

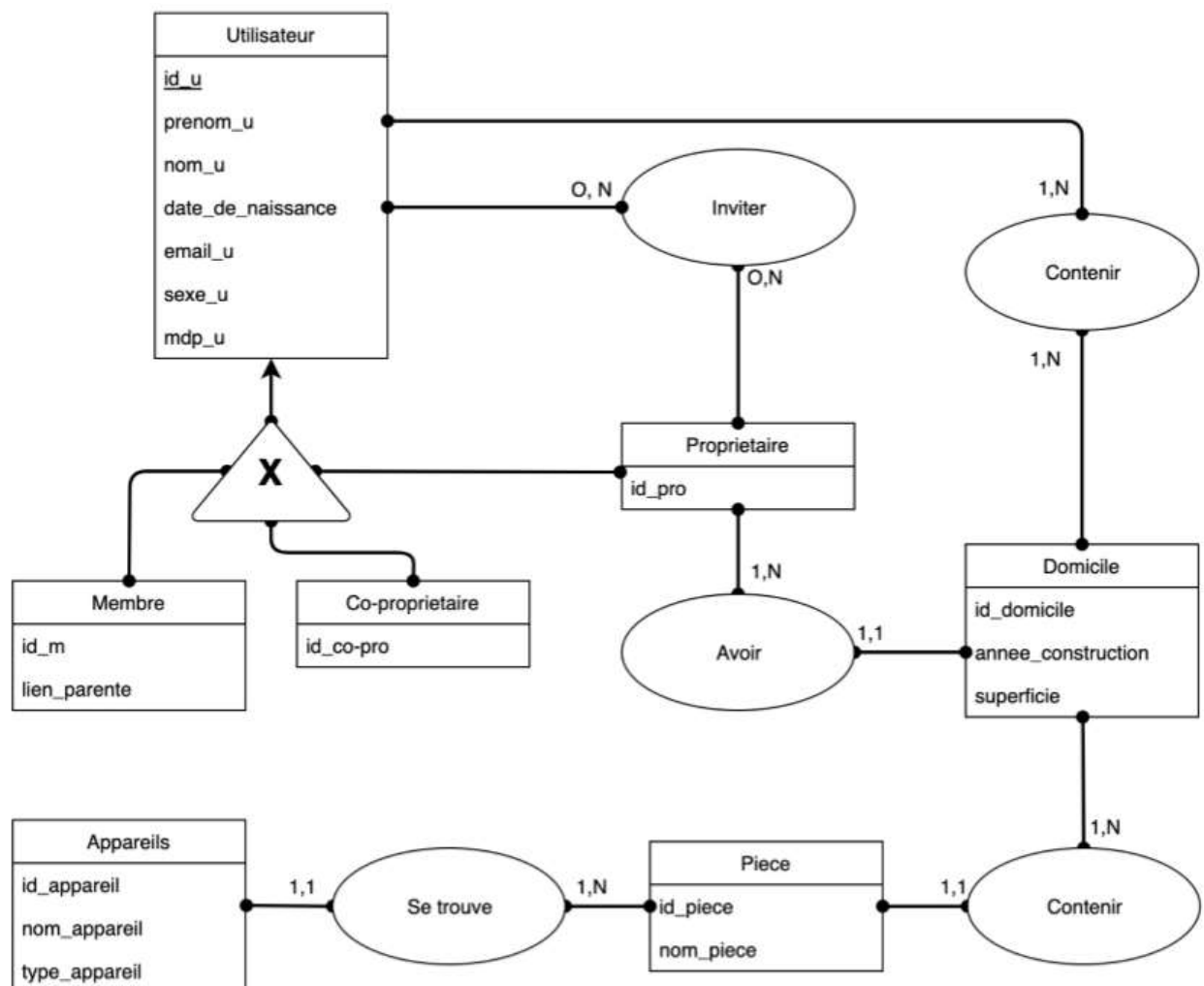
Nom : RABOBA

Prénom : Tokinantenaina Anthonio Sandrica

Exercice :

PROJET JARVIS partie 2

Schéma MCD :



