



Principes Relationnels et Concepts Oracle

Objectifs

A la fin de ce chapitre, vous saurez :

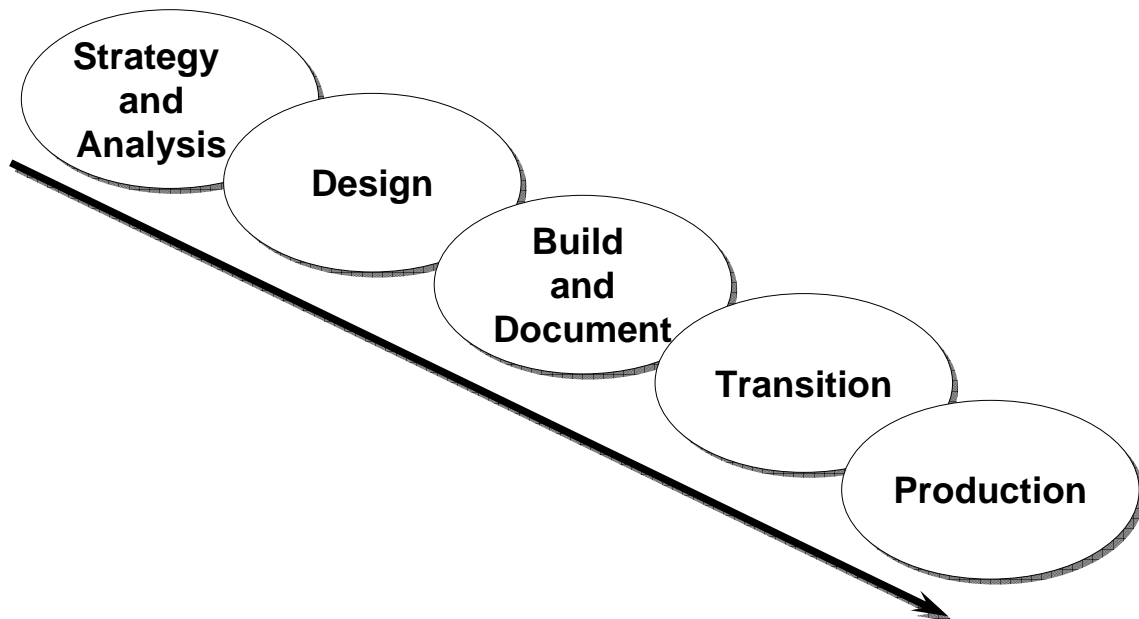
- **Décrire les phases du cycle de vie d'un système**
- **Décrire les aspects théoriques d'une base de données relationnelle**
- **Décrire l'implémentation Oracle des SGBDR et SGBDRO**
- **Décrire l'utilisation de SQL dans les produits Oracle**

I-2

Objectifs

Au cours de ce chapitre, vous allez étudier les principes de base des systèmes de gestion de bases de données relationnelles (SGBDR) et des systèmes de gestion de bases de données relationnelles objet (SGBDRO).

Cycle de Vie d'un Système



I-3

Cycle de Vie d'un Système

Il est possible d'utiliser la méthode du cycle de vie d'un système pour développer une nouvelle base de donnée. Ce cycle est constitué de différentes étapes allant de la stratégie à la production. Cette approche systématique permet de traduire des besoins d'informations de gestion en une base de données opérationnelle.

Strategy and Analysis

- Etude et analyse des besoins. Consultation des utilisateurs et responsables en vue d'identifier les besoins en informations. Intégration des divers rapports d'entreprise et comptes-rendus de mission, ainsi que des éventuelles spécifications futures du système.
- Création de plusieurs modèles du système. Transformation des spécifications écrites en une représentation graphique des besoins et règles d'information de gestion. Révision et confirmation de certains éléments du modèle avec les analystes et les experts.

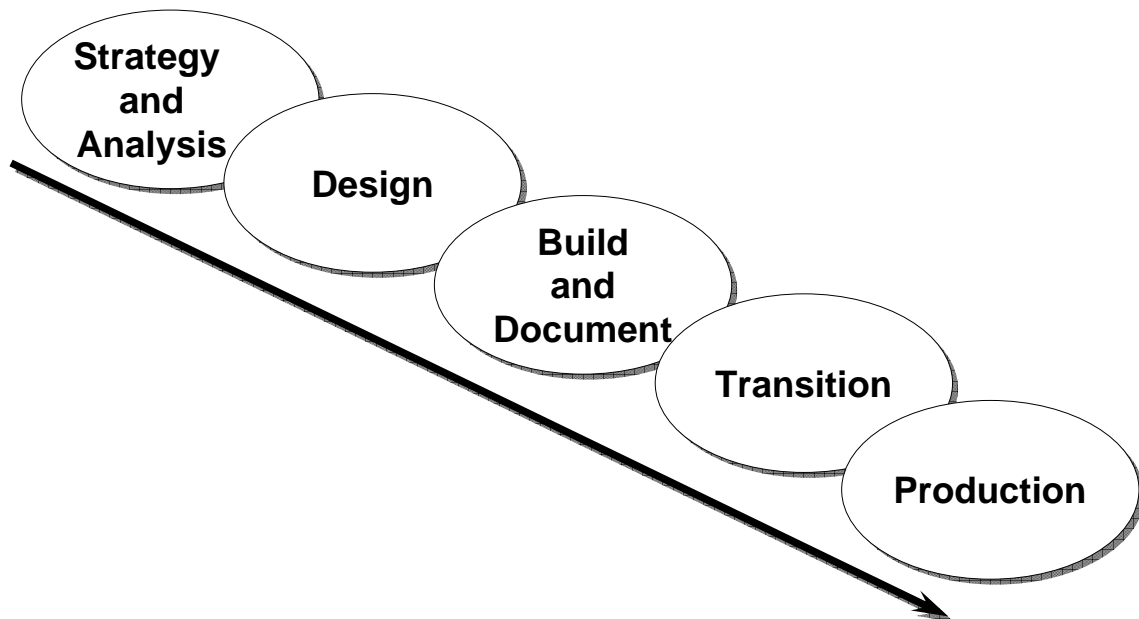
Design

Définition de la base de données à partir du schéma conceptuel des données développé au cours de la phase précédente.

Build and Document

- Génération des commandes de création des tables et des objets de la base de données.
- Développement de la documentation utilisateur, des textes d'aide et des manuels destinés à faciliter l'utilisation et le maniement du système.

