زبان کوئری Hibernate

زبان کوئری hibernate (HQL) زبان کوئری شی گرا و بسیار شبیه به sql هست، با این فرق که به جای اجرا بر روی جداول و ستون ها، hql بر روی persistent object ها و property های اونها کار میکنه. در حقیقت hql وابستگی به جداول دیتابیس نداره و زبان مستقل از دیتابیسی است.

کوئری های نوشته شده به زبان HQL توسط hibernate به کوئری های معمول sql ترجمه می شن که به نوبه خودش روی دیتابیس عملیات انجام میده.

درسته که می تونیم از sql native به صورت مستقیم در hibernate استفاده کنیم، اما بهتره که همیشه تا جایی که امکانش هست از hql استفاده کنیم، به این دلیل که ممکنه در آینده دیتابیس عوض بشه (مثلا اوراکل) و اینکه می تونیم از استراتژی های hibernate sql generation and caching خیلی بهره ببریم.

مزیت های HQL

* مستقل از دیتابیس
* پشتیبانی از کوئری های چند ریختی (polymorphic)
* سادگی در یادگیری برای برنامه نویس

توی زبان HQL کلمات کلیدی مثل SELECT، FROM، WHERE و ... case-sensitive نیستند، اما property هایی مثل ستون و نام جدول حساسند.

واسط Query

یک نمایش شی گرا از کوئری Hibernate است. یک شی از Query توسط صدا زدن متد createQuery() موجود در واسط Session به دست میاد.

واسط Query متد های زیادی داره که معمول ترین اونها در زیر اومده:

|  |  |  |
| --- | --- | --- |
| ردیف | نام تابع | توضیحات |
| 1 | public int executeUpdate() | برای اجرای Updateو Delete استفاده میشه |
| 2 | public List list() | نتیجه جدول رو به صورت یک لیست بر می گردونه |
| 3 | public Query setParameter(int position, Object value) | برای ست کردن یک مقدار به پارامتر های نام گذاری شده در JDBC |
| 4 | public Query setParameter(String name, Object value) | برای ست کردن مقدار در پارامتر نام گذاری شده در کوئری |

عبارت FROM

از عبارت FROM زمانی استفاده می کنیم که بخواهیم یک persistent objectرو به صورت کامل در حافظه لود کنیم.

String hql = "FROM com.erfanfeiz.Employee";

Query query = session.createQuery(hql);

List results = query.list();

عبارت AS

این عبارت زمانی استفاده میشه که بخواهیم اسم مستعاری (alias) رو به کلاسی در کوئری HQLی مون نسبت بدیم. اینکار رو مخصوصاً زمانی انجام میدیم که کوئری طولانی باشه.

String hql = "FROM Employee AS E";

Query query = session.createQuery(hql);

List results = query.list();

کلمه کلیدی AS اختیاری است و می تونیم بدون استفاده از اون بعد نام کلاس اسم مستعار رو بزاریم.

String hql = "FROM Employee E";

Query query = session.createQuery(hql);

List results = query.list();

عبارت SELECT

عبارت SELECT کنترل بیشتری بر روی مجموعه نتایج (result set) نسبت به عبارت FROM فراهم میکنه. اگر بخواهیم property های کمتری از اشیا به جای object کامل بدست بیاریم، از این عبارت استفاده می کنیم. کد زیر تنها فیلد نام رو از شی Employee بر می گردونه:

String hql = "SELECT E.firstName FROM Employee E";

Query query = session.createQuery(hql);

List results = query.list();

توجه کنید که اینجا Employee.firstName فیلد کلاس Employee هستش نه اسم یک ستون جدول.

عبارت WHERE

اگر می خواهیم که شی که از حافظه بر میگردد رو محدود کنیم از عبارت WHERE استفاده می کنیم،

String hql = “FROM Employee E WHERE E.id = 10”;

Query query = session.createQuery(hql);

List results = query.list();

عبارت ORDER BY

برای مرتب کردن نتایج کوئری HQL نیاز داریم که از عبارت ORDER BY استفاده کنیم. می تونیم این مرتب سازی رو بر اساس هر property که در object های result set بازگشتی وجود داره انجام بدیم اون هم بصورت ASC یا DESC.

String hql = “FROM Employee E WHERE E.id > 10 ORDER BY E.salary DESC”;

Query query = session.createQuery(hql);

List results = query.list();

اگر نیاز داریم بر اساس بیشتر از یک ویژگی این مرتب سازی رو انجام بدیم، property های اضافی رو به آخر عبارت order by اضافه کنیم.