1. Diagramme EER

a. MCD 'entité – entité' => MLD

- Utilisateur - (Contenir) - Domicile =>

Utilisateur (id_u, #id_domicile, prenom_u, nom_u, date_de_naissance_u, email_u, sexe_u, mdp_u,) ;

Domicile (id_domicile, annee_domicile, superficie) ;

- Domicile - (Contenir2) - Pièce =>

Piece (id_piece, #id_domicile, nom_piece) ;

Domicile (id_domicile, annee_domicile, superficie) ;

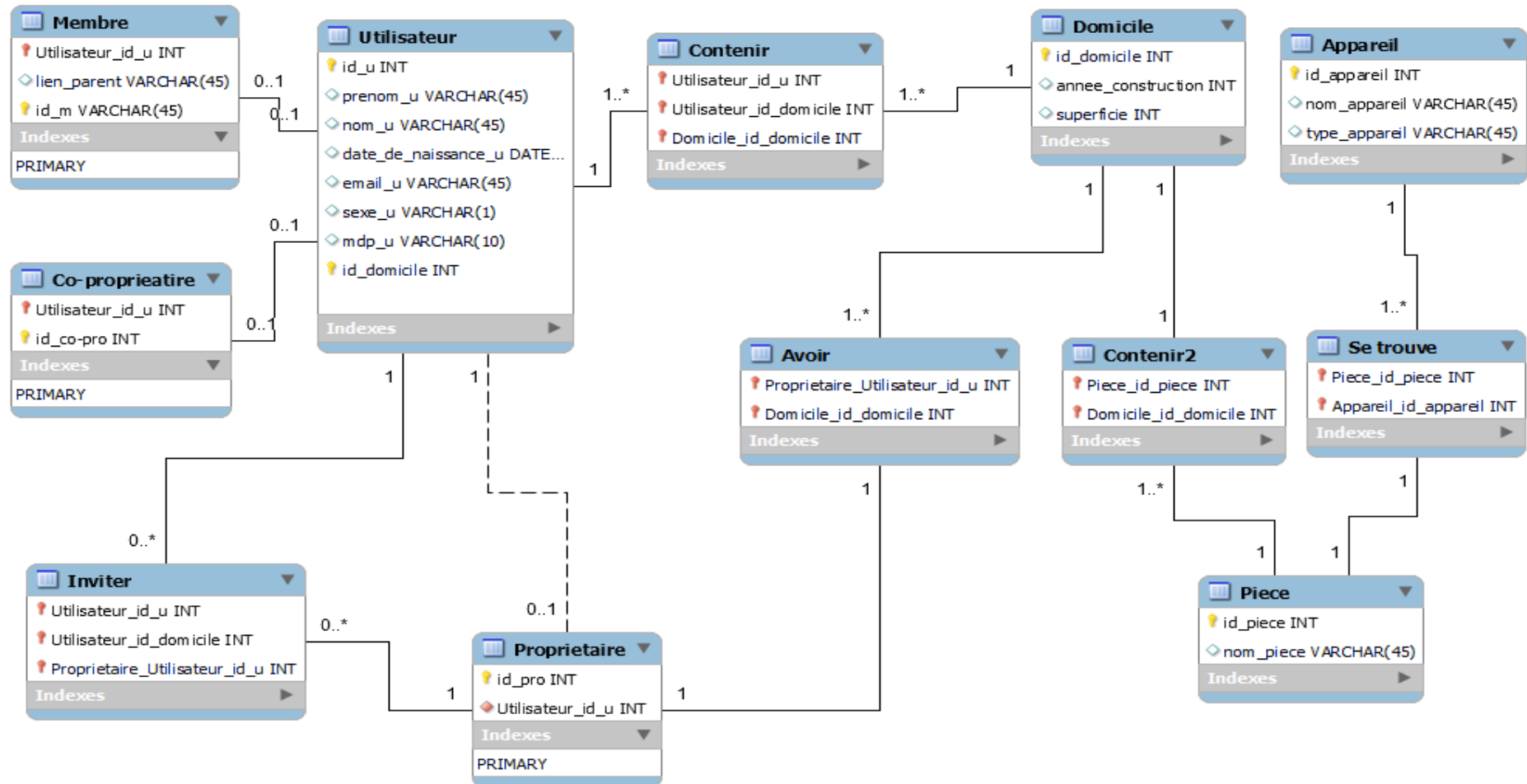
- Piece - (Se trouve) - appareil =>

Appareil (id_appareil, #id_piece, nom_appareil, type_appareil)

