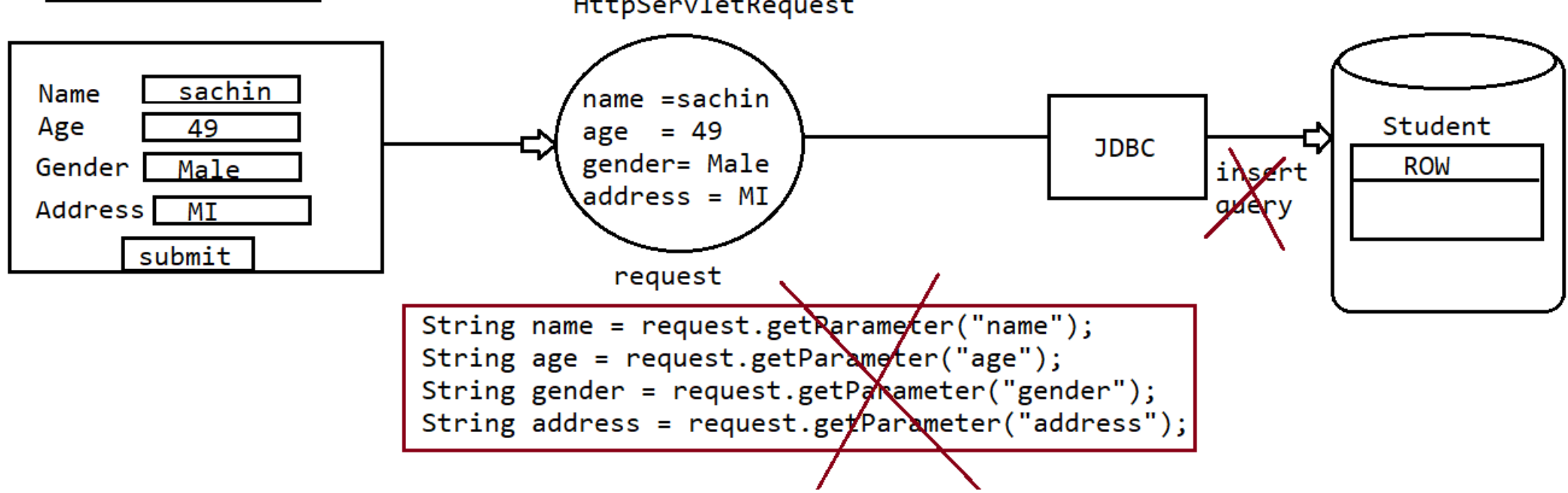
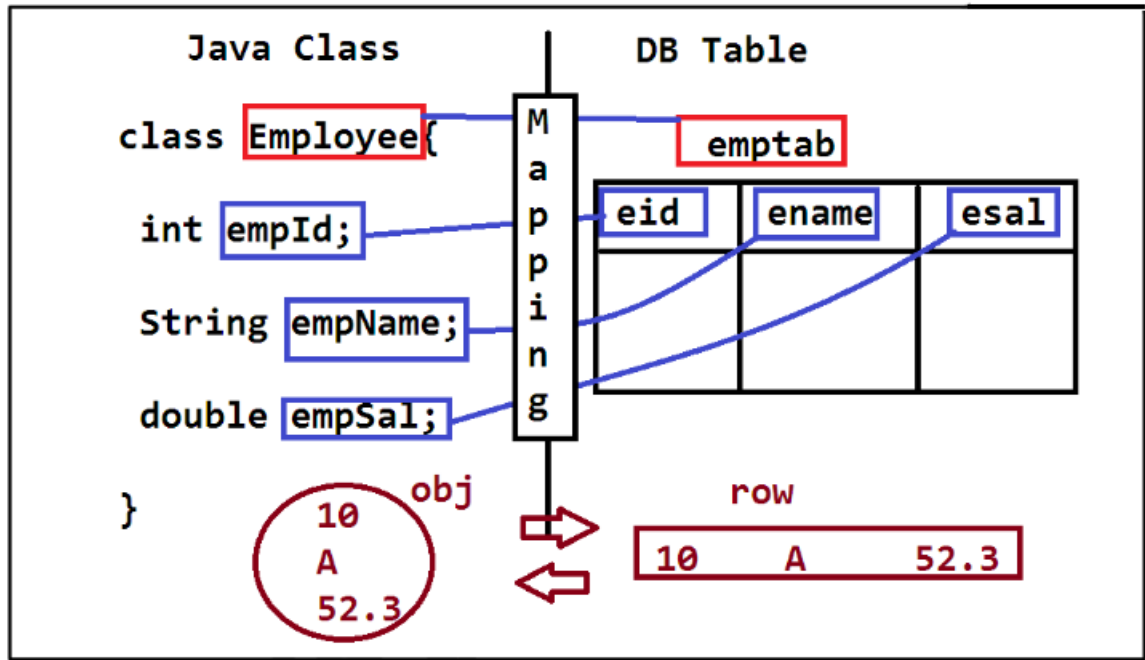
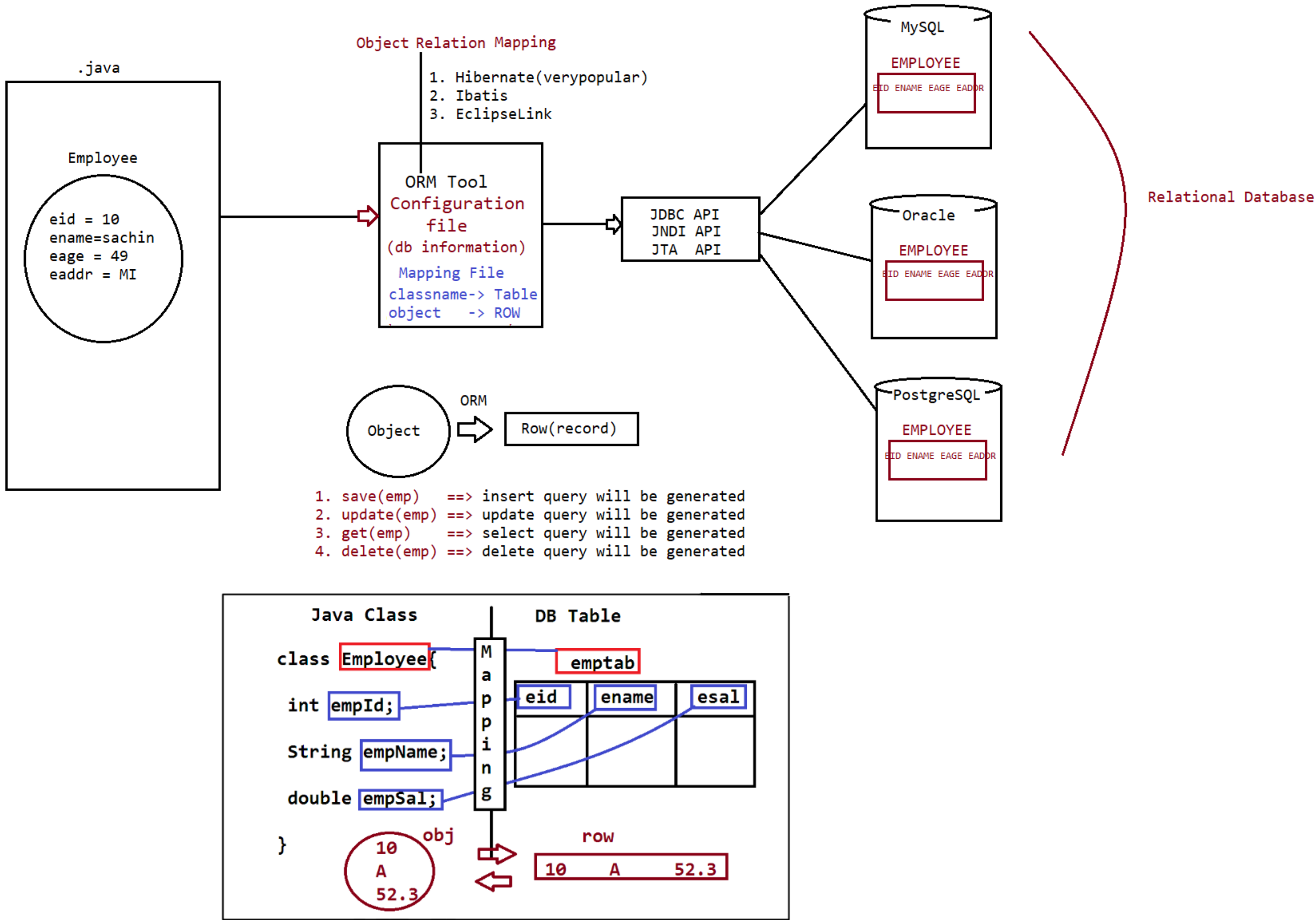
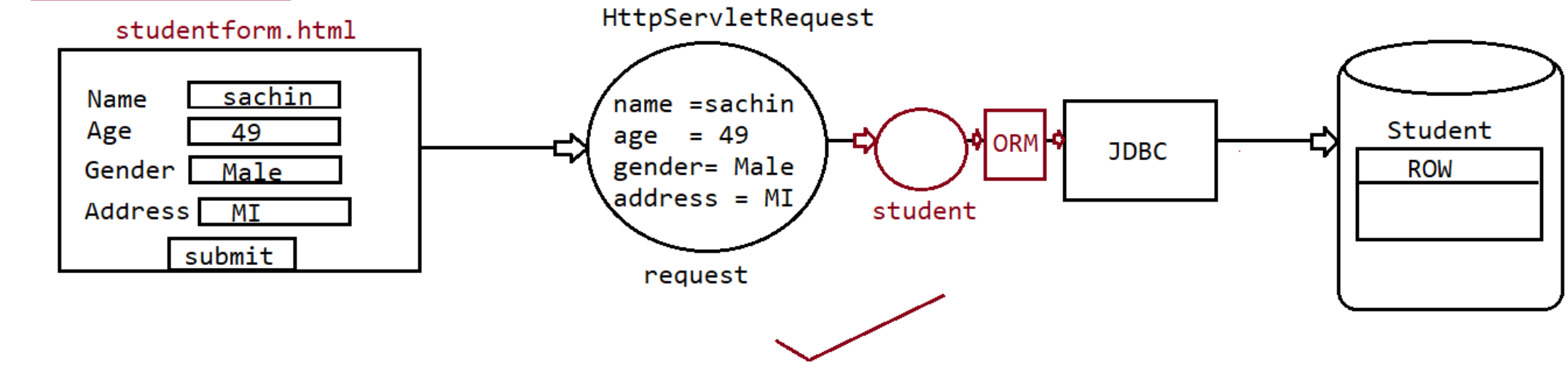


