TP3 – Application Web

Enseignant : Mohammed Salim MELFLAH

Cégep Bois-de-Boulogne

Document d’analyse pour approbation du tp3

Travail d’équipe :

Etudiant 1 : Axel Djellout DA : 2138081

Etudiant 2 : Steven Vuong DA : 6177234

Objectif :

Le but du travail est de s’assurer de la bonne compréhension de la modélisation et de l’implantation d’une base de données, de l’utilisation d’un logiciel de contrôle des sources (git) et finalement la création de pages Web dynamiques qui iront chercher et mettre à jour de l’information dans la base de données. Ceci est un travail d’équipe et doit se faire avec un maximum de deux équipiers.

Consignes générales :

Vous êtes invités à concevoir un projet de site web dynamique de votre choix, tout en respectant les critères suivants : Le projet choisi devra comprendre un minimum de 4 tables et un maximum de 6, et devra comporter un minimum de 3 écrans HTML et un maximum de 5. Vous devrez utiliser des feuilles de style (.CSS) pour mettre en forme les pages HTML de votre site. Un minimum de 2 pages, qui elles devront contenir un/des programmes JavaScript qui iront chercher/mettre à jour de l’information dans la base de données (API REST)

Nature du projet

Ce projet est un site web de commerce électronique simple. La page affiche une liste de produits, chacun avec une image, une description et un prix. Chaque produit a également un bouton "Ajouter au panier", vraisemblablement pour que les utilisateurs puissent ajouter le produit à leur panier d'achat.

Le site web comprend une barre de navigation horizontale en haut de la page, avec un logo, une barre de recherche et une icône de panier. Il y a aussi une barre de navigation verticale avec des liens vers différentes catégories de produits, comme Manga, Jeux Vidéo, Figurines et Cartes. Le site vend une variété de produits reliés aux manga/anime.

Le code est écrit en HTML, qui est utilisé pour structurer le contenu de la page. L'apparence réelle de la page (couleurs, polices, mise en page, etc.) serait contrôlée par CSS, et tout comportement interactif (comme l'ajout d'un produit au panier) serait mis en œuvre avec JavaScript.

Modèle de données logique :

DROP TABLE IF EXISTS manga;

CREATE TABLE manga (

id\_manga INT AUTO\_INCREMENT,

titre VARCHAR(100) PRIMARY KEY,

auteur VARCHAR(100),

annee\_publication INT

stock INT

);

ALTER TABLE manga ADD CONSTRAINT manga\_pk PRIMARY KEY ( id\_manga );

INSERT INTO manga (id\_manga, titre, auteur, annee\_publication, stock) VALUES

('One Piece', 'Eiichiro Oda', 1997, 20),

('Naruto', 'Masashi Kishimoto', 1999, 20),

('Blue Lock', 'Muneyuki Kaneshiro', 2018, 20);

DROP TABLE IF EXISTS carte;

CREATE TABLE carte (

id\_carte INT AUTO\_INCREMENT,

nom VARCHAR(100),

anime VARCHAR(100) PRIMARY KEY,

edition VARCHAR(100),

rarete VARCHAR(100)

stock INT

);

ALTER TABLE carte ADD CONSTRAINT carte\_pk PRIMARY KEY ( id\_carte );

INSERT INTO carte (nom, anime, edition, rarete, stock) VALUES

('Trafalgar Law 047', 'One Piece', 'Bandai', 'Parallel', 20),

('Jiraiya Sage Mode', 'Naruto', 'Bandai', 'Super Rare', 20),

('Charizard', 'Pokemon', 'Wizards of the Coas', 'Rare', 20);

DROP TABLE IF EXISTS jeux\_video;

CREATE TABLE jeux\_video (

id\_jeux\_video INT AUTO\_INCREMENT,

titre VARCHAR(100),

anime VARCHAR(100) PRIMARY KEY,

developpeur VARCHAR(100),

plateforme VARCHAR(100)

stock INT

);

ALTER TABLE jeux\_video ADD CONSTRAINT jeux\_video\_pk PRIMARY KEY ( id\_jeux\_video );

INSERT INTO jeux\_video (titre, anime, developpeur, plateforme, stock) VALUES

('One Piece Odyssey', 'One Piece', 'ILCA, Inc.', 'PC', 20),

('Naruto Shippuden: Ultimate Ninja Storm 3', 'Naruto', 'CyberConnect2', 'PlayStation 3', 20),

('Pokemon FireRed', 'Pokemon', 'Game Freak', 'Game Boy Advance', 20);

DROP TABLE IF EXISTS figurine;

CREATE TABLE figurine (

id\_figurine INT AUTO\_INCREMENT,

nom VARCHAR(100),

anime VARCHAR(100) PRIMARY KEY,

prix DECIMAL(5,2),

fabricant VARCHAR(100)

stock INT

);

ALTER TABLE figurine ADD CONSTRAINT figurine\_pk PRIMARY KEY ( id\_figurine );

