

# Description d'un schéma de bases de données

Suite du cours 4



# CONTRAINTES D'INTEGRITE

- Contrainte de non nullité
- Contrainte de clé
  - *Contrainte de clé primaire*
  - *Contrainte référentielles ou contrainte de clé étrangère*
  - *Contrainte de clé candidate*
- Contrainte de domaine
- Contrainte intra-attribut
- Contrainte inter-attributs
- Contrainte intra-relation
  - *Contraintes intra-lignes*
  - *Contraintes inter-lignes*
- Contrainte inter-relation

# DESCRIPTION D 'UN ATTRIBUT

nomAttribut      type    [DEFAULT expression]    [**Contrainte**]



NOT NULL  
UNIQUE  
PRIMARY KEY  
REFERENCES nomtable (listeAttributs)  
**CHECK (condition)**



- *Contrainte de domaine*
- *Contrainte intra-attribut*
- *Contrainte intra-ligne ou inter-ligne*
- *Contrainte intra-relation*

# SYNTAXE DU CHECK

## CHECK (Condition)



nomattribut **IN** (cste1, cste2, ....)

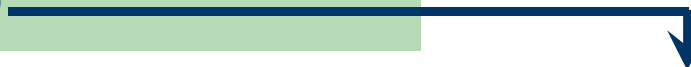
[**NOT**] Prédicat1 [**AND/OR** prédicat2].....

nomattribut **BETWEEN** cste1 **AND** cste2

nomattribut **IN** (requête)



- *Contrainte de domaine*
- *Contrainte intra-ligne*
- *Contrainte intra-attribut*



- *Contrainte de domaine*
- *Contrainte inter-ligne*
- *Contrainte intra-relation*

# SYNTAXE DU CHECK (suite)

CREATE TABLE Relation  
( description des attributs,

**CHECK** (condition sur 1 ou +ieurs attributs), →

- *Contrainte de domaine*
- *Contrainte intra-attribut*
- *Contrainte inter-attribut*
- *Contrainte intra-ligne*
- *Contrainte inter-ligne*
- *Contrainte intra-relation*

CONSTRAINT NomContrainte **CHECK** (condition sur +ieurs attributs)  
);



**ORACLE NE SUPPORTE QUE LES CHECK INTRA-LIGNE**

# EXEMPLE DE CHECK

**CHAMBRE** (NumChambre, Prix, NbrLits, NbrPers, Confort, Équipement)

CREATE TABLE CHAMBRE

( NumChambre NUMBER (3) PRIMARY KEY,

.....

Confort VARCHAR2 (10)

**CHECK (UPPER(Confort) IN ( ' BAIN ', ' DOUCHE '))**

);

# EXEMPLE DE CHECK

**RESERVATION** ( NumClient, NumChambre, DateArrivée, DateDépart)

CREATE TABLE RESERVATION

( NumClient NUMBER (3) REFERENCES CLIENT,  
NumChambre NUMBER (3) REFERENCES CHAMBRE,  
DateArrivee DATE DEFAULT SYSDATE,  
DateDepart DATE,  
PRIMARY KEY (NumClient, NumChambre),  
**CHECK (DateArrivee >= SYSDATE)**  
);

# EXEMPLE DE CHECK

**RESERVATION** ( NumClient, NumChambre, DateArrivée, DateDépart)

CREATE TABLE RESERVATION

( NumClient NUMBER (3) REFERENCES CLIENT,  
NumChambre NUMBER (3) REFERENCES CHAMBRE,  
DateArrivee DATE DEFAULT SYSDATE,  
DateDepart DATE,  
PRIMARY KEY (NumClient, NumChambre),

**CONSTRAINT C1-RES CHECK (DateArrivee >= SYSDATE)**

**CONSTRAINT C1-RES CHECK (DateDepart >= DateArrivee)**

);



# EXEMPLE DE CHECK

Les chambres dont le prix est  $\leq 20$  sont toutes des chambres à 1 lit.

**CHAMBRE** (NumChambre, Prix, NbrLits, NbrPers, Confort, Equipement)

CREATE TABLE CHAMBRE

( NumChambre NUMBER (3) PRIMARY KEY,

Prix NUMBER (5, 2)

.....

**CHECK (Prix > 20 OR NbrLits = 1)**

);

$\text{Prix} \leq 20 \Rightarrow \text{NbrLits} = 1$

$\Leftrightarrow$

$\text{Prix} > 20 \vee \text{NbrLits} = 1$

# EXEMPLE DE CHECK

Le prix d'une chambre ne peut pas dépasser le prix moyen de plus de 40 €

**CHAMBRE** (NumChambre, Prix, NbrLits, NbrPers, Confort, Équipement)

CREATE TABLE CHAMBRE

( NumChambre NUMBER (3) PRIMARY KEY,

Prix NUMBER (5, 2),

.....

