**Konsep Dasar Database**

1. **Model EER :**

Model Entity Relationship yang ditambah kemampuan semantiknya dengan beberapa konsep yang lebih kompleks. **Konsep-konsepnya** yaitu **:**

1. Subclass

Subset dari suatu entitas yang dikelompokkan dalam pengertian tertentu yang perlu disajikan secara eksplisit.

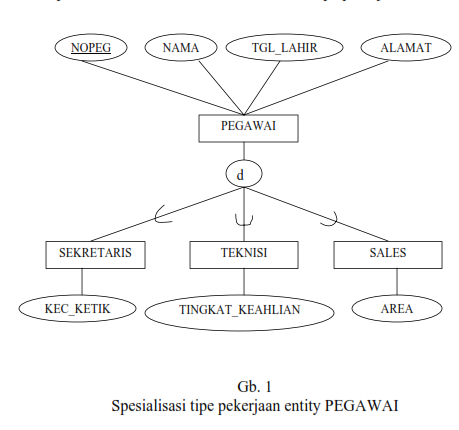
**Contoh** : Entitas PEGAWAI mempunyai beberapa *subclass*, seperti : SEKRETARIS, TEKNISI, AHLI (gb.1).

1. Superclass

Entitas yang merupakan induk dari subclass-subclassnya. Contoh : Subclass SEKRETARIS, TEKNISI, SALES mempunyai *superclass* PEGAWAI (gb.1)

1. Spezialization

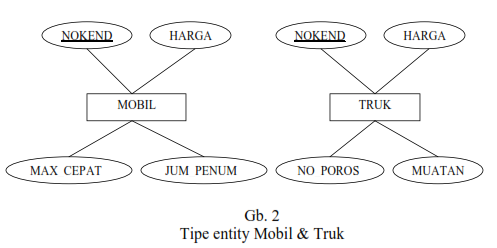
Proses pemecahan entitas menjadi subclass-subclass beserta atribut-atributnya. Terdapat beberapa jenis spesialisasi seperti Disjoint Total, Disjoint Partial, Overlapping Total dan Overlapping Partial. Contoh : Spesialisasi dari PEGAWAI berdasarkan tipe pekerjaan.

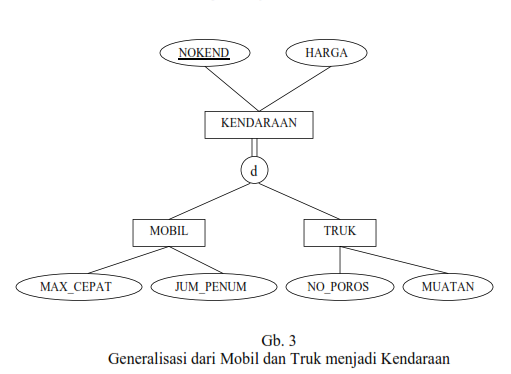


* 1. Specialization Hierarchy
  2. Spezialization Lattice

1. Generalization

Proses penggabungan subclass-subclass menjadi suatu entitas yang lebih umum. Contoh : Generalisasi dari MOBIL dan TRUK menjadi KENDARAAN





1. Attribut Inheritance
2. Shared subclass
3. Categorization

**Subclass / Superclass Relationship :**

Relationship yang menghubungkan subclass dan superclassnya.

Bila suatu entity yang merupakan anggota dari suatu subclass, maka ia juga merupakan anggota dari superclass.

Bila suatu entity merupakan anggota dari suatu superclass maka ia belum tentu merupakan anggota dari subclass.

**Attribut Inheritance :**

• Merupakan pewarisan attribut dari superclass – nya

• Entity yang merupakan anggota dari subclass mewarisi semua atribut dari Entity superclass – nya.

**Disjoint Constraint :**

• Constraint yang menerangkan bahwa subclass-subcla

ss dari spesialisasi saling disjoint, artinya entity merupakan anggota dari salah satu subclass.

• Disjoint Constraint direpresentasikan dengan lambing “d″ yang berarti *disjoint*.

Contoh : entity dari spesialisasi tipe pekerjaan dari PEGAWAI

merupakan anggota dari subclass :

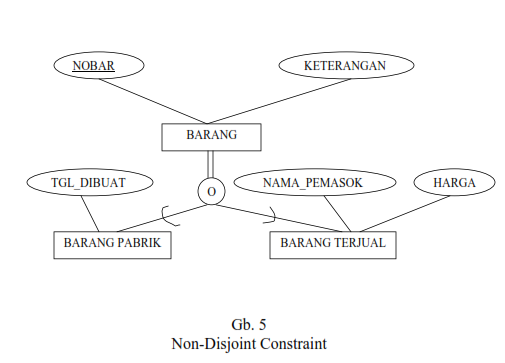
PEGAWAI TETAP atau PEGAWAI HONORER.

**Non – Disjoint Constraint :**

• Constraint yang menerangkan bahwa subclass-subclass dari spesialisasi tidak saling disjoint, artinya entity mungkin anggota lebih dari satu subclass.

• Non-Disjoint digambarkan dengan lambang ″o″ yang berarti *overlapping.*

Contoh : Entity dari spesialisasi tipe barang merupakan anggota dari subclass BARANG PABRIK dan juga anggota dari subclass BARANG TERJUAL.



**Total Specialization Constraint :**

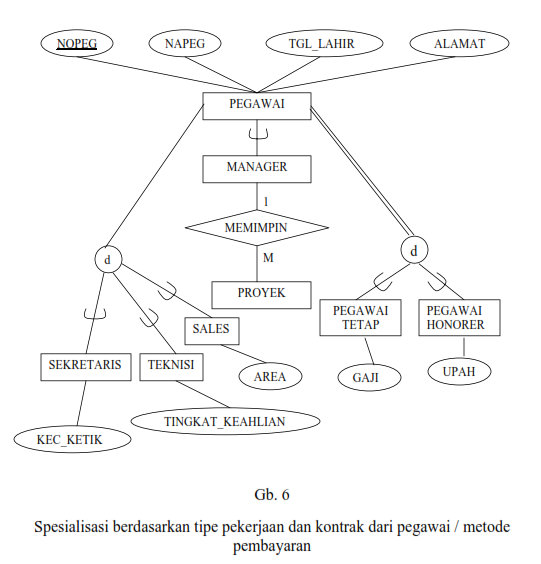
• Constraint yang menerangkan bahwa setiap entity di dalam superclass harus merupakan anggota dari salah satu subclass. Contoh : Entity PEGAWAI harus termasuk subclass dari

PEGAWAI TETAP atau PEGAWAI HONORER. (Gb. 4)

**Partial Specialization Constraint :**

• Constraint yang menerangkan bahwa setiap entity didalam superclass dapat merupakan anggota dari subclass-subclass yang didefinisikan.

Contoh : Entity dari PEGAWAI dapat merupakan anggota dari subclass MANAGER, TEKNISI atau SALES.



**Specialization Hierarchy :**

Spesialisasi bertingkat dimana setiap subclass berpartisipasi didalam satu kelas / subclass relationship

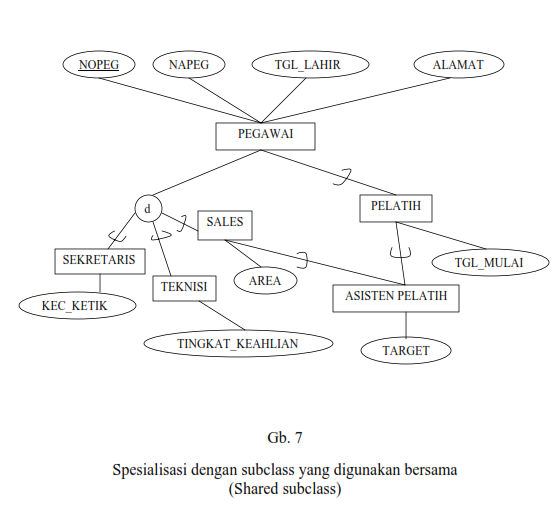
• **Specialization Lattice :**

Spesialisasi bertingkat dimana suatu subclass dapat berpartisipasi didalam beberapa class / subclass relationship.

