

Réseaux 1

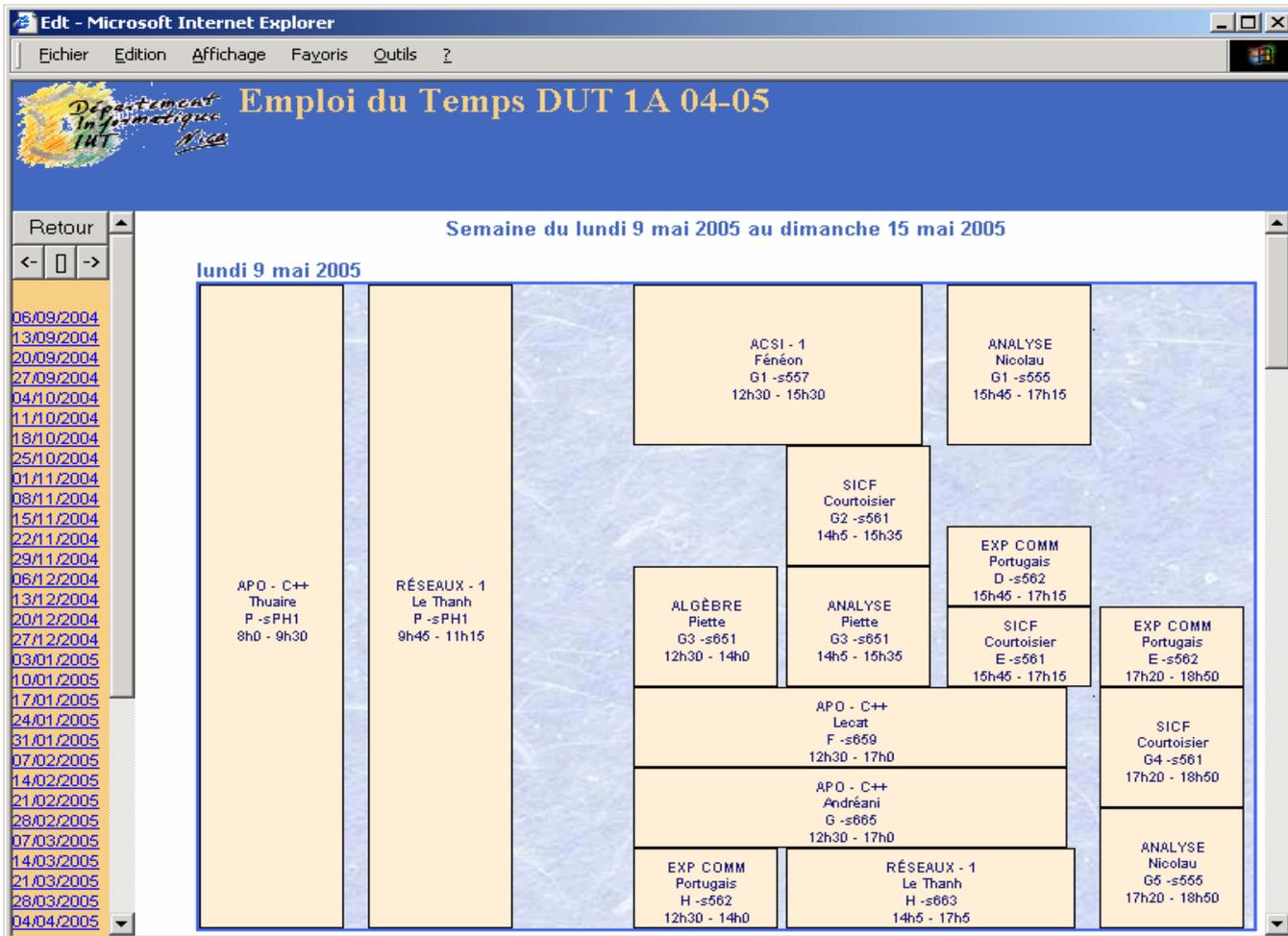
TP 13 – Accès XML à une base de données

Objectif : mettre en pratique XML, CSS et PHP sur un exemple concret

Nous considérons le problème d'affichage des emplois du temps en intranet.

