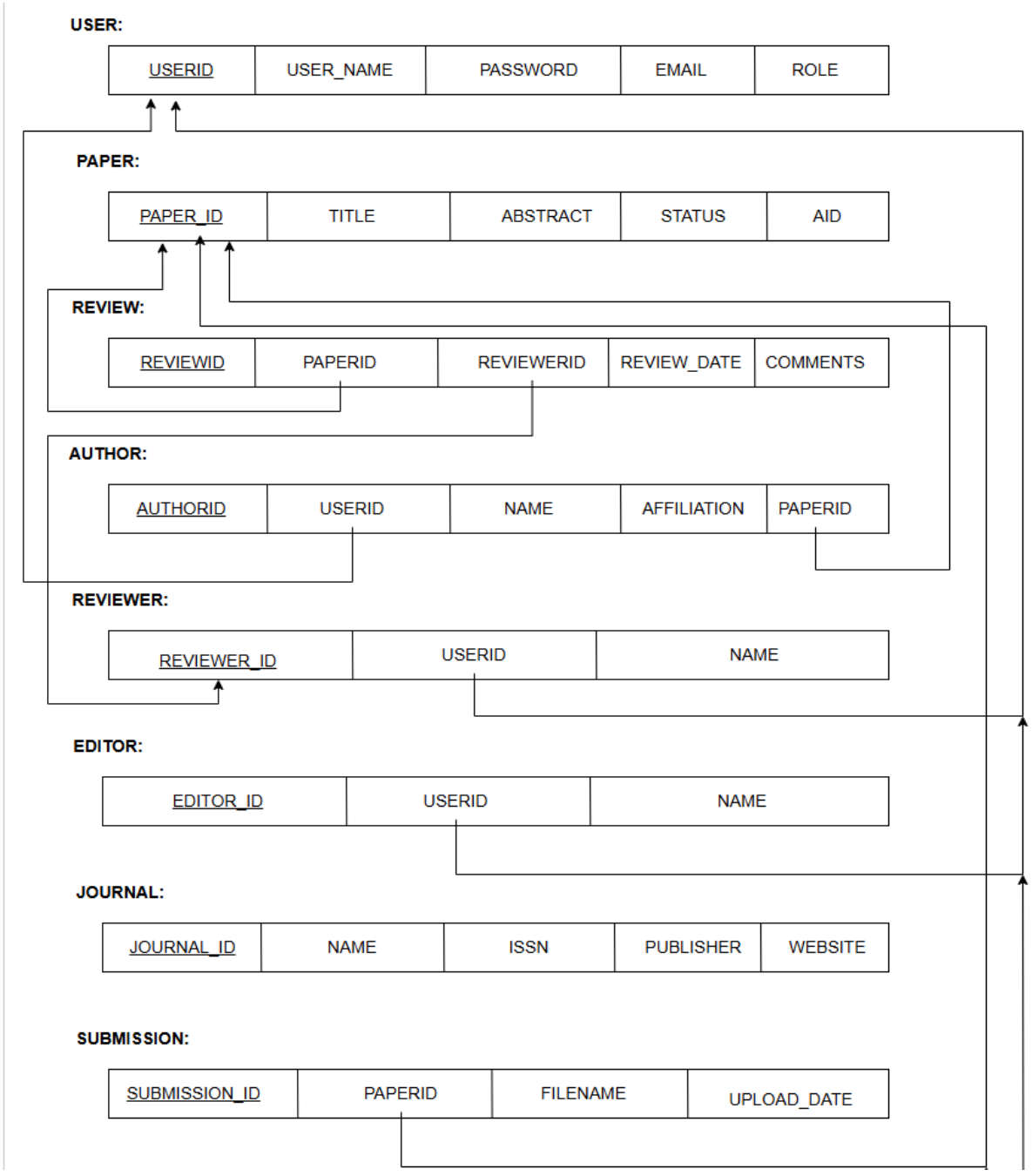
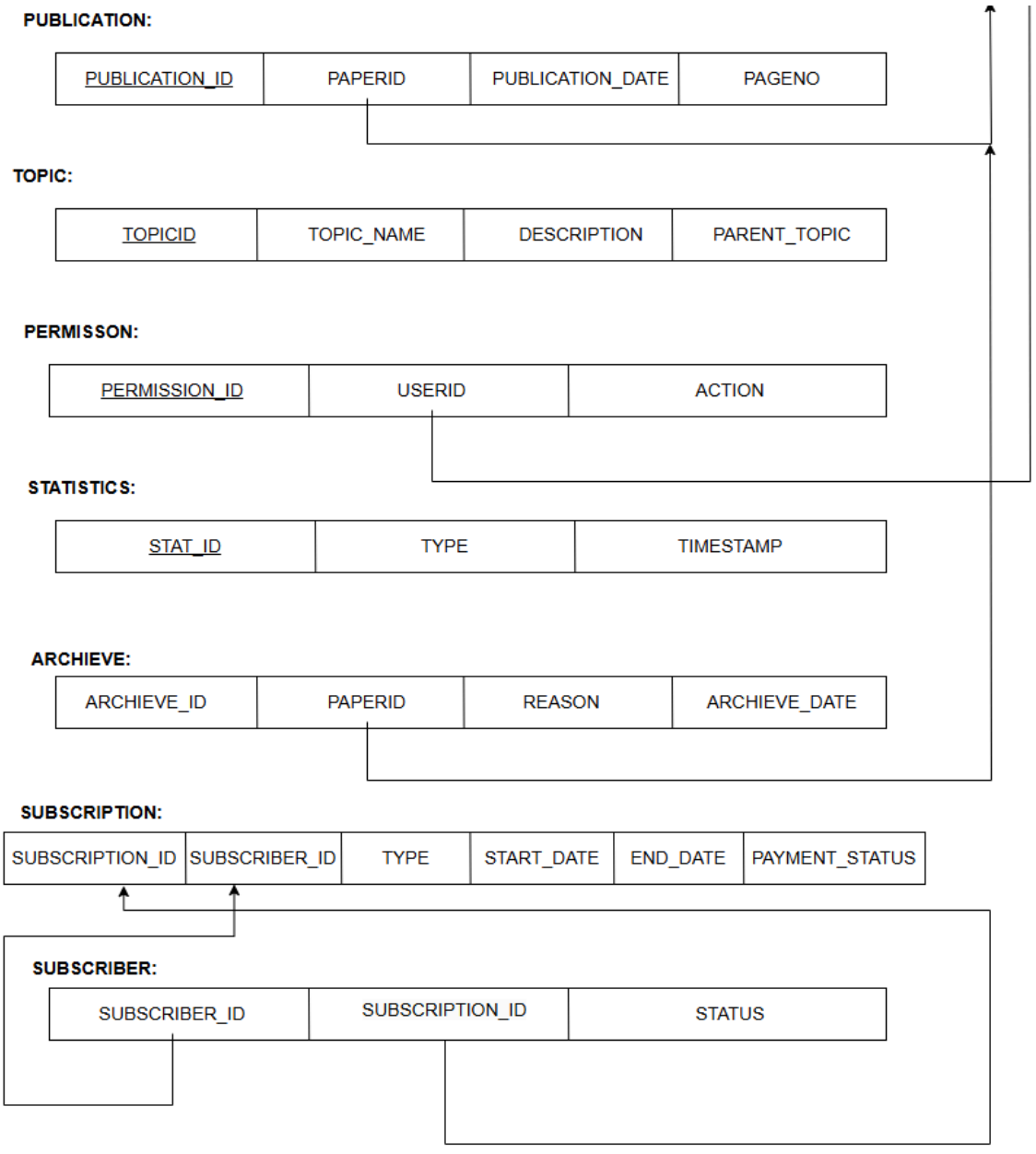
NAME-KEERTHANA.G.S

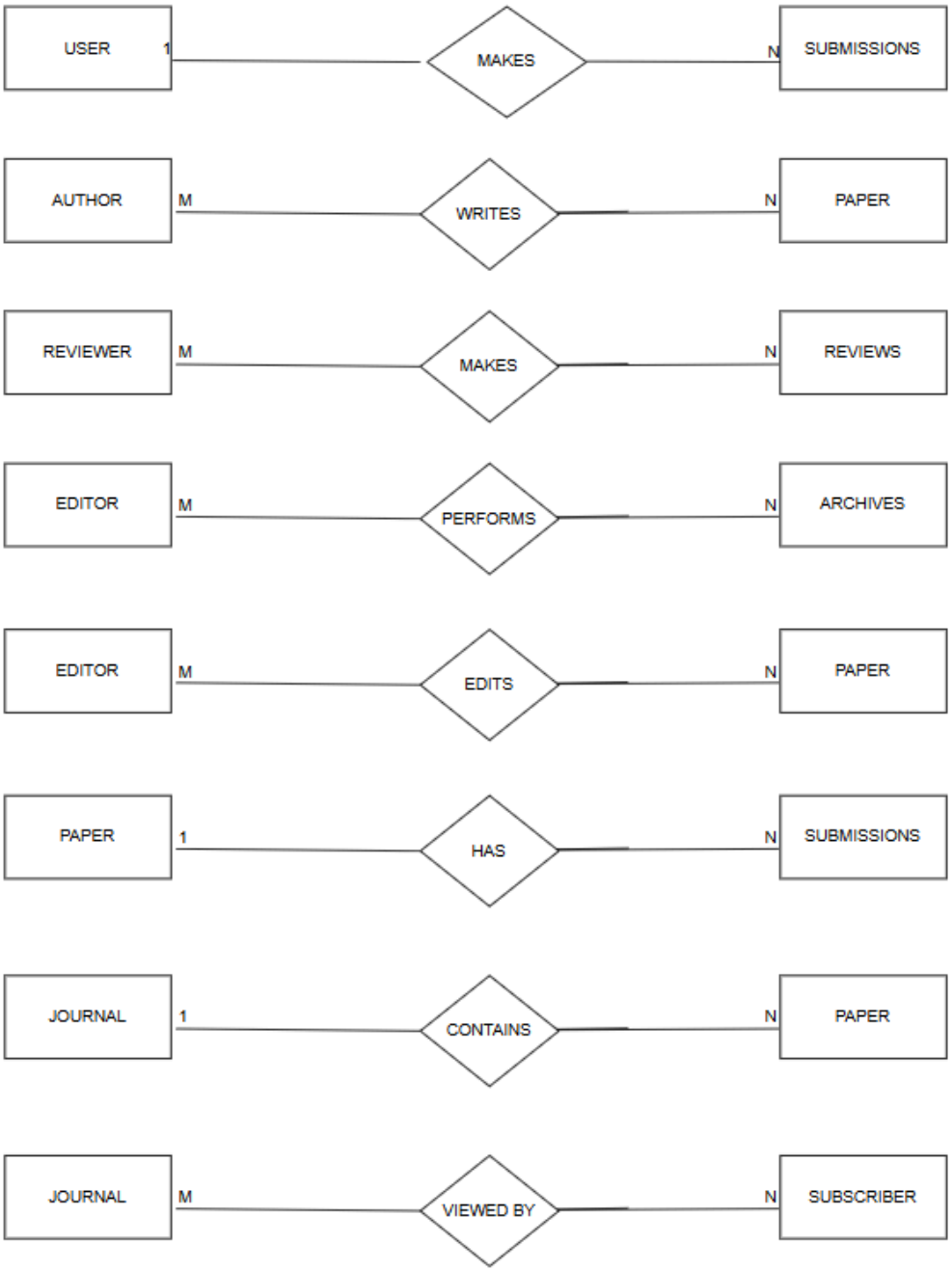
REGISTER NUMBER: 3122225001059

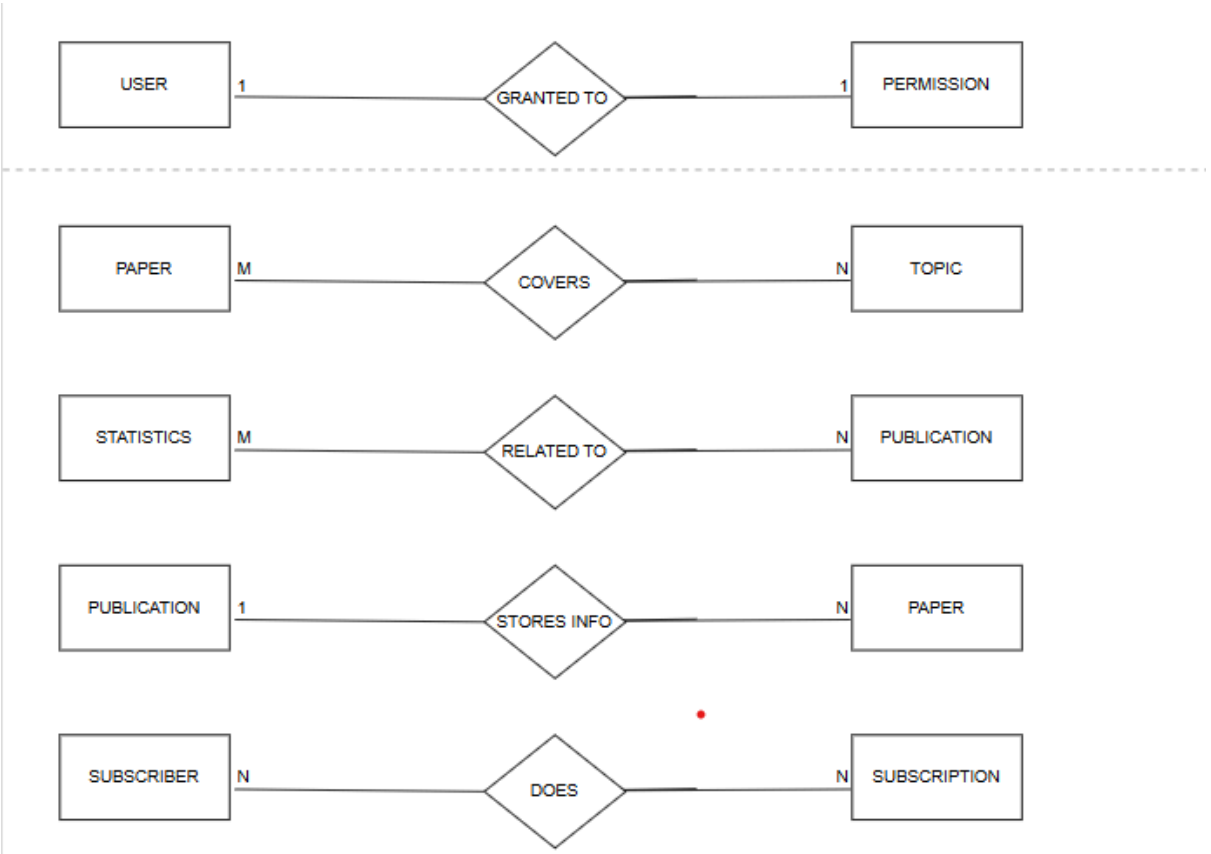
SCHEMA DIAGRAM:




PUBLICATION:
PUBLICATION_ID
PAPERID
PUBLICATION_DATE
PAGENO
TOPIC:
TOPICID
TOPIC_NAME
DESCRIPTION
PARENT_TOPIC
PERMISSON:
PERMISSION_ID
USERID
ACTION
STATISTICS:
STAT_ID
TYPE
TIMESTAMP
ARCHIEVE:
ARCHIEVE_ID
PAPERID
REASON
ARCHIEVE_DATE
SUBSCRIPTION:
SUBSCRIPTION_ID
SUBSCRIBER_ID
TYPE
START_DATE
END_DATE
PAYMENT_STATUS
SUBSCRIBER:
SUBSCRIBER_ID
SUBSCRIPTION_ID
STATUS

CARDINALITY DIAGRAM:





USERS(*userid,username,password,email,role*)

The possible functional depenncies are:

- userid -> username
- userid -> password
- userid -> email
- userid -> role
- username -> userid

- role -> userid
- {userid, username} -> password
- {userid, username} -> email
- {userid, username} -> role
- {userid, password} -> email
- {userid, password} -> role
- {userid, email} -> username

LET

- userid be A
- Username be B
- Password be C
- Email be D
- Role be E

Minimal set of fd's:

Complete set :

A->B,A->C,A->D,A->E,D->A,AB->C,AB->D,AB->E,AC->D ,AC->E,AD->B

Finding Minimal Set:

1.A->B

INCLUDE A+={A,B,C,D,E}

W/O INCLUDE A+={A,C,B,D,E}

2.A->C

INCLUDE A+={A,C,D,E,B}

WITHOUT INCLUDE A+={A,D,E,B,C}

3.A->E

INCLUDE A+={A,E,D,B,C}

WITHOUT INCLUDE A+={A,D,B,C,E}

4.D->A

INCLUDE A+={D,A,B,C,E}

WITHOUT INCLUDE A+={D}

5.AB->C

INCLUDE AB+={A,B,C,D,E}

W/O AB+={A,B,E,D}

6.AB->D

INCLUDE AB+={A,B,D,C,E}

W/O AB+={A,B,C,D,E}

7.AB->E

INCLUDE AB+={A,B,E,C,D}

W/O INCLUDE AB+={A,B,C,DE,}

8.AC->D

INCLUDE AC+={A,B,C,D,E}

W/O AC+={A,B,C,D,E}

9.AC->E

