BERURUTAN

1. Map strong entity with its simple attribute(s)
2. Map weak entity
3. Map 1-1 Relationship
4. Map 1-N Relationship
5. Map N-M Relationship
6. Map N-ary Relationship
7. Map Multivalued attr
8. Map specialization & generalization
9. Map Union

KALAU ADA YG NGEDOBEL, DITIMPA AJA (misal A jadi FK hasil dri step 1 sama 5, ditulisnya satu aja ngga dobel)

1. STRONG ENTITY

- set salah satu attr jadi PK (biasanya dari key attr), kalau atr bertipe composite, gabungan setiap componennya jadi PK

2. WEAK ENTITY

- sama kaya entity, TAPI masukin PK dari entity yg jadi abangnya weak entity jadi FK, trus PK dari weak entity ini gabungan dari FK dan partial key

3. 1-1 Relationship

- Kalo ada total sama partial participation, PK dari si partial jadi FK di si total. Terus, properti (atribut dari relationship) juga masuk ke si total

- kalo dua2nya total participation(trus juga 1-1), harusnya ini gaada (mungkin keasalahan dari skema) jadinya sbenerny ini satu entity yg sama. Jadi, tabelnya digabung, mau E1 gabung ke E2 atau sebaliknya monggo. Nnti pilih salah satu jadi PK nya (jangan 22nya)

- Kalo dua2nya partial participation, bikin relation tabel baru, isinya semua attr dri kedua entity plus properti juga (namanya cross reference). Tiap PK dari kedua entity tadi jadi PK table ini (digabung jadi PK)

4. 1-N Relationship

- gadiliat tipe participationnya. PK dari si entt-1 bakal jadi FK dari si entt-N, properti juga diadain ke si entt-N

- berlaku juga untuk recursive relationship (bakal ada dua field yg sama, biasanya dikasih kode biar beda contoh: PK Ssn, FK SuperSsn)

5. M-N Relationship

- Buat relation table baru, PK nya gabungan dari dua PK sebelumnya trus properti juga jangan ketinggalan

6. Multivalued attr

- buat relation table baru, isinya PK dri si entitiynya jadi FK trus si attr sendiri jadi attr, PK nya gabungan FK sama si attrnya sendiri tadi

7. N-ary Relationship

- bikin relation table baru, PK tabel ni gabungan dari semua PK dari entity yg nyambung ke relationship. Janlup properti juga masukin

8. superclass subclass

- tipe A: tiap subclass bikin relation tabel baru, isinya PK dari si superclass plus attr yg dia sndiri punya (cocok buat apa aja)

- tipe B: tiap subclass itu hasil dari merge superclass sama dirinya sndiri, dimana nnti si superclassnya bakal dihapus (cocok buat generalization)

- tipe C: digabung jadi satu relation table termasuk attr yg spesifik di subclass2nya, bakal ada field yg **ngebedain** (discriminating). Nnti attr spesifik tadi tergantung field ntu, kalo bukan tipenya brrti nnti jadi null

Contoh: blabla blabla JobType TypeSpeed TechGrade

Blaa blaa Secretary 60 null

Bla bla Technician null 10

(jelekny nnti bakal banyak null, cocok buat total participation)

- tipe D: mirip tipe C tpi bukan discriminating field, tpi pake flag (tipenya bool)

Contoh: blabla blabla SecrFlag TypeSpeed TechFlag TechGrade

Blaa blaa 1 60 0 null

Bla bla 0 null 1 10

(banyak null juga, cocok buat overlapping subclass)

9. Union

- surrogate key: key buatan dari parentclass buat jadi FK si subclass nya

-