• **Shared – Subclass :**

Subclass yang mempunyai lebih dari satu superclass. Contoh : Subclass ASISTEN PELATIH mempunyai dua

superclass yang tipenya sama yaitu SALES & PELATIH.



**Kategori :**

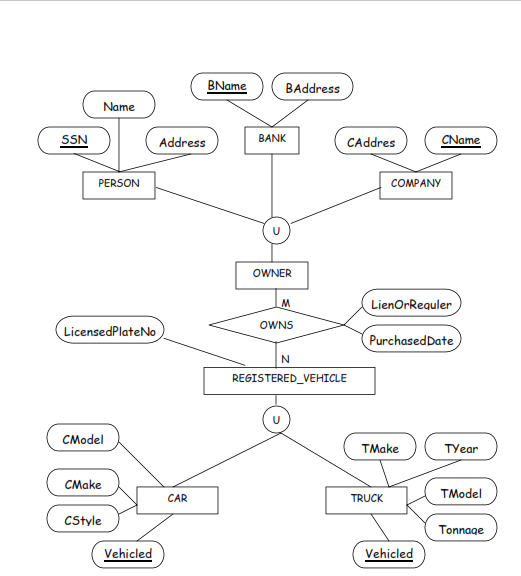
adalah kebutuhan yang timbul untuk model suatu relationship superclass/subclass tunggal dengan lebih dari satu superclass dimana superclass-superclass tersebut menggambarkan jenis entity yang berbeda. Sebuah kategori mempunyai satu atau lebih superclass yang dapat mewakili tiap tipe entity, dimana superclass/subclass lainnya bisa saja hanya mempunyai satu superclass.

Kategori dapat saja untuk seluruh relasi atau hanya sebagian saja. Contoh :

Terdapat 3 jenis entitas yaitu : PERSON, BANK dan COMPANY. Dalam suatu database REGISTERED\_VEHICLE, pemilik kendaraan (OWNER) bisa saja : perorangan, bank atau perusahaan.

Kita perlu membuat suatu class yang terdiri-dari 3 jenis entitas untuk memainkan perannya sebagai pemilik kendaraan.

Maka dibuat suatu kategori OWNER yaitu sebuah subclass dari gabungan (UNION) 3 class : COMPANY, BANK, PERSON



Keterangan :

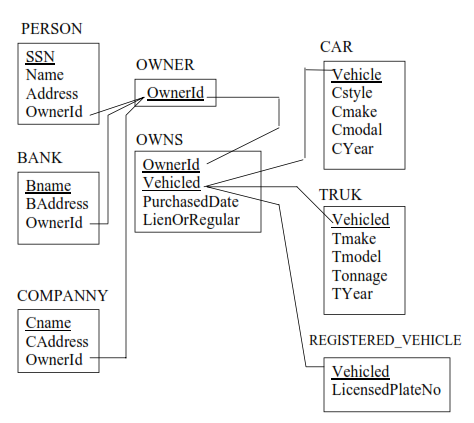
Pada gambar di atas, terdapat 2 kategori yaitu :

1. OWNER yang merupakan sebuah subclass dari gabungan PERSON, BANK dan COMPANY

2. REGISTERED\_VEHICLE yang merupakan subclass dari gabungan

CAR dab TRUCK

Pemetaan EER menjadi model relasional :



**Pemetaan (mapping) EER ke model Relasional**

Algoritma pemetaannya merupakan perluasan atau lanjutan dari pemetaan ER.

**Langkah 13** : Untuk memetakan Spesialisasi.

Attrs( R ) = atribut-atribut relasi R, sedangkan PK( R )=primary key pada R. Untuk setiap spesialisasi dengan m subclass {S1, S2, …, Sm} dan superclass C, dimana atribut C adalah {k, a1, a2, …, an}dan k=PK.

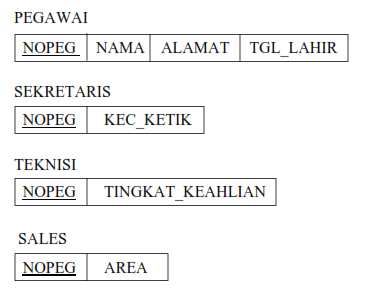
Ini dapat dipetakan dengan beberapa cara berikut.

a. Buat suatu relasi L dengan atribut-atribut Attrs(L)= {k, a1, a2, …, an} dan PK(L)=k.

Buat juga relasi Li untuk setiap subclass Si, 1 < i < m, dengan atribut

Attrs(Li) = {k} ∪ {attrs Si} dan PK(Li) = k.

Perhatikan gb. 1, mappingnya :



b. Buat relasi Li untuk setiap subclass Si, 1 < i < m dengan atribut-atribut

Attrs(Li) = {Attrs Si} ∪ {k, a1, a2, …, an} dan PK(Li) = k. Perhatikan gb. 3 , mappingnya :

MOBIL

|  |  |  |  |
| --- | --- | --- | --- |
| NOKEND | HARGA | MAX\_CEPAT | JUM\_PENUM |

TRUK

|  |  |  |  |
| --- | --- | --- | --- |
| NOKEND | HARGA | NO\_POROS | MUATAN |

c. Buat satu relasi L dengan atribut Attrs(L) = {k, a1, a2, …, an} ∪

{attributes of Si} ∪ … ∪ {attributes dari Sm} ∪ {t} dan PK(L)=k. t adalah atribut-atribut yang dimiliki subclass.

Perhatikan gb. 1, mappingnya :

PEGAWAI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOPEG | NAMA | ALAMAT | TGL\_LAHIR | KEC\_KETIK | **…** |

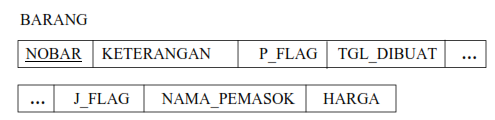
**…** TINGKAT\_KEAHLIAN AREA

d. Buat satu skema relasi L dengan Attrs(L) = {k, a1, a2, …, an} ∪

{attributes of Si} ∪ … ∪ {attributes dari Sm} ∪ { t1, t2, …, tm } dan

PK(L)=k. Setiap ti , 1< i < m adalah atribut Boolean (flag) yang

menunjukkan bukan tuple yang dimiliki oleh subclass Si . Perhatikan gb. 5, mappingnya :



**Transformasi dari EER ke Database Relasional**

Pilihan 1.

Buat relasi dari Super-Kelasnya sendiri ditambah dengan relasi-relasi sebanyak Sub-Kelasnya, dimana relasi untuk tiap Sub-Kelasnya berisi atribut dari masing-masing Sub\_Kelas ditambahkan Primary Key dari Super-Kelasnya sebagai Primary Keynya.

Pilihan 2.

Dibuat relasi-relasi sebanyak Sub-Kelasnya, dimana isi dari relasi tersebut adalah atribut-atribut dari Super-Kelas ditambahkan atribut-atribut dari Sub-Kelasnya sendiri. Primary key menggunakan Primary key Super-Kelasnya.

Disjoint

Dibuat satu relasi yaitu gabungan atribut Super-Kelasnya dengan atribut-atribut Sub-Kelasnya.

overlapping

Dibuat satu relasi yaitu gabungan atribut-atribut Super- Kelasnya dengan atribut-atribut Sub\_Kelasnya, tetapi ditambahkan Flag sebanyak Sub-Kelasnya.

