

## **Projet Bases de données et programmation web : agence immobilière**

Vous êtes chargé de mettre en place une base de données pour une agence immobilière. La base de données est destinée à être utilisée aussi bien par le personnel de l'agence immobilières que par les clients désirant louer ou acheter un bien immobilier.

Ci-dessous, les informations qui doivent être accessibles à partir de la base de données ainsi que les applications qui devront être réalisées.

L'agence immobilière possède un certain nombre de biens immobiliers, proposés soit à la location soit à la vente.

Pour les maisons ou appartements mises en vente, les propriétaires des biens immobiliers décident du prix de vente initiale et indique à l'agence une marge de manœuvre dans la négociation du prix. Les frais d'agence sont négociés entre l'agence immobilière et le propriétaire, ils correspondent entre 3 % et 10 % du prix du bien mise en vente.

Pour les biens immobiliers en location, le propriétaire doit indiquer un loyer mensuel ainsi qu'un montant estimé des charges. Les frais d'agence sont d'au maximum un mois de loyer sans charge. Les clients souhaitent disposer de différentes informations sur les biens immobiliers afin d'effectuer leur choix : le prix, la surface habitable, le nombre de pièce et des images du biens. Pour les pièces, il doit être possible de connaître le nombre de chambres, de salles de bains, de cuisines, la présence d'un garage et d'une cave. Pour les maisons, la taille du terrain doit être accessible. Pour les appartements, l'étage et la présence d'un ascenseur doit être indiqué.

Les clients font leur recherche en indiquant un prix minimum et maximum à l'achat ou à la location pour les biens, la taille minimale ou maximale de la surface, la nombre de pièces souhaités et la région/quartier où se situe le bien.

La base de données contiendra aussi les informations sur le personnel de l'agence. Les membres du personnel peuvent être directeur, secrétaire, personnel administratif ou agent immobilier.

Les clients sont caractérisés par leur nom, leur prénom, leur date de naissance, leur adresse, leur numéro de téléphone, leur adresse e-mail de contact, la date de leur inscription à l'agence, leur login et mot de passe pour accéder à leur compte en ligne. Un client peut aussi enregistrer ses préférences de recherche.

Un client peut déclarer être intéressé par un appartement et réserver un créneau afin d'effectuer une visite avec un agent immobilier.

Chaque vente ou location est enregistré dans la base. Une vente ou une location associe un biens immobilier, un locataire et un agent immobilier (celui qui a réalisé la vente).

Un client peut aussi ajouter des adresses de positions dont il souhaite connaître la distance par rapport aux biens immobiliers examinés. Il pourra par exemple préciser la position de son lieu de travail et pourra ainsi connaître la durée estimé de trajet entre son lieu de travail et le bien examiné. Les informations de distance entre positions seront obtenues par requête sur une API de cartographie.

Des tables d'archivage seront définies : les biens immobiliers louées ou vendus ne seront pas dans la même tables que les biens immobiliers en cours de vente ou en cours de location.

Remarque : les mots de passe (pour se connecter au site) ne sont pas stockés tels quels dans la base de donnée. C'est le résultat d'une fonction de hachage appliquée au mot de passe qui sera stockée dans la base.

1. Etablir le modèle entité-association de la base de données de l'agence immobilière en respectant au maximum les besoins indiqués dans l'énoncé.
  2. Préciser les contraintes d'intégrité sur le modèle (contraintes que les données devront respecter à tout instant).
  3. Etablir le modèle logique relationnel de la base.
- Rendu du projet : un rapport devra être remis contenant le schéma E/A, la liste des contraintes d'intégrités ainsi que le schéma relationnel.

Rendu du projet : les scripts SQL de création des tables, de suppression des tables et d'insertion de données de tests devront être remis.

Ecrire les requêtes permettant d'obtenir les résultats suivants

1. Les propriétaires ayant plus de 5 biens en vente dont la surface habitable est supérieur à 130 m<sup>2</sup>.
2. Pour chaque ville, le nombre de biens immobilier en location, le nombre de biens immobilier en vente et le nombre de clients cherchant un bien dans cette ville.
3. Pour chaque appartement vendu, la différence entre le bénéfice maximum potentiel et le bénéfice réelle de la vente.
4. Pour chaque région, les clients ayant effectué plus de visites d'appartement au cours du dernier mois.
5. Pour chaque client, le bien immobilier le moins cher correspondant à ses critères de recherche.

