Bases de données relationnelles - 1

S8 - BDTN 2025

1

Systèmes de Gestion de Bases de Données (SGBD) relationnels ?

PL/SQL vs SQL

Jointures

PL/SQL : les principaux types de blocs

SGBD Relationnels

Première implémentation au début des années 1980

Oracle est la première base de données commerciale à implémenter le modèle relationnel

Ces SGBD sont les plus répandus aujourd'hui

- Microsoft SQL Server, Oracle
- MySQL, PostgreSQL

Les informations sont stockées
Dans des TABLES comportant des
Lignes et des colonnes

Table CUSTOMER_CST

CST_ID	CST_EMAIL	CST_LAST_NAME	CST_FIRST_NAME	CST_PHONE
1	rnelson@machin.fr	NELSON	Ryan	0683242254
2	barry.robichaux@truc.fr	ROBICHAUX	Barry	0228794613
3	cjones@machin.fr	JONES	Chandi	0659157846
4	djekeye@machin.fr	DJEKEYE	(null)	0659157879
5	imartin567@voila.fr	MARTIN	Isabelle	(null)

Clé primaire

Relation

Clé étrangère

1 colonne
⇒ 1 type de données
⇒ 1 valeur maxi par colonne et par ligne

Table BOOKING_BKG

BKG_ID	BKG_TOTAL_SEAT	BKG_CST_ID	BKG_SHW_ID
100	2	1	10
101	4	3	10
102	1	2	14
103	1	1	21
104	2	4	22

Autres SGBD

Dans la dominante BDTN

- Cassandra
- MongoDB
  - BDD orientée documents

```
{
    _id: "5cf0029caff5056591b0ce7d"
    "nom":"Honkytonk Man",
    "realisateur":{
        "nom":"Eastwood",
        "prenom":"Clint"
    },
    "annee":1982,
    "acteurs":[
        {
            "nom":"Eastwood",
            "prenom":"Kyle"
        },
        {
            "nom":"Eastwood",
            "prenom":"Clint"
        }
    ]
}
```

Document
Liste clé / valeur dans un format similaire à JSON
Identifié par un unique identifiant généré par MongoDB
Une propriété du d'un document peut contenir des tableaux, des structures

