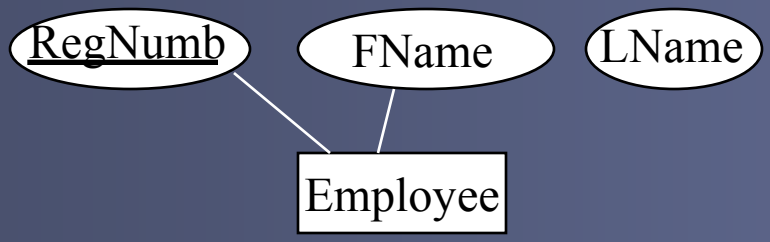
Database Concepts
Entity-Relationship to Relational Schema

How to do you translate a Entity-Relationship Model into a Relational Schema?

(*Part 3)*

Database Concepts
Entity-Relationship to Relational Schema

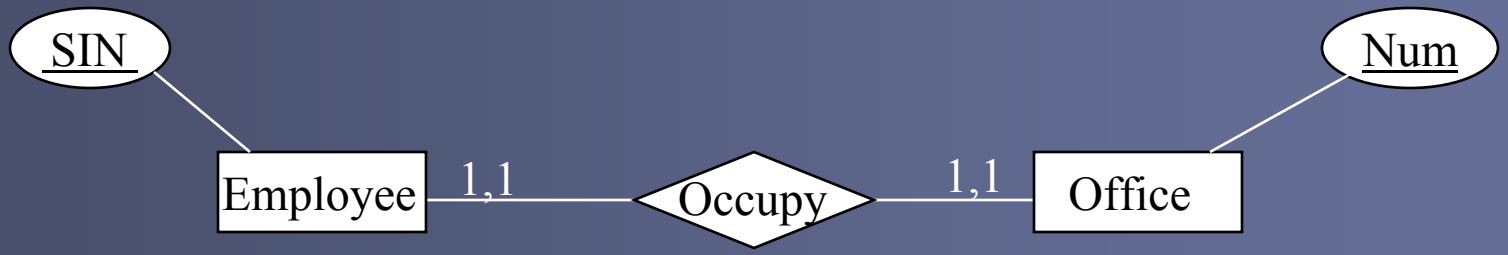
- Entity
  - Each entity type becomes a relation. Its identifier becomes a key and its properties becomes attributes. Multivalued properties are transformed such that they conform to 1NF (attributes must be simple and monovalued).



Employee(RegNumb. , LName, FName, …)

Database Concepts
Entity-Relationship to Relational Schema

- One-to-one relationship {0,1 | 1,1} to {0,1 | 1,1}
  - Each Relationship Type translates into:
    - a fusion of the two corresponding relations
    - and an addition of a foreign key into one or both of the relations.



Option 1 :
Employee(SIN, …, #Num)
Office(Num, …)

Option 2 :
Office(Num, …, #SIN)
Employee(SIN, …)

Database Concepts
Entity-Relationship to Relational Schema

- One-to-many Relationships {0,1 | 1,1} to {0,N | 1,N}
  - each Relationship Type translates by adding a a foreign key.

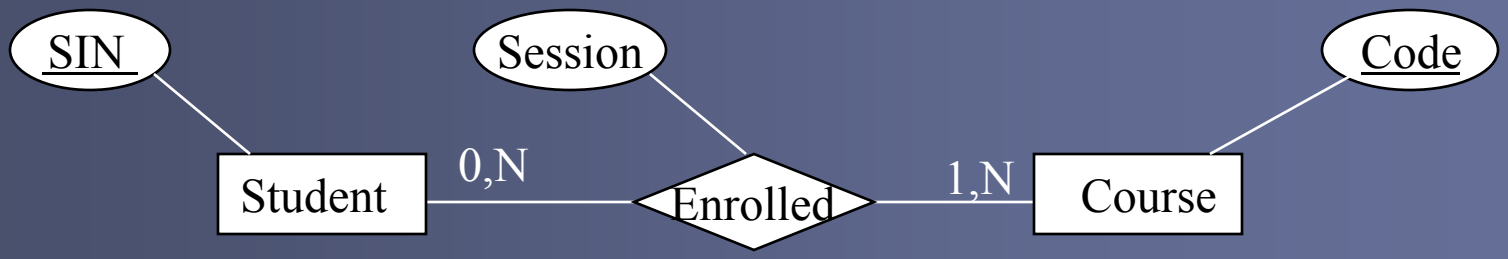
Employee(SIN, …, #Name_D)
Department(Name_D, …)