Cycle de Vie d'un Système



I-4

Transition

Passage en production impliquant les tests d'acceptation utilisateur, la conversion des données existantes et les opérations parallèles. Ajout des modifications nécessaires.

Formation des utilisateurs finaux.

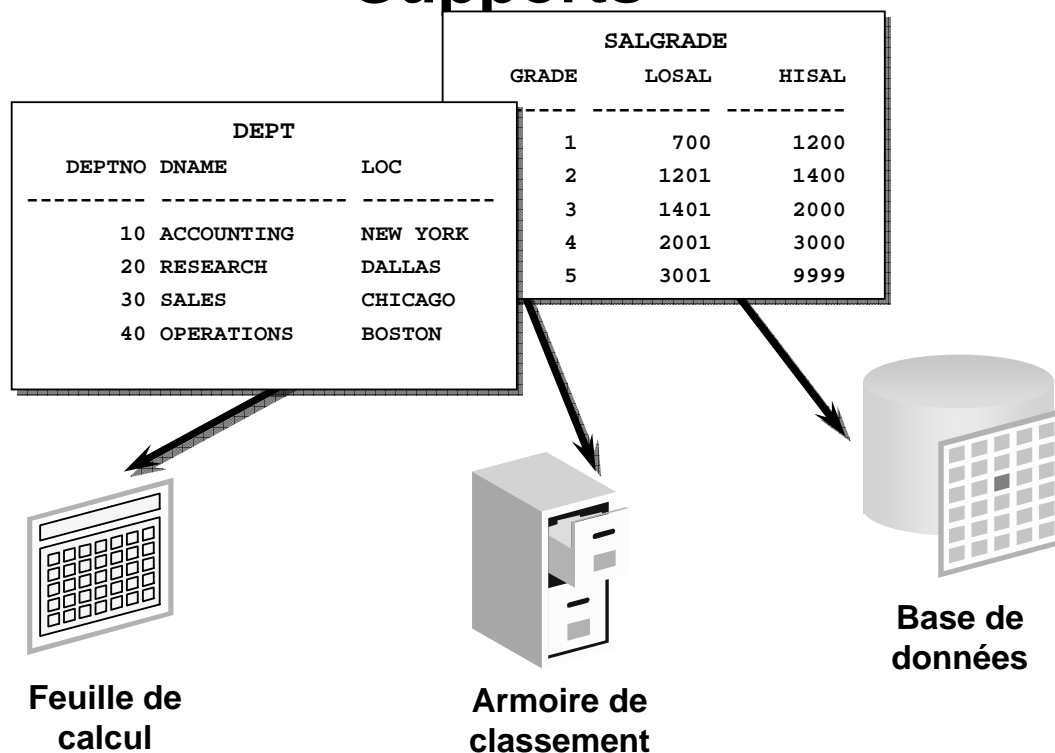
Production

Livraison du système aux utilisateurs et mise en oeuvre. Contrôle des performances et amélioration du système.

Maintenance du système.

Remarque : le cycle de vie d'un système est itératif. Ce cours se concentre sur la phase 'Build and Document'.

Stockage de Données sur Différents Supports



I-5

Stockage de l'Information

Chaque organisation a des besoins particuliers en matière d'information. Par exemple, une bibliothèque tient un registre de ses adhérents, de ses livres, des dates de prêt et des amendes. Une entreprise doit conserver des informations sur ses départements, ses employés et leur salaire. Tous ces éléments d'information sont appelés *données*.

Il est possible de stocker ces données sur différents supports et dans différents formats, par exemple sur un document papier, dans une armoire ou dans une feuille de calcul ou bien encore dans une base de données.

Une *base de données* est un ensemble organisé d'informations.

Pour gérer une base de données, il faut un système de gestion de bases de données (SGBD). Un SGBD est un programme qui permet de stocker, d'extraire et de modifier des données à la demande dans la base de données. Il existe cinq types principaux de bases de données : *hiérarchique*, *réseau*, *relationnel*, *relationnel objet* et *objet*.

Remarque : Oracle7 est un système de gestion de bases de données relationnelles et Oracle8 un système de gestion de bases de données relationnelles objet.

Principe d'une Base de Données Relationnelle

- En 1970, Dr E. F. Codd propose le modèle relationnel pour les systèmes de bases de données.
- Est à la base des systèmes de gestion de bases de données relationnelles (SGBDR).
- Composants du modèle relationnel :
 - Collection d'objets appelés encore relations
 - Ensemble d'opérateurs pour agir sur les relations
 - Règles d'intégrité pour garantir exactitude et cohérence des données

I-6

Le Modèle Relationnel

C'est le Dr. E. F. Codd qui, en juin 1970, a présenté pour la première fois les principes du modèle relationnel pour les bases de données dans un article intitulé "A Relational Model of Data for Large Shared Data Banks".

A cette époque, les modèles les plus utilisés étaient les modèles hiérarchique et réseau, voire de simples structures séquentielles. Les systèmes de gestion de bases de données relationnelles (SGBDR) furent rapidement adoptés en raison, notamment, de leur simplicité d'utilisation et de la flexibilité de leur structure. En outre, plusieurs fournisseurs novateurs, tels que Oracle, ont complété les SGBDR en leur adjoignant une série de puissants outils de développement d'applications et produits utilisateur, offrant ainsi une solution complète.

Composants du Modèle Relationnel

- Collections d'objets appelés encore relations pour stocker les données
- Ensemble d'opérateurs agissant sur les relations afin de produire d'autres relations
- Des règles d'intégrité pour garantir l'exactitude et la cohérence des données