CHECK (Prix- 40 <= (SELECT AVG(Prix) FROM CHAMBRE) )

);



**ORACLE NE PERMET PAS CETTE ECRITURE**

## REMARQUE

**LE CHECK N'EST VERIFIE QUE POUR LA LIGNE QUI EST INSEREE OU MODIFIEE DANS LA TABLE OÙ LE CHECK EST DEFINI.**



**SI UNE INSERTION OU MODIFICATION ENTRAINE UNE VIOLATION DE LA CONTRAINTESUR UNE AUTRE LIGNE QUE CELLE INSEREE OU MODIFIEE LA VIOLATION NE SERA PAS BLOQUEE PAR LE CHECK**



**UNE INSERTION QUI VIOLERAIT LA CONTRAINTES PASSERAIT INAPERÇUE**

# EXEMPLE

Etat initial

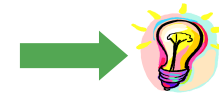
	prix
Ch1	60
Ch2	90

Ajout Ch3 avec un prix de 50?

Possible car :

- $AVG(prix) = 66,7$
- $50 - 40 \leq 66,7$

	prix
Ch1	60
Ch2	90
Ch3	50



*Pas de violation de contraintes pour les autres tuples*

Ajout Ch4 avec un prix de 130?

Impossible car

- $AVG(prix) = 82,5$
- $130 - 40 > 82,5$

	prix
Ch1	60
Ch2	90
Ch3	50

## EXEMPLE

**Baisser le prix de Ch3 à 10**

**Possible car :**

- $AVG(prix) = 53,33$
- $10 - 40 \leq 53,33$

	prix
Ch1	60
Ch2	90
Ch3	10



*Pas de violation de contraintes pour les autres tuples*

**Baisser le prix de Ch1 à 30**

**Possible car :**

- $AVG(prix) = 43,33$
- $30 - 40 \leq 43,33$

	prix
Ch1	30
Ch2	90
Ch3	10



*Violation de la contrainte pour Ch2*

**L'utilisation du check n'est donc pas appropriée dans cet exemple**



**Utiliser dans ce cas là les assertions si le SGBD le permet.**

# LES ASSERTIONS

```
CREATE TABLE CHAMBRE  
  ( NumChambre NUMBER (3) PRIMARY KEY,  
    Prix  NUMBER (5, 2)  
    .....  
  );
```

```
CREATE ASSERTION NomAssertion CHECK
```

```
(Prix- 40 <= (SELECT AVG(Prix) from CHAMBRE) );
```



**ORACLE ne permet pas les assertions mais permet  
l'utilisation des déclencheurs**

# DECLENCHEUR (TRIGGER)

- Définition
  - C'est une opération qui est déclenchée automatiquement par le SGBD suite à certains événements prédéterminés
- Quand l'utiliser?
  - Lorsque une contrainte d'intégrité ne peut être définie par les autres clause que supporte le SGBD (ex : ne pas utiliser un trigger pour maintenir une contrainte de clé candidate si le SGBD supporte la clause UNIQUE)

# DECLENCHEUR (TRIGGER)

- Un trigger est déclenché par un événement de modification d'une table tel que INSERT, DELETE ou UPDATE
- 
- Un trigger peut être exécuté avant ou après événement
- Suite à une opération, le trigger peut être exécuté une fois pour chaque ligne de la table touchée par l'opération de mise à jour
- Le déclenchement d'un trigger peut dépendre de l'évaluation d'une condition. Cette condition dans le cas d'oracle ne peut pas contenir de SELECT
- Un trigger déclenché exécutera une série d'action.
- Dans un trigger, on peut avoir à manipuler l'ancienne valeur d'un attribut et sa nouvelle valeur. On prefixera par **:old** un attribut pour désigner son ancienne valeur et par **:new** pour désigner sa nouvelle valeur.



# DECLENCHEUR (TRIGGER)

```
CREATE TRIGGER NomDéclencheur  
<quand> <évènement> ON NomTable  
[FOR EACH ROW [WHEN <condition>]]  
BEGIN  
<ListeActions>  
END;  
/
```

- Quand :BEFORE, AFTER
- évènement : DELETE, UPDATE ou INSERT séparés par des OR

## Exemple

- Lorsqu'une augmentation du prix d'une chambre est tentée, il faut limiter l'augmentation à 10% du prix en cours

```
CREATE TRIGGER VerifNouveauPrix  
BEFORE UPDATE ON CHAMBRE  
FOR EACH ROW WHEN (new.Prix > old.Prix*1.1)  
BEGIN  
    :new.prix = :old.prix*1.1  
END;  
/
```

## Exemple 2

**CLIENT** (NumClient, Nom, Prénom, Adresse, Tel, Type)

**Pas de réservation pour un mauvais client**

Type ∈ {M(auvais) ou B(on) }.

```
CREATE TRIGGER VERIFIER-CLIENT
BEFORE INSERT ON RESERVATION
DECLARE MB VARCHAR;
BEGIN
    SELECT Type INTO MB FROM CLIENT C
    WHERE C.NumClient= :New.Numclient;
    IF MB = ' M '
    THEN BEGIN
        RAISE_APPLICATION_ERROR(20000, ' Pas de réservation pour ce client ' );
        ROLLBACK WORK;
    END IF;
END;
/
```