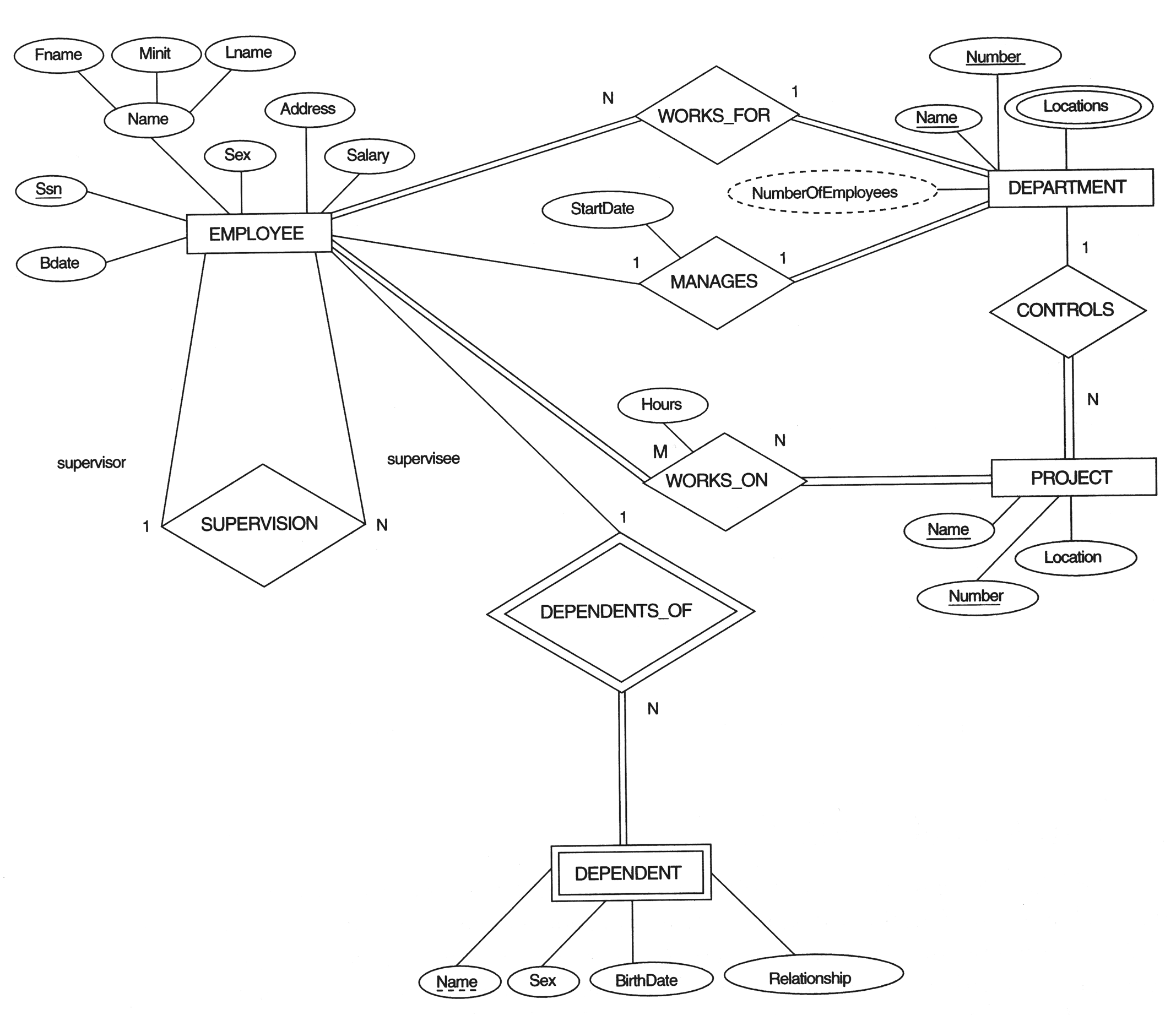
Pemetaan model ER ke skema relasional

Pemetaan diagram (e)er ke skema relasional dibagi menjadi dua:

1. Pemetaan model er ke skema relasioanal

Berikut merupakan urutan tahapan pemetaan diagram ER ke model relasional :

1. Pemetaan Regular Entity Tipes



Employee

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SSN | FName | Minit | Lname | BDate | Address | Sex | Salary |

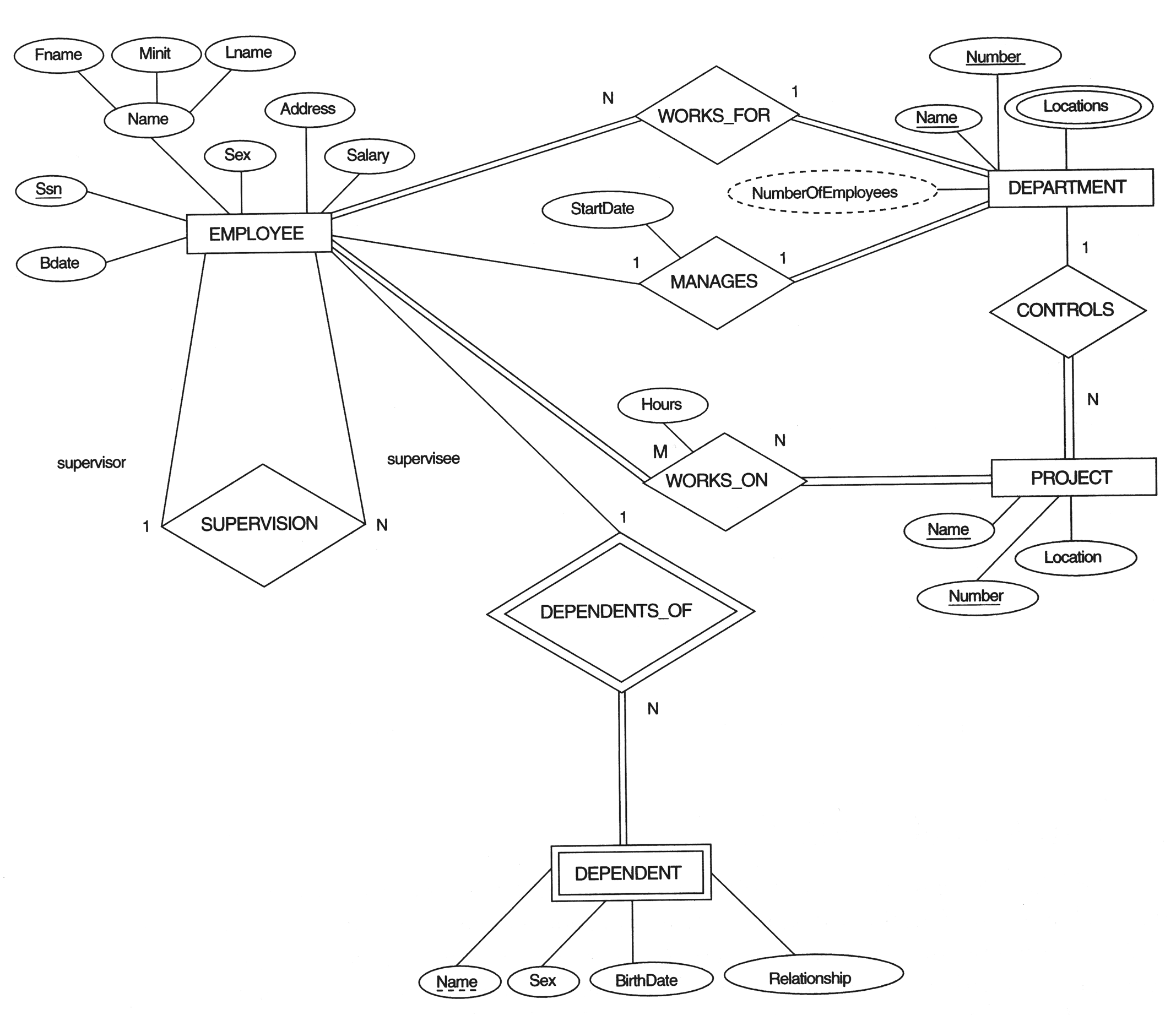
Department

|  |  |
| --- | --- |
| DName | DNumber |

Project

|  |  |  |
| --- | --- | --- |
| PName | PNumber | Plocation |

1. Pemetaan Weak Entity Types



Employee

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SSN | FName | Minit | Lname | BDate | Address | Sex | Salary |

