Ex: 9 NF & ER MODELLING

-S. Vishakan CSE – C 18 5001 196

9A: Normal Forms

The Primary Keys of the given **COMPANY** Schema was found to be **empid & pno** from the given functional dependencies.

{empid, pno}+ : {empid, name, address, bdate, sex, salary, dno, dname, mgr_id, pno, pname, pdno, hrs}
The closure of {empid, pno} contains all the attributes of the relation.

Functional Dependencies

FD1 : empid -> name, address, bdate, sex, salary, dno
FD2 : dno -> dname, mgr_id
FD3 : pno -> pname, pdno
FD4 : empid, pno -> hrs

1NF Normalization

The relations are already in 1NF form as there are no multi-valued attributes or composite attributes.

COMPANY(empid, name, address, bdate, sex, salary, dno, dname, mgr_id, pno, pname, pdno, hrs)

2NF Normalization

Since the attributes pname, pdno are non-key attributes that are not dependent upon empid and the attributes name, address, bdate, sex, salary, dno, dname, mgr_id are not dependent upon pno, we split them into 2 relations **EMPLOYEES** and **PROJECT**, with **pno** and **empid** their respective key attributes, since they violate the 2NF rule of non-prime attributes being fully functionally dependent upon all the prime attributes.

Also, the hrs attribute is completely dependent upon the key attributes, those 3 attributes comprise another relation namely **PROGRESS**, with the **empid and pno** being the key attributes. **Empid** and **pno** are also foreign keys here which are referenced from the **EMPLOYEES** and **PROJECT** relations.

Now, the relations are in 2NF.

EMPLOYEES	(empid, name, address, bdate, sex, salary, dno, dname, mgr_id)
PROJECT	(pno, pname, pdno)
PROGRESS	(empid, pno, hrs)

3NF Normalization

Since dname, mgr_id of **EMPLOYEES** relation are transitively dependent upon empid through dno, they have to be broken down into **EMPLOYEES** and **DEPARTMENTS** respectively, as 3NF disallows transitive dependency. **Dno** is also referenced as a foreign key in **EMPLOYEES** relation from the **DEPARTMENTS** relation where it is a key attribute.

Now, the relations are in 3NF.

EMPLOYEES	(empid, name, address, bdate, sex, salary, dno)
DEPARTMENTS	(dno, dname, mgr_id)
PROJECT	(pno, pname, pdno)
PROGRESS	(empid, pno, hrs)

The **Lossless Join** property has been verified in the **SQL Spool File.**

Proving Preservation of FDs:

Initially we have,

FD1 : empid -> name, address, bdate, sex, salary, dno
FD2 : dno -> dname, mgr_id
FD3 : pno -> pname, pdno
FD4 : empid, pno -> hrs

F = {FD1, FD2, FD3, FD4}

After decomposition to 3NF,

FD1 is preserved in the **EMPLOYEES** relation. F1 = {FD1}
In **EMPLOYEES**, Closure of {empid}+ : {empid, name, address, bdate, sex, salary, dno}
FD2 is preserved in the **DEPARTMENTS** relation. F2 = {FD2}
In **DEPARTMENTS**, Closure of {dno}+ : {dno, dname, mgr_id}
FD3 is preserved in the **PROJECT** relation. F3 = {FD3}
In **PROJECT**, Closure of {pno}+ : {pno, pname, pdno}
FD4 is preserved in the **PROGRESS** relation. F4 = {FD4}
In **PROGRESS**, Closure of {empid, pno}+ : {empid, pno, hrs}

$F1 \cup F2 \cup F3 \cup F4 = \{FD1, FD2, FD3, FD4\} = F$

Thus, functional dependencies are preserved across the different relations even after decomposition.

9B: ER Model and ER to Relational Mapping