

CSIT 460

Développement des applications BD

Plan de module CSCI 430

Chapitre 1 : Introduction to SGBD and Application development

Chapitre 2 : Review: SGBD, MEA, MR, SQL basic statements

Chapitre 3 : Java Programming

Chapitre 4 : Java SWING API (continue)

Chapitre 5 : Database Programming: Oracle 9i PL/SQL

Chapitre 6 : Stored Procedures

Chapitre 7 : Triggers & Constraints

Chapitre 2

Review: SGBD, MEA, MR, SQL basic statements

AReview:

- 1. Data base management systems, (SGBD)**
- 2. Data base Entity Relational Database ERD (MCD)**
- 3. Relational Models RM (MLD)**
- 4. SQL basic statements (MPD-SQL)**

Data base management systems (SGBD)

BASES DE DONNEES

I.1. Définition d'une Base de Données (BD)

Une Base de Données est un **ensemble structuré de données** enregistrées sur des mémoires secondaires créé et tenu à jour pour les besoins d'un ensemble d'utilisateurs.

I.2. Le Système de gestion de bases de Données (SGBD)

C'est **l'outil** permettant de créer des bases de données, mettre à jour les données de la base (ajout, modification, suppression), de rechercher des données, de fournir différentes manières de les visualiser et de les imprimer.

I.3. Le modèle relationnel

Le modèle relationnel est basé sur une organisation des données d'une base de données sous forme de tableaux appelés **tables**.

Data base management systems (SGBD)

BASES DE DONNEES

I.1. Définition d'une Base de Données (BD)

Une Base de Données est un **ensemble structuré de données** enregistrées sur des mémoires secondaires créé et tenu à jour pour les besoins d'un ensemble d'utilisateurs.

I.2. Le Système de gestion de bases de Données (SGBD)

C'est **l'outil** permettant de créer des bases de données, mettre à jour les données de la base (ajout, modification, suppression), de rechercher des données, de fournir différentes manières de les visualiser et de les imprimer. (Oracle, MySQL, Access)

I.3. Le modèle relationnel

Le modèle relationnel est basé sur une organisation des données d'une base de données sous forme de tableaux appelés **tables**.

Data base Entity Relational Database ERD

Modèle Conceptuel des Données

Représentation **statique**, sous forme schématique, de la situation respective des données d'un domaine de gestion.

- ✓ Ce schéma est conçu pour être très stable dans le temps.
- ✓ Son objectif : définir (**identifier**) toutes les données utilisées, les regrouper en ensembles appelés **entités**, et de lier ces entités par des **relations (association)**, dans un modèle défini et compréhensible par toute personne connaissant la "syntaxe" du MCD.
- ✓ Le MCD regroupe les informations à traiter, le "**quoi**" du système.
- ✓ Les étapes du MCD :

1, 2 et 3
Pour
valider
l'entité

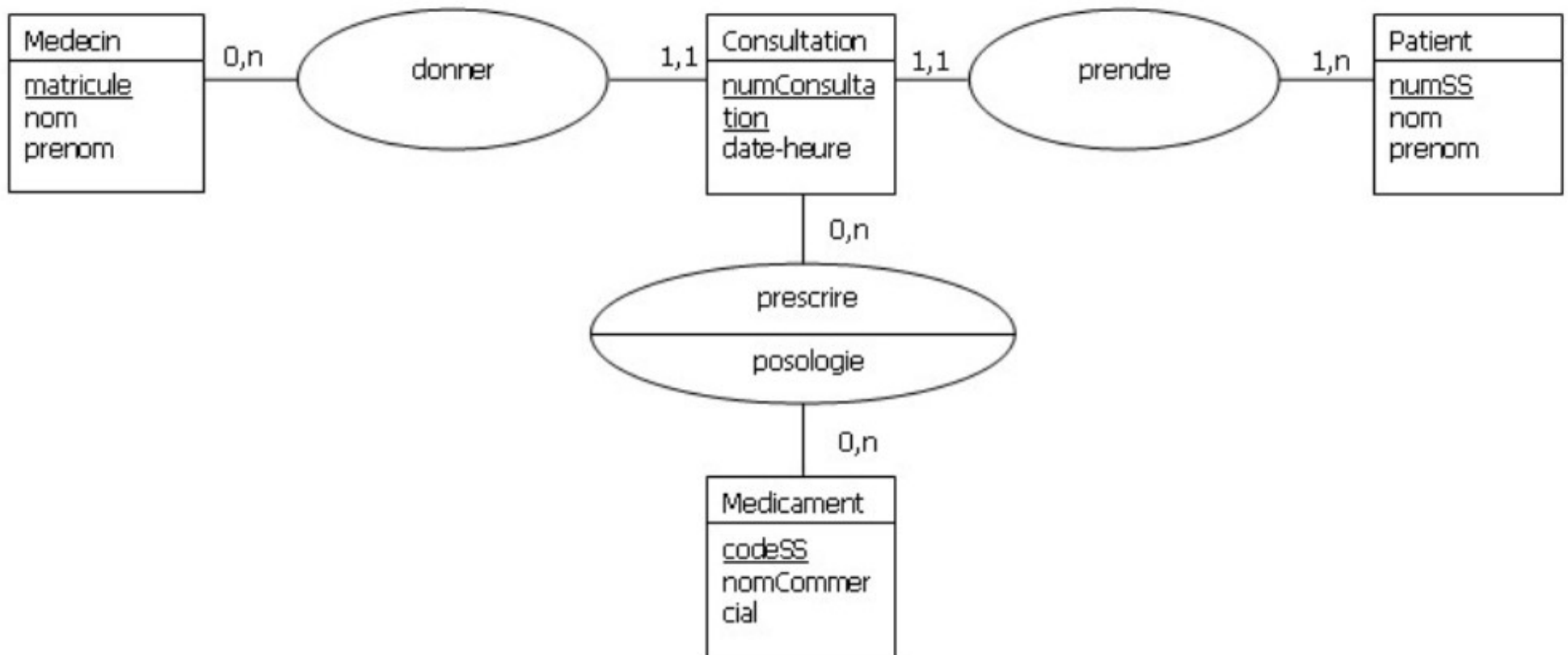
1. Détermination des entités(Nom de l'entité): Personne, Voiture, Rectangle....
2. L'identifiant (figure en premier dans la liste des propriétés et souligné) qui permet d'identifier de façon unique une occurrence de l'entité: matricule de l'étudiant, IDprof, ID...
3. Catalogue des données (propriétés de l'entités): nom, prénom, âge, longueur, largeur, vitesse ..

4 et 5 Pour
valider
l'association

4. Recensement des associations (avec, éventuellement, les propriétés non encore affectées)
5. Détermination des cardinalités

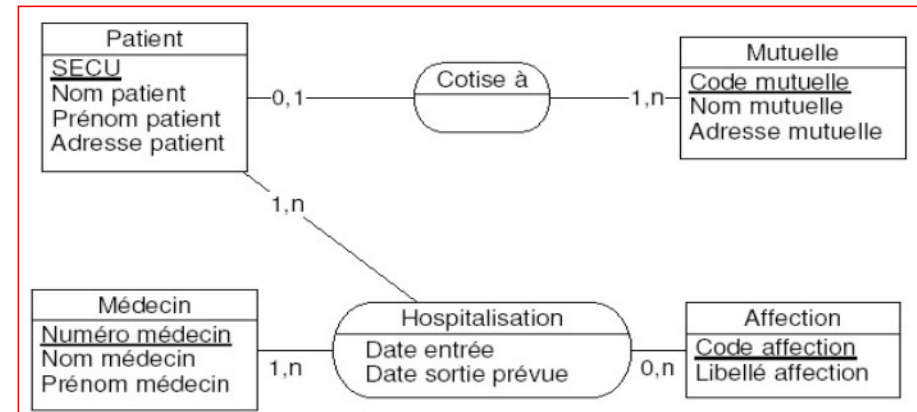
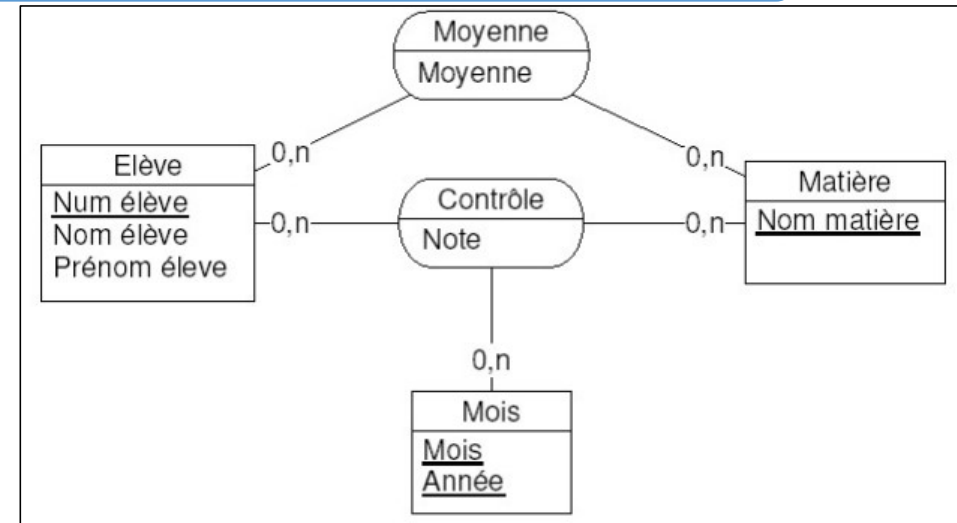
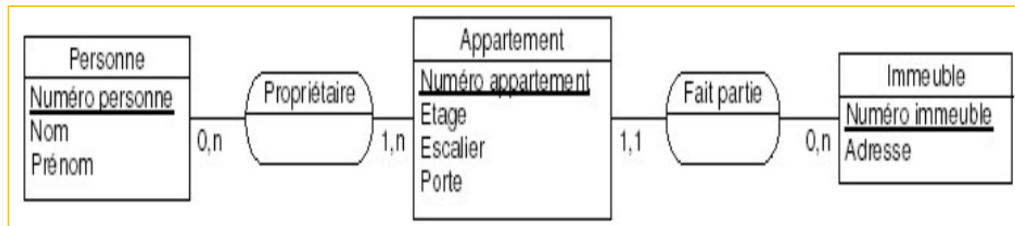
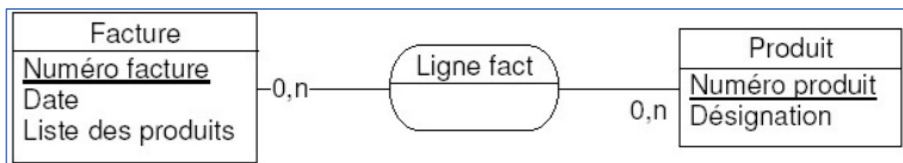
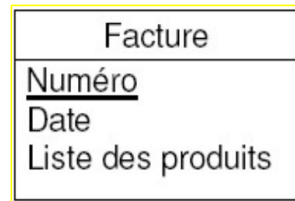
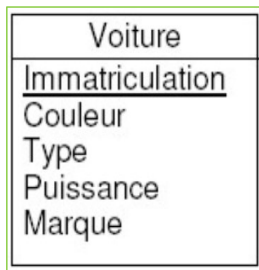
Data base Entity Relational Database ERD