Database Concepts
Entity-Relationship to Relational Schema

- Many-to-many Relationships

  {0,N | 1,N} vers {0,N | 1,N}

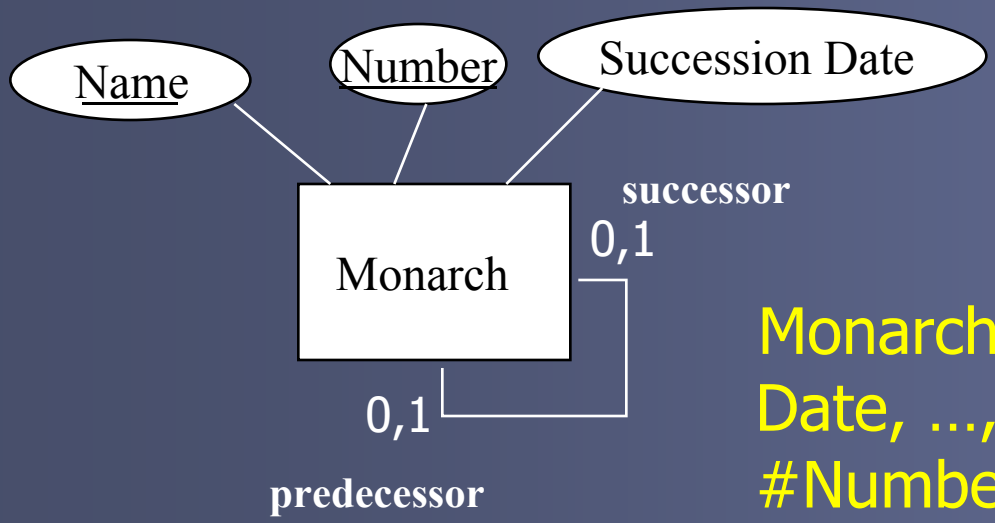
  – Each Relationship Type becomes a Relation
  – The indentifiers of the participating entities becomes the key of the relation.



Enrolled(#SIN, #Code, Session)
SIN and Code are both foreign keys

Database Concepts
Entity-Relationship to Relational Schema

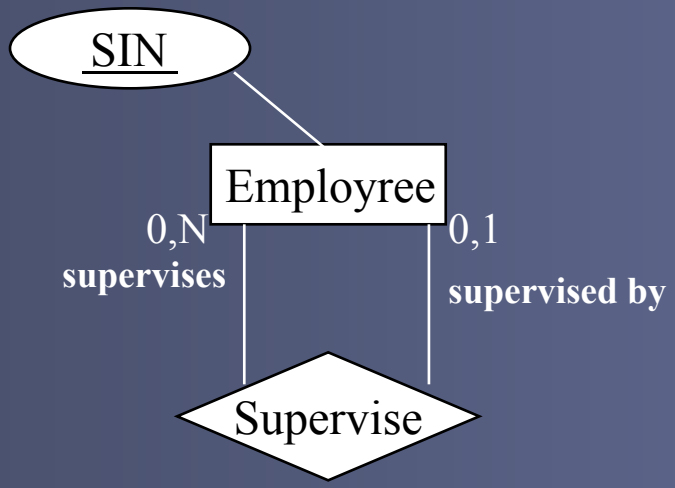
- Recursive One-to-one Relationship
  - This Relationship Type translates into:
    - The addition of two foreign keys into the relation.



Monarch(Name, Number, Succession Date, …, #NamePred, #NumberPred , #NameSucc, #NumberSucc)

