



# @OneToOne – main

# Cas d'une création d'une voiture

```
public static void main(String[] args) {  
  
    . . .  
  
    // créer un objet voiture et un moteur  
    Voiture v= new Voiture ("logan","111-aa-11");  
    Moteur m = new Moteur(90, "essence", 1500);  
  
    //associer les deux objets  
    v.setMoteur(m);  
  
    // récupérer une session ouvrir une transaction  
    session = factory.getCurrentSession();  
    session.beginTransaction();  
  
    session.save(v);  
  
    //commit transaction  
    session.getTransaction().commit();  
    . . .  
}
```



# Créer la bdd

Executer le script create-db.sql

```
DROP SCHEMA IF EXISTS `bdd_one_to_one_uni`;
CREATE SCHEMA `bdd_one_to_one_uni`;

USE `bdd_one_to_one_uni`;

SET FOREIGN_KEY_CHECKS = 0;

DROP TABLE IF EXISTS `voiture`;
DROP TABLE IF EXISTS `moteur`;

-- Table structure for table `moteur`
CREATE TABLE `moteur` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `puissance` INT DEFAULT NULL,
  `carburant` varchar(10) DEFAULT NULL,
  `cylindree` INT DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

-- Table structure for table `voiture`
CREATE TABLE `voiture` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `modele` varchar(45) DEFAULT NULL,
  `immatriculation` varchar(10) DEFAULT NULL,
  `moteur_id` INT DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_moteur_idx` (`moteur_id`),
  CONSTRAINT `FK_to_moteur_id` FOREIGN KEY (`moteur_id`)
  REFERENCES `moteur` (`id`)
  ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

SET FOREIGN_KEY_CHECKS = 1;
```

Création  
de la FK



Administration Schemas create-db

Limit to 1000 rows

SCHEMAS

Filter objects

- bdd\_hibernate\_demo
- bdd\_hibernate\_tp10
- bdd\_one\_to\_one\_uni
  - Tables
    - moteur
      - Columns
        - id
        - puissance
        - carburant
        - cylindree
      - Indexes
      - Foreign Keys
      - Triggers
    - voiture
      - Columns
        - id
        - modele\_name
        - immatriculation
        - moteur\_id
      - Indexes
      - Foreign Keys
      - Triggers

```
1 • DROP SCHEMA IF EXISTS `bdd_one_to_one_uni`;
2 • CREATE SCHEMA `bdd_one_to_one_uni`;
3
4 • USE `bdd_one_to_one_uni`;
5
6 • SET FOREIGN_KEY_CHECKS = 0;
7
8
9
10 -- Table structure for table `moteur`
11 • DROP TABLE IF EXISTS `moteur`;
12
13 • CREATE TABLE `moteur` (
14
```

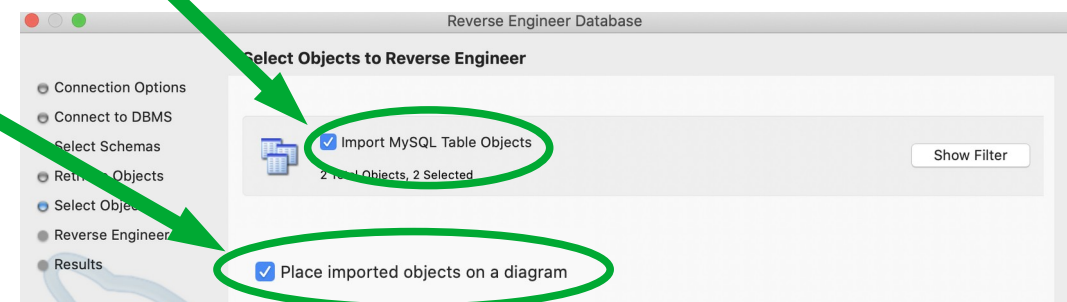
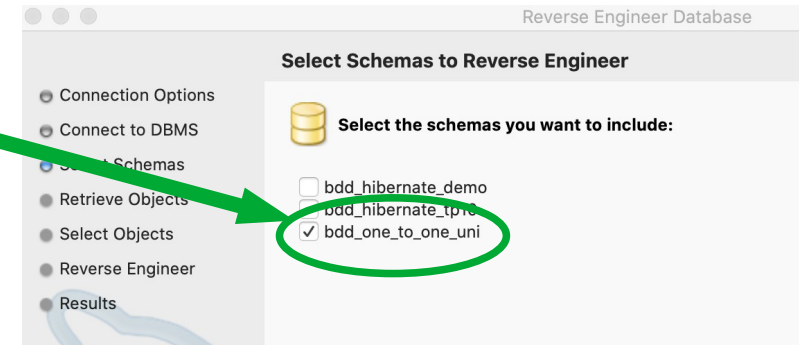
100% 1:5

Action Output

	Time	Action
✓ 1	21:14:49	DROP SCHEMA IF EXISTS `bdd_one_to_one_uni`
✓ 2	21:14:49	CREATE SCHEMA `bdd_one_to_one_uni`
✓ 3	21:14:49	USE `bdd_one_to_one_uni`
✓ 4	21:14:49	SET FOREIGN_KEY_CHECKS = 0
⚠ 5	21:14:49	DROP TABLE IF EXISTS `moteur`
✓ 6	21:14:49	CREATE TABLE `moteur` ( `id` INT NOT NULL AUTO_INCREM
⚠ 7	21:14:49	DROP TABLE IF EXISTS `voiture`
✓ 8	21:14:49	CREATE TABLE `voiture` ( `id` INT NOT NULL AUTO_INCREM

# Reverse Engineer

- Il est possible de générer le diagramme de classe
- Dans l'onglet DataBase de MySQL Workbench
- >Reverse Engineer (lire les tables et générer un diagramme représentatif)
- >Stored Connection select "padawan">Continue
- >Continue (encore)>choisir le schema
- >Continue>Continue
- >Assurez vous que l'import est coché  
ainsi que "Place imports ..."
- >Execute
- >Continue > Close



# De quoi documentez votre application , c'est cadeau.