SQL et PL/SQL

PL/SQL et SQL

SQL

- Standard ANSI (American National Standards Institute)
- Non procédural

PL/SQL

- Oracle
- Extension procédurale
- Utiliser SQL

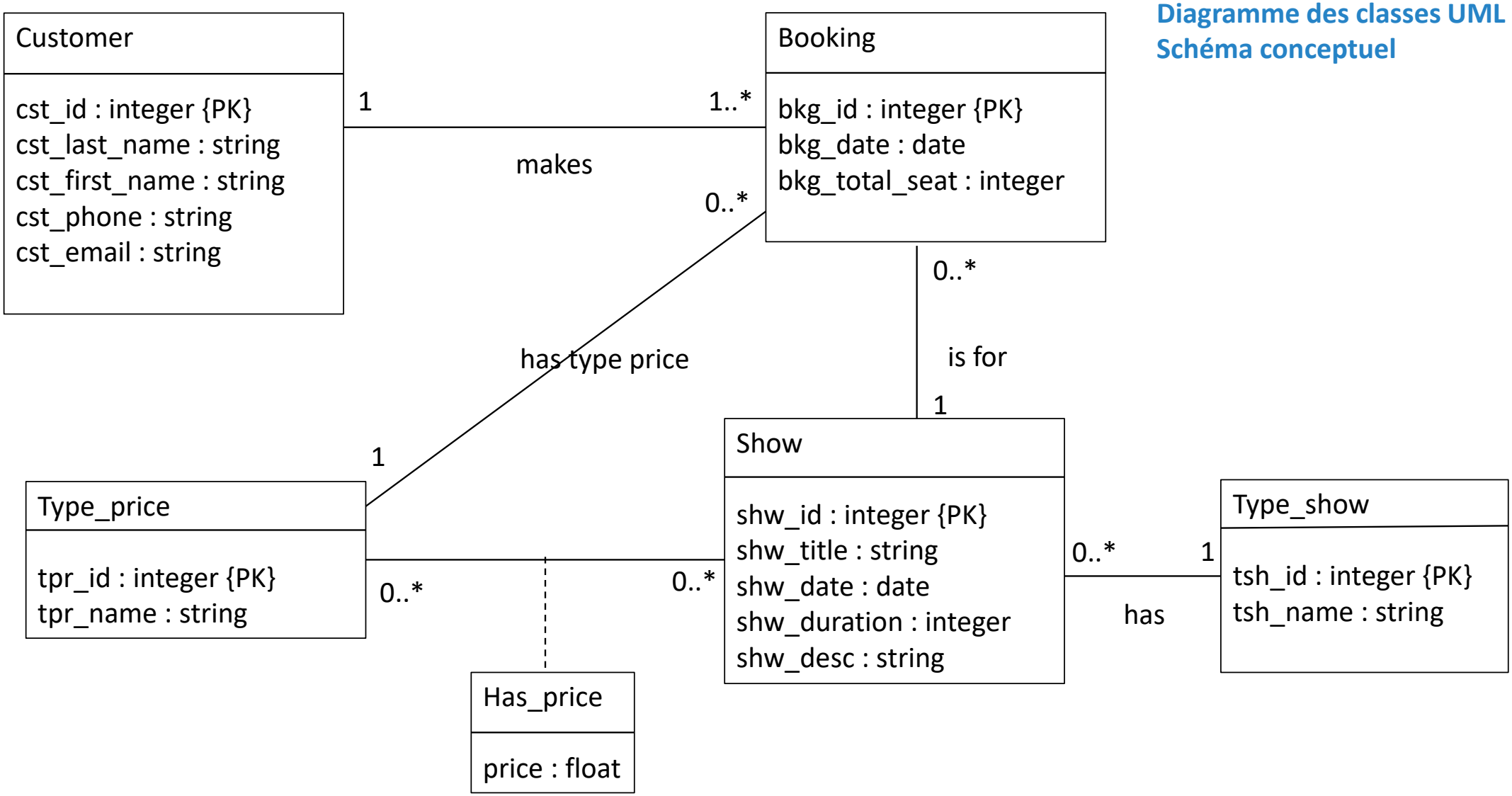
Quelques-uns des bénéfices de PL/SQL

- Meilleures performances
- Portabilité

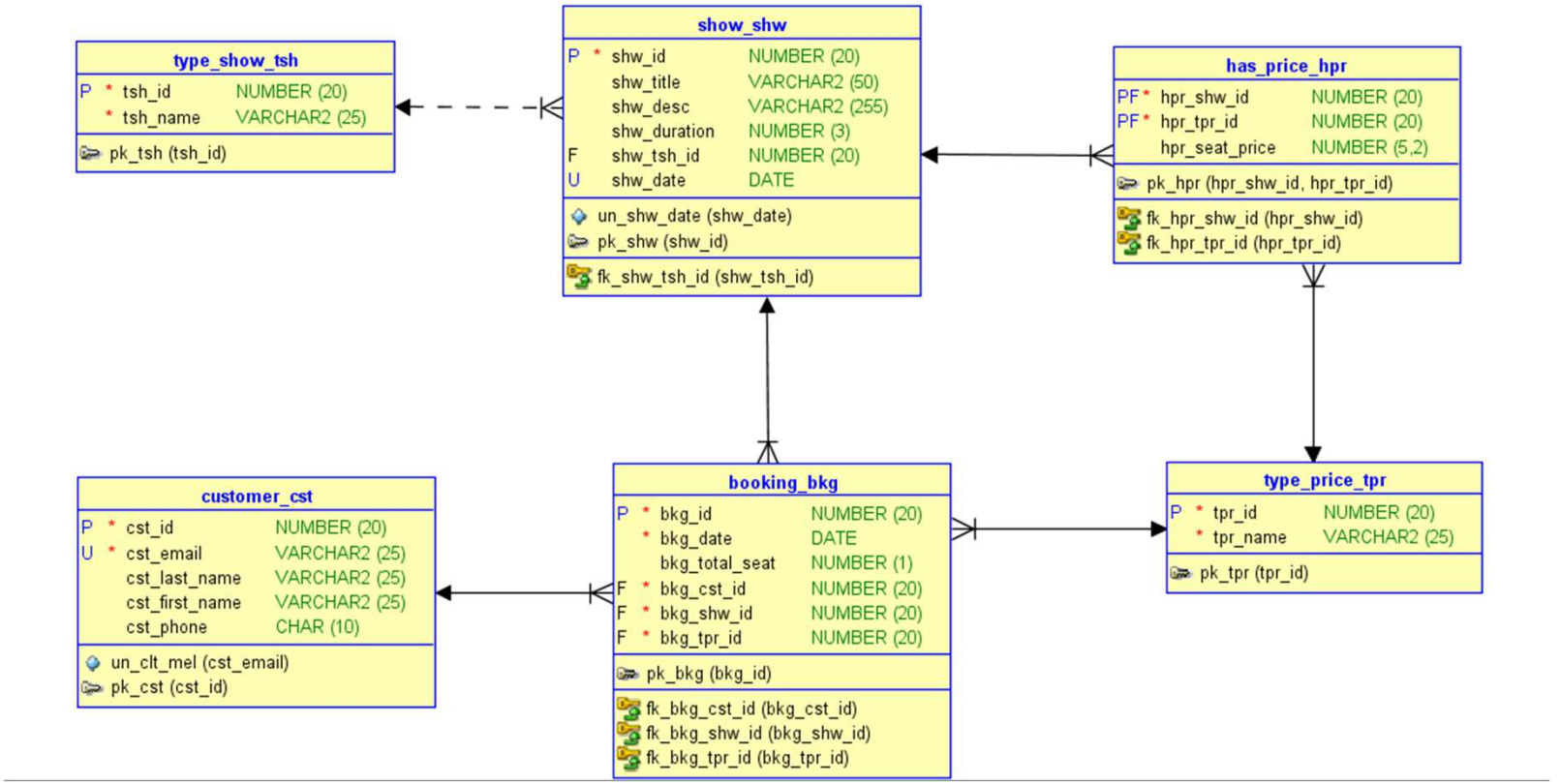
Révisions

1. Modèle conceptuel / Modèle physique
2. Clé primaire
   Définition, exemple, à quoi ça sert
3. Clé étrangère
   Définition, exemple, à quoi ça sert
4. Clé naturelle vs clé de substitution
   Définition, exemple, intérêt de l'un ou de l'autre
5. Contraintes
   Définition, exemple, intérêt
6. Diagramme conceptuel vers schéma relationnel : comment sont traduites les associations 1 à plusieurs
7. Diagramme conceptuel vers schéma relationnel : comment sont traduites les associations plusieurs à plusieurs

Diagramme des classes UML
Schéma conceptuel



Modèle physique
Représentation graphique des tables
avec colonnes, types et contraintes pour Oracle



Oracle Data Modeler