INSERT INTO figurine (nom, anime, prix, fabricant, stock) VALUES

('Grand Ship Collection 01 - Thousand Sunny', 'One Piece', 26.99, 'Bandai', 20),

('Precious G.E.M Uchiha Itachi Susanoo Ver.', 'Naruto', 899.99, 'MegaHouse', 20),

('Nagi Seishiro', 'Blue Lock', 79.99, 'Nendoroid', 20);

CREATE TABLE purchase (

id\_purchase INT AUTO\_INCREMENT,

id\_client INT,

cout DECIMAL(5,2),

purchase\_id\_manga INT,

purchase\_id\_carte INT,

purchase\_id\_jeux\_video INT,

purchase\_id\_figurine INT

);

INSERT INTO purchase (id\_client, cout, purchase\_id\_manga, purchase\_id\_carte, purchase\_id\_jeux\_video, purchase\_id\_figurine) VALUES

(1, 26.99, 1, NULL, NULL, NULL),

(2, 899.99, NULL, NULL, NULL, 2),

(3, 79.99, NULL, NULL, NULL, 3)

ALTER TABLE purchase ADD CONSTRAINT purchase\_pk PRIMARY KEY ( id\_purchase );

ALTER TABLE purchase

ADD CONSTRAINT purchase\_client\_fk FOREIGN KEY ( id\_client )

REFERENCES client ( id\_client );

ALTER TABLE purchase

ADD CONSTRAINT purchase\_manga\_fk FOREIGN KEY ( purchase\_id\_manga )

REFERENCES manga ( id\_manga );

ALTER TABLE purchase

ADD CONSTRAINT purchase\_carte\_fk FOREIGN KEY ( purchase\_id\_carte )

REFERENCES carte ( id\_carte );

ALTER TABLE purchase

ADD CONSTRAINT purchase\_jeux\_video\_fk FOREIGN KEY ( purchase\_id\_jeux\_video )

REFERENCES jeux\_video ( id\_jeux\_video );

ALTER TABLE purchase

ADD CONSTRAINT purchase\_figurine\_fk FOREIGN KEY ( purchase\_id\_figurine )

REFERENCES figurine ( id\_figurine );

INSERT INTO purchase (id\_client, cout, purchase\_id\_manga, purchase\_id\_carte, purchase\_id\_jeux\_video, purchase\_id\_figurine) VALUES

-- Ceci va creer un URI sous le URL qui pourra etre utilise pour y activer les tables en mode REST

BEGIN

ORDS.enable\_schema(

p\_enabled => TRUE,

p\_schema => 'RESTSCOTT',

p\_url\_mapping\_type => 'BASE\_PATH',

p\_url\_mapping\_pattern => 'hr2',

p\_auto\_rest\_auth => FALSE

);

COMMIT;

END;

/

-- Activation de la table manga pour acces REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE, -- Default { TRUE | FALSE }

p\_schema => 'RESTSCOTT',

p\_object => 'MANGA',

p\_object\_type => 'TABLE', -- Default { TABLE | VIEW }

p\_object\_alias => 'MANGA'

);

COMMIT;

END;

/

-- Activation de la table carte pour acces REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE, -- Default { TRUE | FALSE }

p\_schema => 'RESTSCOTT',

p\_object => 'CARTE',

p\_object\_type => 'TABLE', -- Default { TABLE | VIEW }

p\_object\_alias => 'CARTE'

);

COMMIT;

END;

/

-- Activation de la table jeux\_video pour acces REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE, -- Default { TRUE | FALSE }

p\_schema => 'RESTSCOTT',

p\_object => 'JEUX\_VIDEO',

p\_object\_type => 'TABLE', -- Default { TABLE | VIEW }

p\_object\_alias => 'JEUX\_VIDEO'

);

COMMIT;

END;

/

-- Activation de la figurine pour acces REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE, -- Default { TRUE | FALSE }

p\_schema => 'RESTSCOTT',

p\_object => 'FIGURINE',

p\_object\_type => 'TABLE', -- Default { TABLE | VIEW }

p\_object\_alias => 'FIGURINE'

);

COMMIT;

END;

/

-- Activation de la table purchase pour acces REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE, -- Default { TRUE | FALSE }

p\_schema => 'RESTSCOTT',

p\_object => 'PURCHASE',

p\_object\_type => 'TABLE', -- Default { TABLE | VIEW }

p\_object\_alias => 'PURCHASE'

);

COMMIT;

END;

/

-- Confirmation de l'activation du schema

SELECT \*

FROM user\_ords\_schemas;

-- Confirmation de l'activation des tables pour REST

SELECT \*

FROM user\_ords\_enabled\_objects;

Croquis écrans

Accueil

Une image contenant texte, capture d’écran, graphisme, dessin humoristique

Description générée automatiquement

Page N°2

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Page N°3

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Page N°4

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement