

TD complet

Préalable au cours sur PL/SQL

Michel DUBOIS

Nous nous intéresserons à la gestion d'un aéro-club qui est une association à but non lucratif, régie par la loi de 1901. Il offre aux adhérents une flotte importante d'avions récents. Outre les équipements normaux (une piste, une tour de contrôle et un service de maintenance), il dispose de deux hangars. Il donne la possibilité aux adhérents de voler avec des avions de types différents. Selon le cas, l'adhérent sera sous la direction d'un instructeur durant un vol.

On considère le diagramme de classe fourni en annexe 1. Ce diagramme de classe repose sur la règle de gestion suivante : un instructeur est qualifié pour au moins un type d'avions.

En plus du diagramme de classe, on nous indique les contraintes suivantes :

A : Un avion ne peut participer à un vol ayant une date antérieure à sa date de mise en service.

B : Seul un adhérent autonome pour le type d'avion peut voler sans instructeur.

C : Un vol avec un adhérent non autonome a toujours un instructeur. Cet instructeur est unique pour ce vol.

D : Seul un instructeur qualifié pour le type d'avion peut diriger un vol concernant ce même type d'avions.

E : Un instructeur qualifié pour un type d'avion est forcément en tant qu'adhérent, autonome pour ce type d'avion

F : Un adhérent qui n'est pas instructeur doit être suivi par un unique instructeur qui sera son interlocuteur privilégié.

G : Aucun salaire n'est inférieur à la valeur du SMIG.

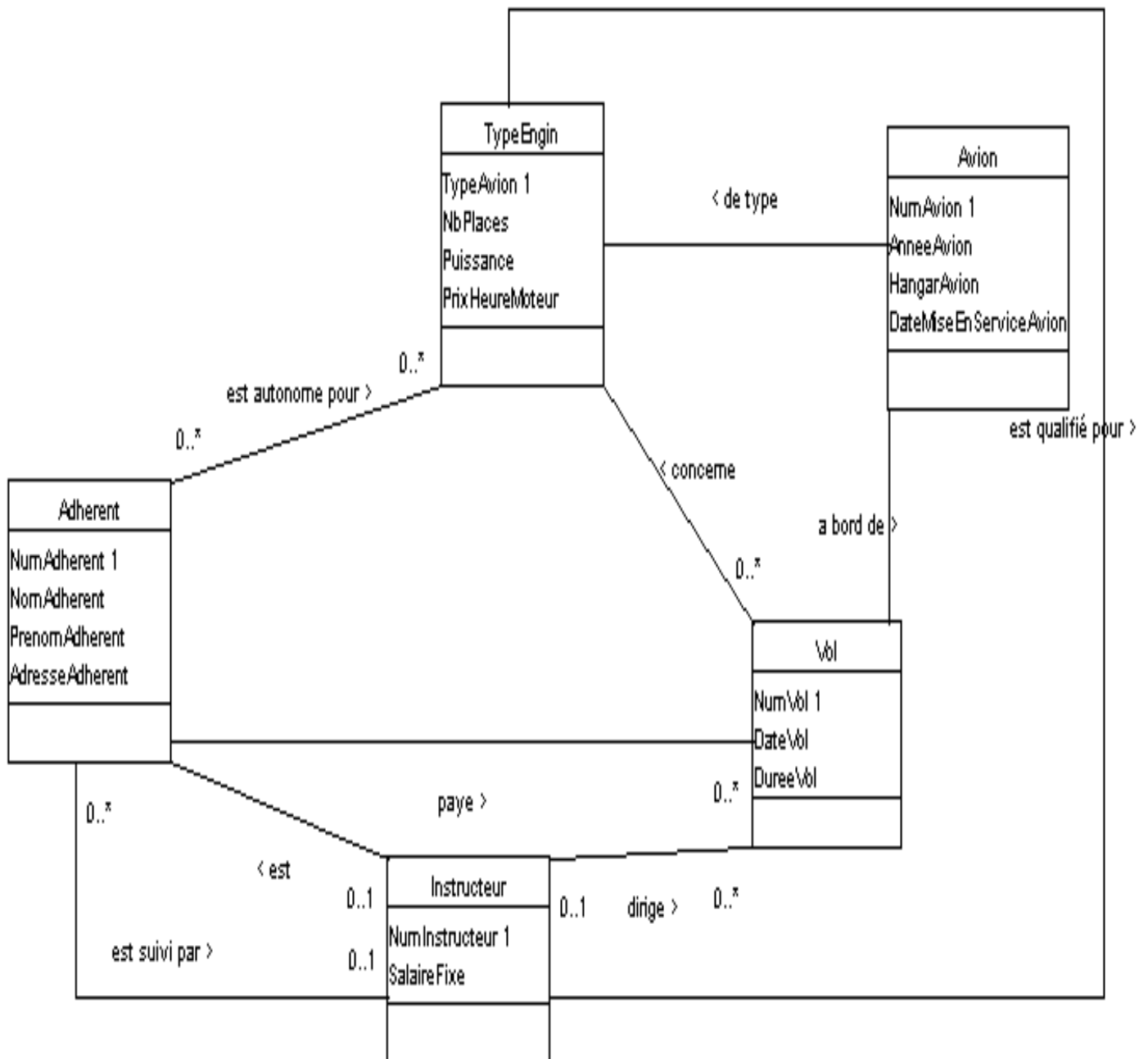
H : Le hangar d'un avion ne peut être que « H1 » ou « H2 », casse indifférente.

