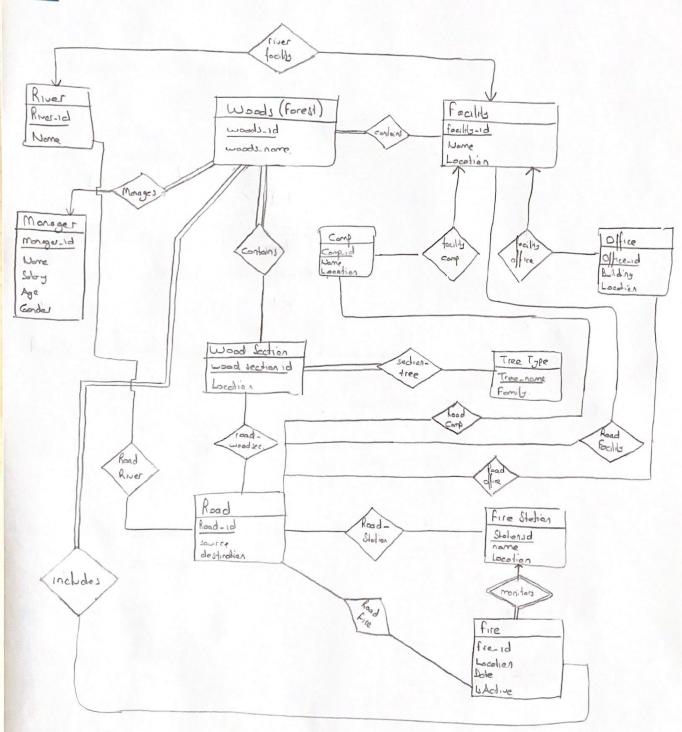
Part 1



Functional Defendencies

-> For forest table

[woods_id, woods_name] -> woods_id

["] -> woods_name.

Trivial fine. dop.

woods_name -> woods_name

woods-id - woods-none -> Non-trivial force dof.

There isn't transitives or multivolved dependency

-> for Facility Table

[focility-id, none, location] -> focility-id

["] -> none

["] -> location

[focility-id, norme] -> focility-id

["] -> none

[| ocility-id, location] -> facility-id

["] -> location

[name, location] -> name

["] -> location

Trivid func. dep.

name -> name

facility-id -> facility-id

location -> location

Focility id -> None, Loadion -> Non-trival func. dep.

There isn't transitive or multivolved def.

-> For River Toble

[River.1d, None] -> River.1d

["] -> None

River.id -> River.id

None -> None

Rivered - None - Non-trivid fore dp.

There isn't trasitive or multivolved dependency.

```
[Manger-id, Name, Solary, Age, Gender] -> Monger-id
                                               [Morger - id, None, Ago, Gender] -> Morger-id
                                                                             ] -> None
                               ] -> None
                                                                             J -> Age
                                7-25-log
                                                                              ] -> Garder
                                J → Age
                                                [Morger_id, Name, Solory, Gerder] -> Morager_id
                                ] -> Gender.
                                                                              -> None
[morger_id, Sobra, Age, Gender] -> morger_id
                                                                              T -> Solos
                             ] -> Solar
                                                                             ] -> Bender
                             → Age.
                                              [Marger-1d, Name, Sdiz, Age] -> Marger-1d
                             ] -> Gender.
                                                                       ] - None
                                                                       ] -> Selvs
                                                                       ] -> Berde
Manager - id -> Manager - id
Done -> None
Solary -> Solar
                                     Alon-ta
Ager -> Age
                                                                      · There 150:+
                                  · Trivial function dependency
Gerder -> conde,
                                                                       transitive and
                                                                      multivolved dependency
```

Manager-id -> Name, Solary, Age, Gender -> Non-trivial fred; dependency.

-> for comp toble

```
[Comp.id, Name, lacoton] -> Comp.id [Name, bootion] -> Name

[ "] -> Name
[ "] -> lacotion
[ Comp.id, lacotion] -> comp.id
[ Comp.id, lacotion] -> comp.id
[ "] -> lacotion
[ Comp.id, Name] -> comp.id
[ "] -> Name
[ "] -> Name
[ "] -> Name
[ "] -> Name
```

Compild -> None, location -> Non-trivial func. dep.
There isn't transitive and multivalved dep.

- office dobb

[office_id, bilding, location] -> office_id

[office_id, bilding, location] -> office_id

[office_id, bilding] -> office_id

Office-1d -> bilding, location -> non-trivial func. dep.
There isn't transitive or multivalved dep.

-> for wood section table

[wood-sectionid, location] -> wood-sectionid]

[wood-sectioned -> wood-sectioned]

Trivial fine dep.

location -> boother

wood-sectioned -> location -> Won-trivial fune dep.

There isn't travitive of multivolved dep.

-> for Tree Type table

[Tree-none, family] -> Tree-none

["] -> family

Tree-none -> Tree-none

family -> family

tree-none -> family -> non-trivial func. dep.