INCLUDE AC+={A,B,C,D,E}

W/O AC+={A,C,D,B}

10.AD->B

INCLUDE AB+={A,D,B,C,E}

W/O AB+={A,D}

SO THE MINIMAL SETS ARE

- A->D
- D->A
- AB->C
- AC->E
- AD->B

PAPERS(*paperid , title, abstract , status , authored***)**

The possible functional dependencies are :

- paperid -> title
- paperid -> abstract
- paperid -> status
- paperid -> authored
- {paperid, title} -> abstract
- {paperid, title} -> status
- {paperid, title} -> authorid
- {paperid, authorid} -> status

Let

- paperid be A
- title be B
- abstract be C
- status be D
- authorid be E

complete set :

- A->B
- A->C

- A->D
- A->E
- AB->C
- AB->D
- AB->E
- AE->D

Finding minimal set of fds:

1.A->B

INCLUDE A+={A,B,C,,D,E}

W/O A+={A,C,D,E}

2.A->C

INCLUDE A+={A,C,B,D,E}

W/O A+={A,B,D,E,C}

3.A->D

INCLUDE A+={A,B,D,E,C}

W/O A+={A,B,E,C,D}

4.A->E

INCLUDE A+={A,B,E,C,D}

W/O A+={A,B,C,D,E}

5.AB->C

INCLUDE AB+={A,B,C,D,E}

W/O AB+={A,B,D,E}

6.AB->D

INCLUDE AB+={A,B,C, D,E}

W/O AB+={A,B,C,D,E}

7.AB->E

INCLUDE AB+={A,B,C,E,D}

W/O AB+={A,B,C}

8.AE->D

INCLUDE AE+={A,B,D,C,E}

W/O AE+={A,E,B,C}

THE MINIMAL SET OF FDS ARE

- A->B
- AB->C
- AB->E
- AE->D

REVIEWS(*reviewid , paperid , reviwerid , reviewdate , comments***)**

The possible functional depenncies are:

- reviewid -> paperid
- reviewid -> reviewerid
- reviewid -> reviewdate
- reviewid -> comments

- paperid -> reviewid
- comments -> reviewid
- {reviewid, paperid} -> reviewerid
- {reviewid, paperid} -> reviewdate
- {reviewid, paperid} -> comments

Let

Ø reviewid be A

Ø paperid be B

Ø reviwerid be C

Ø reviewdate be D

Ø comments be E

THE COMPLETE SET :