Un vol concerne un et un seul type d'avion. Il se fait à bord de un et un seul avion. Il est piloté par un et un seul adhérent. Il est éventuellement dirigé par un seul instructeur. Un avion à un et un seul type d'engin.

Ce TD a pour objectif de proposer une méthodologie de conception de table dans un contexte transactionnel. D'autres méthodes sont possibles et peuvent être proposées dans le cours de conception du Système d'Informations. Elles reposent toutes sur la théorie de la normalisation. Mais ce TD doit vous aider à comprendre le sujet du projet Business Objects qui vous permettra de mettre en oeuvre toutes les parties de la démarche de conception d'un système décisionnel. De plus, le travail fait lors de ce TD manipulant uniquement des concepts sur les Systèmes de Gestion des Bases de Données sera utilisé lors du TP n°1 sur la création d'un univers Business Objects.

A Compléter le diagramme de classe UML ci-joint : précisez les multiplicités (ou cardinalités)

Annexe 1 : Diagramme UML à compléter.



B Traduction de « Vol Concerne TypeEngin »

Soit le modèle relationnel (type MS Access, ou Oracle par BO) à la page suivante. Qu'est devenu l'association « concerne » ? Pourquoi un tel traitement ? Quel type d'objet Oracle (table exclue) permet d'accéder par une requête simple aux informations de « concerne » sans qu'il y ait de redondance dans le schéma physique.

C Graphe des dépendances fonctionnelles de couverture minimale

A partir du tableau des dépendances fonctionnelles élémentaires, directes et canoniques entre attributs non calculés et non paramètres (annexe), proposez le graphe de couverture minimale, le graphe des dépendances fonctionnelles inter-relation et le schéma relationnel.

D Prise en compte des attributs calculés

Comment prendre en compte les attributs calculés du dictionnaire des données dans une bases de données Oracle, MS Access, MySQL ?

E Prise en compte des paramètres

Comment prendre en compte le paramètre SMIG du dictionnaire des données ?

F Optimisation du schéma physique Oracle

On désire créer une vue IndVol permettant d'accéder à des attributs calculés. Voici les règles de calcul :

Supplément de salaire pour le directeur du vol s'il existe pour ce vol= 300* Durée du vol.