Department

|  |  |
| --- | --- |
| DName | DNumber |

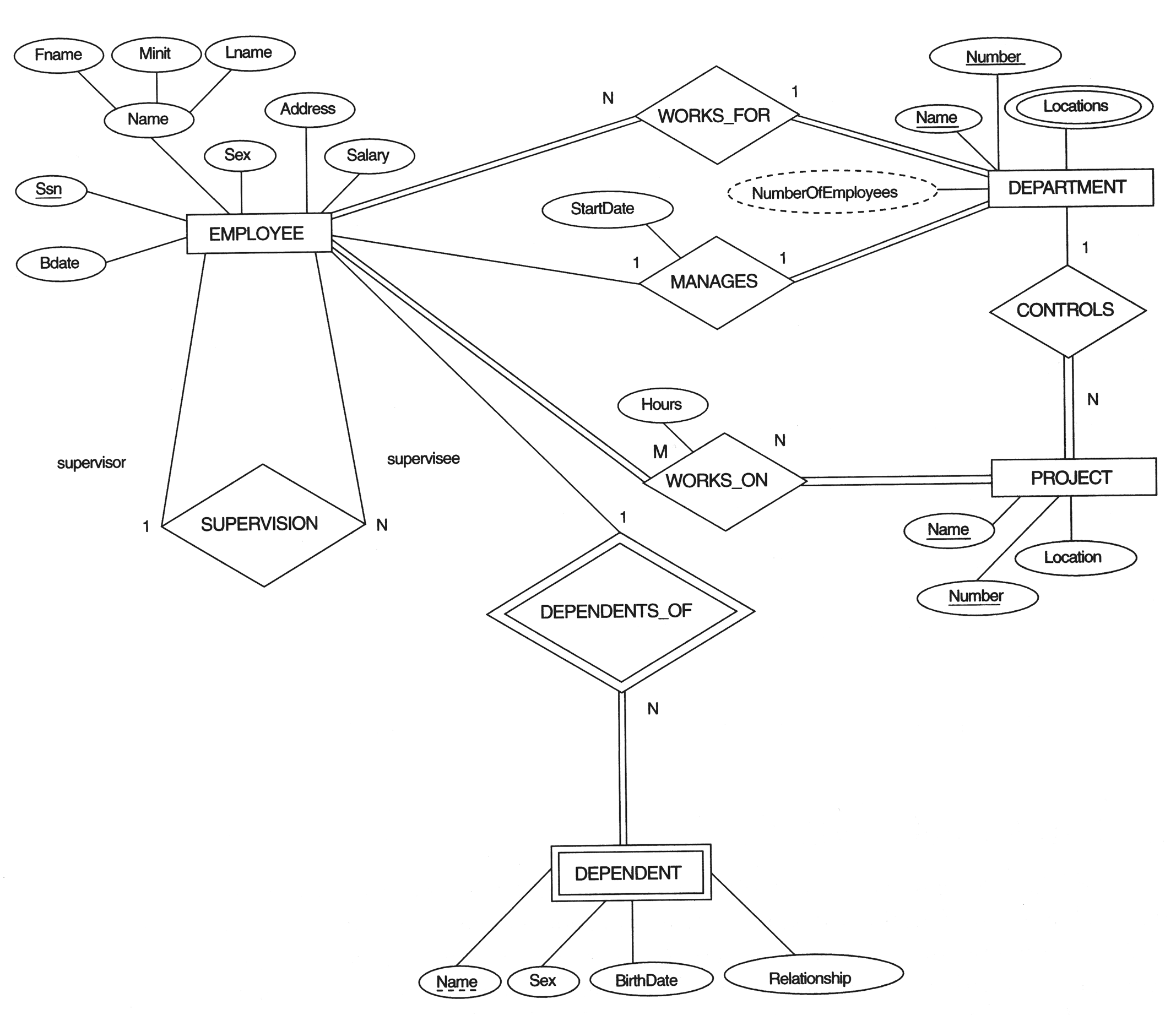
Project

|  |  |  |
| --- | --- | --- |
| PName | PNumber | Plocation |

Dependent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ESSN | Dependent\_Name | BIrdthDate | Sex | Relationship |

1. Pemetaan Binary 1:1Relationship Types



Employee

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SSN | FName | Minit | Lname | BDate | Address | Sex | Salary |

Department

|  |  |  |  |
| --- | --- | --- | --- |
| DName | DNumber | Mgr\_Ssn | Mgr\_start\_date |

Project

|  |  |  |
| --- | --- | --- |
| PName | PNumber | Plocation |

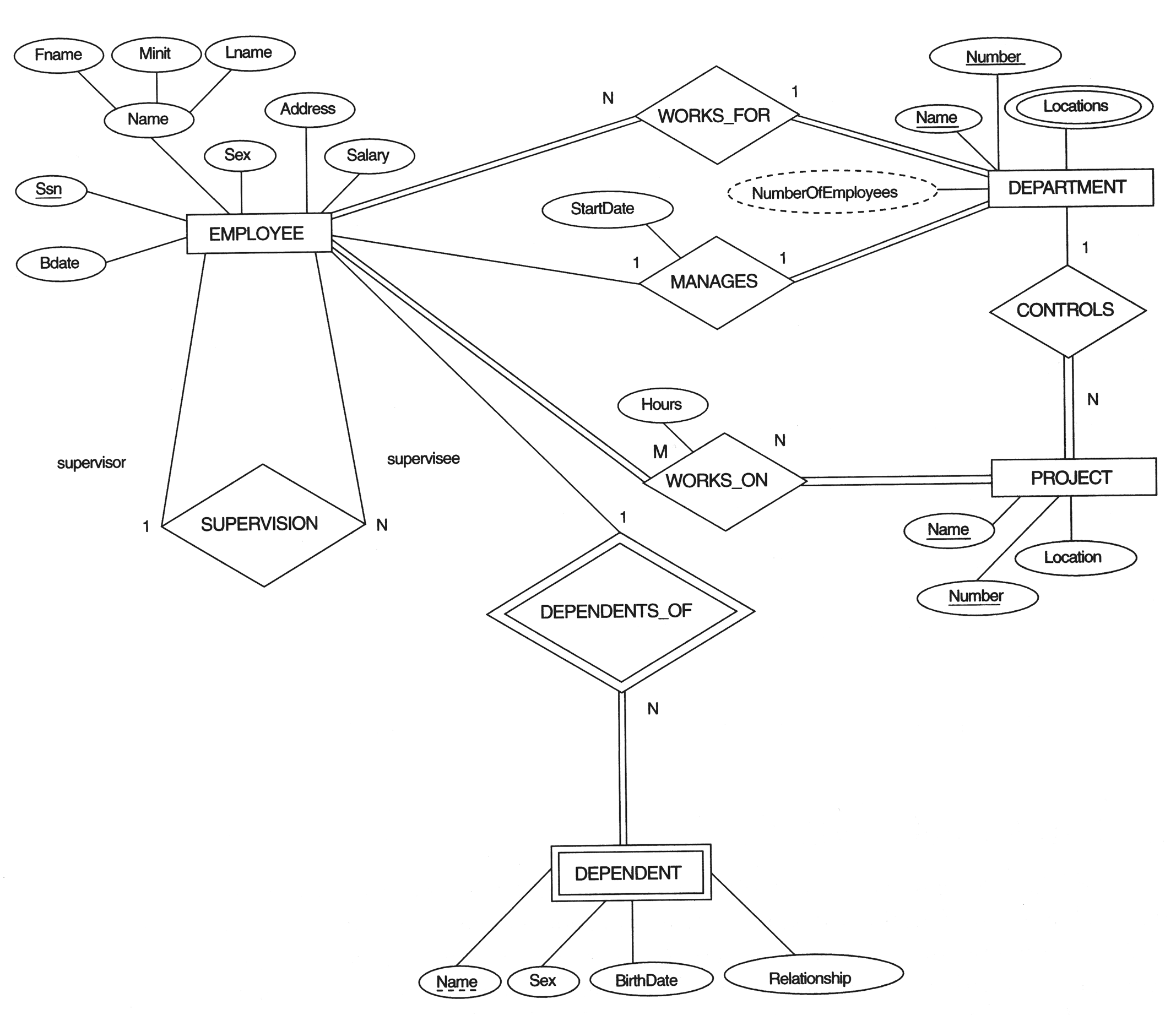
Dependent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ESSN | Dependent\_Name | BIrdthDate | Sex | Relationship |

Tiga pendekatan dalam pemetaan binary 1-1

* Pendekatan foreign key
  + Pilih salah satu relasi (missal S ), masukkan primary key dari relasi T sebagai foreign key pada S.
  + Relasi S yang dipilih sebaiknya yagn berpartisipasi total
* Opsi penggabungan relasi
  + Kedua entity types dan relationship digabungkan menjadi satu relasi.
  + Sesuai jika partisipasi kedua entity types dalam relationship tersebut bersifat total.
* Opsi cross –reference / relationship relation
  + Membuat relasi baru yang mengandung primary key dari kedua types yang terhubung dalam relationship tersebut.
  + Dapat dilakukan pada berbagai kondisi (partisipasi total maupun parsial).