Database Concepts
Entity-Relationship to R elational Schema

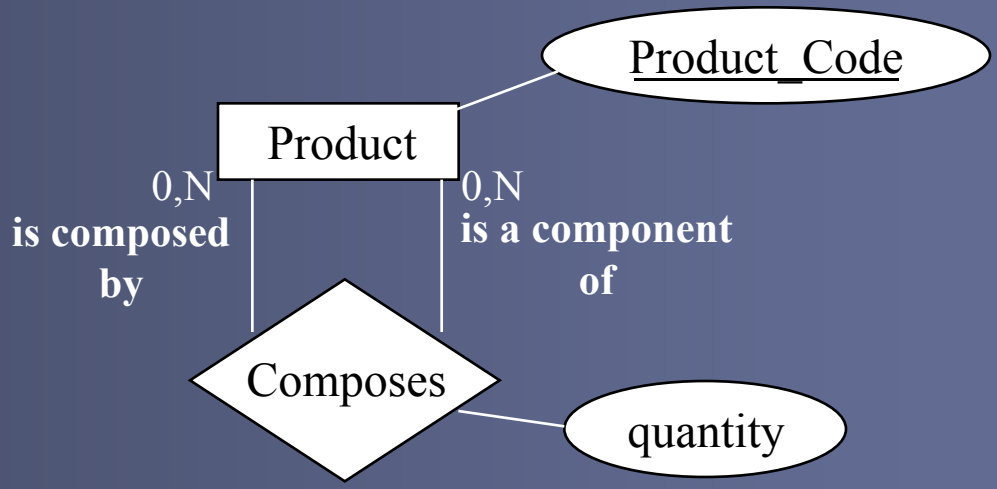
- Recursive one-to-many Relationships:
  - Each Relationship Type translates by the addition of a foreign key



Employee(SIN, …, #SUPSIN)

Database Concepts
Entity-Relationship to R elational Schema

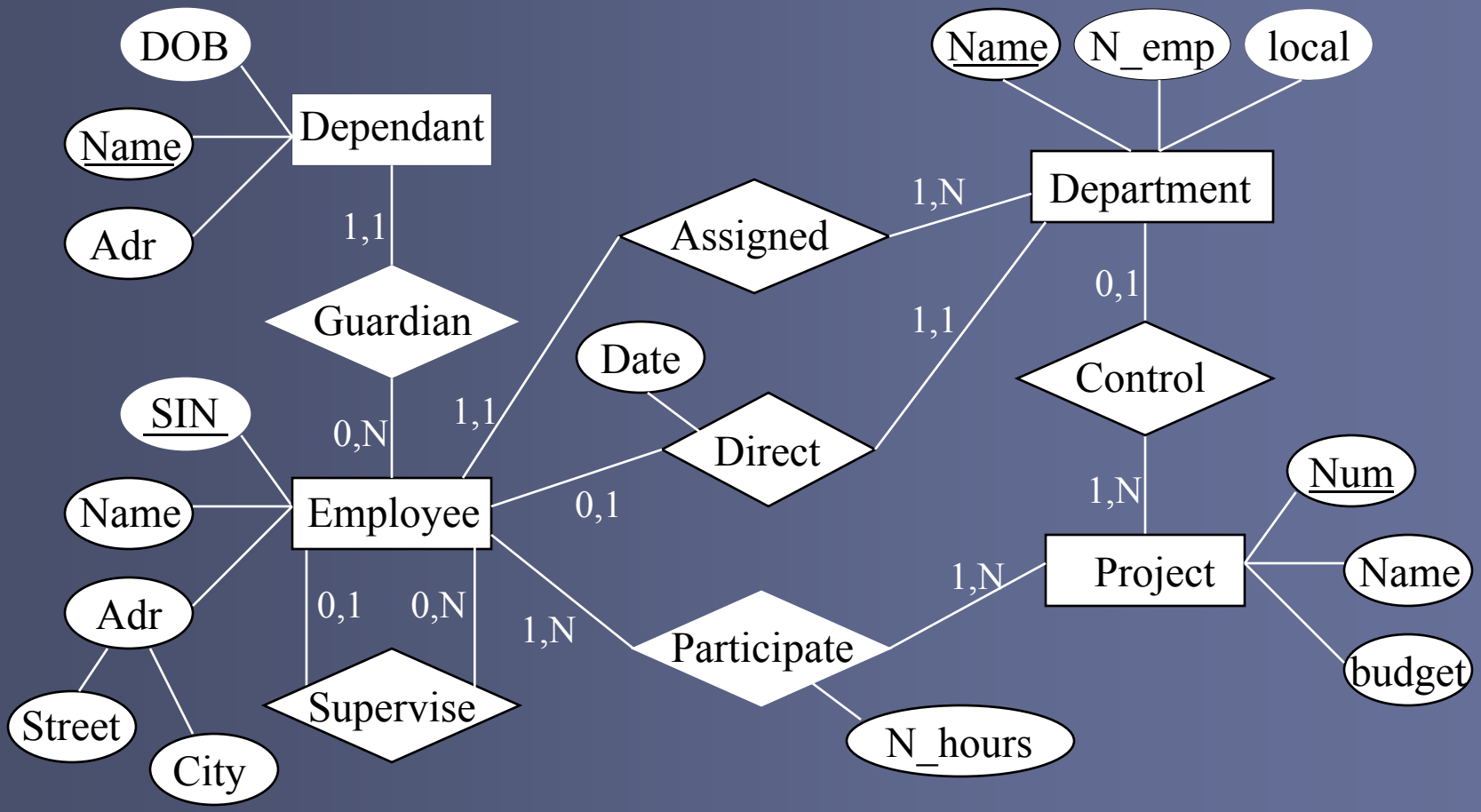
- Recursive many-to-many Relationship Types:
  - Each Relationship Type becomes a Relation.
  - the identifiers of the relationship type becomes becomes the key of the relation.