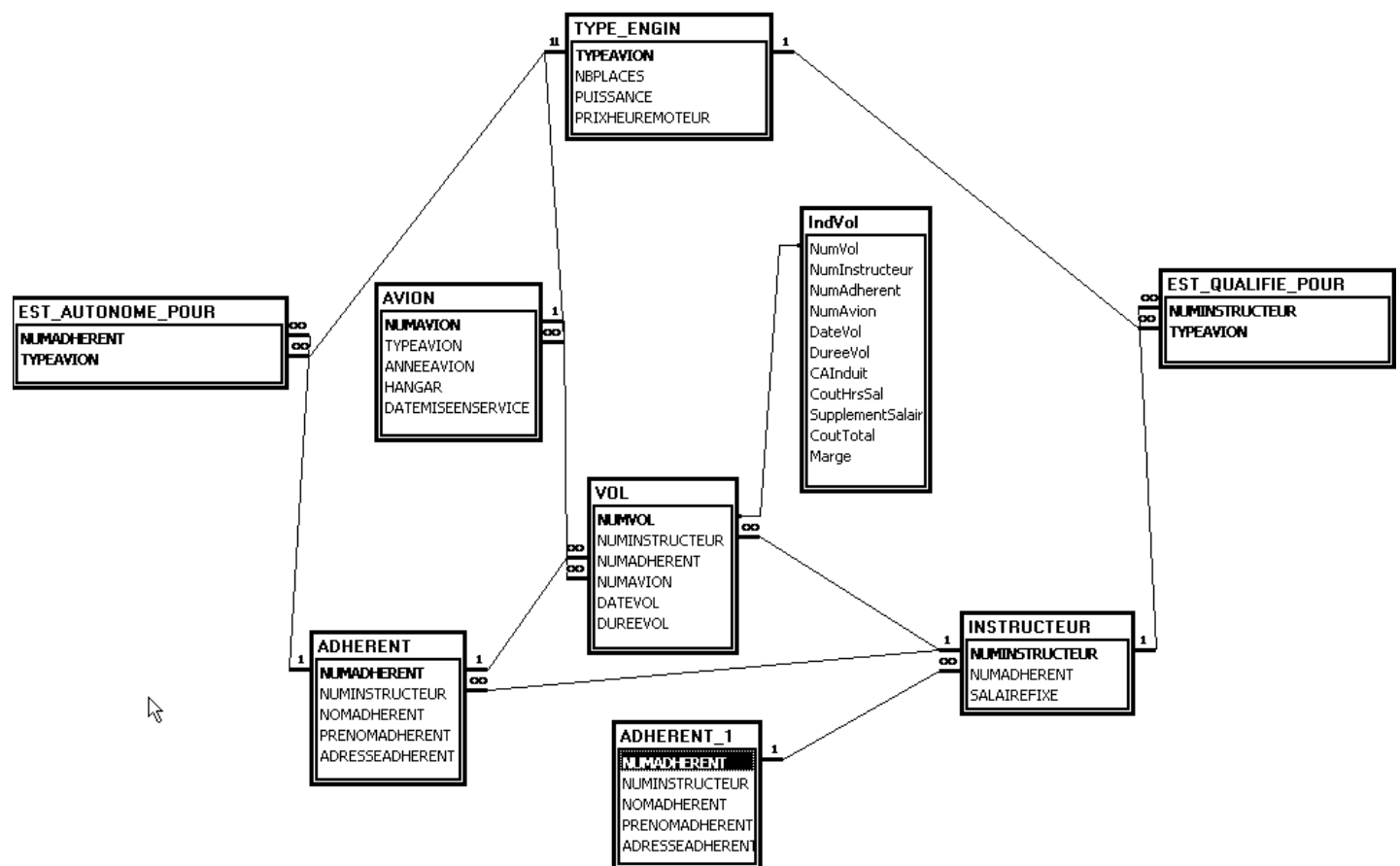
Chiffre d'affaires induit par le vol= Durée du vol*(Prix de l'heure moteur + x) (x=350 si le vol est dirigé, x=0 sinon).

Coûts hors salaire du vol=80% de (Durée du vol*(Prix de l'heure moteur))

Coût total=Coûts hors salaire du vol+ Supplément de salaire pour le directeur du vol.

Marge=CA induit – Coût total.

Annexe 2 : Modèle Physique d'ACCESS.



Complétez l'ordre de création de la vue :

```

DROP VIEW IndVol;
CREATE VIEW IndVol(NumVol, NumInstructeur, NumAdherent, NumAvion, DateVol, DureeVol,
CAInduit, CoutHrsSal, SupplementSalaire, CoutTotal, Marge)AS
SELECT VOL.NumVol, VOL.NumInstructeur, VOL.NumAdherent, VOL.NumAvion, VOL.DateVol,
VOL.DureeVol,
DureeVol*PrixHeureMoteur,
DureeVol*.8*PrixHeureMoteur,
0,
.8*DureeVol*PrixHeureMoteur,
.2*DureeVol*PrixHeureMoteur
FROM VOL, AVION, TYPE_ENGIN
WHERE VOL.NumAvion=AVION.NumAvion
AND TYPE_ENGIN.TypeAvion=AVION.TypeAVION
AND NumInstructeur IS NULL
UNION
SELECT
VOL.NumVol, VOL.NumInstructeur, VOL.NumAdherent, VOL.NumAvion, VOL.DateVol, VOL.DureeVol,
...
...
...
...
.2*DureeVol*PrixHeureMoteur+50*DureeVol
FROM VOL, AVION, TYPE_ENGIN
WHERE VOL.NumAvion=AVION.NumAvion
AND TYPE_ENGIN.TypeAvion=AVION.TypeAVION
AND NumInstructeur IS NOT NULL
;