- 
1. Définir une fonction qui retourne le chiffre d'affaire de vente d'un agent immobilier. Le chiffre d'affaire correspond à la somme du bénéfice de chaque bien immobilier vendu par l'agent.
  2. Définir une procédure qui diminue le prix de vente de 2 % de tous les biens immobiliers proposés à la vente depuis plus d'un mois et qui n'ont pas trouvé acquéreur, jusqu'à la valeur minimale permise (déterminée par la marge de manœuvre dans la négociation des prix défini par le propriétaire). Cette procédure est exécutée au début de chaque mois.
  3. Définir une procédure d'archivage des biens immobilier loués/vendus. La procédure supprime tous les biens vendus de la table courante et ajoute ces biens dans la table d'archivage des biens vendus.  
La table d'archivage est identique à la table des biens immobiliers mis en vente, à l'exception de deux colonnes ajoutées, une contenant la date de vente et l'autre le bénéfice de la vente.
  4. Ecrire une fonction PL/SQL qui retourne une chaîne de caractère correspondant au code HTML d'une liste contenant un top des n agents immobiliers ayant effectués le plus de ventes au cours du trimestre.
  5. Définir une fonction qui renvoie les appartements situés dans une zone géographique. La fonction prend en paramètre les coordonnées d'un rectangle (défini par son coin inférieur gauche et supérieur droit).

---

Réaliser les différentes contraintes d'intégrité spécifiées dans le sujet du projet. Vous pouvez définir des contraintes statiques ou dynamiques. Vous pouvez aussi, si cela est nécessaire, réaliser des

procédures permettant de réaliser des opérations valides. Par exemple, assurez-vous que tous bien supprimé des ventes ou des locations est archivé.

Réaliser aussi les contraintes d'intégrités suivantes :

1. Modifier ou refuser les ajouts des clients dont les paramètres ne sont pas valides :
  - Le nom d'un client est entièrement en majuscule.
  - Le prénom d'un client commence par une majuscule, les autres lettres sont en minuscules.
  - Le login ne contient que des lettre et des chiffres. Le login doit commencer par une lettre.
  - L'email est valide (il contient le caractère @ suivi, plus loin, du caractère '.').
2. Deux biens immobilier d'un même quartier ne peuvent être situés à plus de 15 km l'un de l'autre.
3. Les visites d'appartement peuvent être organisées du lundi au samedi, entre 8h et 20h. Un client ne peut effectuer deux visites en même temps de deux biens situés sur deux sites distants. De même pour un agent.  
Il est par contre possible que jusqu'à trois personnes visitent simultanément un même bien.
4. Définir une vue permettant d'afficher la rentabilité en pourcentage des biens vendus :  
$$((\text{prix de vente réel})/(\text{prix de vente initiale}))*100$$
  
Il devra être possible de modifier le prix de vente d'un appartement par l'intermédiaire de cette vue.
5. Définir une table qui contiendra pour chaque agent son nombre de ventes effectuées ainsi que le total des bénéfices de ses ventes pour chaque année. Définir les déclencheurs permettant de s'assurer que cette table soit à tout instant à jour.

---

Assurez-vous que les procédures précédemment définies pour le projet permettent d'obtenir un résultat correct même si les données qu'elles manipulent sont modifiées par une autre transaction au cours de l'exécution de la procédure.

---

On décide d'enregistrer les projets à destination des étudiants dans une base de données.

Une interface HTML permettra d'ajouter, d'afficher ou de supprimer les projets.

La table projet est définie de la façon suivante :

```
projet(  
idProjet INTEGER,  
auteur VARCHAR2(64),  
texte VARCHAR2(512),  
url VARCHAR2(256)  
)
```

Écrire le formulaire HTML et le script PHP associé.

Astuce : la balise html <textarea> permet de définir une zone de texte à plusieurs lignes.