Pour plus d'informations, reportez-vous à

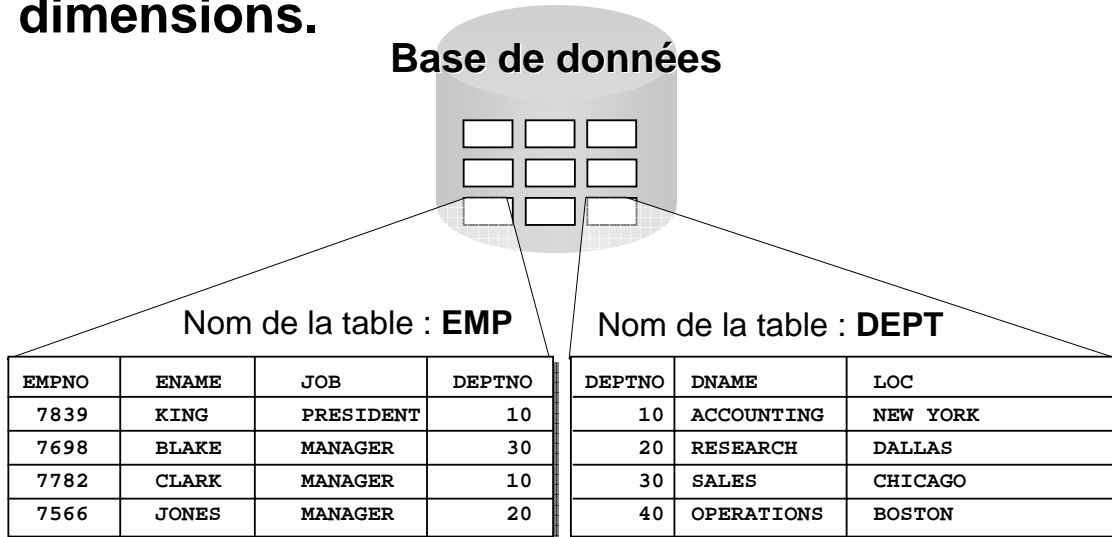


E. F. Codd, *The Relational Model for Database Management Version 2* (Reading, Mass. :

Addison-Wesley, 1990).

Définition

Une base de données relationnelle est un ensemble de relations ou tables à deux dimensions.



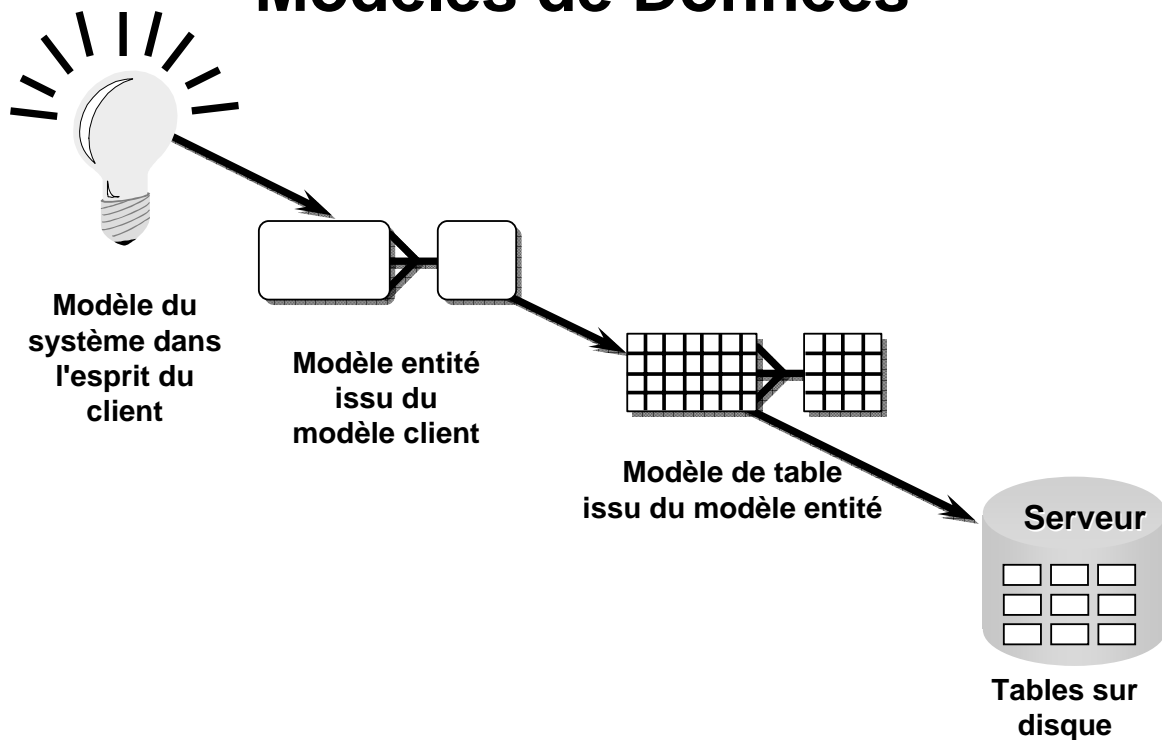
I-7

Base de Données Relationnelle

Une base de données relationnelle stocke l'information au moyen de relations ou tables à deux dimensions.

Supposons, par exemple, que vous souhaitiez stocker toutes les informations concernant les employés de votre société dans une base de données relationnelle. Vous pourriez créer plusieurs tables pour stocker différents éléments d'information sur vos employés, telles que la table des employés, la table des départements, et la table des salaires.

Modèles de Données



I-8

Modèles de Données

Toute conception repose sur un modèle. Ainsi, les ingénieurs automobile commencent par concevoir un modèle de voiture, afin d'en vérifier tous les détails avant de lancer la production. De la même manière, les concepteurs de systèmes développent des modèles pour tester les nouvelles idées et améliorer la conception des bases de données.

Rôle des Modèles

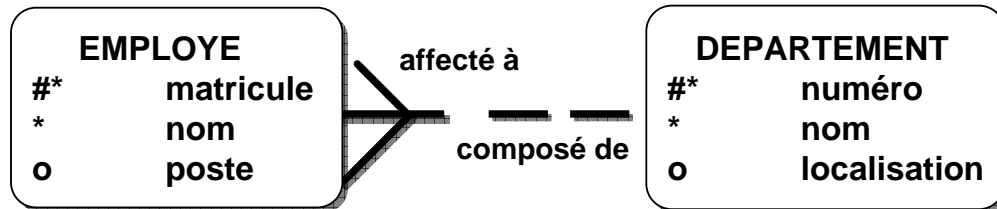
Les modèles facilitent l'explication et la compréhension des concepts. On peut les utiliser pour :

- Communiquer
- Classifier
- Décrire
- Spécifier
- Rechercher
- Evoluer
- Analyser
- Imiter

L'objectif est de produire un modèle adapté au maximum de ces usages, compréhensible par un utilisateur final et suffisamment précis pour permettre au développeur de créer un système de base de données.

Modèle Entité-Relation

- **Création d'un schéma entité-relation à partir de règles de gestion ou de comptes-rendus d'interviews.**



Scénario

- ". . . Affecter un ou plusieurs employés à un département . . ."
- ". . . Certains départements n'ont pas encore d'employés qui leur soient affectés. . ."

