**CPS510 Fall2021 Section 04**

**Group-11**

**Assignment 8**

Normalization / BCNF

**Conversion of 3NF to BCNF Normalize form by using Algorithm**

**Application Name: Online Job Bank System.**

Group Member names:

1. Tusaif Azmat

Student#: 500660278

1. Ankit Dheedsa

Student#: 500975118

1. Mahdi Alam

Student#: 500969935

**-- Normalization/ Conversion of 3NF to BCNF with the help of Algorithm.**

**--- We used Bernstein’s Algorithm to achieve this task:**

**Bernstein’s Algorithm - Broken down into 4 steps:**

## Determine all the functional dependencies

## a) Find and remove redundancies

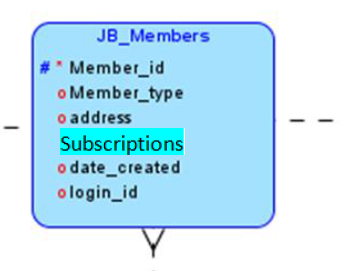
## b) Find and remove partial dependencies

## Find keys

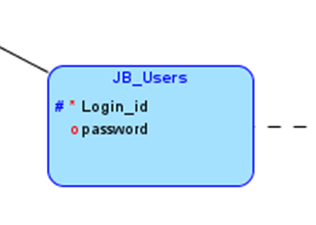
## Create tables

# Step 1 (finding all functional dependencies)

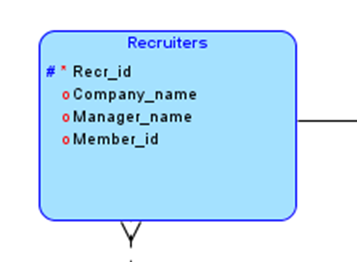
- **Member\_id** 🡪 {Member\_type, address, date\_created, Subscriptions}



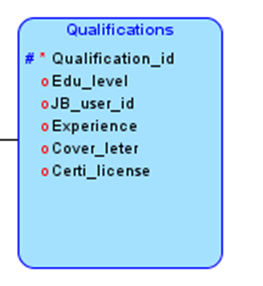
* **Login\_id** 🡪 { password}



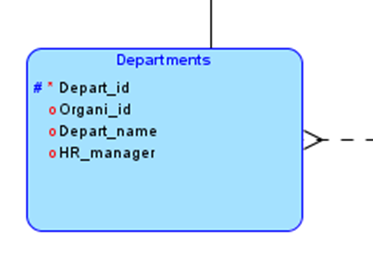
* **Recr\_id** 🡪 {company\_name, Manager\_name}



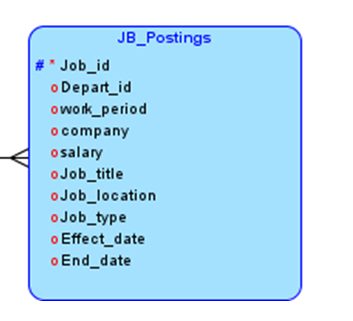
* **Qualification\_id🡪** {Edu\_level, Experience, cover\_letter, Certi\_license}



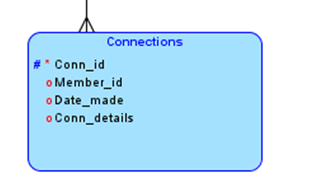
* **depart\_id** 🡪 {depart\_name, HR\_manager}



* **job\_id** 🡪 {work\_period, company, salary, job\_title, job\_location, job\_type, Effect\_date, End\_date}



* **Conn\_id** 🡪 {Date\_made, Conn\_details}



# Step 2a (Break RHS and find redundancies)

Get rid of Redundancies

- **Member\_id** 🡪 {Member\_type, address, date\_created, Subscriptions}

* + Reduced list of FD’s:
  + **Member\_id🡪** {Member\_type}
  + **Member\_id** 🡪{address}
  + **Member\_id** 🡪{date\_created}
  + **Member\_id** 🡪{Subscriptions}

# No redundancies

* **Login\_id** 🡪 { password}
  + Reduced list of FD’s:
  + **Login\_id** -> {password}

# No redundancies

* **Recr\_id** 🡪 {company\_name, Manager\_name}
  + Reduced list of FD’s:
  + **Recr\_id** 🡪 {company\_name}
  + **Recr\_id** 🡪 {Manger\_name}

# No redundancies

* **Qualification\_id 🡪** {Edu\_level, experience, cover\_letter, Certi\_license}
  + Reduced list of FD’s:
  + **Qualification\_id** 🡪 {user\_degree}
* **Qualification\_id** 🡪{ experience}
* **Qualification\_id** 🡪{cover\_letter}
* **Qualification\_id** 🡪{Certi\_license}

# No redundancies

* **depart\_id 🡪** {depart\_name, HR\_manager}
  + Reduced list of FD’s:
  + **depart\_id** 🡪 {depart\_name}
  + **depart\_id** 🡪 {HR\_manager}