Modèle Conceptuel des Données: Exemple complet et correct



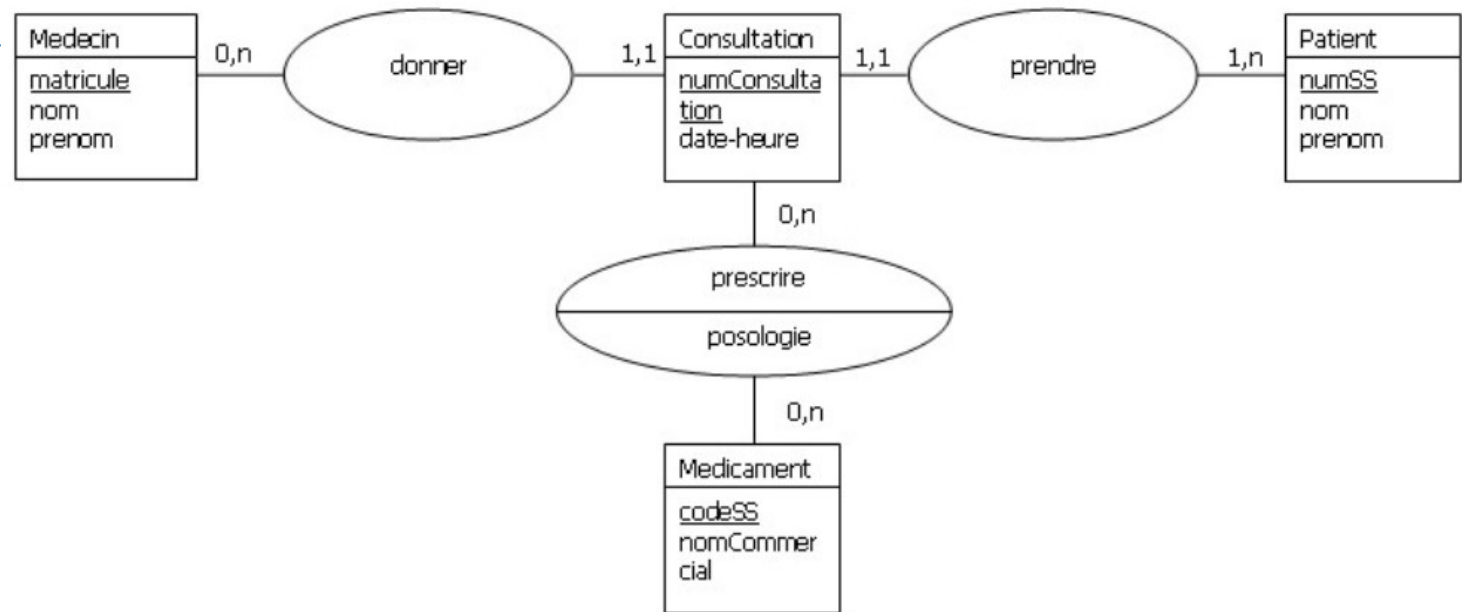
Exercice 1 : Entité valide

Est-ce que les MCD suivants sont valides ou non? Justifier ta réponse et corriger les erreurs de modélisation.



Exercice 2 : Clinique Médical

On vous donne le MCD représentant des visites dans une clinique médical.

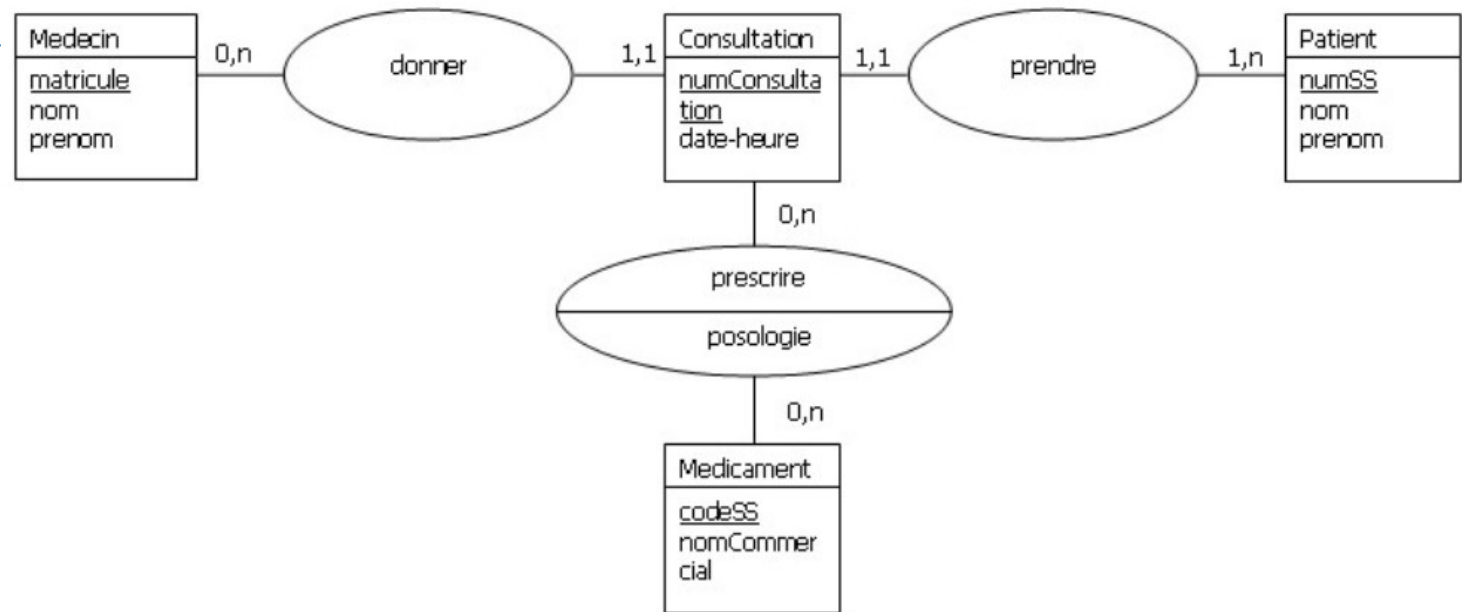


T.A.F:

1. Un patient peut-il effectuer plusieurs consultations ?
2. Un médecin peut-il recevoir plusieurs patients dans la même consultation ?
3. Peut-on prescrire plusieurs médicaments dans une même consultation ?
4. Deux médecins différents peuvent-ils prescrire le même médicament ?

Exercice 1 : Clinique Médical

On vous donne le MCD représentant des visites dans une clinique médical.



T.A.F:

1. Un patient peut-il effectuer plusieurs consultations ?
2. Un médecin peut-il recevoir plusieurs patients dans la même consultation ?
3. Peut-on prescrire plusieurs médicaments dans une même consultation ?
4. Deux médecins différents peuvent-ils prescrire le même médicament ?

Relational Models RM

Modèle Logique de Données: Modèle Relationnel

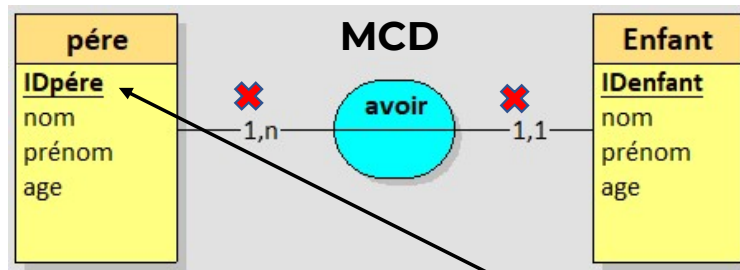
Dans un **modèle logique des données**, ***il n'y a pas des associations*** et il faut, ***au moins, une seule clé étrangère*** entre deux entités reliées par une association dans le MCD

Les règles de passage du MCD en MLD:

1. La règle : 1 – N (père – enfant)
2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
3. La règle : N – N: (Client – produit)

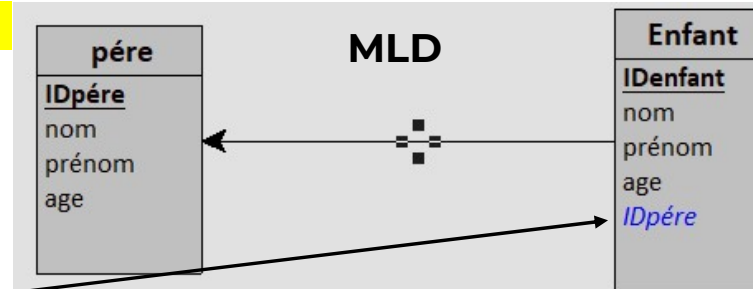
La règle : 1 – N (père – enfant):

éliminer les cardinalités minimums du schéma et l'ID de l'entité de cardinalité max=n(entité père) va migré vers l'entité de cardinalité min=1(entité enfant) et devenir une clés étrangères



Clé primaire

Clé étrangère



Mode d'écriture de MLD

père = (IDpère, nom, prénom, age);

Enfant = (IDenfant, nom, prénom, age, #IDpère);

Relational Models RM

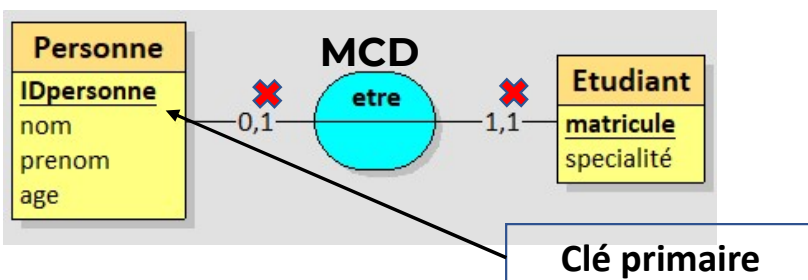
Modèle Logique de Données: Modèle Relationnel

Dans un **modèle logique des données**, ***il n'y a pas des associations*** et il faut, ***au moins, une seule clé étrangère*** entre deux entités reliées par une association dans le MCD

Les règles de passage du MCD en MLD:

1. La règle : 1 – N (père – enfant)
2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
3. La règle : N – N: (Client – produit)

2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
INVERSER LA REGLE (père-enfant): éliminer les cardinalités maximum et l'ID de l'entité de cardinalité min=0 va migré vers l'entité de cardinalité minx=1, et devenir une clés étrangères.