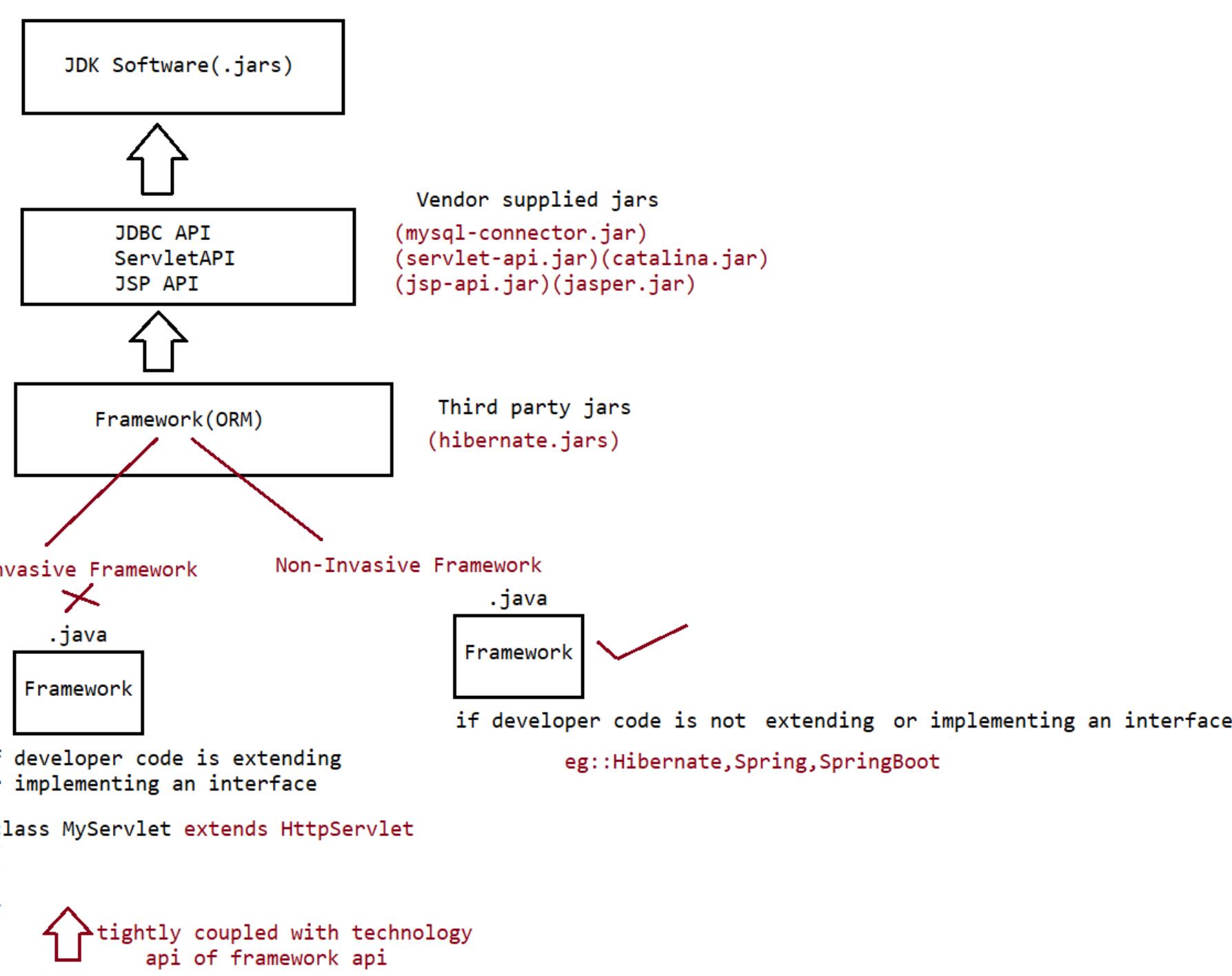
Using Technology

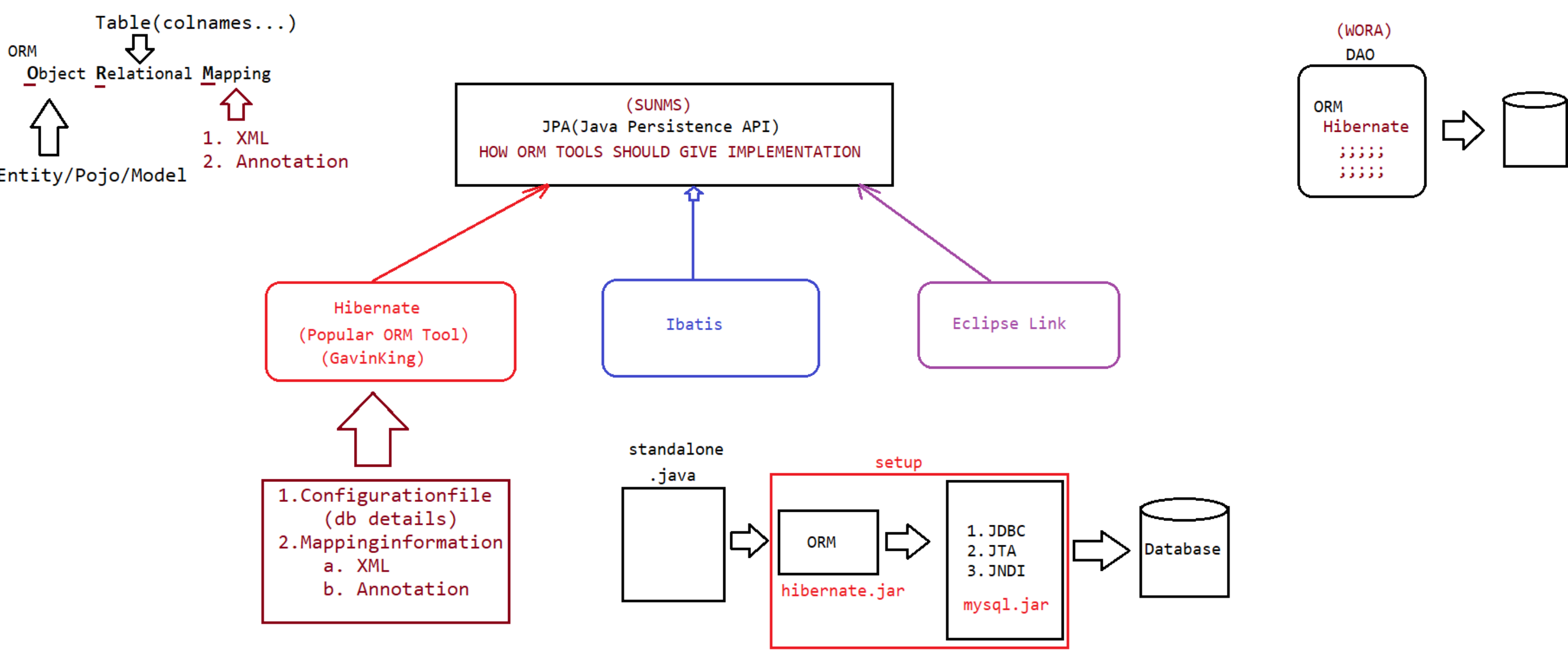


using Framework

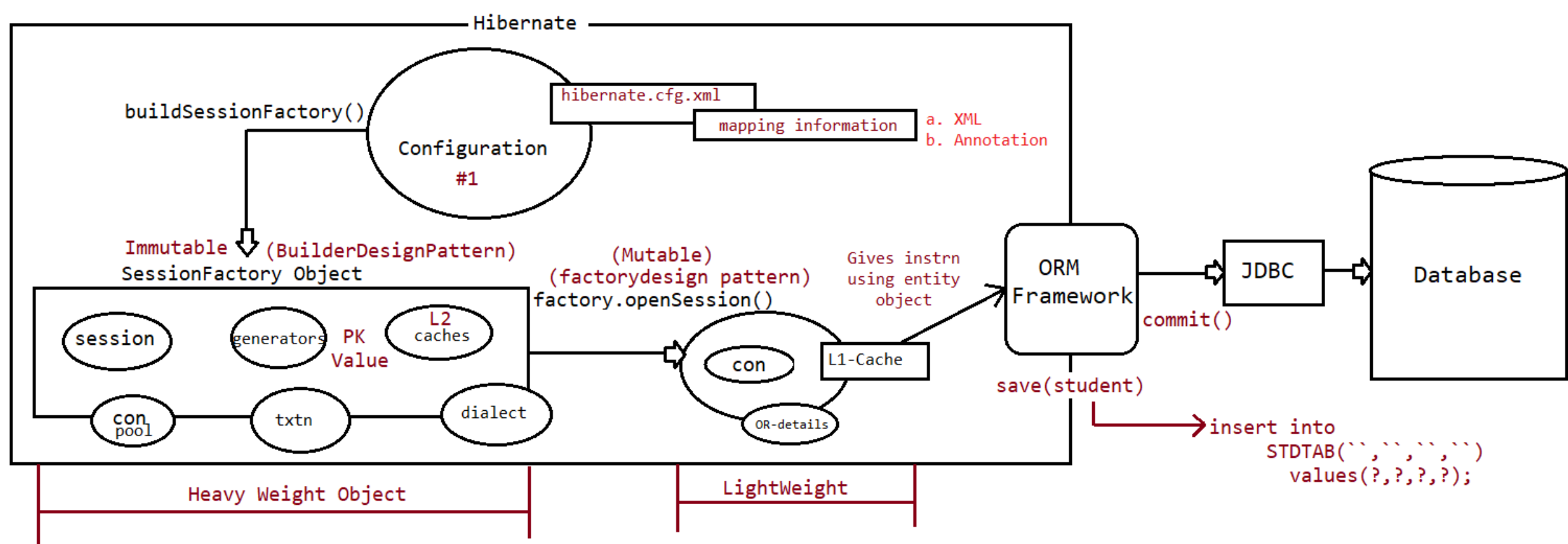
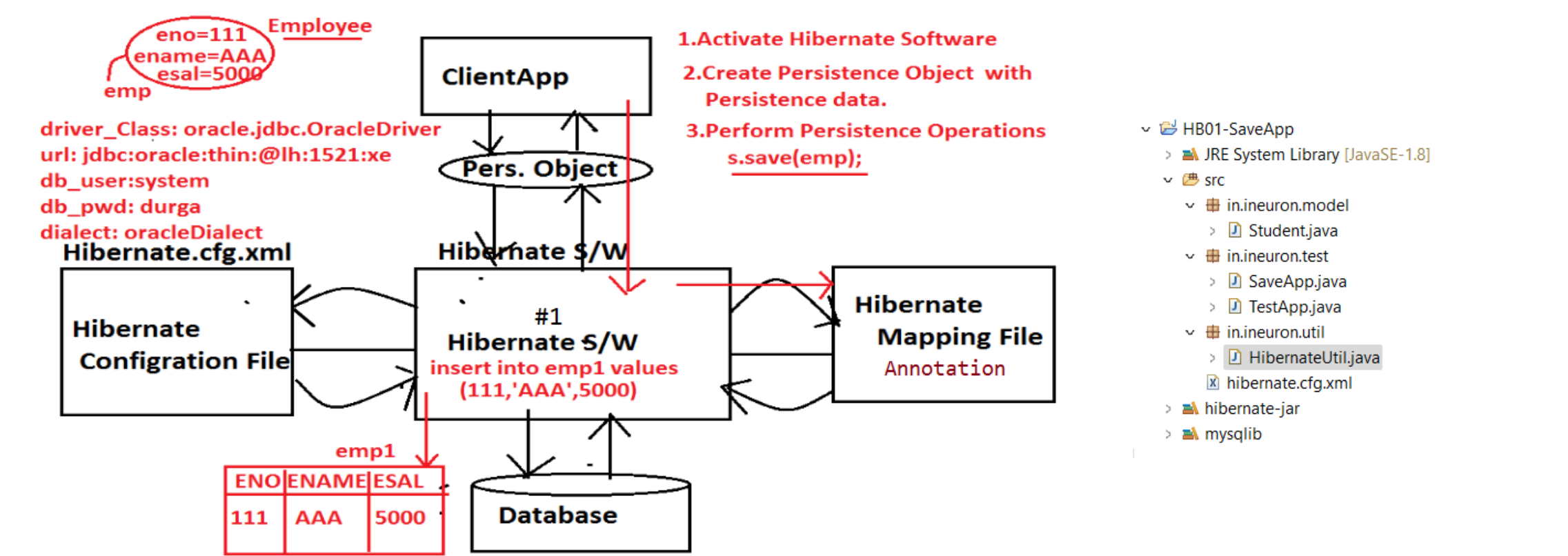


