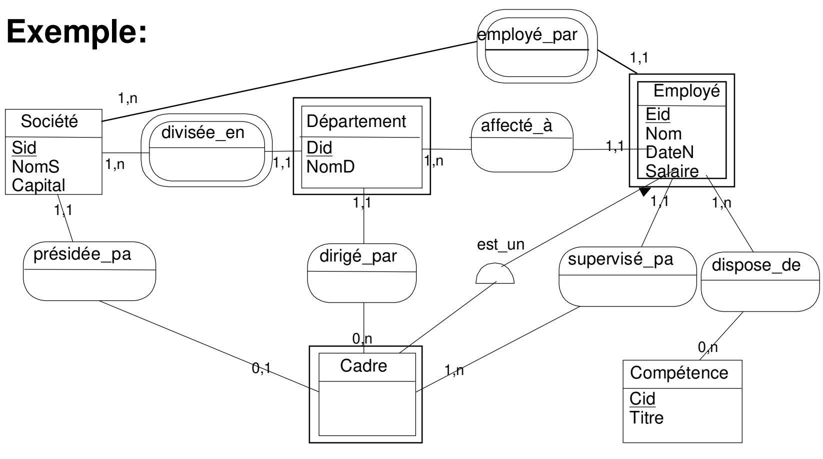
Où en sommes-nous ?

- **Tout est relation**
  - les tables mémorisant les données
  - les résultats des requêtes, mémorisables dans
    - » des vues
    - » des tables temporaires (détruites en fin de session)
- **Redondances => Normalisation**
  - Eviter les anomalies lors des mises à jour de la BD
  - Décomposer sans perte d 'information ni de DF
- **Normalisation => Cohérence**
  - Dans une relation
    - » tout dépend de la clé (par définition de la PK)
    - » de toute la clé (2ème FN)
    - » rien que de la clé (3ème FN)

De la conception E/R à la conception relationnelle

- **Exemple:**



**Etape 1: relations / entités indépendantes (« fortes »)
clés primaires**

Un schéma de BD relationnelle :
des tables et des contraintes d'intégrité

- SOCIETE(<u>SID</u>, NOMS, CAPITAL)
- create table SOCIETE(SID integer primary key,
  NOMS varchar(32),
  CAPITAL integer)
- COMPETENCE(<u>CID</u>, TITRE)
- create table COMPETENCE (CID integer,
  TITRE varchar(32),
  primary key (CID))

**Etape 2: relations / entités dépendantes (« faibles »), sous-entités
clés étrangères**

- DEPARTEMENT(<u>SID, DID</u>, NOMD)
- create table DEPARTEMENT (SID integer, DID integer,
  NOMD varchar(32), primary key (SID,DID),
  foreign key (SID) references SOCIETE)
- EMPLOYE(<u>SID, EID</u>, NOM, DATEN , SALAIRE)

- CADRE(<u>SID, EID</u>)
- create table CADRE (SID integer, EID integer,
  primary key (SID, EID),
  foreign key (SID, EID) references EMPLOYE)

Etape 3: attributs / associations 1-1, 1-N
clés étrangères

- SOCIETE(SID, NOMS, CAPITAL, P_SID, P_EID)
- alter table SOCIETE add (P_SID integer, P_EID integer,
  foreign key (P_SID, P_EID)
  references CADRE)

- DEPARTEMENT(SID, DID, NOMD, [D_SID,] D_EID)
- EMPLOYE(SID, EID, NOM, DATEN, SALAIRE
  [A_SID,] A_DID, [S_SID,] S_EID)

Mars 2001 saglio@enst.fr 5

Etape 4: relations / associations N-M ou n-aires

- DISPOSITION(SID, EID, CID)
- create table DISPOSITION(

```
        SID integer,
        EID integer,
        CID integer,
        primary key (SID, EID, CID),
        foreign key (SID, EID) references EMPLOYE,
        foreign key (CID) references COMPETENCE
            )
```

Mars 2001 saglio@enst.fr 6

Intégrité d'une BD relationnelle

- **attribut(tuple) -> domaine de valeurs**
  - **NULL ou NOT NULL**
  - **préciser le domaine au delà des types de base, par exemple :**
    check CAPITAL > 10 000
    check TITRE in (' Technique ', ' Financière ', ' Humaine ')
- **clé (primaire ou secondaire)**
  par exemple, primary key (SID) dans la table SOCIETE :
  - **check SID not null**
  - **check SID unique**
- **clé étrangère**
  par exemple, dans la table DEPARTEMENT, foreign key (SID) references SOCIETE :
  - π_{SID} **(DEPARTEMENT)** $\subseteq \pi_{SID}$ **(SOCIETE)**

Intégrité : autres expressions

Exprimer le fait qu'un employé ne peut pas gagner plus que celui qui le supervise

- create assertion HIERARCHIE_SALAIRES check
  (not exists (select * from EMPLOYE E, EMPLOYE S
  where E.S_EID=S.EID
  and E.SALAIRE > S.SALAIRE))

Exprimer que le salaire moyen ne doit pas dépasser 200000

- create assertion SALAIRE_MOYEN check (200000 >
  (select avg(SALAIRE) from EMPLOYE))

Comment exprimer qu'un salaire ne peut pas diminuer?