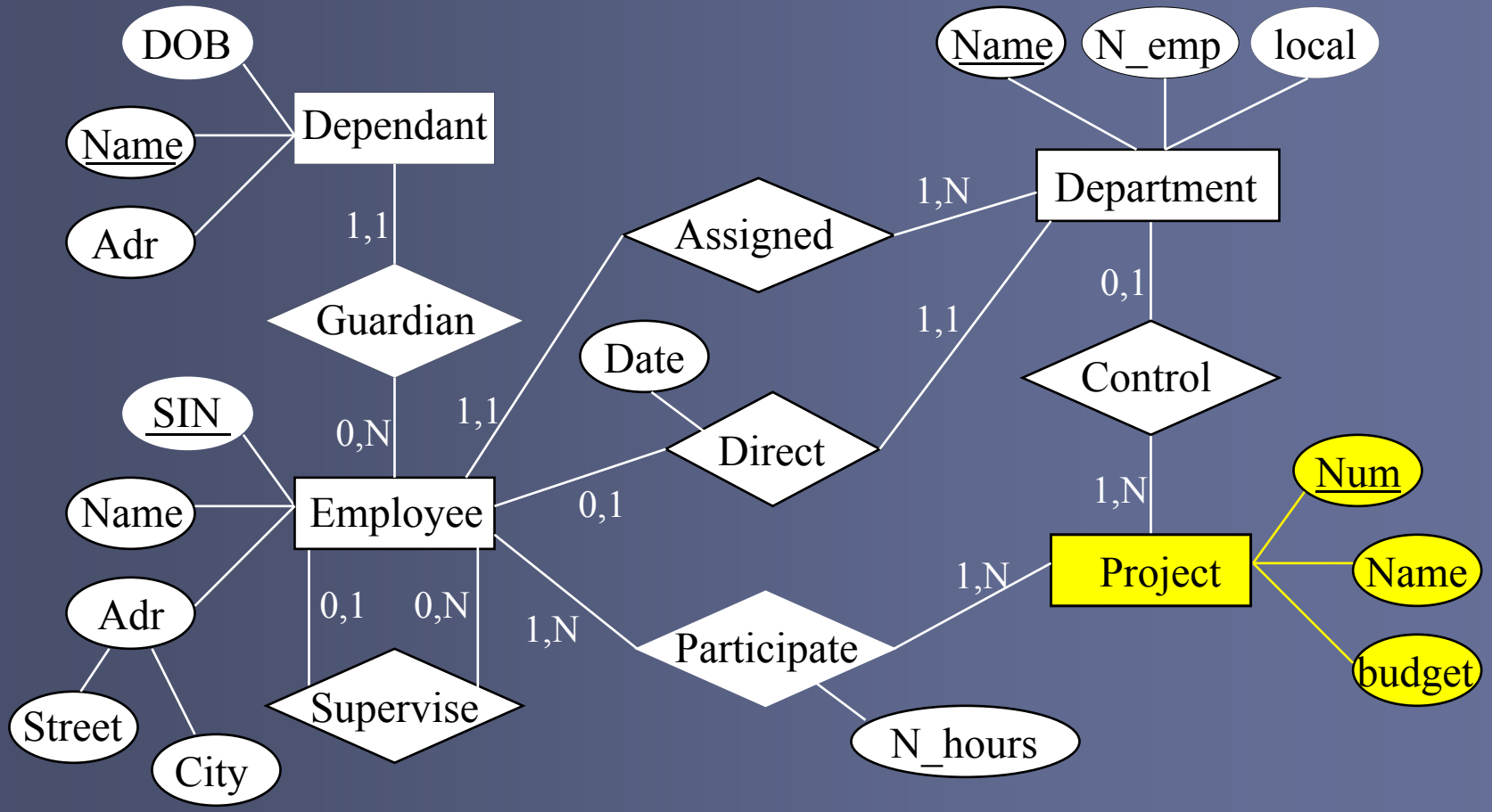
Composes(# Product_Code_CompBy, #Product_Code_CompOf …., quantity)