Framework and its types





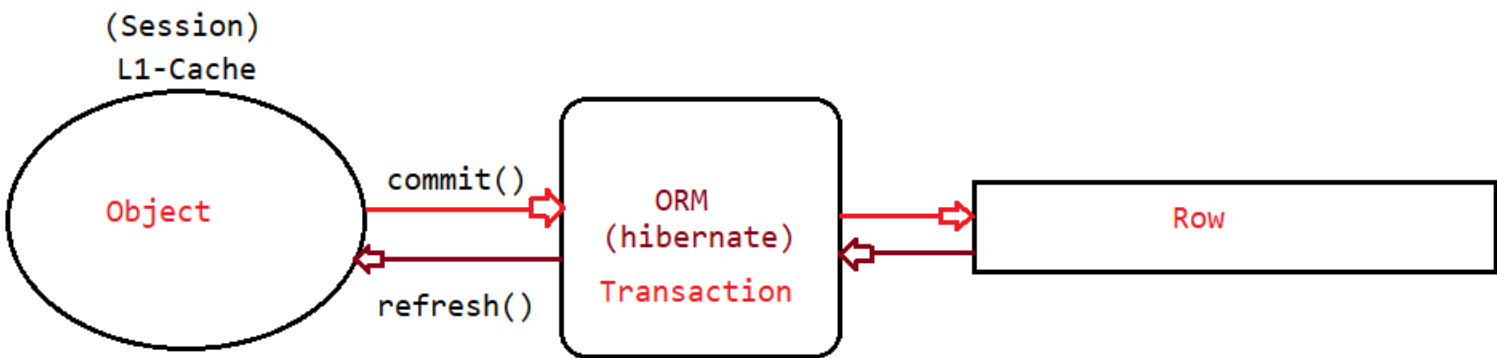
Hibernate Architecture:



*> By default connection object present in connection pool autoCommit mode = false.
connection.setAutoCommit(false)

*> Any persistence operation of non-select type to be done we need to start the transaction
Transaction transaction=session.beginTransaction(); //transaction.begin()

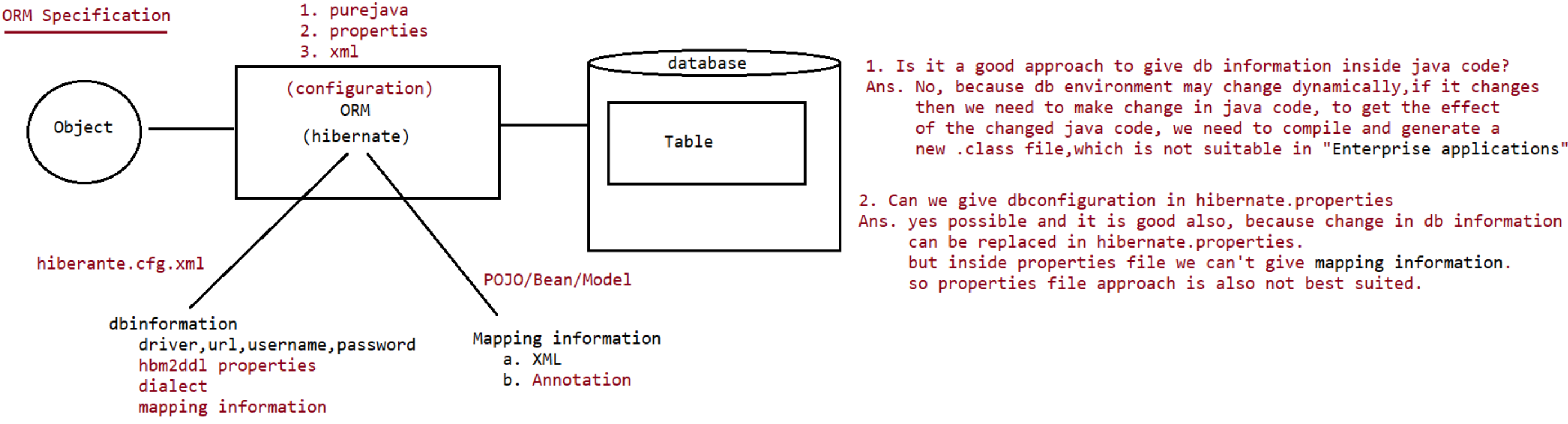
*> If transaction is succesfull, then perform commit operation otherwise perform rollback operation
transaction.commit() //con.commit()
transaction.rollback //con.rollback()



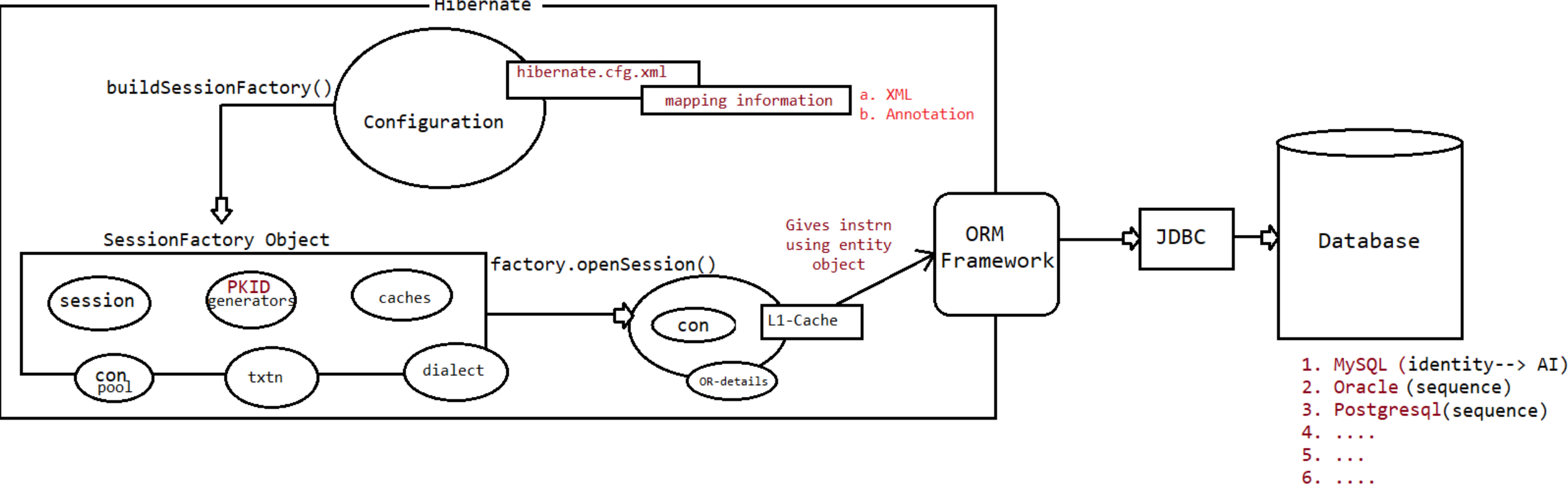
Within in a transaction, any changes made to the object would be automatically reflected to the record in the dbtable once the transaction is committed because of Synchronization.

Within the transaction, any changes made to the record in the table, will it be reflected back to the Object?
Ans. Yes.

The object which is obtained from database by default would be available inside L1-cache, any changes made to the row in the database would not be reflected automatically to the L1-cache object,for the reflection to happen we need to use session.refresh(object).



Generators in hibernate



compositekey ← Take 2 columns inside seperate class and inject that object in main class.

ProgrammerProjectInfo

pid

pname

deptno

projId

projName

100	sachin	10	501	IPL
100	sachin	10	502	IND
101	kohli	19	501	IPL
101	kohli	19	502	IND

(pid,projId)

ProgrammerId

ProgrammerProjectInfo

HB-13-HibernateCompositeldApp

JRE System Library [JavaSE-1.8]

src

in.neuron.model

ProgrammerProjId.java

ProgrammerProjInfo.java

in.neuron.test

InsertRecordApp.java

SelectRecordApp.java

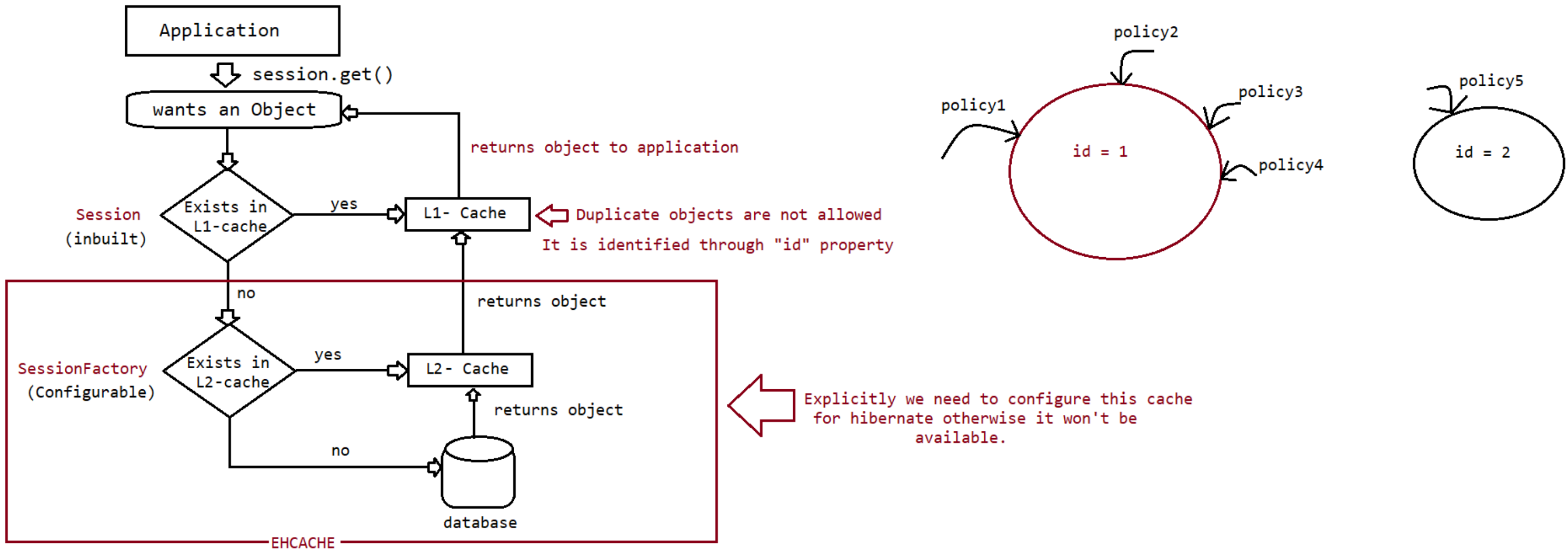
in.neuron.util

HibernateUtil.java

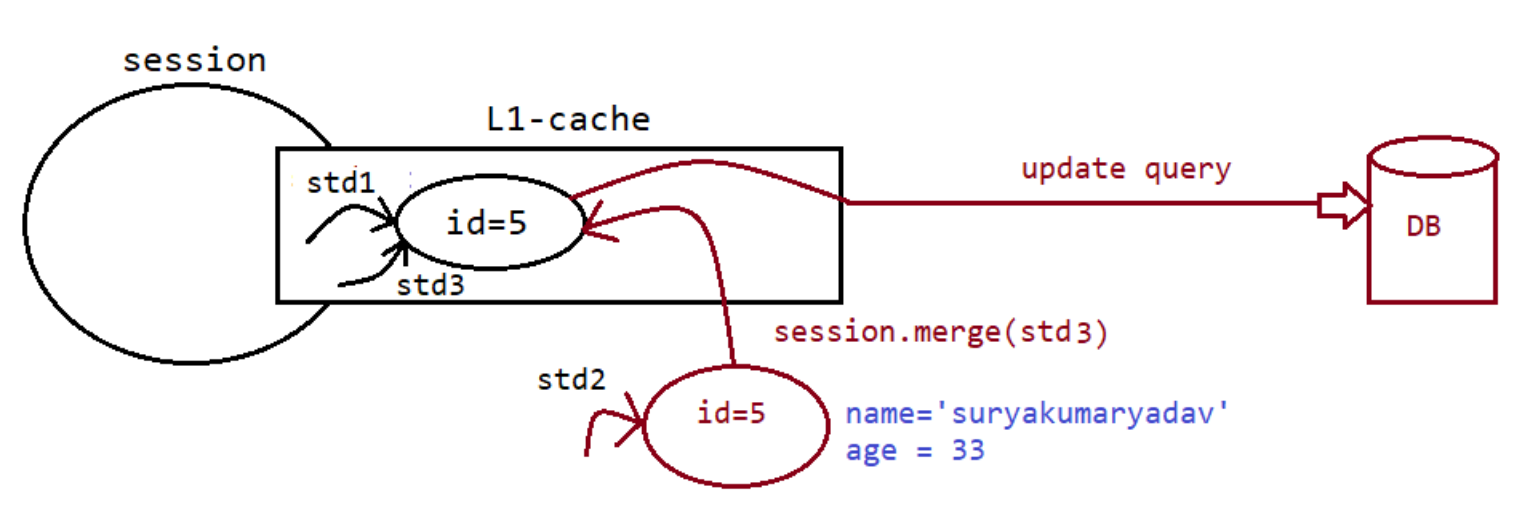
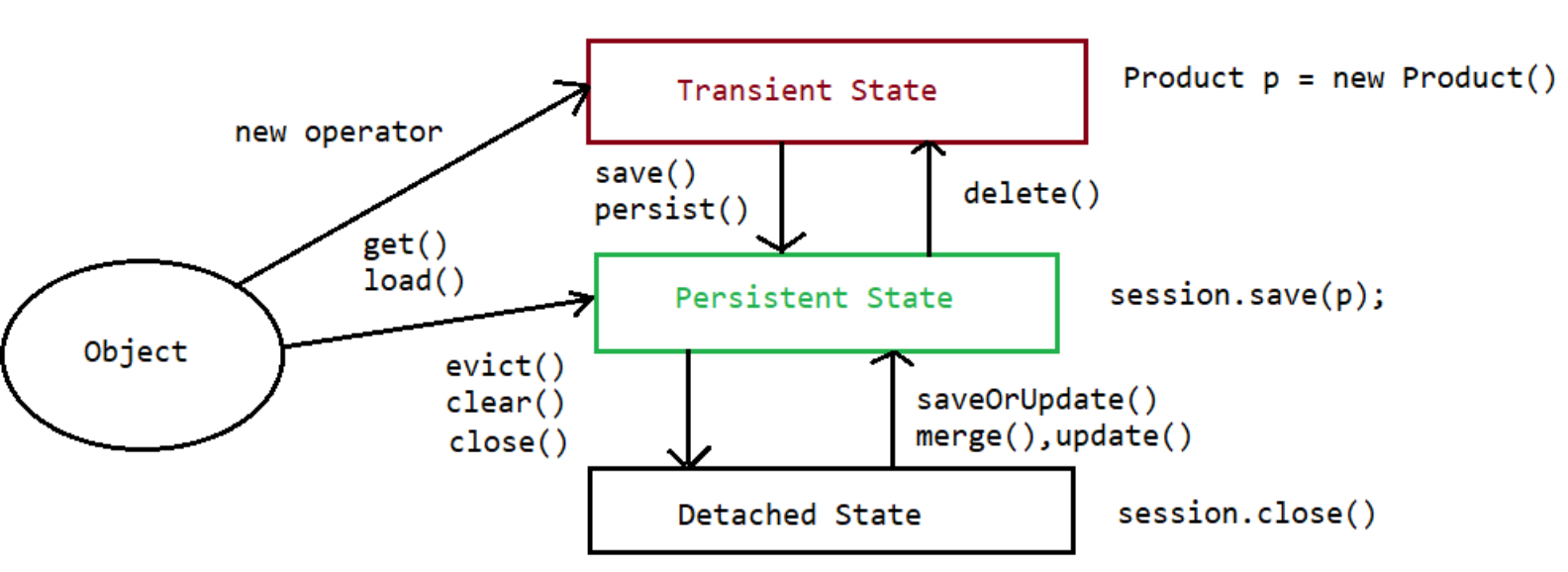
hibernate.cfg.xml

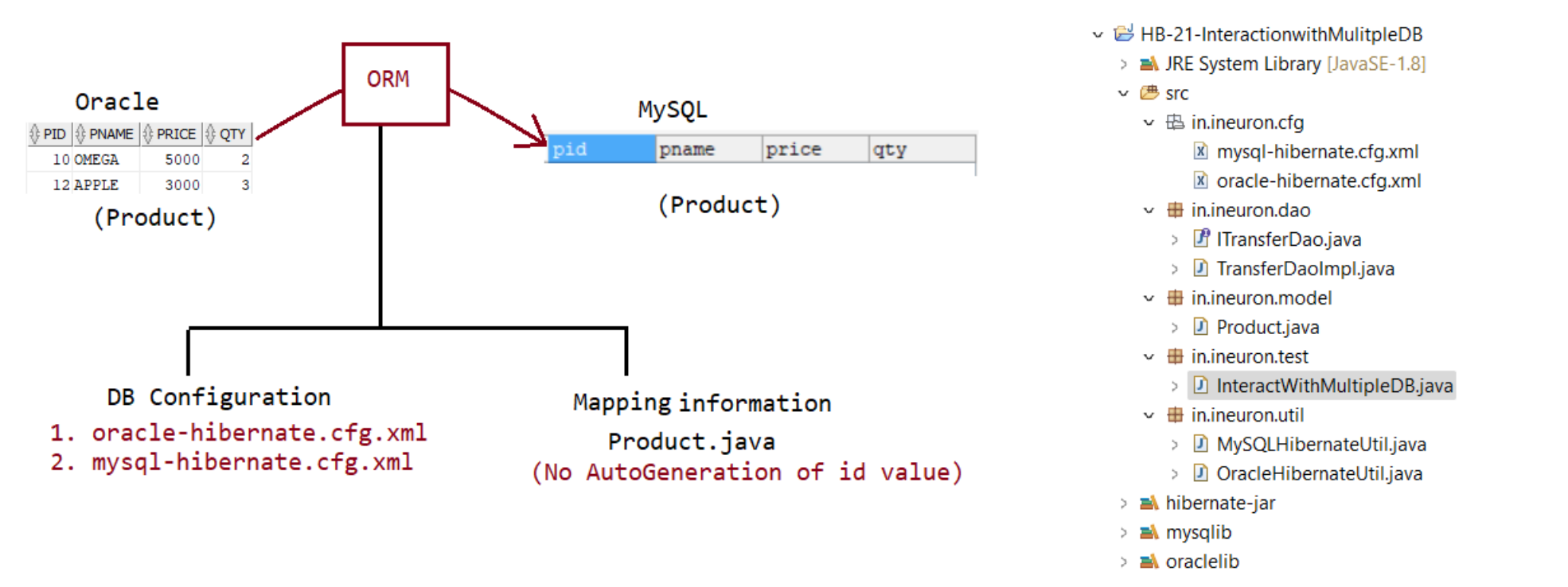
hibernate-jar

mysqlib



Entity Object Life Cycle





HB-21-InteractionWithMultipleDB

JRE System Library [JavaSE-1.8]

