



CS544 EA

# Hibernate

## Optimization: BatchSize

# @BatchSize

- Hibernate extension commonly used for N+1
  - Helps by loading **several collections in one 'batch'**
  - Gets another batch when previous batch is empty
  - Turns  $N + 1$  into:  $\text{ceil}(N / \text{Batchsize}) + 1$

```
import org.hibernate.annotations.BatchSize;

@Entity
public class SalesRep {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    @OneToMany(mappedBy = "salesRep", cascade=CascadeType.ALL)
    @BatchSize(size=3)
    private List<Customer> customers = new ArrayList<>();
}
```

Typical size between  
3 and 15

1 select

Select \* from Salesrep

Salesrep #1

Salesrep #2

Salesrep #3

Salesrep #4

Salesrep #5

N / Batchsize  
selects

Select \* from Customer  
where salesrep\_id in (1, 2, 3)

Works because it knows  
the IDs of next SalesReps

Select \* from Customer  
where salesrep\_id in (4, 5)

Each select returns  
a 'Batch' of collections

Customer #1

Customer #2

Customer #3

Customer #4

Customer #5

Customer #6

Customer #7

Customer #8

Customer #9

Customer #10

Customer #11

Customer #12

Customer #13

Customer #14

Customer #15

# Code

```
em.getTransaction().begin();

SalesRep sr1 = new SalesRep("John Willis");
SalesRep sr2 = new SalesRep("Mary Long");

sr1.addCustomer(new Customer("Frank", "Brown"));
sr1.addCustomer(new Customer("Jane", "Terrien"));
sr2.addCustomer(new Customer("John", "Doe"));
sr2.addCustomer(new Customer("Carol", "Reno"));

em.persist(sr1);
em.persist(sr2);
em.getTransaction().commit();
```

```
System.out.println("Before query");
TypedQuery<SalesRep> query =
    em.createQuery("from SalesRep", SalesRep.class);
List<SalesRep> salesReps = query.getResultList();
System.out.println("After query");
for (SalesRep s : salesReps) {
    s.getCustomers().get(0).getFirstName();
}
System.out.println("After loop");
```

Before query

Hibernate:

```
select
    salesrep0_.id as id1_3_,
    salesrep0_.name as name2_3_
from
    SalesRep salesrep0_
```

After query

Hibernate:

```
select
    customers0_.salesRep_id as salesRep4_1_1_,
    customers0_.id as id1_1_1_,
    customers0_.id as id1_1_0_,
    customers0_.firstName as firstNam2_1_0_,
    customers0_.lastName as lastName3_1_0_,
    customers0_.salesRep_id as salesRep4_1_0_
from
    Customer customers0_
where
    customers0_.salesRep_id in (
        ?, ?
    )
```

Batch loaded when  
customer first needed

After loop

# BatchSize and N+1

- @BatchSize does not completely eliminate N+1
  - It does significantly reduce it
  - N+ 1 becomes:  $\lceil N / \text{batchsize} \rceil + 1$
- Potential Problems:
  - N+1 effects not completely removed (just reduced)
  - Static, always on, no way to not use it
  - May load (batchsize - 1) objects not needed