Database Concepts
Entity-Relationship to Relational Schema

- Convert the following ER model into relational schema:

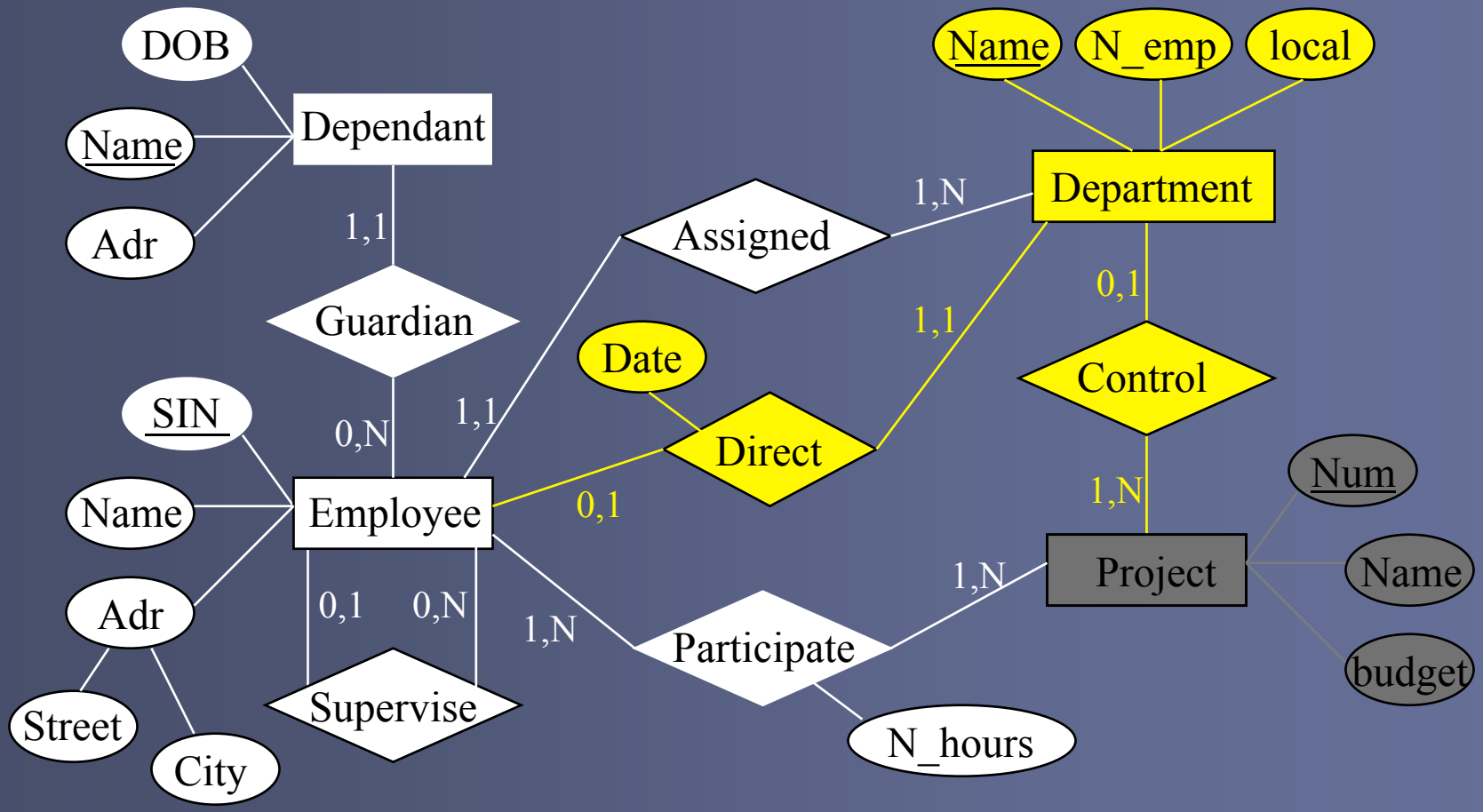