Mode d'écriture de MLD

Personne = (IDpersonne , nom, prenom , age);
Etudiant = (matricule , specialité , #IDpersonne);

Relational Models RM

Modèle Logique de Données: Modèle Relationnel

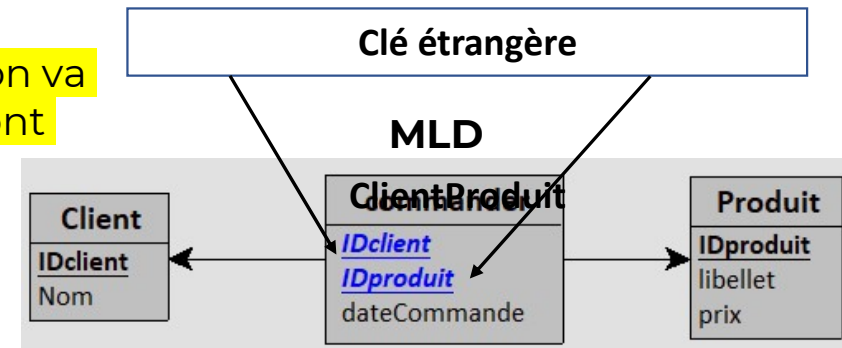
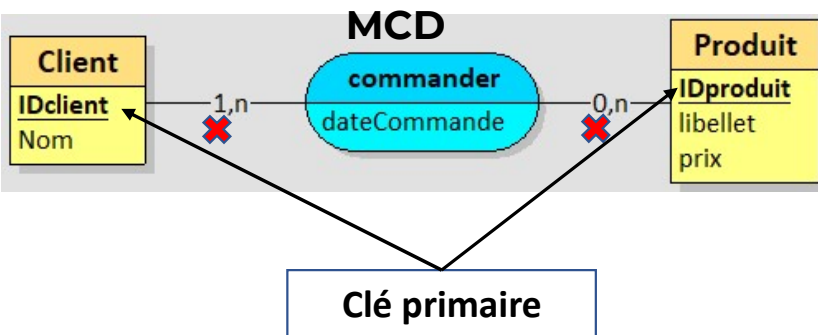
Dans un **modèle logique des données**, ***il n'y a pas des associations*** et il faut, ***au moins, une seule clé étrangère*** entre deux entités reliées par une association dans le MCD

Les règles de passage du MCD en MLD:

1. La règle : 1 – N (père – enfant)
2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
3. La règle : N – N: (Client – produit)

3. La règle : N – N: (Client – produit)

Eliminer les cardinalités minimums du schéma et l'association va devenir une entité, les clés prières des entités du schémas vont devenir des clés étrangères dans la nouvelle entités.



Mode d'écriture de MLD

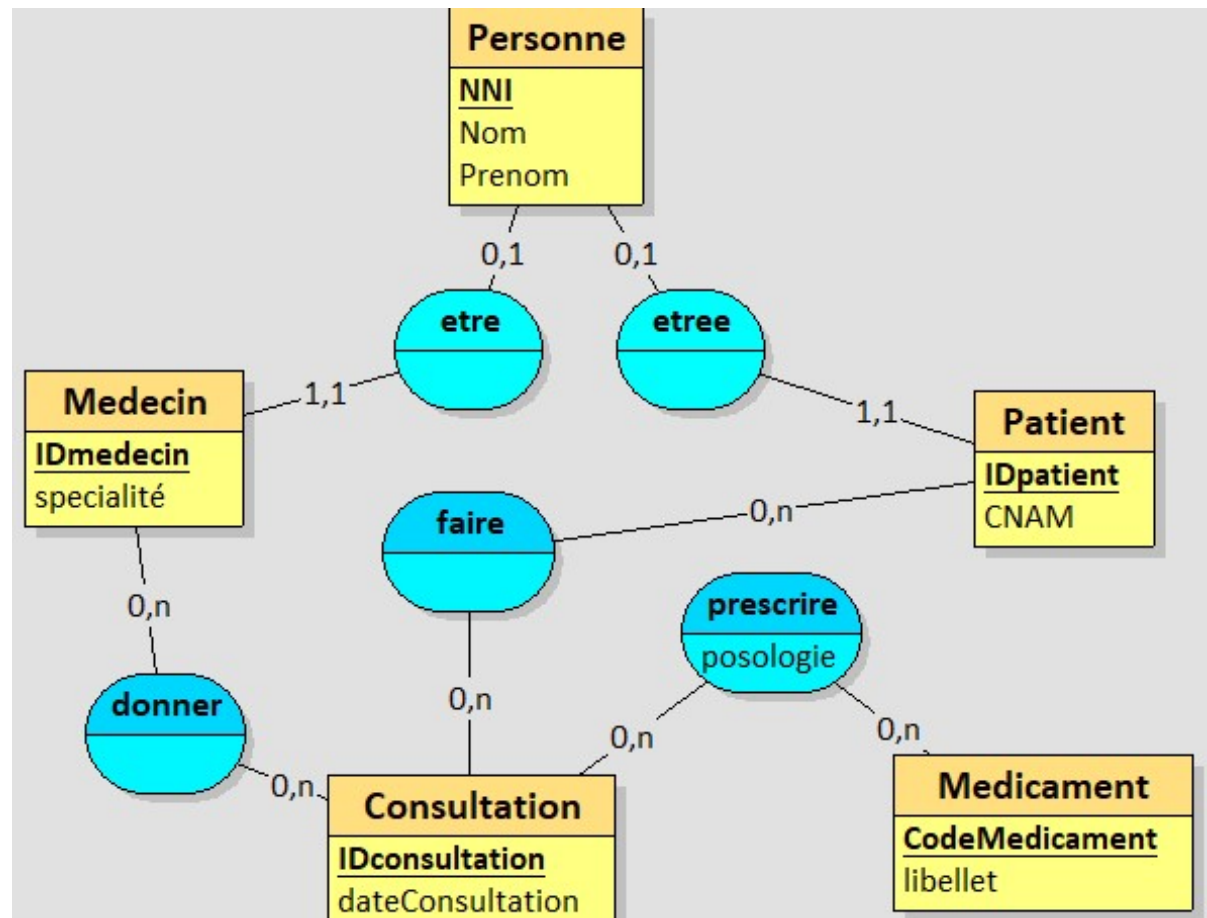
Client = (IDclient, Nom);

Produit = (IDproduit, libellet , prix);

ClientProduit = (#IDclient, #IDproduit, dateCommande);

Relational Models RM: Exercice

Soit le Modèle Conceptuel des Données suivant :



T.A.F:

1. En déduire le MLD

Relational Models RM: Exercice

Exercice 2 : Entreprise et employés

Dans une entreprise, un **département** est identifié par un code et caractérisé par un nom et une localisation.

Un **employé** est identifié par son matricule et caractérisé par son nom, son grade et le département dans lequel il est affecté.

Le numéro d'un employé est unique dans son département mais pas dans l'entreprise dans lequel il travaille.

Une **entreprise** est identifiée par son code et caractérisé par son nom, son statut et son siège.

T.A.F :

1. réaliser le MCD.
2. En déduire le MLD

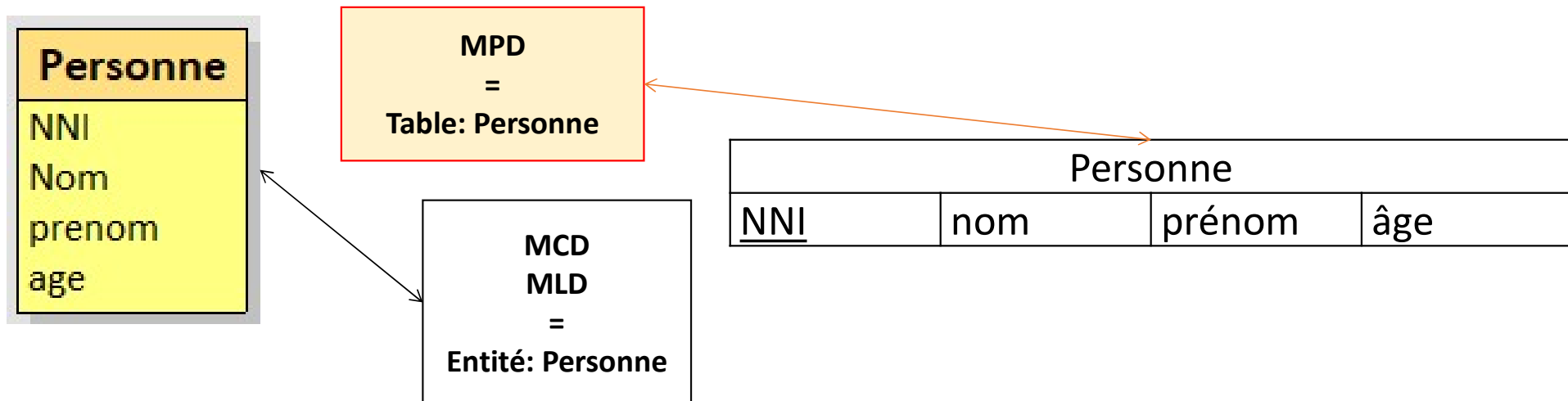
MCD et MLD

Ces deux modèles des données sont **indépendants du SGBD**

Le MPD: le Modèle Physique de Données

Le Modèle Physique de Données **dépendant du SGBD**

MPD: MLD + le type de données de chaque attributs



MPD: Modèle Physique de Données

Transformation en tables les entités du MLD

- Dépend de la base de données cible (Oracle, MySQL, SQLite, PostgreSQL...)
- **Types** de données
- Les attributs qui permettent d'indexer les tables sont des **clés primaires**
- Les attributs (non clés primaires) qui font référence aux clés primaires d'autres tables sont des **clés étrangères**

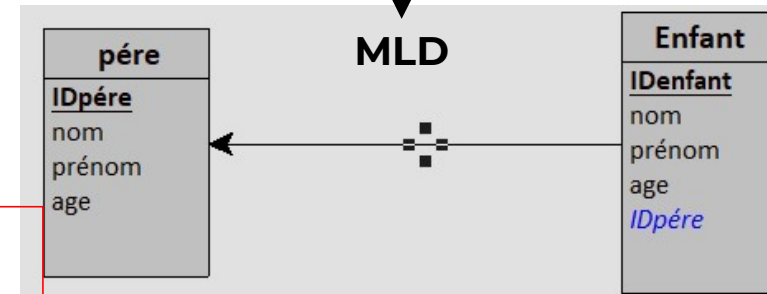
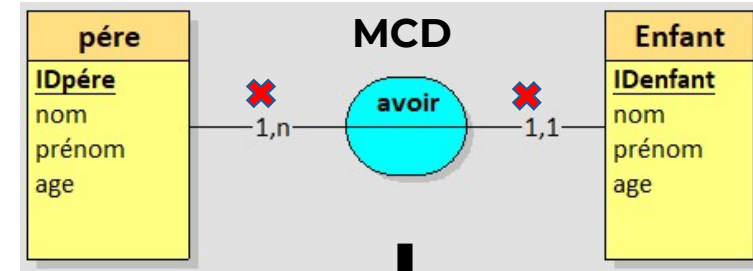
Modèle Physique de Données: SQL

