

UNIVERSITÉ PAUL SABATIER

TOULOUSE III

Bureau des Concours ITRF
118, Route de Narbonne
31062 TOULOUSE CEDEX 04

Concours EXTERNE – BAP E

Corps : Assistant Ingénieur
Spécialité : Développeur d'Applications

Session 2005

Durée : 3 h 00 - Coefficient : 4

Le sujet comporte 8 pages numérotées de 1/8 à 8/8.

Assurez-vous que cet exemplaire est complet.

S'il est incomplet, demandez un autre exemplaire au surveillant de salle.

Il vous est rappelé que votre identité ne doit figurer que sur la première page de la copie d'examen. Toute mention d'identité portée sur toute autre partie de la copie que vous remettrez en fin d'épreuve mènera à l'annulation de votre épreuve.

L'usage de la calculatrice n'est pas autorisé. L'usage du téléphone portable est interdit.
Tout document et autre matériel électronique sont interdits.

Ne pas écrire au crayon à papier sur la copie.

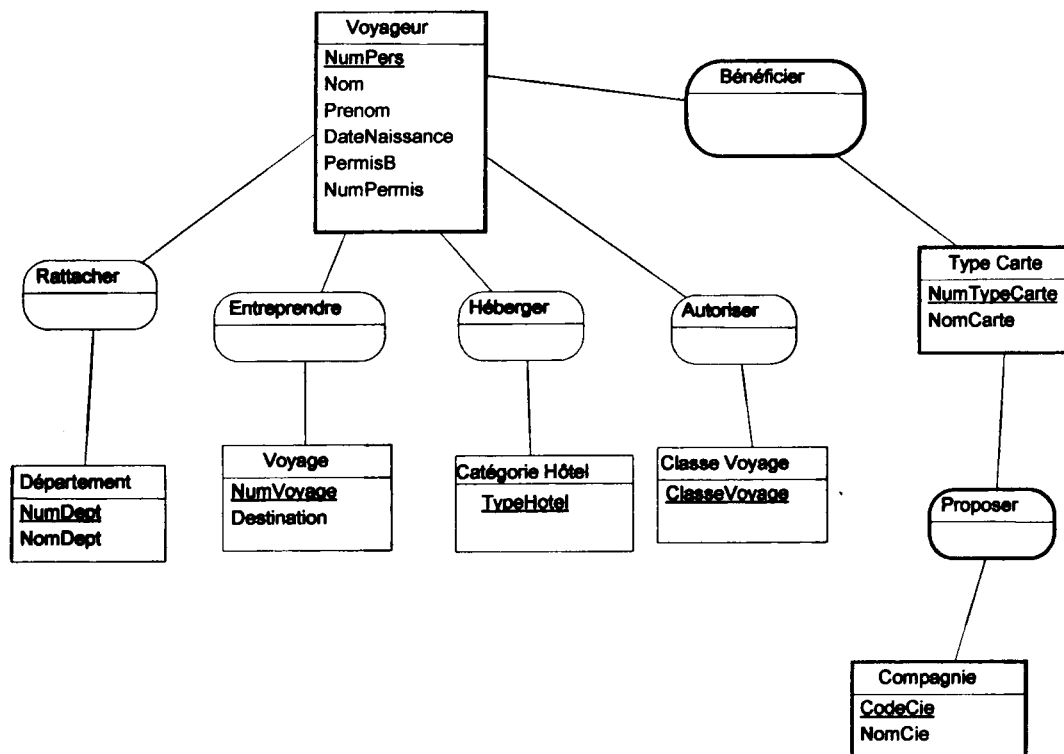
Ne pas répondre sur le sujet mais sur la copie d'examen.

PARTIE 1 : BASES DE DONNEES

Le personnel du Centre de Ressources Informatique (CRI) est amené à voyager fréquemment dans le monde entier. Actuellement, la gestion de ces voyages est réalisée à l'aide d'un tableur. Face à la croissance du nombre de voyages effectués, le responsable du service veut faire migrer cette application dans un environnement de type SGBD relationnel.

Une première analyse a conduit au schéma entité-association ci-joint :

Schéma conceptuel partiel



Règles de gestion :

- Le CRI est divisé en 5 départements : Développement, Gestion, Réseau, Web et Direction.
- Chaque employé est rattaché à un seul département et ne peut voyager que dans l'une des deux classes proposées par les compagnies aériennes (économique ou affaire). De plus, il n'est autorisé à séjourner que dans certaines catégories d'hôtel (2*, 3*, 4*).
- Un voyageur peut posséder plusieurs cartes de différentes compagnies.
- Un voyage ne concerne qu'une seule personne.
- Une compagnie commercialise un ou plusieurs types de carte.
- Un type de carte peut être proposé par plusieurs compagnies avec un prix de vente propre à la compagnie (PrixVenteCarte).
- Un voyageur peut bénéficier de plusieurs cartes de différentes compagnies. Sur chacune de ses cartes figurent un n° de carte (NumCarte) ainsi qu'une date de fin de validité (DateFin)

TRAVAIL À FAIRE

Reproduire le schéma conceptuel sur la copie d'examen.