Database Concepts
Entity-Relationship to Relational Schema



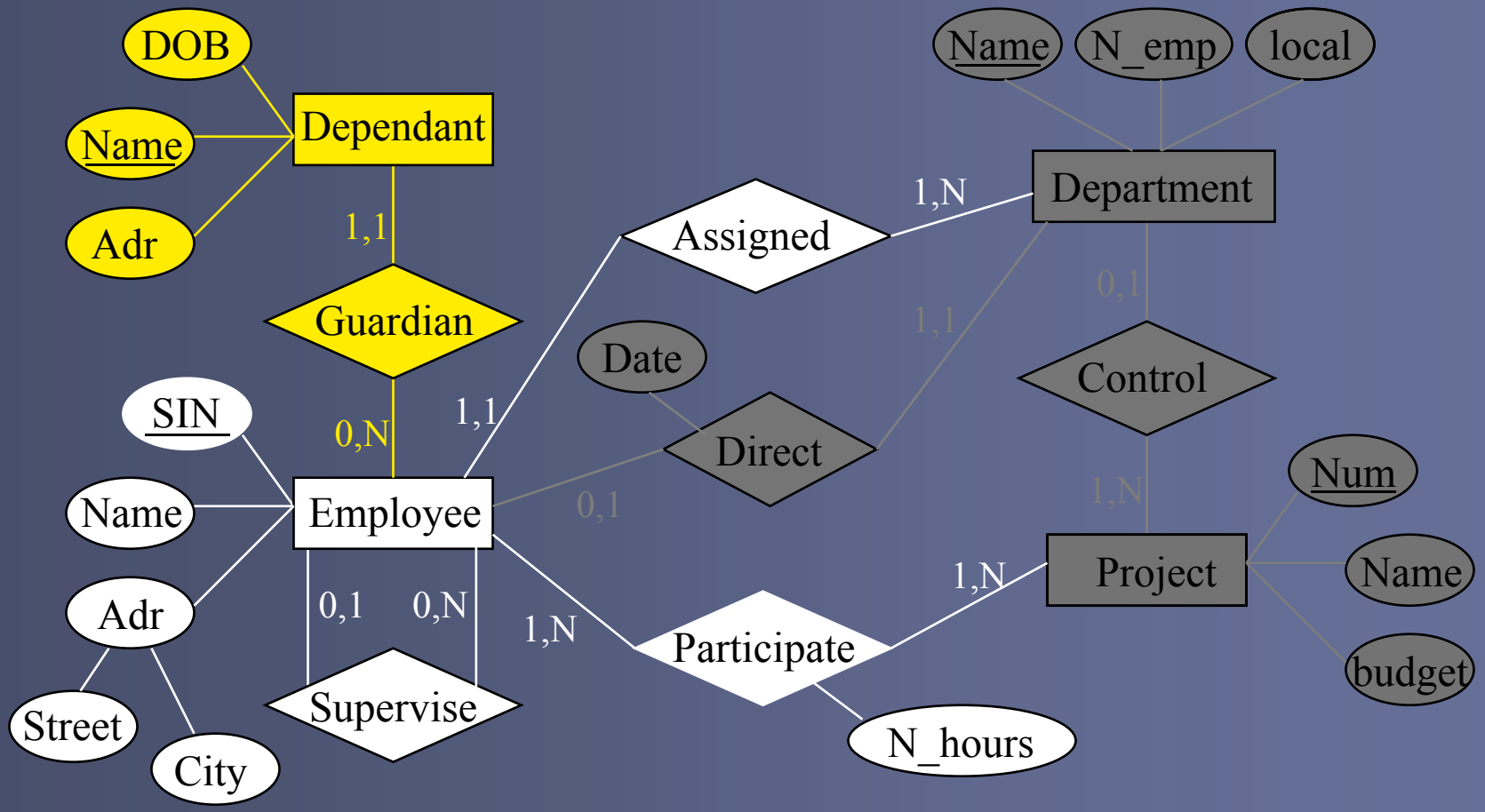
Project(Num, Name, budget)

