

LE LANGAGE DE REQUETES SQL2

- ♦ Les niveaux
- ♦ SQL2 Intermédiaire
- ♦ Introduction à SQL3

1. Les niveaux

- Trois niveaux distingués :
 - Entry SQL2 = SQL89 + manques
 - Intermediate SQL2 = Compléments relationnels
 - Full SQL2 = Gadgets en plus

SQL2 Entry

- Codes réponses SQLSTATE
- Renommage des colonnes résultats
- Mots clés utilisables entre " "
- Méta-base normalisée (schémas)

2. SQL2 Intermédiaire

- SQL relationnel complet
- Support de la notion de domaine
 - Domaine = type de données contraint
- Support du temps
 - Type Date, Time, ... avec opérateurs
- Meilleur support des chaînes de caractères
- Opérateurs algébriques mieux intégrés
- Modification de schémas

SQL2 Intermediate : Création de domaines

CREATE DOMAIN <nom> <type> [valeur] [Contrainte]

<type> ::=

CHAR [(n)] | VARCHAR [(n)]

BIT [**VARYING**] [(n)]

SMALLINT | INTEGER

NUMERIC [p, [q]] | DECIMAL [p, [q]] | FLOAT [n]

DATE | TIME | TIMESTAMP | INTERVAL

<valeur> ::= DEFAULT

constante

USER | NULL

CURRENT_DATE | CURRENT_TIME | CURRENT_TIMESTAMP

Exemples de domaine (1)

- `CREATE DOMAIN cru VARCHAR(20) ;`
- `CREATE DOMAIN couleur_vins CHAR(10)
DEFAULT 'rouge' ;`
- `CREATE DOMAIN date_commande DATE ;`
- `CREATE DOMAIN quantite SMALLINT ;`

Exemple de domaine (2)

- Possibilité de créer des domaines avec contraintes :
 - CREATE DOMAINE MONEY IS DECIMAL (5,2)
 - DEFAULT (-1)
 - CHECK (VALUE = -1 OR VALUE > 0)
 - NOT NULL

SQL2 : Types

- Types de données date avec opérations
 - DATE, TIME et TIMESTAMP
 - Intervalles de temps
- Cascade des mises à jour
 - Suppression en cas d' intégrité référentielle avec options
 - Cascader les suppressions (CASCADE)
 - Rendre nul l'attribut référençant (NULLIFY)
- Différents alphabets et ordres de lettres

Traitement des chaînes de caractères

expression_caractère ::=

'chaîne_caractère'

<nom_colonne>

USER

UPPER (expression_caractère)

LOWER (expression_caractère)

CHARACTER_LENGTH (expression_caractère)

SUBSTRING (expression_caractère FROM début FOR longueur)

POSITION (expression_caractère IN expression_caractère)

CAST (expression AS type | domaine)

expression_caractère | | expression_caractère

Exemple de chaînes de caractères

- Q11 :Donner le numéro et les 5 premières lettres du cru en majuscule pour chaque vin dont le cru possède plus de 10 lettres

```
SELECT V#, UPPER(SUBSTRING(CRU FROM 1 TO 5))  
FROM VINS  
WHERE CHARACTER_LENGTH(CRU) > 10
```

Traitement du type date

expression_date_temps ::=

constante

<nom_colonne>

CURRENT_DATE

CURRENT_TIME [précision]

CURRENT_TIMESTAMP [précision]

EXTRACT (champ FROM source)

CAST (expression AS type | domaine)

Traitement du type date

- Exemples :

```
UPDATE VINS  
SET ANNEE = EXTRACT (YEAR FROM CURRENT_DATE)  
WHERE V# = 10
```

```
UPDATE COMMANDES  
SET DATE_COM = CAST ('1996-10-23' AS DATE)  
WHERE V# = 10
```

SQL2 : Opérateurs

- Jointure externe (outer-join)
 - SELECT ...
 - FROM R1 [NATURAL] [{LEFT | RIGHT}] JOIN R2
[ON (A=B)], ...
 - WHERE ...
- Expressions de SELECT
 - [OUTER] UNION
 - INTERSECT
 - EXCEPT

Exemple de Jointure Externe

- Q12 :Afficher le numéro et le cru des vins ainsi que la quantité commandée pour ceux d'entre eux qui ont été commandés ou non
 - SELECT V#, CRU, QTE
 - FROM VINS NATURAL LEFT JOIN COMMANDES
- ou bien
 - SELECT V#, CRU, QTE
 - FROM VINS LEFT JOIN COMMANDES USING (V#)
- ou encore
 - SELECT V#, CRU, QTE
 - FROM VINS LEFT JOIN COMMANDES ON V.V# = C.V#

SQL2 : Création d'index

```
CREATE [UNIQUE] INDEX [nom_index]  
ON nom_table  
( <nom_colonne [ASC | DESC] > * );
```

Exemple:

```
CREATE UNIQUE INDEX index_exp  
ON EXPEDITION (num_exp ASC) ;
```

SQL2 : Création de schéma

```
CREATE SCHEMA [nom_schéma]  
[ AUTHORIZATION <nom_utilisateur>]  
[DEFAULT CHARACTER SET <jeu_car>]  
[ <éléments_du_schéma> * ];
```

Exemple:

CREATE DATABASE VIGNOBLES // non normalisé

CREATE SCHEMA COOPERATIVE

CREATE DOMAIN ...
CREATE TABLE ...

