

## TP3 Base de données – ESIEA

### Exercice 1 – Importer des Tables du schéma HR

Créer une connexion vers le schéma HR. Avec cette connexion, importez les Tables EMPLOYEES et DEPARTMENTS dans SQL Developer Data Modeler

Créer / Sélectionner une connexion de base de données

Nom de connexion	Détails de connexion
Angelo_Connexion	ANGELO@//192.168.1.132
hr_connexion	HR@//192.168.1.132
Initialisation	SYSTEM@//192.168.1.132
XE	SYSTEM@//192.168.1.132

Name:

Type de base de données:

**Informations utilisateur** Utilisateur proxy

Type d'authentification:

Nom utilisateur:  Rôle:

Mot de passe:  ☒ Enregistrer le mot de passe

Type de connexion:

**Détails** Avancé

Nom d'hôte:

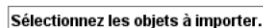
Port:

☐ SID

☒ Nom de service:

Statut : succès

Aide Enregistrer Effacer **Tester** Connexion Annuler



#### 4. Générez la conception.

Sélectionné	Schéma	Nom d'objet
<input type="checkbox"/>	HR	COUNTRIES
<input checked="" type="checkbox"/>	HR	DEPARTMENTS
<input checked="" type="checkbox"/>	HR	EMPLOYEES
<input type="checkbox"/>	HR	JOB_HISTORY
<input type="checkbox"/>	HR	JOBS
<input type="checkbox"/>	HR	LOCATIONS
<input type="checkbox"/>	HR	REGIONS

Tables	Vues	Utilisateurs	Rôles	Répertoires	Tables externes	Journaux de vue matérialisée	Contextes	Clusters
Séquences		Synonymes		Tablespaces		Tablespaces temp.		Dimensions
Package	Procédures stockées		Fonctions	Tablespaces d'annulation (undo)				

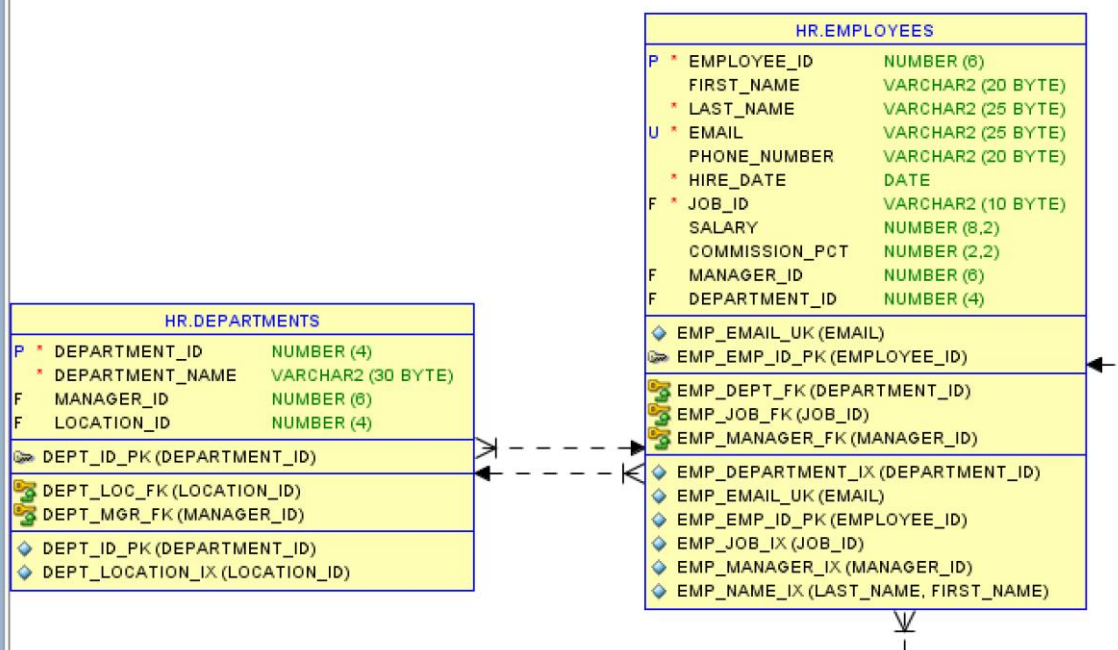
Filtre :

[< Précédent](#)

Suivant >

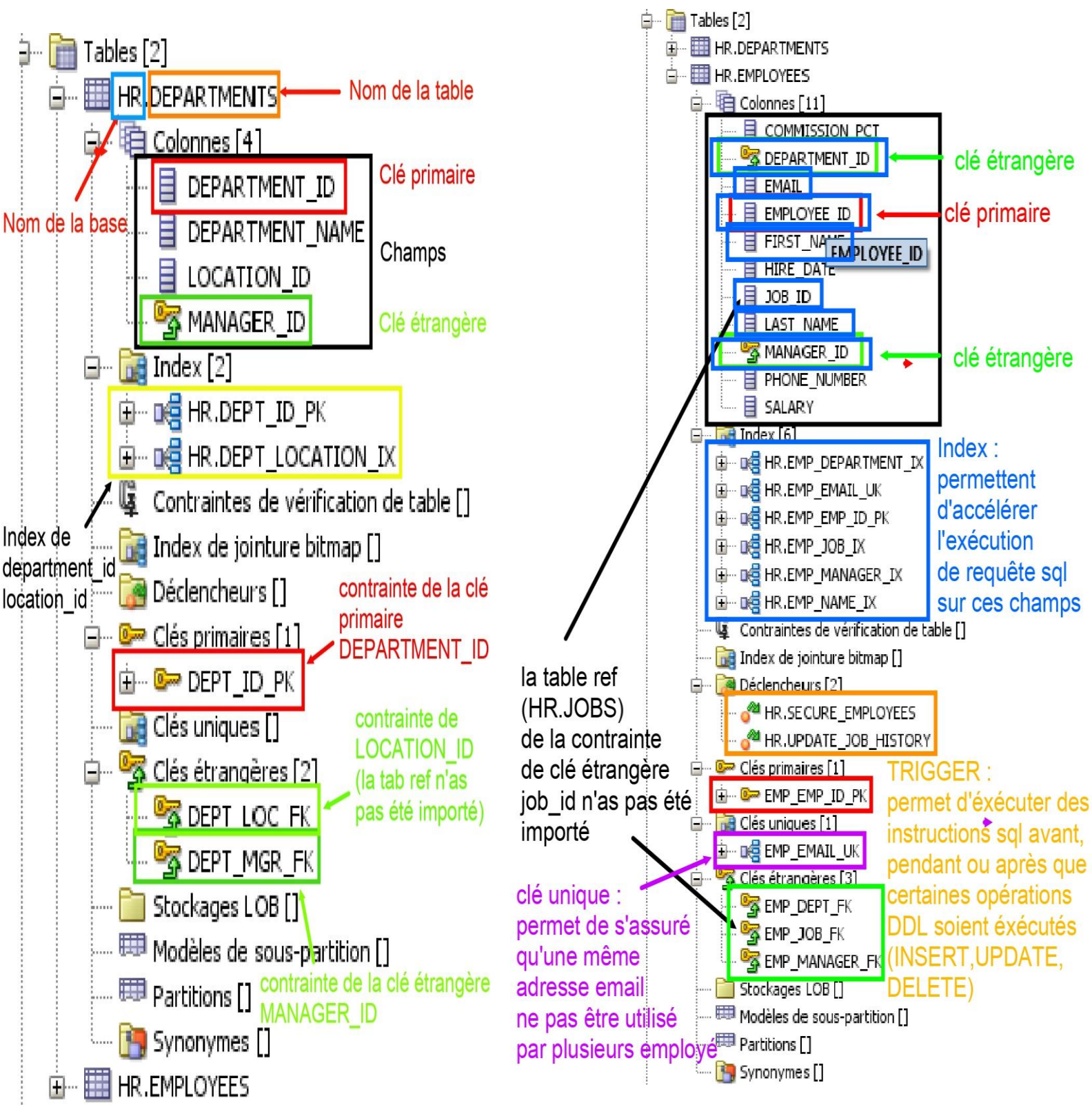
Annuler

Aide



## Exercice 2 – Examen du Modèle Physique

^fggfgfgkjsdshgssetuifdghggds.