SQL

Instruction SQL simple

```
SELECT col1, col2, …, coln
  FROM table
  [WHERE condition]
  ORDER BY col1 [ASC|DESC],
                col2 [ASC|DESC],
                 …
                coln [ASC|DESC] ;
```

Exemple

```
SELECT INITCAP(shw_title) AS titre,
       ROUND(shw_duration/60, 1) || ' h' AS Durée,
       'Excellent' AS  Appréciation,
       shw_date + 7
FROM show_shw
ORDER BY shw_tsh_id, shw_date ASC;
```

TITRE	DURÉE	APPRÉCIATION	SHW_DATE+7
Calacas - Zingaro	1,5 h	Excellent	02/10/24
Luna Rossa	1,3 h	Excellent	03/10/24
Les Aveugles	1,5 h	Excellent	05/10/24
Le Canard À L'Orange	1,7 h	Excellent	02/11/24
Les Faux British	1,7 h	Excellent	16/11/24
Out Of Time	1,3 h	Excellent	06/10/24
Fiordalisi	1,5 h	Excellent	10/10/24
Kazut De Tyr	1,5 h	Excellent	08/10/24
Le Comte Ory	2 h	Excellent	27/10/24
Martialsolal Trio	1,5 h	Excellent	28/10/24
Pffffff !	1,3 h	Excellent	16/10/24
Le Soir Des Monstres	1,3 h	Excellent	19/01/25
Sophia Aram	1,3 h	Excellent	21/01/25

Les principaux opérateurs « JOIN »

