TP : L’instruction Select

# Objectif du TP

L’objectif de cette séance est de manipuler les données d’une base de données à l’aide de l’instruction Select.

REQUÊTES SUR UNE SEULE RELATION

Soit la Base de Données Cinéma représentée par le schéma relationnel suivant :

INDIVIDU (NUMI, Nom, Prénom, DateNaiss, VilleNaiss)

JOUER (NUMA#, NUMF#, Role, Salaire)

FILM (NUMF, NUMR#, Titre, Genre, Annee, Longueur, Budget)

PROJECTION (NUMC#, NUMF#, DateProj)

CINEMA (NUMC, Nom, Adresse, Ville, NombreSalles)

Exercice 1 Exercices sur Papier

|  |  |
| --- | --- |
| a. Afficher tous les genres de films | SELECT DISTINCT \*  FROM FILM ; |
| b. Afficher les films dont la longueur dépasse 180 min | SELECT \*  FROM FILM  WHERE (Longueur > 180) ; |
| c. Afficher le titre et l’année des films de science‐ fiction dont le budget dépasse 5.000.000 $ | SELECT Titre, Annee  FROM FILM  WHERE (Genre = SciFi) AND (Budget > 5.000.000) ; |
| d. Afficher les films triés par ordre décroissant de budget | SELECT \*  FROM FILM  ORDER BY Budget DESC ; |
| e. Afficher les genres et titres des films réalisés en 2008, triés par ordre alphabétique | SELECT Genre, Titre  FROM FILM  WHERE Annee = 2008  ORDER BY Titre ; |
| f. Afficher les individus qui sont nés à Nice, Paris ou Londres | SELECT Nom, Prenom  FROM INDIVIDU  WHERE VilleNaiss IN (‘Nice’, ‘Paris’, ‘Londres’) ; |
| g. Afficher les individus qui sont nés ailleurs qu’à Nice, Paris et Londres | SELECT Nom, Prenom  FROM INDIVIDU  WHERE NOT (VilleNaiss IN (‘Nice’, ‘Paris’, ‘Londres’) ) ; |
| h. Afficher les titres des films ainsi que leur budget en dollars et en euros, triés par ordre décroissant sur le budget en dollars (1 $ = 0,75  €) | SELECT Titre Budget & ‘$’ AS BudgetEnDollars,  0.75\*Budget & ‘€’ AS BudgetEnEuros  FROM FILM  ORDER BY BudgetEnDollars |
| i. Afficher le nom et la ville des cinémas situés à Paris et dont le nom commence par C, avec un R en troisième lettre. Trié croissant par ville. | SELECT Ville, Nom FROM CINEMA  WHERE (Ville = ‘Paris’ OR Ville = ‘Nice’) AND (Nom LIKE ‘C??R\*’)  ORDER BY Ville ; |
| j. Afficher les dates de projection (de la plus récente à la plus ancienne) des films dont les numéros sont compris entre 5 et 12, ou  supérieurs à 23, du cinéma n°10 | SELECT DateProj  FROM PROJECTION  WHERE (NUMF BETWEEN 5 AND 12) AND (NUMF > 23) AND (NUMC = 10)  ORDER BY DateProj DESC; |

Lise BRENAC & Denis PALLEZ Module 1104 (Introduction aux BDs) 1 / 3 IUT de Nice Côte d’Azur Département Informatique TP : L’instruction Select

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| k. | Afficher les numéros des films projetés au mois d’aout. | SELECT NUMF  FROM PROJECTION  WHERE (Month(DateProj) = 08) ; |

Exercice 2 Exercices sur machine

Dans la suite des TPs, nous allons utiliser une BD Jeu de rôle contenant des personnages de différentes classes possèdent des objets. Ils connaissent des sorts qui leur permettent de se défendre. Ces sorts peuvent être composés (ou non) de plusieurs ingrédients. Le schéma relationnel correspondant à cette base de données est le suivant :

CLASSE (IdClasse, Libelle, PointsParNiveau, ArmurePossible)

PERSONNAGE (IdPersonnage, IdClasse#, Nom, Race, Niveau, DateCreation)

OBJET (NumObjet, Libelle, Prix, Categorie)

POSSEDER (IdPersonnage#, NumObjet#, Quantite, DateAcquisition)

CONNAITRE (IdPersonnage#, IdSort#, DateConnu) SORT (IdSort, Nom, Niveau, Action)

COMPOSER (IdSort#, NumIngredient#, Quantite)

INGREDIENT (NumIngredient, Libelle, Prix)

De manière plus précise, la relation CLASSE donne les identifiants, libellé, points à acquérir pour monter de niveau et port d’armure possible des différentes classes de personnages. Une classe est identifiée par l’identifiant de classe IdClasse. La relation PERSONNAGE donne les identifiants nom, race, niveau, date de création des différents personnages, ainsi que l’identifiant de la classe à laquelle appartient le personnage. Un personnage est identifié par IdPersonnage. La relation OBJET donne les identifiants libellé, prix, catégorie des objets. Un objet est identifié par le numéro d’objet NumObjet. La relation POSSEDER précise quel personnage possède quel objet et en quelle quantité, ainsi que la date d’acquisition.

