

# Travail d'intégration (420-KED, 420-KA5)

## Des circuits et des monuments :

---

### Objectifs

Ce travail vise principalement à vous faire expérimenter les aspects suivants:

- Réaliser un travail d'intégration
- Définir un prototype visuel.
- Définir le modèle de données adéquat.
- Écrire des requêtes SQL
- Développer l'application C#, ADO.NET
- Travailler en équipe. **Vous devez être en équipe de deux**

### Mise en contexte :

L'agence Tourism est une agence qui œuvre dans le domaine du tourisme. Une de ces tâches consiste à faire visiter à ces clients de monuments historiques en leur proposant des circuits

Un circuit à une ville début, une ville d'arrivée, un prix, un nom unique, la durée totale pour le visiter et le nombre maximum de clients pour le circuit. Un circuit est identifié par un numéro séquentiel.

Sur un circuit, nous avons des monuments à visiter.

Chaque monument a un numéro unique, un nom, une date de construction, une brève histoire du monument, une d'image qui n'est pas dans la BD, (dans la BD on stocke l'emplacement de l'image), un prix pour une visite guidée et un nombre d'étoiles. Les étoiles vont de 1 à 5 (5 étoiles, veut dire à ne pas rater, 4 il faut absolument le visiter etc.)

Un monument peut se trouver sur plusieurs circuits, et un circuit a plusieurs monuments. Il y a un ordre de visite pour chaque monument sur chacun des circuits.

Les villes ont un code, un nom, un nombre total d'habitants et une brève histoire.

Avant d'aller visiter un circuit, les clients doivent faire une réservation. Une réservation a un numéro unique qui est séquentiel, une date de réservation et une date limite à laquelle on peut annuler sans pénalité. Un client peut faire plus qu'une réservation. Une réservation concerne un client.

Un client peut réserver pour plusieurs circuits. Sur un circuit on peut avoir plusieurs clients.

## Le travail demandé :

Après avoir implémenté le modèle de la base de données, vous devez réaliser une application C# ADO.NET qui permet de faire les opérations suivantes :

1. Ajouter un circuit. Doit utiliser un OracleParameter.
2. Modifier un circuit : modifier le prix d'un circuit.
3. Supprimer un monument d'un circuit.
4. Ajouter un monument. Doit utiliser un OracleParameter.
5. Liste des circuits offerts par l'agence. La recherche doit se faire selon les critères suivants :
  - a. Tous les circuits, **cette liste doit être ordonnée par la cotation (nombre d'étoiles)**. (Afficher les infos suivantes : le nom du circuit, le nom de la ville de départ, le nom de la ville de d'arrivée, le prix du circuit)
  - b. La ville de départ.
  - c. Le prix : liste des circuits dont le coût est inférieur à un certain prix (que l'on doit saisir)
  - d. Contenant un monument donné.
6. Recherche de monuments selon la cotation (nombre d'étoiles). Vous devez afficher les circuits sur lesquels ils se trouvent et sa position dans le circuit (l'ordre dans lequel il sera visité)
7. Rechercher la liste des monuments sur un circuit. (Nom du monument, coût de la visite guidée, brève histoire, l'ordre dans lequel le monument sera visité, la photo du monument). Les indications concernant cet affichage seront fournies par le cours de KA5.
8. Quels est le meilleur circuit (d'abord selon le coût, puis selon le nombre de monuments de trois étoiles ou plus) à choisir pour visiter un monument en particulier.

## Contrainte concernant la base de données

- La Base de données est Oracle.
- Le prix d'un circuit doit être plus grand ou égal à 50 \$.
- Le nombre d'étoile doit être entre 1 et 5.

- Vous devez créer des vues pour les requêtes SELECT les plus complexes.
- Les numéros de circuits, de clients et de réservation doivent être fait par des séquences.
- Vous devez insérer au minimum (par SQLDeveloper)
  - o 15 monuments
  - o 5 circuits, sur chaque circuit, je dois trouver au moins cinq monuments distincts.
  - o En d'autres mots vous devez peupler la base de données de sorte que les recherches donnent des résultats.

## Remise et correction

Vous devez remettre :

1. Le modèle relationnel de la base de données : Le 29 novembre. La remise doit se faire en mains propres et au format papier. Tout retard dans la remise entraine la note zéro.
2. Le script contenant au format SQL
  - a. La création des tables
  - b. la définition des données (ALTER)
  - c. les insertions
  - d. la création des séquences
  - e. la création des vues.
3. L'application C# ADO.Net.

Pour les points de 2 et 3, une boite de remise sera ouverte à cet effet. La date limite pour la remise est le 17 décembre avant la fin des corrections. Aucune remise par courriel n'est acceptée. Tout retard implique la note zéro.

La correction de l'application aura lieu le lundi 17 décembre de 8h à 12h. Un ordre de passage vous sera envoyé par Colnet.

Grille de correction : La partie Bases de données compte pour 20% de la note finale

Critères d'évaluation	Pondération	Date de remise
Normalisation et modèle relationnel	15%	Le 29 novembre
Démonstration du fonctionnement	60%	Le 17 décembre
Efficacité de programmation et validation	25%	Le 17 décembre

**Important :**

1. Les détails concernant l'interface graphique seront fournis par mon collègue Nicolas Chourot.
2. La grille de correction pour la partie 420-KA5 sera fournie par mon collègue Nicolas Chourot.