src

in.ineuron.cfg

mysql-hibernate.cfg.xml
oracle-hibernate.cfg.xml

in.ineuron.dao

ITransferDao.java
TransferDaoImpl.java

in.ineuron.model

Product.java

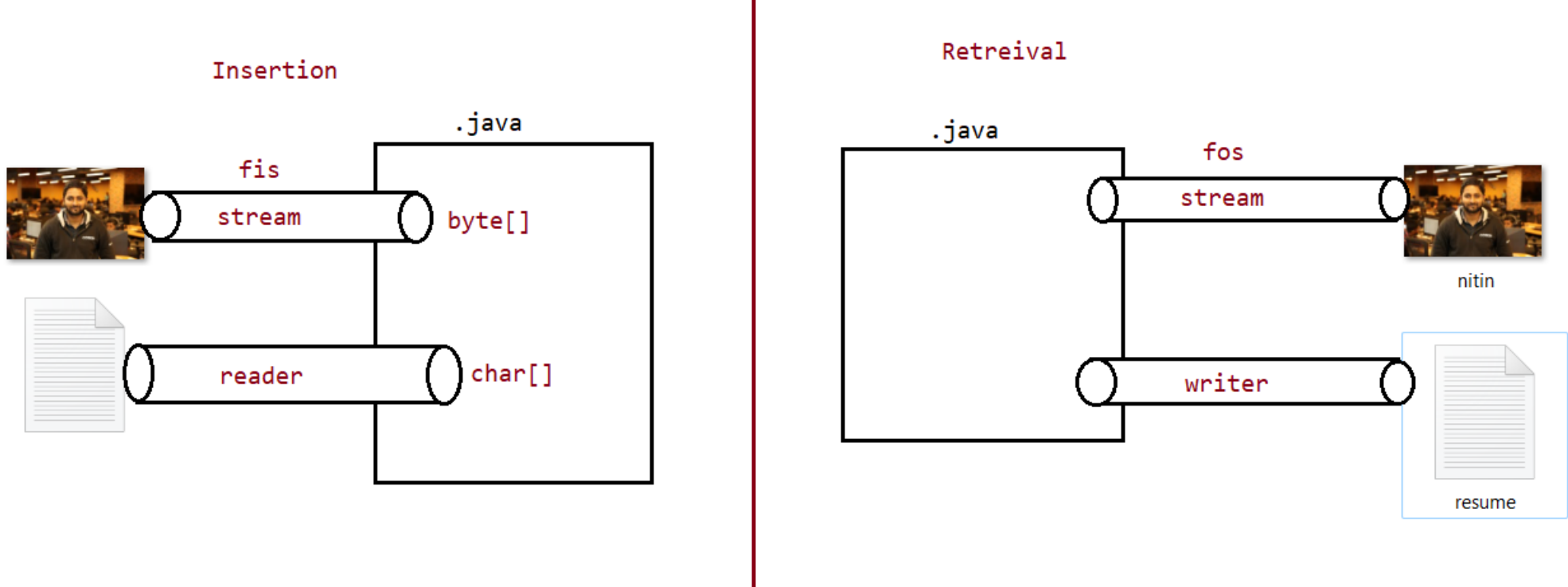
in.ineuron.test

InteractWithMultipleDB.java

in.ineuron.util

MySQLHibernateUtil.java
OracleHibernateUtil.java

hibernate-jar
mysqlib
oraclelib



HB-22-HibernateLobOperation

JRE System Library [JavaSE-1.8]