1. Pemetaan Binary 1:N Relationship Types



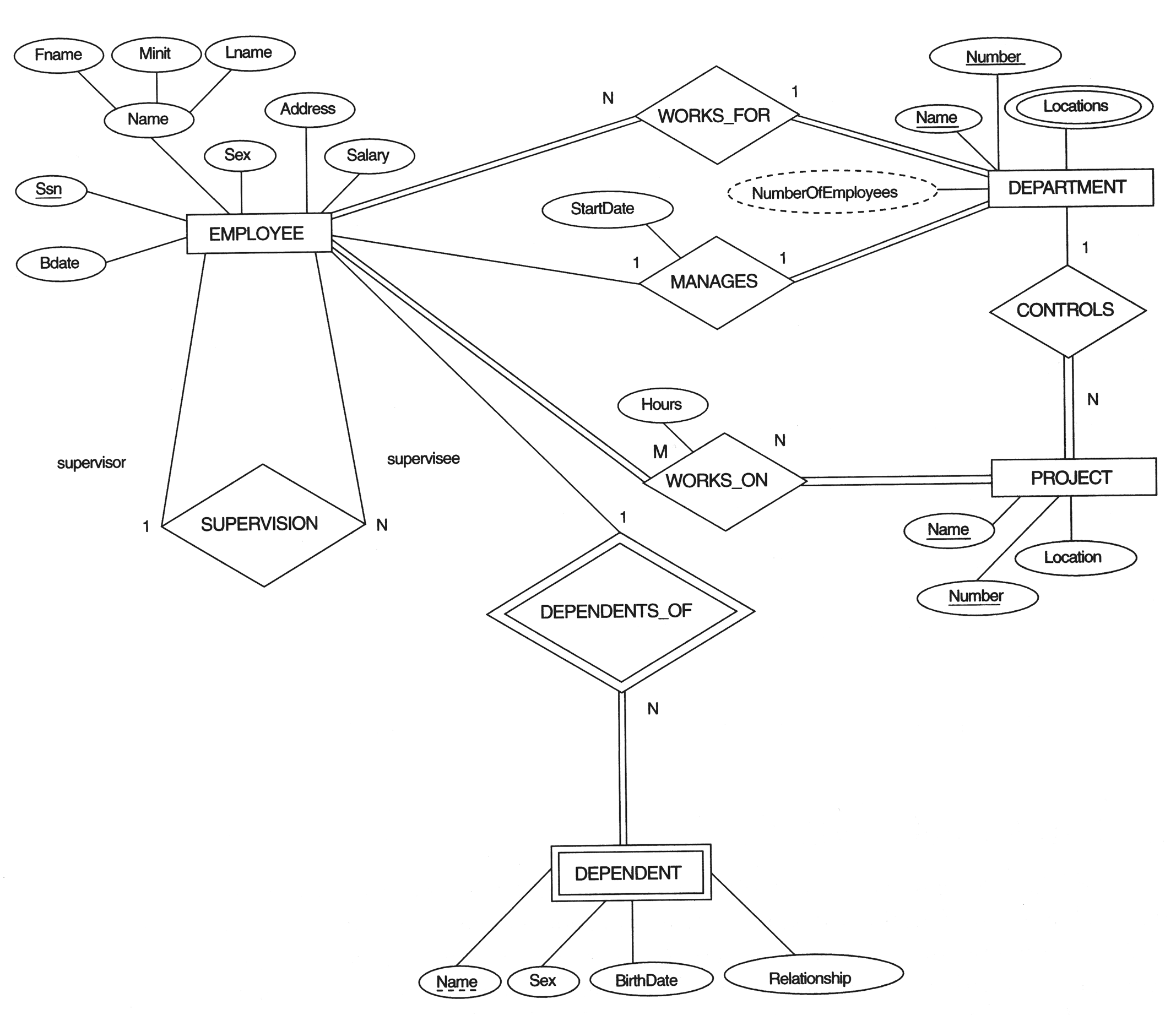
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SSN | FName | Minit | Lname | BDate | Address | Sex | Salary | Super\_Ssn | DNo |

|  |  |  |  |
| --- | --- | --- | --- |
| DName | DNumber | Mgr\_Ssn | Mgr\_start\_date |

|  |  |  |  |
| --- | --- | --- | --- |
| PName | PNumber | Plocation | DNum |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ESSN | Dependent\_Name | BIrdthDate | Sex | Relationship |

1. Pemetaan Binary M:N Relationship Types



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SSN | FName | Minit | Lname | BDate | Address | Sex | Salary | Super\_Ssn | DNo |

|  |  |  |  |
| --- | --- | --- | --- |
| DName | DNumber | Mgr\_Ssn | Mgr\_start\_date |

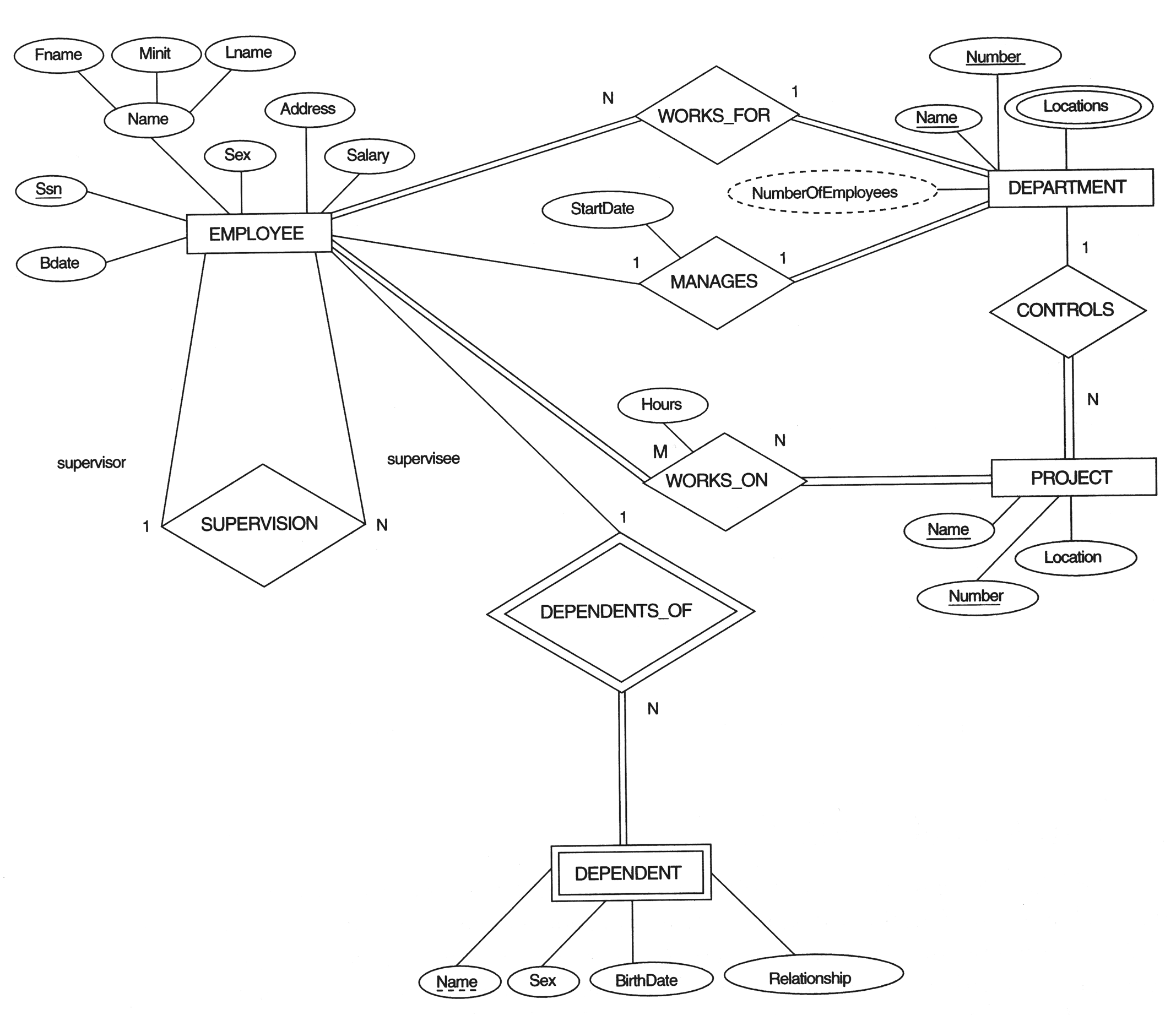
|  |  |  |  |
| --- | --- | --- | --- |
| PName | PNumber | Plocation | DNum |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ESSN | Dependent\_Name | BIrdthDate | Sex | Relationship |