Nommer les contraintes d'intégrité - SQL92

```
CREATE TABLE EMPLOYE
    (SID INTEGER NOT NULL DEFAULT ' ENST ',
     EID INTEGER NOT NULL UNIQUE,
    NOM CHARACTER(30) NOT NULL ,
    SALAIRE DECIMAL(9,2) CHECK (SALAIRE> 7000) CONSTRAINT smic,
    ...
    CONSTRAINT grille_salaires CHECK (SALAIRE < SELECT MIN (SALAIRE)
        FROM CADRE)
    CONSTRAINT e_pk PRIMARY KEY (SID, EID)
    CONSTRAINT e_fk FOREIGN KEY (SID) REFERENCES SOCIETE)
CREATE ASSERTION masse_salariale
    CHECK (select SUM(SALAIRE)  FROM EMP < 500000)
```

Le nommage des CI permet de les annuler/remplacer ou les inhiber

Mars 2001 saglio@enst.fr 9

Intégrité : comment et quand contrôler?

- comment ?
  - par les applications
    - » les CI sont seulement des spécifications pour TOUTES les applications
  - par le SGBD
    - » les CI sont des spécifications pour des « réflexes » du SGBD (« triggers »)
- quand ?
  - après chaque action
    - » possible pour certaines CI seulement
  - seulement à la fin de « transactions »
    - » **Transaction = suite d'actions faisant passer la BD d'un état cohérent à un autre état cohérent**
    - » (au milieu de la transaction la BD n 'est pas nécessairement dans un état cohérent)

Mars 2001 saglio@enst.fr 10

Les triggers: une longue histoire

• Les procédures de Bases de Données en CODASYL (73-77)

on commands (insert, remove, find, store, delete, modify, get)

call procedure (routine du LMD : Cobol)

• System-R (75)

define trigger name **on** event **of** relation **:** SEQUEL statement

• Sybase (88)

create trigger name **on** table **for** event **as** SQL statement

• Ingres (90)

create rule name **after** event **where** condition **execute** procedure name(args)

Mars 2001 saglio@enst.fr 11

Les triggers d'Ingres

CREATE RULE <nom règle>

AFTER <spec>

spec := insert into<nomrel> | update(att) of <nomrel> | delete from <nomrel>

spec := spec, spec

[WHERE <cond>]

cond := expression booléenne portant sur les attributs de <nomrel>

EXECUTE PROCEDURE <nomproc> [(<param>)]

param := nomparam = old|new.attribut

param := param , param

Mars 2001 saglio@enst.fr 12

Les procédures stockées d’Ingres

SQL procédural

= SQL + variables + structures de contrôle + traitement des erreurs

Stockées dans le SGBD

Exécutées dans le noyau

CREATE PROCEDURE <nomproc> (<params>) AS

[DECLARE

<idf> <type ingres>{; <idf> <type ingres>}*]

BEGIN

<suite d'instructions> select, insert, update, delete, =, if else endif, whileendwhile, endloop, message, return, raise error, register dbevent, remove dbevent.

END;

Mars 2001 saglio@enst.fr 13

Trigger Ingres : Sémantique

- **Tuple à tuple**
- **Couplage immédiat**
- **Exécution dans la même transaction**
- **Action: rollback** (par le SGBD) **ou commit** (comme demandé par l ’utilisateur)
- **Déclenchement en cascade : en profondeur d'abord**
- **INGRES : conclusion**

ça marche… : -)

On peut faire beaucoup de choses (contraintes dynamiques)

…bizarrement : - (

couplage immédiat : on ne peut pas vérifier contrainte référentielle, on s’en remet à la cohérence des actions différées à la fin de la transaction

Pas de structuration des règles (toutes au même niveau)

Sémantique difficile à maîtriser

+ **Pbms de performance** (détection des règles actives + sémantique tuple à tuple)

Mars 2001 saglio@enst.fr 14

trigger or not trigger ?

Les triggers : Solution unifiée aux différents problèmes ?

On Event If Condition Then Action

Composants:

- **Détecteur d'événements**
- **Langages**
- **Modèles d'exécution :**

- immédiate => Prévention ou différée => Détection
- tuple à tuple ou ensembliste
- Les triggers **modifient** la transaction utilisateur

  Ce que utilisateur+SGBD font peut être différent de ce que l'utilisateur veut faire
- pbs: consistance sémantique, terminaison, indéterminisme

Les Index et les Vues : outils auxiliaires du contrôle d'intégrité