src

in.ineuron.model

JobSeeker.java

in.ineuron.test

InsertRecordApp.java
SelectRecordApp.java

in.ineuron.util

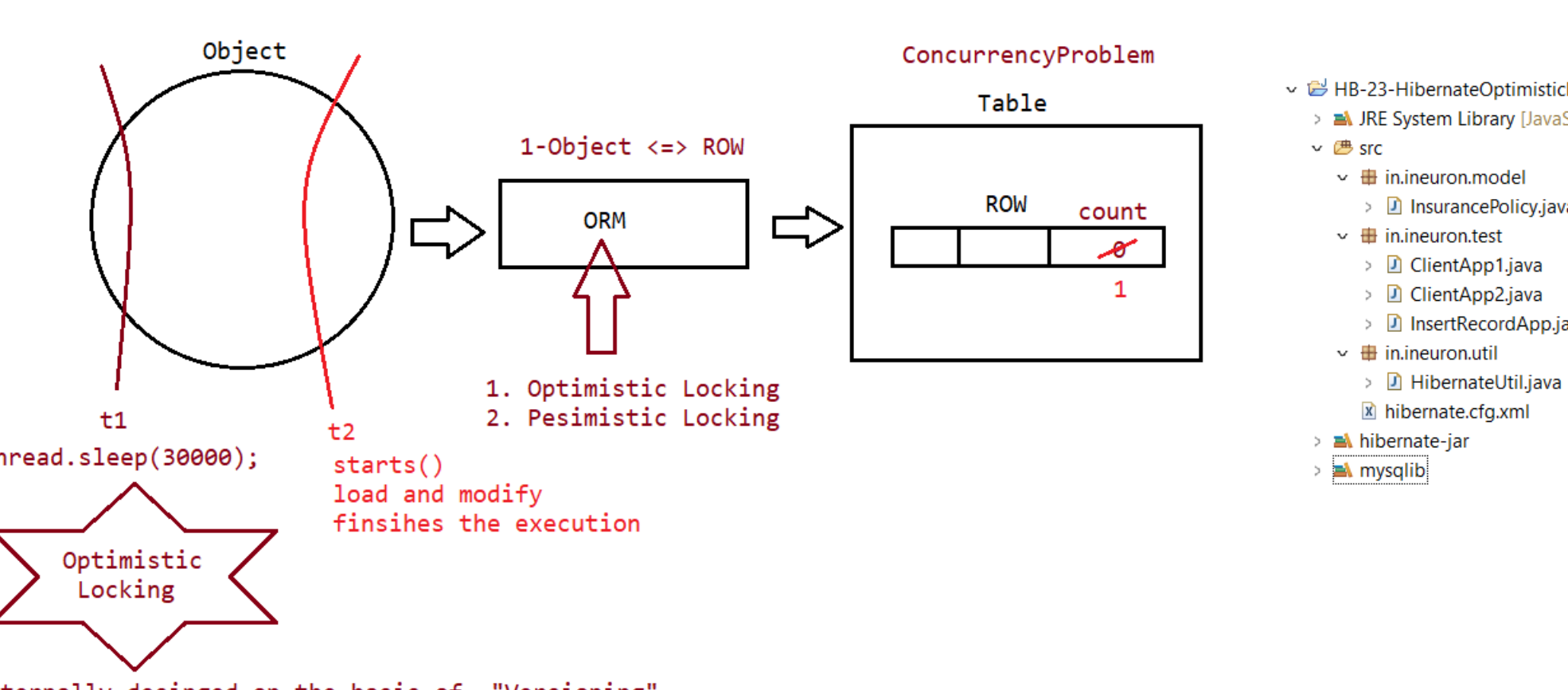
HibernateUtil.java
hibernate.cfg.xml

hibernate-jar
mysqlib

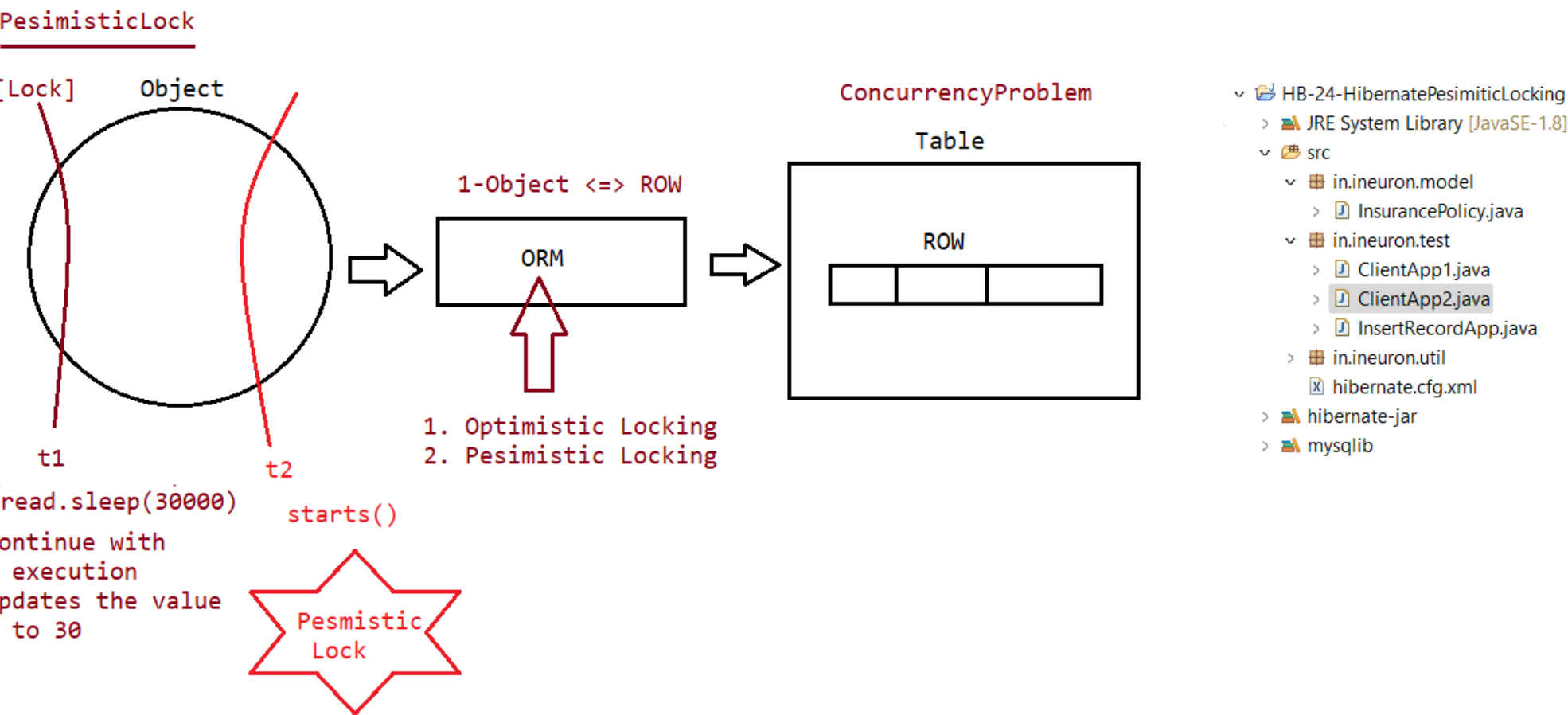
store

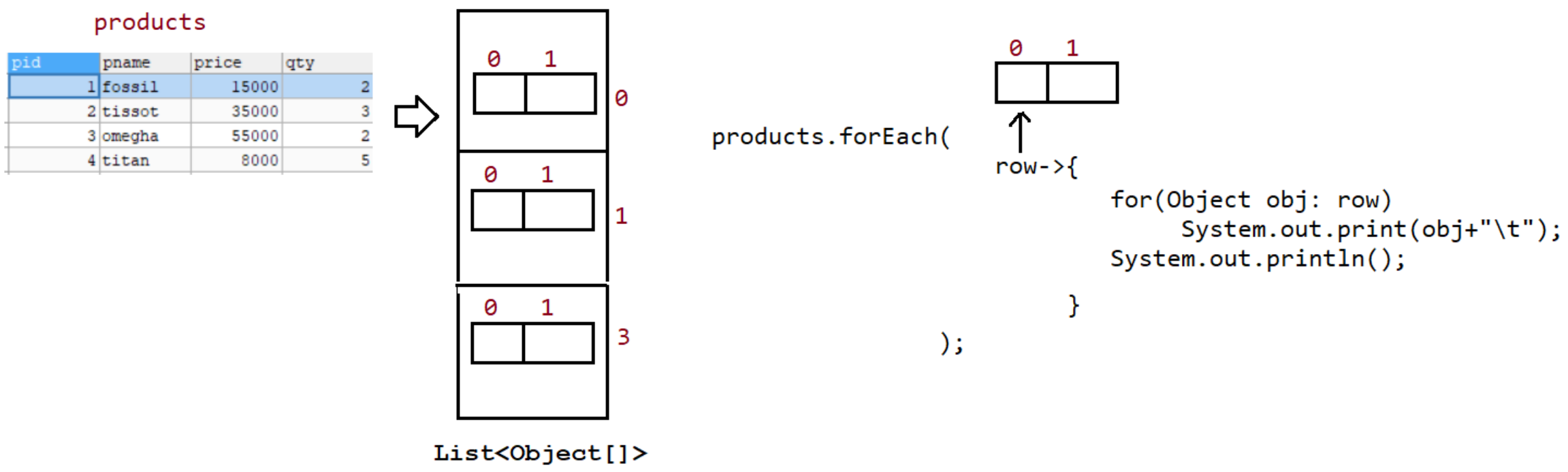
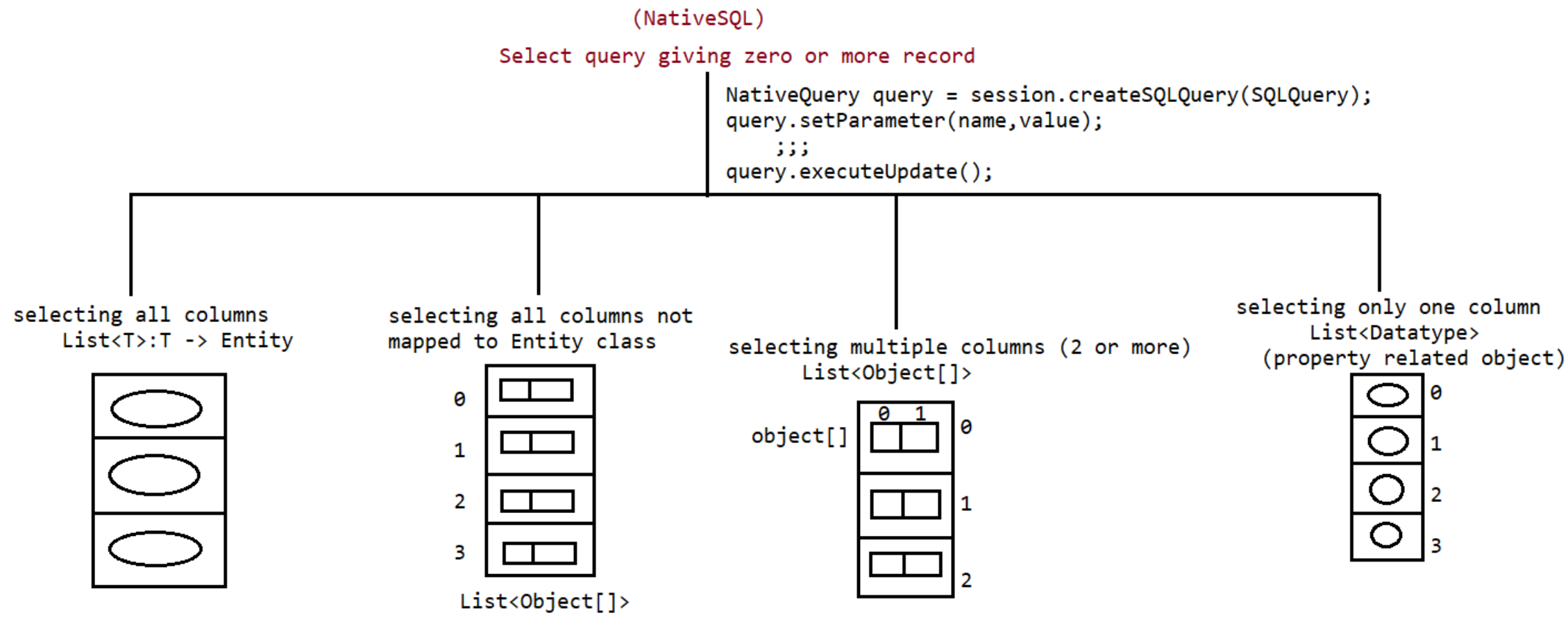
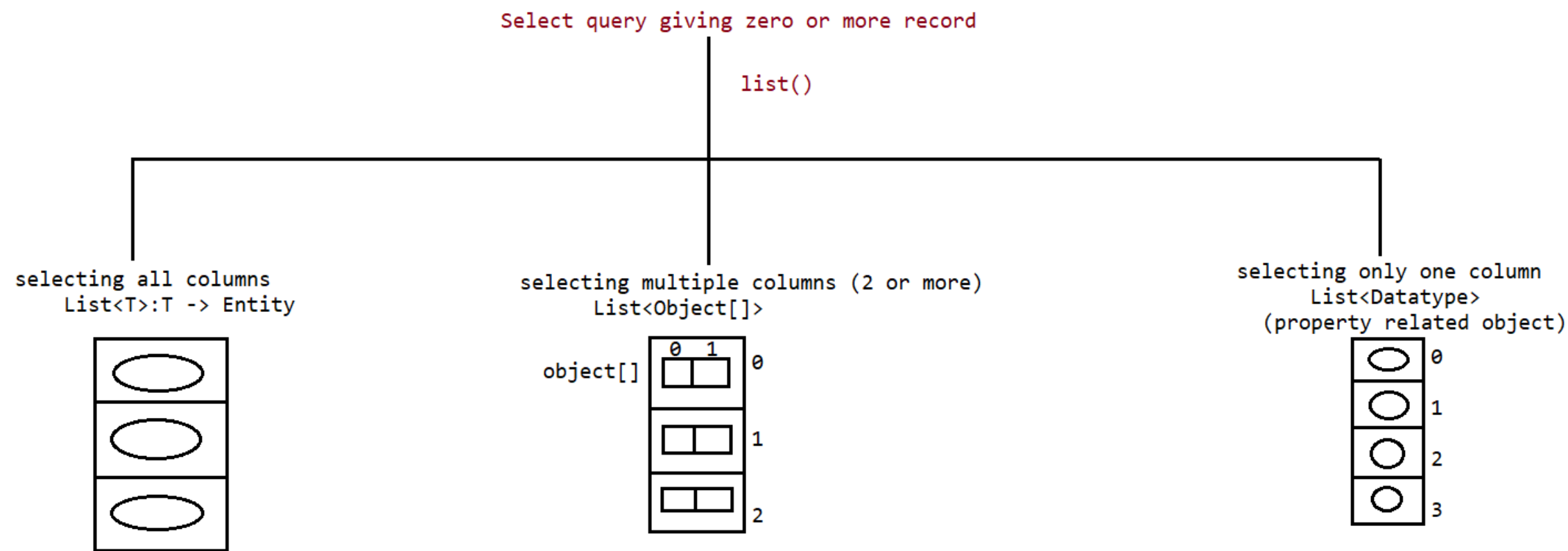
nitin.jpg
resume.txt