- A->B
- A->C
- A->D
- A->E
- E->A
- AB->C
- AB->D
- AD->E

FINDING MINIMAL SET OF FDS:

1.A->B

INCLUDE A+={A,B,C,D,E}

W/O A+={A,C,D,E}

2.A->C

INCLUDE A+={A,B,C,E,D}

W/O A+={A,B,D,E,C}

3.A->D

INCLUDE A+={A,B,D,E,C}

W/O A+={A,B,E,D,C}

4.A->E

INCLUDE A+={A,B,E,C,D}

W/O A+={A,B,C,D,E}

5.E->A

INCLUDE E+={E,A,B,C,D}

W/O E+={E}

6.AB->C

INCLUDE AB+={A,B,C,D,E}

W/O AB+={A,B,D,E}

7.AB->D

INCLUDE AB+={A,B,C,,D,E}

W/O AB+={A,B,C}

8.AD->E

INCLUDE AD+={A,B,C,D,E}

W/O AD+={A,B,C,D}

THE MINIMAL SET OF FD'S ARE

- A->B
- E->A
- AB->C
- AB->D
- AD->E

AUTHORS(*authorid,userid,name,affiliation,paperid*)

The possible functional depenncies are:

· authid -> userid

· authid -> name

· authid -> affiliation

· authid -> paperid

· userid -> authid

· name -> authid

· paperid -> authid

· {authid, userid} -> name

· {authid, userid} -> affiliation

· {authid, userid} -> paperid

· {authid, name} -> userid

Let

- authorid be A
- userid be B
- name be C
- affiliation be D
- paperid be E

COMPLETE SET:

- A->B
- A->E
- B->A
- E->C

- AB->D

1.A->B

INCLUDE A+={A,B,C,D,E}

W/O A+={A,E,C}

2.A->E

INCLUDE A+={A,B,E,C,D}

W/O A+={A,B,D,C,E}

3.A->D

INCLUDE A+={A,B,C,D,E}

W/O A+={A,B,D}

4.E->C

INCLUDE E+={E,C}
W/O E+={E}

5.AB->D

INCLUDE AB+={A,B,C,D,E}

W/O AB+={A,B,E,C}

THE MINIMAL SET OF FDS ARE

- A->B
- A->E

- B->A
- E->C
- AB->D

REVIEWERS *(reviewerid,userid,name)*

The possible functional depenncies are:

- reviewerid -> userid
- reviewerid -> name
- userid -> reviewerid

let

- reviewerid be A
- userid be B
- name be C

THE COMPLETE SET:

- A->B
- A->C
- B->A

THE MINIMAL SET OF FDS

1.A->B

INCLUDE A+={A,B,C}

W/O A+={A,C}

2.A->C

INCLUDE A+={A,C,B}

W/O A+={A,B}

3.B->A

INCLUDE A+={B,A,C}

W/O B+={B}

THE MINIMAL SET OF FDS

- A->B
- A->C
- B->A

EDITORS*(editorid, userid, name)*

The possible functional dependencies are:

- editorid -> userid
- editorid ->name
- userid -> editorid
- userid -> name
- editorid, userid -> name
- editorid, name -> userid
- userid, name -> editorid

LET

➢ editorid be A

➢ userid be B

➢ name be C

Complete set of fd's:

A->B, A->C, B->A, B->C, AB->C, AC->B, BC->A

Finding Minimal Set:

1. A->B

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {A,C,B}

2. A->C

INCLUDE A+= {A,C,B}

W/O INCLUDE A+= {A}

3. B->A

INCLUDE B+= {B,A,C}

W/O INCLUDE B+= {B,C,A}

4. B->C

INCLUDE B+= {B,C,A}

W/O INCLUDE B+= {B}

5. AB->C

INCLUDE AB+= {A,B,C}

W/O INCLUDE AB+= {A,B,C}

6. AC->B

INCLUDE AC+= {A,C,B}

W/O INCLUDE AC+= {A,C}

7. BC->A

INCLUDE BC+= {B,C,A}

W/O INCLUDE BC+= {B,C}

SO THE MINIMAL SETS ARE

- A->C
- B->C
- AC->B
- BC->A

JOURNAL(*journalid, name, ISSN, publisher, website*)

The possible functional dependencies are:

- journalid->name
- journalid->ISSN
- journalid->publisher
- journalid->website
- ISSN->journalid
- publisher->journalid
- journalid,publisher ->website

- journalid,ISSN->name

LET

➢ journalid be A

➢ name be B

➢ ISSN be C

➢ publisher be D

➢ website be E

Complete set of fd's:

A->B, A->C, A->D, A->E, C->A, D->A, AD->E, AC->B

1.A->B

INCLUDE A+= {A,B,C,D,E}

W/O INCLUDE A+= {A,C,D,E,B}

2.A->C

INCLUDE A+= {A,C,D,E,B}

W/O INCLUDE A+= {A,D,E,}

3.A->D

INCLUDE A+= {A,D,C,E,B}

W/O INCLUDE A+= {A,C,E,B}

4.A->E

INCLUDE A+= {A,E,C,D,B}

W/O INCLUDE A+= {A,C,D,B,E}

5.C->A

INCLUDE C+= {C,A,D,E,B}

W/O INCLUDE C+= {C}

6.D->A

INCLUDE D+= {D,A,C,B,E}

W/O INCLUDE D+= {D}

7.AD->E

INCLUDE AD+= {A,D,E,B,C}

W/O INCLUDE AD+= {A,D,C,B}

8.AC->B

INCLUDE AC+= {A,C,B,D,E}

W/O INCLUDE AC+= {A,C,D,E}

SO THE MINIMAL SETS ARE

- A->C
- A->D
- C->A
- D->A
- AD->E
- AC->B

SUBMISSION(*subid,paperid,fiename,upload_date*):

The possible functional dependencies are:

- subid->paperid
- subid->filename
- subid->upload_date
- subid,paperid->filename
- subid,paperid->upload_date
- subid,filename->paperid
- subid,upload_date->paperid

LET

➢ subid be A

➢ paperid be B

➢ filename be C

➢ upload_date be D

Complete set of fd's:

A->B, A->C, A->D, AB->C, AB->D, AC->B, AD->B

1.A->B

INCLUDE A+= {A,B,C,D}

W/O INCLUDE A+= {A,C,D,B}

2.A->C

INCLUDE A+= {A,C,B,D}

W/O INCLUDE A+= {A,D,B,C}

3.A->D

INCLUDE A+= {A,D,B,C}

W/O INCLUDE A+= {A}

4.AB->C

INCLUDE AB+= {A,B,C,D}

W/O INCLUDE AB+= {A,B,D}

5.AB->D

INCLUDE AB+= {A,B,D,C}

W/O INCLUDE AB+= {A,B,C,D}

6.AC->B

INCLUDE AC+= {A,C,B,D}

W/O INCLUDE AC+= {A,C,D,B}

7.AD->B

INCLUDE AD+= {A,D,C,B}

W/O INCLUDE AD+= {A,D}

SO THE MINIMAL SETS ARE

- A->D
- AB->C
- AD->B

PUBLICATION(*publicationid,paperid,date,pageno*):

The possible functional dependencies are:

- publicationid->paperid
- publicationid->date
- publicationid->pageno
- publicationid,date->pageno
- publicationid,pageno->paperid
- publicationid,paperid->pageno

LET

➢ subid be A

➢ paperid be B

➢ filename be C

➢ upload_date be D

Complete set of fd's:

A->B, A->C, A->D, AC->D, AD->C, AB->D

1.A->B

INCLUDE A+= {A,B,C,D}

W/O INCLUDE A+= {A,C,D}

2.A->C

INCLUDE A+= {A,C,D,B}

W/O INCLUDE A+= {A,D,C,B}

3.A->D

INCLUDE A+= {A,D,B,C}

W/O INCLUDE A+= {A,B,D,C}

4.AC->D

INCLUDE AC+= {A,C,D,B}

W/O INCLUDE AC+={A,C,B,D}

5.AD->C

INCLUDE AD+= {A,D,C,B}

W/O INCLUDE AD+= {A,D,B}

6.AB->D

INCLUDE AB+= {A,B,D,C}

W/O INCLUDE AB+= {A,B}

SO THE MINIMAL SETS ARE

- A->B
- AD->C
- AB->D

TOPICS(*topicid,topicname,description,parenttopic*):

The possible functional dependencies are:

- topicid->topicname
- topicid->description
- topicid->parenttopic
- topicid,parenttopic->topicname
- topicname->topicid
- description,parenttopic->topicname
- topicname,description->parenttopic

LET

➢ topicid be A

➢ topicname be B

➢ description be C

➢ parenttopic be D

Complete set of fd's:

A->B, A->C, A->D, AD->B, B->A, CD->B, BC->D

1.A->B

INCLUDE A+= {A,B,C,D}

W/O INCLUDE A+= {A,C,D,B}

2.A->C

INCLUDE A+= {A,C,D,B}

W/O INCLUDE A+= {A,D,B}

3.A->D

INCLUDE A+= {A,D,C,B}

W/O INCLUDE A+= {A,C}

4.AD->B

INCLUDE AD+= {A,D,B,C}

W/O INCLUDE AD+= {A,D,C,B}

5.B->A

INCLUDE B+= {B,A,C,D}

W/O INCLUDE B+= {B}

6.CD->B

INCLUDE CD+= {C,D,B,A}

W/O INCLUDE CD+= {C,D}

7.BC->D

INCLUDE BC+= {B,C,D,A}

W/O INCLUDE BC+= {B,C,A,D}

SO THE MINIMAL SETS ARE

- A->C
- A->D
- B->A
- CD->B
-

$$$1111111111111111111111111

PERMISSIONS_(permission_id, user_id, action)_:

The possible functional dependencies are:

- permission_id->user_id
- permission_id->action
- user_id,action->permission_id
- user_id->action

LET

➢ permission_id be A

➢ user_id be B

➢ action be C

Complete set of fd's:

1.A->B

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {A,C}

2.A->C

INCLUDE A+= {A, B, C}

W/O INCLUDE A+= {A, B, C}

3.BC->A

INCLUDE BC+= {A,B,C}

W/O INCLUDE A+= {B,C}

4.B->C

INCLUDE AD+= {A,B,C}

W/O INCLUDE AD+= {B}

SO THE MINIMAL SETS ARE

- A->B
- BC->A
- B->C

$$2222222222

SUBSCRIBER*(subscriber_id, subcription_id, status)*

The possible functional dependencies are:

- subscriber_id->subscription_id
- subscriber_id->status
- subscription_id,status->subscriber_id
- subscriber_id,status->subscription_id

LET

➢subscriber_id be A

➢ subscription_id be B

➢ status be C

Complete set of fd's:

1.A->B

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {A,B,C}

2.A->C

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {A}

3.BC->A

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {B,C}

4.AC->B

INCLUDE AD+= {A,C,B}

W/O INCLUDE AD+= {A,C}

SO THE MINIMAL SETS ARE

- A->C
- BC->A
- AC->B

$$3333333333

SUBCRIPTION*(subscription_id, subscriber_id, subcription_type, start_date, end_date, payment_status)*

The possible functional dependencies are:

- subscription_id->subscriber_id
- subscription_id->subscription_type
- Subscription_id->start_date
- Subscription_id->end_date
- subscription_id->payment_status
- subscription_id, subscription_type->start_date, end_date
- subscription_id, subscription_type->payment_status

LET

➢ subscription_id be A

➢ subscriber_id be B

➢ subscription_type be C

➢start_date be D

➢end_date be E

➢payment_status be F

Complete set of fd's:

1.A->B

INCLUDE A+= {A,B,C,D,E,F}

W/O INCLUDE A+= {A,C,D,E,F}

2.A->C

INCLUDE A+= {A,B,C,D,E,F}

W/O INCLUDE A+= {A,D,B}

3.A->D

INCLUDE A+= {A,B,C,D,E,F}

W/O INCLUDE A+= {A,B,C,D,E,F}

4.A->E

INCLUDE A+= {A,B,C,D,E,F}

W/O INCLUDE A+= {A,B,C,D,E,F}

5.A->F

INCLUDE A+= {A,B,C,D,E,F}

W/O INCLUDE A+= {A,B,C,D,E,F}

6.AC->DE

INCLUDE AC+= {A,B,C,D,E,F}

W/O INCLUDE AC+= {A,B,C}

7.AC->F

INCLUDE AC+= {A,B,C,D,E,F}

W/O INCLUDE AC+= {A,B,C,D,E}

SO THE MINIMAL SETS ARE

- A->B
- A->C
- AC->DE
- AC->F

4444444444444444444444444444444444444444444444444

STATISTICS*(statistic_id, type, timestamp)*

The possible functional dependencies are:

- statistic_id->type
- statistic_id->timestamp
- type,timestamp->statistic_id

LET

➢ statistic_id be A

➢ type be B

➢ timestamp be C

Complete set of fd's:

1.A->B

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {A,C}

2.A->C

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {A,B}

3.BC->A

INCLUDE A+= {A,B,C}

W/O INCLUDE A+= {B,C}

SO THE MINIMAL SETS ARE

- A->B
- A->C
- BC->A

55555555555555555555555555555555555555

ARCHIVES*(archive_id, paper_id, reason, archive_date)*

The possible functional dependencies are:

- archive_id->paper_id
- archive_id->reason
- archive_id->archive_date
- Paper_id, archive_date->reason
- Paper_id, reason->arch_date

LET

➢ archive_id be A

➢ paper_id be B

➢ reason be C

➢ archive_date be D

Complete set of fd's:

1.A->B

INCLUDE A+= {A,B,C,D}

W/O INCLUDE A+= {A,B,D}

2.A->C

INCLUDE A+= {A,C,D,B}

W/O INCLUDE A+= {A,D,B}

3.A->D

INCLUDE A+= {A,D,C,B}

W/O INCLUDE A+= {A,B,C}

4.BD->C

INCLUDE BD+= {D,B,C}

W/O INCLUDE BD+= {B,D}

5.BC->D

INCLUDE BC+= {B,C,D}

W/O INCLUDE BC+= {B,C}

SO THE MINIMAL SETS ARE

- A->B
- A->C
- BD->C
- BC->D

Finding the Candidate Keys:

1. **USERS** *(userid, username, password, email, role)*
    - Minimal Set: A -> D, D -> A, AB -> C, AC -> E, AD -> B
    - **Candidate Keys:** A,D
    - **Prime Attributes:** A, B, D
    - **Non-Prime Attributes:** C, E

2. **PAPERS** *(paperid, title, abstract, status, authored)*
    - Minimal Set: A -> B, A->E, B->C, AB->D
    - **Candidate Keys:** A
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** B,C, E, D

3. **REVIEWS** *(reviewid, paperid, reviewerid, reviewdate, comments)*
    - Minimal Set: A -> B, E -> A, AB -> C, AB -> D, AD -> E
    - **Candidate Keys:** A, B
    - **Prime Attributes:** A, B
    - **Non-Prime Attributes:** C, E, D

4. **AUTHORS** *(authorid, userid, name, affiliation, paperid)*
    - Minimal Set: A -> B, A->E,B->A,E->C,AB->D
    - **Candidate Keys:** A, B
    - **Prime Attributes:** A, B
    - **Non-Prime Attributes:** C, E, D

5. **REVIEWERS** *(reviewerid, userid, name)*
    - Minimal Set: A -> B, A -> C, B -> A
    - **Candidate Keys:** A, B
    - **Prime Attributes:** A, B
    - **Non-Prime Attributes:** C

6. **EDITORS** *(editorid, userid, name)*
    - Minimal Set: A -> C, B -> C, AC -> B, BC -> A
    - **Candidate Keys:** A, B
    - **Prime Attributes:** A, B
    - **Non-Prime Attributes:** C

7. **JOURNAL** *(journalid, name, ISSN, publisher, website)*
    - Minimal Set: A -> C, A -> D, C -> A, D -> A, AD -> E, AC -> B
    - **Candidate Keys:** A, C, D
    - **Prime Attributes:** A, C, D
    - **Non-Prime Attributes:** B

8. **SUBMISSION** *(subid, paperid, filename, upload_date)*
    - Minimal Set: A -> D, AB -> C, AD -> B
    - **Candidate Keys:** A
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** B, C, D

9. **PUBLICATION** *(publicationid, paperid, date, pageno)*
    - Minimal Set: A -> B, AD -> C, AB -> D
    - **Candidate Keys:** A
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** B, C, D

10. **TOPICS** *(topicid, topicname, description, parenttopic)*
    - Minimal Set: A -> C, A -> D, B -> A, CD -> B
    - **Candidate Keys:** A, B, CD

- **Prime Attributes:** A, B, C, D
- **Non-Prime Attributes:** None

11. **PERMISSIONS***(permission_id, user_id, action)*
    - Minimal Set:
    - **Candidate Keys:** A
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** B,C
12. **SUBSCRIBER***(subscriber_id, subcription_id, status)*
    - Minimal Set:
    - **Candidate Keys:**A, BC
    - **Prime Attributes:** A, B, C
    - **Non-Prime Attributes:**None
13. **SUBCRIPTION***(subscription_id, subscriber_id, subcription_type, start_date, end_date, payment_status)*
    - Minimal Set: A->B,A->C,AC->DE,AC->F
    - **Candidate Keys:** A
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** B.C,D,E,F
14. **STATISTICS***(statistic_id, type, timestamp)*
    - Minimal Set: A->B,A->C,BC->A
    - **Candidate Keys:**A
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** None
15. **ARCHIVES***(archive_id, paper_id, reason, archive_date)*
    - Minimal Set: A->B,A->C,BD->C,BC->D
    - **Candidate Keys:** A, BD, BC
    - **Prime Attributes:** A
    - **Non-Prime Attributes:** B,C,D

NORMALIZATION:

USERS:

- **Attributes**: {A,B,C,D,E}
- **Functional Dependencies (FDs)**:
    - Minimal Set: A -> D, D -> A, AB -> C, AC -> E, AD -> B

First Normal Form (1NF)

1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes the table is in 1NF.

Second Normal Form (2NF)

2NF requires that the table is in 1NF and that all non-prime attributes are fully functionally dependent on the entire candidate key.

CHECKING 2NF:

For the candidate key A:

- A->D
- A->B

Fro the candidate key D:

- D->A
- D->B

All the non prime attributes Cand E are functionally dependent on the combinations of prime attributes:

- C depends on AB

- E depends on AC

Since non-prime attributes C and E are dependent on combinations of prime attributes and not just a part of a candidate key, **the table is in 2NF.**

Third Normal Form (3NF)

3NF requires that the table is in 2NF and that all attributes are non-transitively dependent on the candidate key.

Checking 3NF:

- A→D (A is a candidate key, so it's already in 3NF)
- D→ (D is a candidate key, so it's already in 3NF)
- AB→C (AB is a superkey, as it includes the candidate key A)
- AC→ E (AC is a superkey, as it includes the candidate key A)
- AD→ (AD is a superkey, as it includes the candidate key A)

Since all non-prime attributes are directly dependent on a candidate key or a combination of prime attributes without any transitive dependencies through other non-prime attributes, **the table is in 3NF.**

Boyce-Codd Normal Form (BCNF)

BCNF is a stricter form of 3NF where for any non-trivial functional dependency X→ Y, X must be a superkey.