Les règles de passage du MCD en MLD:

1. La règle : 1 – N (père – enfant)
2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
3. La règle : N – N: (Client – produit)

La règle : 1 – N (père – enfant):

éliminer les cardinalités minimums du schéma et l'ID de l'entité de cardinalité max=n(entité père) va migré vers l'entité de cardinalité min=1(entité enfant) et devenir une clés étrangères



MPD SQL

```
CREATE TABLE père(
  IDpère INT,
  nom VARCHAR(50),
  prénom VARCHAR(50),
  age INT,
  PRIMARY KEY(IDpère)
);
```

```
CREATE TABLE Enfant(
  IDenfant INT,
  nom VARCHAR(50),
  prénom VARCHAR(50),
  age INT,
  IDpère INT NOT NULL,
  PRIMARY KEY(IDenfant),
  FOREIGN KEY(IDpère) REFERENCES père(IDpère));
```

Autre mode d'écriture de MPD:

père = (IDpère INT, nom VARCHAR(50), prénom VARCHAR(50), age INT);
 Enfant = (IDenfant INT, nom VARCHAR(50), prénom VARCHAR(50), age INT, #IDpère);

Modèle Physique de Données: SQL

Les règles de passage du MCD en MLD:

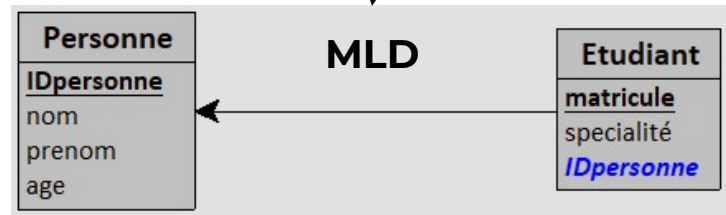
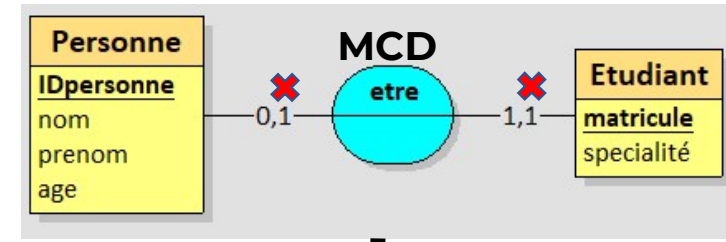
1. La règle : 1 – N (père – enfant)
2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
3. La règle : N – N: (Client – produit)

2. **L'inverse de la règle (1 – N): 1 – 0** (Etudiant - Personne)
 INVERSER LA REGLE (père-enfant): éliminer les cardinalités maximum et l'ID de l'entité de cardinalité min=0 va migré vers l'entité de cardinalité minx=1, et devenir une clés étrangères.

MPD SQL

```
CREATE TABLE Personne(
  IDpersonne INT,
  nom VARCHAR(50),
  prenom VARCHAR(50),
  age INT,
  PRIMARY
  KEY(IDpersonne)
);
```

```
CREATE TABLE Etudiant(
  matricule VARCHAR(50),
  specialité VARCHAR(50),
  IDpersonne INT NOT NULL,
  PRIMARY KEY(matricule),
  UNIQUE(IDpersonne),
  FOREIGN KEY(IDpersonne) REFERENCES
  Personne(IDpersonne));
```



Autre mode d'écriture de MPD:

Personne = (IDpersonne INT, nom VARCHAR(50), prenom VARCHAR(50), age INT);
Etudiant = (matricule VARCHAR(50), specialité VARCHAR(50), #IDpersonne);

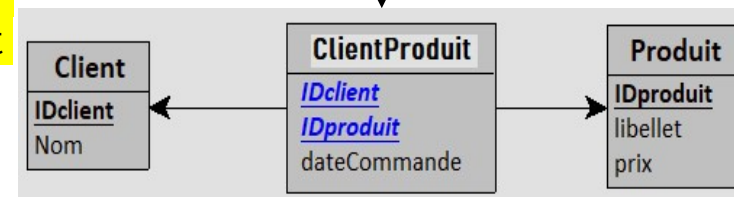
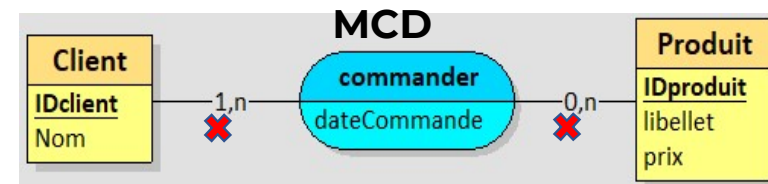
Modèle Physique de Données: SQL

Les règles de passage du MCD en MLD:

1. La règle : 1 – N (père – enfant)
2. L'inverse de la règle (1 – N): 1 – 0 (Etudiant - Personne)
3. La règle : N – N: (Client – produit)

3. La règle : N – N: (Client – produit)

Eliminer les cardinalités minimums du schéma et l'association va devenir une entité, les clés prières des entités du schémas vont devenir des clés étrangères dans la nouvelle entités.



MPD SQL

```
CREATE TABLE Client(
  IDclient INT,
  Nom VARCHAR(50),
  PRIMARY KEY(IDclient)
);
CREATE TABLE Produit(
  IDproduit INT,
  libellet VARCHAR(50),
  prix DECIMAL(15,2),
  PRIMARY KEY(IDproduit)
);
```

```
CREATE TABLE ClientProduit(
  IDclient INT,
  IDproduit INT,
  dateCommande DATE,
  PRIMARY KEY(IDclient, IDproduit),
  FOREIGN KEY(IDclient) REFERENCES
  Client(IDclient),
  FOREIGN KEY(IDproduit) REFERENCES
  Produit(IDproduit));
```

Autre mode d'écriture de MPD:

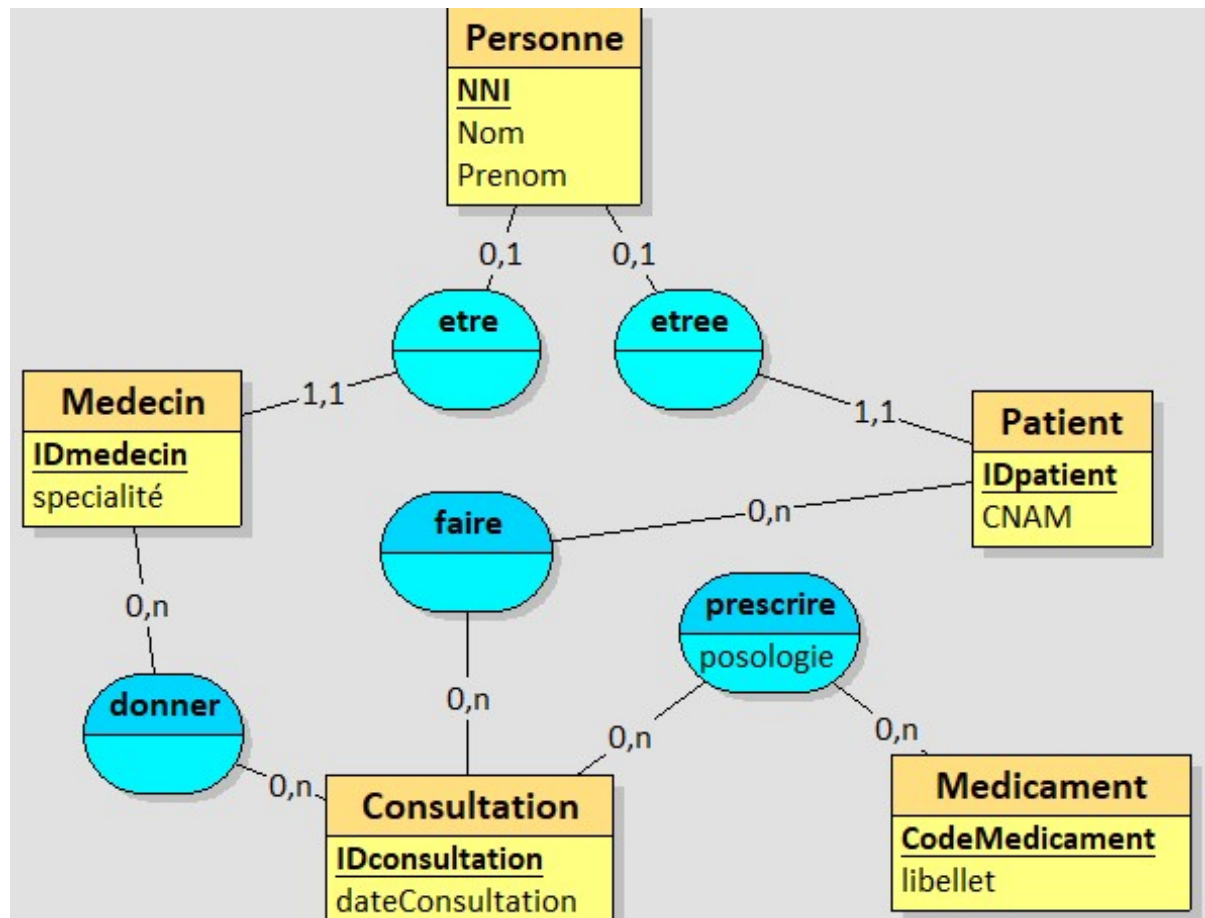
Client = (IDclient INT, Nom VARCHAR(50));
Produit = (IDproduit INT, libellet VARCHAR(50), prix DECIMAL(15,2));
ClientProduit = (#IDclient, #IDproduit, dateCommande DATE);

MPD (SQL) : Exercice

Soit le Modèle Conceptuel des Données suivant :

T.A.F:

1. En déduire le MLD
2. Ecrire les scripts SQL du MPD



Relational Models RM: Exercice

Exercice 1 : Bibliothèque

On considère une bibliothèque contenant des livres pouvant être empruntés.

Un **livre** est caractérisé par un numéro unique, un titre, un **auteur** et un **éditeur**. En outre, on décrit un livre par un certain nombre de **mots-clés** qui indiquent les sujets qui y sont traités.

La bibliothèque dispose d'un ou plusieurs exemplaires de chaque livre, **L'exemplaire** est identifié par un numéro et caractérisé par sa position dans les rayonnages et sa date d'achat.