## Exercice 3 – Créer une nouvelle table

Créer une nouvelle table PROJECTS. Cette table a quatre colonnes: PROJECT\_ID, PROJECT\_NAME, PROJECT\_START\_DATE et PROJECT\_END\_DATE. Ajoutez les colonnes suivantes : PROJECT\_ID, type NUMERIC, précision 6, clé primaire  
PROJECT\_NAME, type VARCHAR longueur 100 PROJECT\_START\_DATE, type DATE  
PROJECT\_END\_DATE, type DATE

Propriétés de la table - TABLE\_4

Général

Nom: PROJECTS

Nom complet: TABLE\_4

Abréviation:

Régénérer: ☒

Nom de clé primaire:

Basé sur le type structuré:

Schéma:

Enregistrer en tant que table spatiale: ☐

L'identificateur d'objet est une clé primaire: ☐

Autoriser la substitution de type: ☒

Générer dans le DDL: ☒

Régénérer en tant que relation: ☐

Autoriser la réorganisation des colonnes lors de la régé...: ☐

En phase d'abandon: ☐

Utiliser le script de transformation DDL: ☐

Table de requête matérialisée: ☐ Sur la table préconstruite ☐

Archivage de ligne: ☐

OK Appliquer Règles de dénomination Annuler Aide

Propriétés de la table - TABLE\_4

Colonnes

Détails Aperçu Sécurité UDP

Colonnes :

Nom	Type de données
1 PROJECT_ID	NUMBER (6)
2 PROJECT_NAME	VARCHAR2 (100)
3 PROJECT_START_D...	DATE
4 PROJECT_END_DATE	DATE

Propriétés de la colonne

Nom: PROJECT\_ID

Type de données: ☐ Domaine ☒ Logique ☐ Distinct

☐ Structuré ☐ Ensemble

Type de source: NUMERIC Enregistré ☐

Précision: 6

Echelle:

☒ Clé primaire ☐ Clé étrangère ☒ Obligatoire ☐ En phase d'abandon

Commentaires dans le SGBDR: Commentaires Remarques

OK Appliquer Règles de dénomination Annuler Aide



Propriétés de la table - TABLE\_4

Général  
**Colonnes**  
 Clé primaire  
 Contraintes uniques  
 Index  
 Contraintes de niveau table  
 Dépendances d'existence  
 Clés étrangères  
 Colonnes imbriquées  
 Dimensions Temps valides  
 Requête matérialisée  
 Propriétés du volume  
 Propriétés spatiales  
 Groupes de colonnes  
 Commentaires  
 Commentaires dans le SGBDR  
 Remarques  
 Analyse d'impact  
 Mesures  
 Demandes de modification  
 Responsables  
 Documents  
 Scripts  
 Propriétés dynamiques  
 Propriétés définies par l'utilisateur  
 Stratégie de protection par occ  
 Types de classification  
 Services REST  
 Récapitulatif

Colonnes

Nom	Type de données
1 PROJECT_ID	NUMBER (6)
2 PROJECT_NAME	VARCHAR2 (100)
3 PROJECT_START_D...	DATE
4 PROJECT_END_DATE	DATE

Propriétés de la colonne

Nom: PROJECT\_END\_DATE

Type de données: ☒ Logique ☐ Distinct ☐ Ensemble

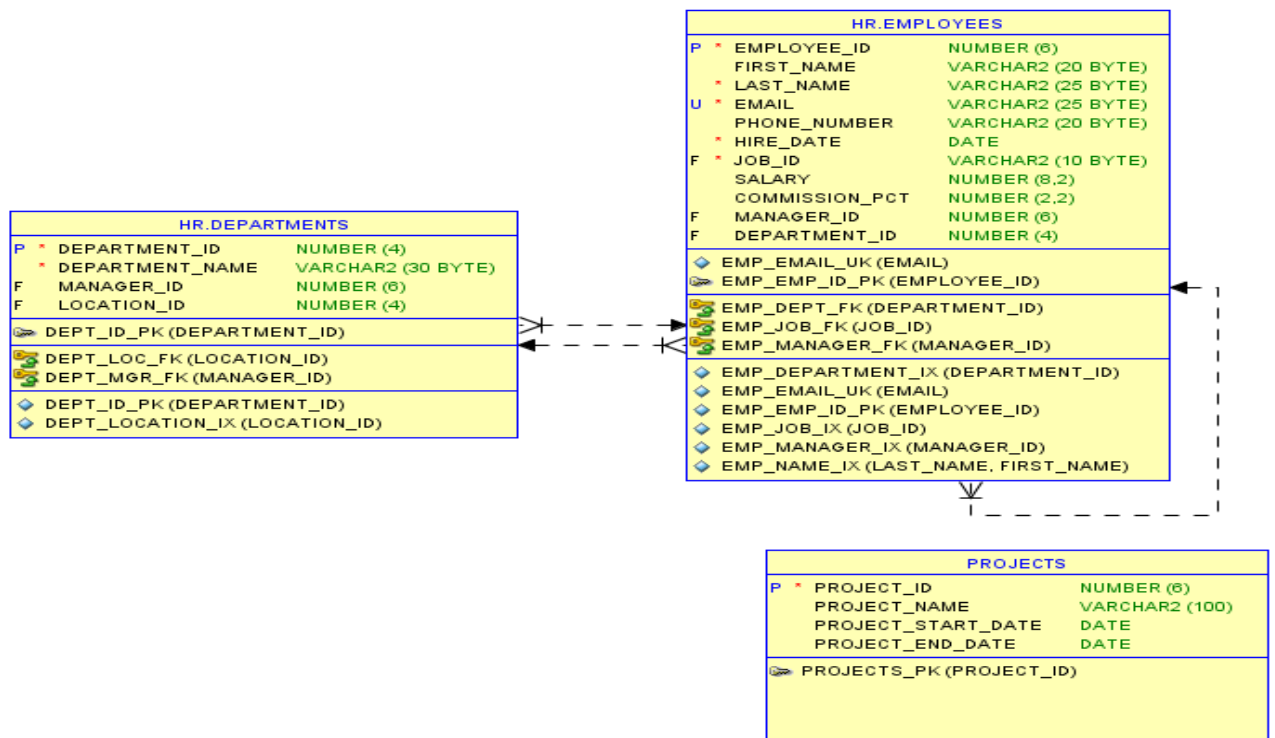
Type de source: Date Enregistré ☐

☐ Clé primaire ☐ Clé étrangère ☐ Obligatoire ☐ En phase d'abandon

Commentaires dans le SGBDR

Commentaires Remarques

OK Appliquer Règles de dénomination Annuler Aide





## Exercice 4 – Associer la nouvelle table avec les autres

Créer une clé étrangère entre les Tables PROJECTS et DEPARTMENTS.  
Dé sélectionnez la case à cocher « Obligatoire » et cliquez sur OK.