Requêtes sur une seule relation

1. Donner toutes les informations sur la relation Personnage.
2. Donner les différentes races de personnages.
3. Donner toutes les informations sur les sorts, triés par niveau décroissant.
4. Donner toutes les informations sur les sorts, triés par niveau décroissant. Si plusieurs sorts possèdent le même niveau alors ils devront être triés par ordre alphabétique.
5. Donner les numéros des objets possédés en renommant temporairement l’attribut numObjet en

Objet\_Possede.

1. Donner les identifiants des personnages pour lesquels le nom n’est pas connu. *Si vous n’avez aucun résultat, vérifiez manuellement que vous avez des enregistrements dans la table qui correspondent à vos conditions. Dans le cas contraire, ajouter des enregistrements qui vérifient les conditions de la requête pour vérifier que votre requête est correcte.* De ce cas précis, ajouter un nouvel enregistrement dans la table Personnage avec un nom non renseigné ! (faites‐le en SQL afin de vous entrainer).
2. Donner toutes les informations sur les personnages humains dont le niveau est inférieur ou égal à 10.
3. Donner toutes les informations sur les objets qui ne sont pas des armes.
4. Donner toutes les informations sur les armes et les armures.
5. Donner toutes les informations sur les objets qui ne sont ni des armes, ni des armures.
6. Donner le libellé, la catégorie et le prix majoré de 15% de tous les objets magiques (sans modifier les valeurs dans la table).

Lise BRENAC & Denis PALLEZ Module 1104 (Introduction aux BDs) 2 / 3 IUT de Nice Côte d’Azur Département Informatique TP : L’instruction Select

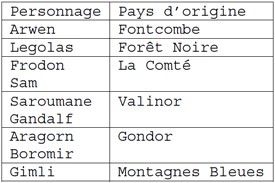
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Donner l’identifiant des sorts appris pendant l’été (de juin à septembre).
2. Donner le nom des sorts qui commencent par ‘Soins’. Rendez ensuite votre requête indépendante à la casse.
3. Donner le nom des sorts pour lesquels l’action contient ‘dégat’. En analysant les tuples de la relation, on constate qu’il devrait y avoir 2 enregistrements. Comment modifier la requête pour que les 2 tuples apparaissent résultats ? On souhaite créer des codes pour les personnages de la façon suivante : Pour le personnage ‘ARA’,’RO’, ‘Aragorn’, ‘Humain’, #01/10/2006#, 10 ; on souhaite créer le code ara\_RO\_aragorn\_H\_06. ara sera forcément en minuscule, RO forcément en majuscule, aragorn forcément en minuscule, H pour Humain en majuscule, 06 pour l’année de création après 2000. Créer la requête SQL qui affiche tous ces codes. Vous pourrez vous aider de l’aide en ligne disponible ici.



# Pour aller plus loin

MODIFICATION DU SCHÉMA

1. Dans la table Personnage, ajouter les attributs Sexe (un caractère qui désigne le sexe de l’individu (‘F’ pour Féminin ou ‘M’ pour Masculin) et Pays (une chaîne qui désigne le pays d’origine du personnage). Vous trouverez ci‐contre les différents pays des personnages.
2. Dans la table Connaître, en plus de la date de connaissance, on souhaite ajouter le prix payé par un personnage pour connaître un sort.

REQUÊTES DE MISE À JOUR

1. Majorer de 10% tous les prix des objets de la catégorie « Armes ».
2. Créer une nouvelle table Archive ayant la même structure que la table Posséder, puis ajouter dans la table Archive toutes les lignes de la table Posséder dont la date d’acquisition est antérieure au 31 décembre 2007.
3. Supprimer de la table Posséder toutes les lignes dont la date d’acquisition est antérieure au 31 décembre 2007.
4. Ajouter dans la table Posséder toutes les lignes de la table Archive.

# Sources pour ce TP

Lise BRENAC & Denis PALLEZ Module 1104 (Introduction aux BDs) 3 / 3