Works\_On

|  |  |  |
| --- | --- | --- |
| ESsn | PNo | Hours |

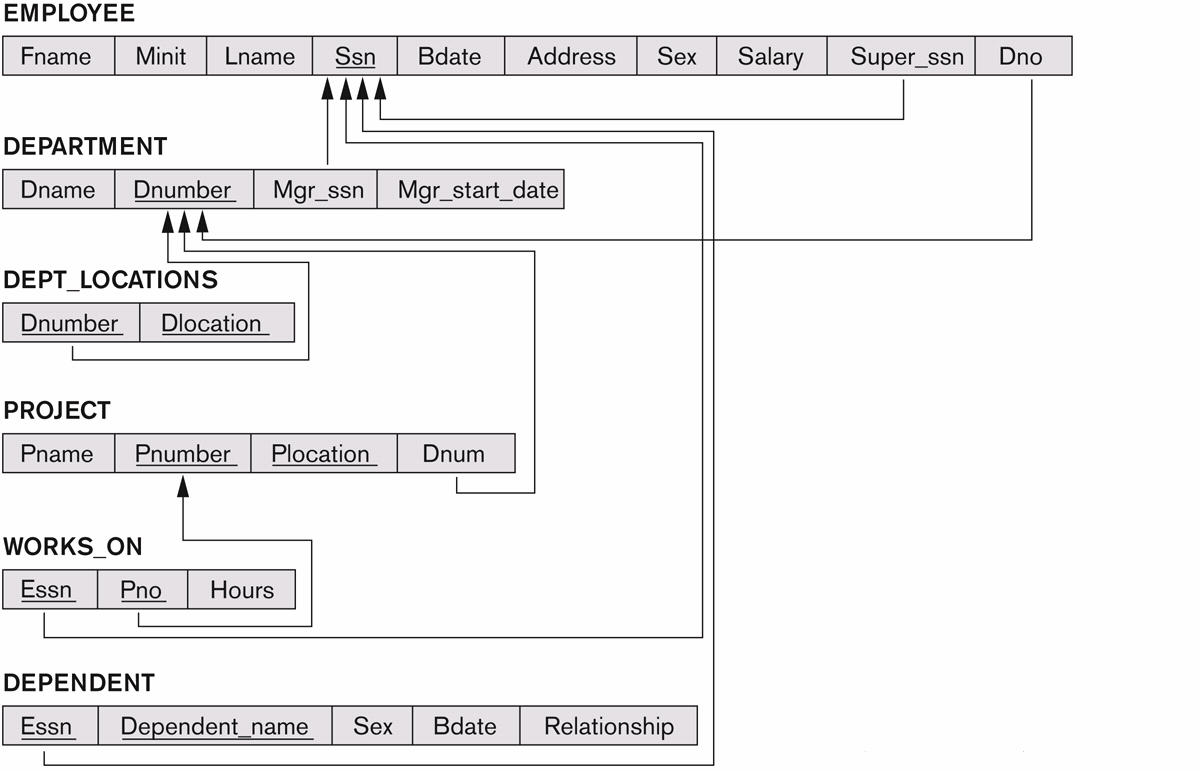
1. Pemetaan Multivalued attributes



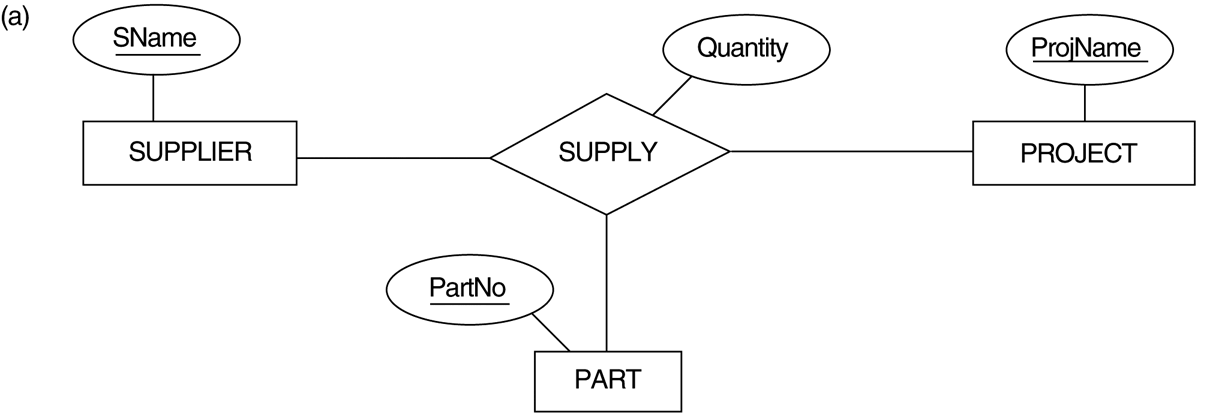
|  |  |  |  |
| --- | --- | --- | --- |
| DName | DNumber | Mgr\_Ssn | Mgr\_start\_date |

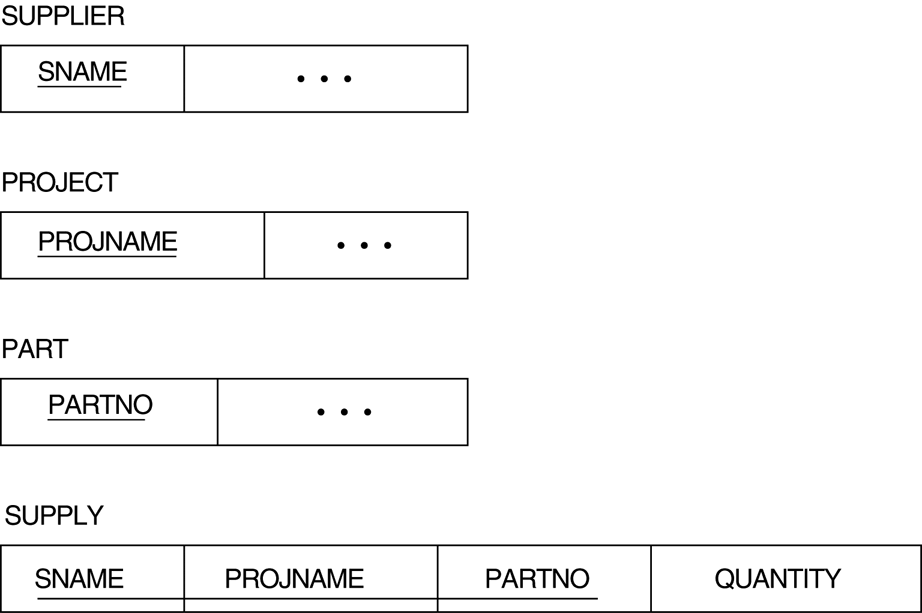
|  |  |  |  |
| --- | --- | --- | --- |
| PName | PNumber | Plocation | DNum |