Propriétés de la clé étrangère - PROJECTS\_DEPARTMENTS\_FK

Général

Nom: PROJECTS\_DEPARTMENTS\_FK Table: PROJECTS

Index de clé primaire/clé unique: DEPARTMENTS.DEPT\_ID\_PK Règle de suppression: NO ACTION

Synonyme de table source: DEPARTMENTS Synonyme de table cible: PROJECTS

Obligatoire: ☐ En phase d'abandon: ☐

Transférable (mise à jour possible): ☒ Générer dans le DDL: ☒

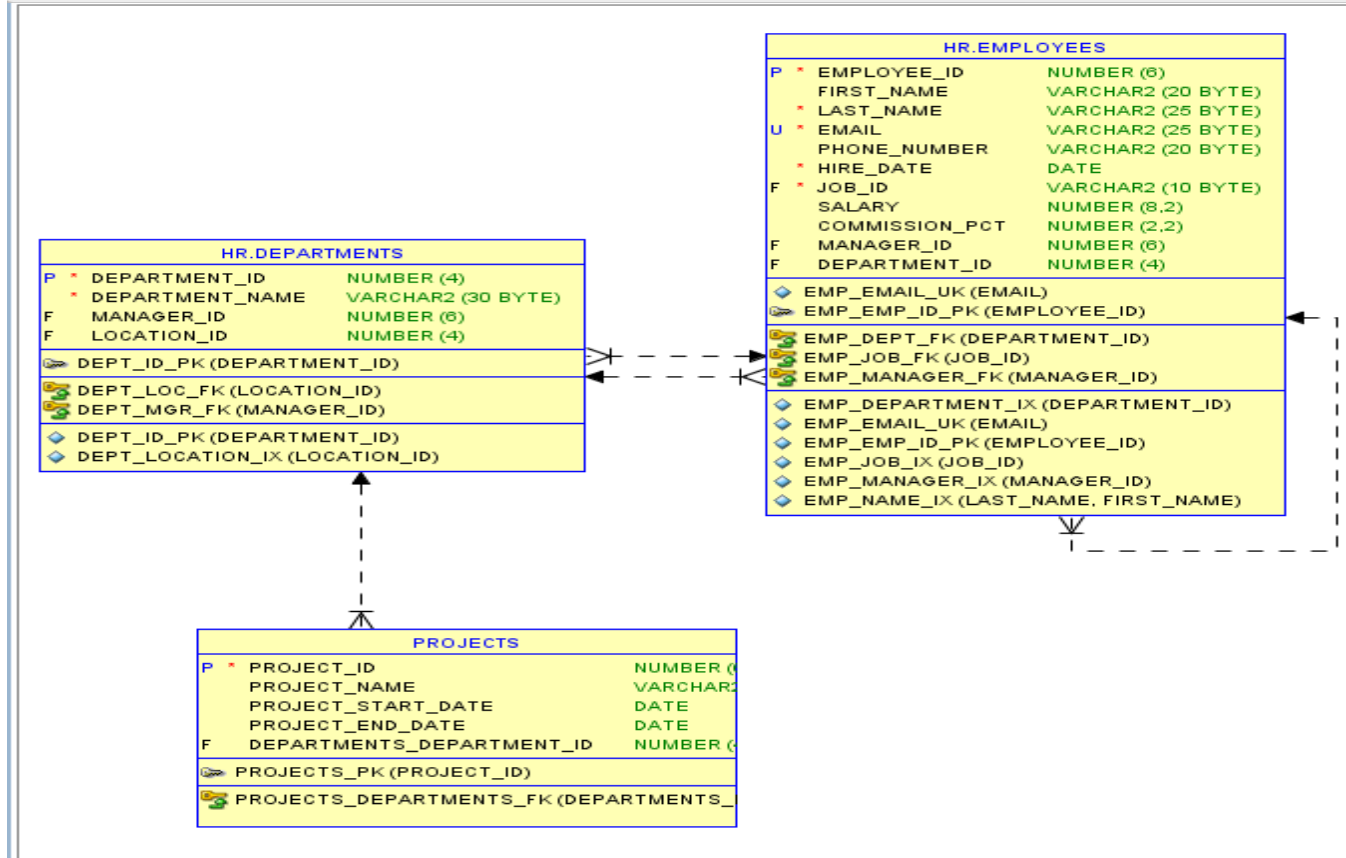
Dans l'arc: ☐ Dépendance: NO

Colonne de discrimination: Valeur de discriminateur:

Colonnes associées

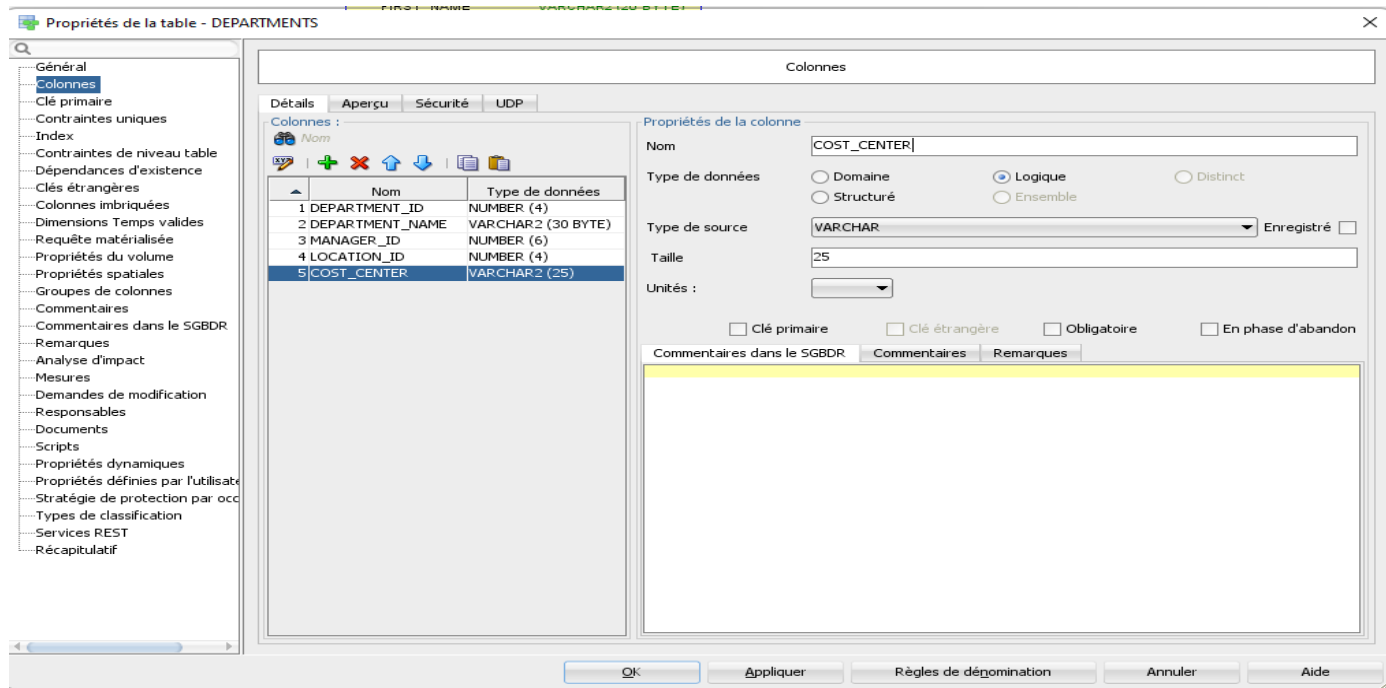
Colonne référencée	Colonne	Obligatoire
DEPARTMENT_ID	DEPARTMENTS_DEPARTMENT_ID	<input type="checkbox"/>