I-9

Modélisation ER (Entité-Relation)

Dans un système réel, les données sont divisées en catégories discrètes ou entités. Un modèle *ER* est la représentation des diverses entités qui existent au sein d'une société et de leurs interrelations. Un modèle ER est issu de règles de gestion ou de compte-rendus d'interview et est conçu au cours de la phase d'analyse du cycle de vie du système. Dans les modèles ER, l'information nécessaire à une société est séparée des activités de cette société. En effet, même si l'activité change, le type d'information, lui, a tendance à ne pas varier. Ainsi les structures de données tendent elles aussi à être constantes.

Avantages de la Modélisation ER

- Documente l'information dans un format clair et précis.
- Offre une idée précise de l'étendue des besoins d'information.
- Fournit une représentation graphique simple pour la conception de la base de données.
- Offre un cadre efficace pour l'intégration de plusieurs applications.

Les Composants Clés

- **Entité** : Élément d'une organisation, concret ou abstrait, identifiable, porteur d'informations et important pour l'activité considérée. Exemple : les départements, les employés, les commandes d'une société.
- **Attribut** : élément qui décrit ou qualifie une entité. Par exemple, pour l'entité employé, les attributs peuvent être le matricule, le nom, l'intitulé du poste, la date d'embauche et le numéro de département, etc., de l'employé. Chaque attribut est optionnel ou obligatoire.
- **Lien** : association entre entités ayant un nom et caractérisée par son caractère obligatoire ou optionnel et son degré. Exemple : employés et départements, commandes et articles.

Conventions de Modélisation selon le Modèle Entité-Relation

Entité

Rectangle

Nom singulier, unique

Majuscules

Synonyme entre parenthèses

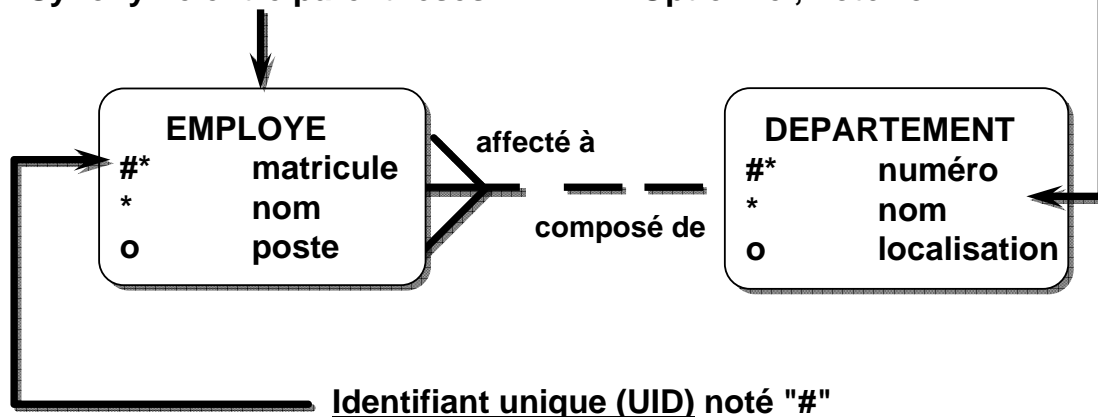
Attribut

Nom singulier

Minuscules

Obligatoire, noté "*"

Optionnel, noté "o"



I-10

Entités

Pour représenter une entité dans un modèle ER, on utilise les conventions suivantes :

- Rectangle de dimensions quelconques
- Nom unique et au singulier
- Nom en majuscules
- Synonymes optionnels, en majuscules et entre parenthèses : ()

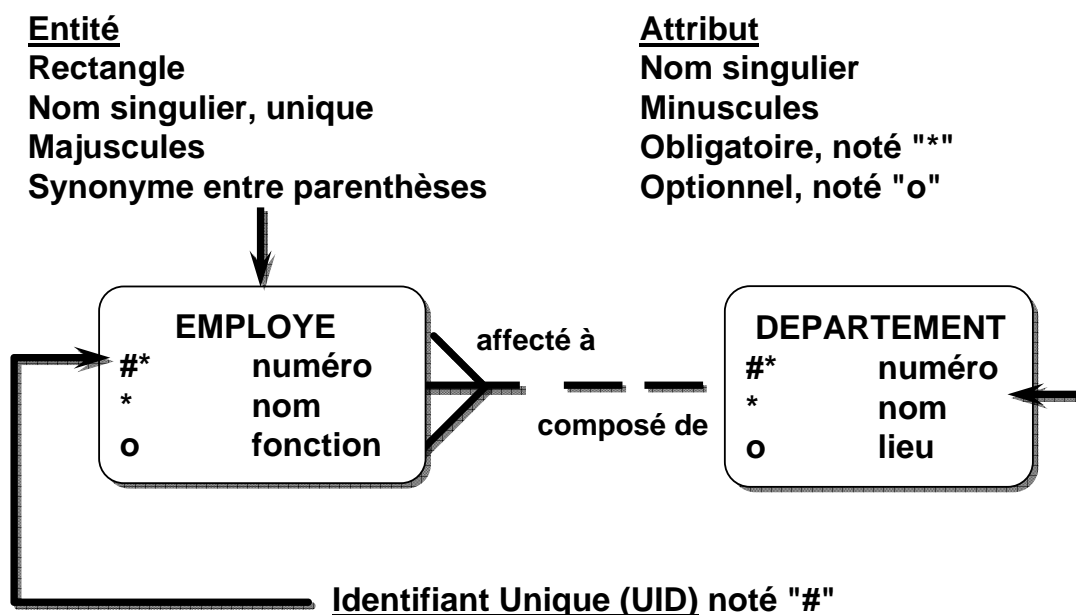
Attributs

Symbole	Description
Ligne discontinue	Élément optionnel signifiant "peut être"
Ligne continue	Élément obligatoire signifiant "doit être"
Patte d'oie	Élément de degré signifiant "un ou plus"
Trait	Élément de degré signifiant "un et seulement un"

Pour représenter un attribut dans un modèle, on utilise les conventions suivantes :

- Noms au singulier et en minuscules
- Astérisque pour baliser les attributs obligatoires ou les valeurs indispensables : *
- Lettre o pour baliser les attributs optionnels ou les valeurs non indispensables

Conventions de Modélisation selon le Modèle Entité-/Relation



I-11

Relations

Dans chaque sens, un lien comporte :

- Une désignation, par exemple, *enseigné par* ou *affecté à*
- Un caractère obligatoire ou facultatif : *doit être* ou *peut être*
- Un degré : *un(e)* et *un(e) seul(e)* ou *un(e) ou plus*

Remarque : le terme cardinalité est synonyme de degré.

