Midterm 2

- Functional dependency and normal forms
  - Keys and superkeys
  - Closure algorithm
  - 2NF, 3NF, BCNF
- Integrity constraints
  - Check
  - Assertion
  - Trigger
- SQL and Datalog

Keys & Superkeys

- A superkey is ***a set of attributes*** that determines all other attributes
  - The closure of a superkey contains all attributes

- A key is a ***minimal*** superkey
  - No proper subset of a key can be a superkey
  - Different keys can contain different number of attributes

How to find the keys?

- R(A, B, C, D, E, F, G)
- A ⟶ C, DE ⟶ B, BCF ⟶ E

- Step 1: find attributes that do not appear on the right side of any FD
- They must be part of every key because
  they cannot be determined by the other attributes
- So {A, D, F, G} must be in every key

How to find the keys?

- R(A, B, C, D, E, F, G)
- A ⟶ C, DE ⟶ B, BCF ⟶ E

- Step 2: compute the closure of each superset of {A, D, F, G} using the closure algorithm
- {A, D, F, G}+ = {A, D, F, G}, trivial
- {A, D, F, G}+ = {A, D, F, G, C}, using A ⟶ C
- {A, D, F, G} is not a superkey

How to find the keys?

- R(A, B, C, D, E, F, G)
- A ⟶ C, DE ⟶ B, BCF ⟶ E

- Step 2: compute the closure of each superset of {A, D, F, G} using the closure algorithm
- {A, D, F, G, B}+ = {A, D, F, G, C, B}, using {A, D, F, G}+ = {A, D, F, G, C}
- {A, D, F, G, B}+ = {A, D, F, G, C, B, E}, using BCF ⟶ E
- {A, D, F, G, B} is a key
- All supersets of {A, D, F, G, B} are superkeys

How to find the keys?

- R(A, B, C, D, E, F, G)
- A ⟶ C, DE ⟶ B, BCF ⟶ E

- Step 2: compute the closure of each superset of {A, D, F, G} using the closure algorithm
- {A, D, F, G, E}+ = {A, D, F, G, C, E}, using {A, D, F, G}+ = {A, D, F, G, C}
- {A, D, F, G, E}+ = {A, D, F, G, C, E, B}, using DE ⟶ B
- {A, D, F, G, E} is a key
- All supersets of {A, D, F, G, E} are superkeys

How to find the keys?

- R(A, B, C, D, E, F, G)
- A ⟶ C, DE ⟶ B, BCF ⟶ E

- Step 2: compute the closure of each superset of {A, D, F, G} using the closure algorithm
- {A, D, F, G, C}+ = {A, D, F, G, C}, trivial
- {A, D, F, G, C} is not a superkey
- All proper supersets of {A, D, F, G, C} contain B or E
  - They are superkeys, but not keys

How to find the keys?

- R(A, B, C, D)
- A ⟶ CD, BC ⟶ A, D ⟶ AB

- All attributes appear at least once on the right side
  - Start with sets of single attribute
- {A}+ = {A, C, D, B}, {B}+ = {B}, {C}+ = {C}, {D}+ = {D, A, B, C}
- {A} and {D} are two keys
  - No need to compute closure for proper supersets {A} or {D}
- {B, C}+ = {B, C, A, D}
- {B, C} is also a key

2NF & Partial Dependency

- A functional dependency $X \longrightarrow Y$ is a ***partial dependency*** if for some $A \in X$, $(X - \{A\}) \longrightarrow Y$