Berikut hasil pemetaan diagram mER Company tersebut



1. Pemetaan N-ary (Higher Degree) Relationship Types

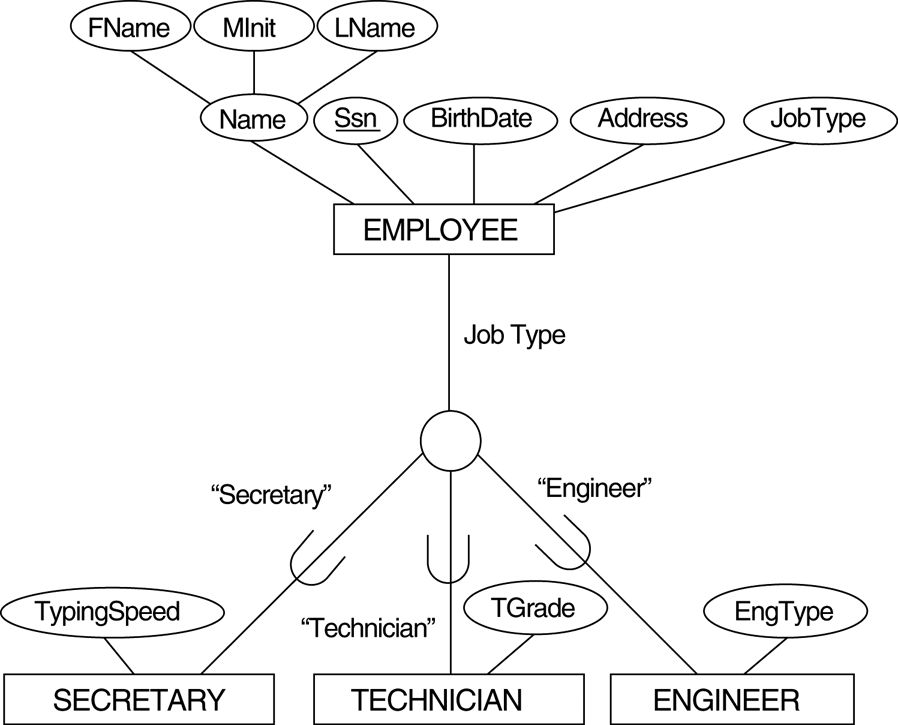


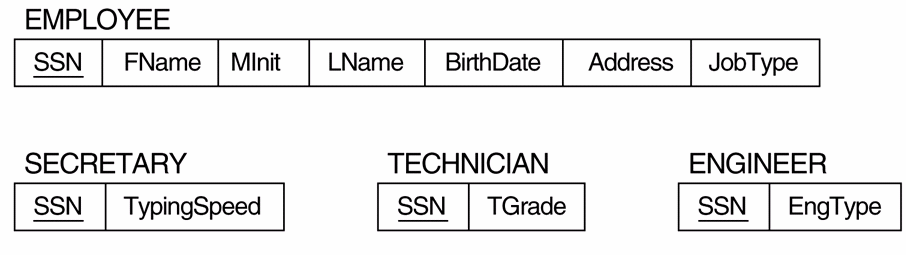


Pemetaan *Derived* *Attribute*

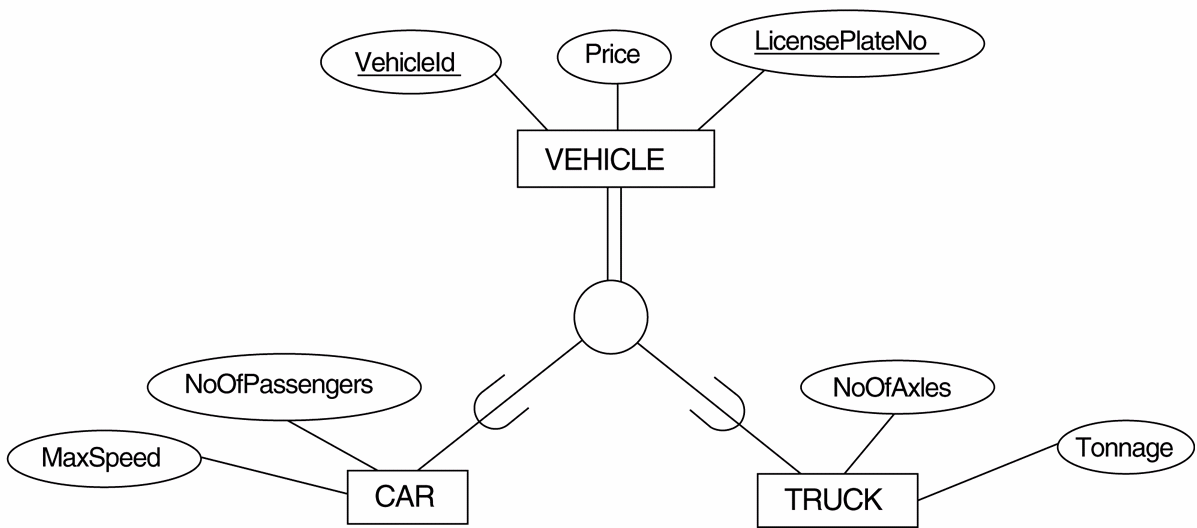
* Opsi 1: tidak disimpan pada basis data
  + Nilai dari derived attribute dihitung pada saat diperlukan (perlu waktu komputasi).
  + Menghemat storage.
  + Konsistensi antara nilai derived attribute dengan stored attribute yang berkaitan dapat dijaga.
* Opsi 2: disimpan pada basis data
  + Derived attribute diperlakukan sebagaimana simple attribute.
  + Untuk menjamin konsistensi, nilai dari derived attribute ini sebaiknya jangan dimasukkan oleh operator, namun dilakukan secara otomatis oleh sistem yang kita buat.

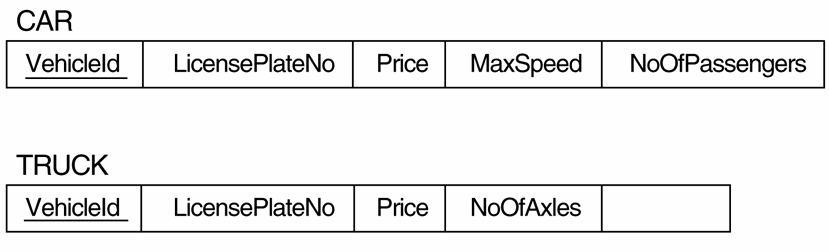
1. Pemetaan model eer ke skema relasioanal
   1. Pemetaan spesialisasi dan generalisasi
      1. Banyak relasi (*superclass* dan *subclass*)



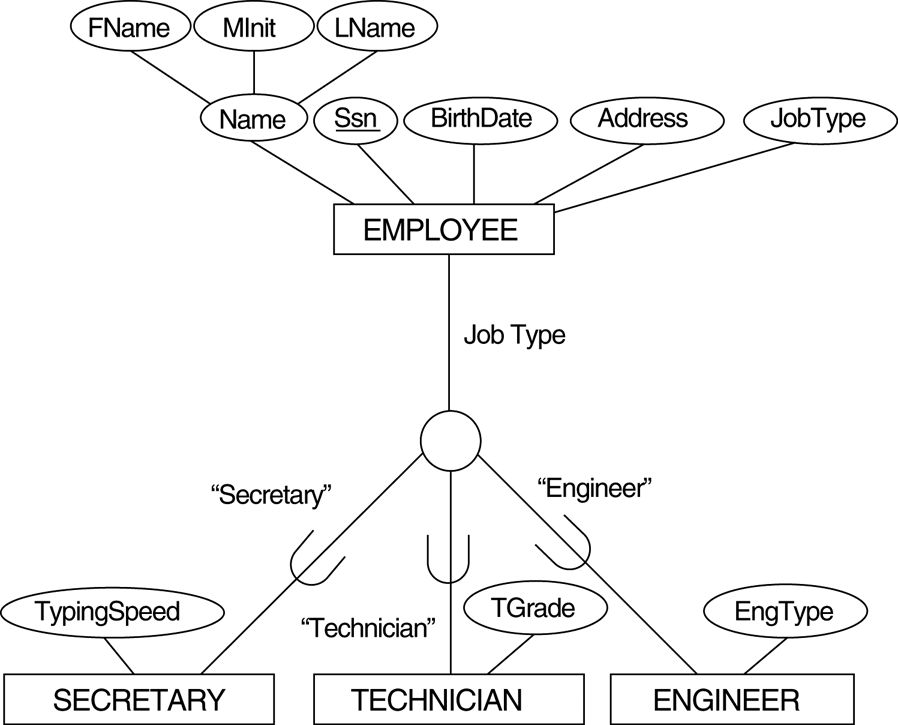


* + 1. Banyak relasi (*subclass* saja)





