

Proposition de correction de l'examen du cours *Systemes d'information*

Fabien Coelho et Claire Medrala – Mines Paris – PSL

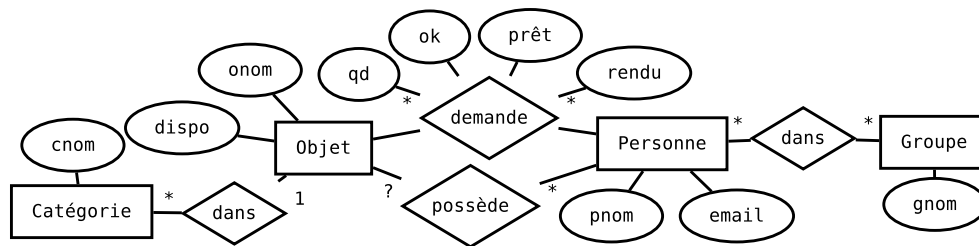
Janvier 2024

Bravo Hobbes! 20/20

1 Modélisation entité-association

5/5

Voici une proposition de modèle E/A pour SPAP, l'application de prêts :



Remarque : il n'est pas clair de savoir si une personnes peut-être dans plusieurs groupes, nous avons choisi l'affirmative.

2 Traduction relationnelle

3/3

A partir du modèle E/A précédent, construisez un modèle relationnel.

```
CT Groupe(gid SPK, gnom TUNN);
CT Personne(pid SPK, pnom TUNN, email TUNN);
CT GroupePersonne(gid INNER Groupe, pid INNER Personne, U(gid,pid));
CT Categorie(cid SPK, cnom TUNN);
CT Objet(oid SPK, onom TNN, dispo BNN, cid INNER Categorie, pid IR Personne,
        U(onom,pid));
CT Demande(did SPK, pid INNER Personne, oid INNER Objet, quand TsNN,
        ok Ts, pret Ts, rendu Ts, U(pid,oid,quand));
```

3 Requêtes et index

8/8

1. Quels sont les noms des étapes du métier *Peintre* auxquelles intervient l'entreprise *Gauguin* sur le chantier *Piscine* sur la commune de *Briare* (45250).

-- attention à l'ordre car commune lié à Chantier et à Entreprise

```
SELECT etnom
FROM Metier
JOIN Etape USING (mid)
JOIN Chantier USING (cid)
JOIN Commune USING (coid) -- on veut la commune du CHANTIER
JOIN Entreprise USING (eid)
WHERE enom = 'Gauguin'
      AND mnom = 'Peintre'
      AND cnom = 'Piscine'
      AND conom = 'Briare' -- non unique
      AND cocp = 45250; -- non unique
```

2. Suggestions d'index pour cette requête :

```
-- 1. en partant du nom ou du cp la commune
-- Commune(conom) couvert par U(conom, cocp)
CREATE INDEX CommuneCP ON Commune(cocp);
CREATE INDEX ChantierCoid ON Chantier(coid);
CREATE INDEX EtapeCid ON Etape(cid);
-- Entreprise(eid) PK, Métier(mid) PK

-- 2. en partant du métier
-- Métier(mnom) U
CREATE INDEX EtapeMid ON Etape(mid);
-- Chantier(cid) PK, Entreprise(eid) PK, Commune(coid) PK

-- 3. en partant du nom de l'entreprise
-- Entreprise(enom) couvert par U(enom, coid)
CREATE INDEX EtapeEid ON Etape(eid);
-- Chantier(cid) PK, Commune(coid) PK, Métier(mid) PK

-- 4. en partant du nom du chantier
-- Chantier(cnom) couvert par U(cnom, coid)
-- CREATE INDEX EtapeCid ON Etape(cid); (voir 1.)
-- Commune(coid) PK, Métier(mid) PK, Entreprise(eid) PK
```

3. Combien de chantiers démarrés en 2023 impliquent des entreprise du département 29 (2 premiers chiffres du CP) ?

```
-- compter chaque chantier une seule fois, sinon compte les étapes
SELECT COUNT(DISTINCT cid) AS "Chantiers 2023 des entreprises du Finistère"
-- attention à l'ordre des jointures, commune de l'entreprise
FROM Commune
JOIN Entreprise USING (coid)
JOIN Etape USING (eid)
JOIN Chantier USING (cid)
WHERE cocp / 1000 = 29
AND EXTRACT(YEAR FROM cdebut) = 2023;
```

4. Afficher pour toutes les communes le nombre de chantiers et le nombre total d'étapes dans l'ordre des CP et des noms.

```
-- attention pas de COUNT(*) avec LEFT JOIN
SELECT cocp, conom,
COUNT(DISTINCT cid) AS "# chantiers",
COUNT(eid) AS "# étapes"
FROM Commune
LEFT JOIN Chantier USING (coid)
LEFT JOIN Etape USING (cid)
GROUP BY 1, 2
ORDER BY 1, 2;
```

5. Afficher les noms et la ville (CP et nom de commune) des entreprises qui ne sont jamais intervenues dans un chantier.

```
-- version avec jointure externe
```

```
SELECT enom, cocp, conom
FROM Entreprise
JOIN Commune USING (coid)
LEFT JOIN Etape USING (eid)
WHERE etid IS NULL;
```

```
-- version avec opérateur ensembliste
```

```
SELECT enom, cocp, conom
  FROM Entreprise
  JOIN Commune USING (coid)
EXCEPT
SELECT enom, cocp, conom
  FROM Entreprise
  JOIN Commune USING (coid)
  JOIN Etape USING (eid);
```

6. Afficher les noms et communes (CP et nom) des chantiers dont l'écart entre coût réel et le devis est maximal ou minimal. Une première colonne indique s'il s'agit de l'écart *min* ou *max*, une autre colonne indique le montant de l'écart.

```
WITH
```

```
  CoutChantier AS (
    SELECT cid, SUM(etdevis) AS devis, SUM(etreel) AS reel
    FROM Chantier
    JOIN Etape USING (cid)
    GROUP BY 1
  )
```

```
-- attention, il peut y avoir des égalités
```

```
SELECT 'max' AS dépassement, cnom, cocp, conom, reel-devis AS montant
  FROM Commune
  JOIN Chantier USING (coid)
  JOIN CoutChantier USING (cid)
  WHERE reel-devis = (SELECT MAX(reel-devis) FROM CoutChantier)
UNION
SELECT 'min', cnom, cocp, conom, reel-devis
  FROM Commune
  JOIN Chantier USING (coid)
  JOIN CoutChantier USING (cid)
  WHERE reel-devis = (SELECT MIN(reel-devis) FROM CoutChantier);
```

7. Pour *tous* les chantiers et *tous* les métiers, afficher le pourcentage du coût réel de ce métier dans le coût total réel du chantier, par ordre des chantiers (nom du chantier, nom et CP de la commune associée) et des noms de métiers.

```
-- version avec fonctions de fenêtres
WITH -- pour chaque chantier le % de coût de chaque corps de métier
  PcMetierChantier AS (
    SELECT DISTINCT cid, mid,
      ROUND((100.0 *
        SUM(etreel) OVER (PARTITION BY cid, mid) /
        SUM(etreel) OVER (PARTITION BY cid))::NUMERIC, 1) AS "%"
    FROM Etape)
SELECT cnom, cocp, conom, mnom, COALESCE("%", 0.0)
FROM Chantier CROSS JOIN Metier -- des % pour tous ceux là
JOIN Commune USING (coid)
LEFT JOIN PcMetierChantier USING (cid, mid)
ORDER BY 1, 2, 3, 4;

-- version sans fonction de fenêtres
WITH
  ChantierMetierCout AS ( -- calcul des coûts par métiers
    SELECT cid, mid, SUM(etreel) AS cout
    FROM Chantier CROSS JOIN Metier
    LEFT JOIN Etape USING (cid, mid)
    GROUP BY 1, 2),
  ChantierTotal AS ( -- calcul du total du chantier
    SELECT cid, SUM(etreel) AS total
    FROM Chantier LEFT JOIN Etape USING (cid)
    GROUP BY 1)
-- rejoinctures pour le calcul du %
SELECT cnom, cocp, conom, mnom,
  ROUND(100.0 * COALESCE(cout / total, 0.0)::NUMERIC, 1) AS "%"
FROM ChantierMetierCout
JOIN Chantier USING (cid)
JOIN Commune USING (coid)
JOIN Metier USING (mid)
JOIN ChantierTotal USING (cid)
ORDER BY 1, 2, 3, 4;
```

4 Questions de cours

4/4

Thème : ACID (*Advanced Chanting In Databases*)

Ce thème est très passionnant, j'ai appris plein de choses super intéressantes qui éclairent parfaitement le fonctionnement de cet aspect des bases de données à la fois théorique et pratique, et me permettent de bien comprendre les caractéristiques générales et particulières des multiples facettes de ce thème dans une perspective transversale de mise en application concrète du modèle relationnel sur des problèmes réels, tout en gardant, au delà du simple niveau fonctionnel, une maîtrise généraliste des aspects opérationnels sur Postgres qui gère le stockage à un prix compatible avec toutes les bourses, une bonne nouvelle pour nos budgets !

Il y a 3 prix Turing liés aux recherches sur les bases de données relationnelles : *Edgard Codd*, *James Gray* et *Michael Stonebraker*.