Gyakorlat 3 óra - (entities + queries)

1. Importáljuk a lab\_3 projektet (Import/Existing Maven project/Mappa kiválasztása, ahol a pom file található).
2. A kód jelenleg hibás, ezeket a hibákat fogjuk először javítani. Az Order osztály-on el vannak helyezve az annotációk, de a Customer és Product osztályról hiányoznak. Az Order osztály példája, illetve az órai diák segítségével pótoljuk a hiányzó annotációkat. Az oszlopok nevei a mellékelt adatbázis modellből kiolvashatóak.

A **Customer** osztály következő attributúmait lássuk el annotációval: **id**, **username, lastname, firstname, password, orders**.

A **Product** osztály következő attribútumait is lássuk el annotációval: **id**, **productname, price.**

1. A ProductService, OrderService, és CustomerService osztályokban használni szeretnénk egy EntityManager-t. Injektáljunk ezekbe az osztályokba egy EntityManager példányt.

Telepítsük a projektet a szerverre és indítsuk el. A főoldalon megjelenik a termékek listája, de nem működik se a törlés, se a szerkesztés, se az új termék felvétele funkció.

1. Valósítsuk meg a ProductService#addProduct függvényét (segítség: entityManager.persist)
2. Valósítsuk meg a ProductService#updateProduct függvényét (segítség: entityManager.merge)
3. Valósítsuk meg a ProductService#removeProduct függvényét (segítség: entityManager.remove + figyeljünk arra, hogy törlés előtt menedzseltté kell tennünk az objektumot. Próbáljuk ki azt is, milyen hibát kapunk, ha a paraméterként megkapott product-ra hívjuk meg a remove-ot).

Telepítsük a módosításokat a szerverre, és próbáljuk ki a termékek létrehozását/módosítását/törlését. Érdemes megfigyelni a szerver logban, hogy a perzisztenciában végzett műveletek milyen SQL-é alakulnak.

1. Valósítsuk meg az CustomerService#addCustomer függvényét.
2. Valósítsuk meg az OrderService#addOrder függvényét.

a.) Hozzunk létre egy új vásárlót (ez a része már implementálva van)

b.) Hozzunk létre egy új megrendelést, és állítsuk be vásárlójaként az előbb létrehozott vásárlót, a vásárlás dátumaként pedig a mai dátumot.

c.) Adjuk hozzá a megrendelés terméklistájához a paraméterként kapott termékeket.

d.) Mentsük az új megrendelést.

Telepítsük a szerverre. A rendelések megtekintése alatt megjelenik az általunk felvett rendelés.

**Lekérdezések:**

1. A ProductService#getProducts metódus jelenleg dinamikus lekérdezést használ. Alakítsuk át statikus NamedQuery-vé. Ehhez definiáljuk a Product osztályon egy NamedQuery-t, amely lekérdezi az összes Product sort. Használjuk a getProducts lekérdezésben.
2. Valósítsuk meg az OrderService#getOrders(String username) metódusát, hogy csak azokat a megrendeléseket adja vissza, amelyeket a username paraméterben megkapott felhasználónevű vásárló adott föl.

Ehhez definiáljunk egy NamedQuery-t az Order osztályon, amely fogadja a username paramétert, majd használjuk a NamedQuery-t a metódusban.

1. Valósítsuk meg CustomerService#getCustomers metódusát, amely visszaadja az összes, az adatbázisban tárolt vásárlót. Ehhez használjuk NamedQuery-t.