```

G Problèmes de mises à jour des tables ADHERENT et INSTRUCTEUR

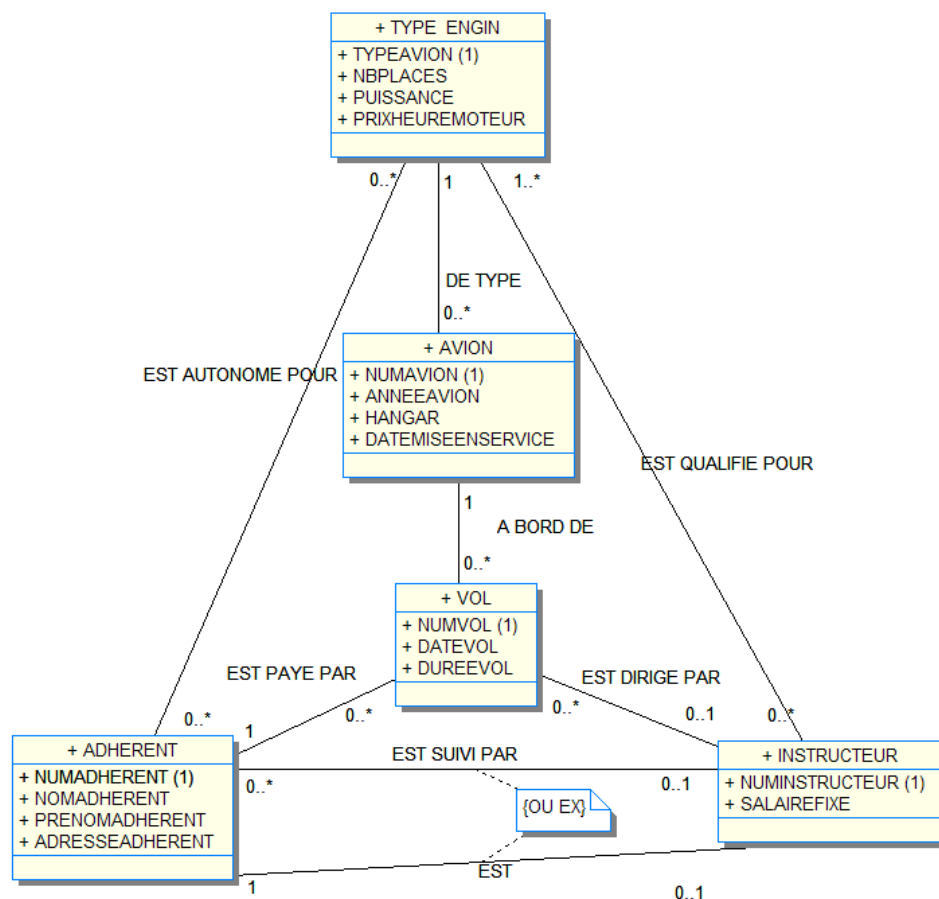
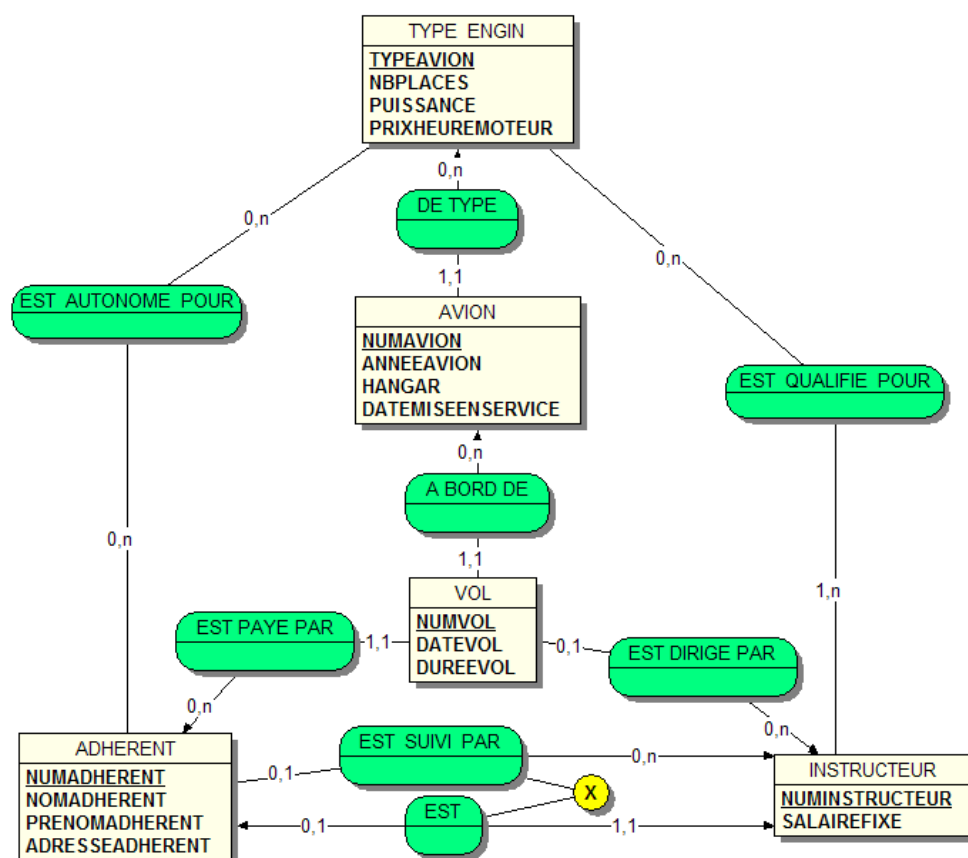
Les tables ADHERENT et INSTRUCTEUR se référencent mutuellement dans les deux schémas physiques. Ceci peut poser des problèmes lors de l'insertion de tuples dans les deux tables. Décrivez les. Donnez une solution permise par Oracle. Ces problèmes de mise à jour concernent ils Business Objects ?