OK Appliquer Annuler Aide



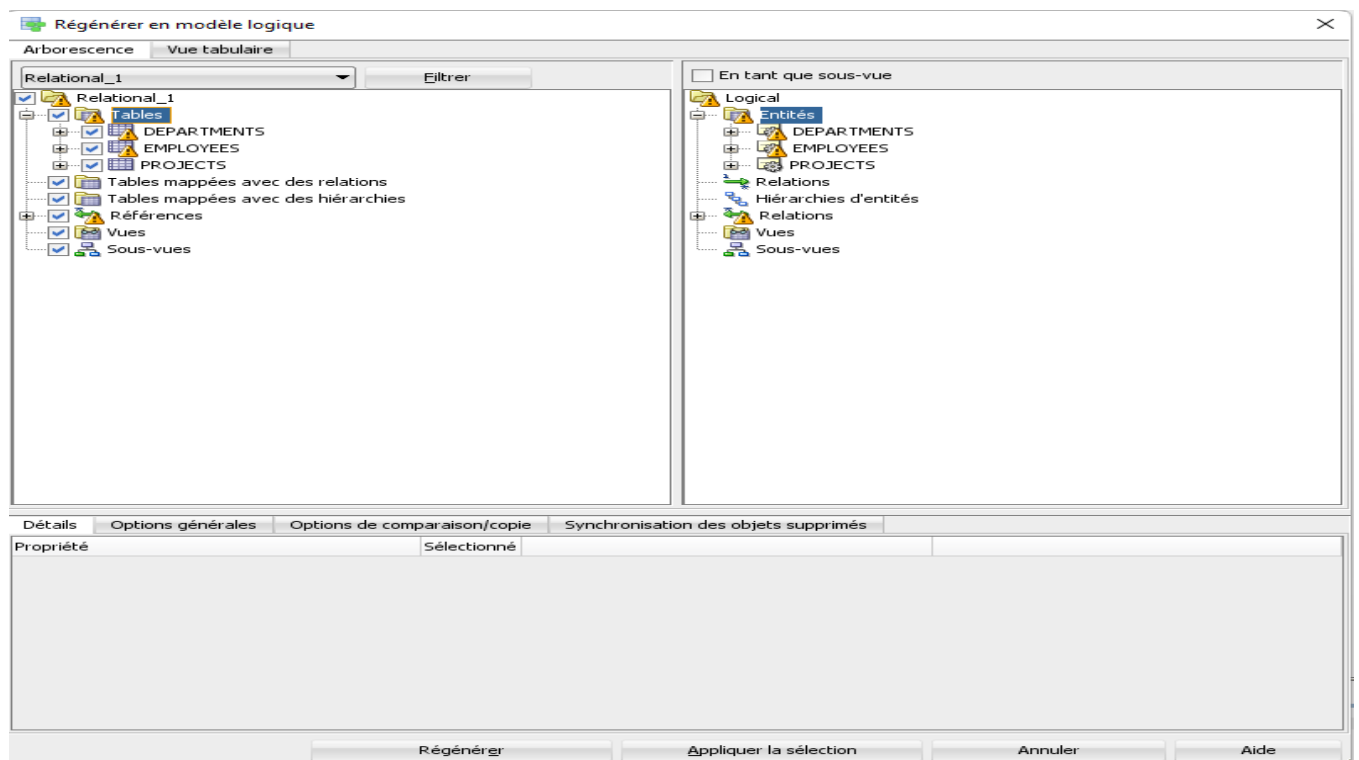
## Exercice 5 – Modification d’une Table existante

Ajouter la colonne **COST\_CENTER** à la Table **DEPARTMENTS**.  
Type **VARCHAR**, longueur 25

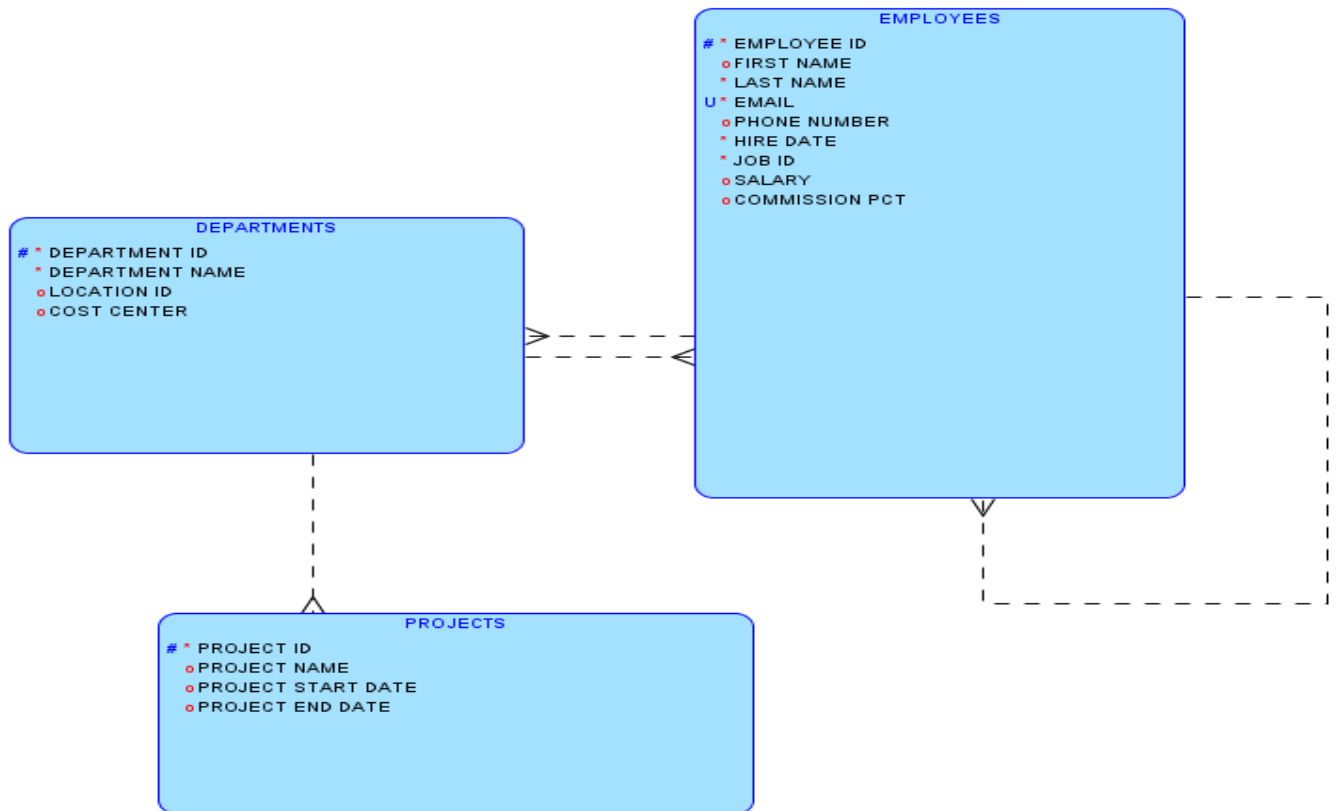


## Exercice 6 – Reverse Engineering du Modèle Relationnel vers le Modèle Conceptuel

Transformer le Modèle Relationnel pour créer un Modèle Conceptuel (MCD) ou Modèle Logique.







### Exercice 7 – Modification du Modèle Conceptuel

Enrichir le Modèle logique. Ajouter des attributs à l'entité **PROJECTS**.  
 Nom **STATUS**, Type **VARCHAR** longueur **30**.

Modifier le style de représentation « *Barker* » en notation « *Bachman*. »

Ajouter une nouvelle entité dans le modèle Logique.  
 Nom d'entité **TASKS**.

Attributs :