Piece (id_piece, #id_domicile, nom_piece) ;

b . Le diagramme EER

[EER Jarvis2.mwb](#)



2. Les requêtes

a. Création de la base de donnée Jarvis2 :

```
CREATE SCHEMA IF NOT EXISTS `Jarvis2` DEFAULT CHARACTER SET utf8
```

b. Création des tables dans la base de donnée :

utilisateur, proprietaire, co-proprietaire, membre, domicile, piece, appareil

b.1- utilisateurs (id_u, #id_domicile, prenom_u, nom_u, date_de_naissance_u, email_u, sexe_u, mdp_u, lien_parente)

```
CREATE TABLE IF NOT EXISTS `jarvis2`.`utilisateur` (  
  `id_u` INT(11) NOT NULL,  
  `prenom_u` VARCHAR(45) NULL DEFAULT NULL,  
  `nom_u` VARCHAR(45) NULL DEFAULT NULL,  
  `date_de_naissance_u` DATE NULL DEFAULT NULL,  
  `email_u` VARCHAR(45) NULL DEFAULT NULL,  
  `sexe_u` VARCHAR(1) NULL DEFAULT NULL,  
  `mdp_u` VARCHAR(10) NULL DEFAULT NULL,  
  `id_domicile` INT(11) NOT NULL DEFAULT '0',  
  `lien_parente` INT(1) NULL DEFAULT '0',  
  PRIMARY KEY (`id_u`, `id_domicile`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8
```

b.2- propriétaire (id_pro, #Utilisateur_id_u)

```
CREATE TABLE IF NOT EXISTS `jarvis2`.`proprietaire` (  
  `id_pro` INT(11) NOT NULL AUTO_INCREMENT,  
  `Utilisateur_id_u` INT(11) NOT NULL,  
  PRIMARY KEY (`id_pro`),  
  INDEX `Utilisateur_id_u_idx` (`Utilisateur_id_u` ASC) VISIBLE,  
  CONSTRAINT `fk_Proprio_Utilisateur`  
    FOREIGN KEY (`Utilisateur_id_u`)  
    REFERENCES `jarvis2`.`utilisateur` (`id_u`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8
```

b.3- co-proprietaire (id_co-pro, #Utilisateur_id_u, lien_parente)

```
CREATE TABLE IF NOT EXISTS `jarvis2`.`co-proprietaire` (  
  `id_co-pro` INT(11) NOT NULL AUTO_INCREMENT,  
  `Utilisateur_id_u` INT(11) NOT NULL,  
  `lien_parente` INT(1) NULL DEFAULT NULL,  
  PRIMARY KEY (`id_co-pro`),  
  INDEX `fk_Co-proprietaire_Utilisateur` (`Utilisateur_id_u` ASC) VISIBLE,  
  CONSTRAINT `fk_Co-proprietaire_Utilisateur`  
    FOREIGN KEY (`Utilisateur_id_u`)  
      REFERENCES `jarvis2`.`utilisateur` (`id_u`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
AUTO_INCREMENT = 2  
DEFAULT CHARACTER SET = utf8
```

b.4- membre (**id_m**, Utilisateur_id_u, lien_parente)

```
CREATE TABLE IF NOT EXISTS `jarvis2`.`membre` (  
  `id_m` INT(11) NOT NULL AUTO_INCREMENT,  
  `Utilisateur_id_u` INT(11) NOT NULL DEFAULT '0',  
  `lien_parente` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`id_m`),  
  INDEX `fk_Membre_Utilisateur` (`Utilisateur_id_u` ASC) VISIBLE,  
  CONSTRAINT `fk_Membre_Utilisateur`  
    FOREIGN KEY (`Utilisateur_id_u`)  
      REFERENCES `jarvis2`.`utilisateur` (`id_u`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
AUTO_INCREMENT = 19  
DEFAULT CHARACTER SET = utf8
```