Database Concepts
Entity-Relationship to Relational Schema



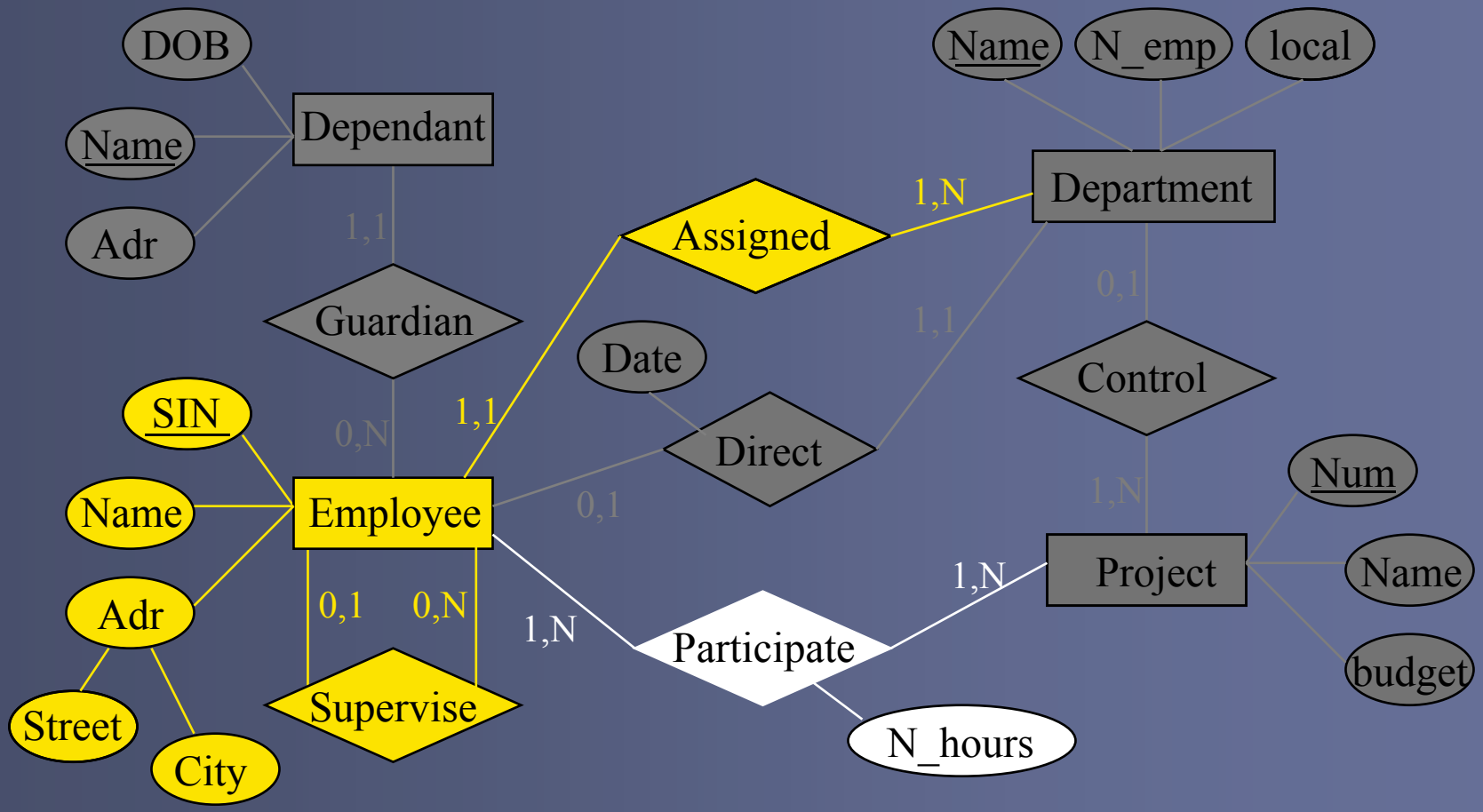
Department(Name, N_emp, local , #NumProject, #NAS_Director, Date)

