❖ MINIMAL FD SET OF THE RELATION

UserID → Fname
UserID → Mname
UserID → Lname
UserID → Gender
UserID → Email
UserID → Password
UserID → Contact_No
UserID → Country
UserID → linkedin_profile
Email → UserID
Contact_No → UserID
linkedin_profile → UserID
FreelancerID → Bio
FreelancerID → hourly_wage
FreelancerID → experience_level
ClientID → hourly_wage
ClientID → payment_authentication
ClientID → company_name
ClientID → Bio
ProjectID → project_title
ProjectID → category
ProjectID → description
ProjectID → required_exp_level

ProjectID → budget
ProjectID → posted_on
ProjectID → deadline
ProjectID → payment_type
ProjectID → status
ProjectID → ClientID
{ProjectID, FreelancerID} → proposed_amount
{ProjectID, FreelancerID} → pitch
{ProjectID, FreelancerID} → req_time
{ProjectID, FreelancerID} → hourly_wage
{ProjectID, FreelancerID} → started_on
{ProjectID, FreelancerID} → deadline
{ProjectID, FreelancerID} → working_status
{ProjectID, FreelancerID} → completed_on
TranscationID → settled_on
TranscationID → method
TranscationID → amount
TranscationID → ProjectID
TranscationID → FreelancerID
TranscationID → ClientID
{ClientID, FreelancerID, ProjectID, comment}→ rating
TokenNo → issue_subject
TokenNo → issue_description
TokenNo → issue_category
TokenNo → issue_status

TokenNo → UserID
ForumNo → title
ForumNo → content
ForumNo → category
ForumNo → access_level
ForumNo → posted_on
ForumNo → UserID

❖ Proof that relations are in BCNF:

1. ‘User_Registration’ relation:

Attributes (UserID, Fname, Mname, Lname, gender, email, password, contact_no, country, linkedin_profile)

Functional Dependencies:

UserID → Fname
UserID → Mname
UserID → Lname
UserID → Gender
UserID → Email
UserID → Password
UserID → Contact_No
UserID → Country
UserID → linkedin_profile
Email → UserID
Contact_No → UserID
linkedin_profile → UserID

Let **X = UserID**
Clouse of X:

X^+ = {UserID, Fname, Mname, Lname, gender, email, password, contact_no , country, linkedin_profile}

Let **Y = email**
Clouse of Y:

Y^+ = {UserID, Fname, Mname, Lname, gender, email, password, contact_no , country, linkedin_profile}

Let **Z = linkedin_profile**
Clouse of Z:

Z^+ = {UserID, Fname, Mname, Lname, gender, email, password, contact_no , country, linkedin_profile}

Let **W = contact_no**
Clouse of W:

W^+ = {UserID, Fname, Mname, Lname, gender, email, password, contact_no , country, linkedin_profile}

Hence, **UserID, email, linkedin_profile, contact_no** are candidate keys of the relation**.**

The left side of FDs in the minimal FD set for the relation **'User_Registration'** is either **UserID, email, linkedin_profile or contact_no**, which are the candidate keys of this relation. Hence, relation **'User_Registration'** is in BCNF.

2. 'Freelancer' relation

Attributes (FreelancerID, hourly_wage, bio, experience_level)

Functional Dependencies:

FreelancerID → Bio
FreelancerID → hourly_wage
FreelancerID → experience_level

Let **X = FreelancerID**

Clouse of X:

X^+ = {FreelancerID, hourly_wage, bio, experience_level}

Since closure of **FreelancerID** has all the attributes of **'Freelancer'** relation, **FreelancerID** is the candidate key of this relation.

The left side of all the minimal FDs of the relation '**Freelancer**' is **FreelancerID**, which is the key of this relation, so '**Freelancer**' is in BCNF.

3. 'freelancer_skills' relation

Attributes (FreelancerID, skills)

Here, **Candidate Key** = {FreelancerID,skills}

In this relation, all attributes create a candidate key.
According to the theorem, such relations are always in BCNF.

So, '**freelancer_skills**' is in BCNF.

4. 'freelancer_edu' relation

Attributes (FreelancerID, education)

Here, **Candidate Key** = {FreelancerID, education}

All attributes are a part of the candidate key.
So, '**freelancer_edu**' is in BCNF.

5. 'freelancer_known_languages' relation

Attributes (FreelancerID, language)

Here, **Candidate Key** = {FreelancerID, language}

According to the theorem, if all the attributes of a relation make a candidate key, then the relation is always in BCNF.

So, '**freelancer_known_languages**' is in BCNF.

6. 'Client' relation

Attributes(ClientID, hourly_wage, payment_authentication, company_name, bio)

Functional Dependencies:
ClientID → hourly_wage
ClientID → payment_authentication
ClientID → company_name
ClientID → Bio

Let **X = ClientID**
Clouse of X:

X^+ = {ClientID, hourly_wage, payment_authentication, company_name, bio}

ClientID is the candidate key of a minimal FD set.

The left side of all the minimal FDs in the relation '**Client**' is **ClientID**, which is the key of this relation, so '**Client**' is in BCNF.

7. 'client_skills_expectation' relation

Attributes(ClientID,expected_skills)

Here **Candidate Key** = {ClientID,expected_skills}

According to the theorem, if all the attributes of a relation make a candidate key, then the relation is always in BCNF.

So, '**client_skills_expectation**' is in BCNF.