Un exemplaire peut être emprunté par un **emprunteur**. Ces derniers sont identifiés par un numéro d'emprunteur et possèdent un nom et une adresse

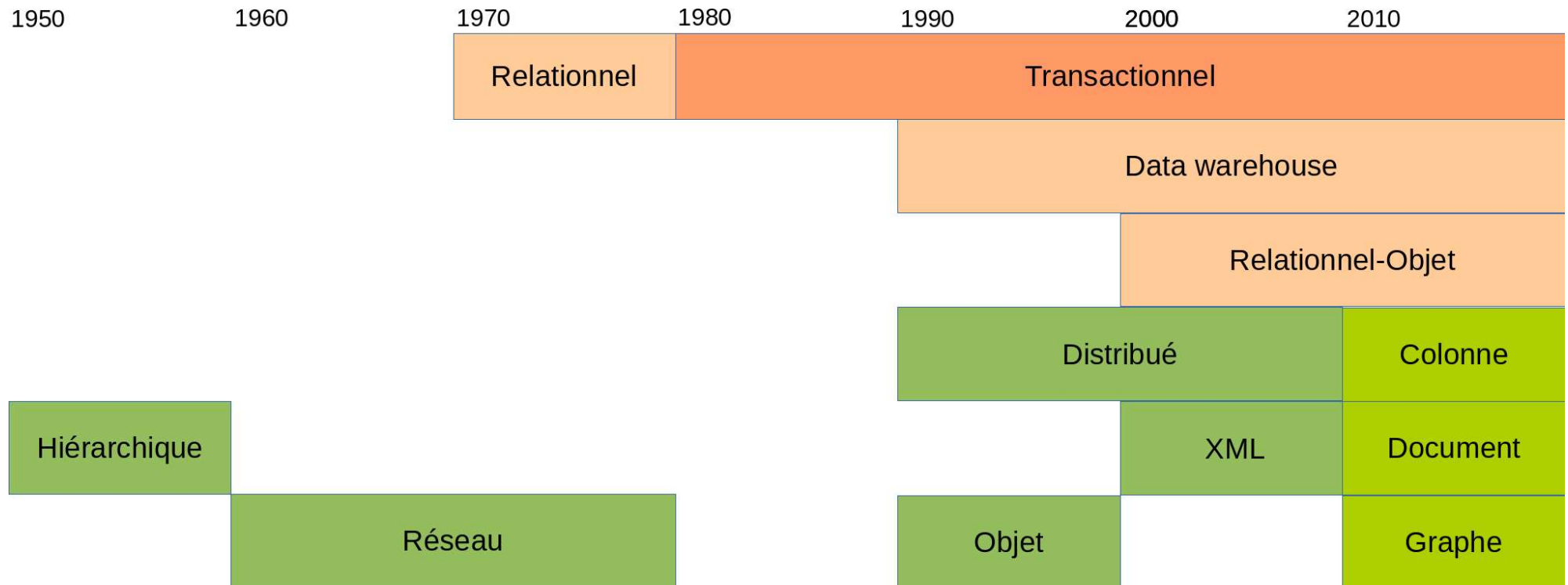
T.A.F :

1. Donner le MCD.
2. En déduire le MLD
3. Ecrire les scripts SQL du MPD

Définition du modèle relationnel

1. Le modèle relationnel n'a pas été le seul et unique modèle des bases de données : différents autres modèles et technologies ont été explorés au cours des années. À la naissance de l'informatique, plusieurs modèles de stockage de l'information sont explorés, comme les modèles hiérarchiques ou réseau.
2. Historique des technologies des bases de données
3. C'est néanmoins le modèle relationnel qui l'emporte dans les années **1970** pour sa simplicité et sa rigueur. Ce modèle présente ce qu'il y a de mieux pour assurer le contrôle de l'intégrité des données.
4. Depuis l'essor de l'informatisation dans les années **1990** puis celui du Web dans les années 2000, la question de la gestion des données se pose différemment. D'une part la question de la **cohérence** n'est plus de premier plan dans tous les contextes de gestion des données, par exemple lors de l'étude statistique de données ou lors de la gestion de contenus non critiques. D'autre part de plus en plus d'applications requièrent la gestion de très grandes quantités d'informations, fortement liées entre elles, pour lesquelles la représentation relationnelle n'est plus adaptée. Cette inadéquation pose des problèmes de performance, par exemple pour la gestion des réseaux sociaux.

1. Méthodologies de modèles de données



Le modèle relationnel: Mise en situation

Les bases de données dites **relationnelles** sont les plus connues et les plus utilisées. Mais au-delà du simple fait de stocker des données, ces bases de données répondent à des contraintes fortes, dénotées par le mot **relationnel**. Réfléchissez à ce qu'il se passe lorsque vous insérez des données en base : elles doivent respecter une structure prédéfinie, celle des tables, ainsi que des contraintes, comme l'unicité de la clé primaire. Aussi, un attribut ne doit pas contenir plusieurs informations : on sépare assez naturellement le nom, le prénom et la date de naissance dans une table. Tous ces réflexes sont une conséquence des fondements théoriques du modèle relationnel.



Fonction première de toute base de données : stocker et retrouver

Fondamental

La première fonction d'une base de données est de permettre de **stocker** et **retrouver** l'information.

Fonction secondaire d'une base de données relationnelle : la cohérence des données

Fondamental

Une base de données relationnelle assure la **cohérence** des informations relativement à un **schéma** exprimé a priori et ce quelles que soient ses conditions d'utilisation.

Une base de données n'est pas seulement un outil pour stocker et retrouver de l'information, c'est aussi un outil permettant d'exprimer un ensemble de règles relatives à ces données (type, format, optionnalité, unicité, références, etc.) et assurant le respect de ces règles **en premier lieu**.

Le modèle relationnel

Corollaire

Remarque


- Une base de données relationnelle bien conçue est toujours cohérente par rapport à son schéma.
C'est donc un outil idéal pour réaliser des applications pour lesquelles l'intégrité des données est non négociable, par exemple pour une application de transaction bancaire.
- Rien ne peut outrepasser le respect des règles de cohérence. La préservation de l'intégrité sera toujours
- préférée à la rapidité d'exécution de la fonction de stockage ou de recherche.
C'est donc un outil qui peut s'avérer contre-productif pour des applications pour lesquelles la cohérence n'est pas indispensable mais où les performances le sont, par exemple pour un outil de recherche documentaire.



Relationnel et contrôle de l'intégrité des données

Rappel

Le relationnel propose un modèle théorique puissant et simple pour gérer le contrôle et l'intégrité des données. On notera en particulier les notions suivantes :

- 
- ☐ **Identification par les données**
 - ☐ **L'atomicité**
 - ☐ **Le schéma**
 - ☐ **La non-redondance**

1. L'identification par les données

Définition

Toute relation contient au moins une clé (un ensemble de données unique et non nul).

La clé permet d'identifier un enregistrement. Il n'existe aucune autre technique pour identifier une donnée.

Cette règle de conception est issue de la première forme normale (1NF).

Elle assure l'indépendance entre les données et le système physique de stockage des données.


Si l'on porte les données d'un système à un autre, aucune information n'est perdue.



Relationnel et contrôle de l'intégrité des données

Rappel

Le relationnel propose un modèle théorique puissant et simple pour gérer le contrôle et l'intégrité des données. On notera en particulier les notions suivantes :

- 
- ☐ Identification par les données
 - ☐ L'atomicité
 - ☐ Le schéma
 - ☐ La non-redondance

1. L'identification par les données
2. L'atomicité

Définition

Les informations doivent être découpées afin que chaque case d'une relation ne contiennent qu'une seule donnée à la fois.

Il n'existe qu'une structure de données, la relation. Une relation est un tableau dont toutes les lignes et colonnes sont identiques en termes de structure.

Cette règle est issue de la première forme normale (1NF).

Elle assure que les données sont toujours accessibles à travers le formalisme de l'algèbre relationnelle et donc qu'une requête SQL simple permet toujours de trouver le niveau d'information requis. Par exemple si on sépare bien le nom et le prénom dans deux colonnes séparées, on n'a pas besoin de fonction de traitement pour extraire l'un ou l'autre et on ne risque pas de se tromper (inversion nom et prénom, prénom composé, etc.).



Relationnel et contrôle de l'intégrité des données

Rappel

Le relationnel propose un modèle théorique puissant et simple pour gérer le contrôle et l'intégrité des données. On notera en particulier les notions suivantes :

- 
- ☐ **Identification par les données**
 - ☐ **L'atomicité**
 - ☐ **Le schéma**
 - ☐ **La non-redondance**

1. **L'identification par les données**
2. **L'atomicité**
3. **Le schéma**

Définition

Il est nécessaire d'exprimer les règles de cohérence concernant les données *a priori*.

L'expression de ces règles permet de déléguer leur contrôle au système.


Ainsi une base de données relationnelle est assurée d'être, en toutes circonstances, cohérente relativement à son schéma.



Relationnel et contrôle de l'intégrité des données

Rappel

Le relationnel propose un modèle théorique puissant et simple pour gérer le contrôle et l'intégrité des données. On notera en particulier les notions suivantes :

- 
- ☐ Identification par les données
 - ☐ L'atomicité
 - ☐ Le schéma
 - ☐ La non-redondance

1. L'identification par les données
2. L'atomicité
3. Le schéma
4. La non-redondance

Définition

Toute information n'est inscrite qu'une seule fois dans la base de données. On utilise un principe de décomposition pour éviter la redondance (l'information est découpée au sein de plusieurs tables). On utilise l'opération de jointure pour retrouver l'information consolidée.

Cette règle est issue de la théorie de la normalisation relationnelle (et en particulier de la troisième forme normale ou 3NF). Elle permet de faire en sorte que la base ne contiennent jamais d'information contradictoire. En effet si on autorise une donnée à être inscrite plusieurs fois alors il arrivera que des inscriptions soient contradictoire : la donnée aura plusieurs valeurs différentes. Dans ce cas précis, on ne peut plus décider quelle valeur est la bonne, ni même si l'une des valeurs inscrites est la bonne : l'information est perdue.

Exercice : Appliquer la notion

Soit la base de données suivante représentant des recettes de cuisine :

```
CREATE TABLE recipe (  
  name VARCHAR(255),  
  ingredient VARCHAR(255) NOT NULL,  
  PRIMARY KEY (name, ingredient)  
INSERT INTO recipe VALUES (  
  'Mayonnaise',  
  '2, cl, huile'  
INSERT INTO recipe VALUES (  
  'Mayonnaise',  
  '1, cl, vinaigre'  
INSERT INTO recipe VALUES (  
  'Mayonnaise',  
  '1, cl, moutarde'
```

Name	ingrédient
•Mayonnaise	2, cl, huile
•Mayonnaise	1, cl, vinaigre
•Mayonnaise	1, cl, moutarde

Question 1

La base n'est pas en première forme normale car un de ses attributs n'est pas atomique, lequel ?