Chaque entité source {peut être | doit être} désignation du lien {un(e) et un(e) seul(e) | un(e) ou plus} entité cible.

Remarque : la convention se lit dans le sens des aiguilles d'une montre.

Identifiant Unique

Un identifiant unique (UID) est une combinaison d'attributs ou de liens ou des deux, permettant de distinguer les différentes occurrences d'une entité. Chaque occurrence d'une même entité doit être identifiée de façon unique.

- Balisage de chacun des attributs de l'UID avec le symbole numérique : #

Remarque : si vous souhaitez obtenir plus d'informations sur la modélisation ER, vous pouvez assister aux cours de modélisation.

Terminologie des Bases de Données Relationnelles

2	3	4	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
	6									
			7839	KING	PRESIDENT		17-NOV-81	5000		10
			7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
			7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
			7566	JONES	MANAGER	7839	02-APR-81	2975		20
			7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
			7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
		5	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
			7900	JAMES	CLERK	7698	03-DEC-81	950		30
			7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
			7902	FORD	ANALYST	7566	03-DEC-81	3000		20
			7369	SMITH	CLERK	7902	17-DEC-80	800		20
			7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
			7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
1			7934	MILLER	CLERK	7782	23-JAN-82	1300		10

I-12

Terminologie des Bases de Données Relationnelles

Une base de données relationnelle peut contenir une ou plusieurs tables. La *table* est la structure de stockage élémentaire d'un SGBDR. Elle contient toutes les données nécessaires relatives à des éléments du monde réel, par exemple, des employés, des factures, des clients.

La diapositive ci-dessus montre le contenu de la *table* ou *relation* EMP. Les numéros désignent les éléments suivants :

1. Une *ligne unique* ou *tuple* indiquant toutes les données concernant un employé particulier. Chaque ligne de la table doit être identifiée par une clé primaire, ce qui permet d'éviter les doublons. L'ordre des lignes n'a pas d'importance ; il peut être spécifié lors de l'extraction des données.
2. Cette *colonne* ou *attribut* contenant le matricule des employés, est la clé primaire. Le matricule identifie un employé *unique* dans la table EMP. La colonne clé primaire doit obligatoirement être renseignée.
3. Cette colonne ne contient pas de valeur clé. Dans une table, une colonne représente un type de données ; ici, il s'agit des intitulés de postes de tous les employés. L'ordre des colonnes n'a pas d'importance pour le stockage des données ; il peut être spécifié lors de l'extraction des données.
4. Cette colonne contenant le numéro du département fait également office de *clé étrangère*. Une clé étrangère est une colonne qui définit la manière dont les tables sont liées entre elles. Elle fait référence à une clé primaire ou à une clé unique d'une autre table. Dans l'exemple, DEPTNO identifie de façon *unique* un département de la table DEPT.
5. Un *champ* se situe à l'intersection d'une ligne et d'une colonne. Il ne peut contenir qu'une seule valeur.
6. Un champ peut ne contenir aucune valeur. On parle alors de *valeur NULL*. Dans la table EMP, seules les lignes des employés ayant le statut de vendeur contiennent une valeur dans le champ COMM (commission).

Remarque : les valeurs NULL sont traitées plus en détails dans les chapitres suivants.

Relier Plusieurs Tables

- Chaque ligne de données d'une table est identifiée de manière unique par une clé primaire (PK).
- Les données de plusieurs tables peuvent être liées logiquement à l'aide de clés étrangères (FK).

Nom de la table : EMP

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20

Nom de la table : DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



Clé primaire



Clé étrangère



Clé primaire

I-13

Liaison de Plusieurs Tables

Chaque table contient des données qui décrivent une et une seule entité. La table EMP, par exemple, contient des données sur les employés. Un format de table permet immédiatement de visualiser, comprendre et utiliser l'information.

Les données concernant chaque entité distincte étant stockées dans différentes tables, il peut être nécessaire de combiner deux ou plusieurs tables afin de répondre à une question particulière. Supposons, par exemple, que vous vouliez savoir où se situe le département d'un employé. Vous avez besoin pour cela des données de la table EMP (contenant des informations sur les employés) et de la table DEPT (contenant des informations sur les départements). Un SGBDR permet de lier les données de deux tables au moyen de la clé étrangère. La clé étrangère est une colonne ou un ensemble de colonnes faisant référence à une clé primaire de la même table ou d'une autre table.

La possibilité de relier les données d'une table à celles d'une autre table permet d'organiser l'information en unités séparées et faciles à gérer. Ainsi, vous pouvez séparer de façon logique les données concernant les employés de celles concernant les départements, en les stockant dans des tables différentes.

Règles concernant les Clés Primaires et Etrangères

- La clé primaire ne doit comporter aucun doublon.
- La clé primaire ne peut généralement pas être modifiée.
- Les clés étrangères sont basées sur des valeurs des données. Ce sont des pointeurs purement logiques et non physiques.
- Une valeur de clé étrangère doit renvoyer à une valeur de clé primaire ou unique existante, ou bien à une valeur NULL.

Propriétés des Bases de Données Relationnelles

Une base de données relationnelle :

- **Peut être consultée et modifiée à l'aide d'ordres SQL (Structured Query Language)**
- **Contient une collection de tables sans pointeurs physiques**
- **Utilise un ensemble d'opérateurs**

I-14

Propriétés des Bases de Données Relationnelles

Dans une base de données relationnelle, il est inutile d'indiquer le chemin d'accès aux tables ni de connaître l'agencement physique des données.

Pour accéder à la base de données, vous exécutez un ordre du langage SQL (Structured Query Language) qui est le langage standard ANSI des bases de données relationnelles. Il comprend un grand nombre d'opérateurs qui permettent de partitionner et combiner les relations. La base de données peut être modifiée au moyen d'ordres SQL.