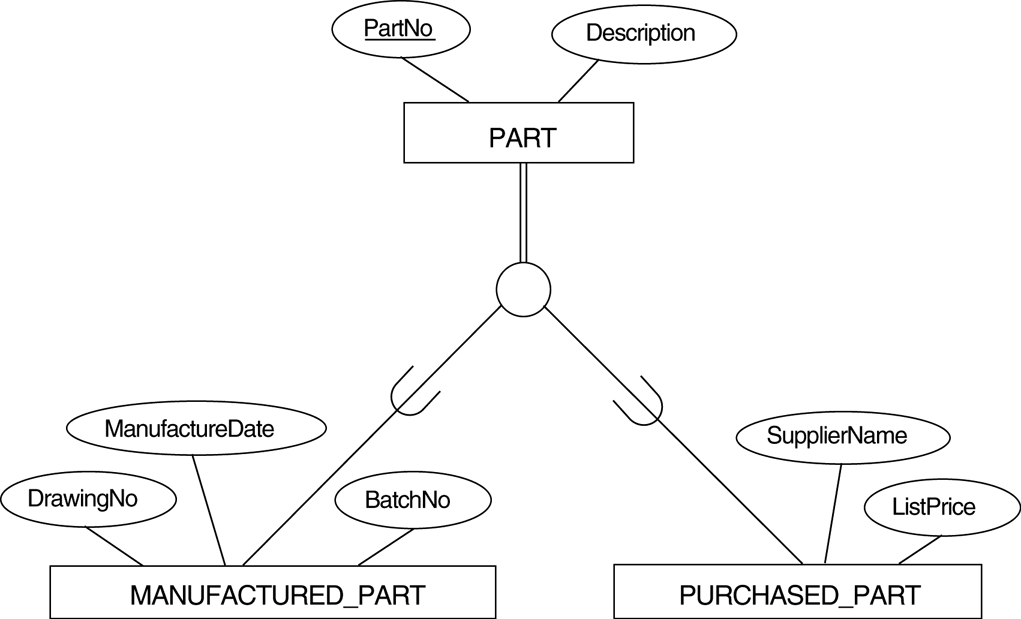
* + 1. Satu relasi dengan satu *type*



Employee

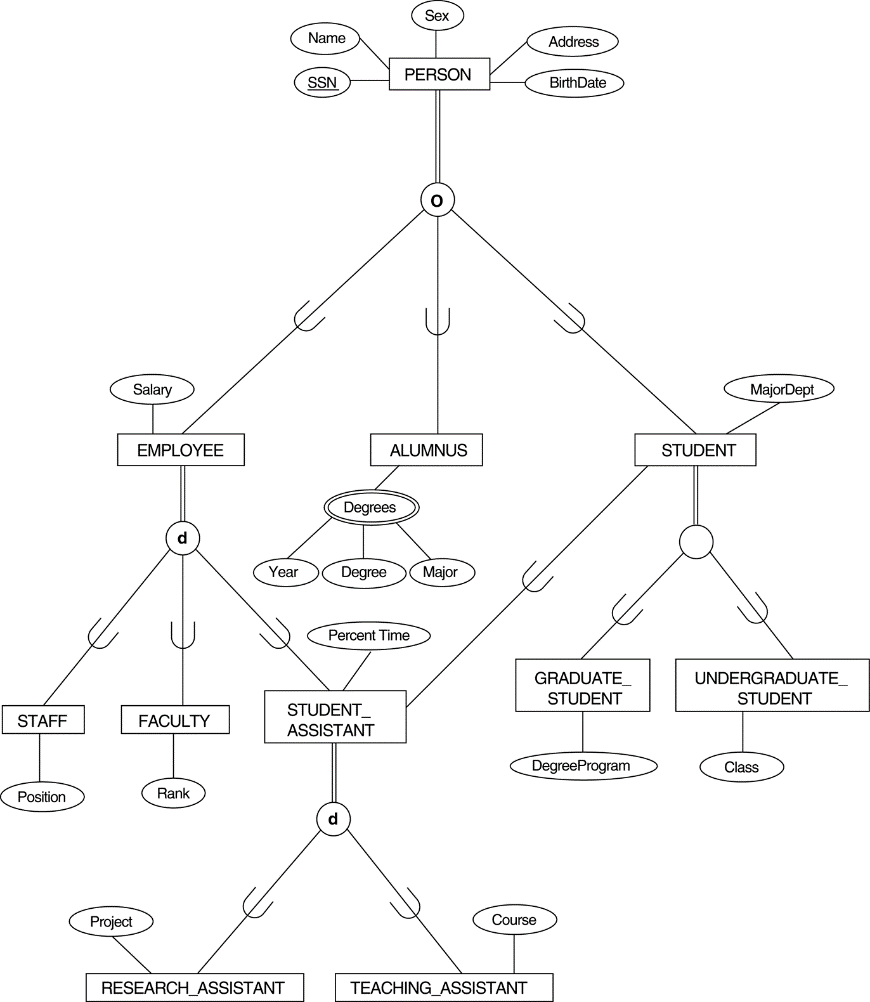


* + 1. Satu relasi dengna banyak *type attribute*



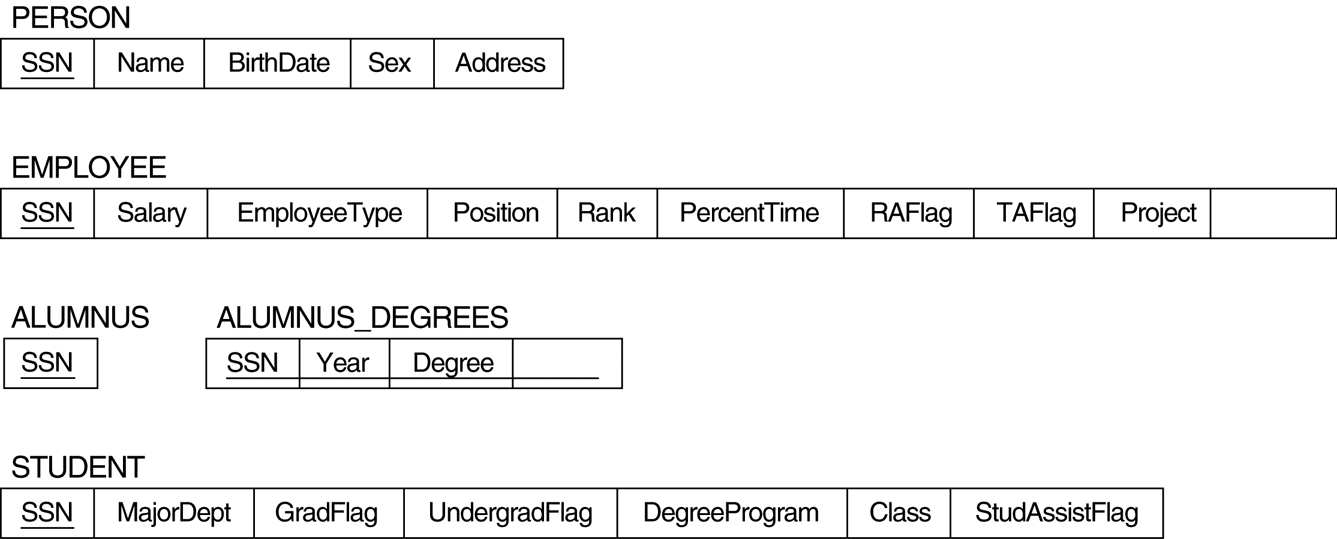


* 1. Pemetaan shared subclasses



Pemetaan shared subclasses dapat memilih opsi 1 sampai opsi 4 yan dijelaskan sebelumnya.

Pada contoh ini digunakan opsi 3 dan 4.



* 1. Pemetaan union types