Jointure naturelle
- NATURAL JOIN

Produit cartésien
- CROSS JOIN

Jointure
- INNER JOIN

Jointure externe
- OUTER JOIN

```
SELECT *
FROM T1 NATURAL JOIN T2;
```

T1

A	B
1	10
2	20
3	30

T2

B	C
10	100
15	150
20	200

A	B	C
1	10	100
2	20	200

```
SELECT *
FROM T1 CROSS JOIN T2;
```

T1

A	B
1	10
2	20
3	30

T2

B	C
10	100
15	150
20	200

A	T1.B	T2.B	C
1	10	10	100
1	10	15	150
1	10	20	200
2	20	10	100
2	20	15	150
2	20	20	200
3	30	10	100
3	30	15	150
3	30	20	200

```
SELECT *
FROM T1 INNER JOIN T2
ON T1.B = T2.B;
```

T1

A	B
1	10
2	20
3	30

T2

B	C
10	100
15	150
20	200

A	T1.B	T2.B	C
1	10	10	100
2	20	20	200

T1

A	B
1	10
2	20
3	30

T2

B	C
10	100
15	150
20	200

SELECT *
FROM T1 **RIGHT** OUTER JOIN **T2**
ON T1.B = T2.B

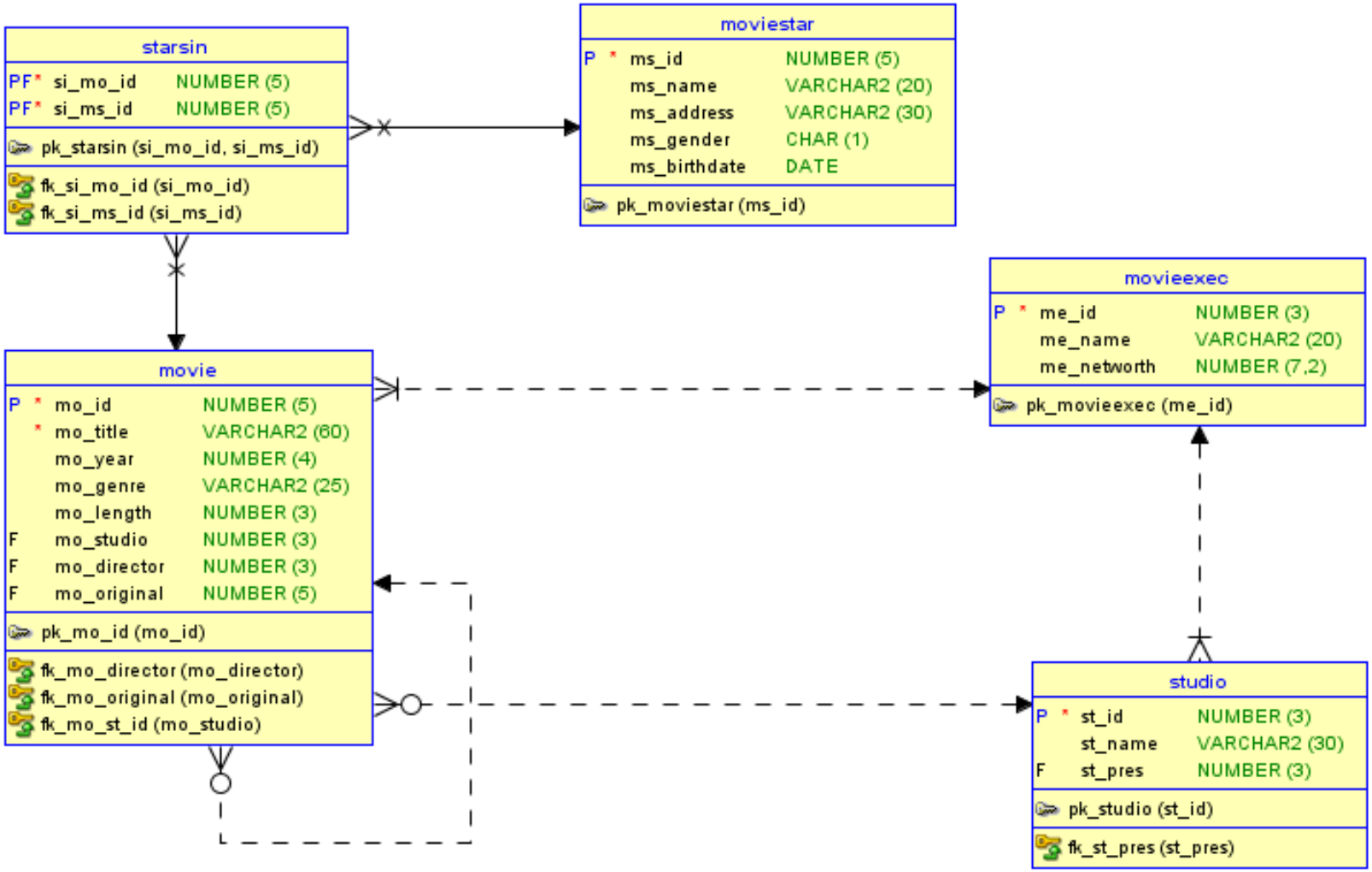
A	T1.B	T2.B	C
1	10	10	100
2	20	20	200
		15	**150**