Question 2

Décomposer cet attribut en trois attributs pour respecter la règle d'atomicité.



جامعة اللبنانية الدولية
LEBANESE INTERNATIONAL UNIVERSITY

CSIT461

Chapitre 5

SGBD-R Oracle

EL BENANY Mohamed Mahmoud

20/12/2021

INTRODUCTION

**Et rendez-vous au chapitre 5 sur Oracle 9g
Introduction essentielle du chapitre 5
Oracle 9g**

ORACLE est un SGBD supportant le model relationnel

Le modèle relationnel permet de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles.

Cette modélisation repose sur des principes mathématiques mis en avant par E.F. Codd est souvent retranscrite physiquement (« implémentée ») dans une base de données.

Les postes principaux d'un model relationnel sont:

1. Une collections des objets(relations)
2. opérateurs qui agissent sur les relations
3. opérations d'intégrité des données (exactitude, cohérence et atomicité)

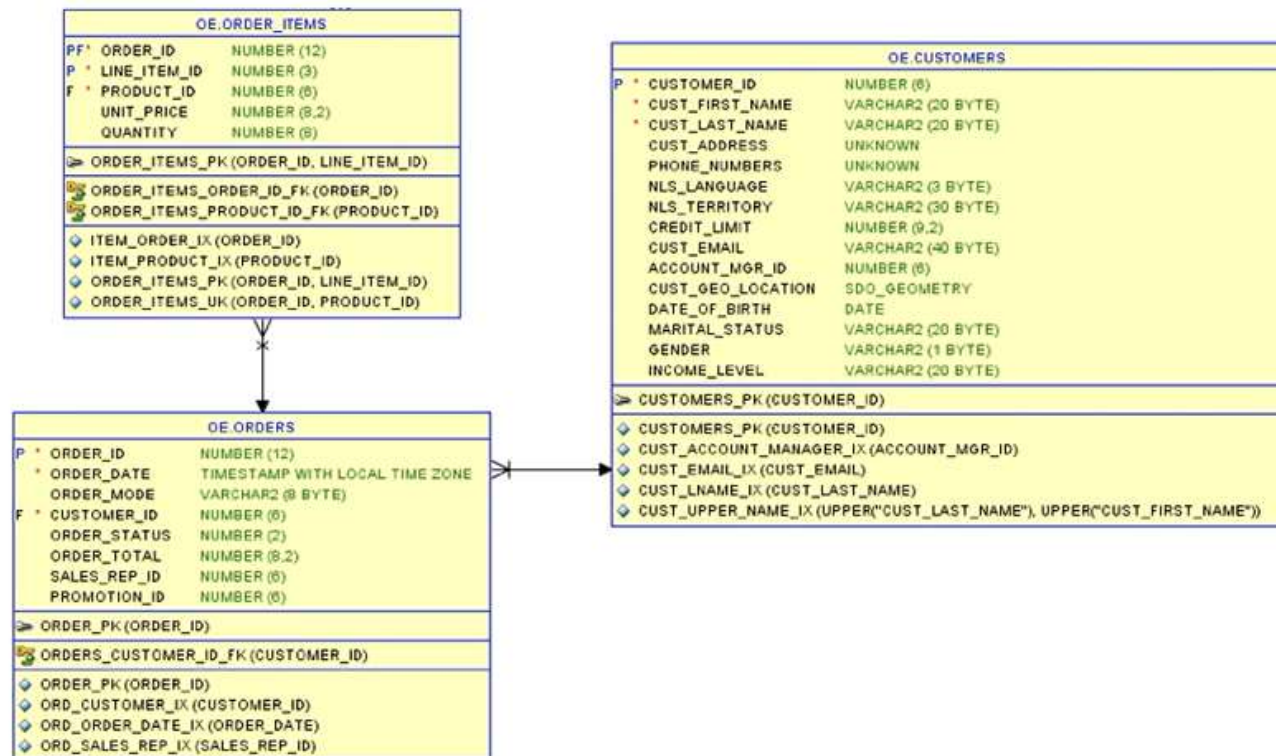
Relations, Opérateurs et l'intégrité de données

Projection:

Select customer_id,
phone_number
from customers;

Selections:

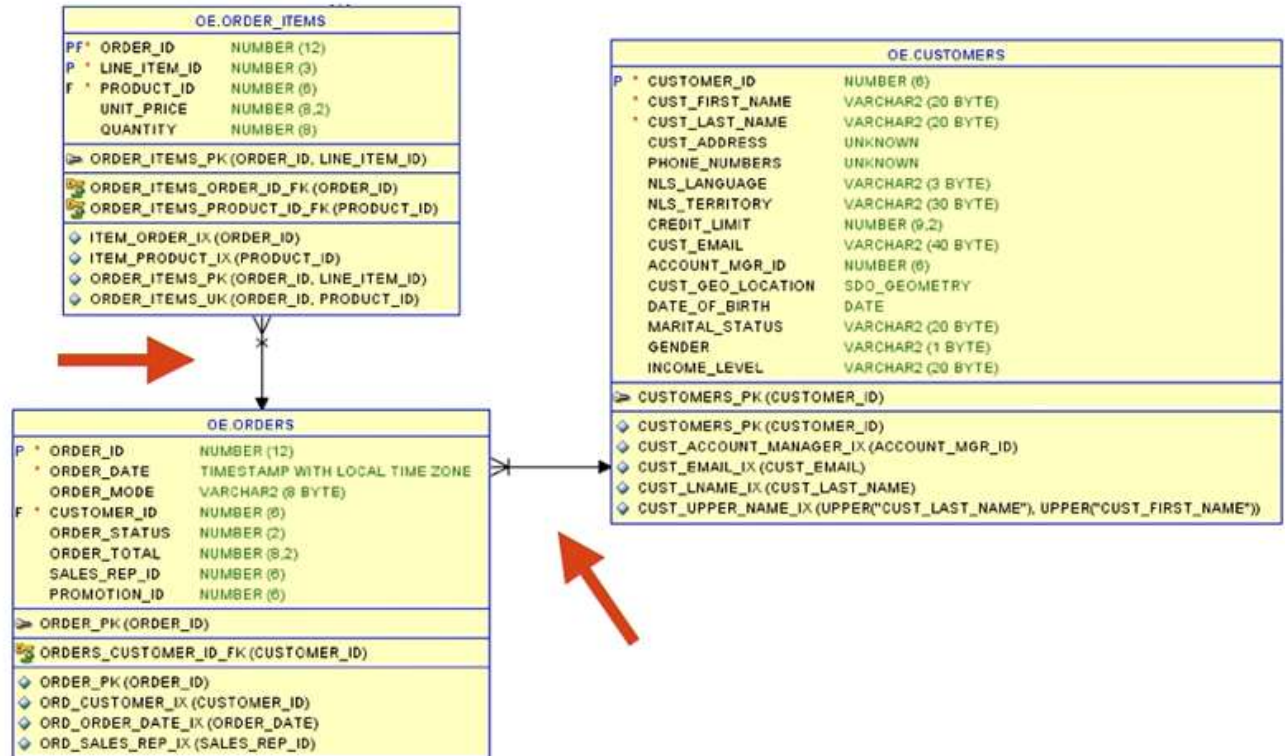
Select odrer_id,
ine_item_id from
order_items where
unit_price >49,99;



Relations, Operateurs et l'intégrité de données

Un article de commande doit avoir une commande associée

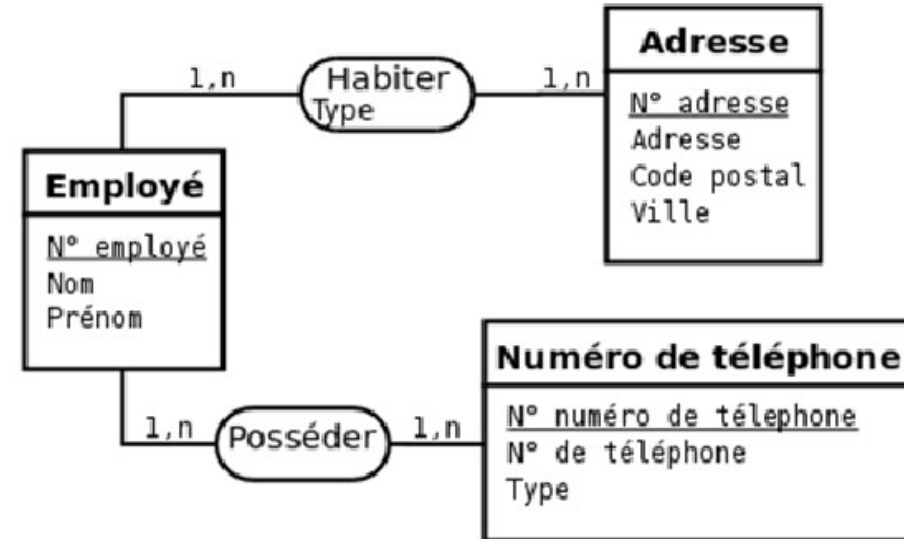
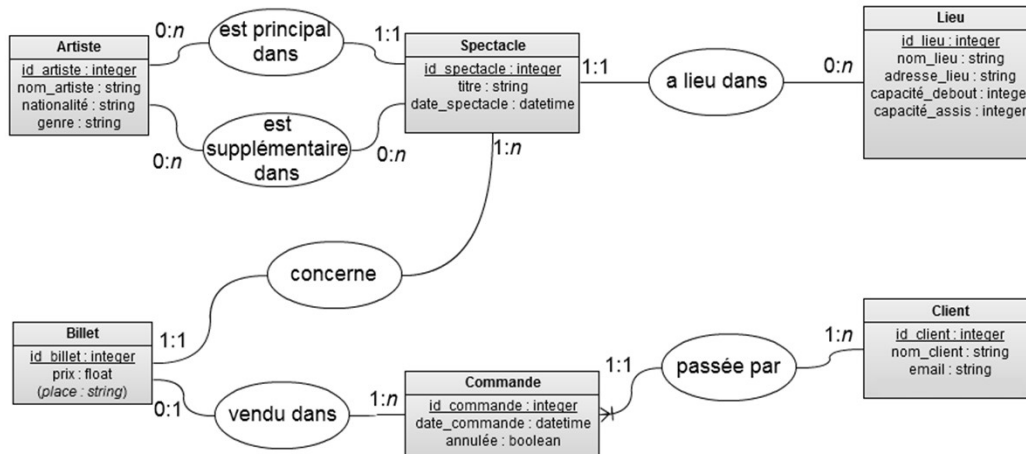
Une commande doit avoir un client pour la passer



