

Hibernate

Matthias Colin

Summary

- ORM
- Java, JDBC, JPA, Hibernate
- Mapping
- CRUD
- Queries
- Spring
- Transaction
- Advanced Topics

ORM

Object Relational Mapper

- RDBMS (SGBDR)
 - 70s : Relational Algebra, SQL: IBM, Oracle
 - now :
 - Oracle Database
 - IBM DB2
 - Microsoft SQL Server
 - MySQL + MariaDB (fork 2009)
 - PostgreSQL
 - SQLite
 - Standard SQL
- OOP : UML + Languages : Java, Php, JavaScript, Python, .NET

m = new Movie("Gemini", 2019)

SQL :

insert into movie (title, year)
values ('Gemini', 2019)

Object model : class Movie
(id included)

table movie		
id	title	year
34	Gemini	2019

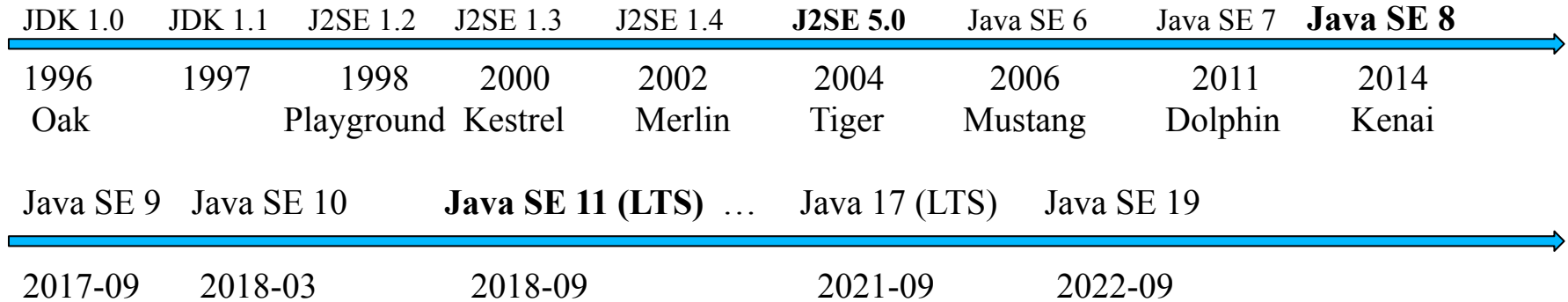
ORM

- mapping Object model vs Relational model
 - Class <-> table
 - attribute, property <-> column
 - association (reference) <-> association (FK -> PK)
 - multiplicity
 - information
 - Inheritance <-> ? (several techniques)
- CRUD
 - insert,update,delete : SQL auto with attributes (get/set + constructor)
 - queries :select
 - Object Vocabulary (class, attribute, association)
 - pseudo SQL : JPQL/HQL, ..., editor independent
 - object methods : Criteria
 - native SQL : editor dependent
- Examples: Hibernate (Java), Doctrine(php), Entity Framework (.NET), SQL Alchemy (python)

ORM Java

- langage Java SE
 - JDBC : Java DataBase Connectivity
 - package java.sql + javax.sql
 - interfaces
 - DataSource (pool)
 - Connection
 - Statement (DML or DDL)
 - PreparedStatement
 - CallableStatement
 - driver JDBC / editor
- framework JEE JPA (Java Persistence API)
 - Hibernate (JPA 2) sponsored by RedHat : hibernate.org

Java SE



Java EE

JDK 1.0	JDK 1.1	J2SE 1.2	J2SE 1.3	J2SE 1.4	J2SE 5.0	Java SE 6	Java SE 7	Java SE 8	Java SE 11
1996	1997	1998	2000	2002	2004	2006	2011	2014	2018
		J2EE 1.2	J2EE 1.3	J2EE 1.4	JEE 5	JEE 6	JEE 7	JEE 8	Jakarta EE
		1999	2001	2003	2006	2009	2013	2017	2018
Servlets :		2.2	2.3	2.4	2.5	3.0	3.1	4.0	
JSP :		1.1	1.2	2.0	2.1	2.2	2.3		
JSTL :			1.0	1.1	1.2	-	-		
EL :					2.1	2.2	3.0		
JPA :					1.0	2.0	2.1	2.2	
Bean validation :					1.0		2.0		
Outils :		JDBC		WS	JAXWS	JAXRS		JSON JSON-B	
		JNDI		JSF	SAAJ	WebProfile	WebSocket		

Serveurs

	J2EE 1.2	J2EE 1.3	J2EE 1.4	JEE 5	JEE 6	JEE 7	JEE 8	
	1999	2001	2003	2006	2009	2013	2017	
Servlet :	2.2	2.3	2.4	2.5	3.0	3.1	4.0	6.0
Tomcat :	3.0	4.1	5.0	6.0	7.0	8.0/8.5	9.0	10
(JSP, Servlet, EL)								
Glassfish :				1, 2	3	4	5.0	
Jboss/Wildfly :			4	5	6	7-11	12-15	
Websphere :		4	5	6	7	8	9	
Weblogic:								

Hibernate

- Versions
 - 4 (2011)
 - 5 (2015) : JPA 2.2 + JPA 3.0 : javax.persistence (package org.hibernate deprecated)
 - typing improved
 - api criteria (new version)
 - 6 (2022) : JPA 3.1 : Jakarta Persistence API

Hibernate : mapping

- 3 ways
 - OOP model => DB model
 - config XML
 - annotation
 - DB model => OOP model (generation auto)
 - Man in the middle : make a correspondance between 2 models
- simple/basic types
 - 8 primitive or Object (null included) types
 - int, Integer
 - long, Long
 - Id, primary key, strategy : @Id, @GeneratedValue
 - String, length, @Nationalized
 - enum @Enumerated
 - temporal types
 - computed field @Formula

Temporal types

Java

- `java.util.Date` (java 1.0)
- `java.util.Calendar` with impl `GregorianCalendar` (java 1.1)
- `java.time` (java 8)
 - `LocalDate`
 - `LocalDateTime`, `ZonedDateTime`
 - `LocalTime`
 - `Duration`
 - ...

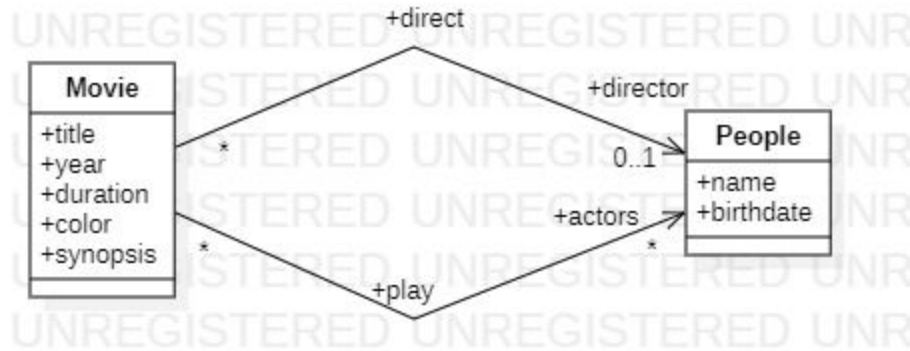
SQL : `Date`, `TimeStamp/DateTime`, `Time` (opt), `Interval`

Hibernate : `@Temporal` for old java types

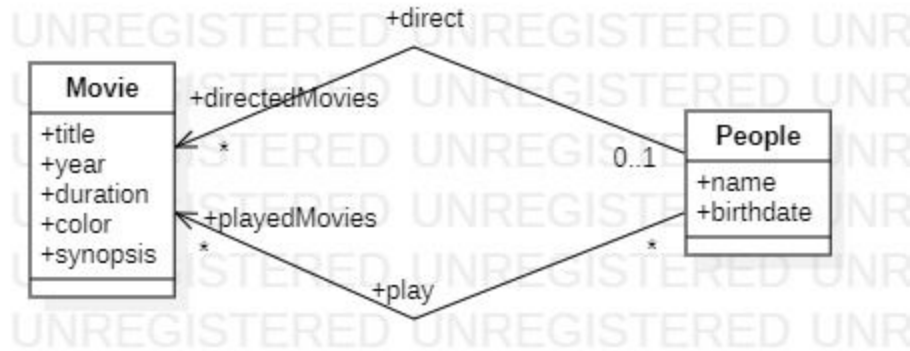
Multiple data

- association
 - multiplicity + navigability
 - One-to-one
 - fetch eager (default)
 - Many-to-one : movie -> director
 - fetch eager (default)
 - One-to-many : people -> directedMovies
 - fetch lazy (default)
 - Many-to-many : movie -> actors, people -> playedMovies
 - fetch lazy (default)
- collection of simple data