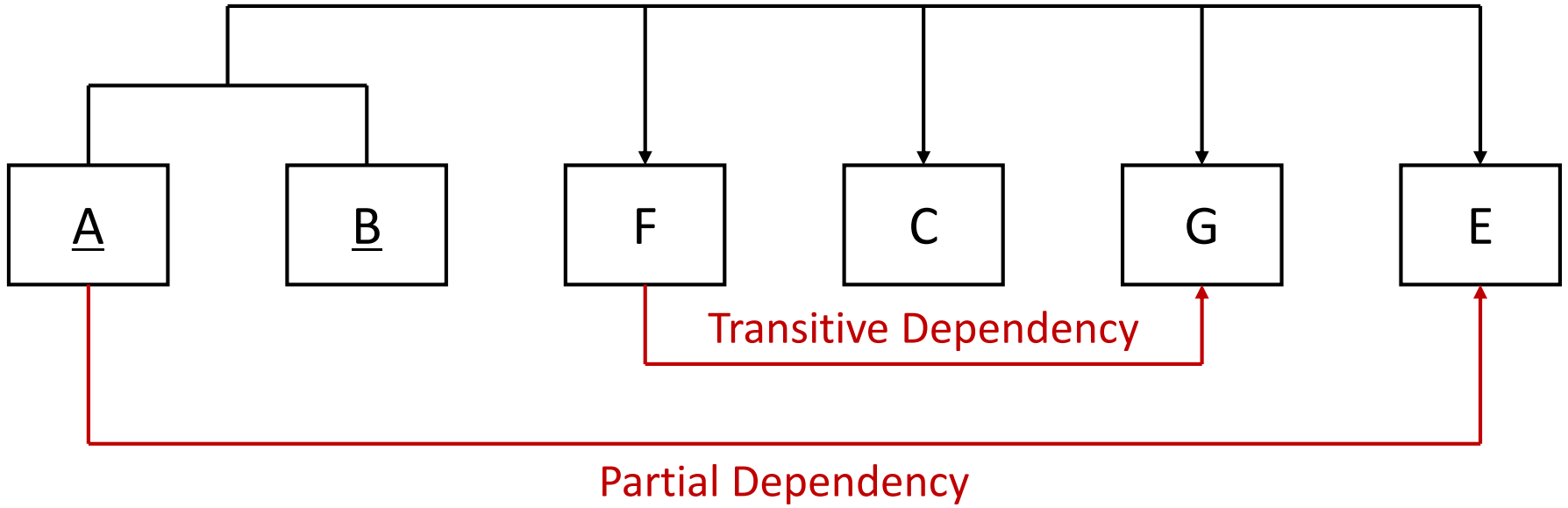
- A relation schema R is in ***second normal form (2NF)*** if there is ***no partial dependency*** from any (composite) key of R to any non-key attribute in R

3NF & Transitive Dependency

- A functional dependency $X \longrightarrow Y$ in a relation schema R is a ***transitive dependency*** if there exists a set of attributes Z in R such that both $X \longrightarrow Z$ and $Z \longrightarrow Y$ hold and that Z is neither a superkey nor a subset of any key of R

- A relation schema R is in ***third normal form (3NF)*** if there is ***no partial dependency or transitive dependency*** from any key of R to any non-key attribute in R

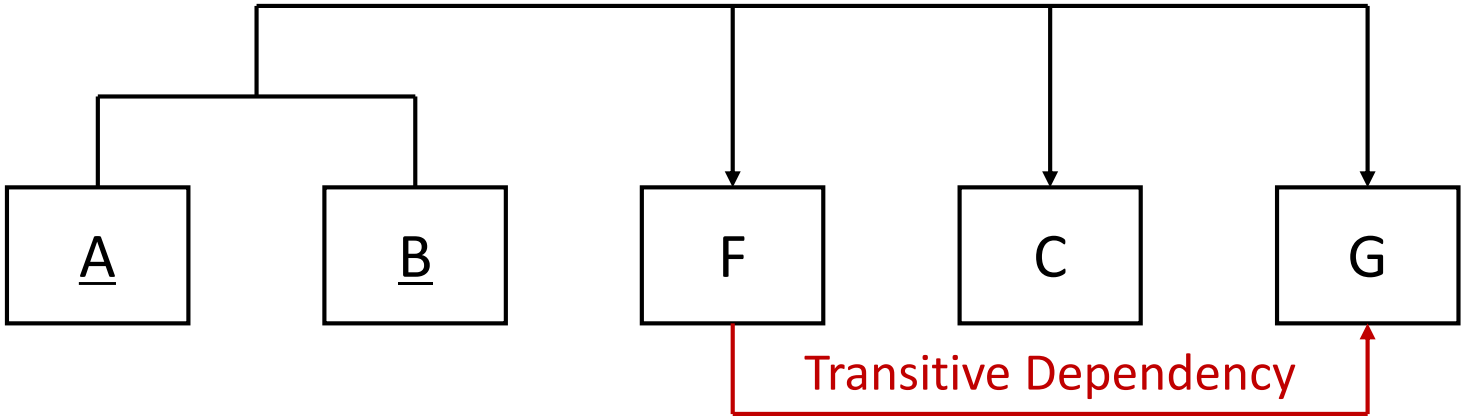
Partial & Transitive Dependency



- {A, B} is the only key
- AB ⟶ E is a partial dependency
- AB ⟶ G is a transitive dependency