Question 1.1 Rattacher au MCD les propriétés NumCarte, DateFin et PrixVenteCarte

Question 1.2 Trouver les cardinalités manquantes à partir des règles de gestion.

Question 1.3 Écrire le schéma relationnel correspondant aux entités et aux associations suivantes : VOYAGEUR, BENEFICIER, TYPE CARTE, PROPOSER, COMPAGNIE.

Pour gérer ses voyages, Le CRI fait appel à des prestataires.

Chaque voyage donne lieu à différentes prestations : transport, hébergement, location de véhicule, ...

L'ensemble des prestations relatives à un même voyage est géré par un seul prestataire à qui l'on règle la totalité des frais. Ces informations sont cependant gérées par le CRI pour des raisons statistiques.

Une ancienne base de données permettait de gérer les factures envoyées par les compagnies aériennes pour les vols effectués par ses employés.

Le schéma relationnel correspondant est le suivant :

COMPAGNIE(noCie, nomCie)

FACTURE(noFact, dateFact, montant, règlement)

EMPLOYE(noEmpl, nomEmpl, preEmpl, dateNaisEmpl, adrEmpl, cpEmpl, villeEmpl, noCateg)

TRANSPORT(noTransp, lieuDép, paysDep, lieuArr, paysArr, heureDep, heureArr, noFact, noCie)

VOYAGER(noTransp, noEmpl)

CATEGORIE (noCateg, libCateg)

Exemple d'enregistrement de la table EMPLOYE :

<u>noEmpl</u>	<u>nomEmpl</u>	<u>preEmpl</u>	<u>dateNaisEmpl</u>	<u>adrEmpl</u>	<u>cpEmpl</u>	<u>villeEmpl</u>	<u>noCateg</u>
56	O'NEILL	Derek	1963-01-25	10 rue de la plage	64000	PAU	19

TRAVAIL À FAIRE

En utilisant le langage SQL, à partir du modèle relationnel ci-dessus réaliser les requêtes suivantes :

- Question 1.4** Créer la table EMPLOYE. On suppose que la table CATEGORIE est déjà créée. On utilisera les types de données SQL fournis en annexe. La déclaration des clés primaire et étrangère est demandée.
- Question 1.5** Insérer dans la table TRANSPORT, le vol n° 1253A de la compagnie 130 partant de Paris à 8 h 53 et arrivant à Madrid en Espagne à 10 h 48. Le numéro de facture n'est pas renseigné pour l'instant.
- Question 1.6** Établir la liste des noms des compagnies qui ont envoyé au moins une facture pour des transports à destination de la Corée et du Japon.
- Question 1.7** Donner le nombre de factures impayées à la compagnie ORIENTAIR (on considère qu'une facture est impayée lorsque le règlement est égal à 0).
- Question 1.8** Attribuer la permission aux utilisateurs "Brenda" et "Nancy" de mettre à jour la colonne règlement de la table FACTURE.
- Question 1.9** Brenda a changé de fonction, elle a en charge la saisie des compagnies et des employés mais ne s'occupe plus des règlements, lui attribuer les droits en conséquence.
- Question 1.10** Afin de faire des tests, l'application est dupliquée sur une autre machine sous un autre nom tous les jours à 23h. Lister et expliciter les actions à faire. (export, import, etc.)

Annexe : Types de données SQL

CHAR(n)	Chaîne de caractères de longueur fixe (n compris entre 1 et 16 383)
VARCHAR(n)	Chaîne de caractères de longueur variable (n compris entre 1 et 16 383)
SMALLINT	Entier signé codé sur 2 octets : entre - 32 768 et + 32 767
INTEGER	Entier signé codé sur 4 octets : entre - 2^{31} et + $2^{31} - 1$
DECIMAL(n, p)	Décimal comportant n chiffres dont p après la virgule
NUMERIC(n)	Nombre relatif comportant n chiffres
FLOAT[(n)]	Si n est inférieur à 25, nombre en virgule flottante sur 32 bits sinon nombre en virgule flottante sur 64 bits (par défaut n = 53)
DATE	AAAA-MM-JJ
TIME	HH:MM:SS
TIMESTAMP	AAAA-MM-JJ:HH:MM:SS

PARTIE 2 - Programmation

2.1 : Que fait le programme suivant ?

```
Variables Nb, i en Entier
Variable Flag en Booleen
Tableau T() en Entier
Debut
Ecrire "Entrez le nombre de valeurs : "
Lire Nb
Redim T(Nb - 1)
Pour i ← 0 à Nb - 1
  Ecrire "Entrez le nombre n° ", i + 1
  Lire T(i)
FinPour
Flag ← Vrai
Pour i ← 1 à Nb - 1
  Si T(i) <> T(i - 1) + 1 Alors
    Flag ← Faux
  FinSi
FinPour
Si Flag Alors
  Ecrire "Les nombres sont ??? "
Sinon
  Ecrire "Les nombres ne sont pas ???"
FinSi
Fin
```

2.2 : Proposez une amélioration de l'algorithme précédent ? Justifiez.