H Scripts SQL de vérification statique des contraintes

On vous demande de proposer les requêtes SQL qui renvoient les anomalies par rapport aux contraintes A, B, C, D, E, F, G et H. La requête ne renverra que l'identifiant de l'objet concerné. Il vous est aussi demandé de vérifier statiquement le contrainte du « OU Exclusif » et de totalité (INSTRUCTEUR [NUMINSTRUCTEUR] = EST_QUALIFIE_POUR [NUMINSTRUCTEUR]).

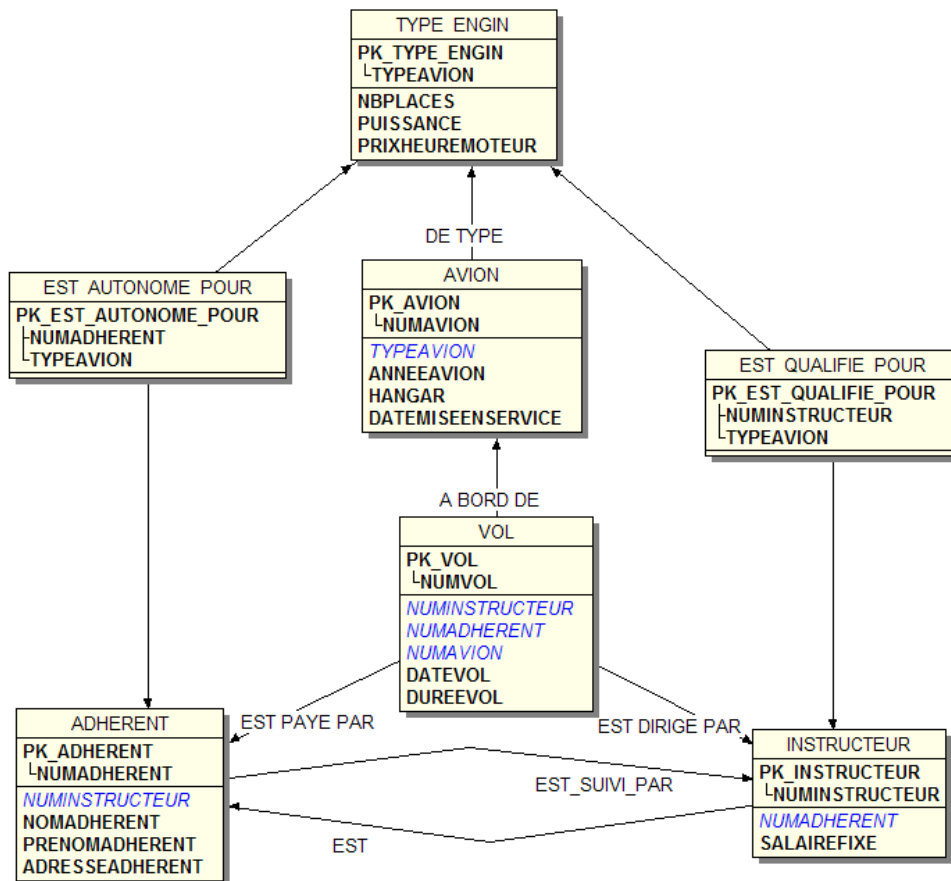
Annexes fournies

- Annexe 1 : Le diagramme de classes partiel.
- Annexe 2 : Représentation du Modèle Physique de MS ACCESS.
- Annexe 3 : Le diagramme de classes complet.
- Annexe 4 : Le Modèle Conceptuel des Données.
- Annexe 5 : Le dictionnaire des données complet.
- Annexe 6 : Le Modèle Logique des Données.
- Annexe 7 : Définition de la vue INDVOL.
- Annexe 8 : Représentation du Modèle Physique d'Oracle.
- Annexe 9 : Tableau des dépendances élémentaires et directes.
- Annexe 10 : Exemple de contraintes exprimées en SQL.

Annexe 3 : Diagramme de classes UML.**Annexe 4 : Modèle Conceptuel des Données.**

Annexe 5 : Dictionnaire des données complet du domaine.