Unidirectional : Movie -> People



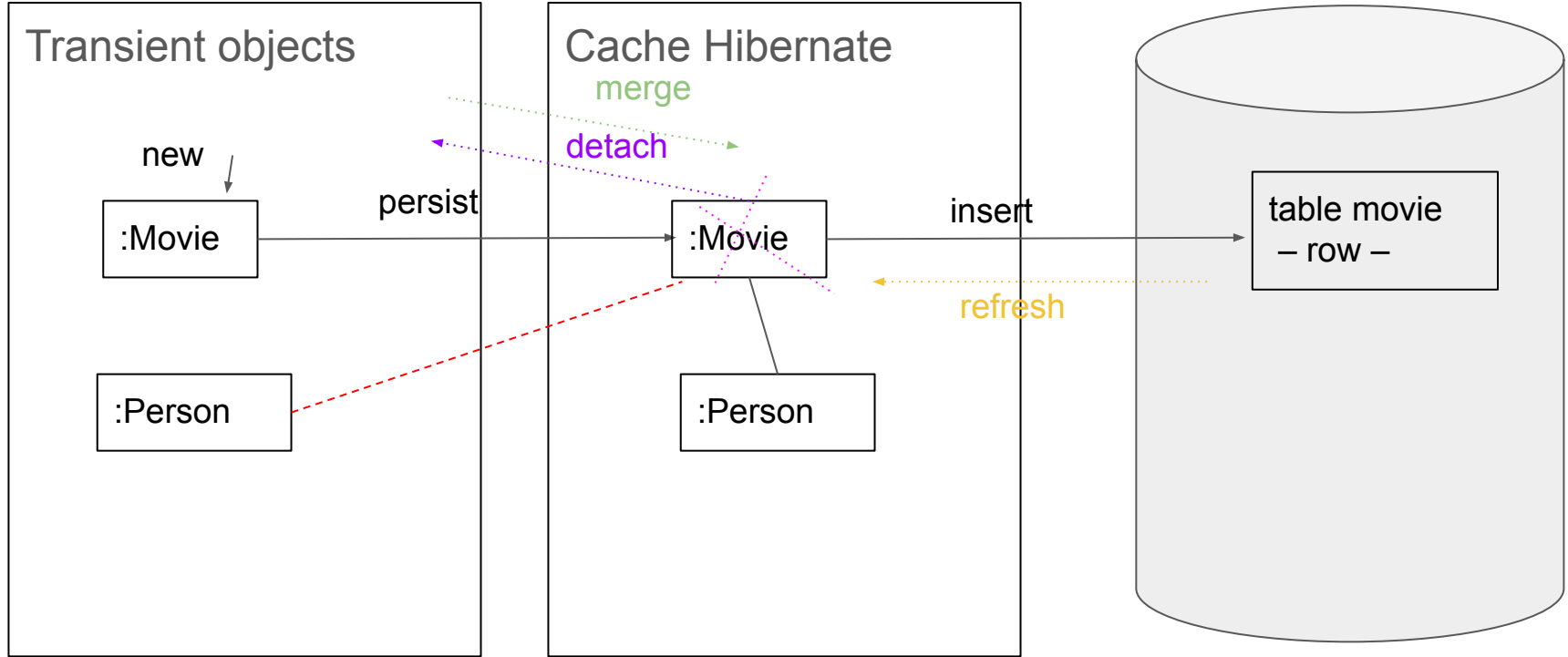
Unidirectional : People -> Movie



Hibernate cache

- JPA : class EntityManager
 - persist
- Hibernate (legacy) : class Session
 - save

Hibernate Cache



Queries

- Spring Data repositories
 - JPQL/HQL + Criteria + native SQL
 - auto SQL methods : `findByTitleAndYear(String title, Integer year)`
 - type checking on entities and keys : `MovieRepo impl JpaRepository<Movie, Integer>`
 - Example
 - Sort
 - Pageable
 - React mode
- Hibernate
 - JPQL/HQL : vendor independent, text
https://docs.jboss.org/hibernate/orm/5.6/userguide/html_single/Hibernate_User_Guide.html#hql
 - API Criteria : vendor independent, OOP (select, where, join, ...)
 - api Criteria v4 deprecated in v5
https://docs.jboss.org/hibernate/orm/5.6/userguide/html_single/Hibernate_User_Guide.html#appendix-legacy-criteria
 - api Criteria v5 new : type system improved
https://docs.jboss.org/hibernate/orm/5.6/userguide/html_single/Hibernate_User_Guide.html#criteria
 - native SQL : vendor dependent, text

Projection

- entity
- custom
 - Object[]
 - Tuple
 - DTO : class or interface (spring data only)