Communiquer avec un SGBDR au Moyen de SQL

Saisie de l'ordre SQL

```
SQL> SELECT loc  
2 FROM dept;
```

L'ordre est envoyé à
la base de données

Base de données

Affichage des données

```
LOC  
-----  
NEW YORK  
DALLAS  
CHICAGO  
BOSTON
```

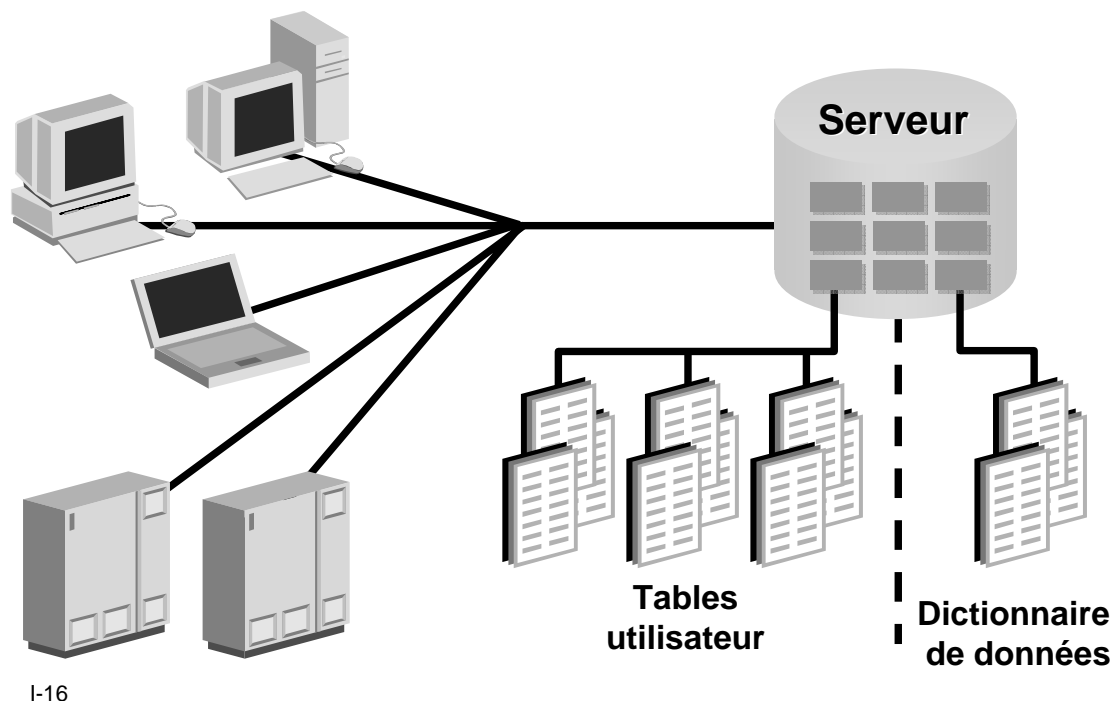
I-15

SQL

SQL permet de communiquer avec le serveur et présente plusieurs avantages. Il est :

- Efficace
- Facile à apprendre et à utiliser
- Complet sur le plan fonctionnel. SQL permet de définir, d'extraire et de manipuler des données des tables.

Système de Gestion de Bases de Données Relationnelles



I-16

Système de Gestion de Base de Données Relationnelle

Oracle offre un SGBDR flexible, Oracle Server. Ce système permet de stocker et gérer des données, avec tous les avantages que procure une structure relationnelle combinée à PL/SQL, moteur permettant de stocker et d'exécuter des unités de programme. Oracle Server permet aux utilisateurs des accès optimisés aux données. Il inclut des fonctions de sécurité qui contrôlent les accès à la base de données et son utilisation. Il comprend aussi des mécanismes de verrouillage qui assurent la cohérence et la protection des données.

Les applications Oracle peuvent être exécutées sur le même ordinateur qu'Oracle Server. Mais il est également possible d'exécuter les applications sur un ordinateur local et Oracle Server sur un autre système (architecture client-serveur). Un tel environnement client-serveur peut faire appel à de nombreuses ressources de traitement. Par exemple, une application de réservation de billets d'avion peut tourner sur un PC client, avec accès aux données de vol gérées par un système Oracle Server sur ordinateur central.



Pour plus d'information, reportez-vous à
Oracle8 Server Concepts Manual, Release 8.

Oracle8 : Système de Gestion de Bases de Données Relationnelles Objet

- **Types de données et objets définis par l'utilisateur**
- **Compatibilité relationnelle totale**
- **Support des objets de type multimedia et des large objects**
- **Fonctions de serveur de bases de données haut de gamme**

I-17

Oracle8

Oracle8 Server est la première base de données objet développée par Oracle. Les fonctions de modélisation des données d'Oracle7 Server ont été étendues pour prendre en charge le nouveau modèle de base de données relationnelle objet. Oracle8 Server fonctionne avec un nouveau moteur permettant la programmation orientée objet, les types de données et les objets de gestion complexes, ainsi qu'une compatibilité totale avec le monde relationnel.

Oracle8 Server offre de nombreuses améliorations par rapport à Oracle7 Server. Notamment, il inclut de nombreuses fonctions destinées à augmenter les performances et fonctionnalités des applications de traitement transactionnel en ligne (OLTP) comme, par exemple, un meilleur partage des structures de données lors de l'exécution, des buffers et des caches de taille supérieure, et des contraintes dont il est désormais possible de différer la vérification. Les applications de data warehouse vont bénéficier d'améliorations telles que l'exécution en parallèle des opérations d'insertion, de mise à jour et de suppression, le partitionnement et l'optimisation des requêtes en parallèle. Parce qu'il fonctionne dans le cadre de l'architecture NCA (Network Computing Architecture), Oracle8 Server est compatible avec les applications client-serveur et Internet distribuées et multi-niveaux.

Oracle8 Server accepte des dizaines de milliers d'utilisateurs simultanés, prend en charge jusqu'à 512 péta-octets et peut manipuler n'importe quel type de données (texte, spatial, image, son, image vidéo, série chronologique, mais aussi des données structurées traditionnelles).