2.3 : Ecrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une promotion. Le nombre d'étudiants de la promotion sera également saisi. Le programme, une fois la saisie terminée, renvoie le nombre d'étudiants ayant une note supérieure à la moyenne de la classe.

2.4 : Est-il possible de transférer le contenu d'un fichier type CSV dans une base de données ? Expliquez brièvement la méthode que vous choisiriez.

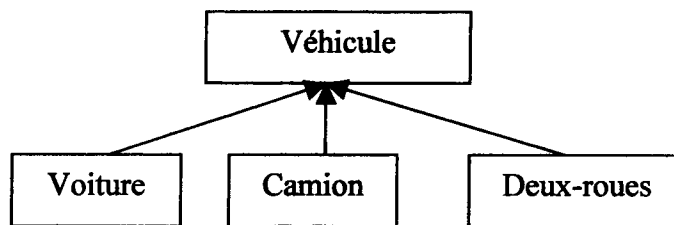
2.5 : Que représente un schéma XML ?

PARTIE 3 - Programmation Orientée Objet (POO)

3.1: Qu'est-ce que le concept d'encapsulation des données ?

3.2: Dans le monde de la POO, il existe trois types de polymorphisme, citez-les et donnez un exemple d'utilisation de chacun d'entre eux.

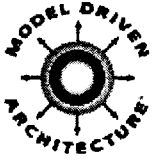
3.3 : Quels sont les relations représentées dans le schéma ci-dessous :



PARTIE 4 : Anglais technique

4.1 : Lisez l'article suivant extrait du site de l'OMG (Object Management Group) relatif à l'architecture MDA (Model Driven Architecture). Suite à cette lecture, résumez en une dizaine de lignes les spécificités et intérêts de cette architecture logicielle.

**Model Driven Architecture®
The Architecture of Choice for a Changing World®**



"The Architecture of Choice for a Changing World®"

Computing infrastructures are expanding their reach in every dimension. New platforms and applications must interoperate with legacy systems. Virtual enterprises span multiple companies. The Internet is imposing new integration challenges as it extends into every corner of every organization. New implementation platforms are continually coming down the road, each claiming to be "the next big thing."

Those who architect computer systems, whether for banks or battleships, face daunting technology choices. To protect their investments and maximize flexibility, they buy hardware that implements open interconnection standards like Ethernet and USB, and software that uses open interface standards like CORBA®. It's the only sensible course in today's rapidly changing, multi-vendor computing environment.

But as computers and networks become faster and cheaper, even interconnection standards must evolve. New technologies constantly appear for new application niches. One need look no further than the recent rise of XML to see how quickly this can happen. How can organizations ensure that their mission-critical information systems are rooted in standards that will adapt to new hardware capabilities and software platforms?

Now, the Object Management Group™ (OMG™) addresses this reality with MDA®, the Model Driven Architecture®. MDA supports evolving standards in application domains as diverse as enterprise resource planning, air traffic control and human genome research; standards that are tailored to the needs of these diverse organizations, yet need to survive changes in technology and the proliferation of different kinds of middleware. The OMG Model Driven Architecture™ addresses the complete life cycle of designing, deploying, integrating, and managing applications as well as data using open standards. MDA-based standards enable organizations to integrate whatever they already have in place with whatever they build today...and whatever they build tomorrow.

Building on a Solid Foundation

The product of OMG's proven, open standards adoption process, MDA represents a significant though evolutionary-step forward. It is built on the solid foundation of well-established OMG standards, including: Unified Modeling Language™ (UML®), the ubiquitous modeling notation used and supported by every major company in the software industry; XML Metadata Interchange (XMI™), the standard for storing and exchanging models using XML; and CORBA™, the most popular open middleware standard.

The OMG MDA® separates the fundamental logic behind a specification from the specifics of the particular middleware that implements it. This allows rapid development and delivery of new interoperability specifications that use new deployment technologies but are based on proven, tested business models. Organizations can use MDA to meet the integration challenges posed by new platforms, while preserving their investments in existing business logic based on existing platforms.

MDA addresses the challenges of today's highly networked, constantly changing systems environment, providing an architecture that assures:

- Portability, increasing application re-use and reducing the cost and complexity of application development and management, now and into the future.
- Cross-platform Interoperability, using rigorous methods to guarantee that standards based on multiple implementation technologies all implement identical business functions.
- Platform Independence, greatly reducing the time, cost and complexity associated with re-targeting applications for different platforms-including those yet to be introduced.
- Domain Specificity, through Domain-specific models that enable rapid implementation of new, industry-specific applications over diverse platforms.
- Productivity, by allowing developers, designers and system administrators to use languages and concepts they are comfortable with, while allowing seamless communication and integration across the teams.

Bottom-line Benefits

The benefits of MDA are significant-to business leaders and developers alike:

- Reduced cost throughout the application life-cycle
- Reduced development time for new applications
- Improved application quality
- Increased return on technology investments
- Rapid inclusion of emerging technology benefits into their existing systems

MDA provides a solid framework that frees system infrastructures to evolve in response to a never-ending parade of platforms, while preserving and leveraging existing technology investments. It enables system integration strategies that are better, faster and cheaper.