Les informations sont organisées dans une base de données ORACLE (vous trouvez le schéma en Annexe).

On désire pouvoir afficher l'emploi du temps d'une semaine comme dans l'exemple ci-dessous.



Une page de garde permettra de sélectionner l'année universitaire, la formation et l'année d'étude.

La page principale comporte 3 cadres :

- le cadre supérieur affiche le code de la formation, l'année d'étude et l'année universitaire
- le cadre de gauche permet de sélectionner une semaine de l'année universitaire, et comporte trois boutons de navigation : semaine précédente, semaine courante ou semaine suivante
- le cadre central affiche l'emploi du temps de la semaine sélectionnée (au démarrage la semaine courante)

Nous allons procéder en 3 étapes :

1. Définir le balisage XML permettant de décrire le contenu de l'emploi du temps d'une semaine
2. Créer la feuille de style CSS permettant de visualiser le code XML dans IE
3. Créer les scripts PHP permettant d'extraire les données de la base ORACLE, et de les renvoyer en XML

1. Définir le balisage XML

- 1.1. Définissez le balisage XML permettant de décrire le contenu de l'emploi du temps d'une semaine.

2. Créer la feuille de style CSS pour visualiser le code XML

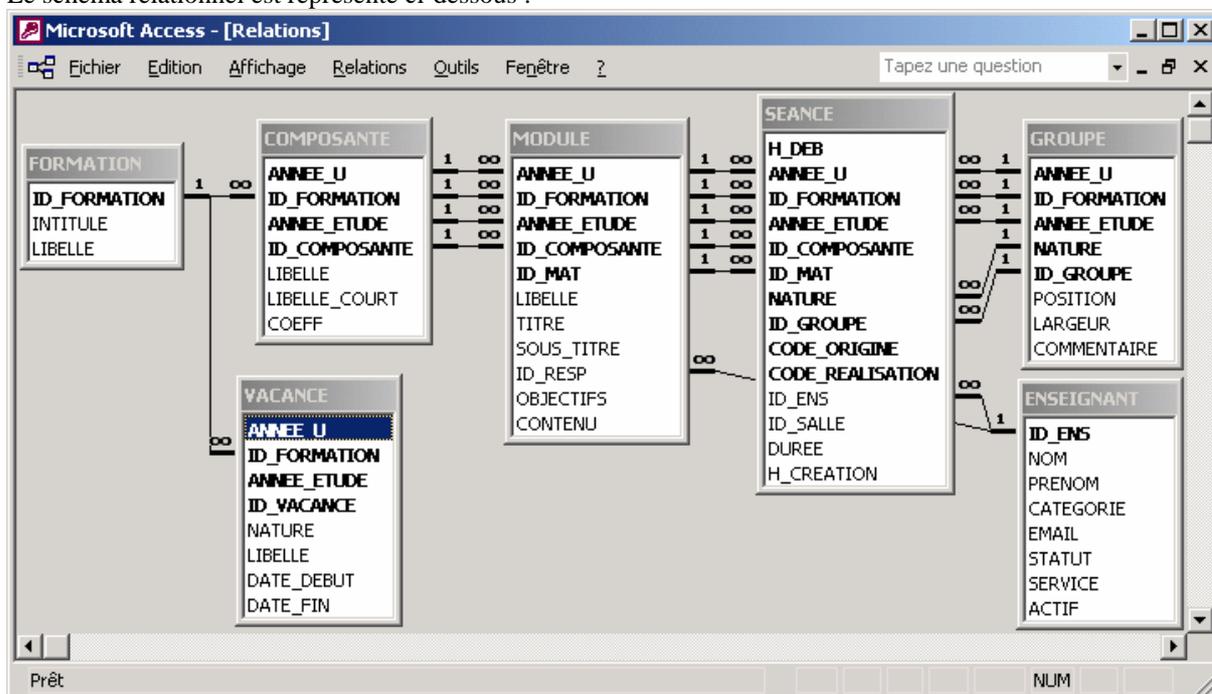
- 2.1. Créez la feuille de style CSS permettant de visualiser dans IE les données XML définie dans la question 1.1 ; tester à l'aide d'un exemple saisi à la main.