Pour plus d'informations, reportez-vous à
Oracle8 Concepts Manual, Release 8

Définition de l'Objet

Un objet :

- **Peut être une personne, un lieu ou une chose**
- **Connaît ses caractéristiques et exécute des actions**
- **A une identité**



**Je suis une pendule, je
connais mon *fuseau*
horaire et je peux
*indiquer l'heure***

I-18

Les Objets

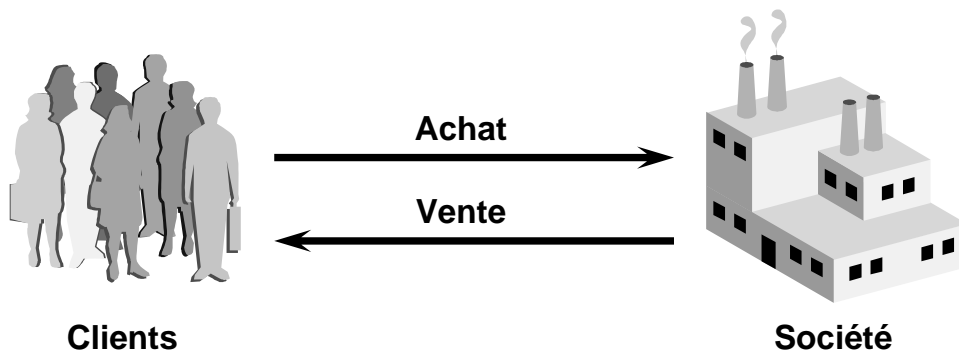
Les objets sont souvent considérés comme étant une représentation d'une chose appartenant au monde réel. On dit d'eux qu'ils "savent" faire des choses. Il est possible de "demander" à un objet employé de calculer ses charges salariales ou à un objet commande de s'auto-expédier.

Voici quelques définitions :

- "Un objet est un 'package' logiciel contenant un ensemble de procédures (méthodes) et de données (variables) associés." David Taylor, *Object-Oriented Technology: A Manager's Guide* (Reading, Mass.: Addison-Wesley, 1981)
- "Un objet est un concept, une abstraction ou une chose ayant des limites et un sens clairs pour le problème à résoudre." James Rumbaugh, et. al., *Object-Oriented Modeling and Design* (Englewood Cliffs, N.J.: Prentice-Hall, 1991)

Utilisation d'un Modèle Objet

- Les objets modélisent le problème à résoudre.
- Le modèle exprime les interactions existant entre les objets
- Les modèles objet reflètent la réalité



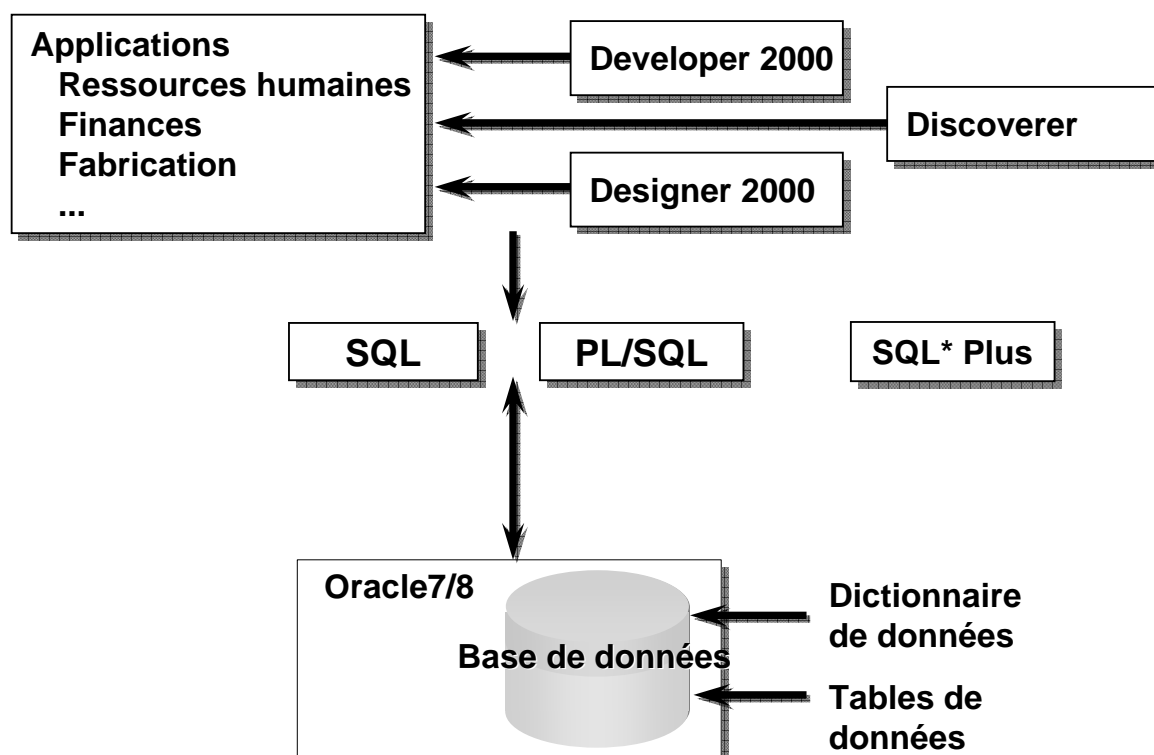
I-19

Les Modèles Objet

La technologie objet a été créée dans le but de modéliser des problèmes de gestion. Le modèle exprime les interactions qui existent entre les objets.

Lorsqu'ils travaillent avec des objets, les développeurs réfléchissent davantage en termes de besoins de l'application qu'en termes d'architecture des systèmes d'exploitation et d'exigences de l'environnement de développement.

La Solution Complète Oracle



I-20

La Solution Complète Oracle

Le système de gestion de bases de données relationnelles Oracle est le principal produit de l'offre Oracle. Il comprend Oracle Server ainsi que divers outils destinés à aider l'utilisateur pour la mise à jour, le contrôle et l'utilisation des données. Le dictionnaire de données Oracle est l'un des composants les plus importants de Oracle Server. Il est constitué d'un ensemble de tables et de vues permettant un accès en lecture seule à la base de données.

Le SGBDR gère des tâches telles que les suivantes :

- Stockage et définition des données
- Contrôle et restriction des accès aux données
- Sauvegarde et restauration
- Interprétation des ordres SQL et PL/SQL

Remarque : PL/SQL est un langage procédural qui complète les possibilités de SQL avec une logique d'application.

Les ordres SQL et PL/SQL sont systématiquement utilisés pour rechercher et manipuler les données stockées dans la base Oracle. Cependant, dans le cadre de programmes d'application, il arrive souvent que vous accédiez à la base de données sans utiliser directement SQL ou PL/SQL : vous vous contentez de cliquer sur un bouton ou de cocher une case par exemple, mais en fait, les applications utilisent implicitement SQL ou PL/SQL lorsqu'elles exécutent la requête.

SQL*Plus est un outil Oracle qui reconnaît les ordres SQL et PL/SQL et en lance l'exécution à partir du serveur. Il a son propre langage de commande.