# No redundancies

* **job\_id** 🡪 {work\_period, company, salary, job\_title, job\_location, job\_type, Effect\_date, End\_date}
  + Reduced list of FD’s:
  + **job\_id** 🡪{work\_period}
  + **job\_id** 🡪{company}
  + **job\_id** 🡪 {salary}
  + **job\_id** 🡪 {job\_title}
  + **job\_id** 🡪 {job\_location}
  + **job\_id** 🡪{job\_type}
  + **job\_id** 🡪{Effect\_date}
  + **job\_id** 🡪{End\_date}

# No redundancies

* **Conn\_id** 🡪 {money}
  + Reduced list of FD’s:
  + **conn\_id 🡪** {Date\_made}
  + **conn\_id 🡪** {conn\_details}

# No redundancies

**Step 2b (Minimize LHS, find and remove partial dependencies)**

* LHS is already minimized, therefore there are no partial dependencies

# Step 3(Find keys) (relational schema)

* **Member\_id🡪** {Member\_type, address, date\_created, subscriptions}
  + Attributes on RHS but not on LHS (cannot be keys)
    - Member\_type
    - Address
    - Date\_created
    - Subscriptions
  + Possible Keys
    - Member\_id
* **Login\_id** 🡪 { password}
  + Attributes on RHS but not on LHS (cannot be keys)
    - password
  + Possible Keys
    - Login\_id
* **Recr\_id** 🡪 {company\_name, manager\_name}
  + Attributes on RHS but not on LHS (cannot be keys)
    - Company\_name
    - Manger\_name
  + Possible Keys
    - Recr\_id
* **depart\_id** 🡪 {depart\_name, HR\_manager}
  + Attributes on RHS but not on LHS (cannot be keys)
    - depart\_name
    - HR\_Manger
  + Possible Keys
    - depart\_id
* **qualification\_id** 🡪 {Edu\_level, experience, cover\_letter, Certi\_license}
  + Attributes on RHS but not on LHS (cannot be keys)
    - Edu\_level
    - Experience
    - Cover\_letter
    - Certi\_license
  + Possible Keys
    - Qualification\_id
* **job\_id** 🡪 {work\_period, company, salary, job\_title, job\_location, job\_type, Effect\_date, End\_date}
  + Attributes on RHS but not on LHS (cannot be keys)
    - work\_period
    - company
    - salary
    - job\_title
    - job\_location
    - job\_type
    - Effect\_date
    - End\_date
  + Possible Keys
    - job\_id
* **Conn\_id** 🡪 {Date\_made, Conn\_details}
  + Attributes on RHS but not on LHS (cannot be keys)
    - Date\_made
    - Conn\_details
  + Possible Keys
    - Conn\_id

**Step 4(Make tables)**

**R1** (**member\_id**, member\_type, address, date\_created, subscriptions, login\_id)

With

FD: **member\_id** 🡪 { member\_type, address, date\_created , subscriptions }

**R2** (**Login\_id**, password)

With

FD: **user\_id** 🡪{ user\_password}

**R3** (**recr\_id**, company\_name, manager\_name, member\_id)

With

FD: **recr\_id** 🡪{company\_name, manager\_name }

**R4** (**depart\_id**, depart\_name, hr\_manager,Org\_id)

With

FD: **depart\_id** 🡪{ depart\_name, hr\_manager }

**R5** (**qualification\_id**, edu\_level, experience, cover\_letter, Certi\_license, user\_id)

With

FD: **qualification\_id** 🡪 { edu\_level, experience, cover\_letter, Certi\_license }

**R6** (**job\_id**, work\_period, company, salary, job\_title, job\_location, job\_type, effect\_date,end\_date, depart\_id)

With

FD: **job\_id** 🡪{, work\_period, company, salary, job\_title, job\_location, job\_type, effect\_date,end\_date }

**R7** (**conn\_id**, date\_made, conn\_details, member\_id)

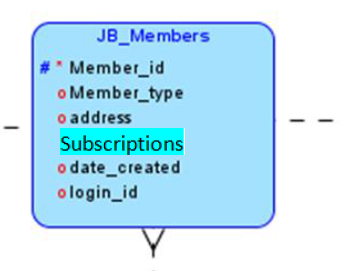
With

FD: **conn\_id** 🡪 { date\_made, conn\_details }

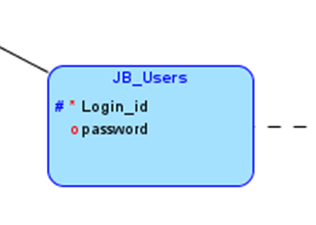
**BCNF (Boyce/Codd Normal Form)**

# Step 1 (finding all functional dependencies (List of all attributes and FDs))

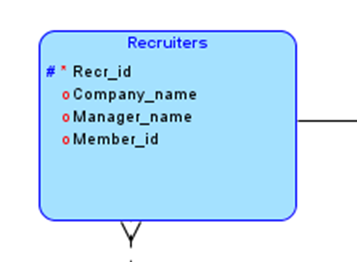
- **member\_id** 🡪 { member\_type, address, date\_created, subscriptions, login\_id }



