

JPA - zapytania JPQL

→ JPQL (ang. *Java Persistence Query Language*)

- ◆ niezależny od bazy danych zorientowany obiektowo język zapytań
- ◆ używa się go do wykonywania zapytań do bazy danych
- ◆ przypomina język SQL ale operuje na encjach JPA (obiektach), a nie na tabelach bazy danych
- ◆ wspierane zapytania to: SELECT, UPDATE, DELETE.

→ HQL (ang. *Hibernate Query Language*)

- ◆ język zapytań biblioteki Hibernate
- ◆ stworzony wcześniej niż JPQL
- ◆ JPQL był mocno inspirowany podczas tworzenia specyfikacją HQL

<https://www.thoughts-on-java.org/jpql/>

Zadanie nr 9:

1. Przejdź do modułu **jpa-starter** i sprawdź konfigurację bazy danych w pliku **persistence.xml** dla Persistence Unit o nazwie: **pl.sda.jpa.starter.queries**. Powinna wskazywać na bazę: **jpa_queries_test**. Stwórz nową bazę danych o takiej nazwie.
2. Znajdź klasę **JpaQueries**. Uruchom metodę **simpleQuery()**. Przeanalizuj kod. Następnie dodaj kod który:
 - a. Wyświetli wszystkich studentów posortowanych rosnąco po nazwie
 - b. Wyświetli trzech ostatnich studentów posortowanych malejąco
 - c. Wyświetli tylko studentów (tylko nazwy) którzy mają więcej niż 25 lat
 - d. Wyświetli wiek i ilość studentów w tym wieku na podstawie tabelki **students** - dane należy pobrać jako listę obiektów klasy (stwórz tą klasę): **StudentsStats** z polami: *age*, *studentsNumber*
3. Znajdź klasę **JpaQueries**. Uruchom metodę **relationsQuery()**. Przeanalizuj kod.
4. W klasie **CourseEntityDaoExt** uzupełnij metody, które wykorzystają JPQL, zgodnie z tym co zapisane jest w komentarzach do metod:
 - a. **findByCity()**,
 - b. **findByName()**,
 - c. **findByDateRange()**,
 - d. **findByCities()**
5. W klasie **StudentEntityDaoExt** uzupełnij metody, które wykorzystają JPQL, zgodnie z tym co zapisane jest w komentarzach do metod:
 - a. **findBySeatRow()**,
 - b. **findByCityAddress()**,
 - c. **(dla chętnych)** **findBySkills()**,
 - d. **(dla chętnych)** **findTheMostSkilled()**