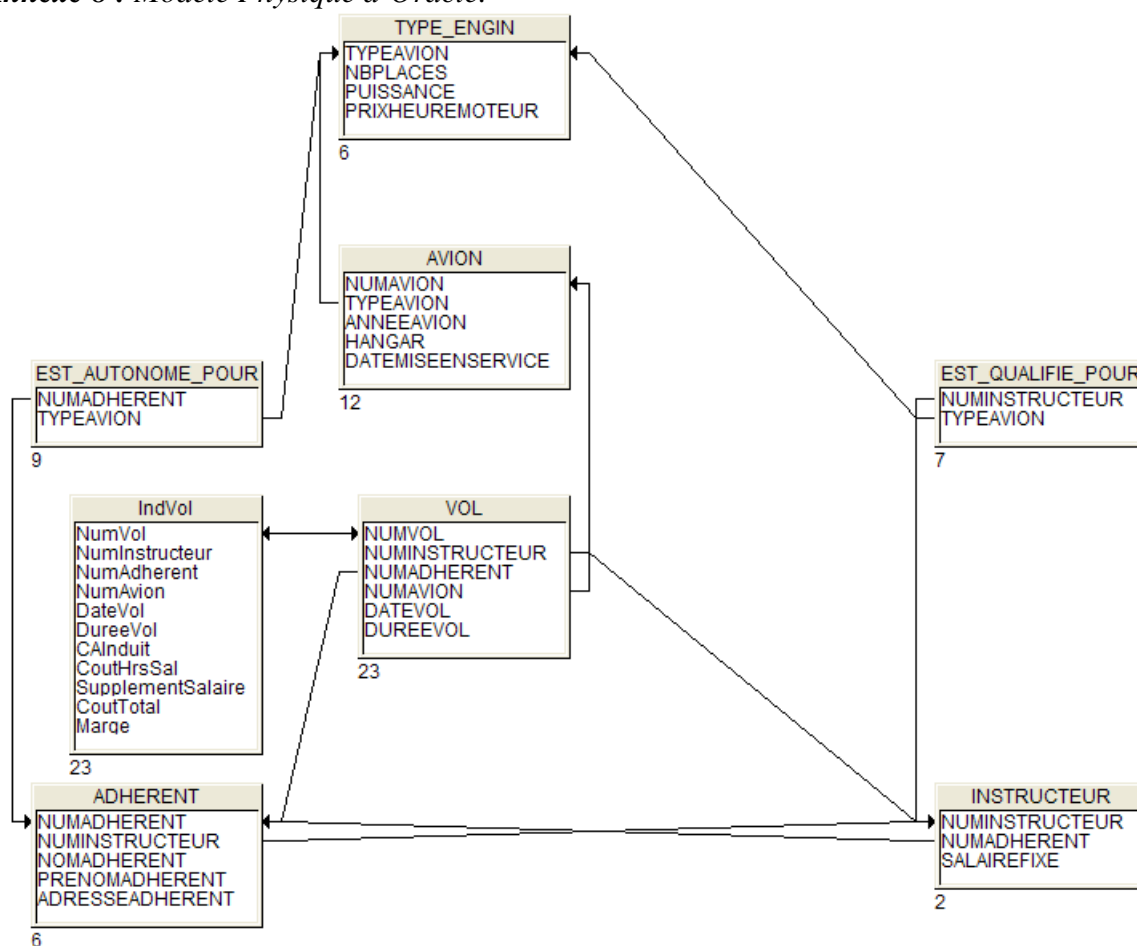
Code	Libellé	Nature	Catégorie	Domaine	Commentaire
NUMADHERENT	Numéro de l'adhérent	Incrémentée	Stable	Numérique Obligatoire.	
NOMADHERENT	Nom de l'adhérent	Saisie	Evolutive	Texte(32) Obligatoire.	
PRENOMADHERENT	Prénom de l'adhérent	Saisie	Stable	Texte(32) Obligatoire.	
ADRESSEADHERENT	Adresse de l'adhérent	Saisie	Evolutive	Texte(200) Obligatoire.	
NUMINSTRUCTEUR	Numéro de l'instructeur	Incrémentée	Stable	Numérique Obligatoire.	
SALAIREFIXE	Salaire fixe de l'instructeur	Saisie	Evolutive	Monétaire Obligatoire.	
TYPEAVION	Type d'avion	Saisie	Stable	Texte (32) Obligatoire	
NBPLACES	Nombre de place de l'avion	Saisie	Stable	Numérique Obligatoire	
PUISSANCE	Puissance du moteur	Saisie	Stable	Numérique Obligatoire	
PRIXHEUREMOTEUR	Prix de l'heure du moteur	Saisie	Evolutive	Monétaire Obligatoire	
NUMAVION	Numéro d'immatriculation	Saisie	Stable	Texte(32) Obligatoire	
ANNEEAVION	Année de fabrication	Saisie	Stable	Numérique Obligatoire	
HANGAR	Hangar	Saisie	Stable	Texte (2) Obligatoire. dans « H1, H2 », casse indifférente.	
DATEMISEENSERVICE	Date de mise en service	Saisie	Stable	Date Obligatoire.	
NUMVOL	Numéro du vol	Incrementée	Stable	Numérique Obligatoire	
DATEVOL	Date du vol	Saisie	Stable	DATE Obligatoire Valeur par défaut : date du jour.	
DUREEVOL	Durée du vol	Saisie	Stable	Numérique Obligatoire.	
CAInduit	Chiffre d'affaires induit par le vol	Calculée	Stable	Monétaire Obligatoire.	=DureeVol*PrixHeureMoteur+(350 si directeur)
CoutHrsSal	Coûts hors salaires	Calculée	Stable	Monétaire Obligatoire.	=DureeVol*.8*PrixHeureMoteur