String hql = “SELECT SUM(E.salary), E.firtName FROM Employee E “ +

“GROUP BY E.firstName”;

Query query = session.createQuery(hql);

List results = query.list();

عبارت GROUP BY

این عبارت به hibernate اجازه میده که اطلاعات رو از دیتابیس بیرون بکشه و بر اساس مقدار یک attribute اونها رو گروه بندی کنه. معمولا از نتیجه برای ساختن یک مقدار تجمعی (aggregate value) استفاده می کنه.

String hql = “SELECT SUM(E.salary), E.firtName FROM Employee E “ +

“GROUP BY E.firstName”;

Query query = session.createQuery(hql);

List results = query.list();

استفاده از پارامتر های نام گذاری شده(Name Parameters)

Hibernate از پارامترهای نام گذاری شده در کوئری های HQL پشتیبانی می کنه، این ویژگی به ما کمک می کنه تا کوئری های بنویسیم که ورودی کاربر رو به راحتی بگیره و دیگه نگران حملات SQL injection نباشیم.

String hql = "FROM Employee E WHERE E.id = :employee\_id";

Query query = session.createQuery(hql);

query.setParameter("employee\_id",10);

List results = query.list();

عبارت UPDATE

بروزرسانی های تجمعی از hibernate 3 اضافه شدند و DELETE در این نسخه با نسخه 2 متفاوت است. واسط Query متدی به نام executeUpdate() برای اجرای عبارات UPDATE و DELETE در HQL استفاده میشه.

عبارت UPDATE می تونه برای بروز کردن یک یا چند property متعلق به یک یا چند object استفاده بشه.

String hql = "UPDATE Employee set salary = :salary " +

"WHERE id = :employee\_id";

Query query = session.createQuery(hql);

query.setParameter("salary", 1000);

query.setParameter("employee\_id", 10);

int result = query.executeUpdate();

System.out.println("Rows affected: " + result);

عبارت DELETE

این عبارت برای حذف کردن یک یا چند Object استفاده میشه.

String hql = "DELETE FROM Employee " +

"WHERE id = :employee\_id";

Query query = session.createQuery(hql);

query.setParameter("employee\_id", 10);

int result = query.executeUpdate();

System.out.println("Rows affected: " + result);

عبارت INSERT

HQL از عبارت INSERT INTO در تنها در جاهایی که رکورد ها بتونن از یک شی داخل یک شی دیگر insert بشن، پشتیبانی می کنه.

String hql = "INSERT INTO Employee(firstName, lastName, salary)" +

"SELECT firstName, lastName, salary FROM old\_employee";

Query query = session.createQuery(hql);

int result = query.executeUpdate();

System.out.println("Rows affected: " + result);

متدهای تجمعی (Aggregate)

HQL از تعدادی متدهای تجمعی، شبیه به SQL، پشتیبانی می کنه. در اینجا به تعدادی تابع موجود اشاره می کنیم:

|  |  |  |
| --- | --- | --- |
| ردیف | نام تابع | توضیحات |
| 1 | Avg(property name) | میانگین مقدار یک ویژگی |
| 2 | Count(property name or \*) | تعداد دفعاتی که یک ویژگی در نتایج تکرار شده |
| 3 | Max(property name) | ماکسیمم مقدار مقادیر یک ویژگی |
| 4 | Min(property name) | مینیمم مقدار مقادیر یک ویژگی |
| 5 | Sum(property name) | مجموع کل مقادیر یک ویژگی |

کلمه کلیدی distinct تنها مقادیر یکتا(unique) رو در row set می شمارد.

String hql = "SELECT count(distinct E.firstName) FROM Employee E";

Query query = session.createQuery(hql);

List results = query.list();

Pagination با استفاده از کوئری

2 متد برای استفاده از pagination در واسط Query وجود دارد.

|  |  |  |
| --- | --- | --- |
| ردیف | متد | توضیحات |
| 1 | Query setFirstResult(int startPosition) | این متد یک مقدار int می گیره که اولین ردیف رو در result set شما نشان می ده، با ردیف 0 شروع می شود. |
| 2 | Query setMaxResults(int maxResult) | این متد به hibernate می گوید که تعداد ثابتی maxResults از اشیا را برگرداند |

با استفاده از 2 متد بالا می تونیم یک کامپوننت paging در برنامه خودمون داشته باشیم. کد زیر در هر زمان 10 ردیف بر می گرداند.

String hql = "FROM Employee";

Query query = session.createQuery(hql);

query.setFirstResult(1);

query.setMaxResults(10);

List results = query.list();