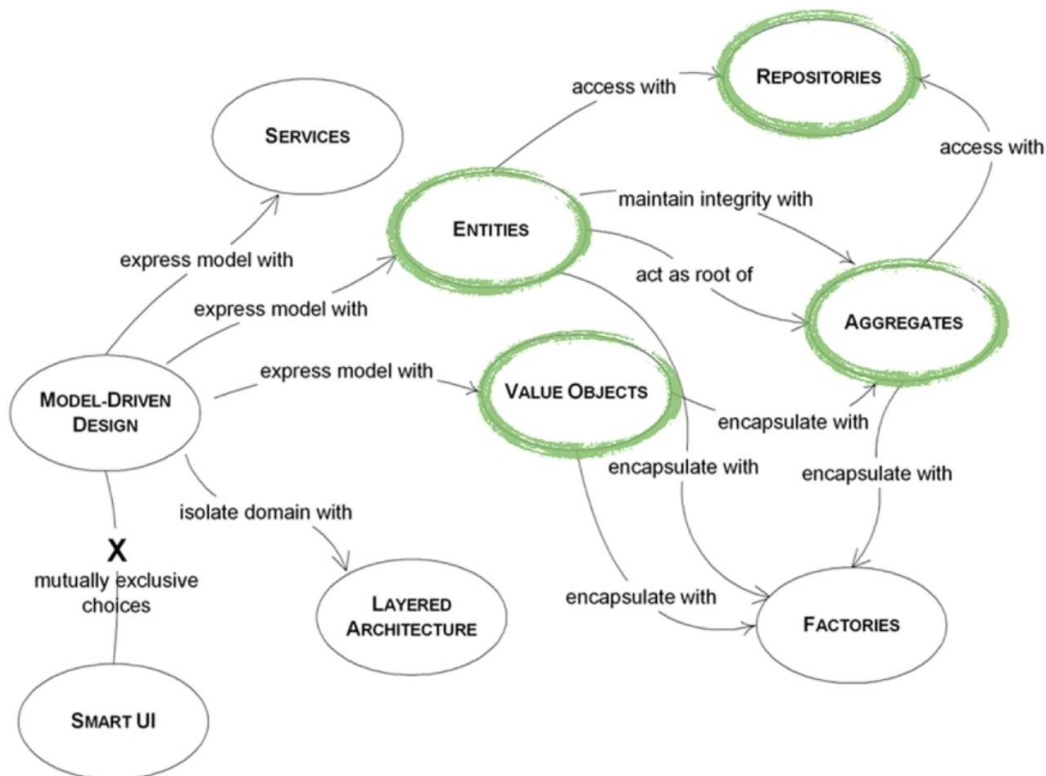


## Domain Driven Design – Repository

**Domain Driven Design (DDD)** a fost introdus de Eric Evans în 2004 și este un design pattern pentru dezvoltarea softurilor cu nevoi complexe în domenii ce folosesc concepte de nivel înalt. DDD pune foarte mult accentul pe necesitatea experților de domeniu de a genera un limbaj omniprezent pe care se bazează proiectul ce poate fi înțeles pe deplin de toți membri echipei și deasemenea flexibil pentru modificări.

Modele high-level ce pot fi folosite pentru crearea, citirea, modificarea și ștergerea obiectelor din agregat : Entity, Value Object, Domain Event, Aggregate, Service, **Repositories**, Factories.



**Repository** – este un serviciu care utilizează o interfață globală ce mediaza între layerele de domeniu și mapare a datelor, pentru a da acces la toate entitățile și obiectele de valoare dintr-o anumită colecție agregată. O dată implementat nu este necesar să fie schimbat în cazul în care se dorește să fie folosit și pentru o altă aplicație, adică serviciul oferă portabilitate.

Este important de știut că trebuie să definiți doar un depozit pentru fiecare rădăcină agregată. Pentru fiecare agregat, trebuie să se creeze o clasă de repository. Într-un microserviciu bazat pe DDD, singurul canal care ar trebui utilizat pentru actualizarea bazei de date ar trebui să fie repository. Acest lucru se datorează faptului că acestea au o relație unu-la-unu cu rădăcina agregată, care controlează invarianții agregatului și consistența tranzacțională. Pentru a atinge scopul rădăcinii agregate pentru a menține coerența tranzacțională între toate obiectele din cadrul agregatului, nu trebuie să creați niciodată un depozit pentru fiecare tabel din baza de date.

Deci un repository este o clasă sau o componentă care încapsulează logica necesară pentru accesarea surselor de date. Ele centralizează funcționalitatea comună de acces la date, oferind o mai bună întreținere și decuplarea infrastructurii sau a tehnologiei utilizate pentru a accesa baza de date din stratul de model de domeniu.

### Repository :

- Nu este un strat de acces la date
- Oferă un nivel mai ridicat de manipulare a datelor
- Este o colecție de rădăcini agregate
- Oferă un mecanism de gestionare a entităților

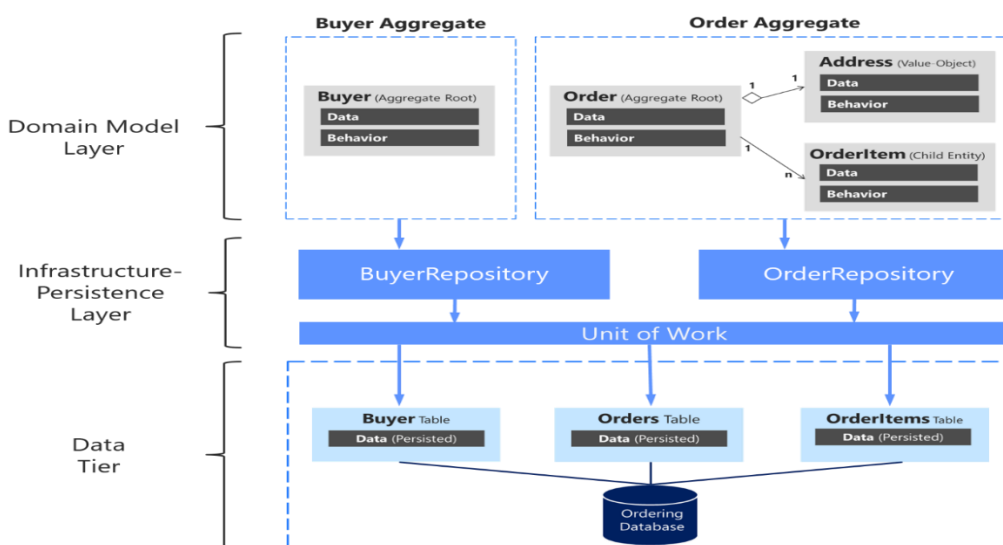


Fig. Relația dintre depozite, agregate și tabele de baze de date