SQL2 : Modification de schéma

DROP

DOMAIN <nom_domaine>

TABLE <nom_table>

INDEX <nom_index>

SCHEMA <nom_schema>

[**CASCADE** | **RESTRICT**] ;

RESTRICT interdit la destruction d'un objet référencé (via une contrainte d'intégrité, une vue, ...) alors que **CASCADE** propage la destruction

SQL2 : Modification de schéma

ALTER TABLE <nom_table>

| **ADD**

| | **COLUMN** <def_colonne>

| | **CONSTRAINT** <def_contrainte_table >

| **ALTER** <def_colonne>

| **DROP**

| | **COLUMN** <nom_colonne>

| | **CONSTRAINT** <nom_contrainte >

Exemples de modification de schema

ALTER TABLE EXPEDITION

ADD COLUMN adresse_livraison **VARCHAR(30)**

ALTER quantite **INTEGER NOT NULL ;**

SQL2 Full

- Extension des dates et temps
- Expressions étendues avec correspondances de colonnes
- Possibilité de SELECT en argument d'un FROM
- Vues concrètes
- Contraintes d'intégrité multi-tables
- Contrôles d'intégrité différés

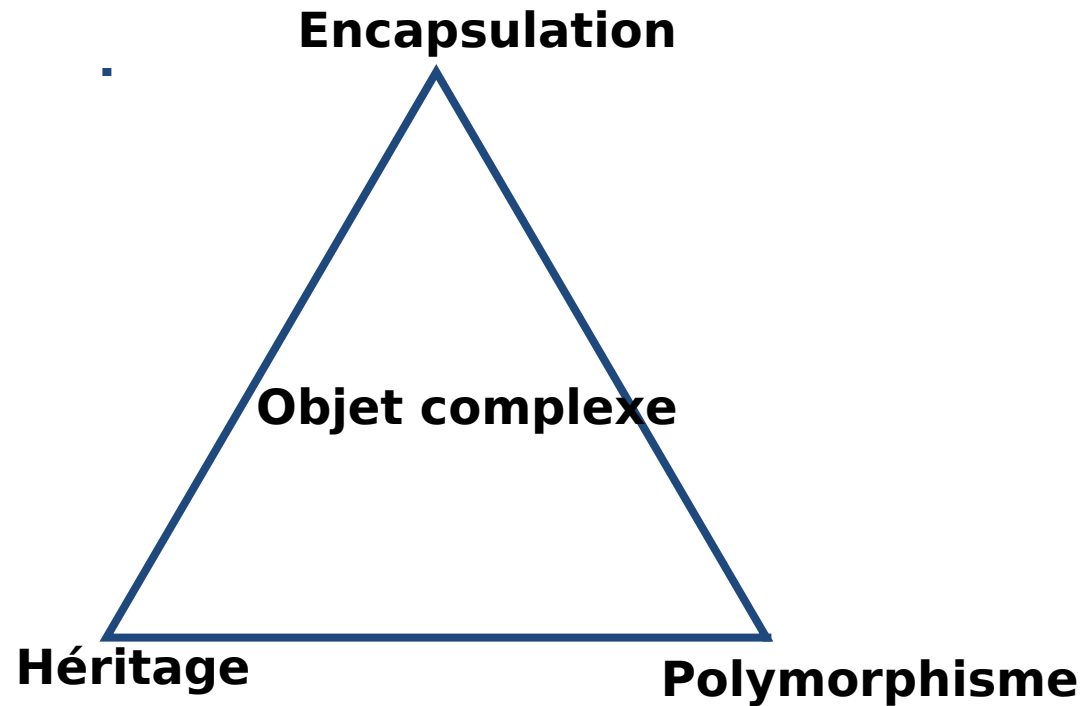
3. SQL3 et l'objet-relationnel

- Standard ISO accepté depuis 1999
 - SQL89, SQL92, SQL99
- Objectif :
 - étendre le standard SQL2 (adopté par l'ISO en 1992) avec des fonctionnalités objets et autres
- Forme :
 - un ensemble de composants dont foundation, PSM, CLI, ...
- Implémentation progressive dans les SGBD
 - DB2, Informix, Oracle, Versant, ...