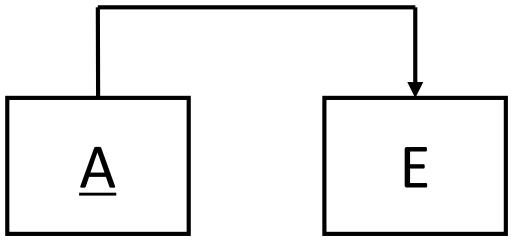
2NF Decomposition

R




```
PRIMARY KEY (A, B)
FOREIGN KEY (A)
    REFERENCES S(A)
```

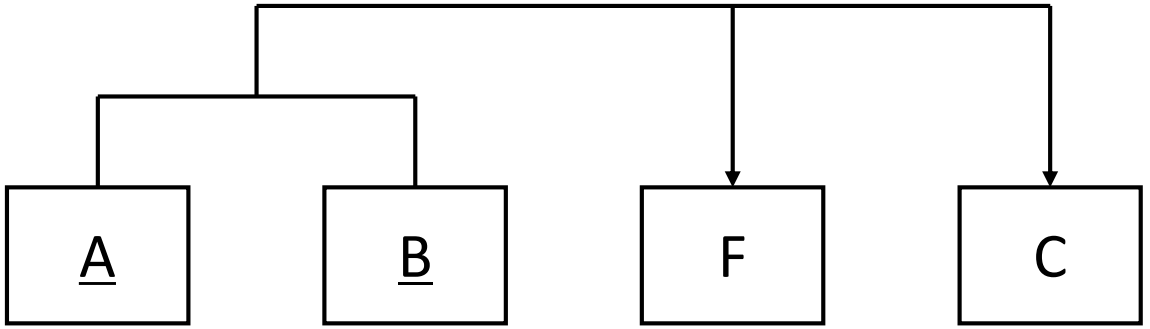
S




```
PRIMARY KEY (A)
```

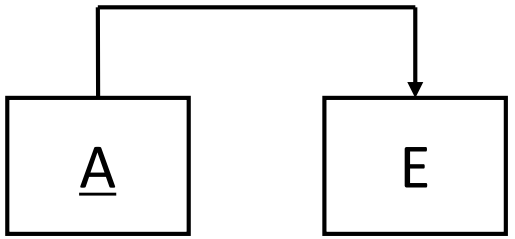
3NF Decomposition

R



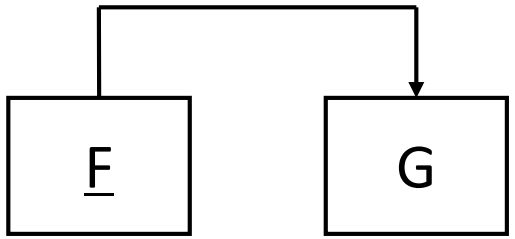

```
PRIMARY KEY (A, B)
FOREIGN KEY (A)
    REFERENCES S(A)
FOREIGN KEY (F)
    REFERENCES T(F)
```

S




```
PRIMARY KEY (A)
```

T




```
PRIMARY KEY (F)
```

3NF & BCNF

- A relation schema R is in ***3NF*** if
  for each *non-trivial* FD: X ⟶ Y
  either X is a superkey or Y is in some key

- A relation schema R is in ***BCNF*** if
  for each *non-trivial* FD: X ⟶ Y
  X is a superkey

- *Non-trivial* means Y is not a subset of X

Integrity Constraints

- Check: single attribute/table, specified inside table definition

```
[CONSTRAINT [symbol]] CHECK (expr)
```

- Assertion: multiple tables

```
CREATE ASSERTION <name> CHECK (
     NOT EXISTS (
          <SELECT-FROM-WHERE>
     )
);
```

Integrity Constraints

- Trigger: event-condition-action

```
CREATE TRIGGER trigger-name
    {BEFORE | AFTER} {INSERT | DELETE | UPDATE [OF column]}
    ON table-name
        [REFERENCING [OLD AS oldTuple]
                     [NEW AS newTuple]]
        [FOR EACH ROW]
        [WHEN (condition)]
        BEGIN
            statement list
        END;
```