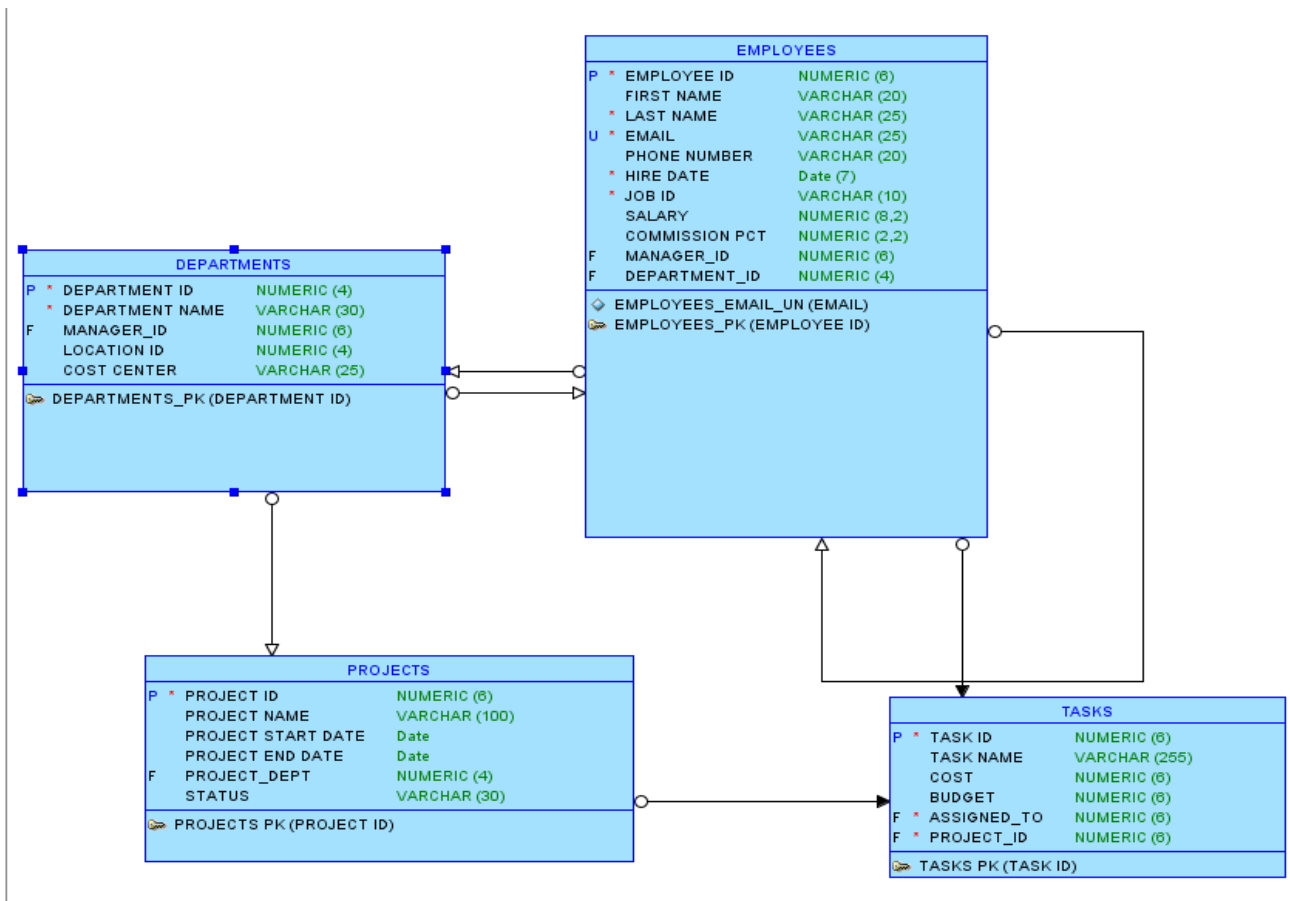
**TASK\_ID**, type **NUMERIC**, précision **6**, **Primary UID**.

**TASK\_NAME**, type **VARCHAR**, taille **255**.

**COST**, type **NUMERIC**, longueur **6**.

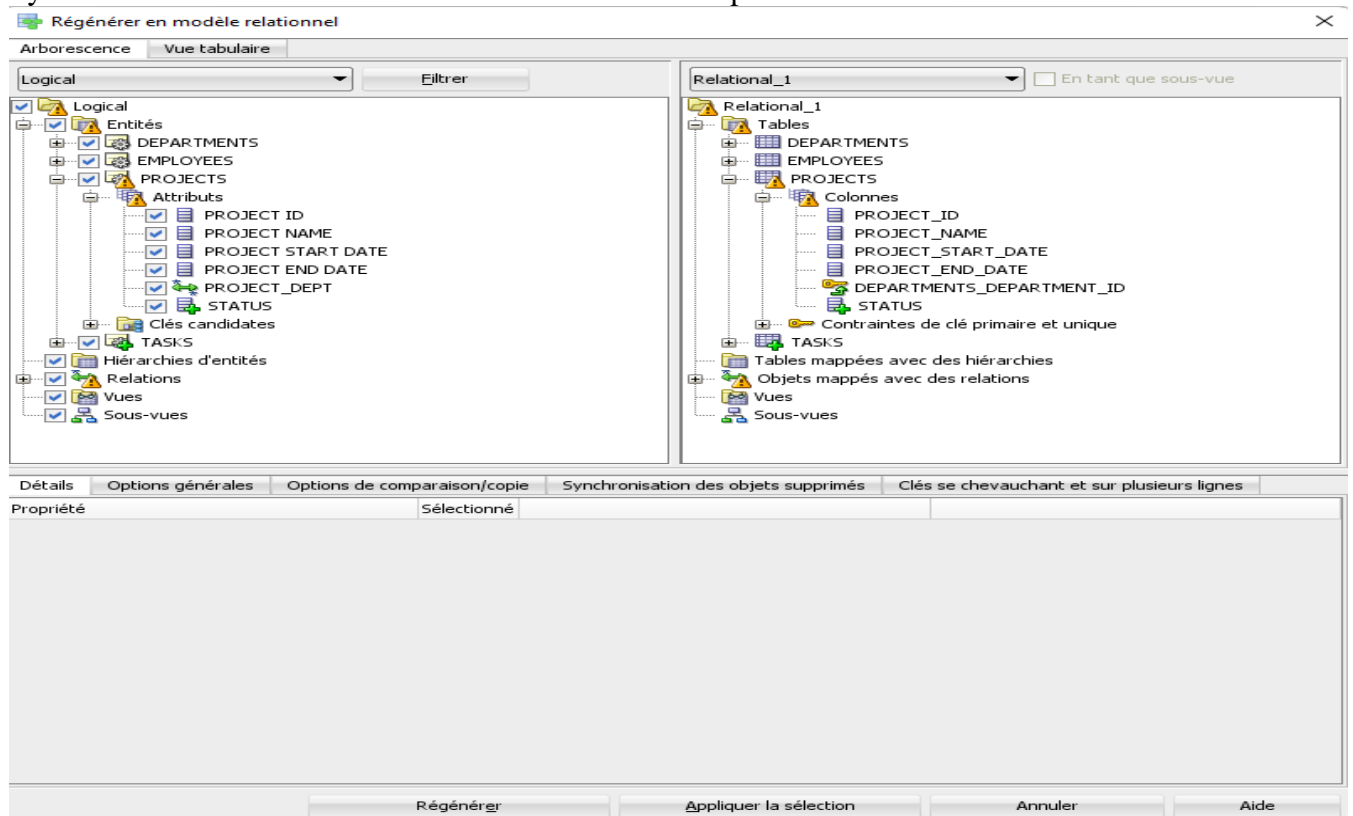
**BUDGET**, type **NUMERIC**, longueur **6**

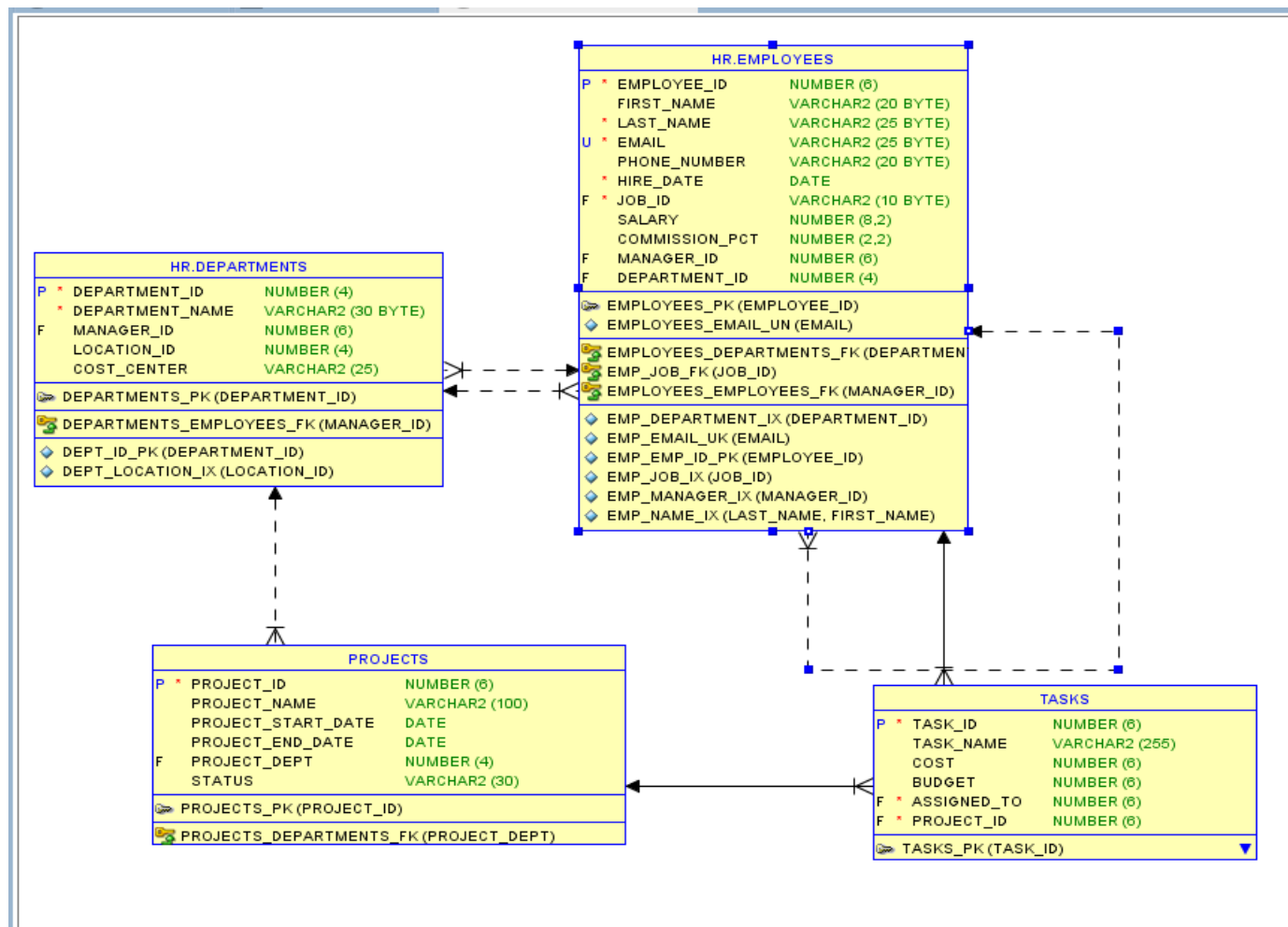
Créer deux associations entre les entités **EMPLOYEES** et **TASKS** et entre **PROJECTS** et **TASKS**.



