Relations Based on E/R Diagram

Relations

Entities

Song(sID, title, length, date_recorded, bpm, lyrics)
Artist(artID, name, description, date_formed)
Album(albID, title, date_released, artwork)
Product(pID, price)
Genre(genID, name)
Customer(cID, name, email, credit_card_number, birthdate, password)
Order(oID, date, price)
Country(coID, name)
Language(langID, name)
Format(fID, extension)

Relationships

wrote(artID, sID)
album_artist_in(artID, albID)
track_in(sID, albID, track_number)
associated_with(sID, genID)
places(cID, oID)
contains_product(oID, pID)
song_is_a(pID, sID)
album_is_a(pID, albID)
rates(sID, cID, rating)
is_located_in(cID, coID)
is_from(artID, coID)
speaks(coID, langID)
communicates_in(cID, langID)
is_written_in(sID, langID)
is_in_format(sID, fID)

Opportunities to Combine Redundancies

As far as we can tell we aren't storing any redundant any information. Arguments can be made for removing certain ID attributes from some entities but music and everything associated to it are not constant. What may be called heavy metal today may not be the same heavy metal 20 years from now. Our justification for IDs on every entity allows the database to keep with the ever changing music without having to constantly change the database entries.