SELECT *
FROM **T1 LEFT** OUTER JOIN T2
ON T1.B = T2.B

A	T1.B	T2.B	C
1	10	10	100
2	20	20	200
3	**30**		

SELECT *
FROM T1 **FULL** OUTER JOIN T2
ON T1.B = T2.B

A	T1.B	T2.B	C
1	10	10	100
2	20	20	200
3	**30**		
		15	**150**


starsin
PF* si_mo_id NUMBER (5)
PF* si_ms_id NUMBER (5)
pk_starsin (si_mo_id, si_ms_id)
fk_si_mo_id (si_mo_id)
fk_si_ms_id (si_ms_id)
moviestar
P * ms_id NUMBER (5)
ms_name VARCHAR2 (20)
ms_address VARCHAR2 (30)
ms_gender CHAR (1)
ms_birthdate DATE
pk_moviestar (ms_id)
movieexec
P * me_id NUMBER (3)
me_name VARCHAR2 (20)
me_networth NUMBER (7,2)
pk_movieexec (me_id)
movie
P * mo_id NUMBER (5)
* mo_title VARCHAR2 (60)
mo_year NUMBER (4)
mo_genre VARCHAR2 (25)
mo_length NUMBER (3)
F mo_studio NUMBER (3)
F mo_director NUMBER (3)
F mo_original NUMBER (5)
pk_mo_id (mo_id)
fk_mo_director (mo_director)
fk_mo_original (mo_original)
fk_mo_st_id (mo_studio)
studio
P * st_id NUMBER (3)
st_name VARCHAR2 (30)
F st_pres NUMBER (3)
pk_studio (st_id)
fk_st_pres (st_pres)

Jointure d'une table à elle-même

```
SELECT movie.mo_title, movie.mo_year,
       orig.mo_title AS "ORIGINAL TITLE", orig.mo_year
FROM movie INNER JOIN movie orig
  ON movie.mo_original = orig.mo_id
```

MO_TITLE	MO_YEAR	ORIGINAL TITLE	MO_YEAR
The Empire strikes back	1980	Star wars	1977
Indiana Jones and the temple of doom	1984	Indiana Jones and the raiders of the lost ark	1981
Indiana Jones and the last crusade	1989	Indiana Jones and the temple of doom	1984

3 rows returned in 0,02 seconds CSV Export

MO_ID	MO_TITLE	MO_YEAR	MO_ORIGINAL
1	12 angry men	1957	(null)
2	Gone with the wind	1939	(null)
3	Star wars	1977	(null)
10	The Empire strikes back	1980	3
4	Titanic	1997	(null)
5	Indiana Jones and the raiders of the lost ark	1981	(null)
6	Indiana Jones and the temple of doom	1984	5
7	Indiana Jones and the last crusade	1989	6
8	King Kong	2005	(null)
9	King Kong	1933	(null)

movie

Movie.mo_original

Orig.mo_id

orig

MO_ID	MO_TITLE	MO_YEAR	MO_ORIGINAL
1	12 angry men	1957	(null)
2	Gone with the wind	1939	(null)
3	Star wars	1977	(null)
10	The Empire strikes back	1980	3
4	Titanic	1997	(null)
5	Indiana Jones and the raiders of the lost ark	1981	(null)
6	Indiana Jones and the temple of doom	1984	5
7	Indiana Jones and the last crusade	1989	6
8	King Kong	2005	(null)
9	King Kong	1933	(null)

Dictionnaire de données

Dictionnaire de données

USER

- USER_TABLES
- USER_CONSTRAINTS
- USER_INDEXES
- …

ALL

- ALL_TABLES..
- Uniquement sur les objets pour lesquels vous avez un privilège

Dictionnaire de données

Accéder à la description de la table

- DESCRIBE
- DESC USER_TABLES

Interroger le dictionnaire de données

- SELECT COUNT(*)
- FROM user_tables;

PL/SQL

Blocs PL/SQL

3 types de blocs

- Anonymes
- Sous-programmes
  - Procédures
  - Fonctions
- Trigger

Structure commune

- Section déclaration
- Section instruction commençant par BEGIN, se terminant par END;
- Section gestion des exceptions

Blocs anonymes

Ne sont pas nommés

Ne sont pas stockés

Sont compilés à chaque fois qu'ils doivent être exécutés

Ne peuvent pas être appelés

Oracle Student Guide 1.3

Anonymous Blocks – Basic Structure

- Basic structure of an anonymous block:

```
[DECLARE]

BEGIN
  --statements

[EXCEPTION]

END;
```

- The DECLARE and EXCEPTION keywords/sections are optional.

```
DECLARE
   v_date_of_birth DATE;

BEGIN
   v_date_of_birth := SYSDATE ;

   DBMS_OUTPUT.PUT_LINE(v_date_of_birth);

END;
```

Sous-programmes

Nommés

Sont compilés une fois et sont stockés

Peuvent être réutilisés, partagés

Placés dans des packages

Procédures vs Fonctions

Procédures

```
CREATE [OR REPLACE] PROCEDURE procedure_name (Obligatoire)
[(parameter1 [mode] datatype1,...)]     (Optionnel)
IS|AS                                   (Obligatoire)
[local_variable_declarations; …]        (Optionnel)

BEGIN          (Obligatoire)
  SQL and PL/SQL statements;
  EXCEPTION (Optionnel)
   WHEN exception-handling actions;
END [name]; (Obligatoire)
```

```
CREATE OR REPLACE PROCEDURE add_tsh (
    p_tsh_name IN type_show_tsh.tsh_name%TYPE )
IS
BEGIN
    INSERT INTO type_show_tsh(tsh_name)
    VALUES(p_tsh_name);
END add_tsh;
/

BEGIN
  add_tsh('Cirque');
END;
/
```

Fonctions

```
CREATE [OR REPLACE] FUNCTION function_name
[(parameter1 [mode1] datatype1, ...)]
RETURN datatype
IS|AS
  [local_variable_declarations; …]
BEGIN
-- actions;
RETURN expression;
END [function_name];
```

```
14 CREATE OR REPLACE FUNCTION type_jour(p_date IN date)
15   RETURN VARCHAR2
16   IS
17   BEGIN
18   IF TO_CHAR(p_date, 'D') IN (6, 7) THEN
19     RETURN 'WE';
20   ELSIF TO_CHAR(p_date, 'D') BETWEEN 1 and 5 THEN
21         RETURN 'S';
22      ELSE
23         RETURN NULL;
24   END IF;
25 END type_jour;
26 /
```

SELECT bkg_id, bkg_total_seat, type_jour(bkg_date)
FROM booking_bkg;

SELECT type_jour(NULL) FROM dual;

Oracle Student Guide 2.7

Variable Case Conventions

- Case Conventions are shown below.
- The following table provides guidelines for writing code in uppercase and lowercase to help you distinguish keywords from named objects.

Category	Case Convention	Examples
SQL keywords	Uppercase	`SELECT, INSERT`
PL/SQL keywords	Uppercase	`DECLARE, BEGIN, IF`
Data types	Uppercase	`VARCHAR2, BOOLEAN`
Identifiers (variables, etc.)	Lowercase	`v_salary, emp_cursor, c_tax_rate, p_empno`
Tables and columns	Lowercase	`employees, dept_id, salary, hire_date`

ORACLE Academy

PLSQL S2L7
Good Programming Practices

10

Oracle Student Guide 2.7

Naming Conventions

- The naming of identifiers should be clear, consistent, and unambiguous.
- One commonly-used convention is to name:
  - Variables starting with v_
  - Constants starting with c_
  - Parameters starting with p_ (for passing to procedures and functions)