Checking BCNF:

- A→ D (A is a candidate key, so it's a superkey)
- D→ A (D is a candidate key, so it's a superkey)

· AB→ C (AB is a superkey, as it includes the candidate key A)

· AC→ E (AC is a superkey, as it includes the candidate key A)

· AD→ B (AD is a superkey, as it includes the candidate key A)

Since every determinant in each functional dependency is a superkey, **the table is in BCNF.**

The USERS table is in BCNF as it satisfies all conditions for 1NF, 2NF, 3NF, and BCNF.

PAPERS:

· **Attributes**: {A,B,C,D,E}\{A, B, C, D, E\}{A,B,C,D,E}

· **Functional Dependencies (FDs)**:

- A→ B
- A→ E
- B→ C
- AB→D

First Normal Form (1NF)

1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes the table is in 1NF.

Second Normal Form (2NF)

2NF requires that the table is in 1NF and that all non-prime attributes are fully functionally dependent on the entire candidate key.

For he candidate keys

A->B

A->E

AB->D

The A is the candidate key

For the candidate key:

B->C ,B is no the partial subset of the candidate key.

As the FDS'S contain only onoe candidate key the table is in 2NF.

Third Normal Form (3NF)

3NF requires that the table is in 2NF and that all attributes are non-transitively dependent on the candidate key.

- A→B (B is directly dependent on A, so it's OK for 3NF)
- A→ E (E is directly dependent on A, so it's OK for 3NF)
- B→ C (C is transitively dependent on A through B, which violates 3NF)
- AB→ D (D is directly dependent on the composite key AB, so it's OK for 3NF)

To achieve 3NF, we need to decompose the table to remove the transitive dependency B→ C.

DECOMPOSITION:

A	B	C	D	E

R1:

B	C

R2:

A	D	E	B

Boyce-Codd Normal Form (BCNF)

BCNF is a stricter form of 3NF where for any non-trivial functional dependency X→ Y, X must be a superkey.

Checking BCNF:

Table 1 (Papers):

- A→B (A is a superkey)
- A→ E (A is a superkey)
- AB->D (A is a superkey)

Table 2 (B_C Relationship):

- B→ C (B is a superkey in this table)

Since every determinant in each functional dependency is a superkey, the decomposed **tables are in BCNF.**

AUTHORS:

· **Attributes**: {authorid, userid, name, affiliation, paperid}

· **Functional Dependencies (FDs)**:

- A→ B
- A→ E
- B→ A
- E→ C
- AB→ D

First Normal Form (1NF)

The table seems to be in 1NF already as there is no mention of multi-valued attributes.

Second Normal Form (2NF)

2NF requires that the table is in 1NF and that all non-prime attributes are fully functionally dependent on the entire candidate key.

Checking 2NF:

All non-prime attributes {C, E, D} are not fully functionally dependent on any partial key; they are fully dependent on the entire candidate keys {A, B}. Hence, **the table is in 2NF.**

Third Normal Form (3NF)

3NF requires that the table is in 2NF and that all attributes are non-transitively dependent on the candidate key.

DECOMPOSITION:

A	**B**	**C**	**D**	**E**

R1:

E	C

R2:

A	B	D	E

Boyce-Codd Normal Form (BCNF)

BCNF is a stricter form of 3NF where for any non-trivial functional dependency X→ Y, X must be a superkey.

Table 1 (AUTHORS):

- A→B (A is a superkey)
- B->A(B is a superkey)
- A→ E (A is a superkey)
- AB->D (A is a superkey)

Table 2 (E_C Relationship):

- E→ C (B is a superkey in this table)

Since every determinant in each functional dependency is a superkey, the decomposed **tables are in BCNF.**

REVIEWS:

· **Attributes**: {reviewid, paperid, reviewerid, reviewdate, comments}

· **Functional Dependencies (FDs)**:

- A→ B
- E→ A
- AB→ C
- AB→ D
- AD→ E

First Normal Form (1NF)

· 1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes the table is in 1NF.

Second Normal Form (2NF)

· 2NF requires that the table is in 1NF and that all non-prime attributes are fully functionally dependent on the entire candidate key.

· Since non-prime attributes C ,D and E are dependent on combinations of prime attributes and not just a part of a candidate key, **the table is in 2NF.**

Third Normal Form (3NF)

3NF requires that the table is in 2NF and that all attributes are non-transitively dependent on the candidate key.

Checking 3NF:

· A→B (A is a candidate key, so it's already in 3NF)

· E->A(D is a candidate key, so it's already in 3NF)

· AB→C (AB is a superkey, as it includes the candidate key A)

· AB→ D (AB is a superkey, as it includes the candidate key A)

· AD→ E(AD is a superkey, as it includes the candidate key A)

Since all non-prime attributes are directly dependent on a candidate key or a combination of prime attributes without any transitive dependencies through other non-prime attributes, **the table is in 3NF.**

· **Boyce-Codd Normal Form (BCNF)**

· BCNF is a stricter form of 3NF where for any non-trivial functional dependency X→ Y, X must be a superkey.

· Since every determinant in each functional dependency is a superkey, the decomposed **tables are in BCNF.**

REVIEWERS

· **Attributes**: {reviewerid, userid, name}

· **Functional Dependencies (FDs)**:

- A→ B
- A→ C

- B→ A

First Normal Form (1NF)

The table seems to be in 1NF already as there is no mention of multi-valued attributes.

Second Normal Form (2NF)

· 2NF requires that the table is in 1NF and that all non-prime attributes are fully functionally dependent on the entire candidate key.

· All non-prime attributes {C} are fully functionally dependent on the entire candidate keys {A, B}. Hence, the table is in 2NF.

Third Normal Form (3NF)

· 3NF requires that the table is in 2NF and that all attributes are non-transitively dependent on the candidate key.

· Since all non-prime attributes are directly dependent on a candidate key or a combination of prime attributes without any transitive dependencies through other non-prime attributes, **the table is in 3NF**

Boyce-Codd Normal Form (BCNF)

· BCNF is a stricter form of 3NF where for any non-trivial functional dependency X→ Y, X must be a superkey.

· Since every determinant in each functional dependency is a superkey, the decomposed **tables are in BCNF.**

EDITORS:

- **Attributes:***(editorid, userid, name)*
- **Minimal set**=A->C, B->C, AC->B, BC->A
- **Candidate Keys:** A, B

First Normal Form (1NF)

1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes **the table is in 1NF.**

Second Normal Form (2NF)

For the candidate key A:

- A→C
- Here, C is dependent on A, but A is a whole candidate key, not a part of one. Hence, this is not a partial dependency.

For the candidate key B:

- B→C
- Here, C is dependent on B, but B is a whole candidate key, not a part of one. Hence, this is not a partial dependency.

So the relation is in 2NF.