3. Créer les scripts PHP pour extraire les données de la base ORACLE, et renvoyer les données XML

- 3.1. Créez le script PHP permettant d'envoyer les données XML à partir d'ORACLE. Vous utilisez la vue VUE_EDT pour extraire l'emploi du temps d'une semaine.
- 3.2. Créez le script PHP permettant d'afficher dans le cadre de gauche les liens permettant de sélectionner une semaine dans l'année universitaire, et de créer les 3 boutons de navigation.
- 3.3. Créez le script PHP qui affiche dans le cadre supérieur le code de la formation, l'année d'étude et l'année universitaire.
- 3.4. Créez le script PHP qui affiche la page principale composée des trois cadres.

4. Annexe – Schéma ET (jeu d'essai sous le compte oracle ET)

Le schéma relationnel est représenté ci-dessous :



- La table FORMATION contient les intitulés et les libellés des différentes formations (DUT, LP ...)
 - le champ ID_FORMATION est le code identifiant une formation (clé primaire)

Les données des tables COMPOSANTE, MODULE, SEANCE, GROUPE et VACANCE sont relatives à une année universitaire, une formation, et une année d'étude.

- La table COMPOSANTE contient les libellés et les coefficients des composantes (UEs ou matière) d'enseignement.
 - le champ ID_COMPOSANTE est numérique sur deux chiffres
 - le premier chiffre représente l'UE
- la table MODULE contient les informations relatives aux modules d'enseignement
 - le champ ID_RESP fait référence à l'enseignant responsable du module
- la table SEANCE contient les séances de cours dans l'emploi du temps
 - le champ H_DEB donne la date et l'heure de la séance
 - le champ DUREE donne la durée de la séance en minutes
- La table GROUPE contient les groupes d'étudiants
 - Le champ NATURE fait référence à un type de cours (C, TD, TP ...)
 - Les champs POSITION et LARGEUR servent pour l'édition des tableaux d'emploi du temps, comme dans l'exemple ci-dessous (ou un groupe de TP fait 4 unités de largeur)

Groupes	G1		G2		G3		G4		G5							
Position	0		8		14		20		26							
Largeur	8		6		6		6		6							
Groupes	A	B	C	D	E	F	G	H								
Position	0		4		8		12		16		20		24		28	
Largeur	4		4		4		4		4		4		4			

- la table VACANCE contient les dates des semaines de vacances ou spéciales
 - le champ nature est égal à « V » pour les vacances, « X » pour les semaines spéciales
- la table ENSEIGNANT contient les informations relatives aux enseignants