There isn't travitive or multivalued dep.

-> For Aoad table

Road-id -> source, destination -> Non-truial function appendency
There isn't transitive or multivalued appendency

-> for fire Addien toble

[station-id, name, location] -> station-id [name, location] -> name

["] -> name

["] -> location

[station-id, name] -> station-id

[station-id, name] -> station-id

[name -> name

[station-id, name] -> station-id

[name -> name

[location] -> name

[station-id, location] -> station-id

[station-id, location] -> station-id

[name -> location]

Trivial fine.

[station-id, location] -> station-id

[name -> location]

Trivial fine.

[station-id, location] -> station-id

[name -> location]

Trivial fine.

Station-id -> none, location -> non-trivial func. dap.

There isn't trasitive or multivolved dep.

-> for fire table

[fire_id, location, date, is Active] -> fire_id

[fire_id, location, date, is Active] -> fire_id

[fire_id, location, is Active] -> fire_id

[fire_id, location, is Active] -> fire_id

fire-d > location, date is Active - o non-trival func. dep.
There isn't travitive or multivalued dep.

a) o Facility table holds the criterios of Doyce-Codd Normal form

-> facility (fectity-id, name, location)

Roaf: A relation schema R is in BCNF with respect to a set F of functional defendences in Ft of the form

d-)B

where X ER and \$ ER, at least one of the following holds:

→ d → p is trivial (B ⊆ d)

If we look at the trivial and non-trivial defendencies shown in the facility table.

In Part 2. The trivial functional dependencies meets the X-> \beta is trivial or \beta \le X rule

Thus, BCNF is provided with this conditions

-other trivial func dep. examples:

[focility-id, name, location] -> focility-id

[focility-id, none] -> focility-id

name-snone

facility-id -> facility-id

lacotion -> location

-> non-trivial fine. dep. examples:

facility id -> name, location => These are non-trivial
but facility-id is superkey
so these holds the BCNF

So the facility table holds the BCNF

· wood Section lide holds the criteria of Bayce-codd Normal form

-> wood Section (wood-section-id, because)

Proof: A relation scheme. R is in BCNF with respect to a set F of functional dependencies $x \rightarrow b$

where $\alpha \subseteq R$ and $\beta \subseteq R$, of least one of the following holds:

-> d -> b 15 trivial (\$ Ed)

-> the trivial func dep. examples:

[wood-section_id, location] => wood-section_id

wood-section-id-> wood-section-id

-> non-trivial fux. dep. examples:

wood. section_id -> location => There are non-trivial but wood-section-id

15 superkey so these holds the BCMF

So the wood section toble holds the BCNF

b) Because of all the functional dependencies specified in Port 2, there is no toble that also not tit the BCNF form.

Part 4

a) . Road table holds the criterios of 3NF

-> Road (road-id, source, destination)

Proof: A relation schema R is in third normal form it for all.

at least one of the following holds

-> d -> p 13 trivial (i.e., ped)

-> of 10 a superkey for R

-> Each attribute A in B-er is contained in a condidate key for R

If a relation is in BCUF H is in BUF (Since in BCUF one of the first two conditions above must hold)

Third condition is a minimal relaxation of BCNF to ensure depending preservation

-> If we look at the trivial and non-trivial dependencies shown in the Road table in part 2. The trivial function dependencies mades the X-2 B in F. Thus these holds criterias of 3NF

-> the trivial func. dop. examples	Polih Selin JAKAR 161044054
[Road_id, source, destination] -> road-id	
["] -> source	
[] -> destination	
[Road-id, source] -> Road-id	
Rood-id-> Rood-id	
Surce -> Sauce	
noition that combination	
-> non-trival fine. dep. examples	
road-id -> source, deplication => These are non-trivial but road-id is experted so these holds the 3UF	
· fire Stotion toble holds the criterios of JNF	
-> Fire Station (station-id, name, location)	
Proof: A relation schema R is in third normal form it for all.	
at least one of the following holds:	
-> d -> B is trivial (i.e., BE d)	
-> or 15 a superkey for R -> Each attribute An B-or 15 contained in a condidate key for R	
If a relation is in BCNF it is in 3NF (since in BCNF one of the first two a	endobers observe.
Third condition is a minimal relaxation of CUF to entire dependency produc	notes
-> If we look at the truid and non-trivial dependencies shown in the in path. The trivial function dependencies meets the x-> f in ft. Thus these of 3NF	
-> the trivial func. dop. examples	
[Station-id, nome, location] -> station-id Station-id	
[none -> nome	
[//] -> location -> location	
[stetrer - id, none] - stetron -id	
[//] -> Name.	
8	

Stationard > name, location => These are non-trivial but

stationard is superkey so those
holds the SUF

6) because of all the frechenol dependencies speciatived in ford 2, there is no table that also not fit the 3UF form