b.5 - domicile (**id_domicile**, annee_domicile, superficie)

```
CREATE TABLE IF NOT EXISTS `jarvis2`.`domicile` (  
  `id_domicile` INT(11) NOT NULL,  
  `annee_construction` INT(11) NULL DEFAULT NULL,  
  `superficie` INT(11) NULL DEFAULT NULL,  
  PRIMARY KEY (`id_domicile`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8
```

b.6 – appareil (**id_appareil**, #**id_piece**, **nom_appareil**, **type_appareil**)

```
CREATE TABLE IF NOT EXISTS `jarvis2`.`appareil` (  
  `id_appareil` INT(11) NOT NULL AUTO_INCREMENT,  
  `nom_appareil` VARCHAR(45) NULL DEFAULT NULL,  
  `type_appareil` VARCHAR(45) NULL DEFAULT NULL,  
  `id_piece` INT(11) NOT NULL,  
  PRIMARY KEY (`id_appareil`),  
  INDEX `fk_Appareil_piece_idx` (`id_piece` ASC) VISIBLE,  
  CONSTRAINT `fk_Appareil_piece`  
    FOREIGN KEY (`id_piece`)  
      REFERENCES `jarvis2`.`piece` (`id_piece`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB  
AUTO_INCREMENT = 6  
DEFAULT CHARACTER SET = utf8
```


- c. Création d'une view synthétisant user, membre, co-proprétaire et propriétaire

```
CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `status_utilisateur` AS

SELECT

  `utilisateur`.`nom_u` AS `nom`,

  `utilisateur`.`prenom_u` AS `prenom`,

  `utilisateur`.`email_u` AS `email`,

  (CASE

    WHEN (`utilisateur`.`lien_parente` = 0) THEN 'membre'

    WHEN (`utilisateur`.`lien_parente` = 1) THEN 'co-proprio'

  END) AS `status`

FROM

  ((`utilisateur`

  JOIN `membre`)

  JOIN `co-propretaire`)
```

- d. Liste :

- d.1 - Toutes les pièces d'un domicile

```
SELECT nom_piece FROM piece

where piece.id_domicile = (<{id_domicile}>);
```

exemple :

```
SELECT nom_piece FROM piece

where piece.id_domicile = 5;
```

d.2- Lister tous les appareils d'un domicile

```
SELECT nom_appareil, type_appareil, nom_piece, appareil.id_piece, id_domicile
FROM jarvis2.piece,
jarvis2.appareil
where appareil.id_piece = piece.id_piece and id_domicile = (<{id_domicile}>);
```

exemple :

```
SELECT nom_appareil, type_appareil, nom_piece, appareil.id_piece, id_domicile
FROM jarvis2.piece,
jarvis2.appareil
where appareil.id_piece = piece.id_piece and id_domicile = 5;
```

d.3- Lister tous les utilisateurs d'un domicile

```
SELECT nom_u, prenom_u, email_u FROM jarvis2.utilisateur
where id_domicile = (<{id_domicile}>) ;
```

exemple :

```
SELECT nom_u, prenom_u, email_u FROM jarvis2.utilisateur
where id_domicile = 2;
```

d.4- Lister les propriétaires et co-propriétaires

```
SELECT nom, prenom, email
FROM status_utilisateur where `status` = 'proprietaire' | 'co-proprio';
NB : status_utilisateur est table 'create view' qu'on avait determiner au
question d.1-
```

d.5- Lister tous les appareils d'une pièce

```
SELECT nom_appareil, type_appareil FROM appareil where id_piece = (<{id_domicile}>);
```

Exemple :

```
SELECT nom_appareil, type_appareil FROM appareil where id_piece = 2 ;
```

d.6 - Lister toutes les invitations en attentes