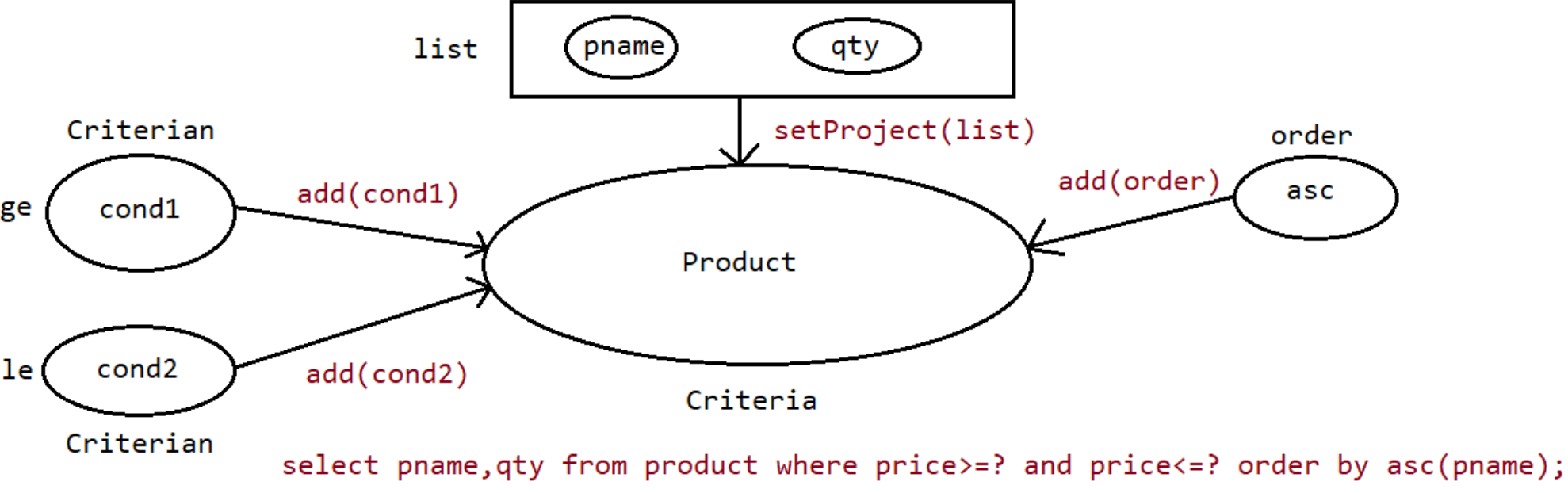
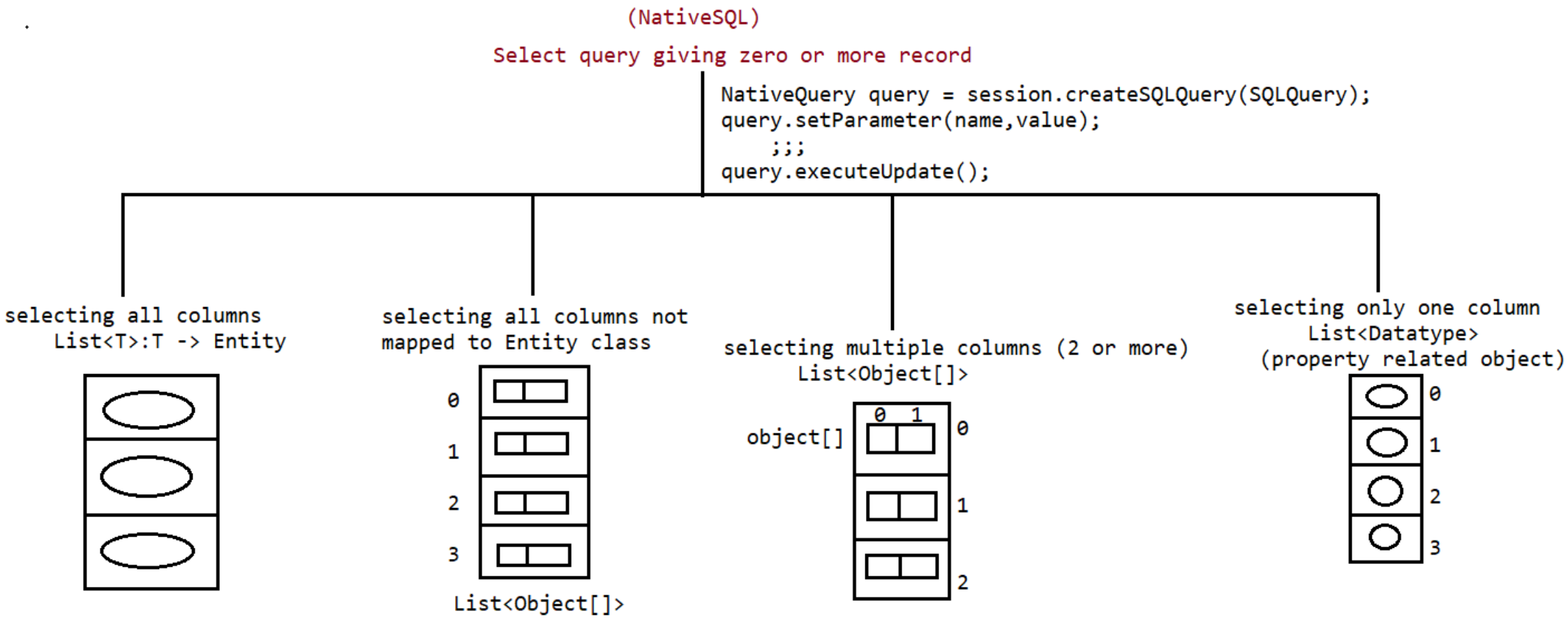
Locking in hibernate



Internally desinged on the basis of "Versioning".

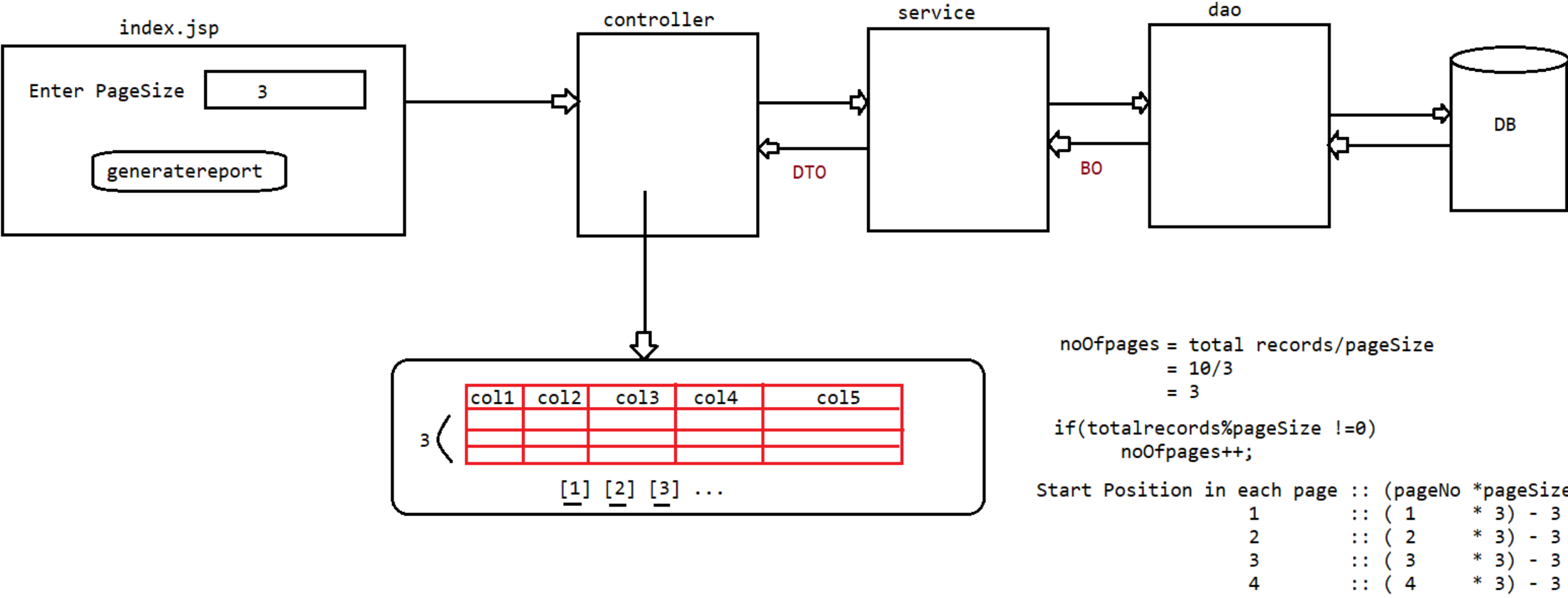






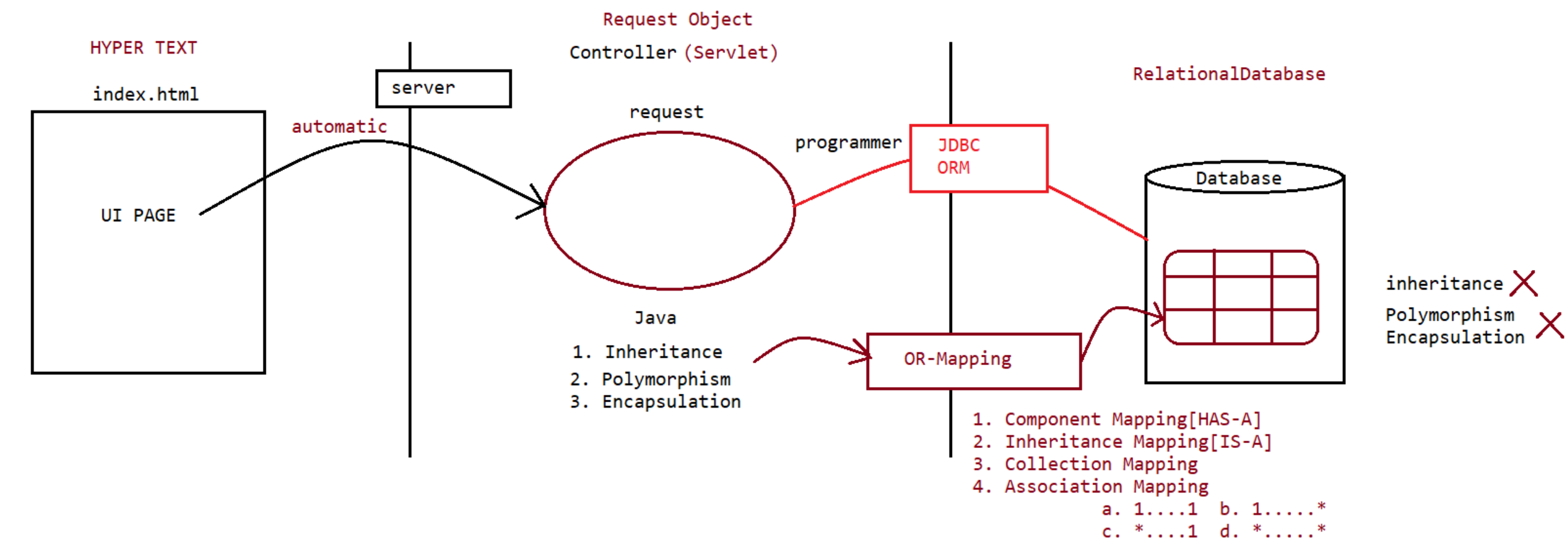
Pagination

page-1	0	1111	FamilyHealth	monthly	HDFC	23
	1	1234	JeevanAnand	quarterly	Prudential	35
	2	2211	JeevanJyothi	monthly	ICICI	24
page-2	3	2233	JeevanBhima	monthly	LIC	24
	4	3322	JeevanSuraksha	yearly	Kotak	20
	5	5555	RetirementFund	monthly	TATA	50
page-3	6	6666	SBI LIFE	quarterly	SBI	45
	7	7777	JeevanLabh	monthly	Bejaaj	35

