
Databasesystemer, forår 2005
IT Universitetet i København

Forelæsning 4: Mere om E-R modellering

24. februar 2005

Forelæser: Rasmus Pagh

— Forelæsningen i dag —

- Fortsættelse af E-R model: Attributtyper, identifiere, svage entiteter, associative entiteter. (Slides fra MDM)
- At gå fra en E-R model til et relationelt databaseskema.
- Udvidede E-R (EER) konstruktioner: Super/subtyper, clustering. (Slides fra MDM)

At gå fra en E-R model til et relationelt databaseskema.

— Alternativ forklaring af konvertering —

MDM beskriver en metode for E-R til relation konvertering med 7 skridt, herunder hvordan man håndterer 10 forskellige tilfælde.

På de følgende slides gennemgås en alternativ måde at se metoden på, hvor de mange skridt og tilfælde er slået sammen til færre, mere generelle tilfælde.

Vi springer i denne omgang over definition af “nøgler” og “fremmednøgler” i relationerne.

— Punkt 1: Skab identifiere for alle entiteter —

- Stærke entitetstyper har allerede identifiere, så dér skal intet gøres.
- For svage entitetstyper skabes en identifier bestående af:
 - Entitetstypens egen partielle identifier.
 - Identifier(s) fra “identifying owner(s)”.

— Punkt 2: Konvertér entitetstyper —

For hver entitetstype laves en relation, der har flg. attributmængde:

- Alle attributter fra entitetstypens identifier.
- Alle simple attributter fra entitetstypen.
- Alle komponentattributter i sammensatte attributter.

For hver multiværdiattribut A laves en relation med flg. attributter:

- Entitetstypens identifier
- A, eller A's komponenter hvis A er en sammensat attribut.

— Punkt 3: Konvertér relationship types —

Generelt set kan et relationship (herunder en associative entiteter) oversættes til en relation med flg. attributter:

- En identifikator for hver involveret entitetstype.
- Hvis en entitetstype er med mere end en gang (f.eks. i et unært relationship) skal der være tilsvarende flere kopier af dens identifikator attribut(ter).
- Eventuelle egne attributter for relationship'et.

Bemærk at det kan være nødvendigt at omnavngive attributter, hvis det samme navn forekommer flere gange.

Ovenstående benyttes hvis der **ikke** er tale om et 1:1 eller 1:M relationship.

— Punkt 3.b: 1:1 og 1:M relationships —

Hvis der er tale om et 1:1 eller 1:M relationship, er det ikke nødvendigt at skabe en relation.

Tilføj i stedet attributter til den ene af de to relationer, der er skabt ud fra de indgående entitetstyper – en identifier for den anden entitetstype.

Hvis der er tale om et 1:M relationship skal attributterne placeres ved entitetstypen på “mange” siden.

— Kommentarer om CASE værktøjer —

Mange “CASE” værktøjer til tegning af E-R diagrammer kan også automatisk transformere diagrammet til relationskemaer.

Grunde til, at det er vigtigt at vide, hvordan det går til:

- Mange CASE værktøjer understøtter ikke mere avancerede modelleringsnotationer, så der er tilfælde hvor manuel konvertering er nødvendig.
- Et værktøj kan lave vilkårlige beslutninger i tilfælde hvor en speciel beslutning er at foretrække (f.eks. 1:1 relationships).
- Det kan være en god idé at tegne E-R diagrammet således at konverteringen giver det ønskede relationskema, f.eks. ét der er normaliseret.

— Konvertering af supertype-subtype relationer .

En supertype-subtype relation er en særlig slags 1:1 relationship.

Derfor er udmøntningen i relationskemaerne den samme, som for et sådant relationship:

I subtypens relation inkluderes supertypens identifier.

— **Vigtigste ting fra sidste gang og denne gang -**

- Business rules (to be continued)
- E-R begreber: Entitetstype, relationship type, stærke og svage entitetstyper, attribut, grad af relationship, cardinality constraints.
- EER begreber: Clustering, supertype/subtype relation.
- Grafisk repræsentation af E-R diagrammer og EER diagrammer, særligt afbildning af ovenstående begreber.
- Forståelse af konvertering af E-R diagram til relationskemaer.

Næste gang

Constraints/business rules i databasen (dvs. egenskaber ved data som DBMSen opretholder):

- Referentiel integritet.
- CHECK constraints
- Triggers.

Gæst: Helen Lunau fra ATP vil fortælle om sine erfaringer med “business rules i DBMSer”.