Dataforståelse vinter 2022

Minimere

1. **Filmproduktion**
2. Problem

Lave en database til at støtte filmproduktion.

* Identificere entiteter og attributter og byg en database.

1. ER

Stærke entiteter: genre (genre\_id, navn, beskrivelse), film (film\_id, film\_titel, udgiveleseår), sprog (sprog\_id, sprog, land), skuespiller (skusp\_id, navn, fødselsår, kvalifikationer, billed).

Svage entiteter: rolle (rolle\_id, navn, beskrivelse)

1. Relation

Diagrammet fortolkes således, en film har et sprog, en genre har en eller flere film, en film kan have en eller flere roller og en film kan have en eller flere skuespiller. En skuespiller kan have en eller flere roller.

1. Database

Her fyldes databasen med nødvendige data og der laves nøgle udtræk for at illustrere funktionalitet.

1. Film id og titel for alle film der ikke tilhører nogle genrer (join mellem film og genre)
2. Skuespiller som har spillet i drama (join mellem genre, film og skuespiller)
3. Antal film for hver genre (join mellem film og genre og en gruppering/ Count)

\*\*write queries.

\*\*prepare presentation (problem, approach, solution, challenges).

1. **Bibliotek**
2. Problem

Lave en database til at støtte bibliotek

- identificere entiteter og attributter og byg en database.

b. ER

Entiteter: bog (bog\_id, titel, udgivelsesår), forfattere (forfatter\_id, navn, fødselsdag, dødsdato), genre (genre\_id, navn), emner (emne\_id, navn), forlag (forlag\_id, navn)

1. Relation

På diagrammet kan det ses, en eller flere bøger kan omhandle et emne, en eller flere bøg tilhører en genre, et forlag har udgivet en eller flere bøger og at en bog kan være skrevet af en eller flere forfatter.

1. Database

Data indsættes og følgende problemer løses (min. 5 linjer i hver tabel og 15 i bog tabellen).

1. Alle bøger sorteret efter forfatters navn.
2. Bøger med mere end en forfatter
3. Bøger skrevet af døde forfatter
4. **Akademi**
5. Problem

Lave en database til at støtte akademiet.

* Identificere entiteter, attributter og byg en database.

1. ER

Entiteter: Studerende (registreringsnummer, navn, adresse), undervisere (underviser\_id, navn, kvalifikation), afdelinger (afdeling\_id, navn, afdelingchef\_navn) og kurser (kur\_kode, navn, semester, forudsætninger).

* En studerende skal tilmeldes i hvilket som helst afdeling
* En afdeling kan have mange studerende
* Underviseren kan tildeles en eller flere kurser.
* Der skal også registreres karakter og om en studerende er bestået eller ej under entitet overblik.

Antagelser: Uden afdeling er der ingen kurser, og uden kurser er ingen undervisere eller studerende. Dvs. at afdeling, studerede og kurser er stærke entiteter, og underviser er en svag entitet. Afdeling chef er ikke en ikke en underviser.

1. Relation

Som det kan ses på er diagrammet, kan en afdeling tilbyde en eller flere kurser, en eller flere kurser har en underviser, og et kursus kan have en eller flere studerende og en studerende kan deltage i en eller flere kurser.

1. Database

Der indsættes og følgende udtrække bygges.

1. Antal studerende per afdeling (joine fra afdeling til studerende og tal studerende)
2. Kurser studerende kan tilmelde sig til på anden semester med bestået forudsætninger.
3. Undervisere som underviser i flere afdelinger
4. **Fodbold**
5. Problem

Lave en database til FIFA WORLD CUP 2022

- identificere entiteter, attributter og byg en database

b. ER

entiteter: hold (nr, navn, land, trainer, gruppe), spiller (nr, navn), kamp (nr, dato, stadium, holdA, holdB), mål (nr, Amål, Bmål, minute, spiller), point (er et udtrække af spiller, mål og hold)

1. Relation

På diagrammet kan det ses at, et hold har 1 eller flere spiller, et hold kan spille 1 eller flere kampe, en eller flere kampe kan spilles på et stadium, en kamp kan have 0 eller flere mål og en spiller kan have 0 eller mange mål. Overveje at lave må om til et associative entity.

d. Database

1. **Taxa**
2. Problem

Lave en database der støtter taxabranchen.

- del tabellen op i logiske entiteter. Tabellen består af 10 variabler. Planen er at dele dem op i mindre tabeller.

b. ER

entiteter: vogn (vogn\_nr), chauffør(id, navn), adresse (nr, vej, hus\_nr ), turtype(type\_nr, kontant, kredit kort), tur (tur\_nr, dato, varighed (start, slut), distance (ad1, ad2), konto, beløb, status)

1. Relation¨

På diagrammet kan det ses at en vogn har en chauffør, en vogn kan have flere tur, en tur kan have en type, flere adresser og en betaling. Betaling er en svag entitet som er afhængig af turen. Det samme kan siges om turen men i denne kontekst er det en stærk entitet.

d. Database

1. **Webshop**
2. Problem

Lave en database til en webshop.

- identificere entiteter og danne en relationsmodel.

- forklar db arkitektur.

- forbedre

b. ER

Baseret på kundens forslag.

entiteter: customers (id, name, mobile, email, address, points), products (id, name, quantityInStock, unitPrice), Shippers (id, name), orders (id, custId, orderdate, status, shippingDate, shipperId), shoppingCart (productID, quantity, price).

Med forbedringer.

Ud fra DBMS-principperne, have databasen følgende entiteter.

Kunder (samme variabler), produkter (antal på lager fjernes), ordrer (samme), shipper (pris/km (for pakker under 20 kg)), lager(id, pId, antal), og shoppingcart (key, price, qnt).

1. Relation

Diagrammet baseret på det oprindelig forslag kan fortolkes på den her måde.

En kunde kan have mange ordrer, en eller flere ordrer kan fragtes af samme shipper, en ordrer kan have mange produkter, og et produkt kan høre til mange ordrer. Derfor har vi en entitet kaldet ”shopping cart”. Som viser et mange til mange forhold.

d. Database