Intégration de l'objet

- 1990 : The Object-Oriented Manifesto
- 1991 : The Third Generation Manifesto
- 1996 : l'objet-relationnel s'impose dans l'industrie
- 1999 : Normalisation du langage SQL3
- Fonctionnalités :
 - extension du relationnel avec les concepts de l'objet
 - support de types de données complexes et extensibles
 - extension de SQL pour le support des objets
 - intégration des règles (triggers)

Concepts fondamentaux



L'apport des modèles objets


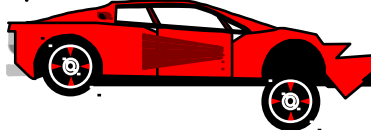

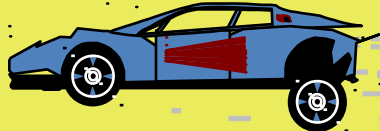

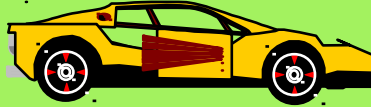
- Identité d'objets
 - introduction de pointeurs invariants
 - possibilité de chaînage
- Encapsulation des données
 - possibilité d'isoler les données par des opérations
 - facilite l'évolution des structures de données
- Héritage d'opérations et de structures
 - facilite la réutilisation des types de données
 - permet l'adaptation à son application
- Possibilité d'opérations abstraites (polymorphisme)
 - simplifie la vie du développeur

Les types d'objets complexes

- Nécessité d'introduire des attributs multivalués
- Fourniture de collections prédéfinies telles liste, ensemble, tableau, ...
- Imbrication des collections pour représenter des objets très compliqués
- Exemple :
 - Type Point {real X, real Y}
 - Type Ligne { list <Points> }
 - Type PolyLigne {list <Lignes> }

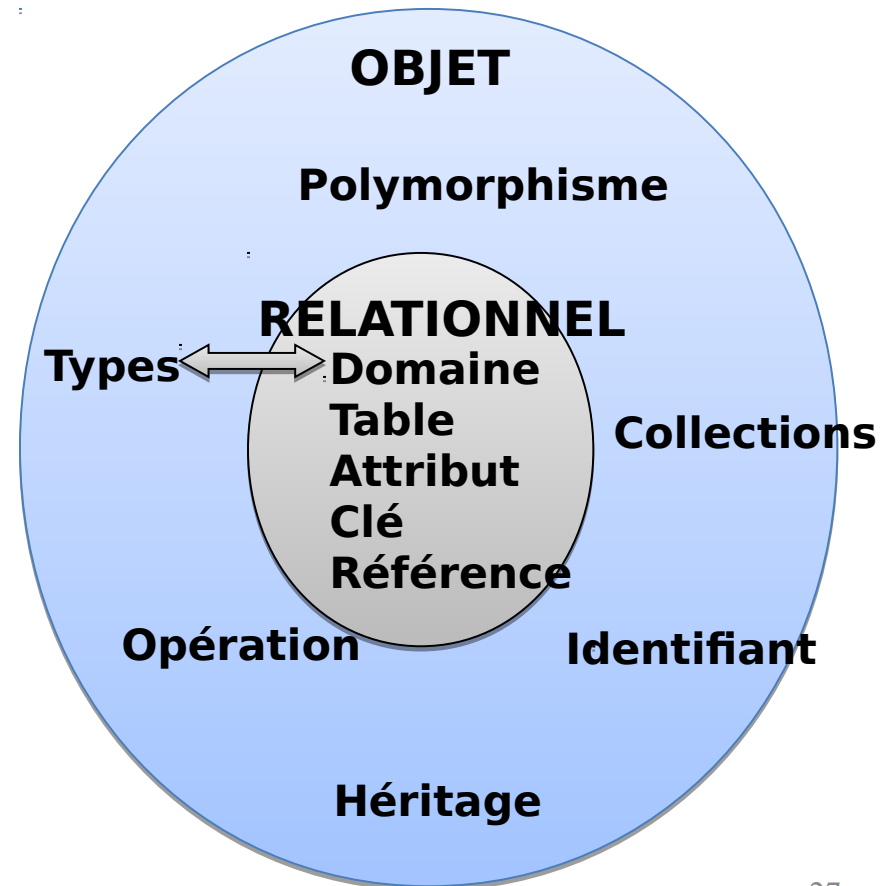
Extension des tables relationnelles

- Peuvent contenir des attributs de type complexe
- Peuvent être des tables d'objets

Incident Table			
Policy#	Incident#	Report	Photo
24	134		
24	219		
24	037		

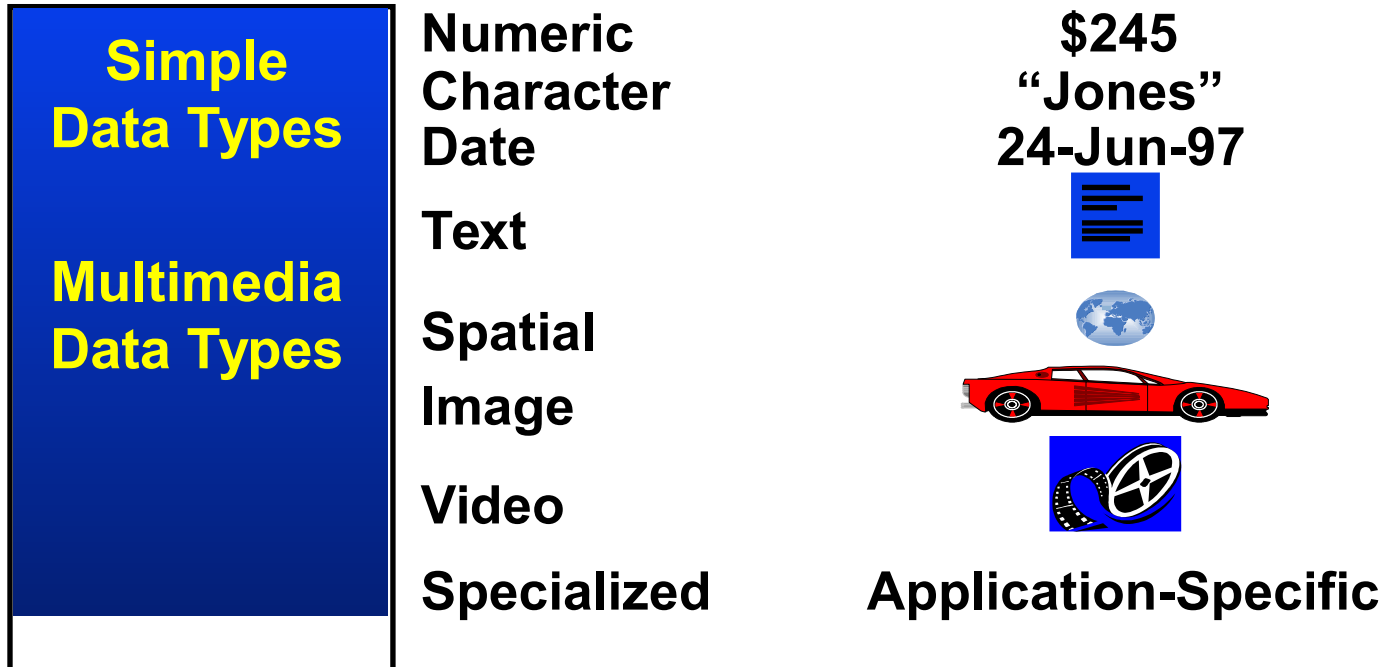
Le modèle objet-relationnel

- Extension du modèle relationnel
 - attributs multivalués : structure, liste, tableau, ensemble
 - héritage sur tables et types
 - domaine type abstrait de données (structure cachée + méthodes)
 - identité d'objets
- Object-Relational en anglais



Exemple de création de type

- **CREATE TYPE** Employe (E# int, Nom varchar(32) Prenom varchar(16), BirthDate date, OPERATION age (BirthDate) int) ;
- Un ensemble de types sont inclus au SGBD



Exemple de création de tables

- **CREATE TABLE** Incident (Police# int, Incident# int, Rapport text, Photo image)
- **CREATE TABLE** Employes of TYPE employe
- **CREATE TABLE** Employes (Description employe)

Extension des requêtes

- peuvent appeler les méthodes
 - argument du **SELECT**
 - argument du **WHERE**
- peuvent traverser les chemins
 - découpage des attributs complexes
 - parcours des associations
- peuvent parcourir les collections
 - en argument du **FROM**
 - possibilité de collection dépendantes

Exemples de requêtes

```
SELECT nom  
FROM Employes e  
WHERE e.age() < 35;
```

```
SELECT nom, adresse  
FROM Employes e  
WHERE e.adresse.dept() = « Mfoundi"  
AND e.adresse.ville = "Yaoundé";
```

4. CONCLUSION

- Un standard de plus en plus complet très suivi
 - Attention aux approximations et imitations incomplètes
 - Tout ou presque existe dans les propositions SQL2 ou SQL3
 - Une référence pour implémenter et utiliser chaque aspect des BD
- Le langage de communication inter-système
 - RDA, TP, SQL ACCESS GROUP (CLI, FAP)
- Le langage universel sur lequel s'appuie les progiciels