SupplementSalaire	Prime du directeur de vol	Calculée	Stable	Monétaire Obligatoire.	= (300*DureeVol si directeur)
CoutTotal	Coûts induits par le vol	Calculée	Stable	Monétaire Obligatoire.	= CoutHrsSal+ SupplementSalaire
Marge	Marge induite par le vol	Calculée	Stable	Numérique Obligatoire.	=(CAInduit- CoutTotal)/ CAInduit
IdSMIG	Identifiant de la table SMIG	Saisie	Stable	Egal à 1 Obligatoire, UQ	Pas une clé primaire (Pas besoin de jointure naturelle !)
Valsmig	SMIG actuel	Paramètre	Evolutive	Monétaire Non obligatoire	Devra éventuellement être stocké dans une table avec tuple unique pour faciliter la mise à jour de ce paramètre

Annexe 6 : Modèle Logique des Données.**Annexe 7 : Définition de la vue SQL.**

```

DROP VIEW IndVol;
CREATE VIEW IndVol AS
SELECT VOL.NumVol, VOL.NumInstructeur, VOL.NumAdherent, VOL.NumAvion, VOL.DateVol,
VOL.DureeVol, DureeVol*PrixHeureMoteur AS CAInduit, DureeVol*.8*PrixHeureMoteur AS
CoutHrsSal, 0 AS SupplementSalaire, .8*DureeVol*PrixHeureMoteur AS
CoutTotal, .2*DureeVol*PrixHeureMoteur AS Marge
FROM VOL, AVION, TYPE_ENGIN
WHERE VOL.NumAvion=AVION.NumAvion
AND TYPE_ENGIN.TypeAvion=AVION.TypeAVION
AND NumInstructeur IS NULL
UNION
SELECT
VOL.NumVol, VOL.NumInstructeur, VOL.NumAdherent, VOL.NumAvion, VOL.DateVol, VOL.DureeVol, Duree
Vol*(PrixHeureMoteur+350) AS CAInduit, DureeVol*.8*PrixHeureMoteur AS CoutHrsSal,
300*DureeVol AS SupplementSalaire, .8*DureeVol*PrixHeureMoteur+300*DureeVol AS
CoutTotal, .2*DureeVol*PrixHeureMoteur+50*DureeVol AS Marge
FROM VOL, AVION, TYPE_ENGIN
WHERE VOL.NumAvion=AVION.NumAvion
AND TYPE_ENGIN.TypeAvion=AVION.TypeAVION
AND NumInstructeur IS NOT NULL
;