8. 'Project' relation

Attributes (ProjectID, project_title, category, description, required_exp_lvl, budget, posted_on, deadline, payment_type, status, ClientID)

Functional Dependencies:

ProjectID → project_title
ProjectID → category
ProjectID → description
ProjectID → required_exp_level
ProjectID → budget
ProjectID → posted_on
ProjectID → deadline
ProjectID → payment_type

ProjectID → status
ProjectID → ClientID

Let **X = ProjectID**
Clouse of X:

X^+ = {ProjectID,project_title,category,description,required_exp_lvl,budget, posted_on,deadline,payment_type,status,ClientID}

Since closure of **ProjectID** has all the attributes of **'Project'** relation, **ProjectID** is the candidate key.

The left side of all the minimal FDs in the relation '**Project**' is **ProjectID** , which is the key of this relation, so '**Project**' is in BCNF.

9. 'project_skills_required' relation

Attributes(ProjectID,required_skills)

Here **Candidate Key** = {ProjectID,required_skills}\

All attributes are a part of the candidate key.
So, **'project_skills_required'** is in BCNF.

10. 'Proposal' relation

Attributes (FreelancerID**,** ProjectID, proposed_amount, pitch, req_time)

Functional Dependencies:
{ProjectID, FreelancerID} → proposed_amount
{ProjectID, FreelancerID} → pitch
{ProjectID, FreelancerID} → req_time

Let **X = {ProjectID, FreelancerID}**
Clouse of X:

X^+ = {FreelancerID**,** ProjectID, proposed_amount, pitch, req_time}
{ProjectID, FreelancerID} is the candidate key of a minimal FD set.

The left side of all the FDs in the minimal set of FDs for the relation '**Proposal**' is **{ProjectID, FreelancerID}**, which is the key of this relation, so '**Proposal**' is in BCNF.

11. 'works_on' relation

Attributes (ProjectID, FreelancerID, hourly_wage, started_on, deadline, working_status, completed_on)

Functional Dependencies:
{ProjectID, FreelancerID} → hourly_wage
{ProjectID, FreelancerID} → started_on
{ProjectID, FreelancerID} → deadline
{ProjectID, FreelancerID} → working_status
{ProjectID, FreelancerID} → completed_on

Let **X = {ProjectID, FreelancerID}**
Clouse of X:
X^+ = {ProjectID,FreelancerID,hourly_wage,started_on,deadline, Working_status, completed_on}

{ProjectID, FreelancerID} is the candidate key of a minimal FD set.

The left side of all the minimal FDs of the relation **'works_on'** is **{ProjectID, FreelancerID},** which is the key of this relation, so **'works_on'** is in BCNF.

12. 'Transactions' relation

Attributes (TransactionID, settled_on, method, amount, ProjectID, FreelancerID, ClientID)

Functional Dependencies:
TranscationID → settled_on

TranscationID → method
TranscationID → amount
TranscationID → ProjectID
TranscationID → FreelancerID
TranscationID → ClientID
{ProjectID, FreelancerID, ClientID} → TransactionID

Let **X = {TranscationID}**
Clouse of X:

X^+ = {TransactionID , settled_on,method,amount,ProjectID,FreelancerID, ClientID}

Let **Y = {ProjectID, FreelancerID, ClientID}**
Clouse of Y:

Y^+ = {TransactionID , settled_on,method,amount,ProjectID,FreelancerID, ClientID}

Thus, **TranscationID, {ProjectID, FreelancerID, ClientID}** is the candidate key of a minimal FD set.

The left side of all the minimal FDs of the relation **'Transactions'** is candidate key, so **'Transactions'** is in BCNF.

13. 'Reviews' relation

Attributes (ProjectID, FreelancerID, ClientID, comment, rating)

Functional Dependencies:
{ClientID, FreelancerID, ProjectID, comment} → rating

Let **X = {ClientID, FreelancerID, ProjectID, comment}**
Clouse of X:

X^+ = {ProjectID,FreelancerID,ClientID,comment,rating}

{ClientID, FreelancerID, ProjectID, comment} is the candidate key of a minimal FD set.

The left side of all the minimal FDs of the relation **'Reviews'** is **{ClientID, FreelancerID, ProjectID, comment}** which is the key of this relation, so **'Reviews'** is in BCNF.

14. 'customer_support' relation

Attributes (TokenNo, issue_subject, issue_description, issue_category, issue_status, UserID)

Functional Dependencies:

TokenNo → issue_subject
TokenNo → issue_description
TokenNo → issue_category
TokenNo → issue_status
TokenNo → UserID

Let **X = TokenNo**
Clouse of X:

X^+ = {TokenNo,issue_subject,issue_description,issue_category, issue_status,UserID}

Since closure of **TokenNo** has all the attributes of **'customer_support'** relation, **TokenNo** is the candidate key.

The left side of all the minimal FDs of the relation **'customer_support'** is **TokenNo**, which is the key of this relation, so **'customer_support'** is in BCNF.

15. 'discussion_forum' relation

Attributes(ForumID,title,content,category,access_level,posted_on,UserID)

Functional Dependencies:

ForumNo → title
ForumNo → content
ForumNo → category
ForumNo → access_level
ForumNo → posted_on

ForumNo → UserID

Let **X = ForumID**
Clouse of X:

X^+ = {ForumID,title,content,category,access_level,posted_on,UserID}

Since closure of **ForumID** has all the attributes of **'discussion_forum'** relation, **ForumID** is the candidate key.

The left side of all the minimal FDs of the relation '**discussion_forum**' is **ForumID**, which is the key of this relation, so '**discussion_forum**' is in BCNF.