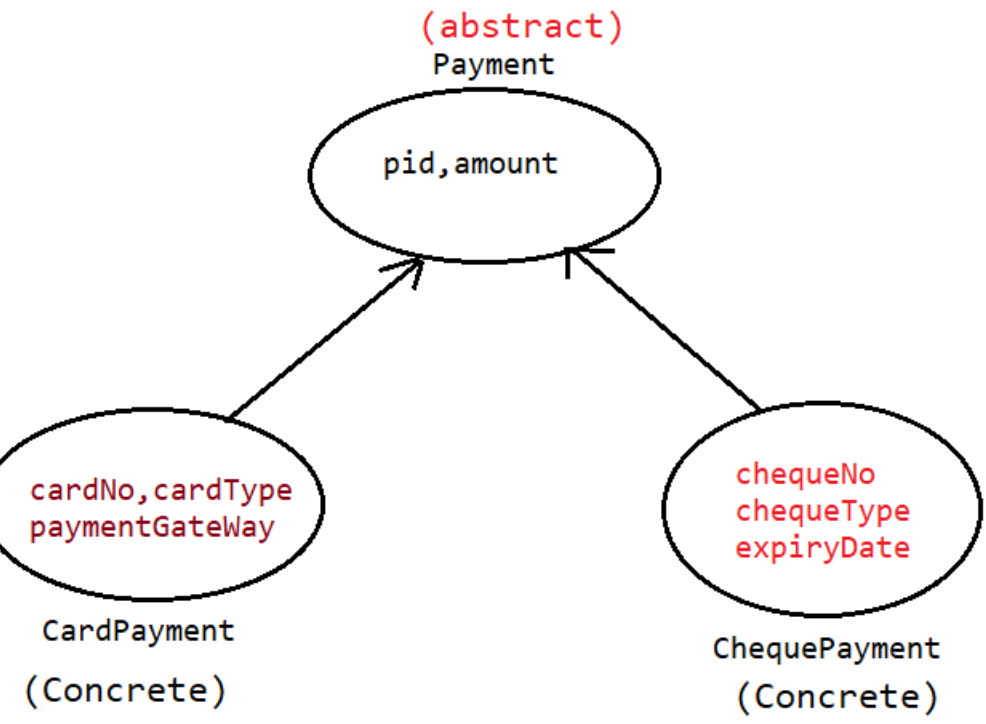


Pagination(Assuming pageSize= 3)

	0		1111	FamilyHealth	monthly	HDFC		23
page-1	1		1234	JeevanAnand	quarterly	Prudential		35
	2		2211	JeevanJyothi	monthly	ICICI		24
page-2	3		2233	JeevanBhima	monthly	LIC		24
	4		3322	JeevanSuraksha	yearly	Kotak		20
	5		5555	RetirementFund	monthly	TATA		50
page-3	6		6666	SBI LIFE	quarterly	SBI		45
	7		7777	JeevanLabh	monthly	Beajaj		35



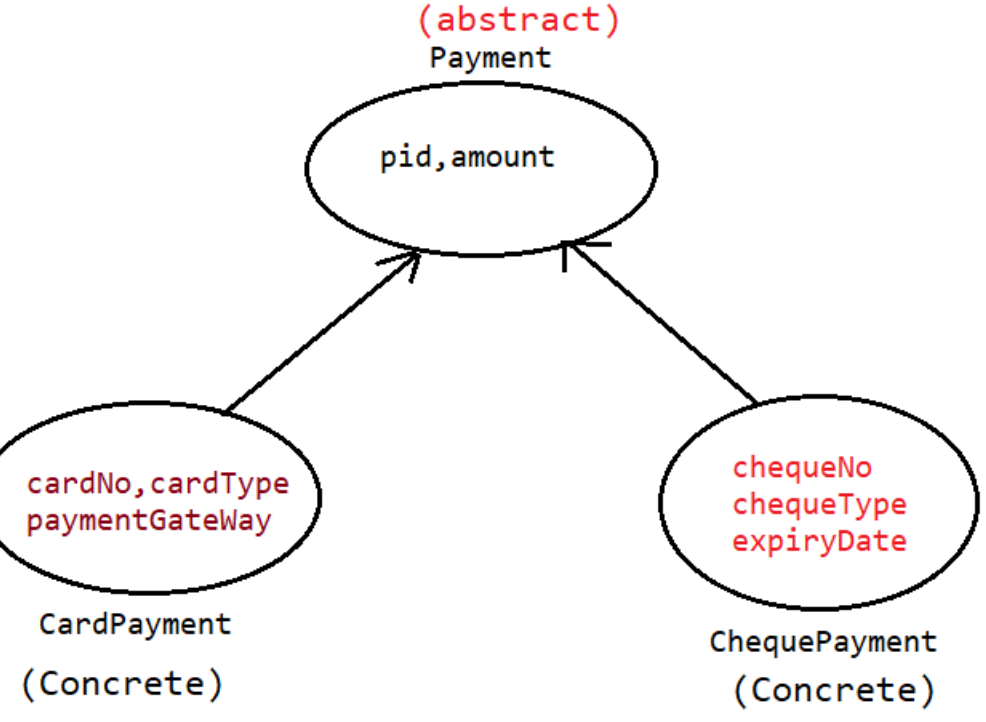
SINGLE_TABLE (DiscriminatorColumn)



Discriminatorcolumn

payment_mode	pid	amount	cardNo	cardType	paymentGateway	chequeNo	chequeType	expriyDate
CARD	1234	8000.5	45678	Debit	visa	(NULL)	(NULL)	(NULL)
CHEQUE	1235	4500.56	(NULL)	(NULL)	(NULL)	546345	self	0023-04-27

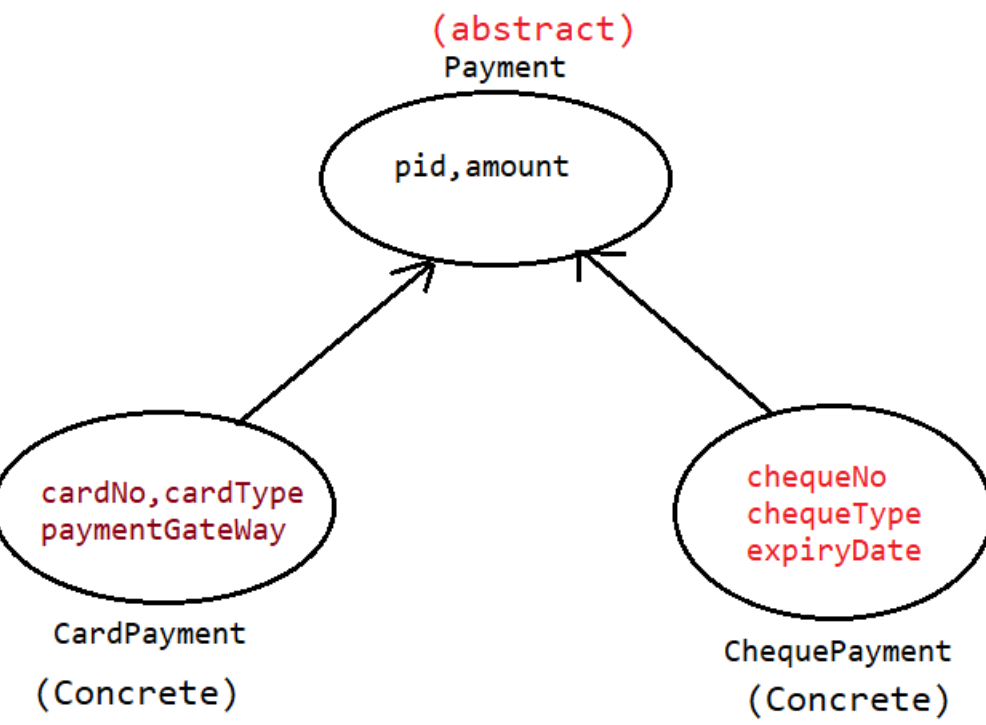
TABLE_PER_CLASS (1:1)



Discriminatorcolumn

CARDPAYMENT TBSC					PAYMENT_TBSC			
cardNo	cardType	paymentGateway	payment_id		payment_mode	pid	amount	
45678	Debit	visa	1234		CARD	1234	8000.5	
					CHEQUE	1235	4500.56	

TABLE PER CONCRETE CLASS



CHEQUE_PAYMENT_TBCC

pid	amount	chequeNo	chequeType	expriyDate
1235	4500.56	546345	self	0023-04-27

CARDPAYMENT_TBCC

pid	amount	cardNo	cardType	paymentGateway
1234	8000.5	45678	Debit	visa

CollectionMapping (1:M)

```
@ElementCollection
@Column(name="friend_name")
@CollectionTable(name = "EMP_FRNDS", joinColumns = @JoinColumn(name = "emp_id", referencedColumnName = "eid"))
@OrderColumn(name = "friend_no")
@ListIndexBase(value = 1)
private List<String> friendList;
```

EMP_FRNDS (fk)		
friend_no	friend_name	emp_id
1	saaurav	10
2	dravid	10
3	sehwagh	10

0	1	2
saruav	dravid	sehwagh

```
@ElementCollection
@Column(name="mobile_no")
@CollectionTable(name = "EMP_PHONE", joinColumns = @JoinColumn(name = "emp_id", referencedColumnName = "eid"))
private Set<Long> phones;
```

999888877	777666777	555666777
-----------	-----------	-----------

EMP_PHONE	
mobile_no	emp_id
999888877	10
777666777	10
555666777	10

eid	eaddress	ename
10	MI	sachin

```
@ElementCollection
@Column(name="account_number")
@CollectionTable(name = "EMP_ACCOUNTS", joinColumns = @JoinColumn(name = "emp_id", referencedColumnName = "eid"))
@MapKeyColumn(name = "bankName", length = 10)
private Map<String,Long> bankAccounts;
```

K	V
SBI	112233
HDFC	223344
ICICI	112244

bankName	account_number	emp_id
SBI	112233	10
HDFC	223344	10
ICICI	112244	10