```

Annexe 8 : Modèle Physique d'Oracle.**Annexe 9 : Tableau des dépendances fonctionnelles directes et élémentaires entre attributs non calculés et non paramètres : recensement complet.**

N°	Dépendances Fonctionnelles élémentaires et directes mais pas canoniques		
1	NUMADHERENT	→	NOMADHERENT, PRENOMADHERENT, ADRESSEADHERENT, NUMINSTRUCTEUR (Df faible, instructeur suiveur), NUMINSTRUCTEUR (Df faible, est peut être).
2	NUMINSTRUCTEUR	→	NUMADHERENT (est à coup sûr), SALAIREFIXE.
3	TYPEAVION	→	NBPLACES, PUISSANCE, PRIXHEUREMOTEUR.
4	NUMAVION	→	ANNEEAVID, HANGAR, DATEMISEENSEERVICE.
5	NUMVOL	→	DATEVOL, DUREEVOL, NUMADHERENT, NUMINSTRUCTEUR (Df faible), NUMAVION.

Annexe 10 : Contraintes exprimées en SQL.

/*

Contrainte A :**Les vols ayant une mauvaise date**

*/

```
SELECT VOL.NumVol
FROM VOL,AVION
WHERE VOL.NumAvion=AVION.NumAvion
AND DateVol<DateMiseEnService
;
```

/*

Contrainte B :**Les vols non dirigés et pourtant non autonomes !**

*/

```
SELECT VOL.NumVol
FROM VOL,AVION
WHERE VOL.NumAvion=AVION.NumAvion
AND (NumAdherent,TypeAvion) NOT IN (
SELECT NumAdherent,TypeAvion
FROM EST_AUTONOME_POUR
)
AND NumInstructeur IS NULL
;
```

/*

Contrainte D :**Les vols dirigés par un instructeur non qualifié**

*/

```
SELECT VOL.NumVol
FROM VOL,AVION
WHERE VOL.NumAvion=AVION.NumAvion
AND (NumInstructeur,TypeAvion) NOT IN (
SELECT NumInstructeur,TypeAvion
FROM EST_QUALIFIE_POUR
)
;
```

/*

Contrainte E :**Les instructeurs ayant des qualifications non incluses dans leur autonomie**

*/

```
SELECT INSTRUCTEUR.NumInstructeur
FROM EST_QUALIFIE_POUR,INSTRUCTEUR,ADHERENT
WHERE ADHERENT.NumInstructeur=INSTRUCTEUR.NumInstructeur
AND EST_QUALIFIE_POUR.NumInstructeur=INSTRUCTEUR.NumInstructeur
AND (ADHERENT.NumAdherent,TypeAvion) NOT IN (
SELECT NumAdherent,TypeAvion
FROM EST_AUTONOME_POUR
)
;
```

/*

Contrainte H :**Les avions dont le hangar d'un avion n'est ni « H1 » ni « H2 », casse indifférente.**

*/

```
SELECT NumAvion  
FROM AVION  
WHERE UPPER(Hangar) NOT IN ('H1','H2')  
;
```

/*

Les adhérents qui violent le ou-ex

*/

```
SELECT ADHERENT.NumAdherent  
FROM ADHERENT,INSTRUCTEUR  
WHERE ADHERENT.NumInstructeur=INSTRUCTEUR.NumInstructeur  
AND ADHERENT.NumAdherent IN (  
    SELECT NumAdherent  
    FROM INSTRUCTEUR  
)  
UNION  
SELECT ADHERENT.NumAdherent  
FROM ADHERENT  
WHERE ADHERENT.NumInstructeur IS NULL  
AND ADHERENT.NumAdherent NOT IN (  
    SELECT NumAdherent  
    FROM INSTRUCTEUR  
)  
;
```