Model entité-association

1. Entité
2. Association
3. Attributs

Schéma Entité-Association mis à jour



Exemple des tables dans le serveur de BDD

CUSTOMERS

CUSTOMER_ID	CUST_FIRST_NAME	CUST_LAST_NAME	CREDIT_LIMIT	CUST_EMAIL
101	Constantin	Welles	100	Constantin.Welles@ANHINGA.EXAMPLE.COM
102	Harrison	Pacino	100	Harrison.Pacino@ANI.EXAMPLE.COM
103	Manisha	Taylor	100	Manisha.Taylor@AUKLET.EXAMPLE.COM
104	Harrison	Sutherland	100	Harrison.Sutherland@GODWIT.EXAMPLE.COM
105	Matthias	MacGraw	100	Matthias.MacGraw@GOLDENEYE.EXAMPLE.COM
106	Matthias	Hannah	100	Matthias.Hannah@GREBE.EXAMPLE.COM
107	Matthias	Cruise	100	Matthias.Cruise@GROSBEAK.EXAMPLE.COM
108	Meenakshi	Mason	100	Meenakshi.Mason@JACANA.EXAMPLE.COM
109	Christian	Cage	100	Christian.Cage@KINGLEY.EXAMPLE.COM
110	Charlie	Sutherland	200	Charlie.Sutherland@LIMPKIN.EXAMPLE.COM
111	Charlie	Pacino	200	Charlie.Pacino@LONGSPUR.EXAMPLE.COM

ORDER_ITEMS

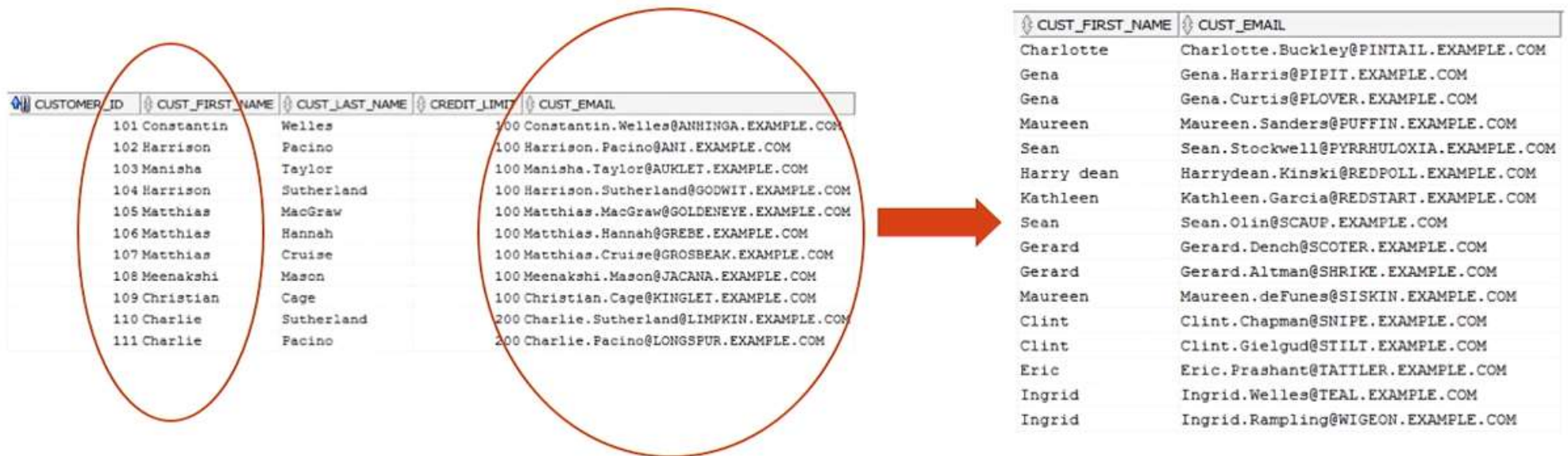
ORDER_ID	LINE_ITEM_ID	PRODUCT_ID	UNIT_PRICE	QUANTITY
2354	1	3106	48	61
2354	2	3114	96.8	43
2354	3	3123	79	47
2354	4	3129	41	47
2354	5	3139	21	48
2354	6	3143	16	53
2354	7	3150	17	58
2354	8	3143	30	61
2354	9	3165	37	64
2354	10	3167	51	68
2354	11	3170	149.2	70
2354	12	3176	113.3	72
2354	13	3182	61	77

ORDERS

ORDER_ID	ORDER_DATE	CUSTOMER_ID	ORDER_TOTAL
2458	2007-08-16 04:34:12.234359000 PM	101	78279.6
2397	2007-11-19 04:41:54.696211000 PM	102	42283.2
2454	2007-10-02 06:49:34.678340000 PM	103	6653.4
2354	2008-07-14 07:18:23.234567000 PM	104	46257
2358	2008-01-08 07:03:12.654278000 PM	105	7826
2381	2008-05-14 09:59:08.843679000 PM	106	23034.6
2440	2007-08-31 10:53:06.008765000 PM	107	70576.9
2357	2006-01-08 10:19:44.123456000 PM	108	59872.4
2394	2008-02-10 11:22:35.564789000 PM	109	21863
2435	2007-09-03 12:22:53.134567000 AM	144	62303
2455	2007-09-20 12:34:11.456789000 PM	145	14087.5
2379	2007-05-16 03:22:24.234567000 AM	146	17848.2
2396	2006-02-02 03:34:56.345678000 AM	147	34930
2406	2007-06-29 05:41:20.098765000 AM	148	2854.2
2434	2007-09-13 06:49:30.647893000 AM	149	268651.8

Exemple des tables dans le serveur de BDD: Operateurs

Projections



Exemple des tables dans le serveur de BDD: Operateurs

Jointure

ORDERS

ORDER_ID	ORDER_DATE	CUSTOMER_ID	ORDER_TOTAL
2458	2007-08-16 04.34.12.234359000 PM	101	78279.6
2397	2007-11-19 04.41.54.696211000 PM	102	42283.2
2454	2007-10-02 06.49.34.678340000 PM	103	6653.4
2354	2008-07-14 07.18.23.234567000 PM	104	46257
2358	2008-01-08 07.03.12.654278000 PM	105	7826
2381	2008-05-14 09.59.08.843679000 PM	106	23034.6
2440	2007-08-31 10.53.06.008765000 PM	107	70576.9
2357	2006-01-08 10.19.44.123456000 PM	108	59872.4
2394	2008-02-10 11.22.35.564789000 PM	109	21863
2435	2007-09-03 12.22.53.134567000 AM	144	62303
2455	2007-09-20 12.34.11.456789000 PM	145	14087.5
2379	2007-05-16 03.22.24.234567000 AM	146	17848.2
2396	2006-02-02 03.34.56.345678000 AM	147	34930
2406	2007-06-29 05.41.20.098765000 AM	148	2854.2
2434	2007-09-13 06.49.30.647893000 AM	149	268651.8

ORDER_ITEMS

ORDER_ID	LINE_ITEM_ID	PRODUCT_ID	UNIT_PRICE	QUANTITY
2354	1	3106	48	61
2354	2	3114	96.8	43
2354	3	3123	79	47
2354	4	3129	41	47
2354	5	3139	21	48
2354	6	3143	16	53
2354	7	3150	17	58
2354	8	3163	30	61
2354	9	3165	37	64
2354	10	3167	51	68
2354	11	3170	145.2	70
2354	12	3176	113.3	72
2354	13	3182	61	77
2397	1	2976	52	2
2397	2	2986	120	8
2397	3	2999	880	16
2397	4	3000	1696.2	16
2454	1	2289	43	120

CUSTOMER_ID	ORDER_ID	ORDER_DATE	LINE_ITEM_ID	PRODUCT_ID	QUANTITY	UNIT_PRICE
102	2397	2007-11-19 04.41.54.696211000 PM	1	2976	2	52
102	2397	2007-11-19 04.41.54.696211000 PM	2	2986	8	120
102	2397	2007-11-19 04.41.54.696211000 PM	3	2999	16	880
102	2397	2007-11-19 04.41.54.696211000 PM	4	3000	16	1696.2

Exemple des tables dans le serveur de BDD: (contraint) l'intégrité de donnée

Ne doit pas être nulle :
note NULL

CUSTOMERS

CUSTOMER_ID	DECODE(CUST_FIRST_NAME,'MANISHA','',CUST_FIRST_NAME)	DECODE(CUST_LAST_NAME,'MACGRAW','',CUST_LAST_NAME)	CREDIT_LIMIT	CUST_EMAIL
101	Constantin	Welles	100	Constantin.Welles@ANHINGA.EXAMPLE.COM
102	Harrison	Pacino	100	Harrison.Pacino@ANI.EXAMPLE.COM
103		Taylor	100	Manisha.Taylor@AUKLET.EXAMPLE.COM
104	Harrison	Sutherland	100	Harrison.Sutherland@GODWIT.EXAMPLE.COM
105	Matthias		100	Matthias.MacGraw@GOLDENEYE.EXAMPLE.COM
106	Matthias	Hannah	100	Matthias.Hannah@GREBE.EXAMPLE.COM
107	Matthias	Cruise	100	Matthias.Cruise@GROSBEAK.EXAMPLE.COM
108	Meenakshi	Mason	100	Meenakshi.Mason@JACANA.EXAMPLE.COM
109	Christian	Cage	100	Christian.Cage@KINGLET.EXAMPLE.COM
110	Charlie	Sutherland	200	Charlie.Sutherland@LIMPKIN.EXAMPLE.COM
111	Charlie	Pacino	200	Charlie.Pacino@LONGSPUR.EXAMPLE.COM