## Exercice 8 – Synchronisation du Modèle Logique avec le Modèle Relationnel

Synchroniser le modèle relationnel avec le model Conceptuel.

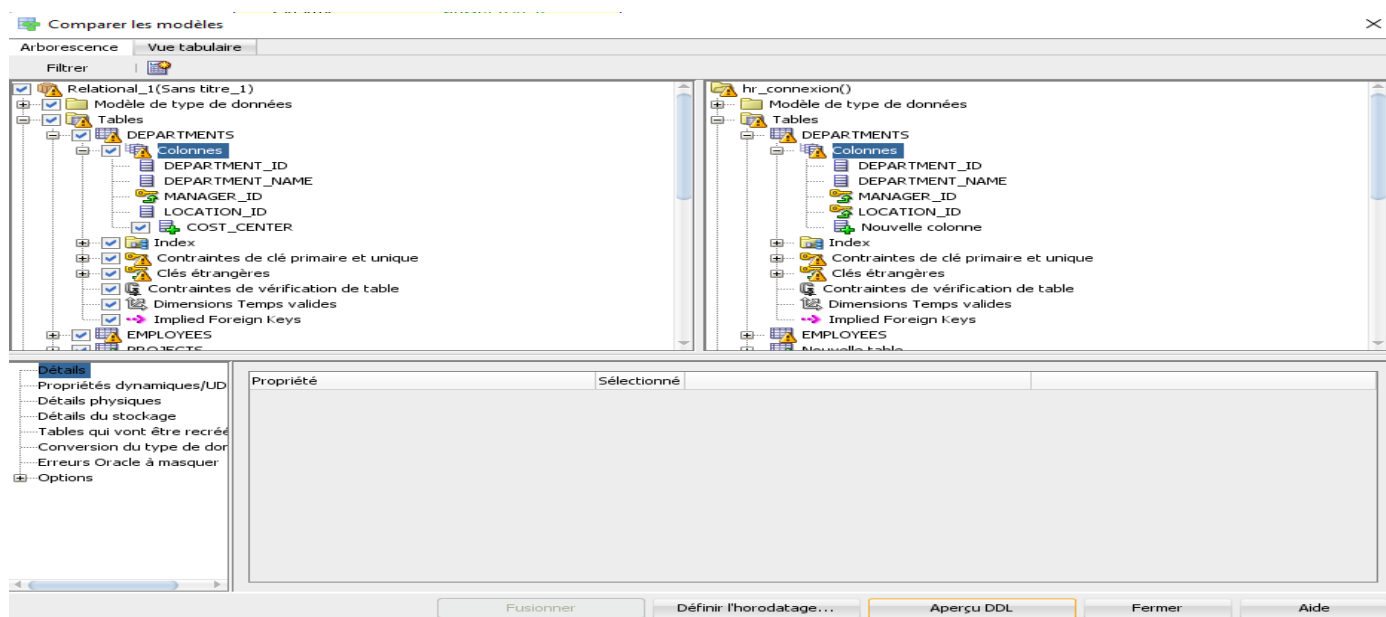




## Exercice 9 – Génération du DDL pour modifier la base de données

Synchroniser le dictionnaire de la base de données (modèle physique) avec le modèle Relationnel et générer le code DDL.

Enregistrer le script DDL. Enregistrer et sauvegarder votre modèle.



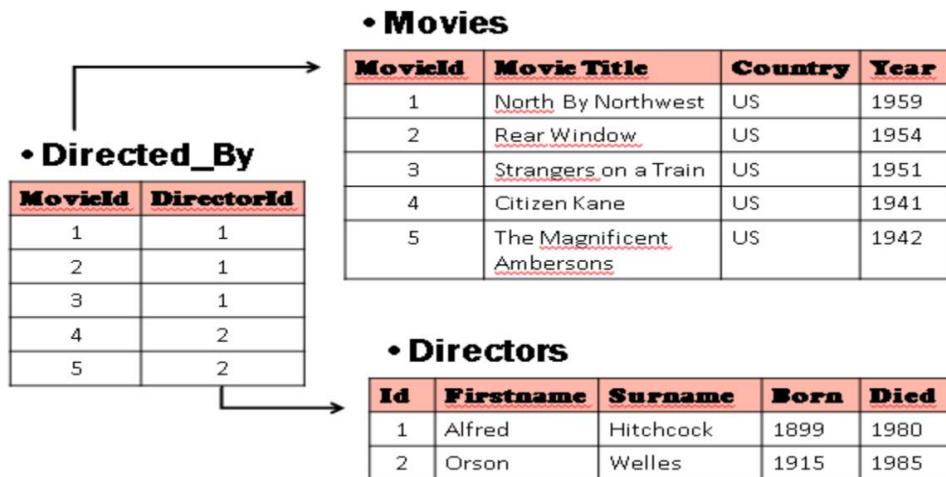
```

1  -- Généré par Oracle SQL Developer Data Modeler 21.2.0.183.1957
2  -- à :      2021-11-26 00:34:17 CET
3  -- site :   Oracle Database 21c
4  -- type :   Oracle Database 21c
5
6
7
8
9  CREATE TABLE projects (
10     project_id      NUMBER(6) NOT NULL,
11     project_name     VARCHAR2(100),
12     project_start_date DATE,
13     project_end_date  DATE,
14     project_dept     NUMBER(4),
15     status           VARCHAR2(30)
16 )
17 LOGGING;
18
19 ALTER TABLE projects ADD CONSTRAINT projects_pk PRIMARY KEY ( project_id );
20 CREATE TABLE tasks (
21     task_id          NUMBER(6) NOT NULL,
22     task_name        VARCHAR2(255),
23     cost             NUMBER(6),
24     budget           NUMBER(6),
25     assigned_to      NUMBER(6) NOT NULL,
26     project_id       NUMBER(6) NOT NULL
27 )
28 LOGGING;
29
30 ALTER TABLE tasks ADD CONSTRAINT tasks_pk PRIMARY KEY ( task_id );
31 ALTER TABLE hr.departments ADD (
32     cost_center VARCHAR2(25)
33 );

```

## Exercice 10 – Création d'un nouveau MCD, MRD et déploiement du script DDL

Créer les différents modèles MCD et MRD ainsi que le script DDL à partir de la figure suivante :