ORACLE Academy

Déclaration de variables

```
DECLARE
  v_mo_length NUMBER;
  v_star_addr VARCHAR2(50) DEFAULT  'Hollywood';
  v_price NUMBER(10,2) := 0;
  v_birthdate DATE := TO_DATE('25/10/2016', 'dd/mm/yyyy');
  v_picture BLOB;

  v_mo_genre movie.mo_genre%TYPE;
```

Attribut %TYPE

Evite certaines erreurs

- Erreurs de type, précision

Changement de type au niveau de la colonne

- Le code n'est pas à reprendre

```
v_tsh_name type_show_tsh.tsh_name%TYPE;
```

Affectation et conversions de type

Conversions implicites

Conversions explicites

- TO_CHAR(value, fmt)
- TO_NUMBER(value, fmt)
- TO_DATE(value, fmt)

```
24  -- Activation du package d'E/S
25  -- A faire une fois
26  SET SERVEROUTPUT ON
27
28  DECLARE
29    v_birthdate DATE;
30  BEGIN
31   v_birthdate := TO_DATE('04/10/2020', 'DD/MM/YYYY');
32   DBMS_OUTPUT.PUT_LINE(v_birthdate);
33   DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_birthdate, 'DD Month YYYY'));
34  END;
35  /
36
37
```

Sortie de script x | Résultat de requête x

Tâche terminée en 0,054 secondes

```
04/10/20
04 Octobre   2020


Procédure PL/SQL terminée.
```

Opérateurs

1. Concaténation : ||
2. IS NULL ou IS NOT NULL
3. LIKE
4. BETWEEN…AND…
5. AND, OR, NOT

CST_ID	CST_EMAIL	CST_LAST_NAME	CST_FIRST_NAME	CST_PHONE
1	rnelson@machin.fr	NELSON	Ryan	0683242254
2	barry.robichaux@truc.fr	ROBICHAUX	Barry	0228794613
3	cjones@machin.fr	JONES	Chandi	0659157846

SQL dans PL/SQL

Peuvent être utilisées

- Instructions de modification de données (LMD - DML)
- Instructions de contrôle des transactions (LCT - TCL)

Ne peuvent pas être utilisées directement

- Instructions de définition de données (LDD)
- Instructions de contrôle de données (LCD)

Instructions SELECT

- 1 ligne => SELECT…INTO
- Plusieurs lignes => curseur explicite

Instructions LMD

```
INSERT INTO show_shw (shw_ID, shw_title, shw_duration, shw_date, shw_tsh_ID)
VALUES (10, 'CALACAS - ZINGARO', '90', TO_DATE('28/09/2012', 'DD/MM/YYYY'),  1);
```

```
UPDATE show_shw
SET shw_desc = 'Fantastic, never seen before....'
WHERE shw_id = 10;
```

```
DELETE FROM type_show_tsh
WHERE tsh_id = 1;
```

Interaction avec la base de données

Colonnes | Données | Model | Contraintes | Droits | Statistiques | Déclenc

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	SHW_ID	NUMBER(20,0)	No
2	SHW_TITLE	VARCHAR2(50 BYTE)	Yes
3	SHW_DESC	VARCHAR2(255 BYTE)	Yes
4	SHW_DURATION	NUMBER(3,0)	Yes
5	SHW_TSH_ID	NUMBER(20,0)	Yes
6	SHW_DATE	DATE	Yes
7	SHW_PICT	BLOB	Yes

Colonnes | Données | Model | Contraintes | Droits | Statistique

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SI
1	FK_SHW_TSH_ID	Foreign_Key	(
2	PK_SHW	Primary_Key	(
3	UN_SHW_DATE	Unique	(

```
CREATE TABLE show_shw(
      shw_id NUMBER(20),
      shw_title VARCHAR2(50),
      shw_desc VARCHAR2(255),
      shw_duration NUMBER(3),
      shw_tsh_id NUMBER(20),
      shw_date DATE CONSTRAINT un_shw_date UNIQUE,
       CONSTRAINT pk_shw PRIMARY KEY(shw_id));
```

```
ALTER TABLE show_shw
ADD CONSTRAINT fk_shw_tsh_id
FOREIGN KEY(shw_tsh_id) REFERENCES type_show_tsh(tsh_id);
```

