu	Fac	Mod
~~1~~	~~Comp~~	~~CS1231~~
1	Sci	MA1521
2	Comp	CS1231

SF → M
M → F
Cand key. {SF, SM}

- Check if relation in some NF : ① Find FD projections (attrib closures on each subset)
  - ② Find min cover from projections + draw HyperGraph
  - ③ Find all key to R → also prime attribs.
    - ↳ start from nodes w/o incoming edge
  - ④ Check props

BCNF : LHS → superkey
3NF : BCNF or RHS is prime attrib

Find 3NF comp by synthesis
- ④ Merge FDs w/ same LHS and RHS (non-prime attrib)
- ⑤ Create tables + key table
- ⑥ Remove redundant table

Dep preserving if
F[R₁] ∪ F[R₂] ... ∪ F[Rₙ] ⊨ F[R].

Check lossless
- ↳ ∀ pair of fragments, the intersection must be key of the other
- ↳ Rinse & repeat

> cartesian prod of same set produce dup

Group By. (for each)
- each col A in SELECT
  - A in Group By
  - Aggregate (A)
- Cand. key (unique) of R appears in GroupBy clause

<v₁, v₂>
<v₁', v₂'> same group if v₁ is NOT DISTINCT from v₁'

ORDER BY.
CASE WHEN title = 'Leader' THEN point END DESC,
CASE WHEN title = 'Minion' THEN point END ASC.

Scanned with CamScanner