list() or getResultList()

===================

1. It internally uses eager loading for bulk operations.

2. It returns the collection directly.

3. Generates only one query to get all the records.

4. suitable for good performance.

5. It won't generate proxy object.

6. It is not deprecated and it is the industry standard approach.

iterate()

=======

1. It internally uses lazy loading for bulk operation

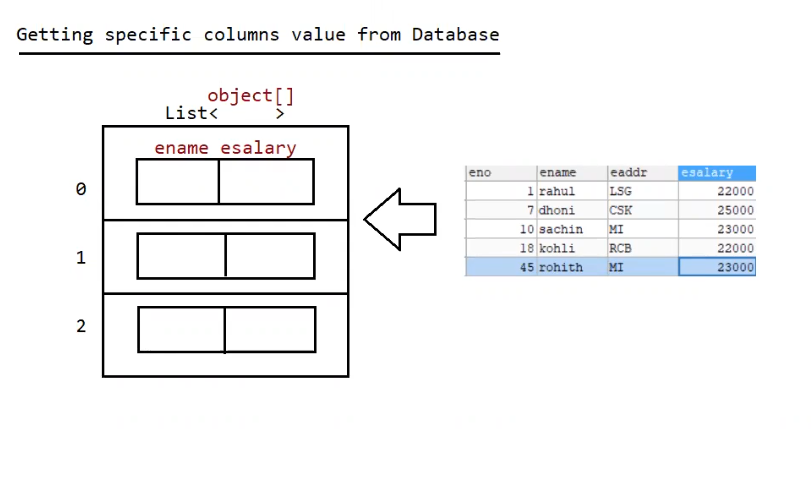
2. It returns the iterator pointing to collection object.

3. Generates n+1 query to get all the records.

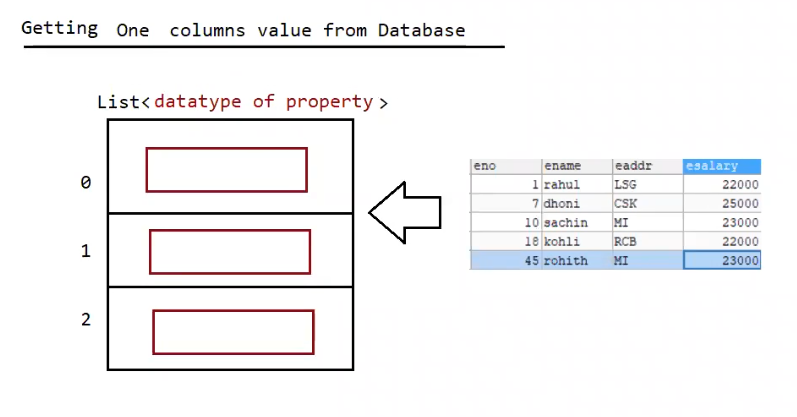
4. Not suitable, it degrades the performance.

5. It generates Proxy object.

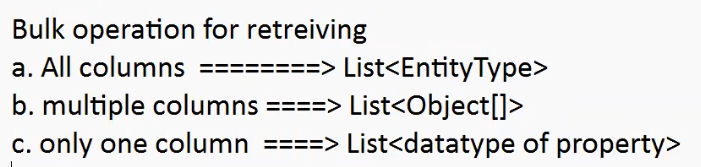
6. It is deprecated because of performance issue.



Eg: HibernateBulkOperationSpecificColumn



Eg: HibernateBulkOperationSingleColumn



**When to Use uniqueResultOptional()**

* Use it when your query **must return either zero or one result** (e.g., searching by primary key or unique column).

Note: To avoid null checking in our application,we can use JDK8 supplied api called "Optional".

Eg: HibernateBulkOperationWithOptional

Note: we should use get()/load() if we are getting record based on primary key value,

we can use uniqueResult()/uniqueResultOptional() for **any query**, including those **based on primary key**, if you're using HQL or Criteria API instead of get().

Eg: HibernateBulkOperationUpdate

Eg: HibernateBulkOperationDelete

HQLInsert operation

==================

It is not possible to insert one record to the database directly using insert query, becoz linking generators with HQL insert query is not possible. so we use session.save() method to insert a record.

We can use HQL insert query to insert bulk record into one db table by selecting them from another db table.

eg: insert into .... values (query is not given)

insert into ... SELECT FROM ...... (given to perform bulk operation)

Eg: HibernateBulkOperationInsert