Third Normal Form (3NF):

Checking 3NF:

- A->C(A is a candidate key (and therefore a superkey) and C is a non-prime attribute.)
- B->C(B is a candidate key (and therefore a superkey) and C is a non-prime attribute.)
- AC->B(AC is a combination of attributes where A is a candidate key and B is a prime attribute)
- BC->A(BC is a combination of attributes where B is a candidate key and A is a prime attribute)

So the relation is in 3NF.

BCNF:

A→C

- A is a candidate key (and therefore a superkey).
- This dependency satisfies BCNF because A is a superkey.

B→C

- B is a candidate key (and therefore a superkey).
- This dependency satisfies BCNF because B is a superkey.

AC→B

- AC includes A, which is a candidate key (and therefore a superkey).
- This dependency satisfies BCNF because AC is a superkey.

BC→A

- BC includes B, which is a candidate key (and therefore a superkey).
- This dependency satisfies BCNF because BC is a superkey.

So the relation is in BCNF.

JOURNAL

- **Attributes**(journalid, name, ISSN, publisher, website)
- **Minimal Set:** A -> C, A -> D, C -> A, D -> A, AD -> E, AC -> B
- **Candidate Keys:** A, C, D

First Normal Form (1NF)

1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes **the table is in 1NF.**

Second Normal Form (2NF):

For candidate key A:

- A→C
- A→D
- Both C and D are prime attributes, so no partial dependency here.

For candidate key C:

- C→A
- A is a prime attribute, so no partial dependency here.

For candidate key D:

- D→A
- A is a prime attribute, so no partial dependency here.

Composite candidate keys:

- AD→E
    - E is a non-prime attribute and dependent on the full composite key AD, not a part of it, so no partial dependency.
- AC→B
    - B is a non-prime attribute and dependent on the full composite key AC, not a part of it, so no partial dependency.

So relation is in 2NF

Third Normal Form (2NF):

- A→C
    - A is a candidate key (superkey).
    - This dependency satisfies 3NF because A is a superkey.
- A→D
    - A is a candidate key (superkey).
    - This dependency satisfies 3NF because A is a superkey.
- C→A
    - C is a candidate key (superkey).
    - This dependency satisfies 3NF because C is a superkey.
- D→A
    - D is a candidate key (superkey).
    - This dependency satisfies 3NF because D is a superkey.

- AD→E
    - AD is a superkey because it includes `A` and `D`, both candidate keys.
    - This dependency satisfies 3NF because AD is a superkey.
- AC→B
    - AC is a superkey because it includes `A` and `C`, both candidate keys.
    - This dependency satisfies 3NF because AC is a superkey.

So relation is in 3NF

BCNF:

- A→C
    - A is a candidate key (superkey).
    - This dependency satisfies BCNF because A is a superkey.
- A→D
    - A is a candidate key (superkey).
    - This dependency satisfies BCNF because A is a superkey.
- C→A
    - C is a candidate key (superkey).
    - This dependency satisfies BCNF because C is a superkey.
- D→A
    - D is a candidate key (superkey).
    - This dependency satisfies BCNF because D is a superkey.
- AD→E
    - AD is a superkey because it includes `A` and `D`, both candidate keys.
    - This dependency satisfies BCNF because AD is a superkey.
- AC→B
    - AC is a superkey because it includes `A` and `C`, both candidate keys.
    - This dependency satisfies BCNF because AC is a superkey.

So relation is in BCNF

PUBLICATION

- **Attributes**(publicationid, paperid, date, pageno)
- **Minimal Set**: A -> B, AD -> C, AB -> D
- **Candidate Keys**: A

First Normal Form (1NF):

1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes **the table is in 1NF.**

Second normal form:

1. A→B
   - A is a candidate key.
   - B is a non-prime attribute.
   - This dependency does not violate 2NF as AAA is the entire candidate key.
2. AD→C
   - AD includes A, which is a candidate key.
   - C is a non-prime attribute.
   - This dependency does not violate 2NF because AD involves the entire candidate key A and another attribute D.
3. AB→D
   - AB includes A, which is a candidate key.
   - D is a non-prime attribute.
   - This dependency does not violate 2NF because AB involves the entire candidate key A and another attribute B.

Since there are no partial dependencies of non-prime attributes on any proper subset of any candidate key, the relation is in 2NF.

So relation is in 2NF

Third Normal Form (3NF):

1. A→B
    - A is a candidate key (and therefore a superkey).
    - B is a non-prime attribute.
    - This dependency satisfies 3NF because A is a superkey.
2. AD→C
    - AD is a superkey.
    - C is a non-prime attribute.
3. AB→D
    - AB is a superkey .
    - D is a non-prime attribute.

So the relation is in 3NF.

So relation is in 3NF

BCNF:

1. A→B
    - A is a candidate key (and therefore a superkey).
    - This dependency satisfies BCNF because A is a superkey.
2. AD→C
    - AD is a superkey.
3. AB→D
    - AB is a superkey.

So relation is in BCNF

TOPICS :

- Attributes(topicid, topicname, description, parenttopic)
- Minimal Set: A -> C, A -> D, B -> A, CD -> B
- **Candidate Keys:** A, B, CD

First Normal Form (1NF):

1NF requires that all attributes contain only atomic (indivisible) values. Since there is no mention of multi-valued attributes **the table is in 1NF.**

Second normal form:

- A→C and A→D
  - Since `A` is a candidate key, these dependencies do not violate 2NF.
- B→A
  - Since `B` is a candidate key, it does not violate 2NF.
- CD→B
  - Since `CD` is a candidate key, this dependency also does not violate 2NF.

So relation is in 2NF

Third normal form:

- A→C:
  - `A` is a candidate key (and therefore a superkey).
  - This dependency satisfies 3NF because `A` is a superkey.
- A→D
  - `A` is a candidate key (and therefore a superkey).
  - This dependency satisfies 3NF because `A` is a superkey.
- B→A
  - `B` is a candidate key (and therefore a superkey).
  - This dependency satisfies 3NF because `B` is a superkey.
- CD→B
  - `CD` is a candidate key (and therefore a superkey).
  - This dependency satisfies 3NF because `CD` is a superkey.

So relation is in 3NF

BCNF:

- A→C:
    - `A` is a candidate key (and therefore a superkey).
    - This dependency satisfies BCNF because `A` is a superkey.
- A→D:
    - `A` is a candidate key (and therefore a superkey).
    - This dependency satisfies BCNF because `A` is a superkey.
- B→A:
    - `B` is a candidate key (and therefore a superkey).
    - This dependency satisfies BCNF because `B` is a superkey.
- CD→B
    - `CD` is a candidate key (and therefore a superkey).
    - This dependency satisfies BCNF because `CD` is a superkey.

So relation is in BCNF