Récupérer une ligne – SELECT … INTO

```
48  DECLARE
49    v_shw_title show_shw.shw_title%TYPE;
50    v_shw_date show_shw.shw_date%TYPE;
51  BEGIN
52    SELECT shw_title, shw_date
53     INTO v_shw_title, v_shw_date
54     FROM show_shw
55     WHERE shw_id = 10;
56    DBMS_OUTPUT.PUT_LINE('Show : ' || v_shw_title || ' ' || v_shw_date);
57  END;
58  /
59
60
```

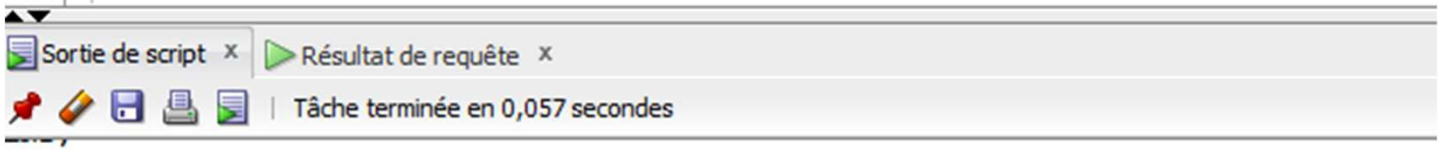
Sortie de script x | Résultat de requête x

Tâche terminée en 2,958 secondes

```
Show : CALACAS - ZINGARO 28/09/18


Procédure PL/SQL terminée.
```

```
48  DECLARE
49    v_shw_title show_shw.shw_title%TYPE;
50    v_shw_date show_shw.shw_date%TYPE;
51  BEGIN
52    SELECT shw_title, shw_date
53     INTO v_shw_title, v_shw_date
54     FROM show_shw;
55    DBMS_OUTPUT.PUT_LINE('Show : ' || v_shw_title || ' ' || v_shw_date);
56  END;
57  /
58
```




```
Rapport d'erreur -
ORA-01422: l'extraction exacte ramène plus que le nombre de lignes demandé
ORA-06512: à ligne 5
01422. 00000 -  "exact fetch returns more than requested number of rows"
*Cause:    The number specified in exact fetch is less than the rows returned.
*Action:   Rewrite the query or change number of rows requested
```

Curseur implicite

Un curseur est :

- Un pointeur vers une zone mémoire dans laquelle Oracle stocke les données traitées par une instructions SQL
- Un nom pour cette zone mémoire

Les curseurs implicites sont définis automatiquement

- Pour toutes les instructions LMD (INSERT, UPDATE, DELETE, MERGE)
- Pour les instructions SELECT (une ligne)

Ils sont désignés par « SQL »

Attributs des curseurs

%ROWCOUNT

- Retourne le nombre de lignes traitées

%FOUND

- TRUE si INSERT, UPDATE, DELETE,MERGE ont traité au moins une ligne

%NOTFOUND

- L'opposé de %FOUND

SQL%ROWCOUNT

```
1  SET SERVEROUTPUT ON
2  SET VERIFY OFF
3
4  DECLARE
5     v_bkg_id booking_bkg.bkg_id%TYPE := &id;
6  BEGIN
7     DELETE FROM booking_bkg
8      WHERE bkg_id = v_bkg_id;
9     IF (SQL%NOTFOUND) THEN
10     DBMS_OUTPUT.PUT_LINE('no row deleted');
11    ELSE
12        DBMS_OUTPUT.PUT_LINE(TO_CHAR(SQL%ROWCOUNT) || ' rows deleted');
13    END IF;
14 END;
15 /
```

Sortie de script x

Tâche terminée en 2,814 secondes

```
1 rows deleted


Procédure PL/SQL terminée.
```

Fonctions

Fonctions Mono-lignes

Chaines de caractères : UPPER, LOWER, INITCAP…

- Nombres : ROUND, TRUNC
- Dates : SYSDATE, CURRENT_DATE, EXTRACT… FROM, …

Fonctions de conversion

- TO_DATE
- TO_CHAR

Fonctions dédiées à la valeur NULL

- NVL, NVL2
- Expression conditionnelles
  - CASE…WHEN

Fonctions multi-lignes (de groupe)

COUNT

AVG

SUM

MAX, MIN

STDDEV

VARIANCE

Egalement

Opérateurs

- Arithmétiques
- Concaténation
  - ||

Constantes