Oracle propose une large gamme d'outils pilotés par une interface graphique (GUI) de pointe pour la conception d'applications de gestion, ainsi qu'un vaste éventail d'applicatifs destinés à de nombreux secteurs du commerce et de l'industrie.

Remarque : les dictionnaires de données Oracle sont traités plus en détails dans les chapitres suivants.

Les Ordres SQL

SELECT	Recherche de données
INSERT UPDATE DELETE	Langage de manipulation des données (LMD)
CREATE ALTER DROP RENAME TRUNCATE	Langage de définition des données (LDD)
COMMIT ROLLBACK SAVEPOINT	Contrôle des transactions
GRANT REVOKE	Langage de contrôle des données (LCD)

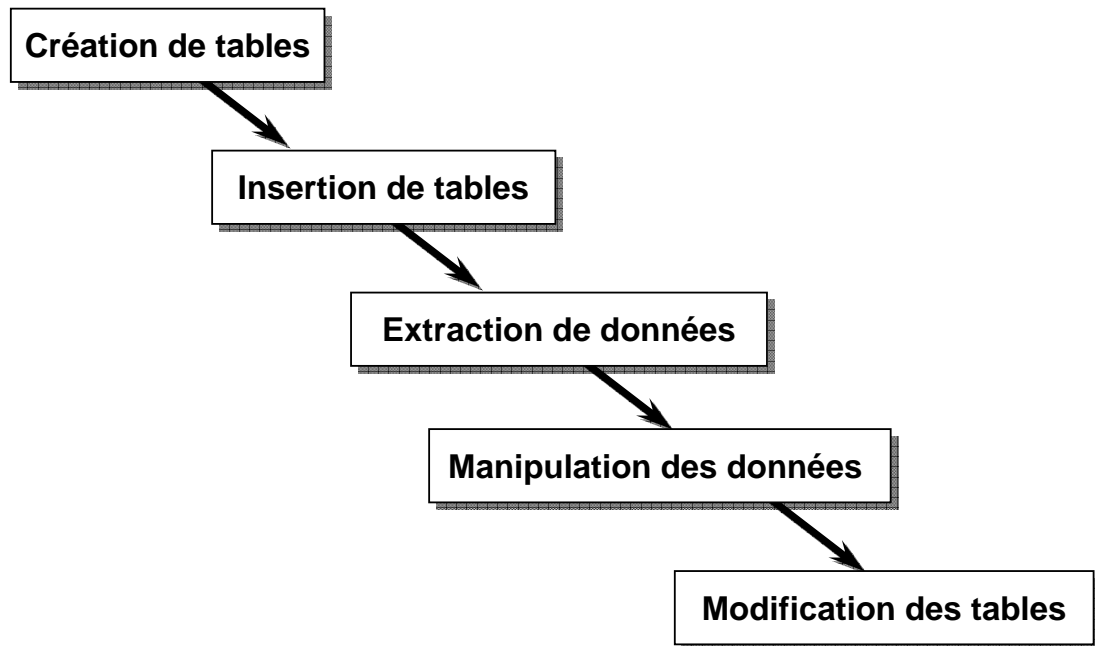
I-21

Les Ordres SQL

Le SQL d'Oracle est conforme aux standards reconnus. La société Oracle garantit la conformité future avec l'évolution des standards en faisant participer certains de ses principaux collaborateurs aux comités de normalisation SQL. Les deux organismes reconnus sont l'ANSI (American Standards Institute) et l'ISO (International Standards Organization). Tous deux ont adopté SQL comme langage standard des bases de données relationnelles.

Ordre	Description
SELECT	Extraction des données de la base de données.
INSERT UPDATE DELETE	Respectivement, ajout, modification et suppression de lignes dans les tables de la base de données. L'ensemble est appelé <i>langage de manipulation des données (LMD)</i> .
CREATE ALTER DROP RENAME TRUNCATE	Initialisation, modification et suppression des structures de données dans une table. L'ensemble est appelé <i>langage de définition des données (LDD)</i> .
COMMIT ROLLBACK SAVEPOINT	Gestion des modifications apportées par les ordres LMD. Les modifications apportées aux données peuvent être regroupées en <i>transactions logiques</i> .
GRANT REVOKE	Accorde ou retire les droits d'accès à la base de données Oracle et aux structures qu'elle renferme. L'ensemble est appelé <i>langage de contrôle des données (LCD)</i> .

Présentation du Cours



I-22

Présentation du Cours

Ce cours comprend dix-neuf chapitres couvrant les sujets suivants :

- Création de tables
- Insertion de données
- Extraction de données
- Manipulation des données
- Modification des tables

Chaque chapitre débute par une présentation des objectifs et se termine par des exercices pratiques.

Ce cours s'attache aux aspects relationnels des systèmes de gestion de bases de données Oracle. Il ne traite pas des objets.

Vous utiliserez SQL*Plus pour saisir et exécuter des ordres SQL.

Tables Utilisées dans le Cours

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7822	MILLER	CLERK	7782	23-JAN-82	1300		10

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Tables Utilisées dans le Cours

Vous utiliserez principalement trois tables dans ce cours :

- La table EMP qui contient des informations sur tous les employés
- La table DEPT, qui contient des informations sur tous les départements
- La table SALGRADE , contenant des informations sur les différents niveaux de salaires en fonction de l'échelon

La structure et les données de chaque table sont données dans l'annexe B.

Résumé

- **Les bases de données relationnelles:**
 - **Sont composées de relations**
 - **Possèdent des opérateurs relationnels**
 - **Sont régies par des contraintes d'intégrité des données**
- **Oracle Server permet de stocker et gérer l'information au moyen du langage SQL.**
- **Oracle8 est fondé sur le système de gestion de bases de données relationnelles objet.**

I-24

Résumé

Les systèmes de gestion de bases de données relationnelles sont constitués de relations. Ils possèdent des opérateurs et sont régis par des contraintes d'intégrité des données.

Oracle Corporation offre des produits et services qui répondent à vos besoins en matière de systèmes de gestion de bases de données relationnelles. Le produit central, Oracle Server, permet de stocker et gérer l'information au moyen de SQL et du moteur PL/SQL pour les structures procédurales.

Oracle Server est fondé sur le standard ANSI pour SQL et comprend des extensions. SQL est le langage qui permet de communiquer avec le serveur pour accéder aux données, les manipuler et les contrôler.