

Toate interfetele sunt tranzactionale

* Totusi, daca de ex metoda save() de la JpaRepository e chemata intr-o tranzactie, ea nu va mai folosi tranzactia proprie, ci va folosi tranzactia metodei de unde e apelata
* 

@Param e cel de dupa : din Query

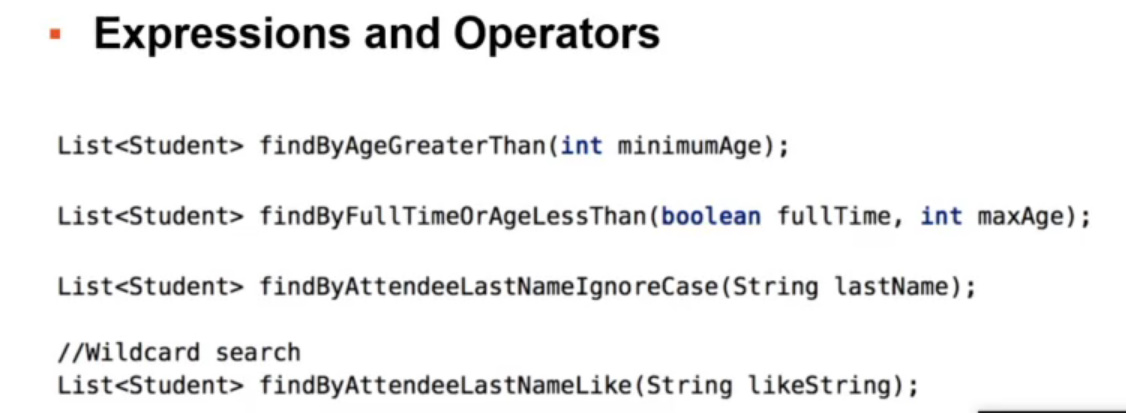
* orice metoda ce face un update necesita anotatia @Modifing, de altfel vom avea exceptie la runtime
* JpaRepository;

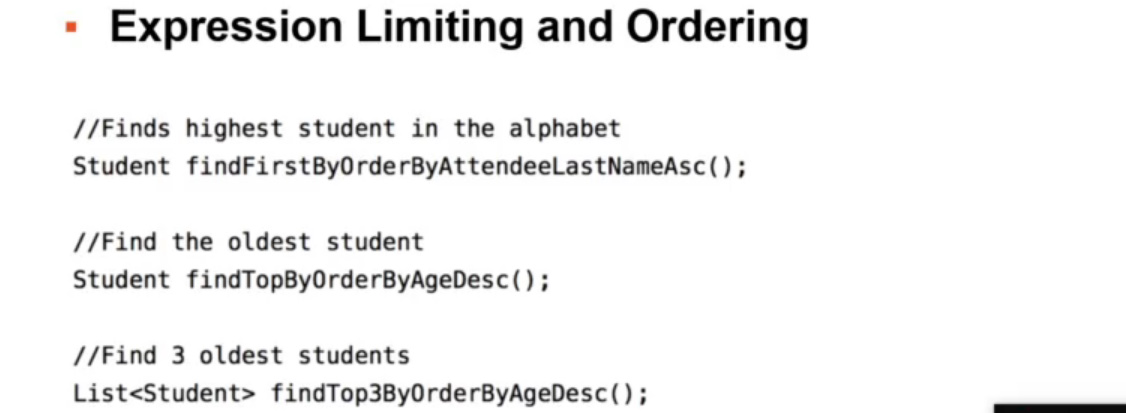


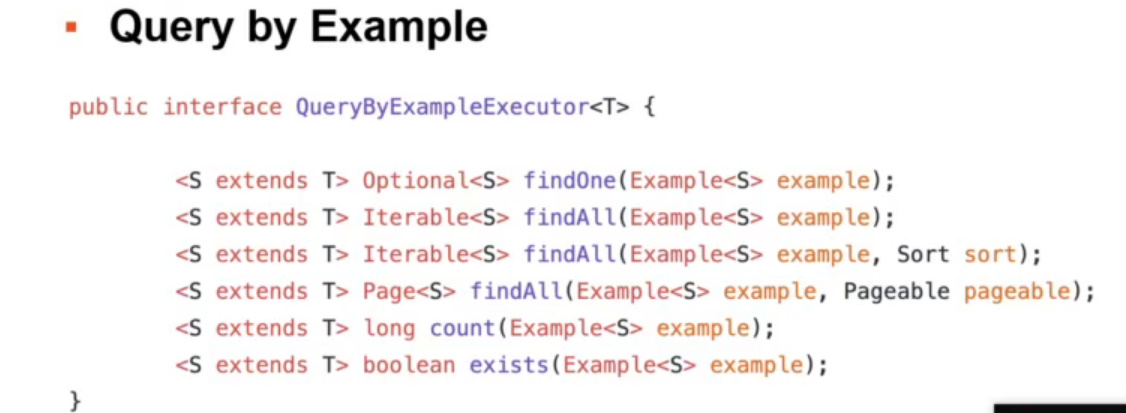
JPA salveaza datele in baza de date cat mai tarziu. save() nu inseamna ca datele vor fi scrise imediat in baza de date, ci mai degraba la final de tranzactie.

**flush()** – face ca queries sa fie executate in baza de date, dar sa nu fie commit inca. E util cand gen vrem sa primit un id de la un user si avem envoie de el tot in tranzactie pana a da commit.

**deleteInBatch()** – deleteAll face un forEach() pentru fiecare item. deleteInBatch creaza un singur query pentru a sterge toate datele

* Spring mereu creaza un object proxy pentru repository creat cu noi de JPA si de asta putem da @Autowired
* 





30:00