Créer un USER avec les commandes suivantes depuis une connexion DBA (user system par exemple) :

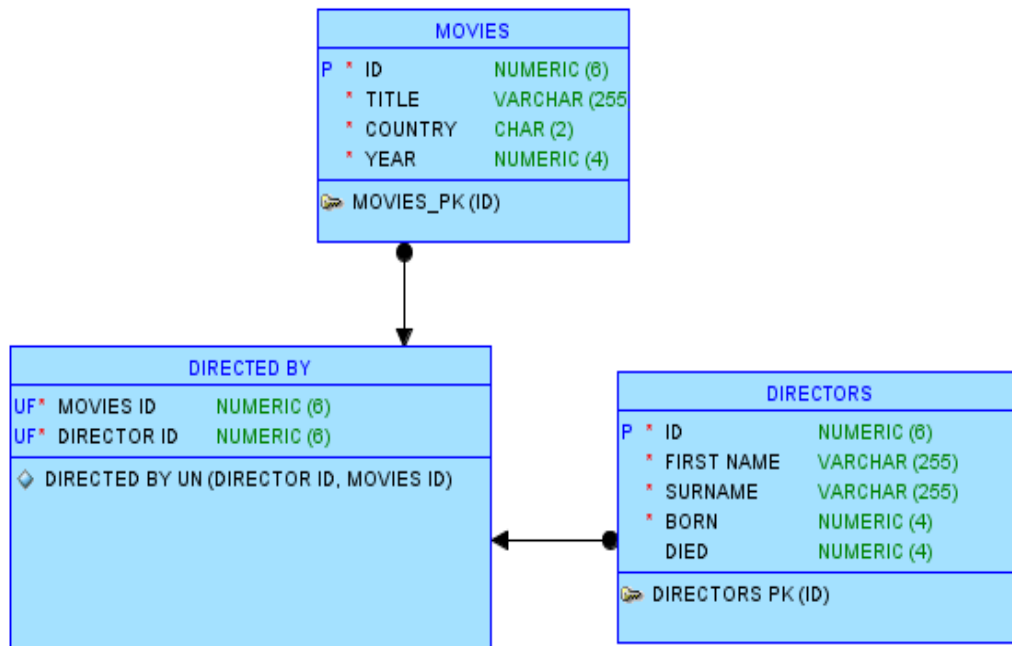
```
SQL> Create user cinema identified by secret default
tablespace users temporary tablespace temp ;
```

```
SQL> Grant connect, resource to cinema;
```

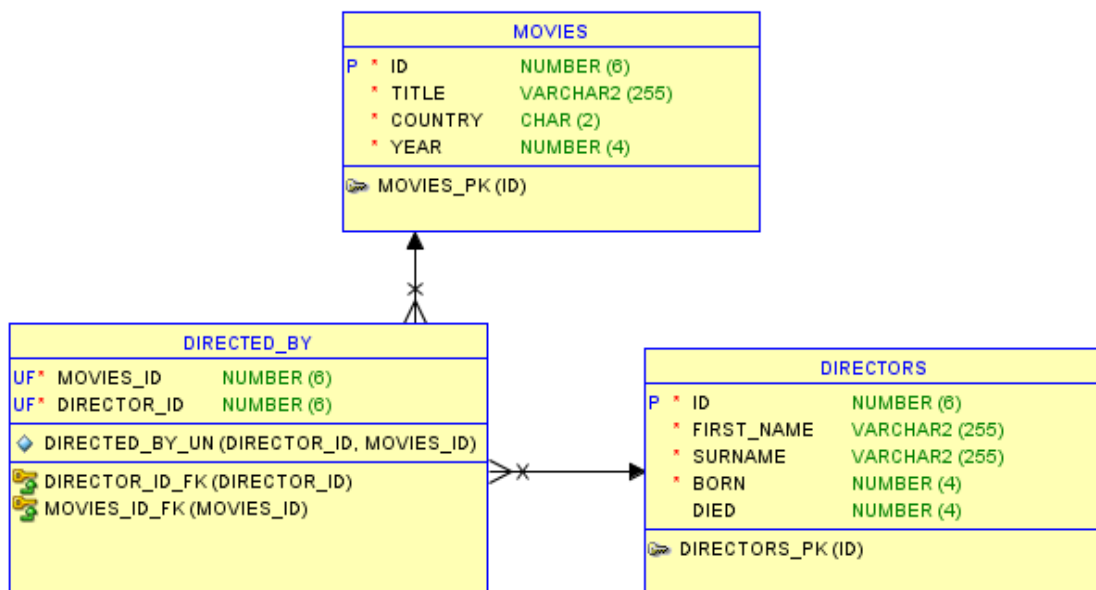
Connectez-vous sous ce nouvel utilisateur et exécuter le script DDL que vous avez créé précédemment.

Insérer les enregistrements qui sont représentés sur la figure ci-dessus.

## Modèle MCD (notation Bachman)



## MRD :



### Script DDL :

```
-- Généré par Oracle SQL Developer Data Modeler 21.2.0.183.1957
-- à :    2021-11-28 21:01:59 CET
-- site :    Oracle Database 11g
-- type :    Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE directed_by (
  movies_id  NUMBER(6) NOT NULL,
  director_id NUMBER(6) NOT NULL
);

// Contrainte unique qui nous permet d'éviter les doublons dans la table DIRECTED_BY
ALTER TABLE directed_by ADD CONSTRAINT directed_by_un UNIQUE ( director_id,
                                                                movies_id );

CREATE TABLE directors (
  id        NUMBER(6) NOT NULL,
  first_name VARCHAR2(255) NOT NULL,
  surname   VARCHAR2(255) NOT NULL,
  born      NUMBER(4) NOT NULL,
  died      NUMBER(4)
);

// Diverses contraintes de vérification qui permettent de vérifier la cohérence des données.
ALTER TABLE directors ADD CHECK ( born > 0 );

ALTER TABLE directors ADD CHECK ( died > born );

ALTER TABLE directors ADD CONSTRAINT directors_pk PRIMARY KEY ( id );

CREATE TABLE movies (
  id    NUMBER(6) NOT NULL,
  title VARCHAR2(255) NOT NULL,
  country CHAR(2) NOT NULL,
  year   NUMBER(4) NOT NULL
);

ALTER TABLE movies ADD CHECK ( year > 0 ); //idem

ALTER TABLE movies ADD CONSTRAINT movies_pk PRIMARY KEY ( id );

ALTER TABLE directed_by
  ADD CONSTRAINT director_id_fk FOREIGN KEY ( director_id )
    REFERENCES directors ( id )
```



```
ON DELETE CASCADE; // permet d'assurer la cohérence des données si un
directeur est supprimé de la table DIRECTORS.
```

```
ALTER TABLE directed_by
ADD CONSTRAINT movies_id_fk FOREIGN KEY ( movies_id )
REFERENCES movies ( id )
ON DELETE CASCADE; // idem mais pour la colonne ID de la table MOVIES.
```

```
-- Rapport récapitulatif d'Oracle SQL Developer Data Modeler :
```

-- CREATE TABLE	3		
-- CREATE INDEX	0		
-- ALTER TABLE	8		
-- CREATE VIEW	0		
-- ALTER VIEW	0		
-- CREATE PACKAGE	0		
-- CREATE PACKAGE BODY		0	
-- CREATE PROCEDURE	0		
-- CREATE FUNCTION	0		
-- CREATE TRIGGER	0		
-- ALTER TRIGGER	0		
-- CREATE COLLECTION TYPE		0	
-- CREATE STRUCTURED TYPE		0	
-- CREATE STRUCTURED TYPE BODY			0
-- CREATE CLUSTER	0		
-- CREATE CONTEXT	0		
-- CREATE DATABASE	0		
-- CREATE DIMENSION	0		
-- CREATE DIRECTORY	0		
-- CREATE DISK GROUP	0		
-- CREATE ROLE	0		
-- CREATE ROLLBACK SEGMENT			0
-- CREATE SEQUENCE	0		
-- CREATE MATERIALIZED VIEW		0	
-- CREATE MATERIALIZED VIEW LOG			0
-- CREATE SYNONYM	0		
-- CREATE TABLESPACE	0		
-- CREATE USER	0		
--			
-- DROP TABLESPACE	0		
-- DROP DATABASE	0		
--			
-- REDACTION POLICY	0		
--			
-- ORDS DROP SCHEMA	0		
-- ORDS ENABLE SCHEMA		0	
-- ORDS ENABLE OBJECT		0	
--			
-- ERRORS	0		
-- WARNINGS	0		

Créer un USER avec les commandes suivantes depuis une connexion DBA (user system par exemple) :

```
SQL> Create user cinema identified by secret default  
tablespace users temporary tablespace temp ;
```

```
SQL> Grant connect, resource to cinema;
```

Connectez-vous sous ce nouvel utilisateur et exécuter le script DDL que vous avez créé précédemment.