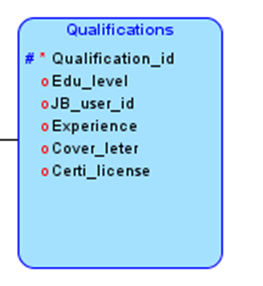
* **Login\_id** 🡪 { password}



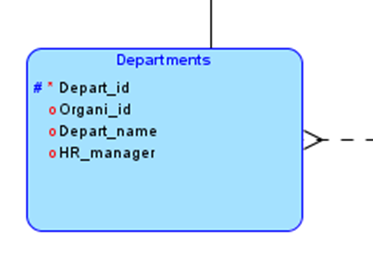
* **Recr\_id** 🡪 {company\_name, Manager\_name, member\_id}



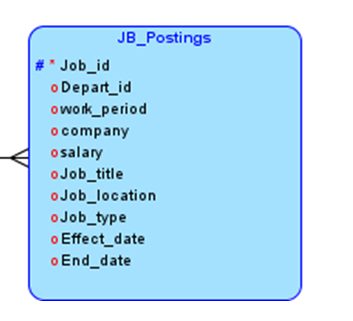
* **Qualification\_id🡪** {Edu\_level, Experience, cover\_letter, Certi\_license, jb\_user\_id}



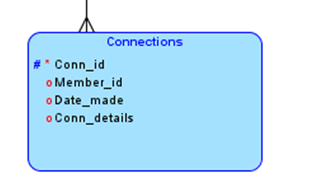
* **depart\_id** 🡪 {depart name, HR\_manager, Organi\_id}



* **job\_id** 🡪 {work\_period, company, salary, job\_title, job\_location, job\_type, Effect\_date, End\_date, depart\_id, Depart\_id}



* **Conn\_id** 🡪 {Date\_made, Conn\_details, member\_id}



# Step 2 (make sure that the left hand side are keys, if not decompose)

Consider the relation schema

**R1** (**member\_id**, member\_type, address, date\_created, subscriptions, login\_id)

With FD

**member\_id** 🡪 { member\_type, address, date\_created , subscriptions }

This schema has one candidate key

member\_id

Therefore this schema is in BCNF

Consider the relation schema

**R2** (**Login\_id**, password)

With FD

**Login\_id** 🡪 {password}

This schema has one candidate key

login\_id

Therefore this schema is in BCNF

Consider the relation schema

**R3** (**recr\_id**, company\_name, manager\_name, member\_id)

With FD

**recr\_id** 🡪 {company\_name, manager\_name }

This schema has one candidate key

recr\_id

Therefore this schema is in BCNF

Consider the relation schema

**R4** (**depart\_id**, depart\_name, hr\_manager,Org\_id)

With FD

**depart\_id** 🡪 { depart\_name, hr\_manager }

This schema has one candidate key

depart\_id

Therefore this schema is in BCNF

Consider the relation schema

**R5** (**qualification\_id**, edu\_level, experience, cover\_letter, Certi\_license, user\_id)

With FD

**qualification\_id** 🡪 { edu\_level, experience, cover\_letter, Certi\_license }

This schema has one candidate key

qualification\_id

Therefore this schema is in BCNF

Consider the relation schema

**R6** (**job\_id**, work\_period, company, salary, job\_title, job\_location, job\_type, effect\_date,end\_date, depart\_id)

With FD

**job\_id** 🡪 { work\_period, company, salary, job\_title, job\_location, job\_type,effect\_date,end\_date}

This schema has one candidate key

job\_id

Therefore this schema is in BCNF

Consider the relation schema

**R7** (**conn\_id**, date\_made, conn\_details, member\_id)

With FD

**conn\_id** 🡪 { date\_made, conn\_details }

This schema has one candidate key

conn\_id

Therefore this schema is in BCNF

**Step 3 (final BCNF schema for R)**

**R1** (**member\_id**, member\_type, address, date\_created, subscriptions, login\_id)

**R2** (**Login\_id**, password)

**R3** (**recr\_id**, company\_name, manager\_name, member\_id)

**R4** (**depart\_id**, depart\_name, hr\_manager,Org\_id)

**R5** (**qualification\_id**, edu\_level, experience, cover\_letter, Certi\_license, user\_id)

**R6** (**job\_id**, work\_period, company, salary, job\_title, job\_location, job\_type, effect\_date,end\_date, depart\_id)

**R7** (**conn\_id**, date\_made, conn\_details, member\_id)

Note: All the tables are of the form BCNF and we didn’t need to combine any tables for this stage.