Database Concepts
Entity-Relationship to Relational Schema



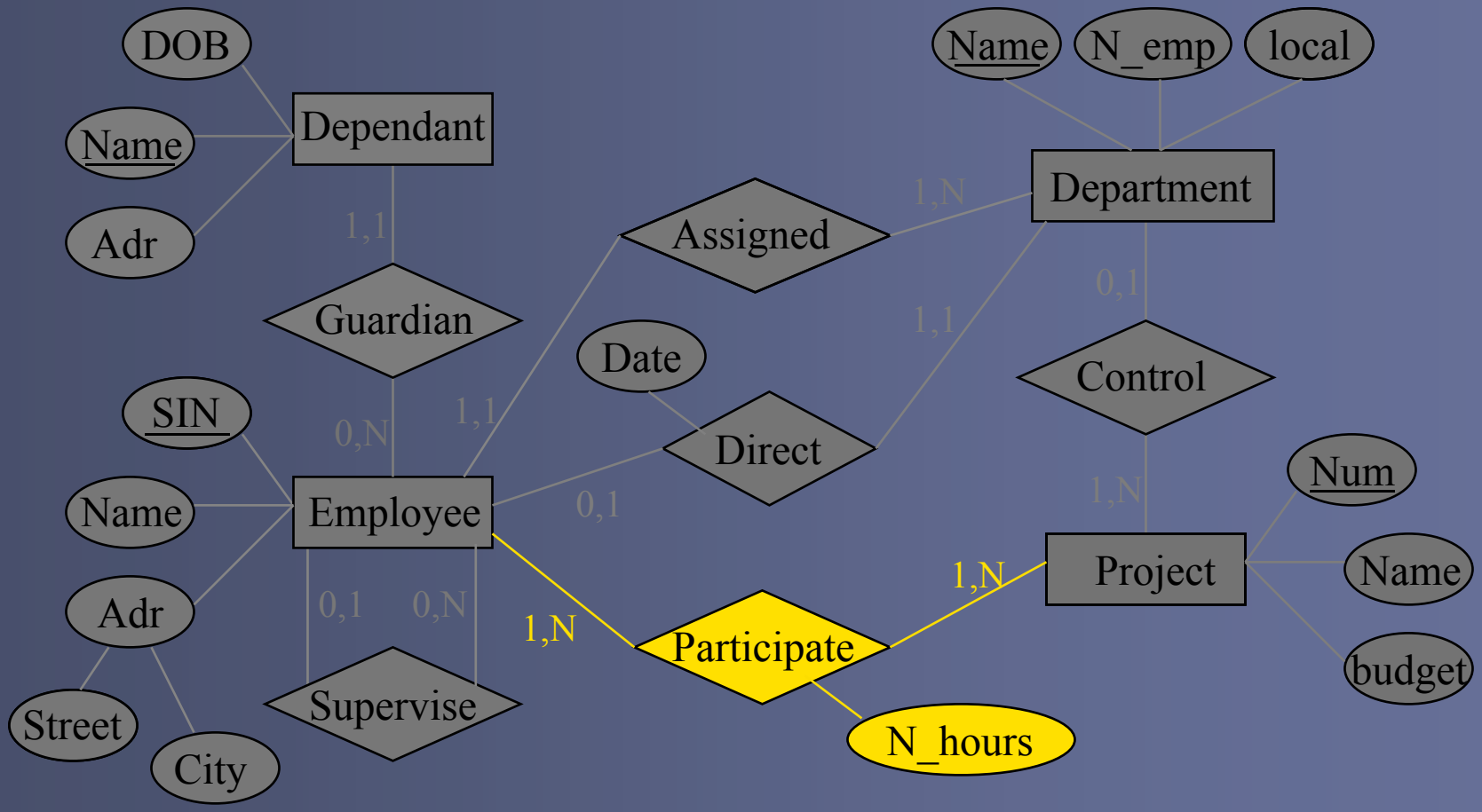
Dependant(Name, DOB, Adr, #SIN_Guarding)

Database Concepts
Entity-Relationship to Relational Schema



Employee(SIN,Adr,#SIN_Superviser,#NameAssDep)

Database Concepts
Entity-Relationship to Relational Schema



Participate(#SIN_Employee,#NumProject, N_hours)

Database Concepts
Entity-Relationship to Relational Schema

Project(Num, Name, budget)

Department(Name, N_emp, local,#NumProject,#DOB_Dependant, Date)

#NumProject references Project.Num , #DOB_Dependant references Employee.SIN

Dependant(Name, DOB, Adr, #SIN_Guardian)

#SIN_Guardian references Employee.SIN

Employee(SIN, Name, Adr, #SIN_Superviser,#NameDepAss)

#SIN_Superviser references Employee.SIN, #NameDepAss references Department.Name

Participate(#SIN_Employee, #NumProject, N_hours)

SIN_Employee references Employee.SIN, #NumProject references Project.Num