Insérer les enregistrements qui sont représentés sur la figure ci-dessus.

```
SQL Plus

SQL*Plus: Release 21.0.0.0.0 - Production on Dim. Nov. 28 19:50:41 2021
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Entrez le nom utilisateur : SYSTEM
Entrez le mot de passe :
Heure de la dernière connexion réussie : Dim. Nov. 28 2021 19:46:20 +01:00

Connecté à :
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CONN system/Oracle21@192.168.1.132:1521/XEPDB1
Connecté.
SQL> Create user cinema identified by secret default tablespace users temporary tablespace temp ;

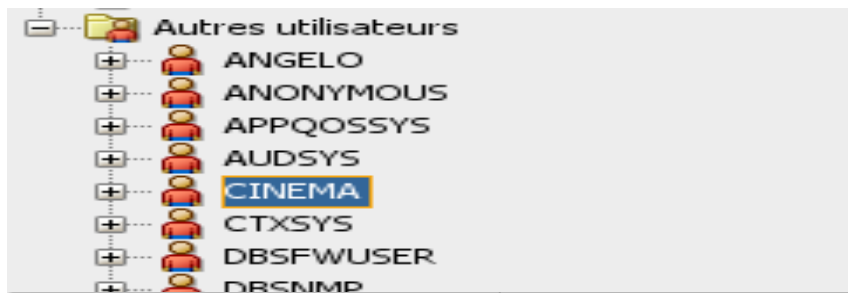
Utilisateur créé.

SQL> Grant connect, ressource to cinema;
Grant connect, ressource to cinema
*
ERREUR à la ligne 1 :
ORA-01919: le rôle 'RESSOURCE' n'existe pas

SQL> Grant connect, resource to cinema;

Autorisation de privilèges (GRANT) acceptée.

SQL>
```



Créer / Sélectionner une connexion de base de données

Nom de connexion	Détails de connexion
Angelo_Connection	ANGELO@//192.168.1.132:1521
cinema_connexion	cinema@//192.168.1.132:1521
hr_connexion	HR@//192.168.1.132:1521
Initialisation	SYSTEM@//192.168.1.132:1521
XE	SYSTEM@//192.168.1.132:1521

Name:  Color:

Type de base de données:

**Informations utilisateur** Utilisateur proxy

Type d'authentification:

Nom utilisateur:  Rôle:

Mot de passe:  ☐ Enregistrer le mot de passe

Type de connexion:

**Détails** Avancé

Nom d'hôte:

Port:

☐ SID

☒ Nom de service

Statut : succès

Aide Enregistrer Effacer Tester Connexion Annuler

Page de bienvenue x Angelo\_Connection x Relational\_1 (Model\_movies) x Model\_movies.sql x

Feuille de calcul SQL Historique

Feuille de calcul Query Builder

```
-- Généré par Oracle SQL Developer Data Modeler 21.2.0.183.1957
-- À : 2021-11-28 19:24:27 CET
-- site : Oracle Database 11g
-- type : Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
-- predefined type, no DDL - XMLTYPE

CREATE TABLE directed_by (
  movies_id NUMBER(6) NOT NULL,
  director_id NUMBER(6) NOT NULL
);

CREATE TABLE directors (
  id NUMBER(6) NOT NULL,
  first_name VARCHAR2(255) NOT NULL,
  surname VARCHAR2(255) NOT NULL,
  born NUMBER(4) NOT NULL,
  died NUMBER(4)
);

ALTER TABLE directors ADD CHECK ( born > 0 );
ALTER TABLE directors ADD CHECK ( died > born );
ALTER TABLE directors ADD CONSTRAINT directors_pk PRIMARY KEY ( id );

CREATE TABLE movies (
  id NUMBER(6) NOT NULL,
  title VARCHAR2(255) NOT NULL,
  country CHAR(2) NOT NULL,
  year NUMBER(4) NOT NULL
);
```

Sélectionner une connexion

Choisissez une connexion existante ou créez-en une pour continuer

Connexion :

Aide OK Annuler

Feuille de calcul Query Builder

```

22 first_name VARCHAR2(255) NOT NULL,
23 surname   VARCHAR2(255) NOT NULL,
24 born      NUMBER(4) NOT NULL,
25 died      NUMBER(4)
26 );
27
28 ALTER TABLE directors ADD CHECK ( born > 0 );
29
30 ALTER TABLE directors ADD CHECK ( died > born );
31
32 ALTER TABLE directors ADD CONSTRAINT directors_pk PRIMARY KEY ( id );
33
34 CREATE TABLE movies (
35 id      NUMBER(6) NOT NULL,
36 title   VARCHAR2(255) NOT NULL,
37 country CHAR(2) NOT NULL,
38 year    NUMBER(4) NOT NULL
39 );
40
41 ALTER TABLE movies ADD CHECK ( year > 0 );
42
43 ALTER TABLE movies ADD CONSTRAINT movies_pk PRIMARY KEY ( id );

```

Sortie de script x | Tâche terminée en 0,187 secondes

Table DIRECTORS modifié(e).

Table DIRECTORS modifié(e).

Table MOVIES créé(e).

Table MOVIES modifié(e).

Table MOVIES modifié(e).

Table DIRECTED\_BY modifié(e).

Table DIRECTED\_BY modifié(e).

```
SQL> ALTER USER cinema quota unlimited on users;
```

Utilisateur modifié.

```
SQL>
```

```

INSERT INTO movies(id,title,country,year) VALUES(1,'North By Northwest','US',1959);
INSERT INTO movies(id,title,country,year) VALUES(2,'Rear Window','US',1954);
INSERT INTO movies(id,title,country,year) VALUES(3,'Strangers on a Train','US',1951);
INSERT INTO movies(id,title,country,year) VALUES(4,'Citizen Kane','US',1941);
INSERT INTO movies(id,title,country,year) VALUES(5,'The Magnificent
Ambersons','US',1942);

```

```

INSERT INTO DIRECTORS(id,first_name,surname,born,died)
VALUES(1,'Alfred','Hitchcock',1899,1980);
INSERT INTO DIRECTORS(id,first_name,surname,born,died)
VALUES(2,'Orson','Welles',1915,1985);

```

```

INSERT INTO DIRECTED_BY(movies_id,director_id) VALUES(1,1);
INSERT INTO DIRECTED_BY(movies_id,director_id) VALUES(2,1);
INSERT INTO DIRECTED_BY(movies_id,director_id) VALUES(3,1);
INSERT INTO DIRECTED_BY(movies_id,director_id) VALUES(4,2);
INSERT INTO DIRECTED_BY(movies_id,director_id) VALUES(5,2);
COMMIT ;

```

Vérification :

The image displays three screenshots of a SQL query builder interface, likely from a database management system. Each screenshot shows a query window with the following SQL query:

```
select * from movies;  
select * from directed_by;  
select * from directors;
```

Below the query window, the results of the query are displayed in a table format. The results are organized into three columns: ID, TITLE, COUNTRY, and YEAR.

ID	TITLE	COUNTRY	YEAR
1	1 North By Northwest	US	1959
2	2 Rear Window	US	1954
3	3 Strangers on a Train	US	1951
4	4 Citizen Kane	US	1941
5	5 The Magnificent Ambersons	US	1942

The three screenshots show the same query and results, but with different window titles and layouts, suggesting different views or states of the application.