**Vue**

**Présentation**

Une vue est une table virtuelle. On peut faire des SELECT et, sous certaines conditions, on peut ajouter, modifier ou supprimer des tuples dans une vue et par conséquent dans la ou les tables associées à la vue.

Le « contenu » d’une vue n’a pas d’existence physique : il est recalculé à chaque utilisation de la vue qui se comporte donc comme un **SELECT**.

Une vue peut être considérée comme une fonction retournant une table.

**Gestion des vues**

* La syntaxe de la création d'une vue est la suivante :

CREATE VIEW nom\_vue AS

SELECT …

* La syntaxe de la suppression d'une vue est la suivante :

DROP VIEW nom\_vue ;

**Les 2 usages des vues**

**1 : décomposer les requêtes en sous-requêtes**

Une vue remplace n’importe quelle table dans un SELECT.

Les vues permettent ainsi de décomposer l’écriture des requêtes complexes.

***Exemple***

On cherche tous les employés travaillant dans un département qui contient au moins un 'ANALYSTE’.

On commence par créer la vue correspondant à la requête : « tous les départements contenant au moins un ANALYSTE :

CREATE or REPLACE VIEW deptAvecAnalyste as

SELECT DISTINCT département FROM employés

WHERE poste = 'ANALYSTE'

ORDER BY département;

Ensuite on traite la requête principale en exploitant la nouvelle vue :

SELECT DISTINCT E.nom

FROM employé E, deptAvecAnalyste D

WHERE E.département=D.département;

**2 : Faciliter et sécuriser l’accès aux données par les utilisateurs**

On peut créer des vues et ne donner des droits d’accès que sur ces vues : ça favorise la sécurisation et ça facilite l’accès, les vues étant orientées « usage ».

La vue permet aussi de restreindre l’accès aux tuples.

**Exemple**

On veut limiter les droits d’accès de l’utilisateur « Dupond » à la consultation du nom et du poste des employés des départements 10 et 20.

CREATE VIEW vueEmp AS

SELECT nom, poste

FROM employés

WHERE département in (10, 20);

GRANT SELECT ON vueEmp TO Dupond;

**Les vues modifiables**

INSERT, UPDATE et REPLACE dans une vue

**Vue mono-table**

On peut faire des INSERT, des UPDATE et des REPLACE dans une vue mono-table : ils passeront dans la table correspondante, à condition que toutes les contraintes d’intégrité soient correctement prises en compte.

**Vue multi-table**

On peut faire des INSERT, un UPDATE ou un REPLACE dans une vue multi-table à condition que toutes les contraintes d’intégrité de la table correspondante soient correctement prises en compte.

Mais on ne peut créer, modifier ou remplacer qu’un seul tuple à la fois.

**DELETE dans une vue**

**Vue mono-table**

On peut faire des DELETE dans une vue mono-table : ils passeront dans la table correspondante, à condition que toutes les contraintes d’intégrité soient correctement prises en compte.

**Vue multi-table**

On ne peut pas faire de DELETE dans une vue multi-table.

**Exemple 1**

Créer une vue ‘comedie’ qui sélectionne les comédies :

CREATE VIEW comedies AS

SELECT \*

FROM films

WHERE genre = 'Comédie';

**Exemple 2**

Créer une vue ‘ventimp’ qui liste les factures dont le montant dépasse 1500 euros, et affiche le nom de l’employé ayant fait la vente :

CREATE VIEW ventimp AS

SELECT numfact, nomemp, prixfact

FROM facture F, employe E

WHERE F.numemp=E.numemp AND (quantfact\*prixfact)>1500;

**Vérification**

SELECT \*

FROM ventimp

ORDER BY numfact;

**Exemple 3**

Créer une vue gde\_qte qui sélectionne dans la table « facture » les factures portant sur plus de 25 articles

CREATE VIEW gde\_qte AS

SELECT numfact, quantfact

FROM facture as F, employe AS E

WHERE F.numemp=E.numemp AND quantfact>25

**Exercices : Vues**

Soit le schéma relationnel :

**EMP(Matr, NomE, Poste, DatEmb, Salaire, Commission, NumDept)**

**DEPT(NumDept, NomDept)**

**PROJET(CodeP, NomP)**

**PARTICIPATION(Matr, CodeP, Fonction)**

1. Créez une vue V\_EMP contenant : le matricule, le nom, le numéro de département, la somme de la commission et du salaire baptisée GAINS, le Nom du département.
2. Sélectionnez les lignes de V\_EMP dont le salaire total est supérieur à 10.000€
3. Mettez à jour le salaire de martin « 1700€ » à travers la vue V\_EMP.
4. Créez une vue VEMP10 qui ne contienne que les employés du département 10 de la table EMP. Insérez dans cette vue l’employé (1117, 'SOULIER', ‘COMMERCIAL', '18/05/81', 10000, 300, 20). Sélectionner ensuite cet employé au moyen de la vue VEMP10 puis au moyen de la table EMP.
5. Détruisez cette vue VEMP10.
6. Créer une vue VEMP\_Com contenant le Nom du Département, Nom du projet, et la Fonction des commerciaux.
7. Modifier le nom du département (Rhône) de l’employé Balard.
8. Liste des salaires des employés avec le pourcentage par rapport au total des salaires de leur département (utilisez une vue qui fournira le total des salaires).
9. Créer une vue qui donne le droit à Dupond de modifier que la colonne poste de la table employés.