Script SQL de création des tables

```
rem $ etblid.sql --construction du schema de base ET
DROP TABLE seance ;
DROP TABLE module ;
DROP TABLE composante ;
DROP TABLE vacance ;
DROP TABLE groupe ;
DROP TABLE formation ;
DROP TABLE enseignant ;

CREATE TABLE enseignant
(
  id_ens          VARCHAR2 (3)   PRIMARY KEY
  , nom           VARCHAR2 (20)
  , prenom        VARCHAR2 (14)
  , categorie     VARCHAR2 (1)
  , email         VARCHAR2 (60)
  , statut        VARCHAR2 (6)
  , service       NUMBER (3)
  , actif         NUMBER (1)
) ;

CREATE TABLE formation /* formations DUT, LP .. */
(
  id_formation   VARCHAR2 (4)   PRIMARY KEY
  , intitule     VARCHAR2 (60)
  , libelle      VARCHAR2 (10)
) ;

CREATE TABLE composante /* composantes d'enseignement UEs et matieres */
(
  annee_u        VARCHAR2 (4)
  , id_formation VARCHAR2 (4)   REFERENCES formation
  , annee_etude  NUMBER (1)    CHECK ( annee_etude IN (1, 2) )
  , id_composante NUMBER (6)
  , libelle      VARCHAR2 (50)
  , libelle_court VARCHAR2 (12)
  , coeff        NUMBER (6,2)
  , PRIMARY KEY (annee_u, id_formation, annee_etude, id_composante)
) ;
```

```

CREATE TABLE module          /* modules d'enseignement */
(
  annee_u          VARCHAR2 (4)
  id_formation     VARCHAR2 (4)
  annee_etude      NUMBER (1)      CHECK ( annee_etude IN (1, 2) )
  id_composante    NUMBER (6)
  id_mat           VARCHAR2 (8)
  libelle          VARCHAR2 (30)
  titre            VARCHAR2 (50)
  sous_titre       VARCHAR2 (30)
  id_resp          VARCHAR2 (3)    REFERENCES enseignant
  objectifs        VARCHAR2 (800)
  PRIMARY KEY (annee_u, id_formation, annee_etude, id_composante, id_mat)
  FOREIGN KEY (annee_u, id_formation, annee_etude, id_composante) REFERENCES composante
)
;

CREATE TABLE vacance        /* dates des vacances */
(
  annee_u          VARCHAR2 (4)
  id_formation     VARCHAR2 (4)    REFERENCES formation
  annee_etude      NUMBER (1)      CHECK ( annee_etude IN (1, 2) )
  id_vacance       VARCHAR2 (8)
  nature           VARCHAR2 (1)
  libelle          VARCHAR2 (20)
  date_debut       DATE
  date_fin         DATE
  PRIMARY KEY (annee_u, id_formation, annee_etude, id_vacance)
)
;

CREATE TABLE groupe        /* groupes d'etudiants */
(
  annee_u          VARCHAR2 (4)
  id_formation     VARCHAR2 (4)    REFERENCES formation
  annee_etude      NUMBER (1)      CHECK ( annee_etude IN (1, 2) )
  nature           VARCHAR2 (6)
  id_groupe        VARCHAR2 (2)
  position         NUMBER (3)
  largeur          NUMBER (3)
  commentaire      VARCHAR2 (15)
  PRIMARY KEY (annee_u, id_formation, annee_etude, nature, id_groupe)
)
;

CREATE TABLE seance        /* seances de cours (C, TD, TP) */
(
  h_deb            DATE
  annee_u          VARCHAR2 (4)
  id_formation     VARCHAR2 (4)
  annee_etude      NUMBER (1)      CHECK ( annee_etude IN (1, 2) )
  id_composante    NUMBER (6)
  id_mat           VARCHAR2 (8)
  nature           VARCHAR2 (6)
  id_groupe        VARCHAR2 (2)
  code_origine     NUMBER (1)
  code_realisation NUMBER (1)
  id_ens           VARCHAR2 (3)    REFERENCES enseignant
  id_salle         VARCHAR2 (3)    REFERENCES salle
  duree            NUMBER (3)
  h_creation       DATE
  PRIMARY KEY (h_deb, annee_u, id_formation, annee_etude, id_composante,
              id_mat, nature, id_groupe, code_origine, code_realisation)
  FOREIGN KEY (annee_u, id_formation, annee_etude, id_composante, id_mat)
              REFERENCES module
  FOREIGN KEY (annee_u, id_formation, annee_etude, nature, id_groupe) REFERENCES groupe
)
;

```

Vue VUE_EDT (en lecture pour tous les utilisateurs oracle)

```

CREATE VIEW vue_edt AS
SELECT
  H.annee_u, H.id_formation, H.annee_etude,
  H.h_deb, H.nature, M.libelle, E.nom, H.duree,
  H.id_groupe, H.id_salle, H.h_creation,
  H.code_realisation, G.position, G.largeur
FROM
  ET.seance H , ET.module M, ET.enseignant E, ET.groupe G
WHERE
  H.annee_u = M.annee_u
  AND H.id_formation = M.id_formation
  AND H.annee_etude = M.annee_etude
  AND H.id_composante = M.id_composante
  AND H.id_mat = M.id_mat
  AND H.id_ens = E.id_ens
  AND H.annee_u = G.annee_u
  AND H.id_formation = G.id_formation
  AND H.annee_etude = G.annee_etude
  AND H.nature = G.nature
  AND H.id_groupe = G.id_groupe ;

```