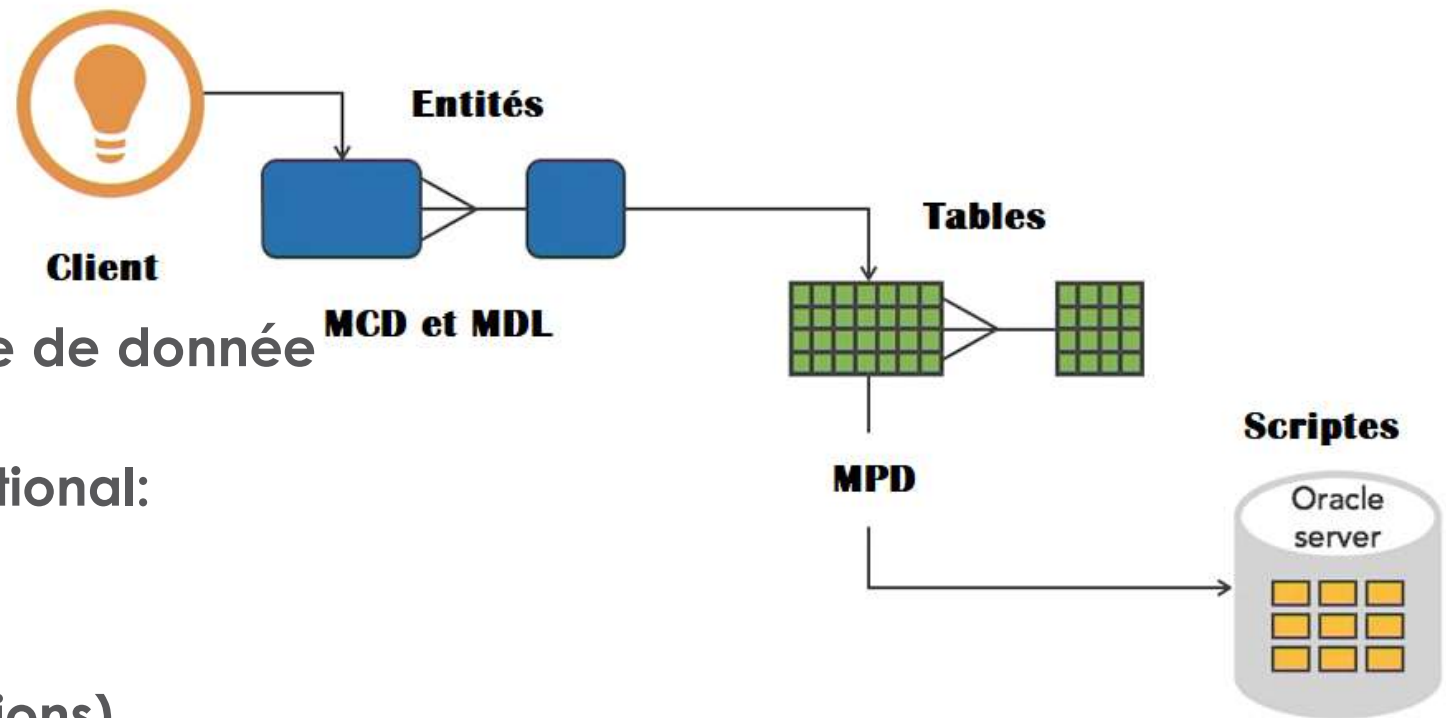
**Exemple des tables dans le serveur de BDD:
(contraint) référentiel des données**

Clé étrangère

ORDER_ID	ORDER_DATE	CUSTOMER_ID	ORDER_TOTAL
2458	2007-08-16 04.34.12.234359000 PM	101	78279.6
2397	2007-11-19 04.41.54.696211000 PM	102	42283.2
2454	2007-10-02 06.49.34.678340000 PM	103	6653.4
2354	2008-07-14 07.18.23.234567000 PM	104	46257
2358	2008-01-08 07.03.12.234567000 PM	105	7826
2381	2008-05-14 09.59.08.843679000 PM	106	23034.8
2440	2007-08-31 10.53.06.008765000 PM	107	70576.9
2357	2006-01-08 10.19.44.123456000 PM	108	59872.4
2394	2008-02-10 11.22.35.564789000 PM	109	21863
2435	2007-09-03 12.22.53.134567000 AM	144	62303
2455	2007-09-20 12.34.11.456789000 PM	145	14087.5
2379	2007-05-16 03.22.24.234567000 AM	146	17848.2
2396	2006-02-02 03.34.56.345678000 AM	147	34930
2406	2007-06-29 05.41.20.098765000 AM	148	2854.2
2434	2007-09-13 06.49.30.647893000 AM	149	268651.8

ORDER_ID	LINE_ITEM_ID	PRODUCT_ID	UNIT_PRICE	QUANTITY
2354	1	3106	48	61
2354	2	3114	96.8	43
2354	3	3123	79	47
2354	4	3129	41	47
2354	5	3139	21	48
2354	6	3143	16	53
2354	7	3150	17	58
2354	8	3163	30	61
2354	9	3165	37	64
2354	10	3167	51	68
2354	11	3170	145.2	70
2354	12	3176	113.3	72
2354	13	3182	61	77
2397	1	2976	52	2
2397	2	2986	120	8
2397	3	2999	880	16
2397	4	3000	1696.2	16
2454	1	2289	43	120

De l'idée à la base de données : C'est la Modélisation



ℳ Oracle est une base de donnée relationnelle;

ℳ Support le Model rational:

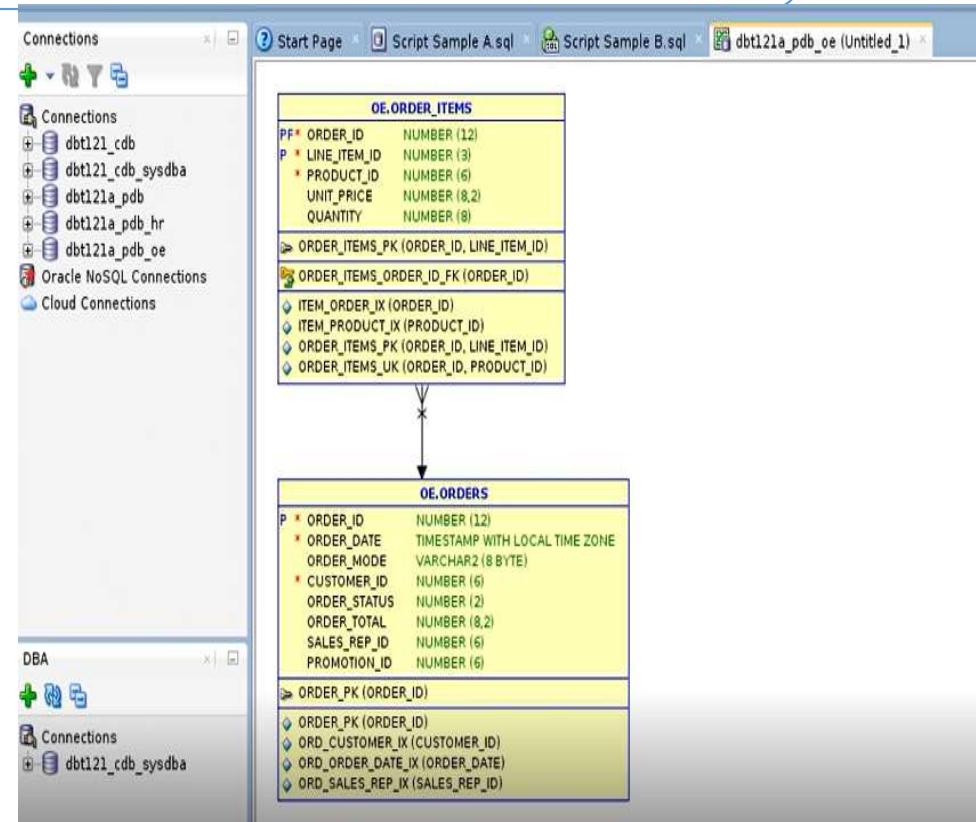
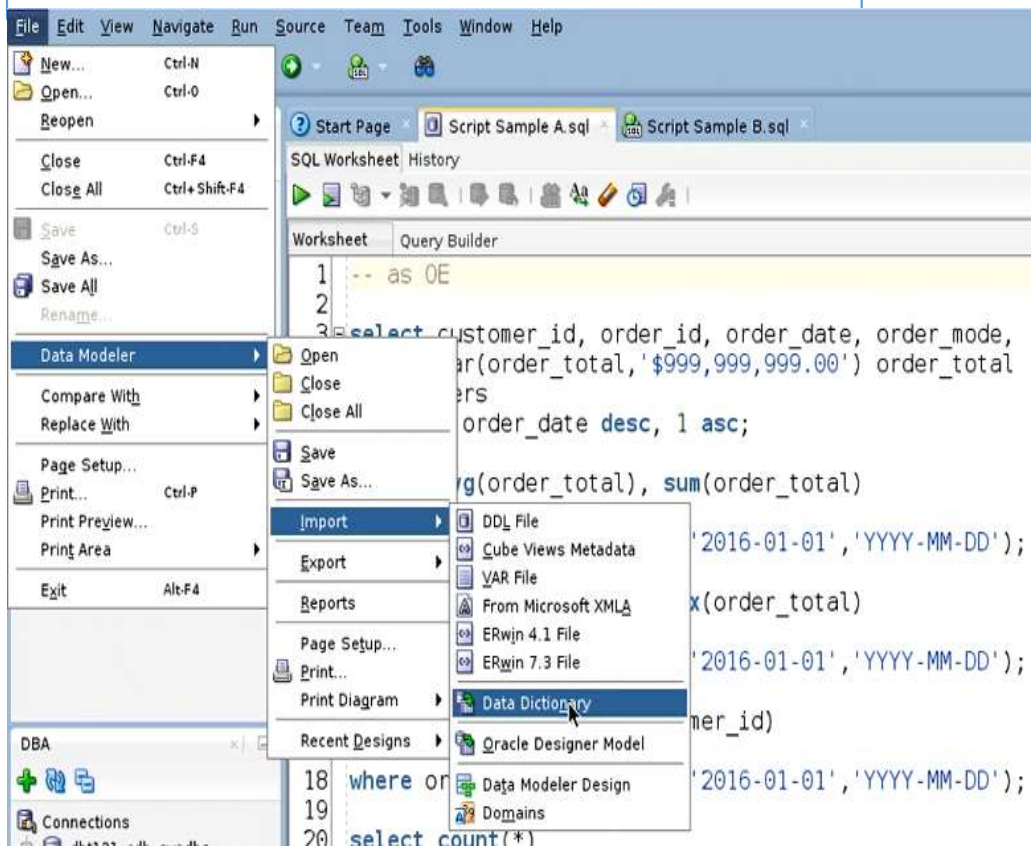
- Tables(entités)
- Linge (tuplet)
- Colone(projections)
- Champs(intersection)
- Value null

Modélisations E/A : Outils

1. JMerise
2. Looping(recommander)
3. ER/Studio(Embarcadero)
4. Erwin(Computer Associate)
5. Oracle Data Modeler
 - Intégrer dans Oracle SQL Developer

SQL Developer

Modeler



SQL Oracle: Outils

1. SQL*Plus
2. Oracle SQL Developer
3. SQL CMD

SQL

1. Langage déclarative

PL/SQL

1. Langage procédural

```
declare
  cursor bq is
    SELECT
      /*+ monitor noparallel */
      * from dual;
  type bqa is table of bq%rowtype;
  bqb bqa;
  rowcount      number := 0;
begin
  open bq;
  loop
    fetch bq bulk collect into bqb limit 5000;
    rowcount := rowcount + bqb.count;
    exit when bq%notfound;
  end loop;
  close bq;
  dbms_output.put_line('Rows: ' || rowcount);
end;
/
```

SQL Developer

Démonstration