

TD N° 3 : Type Date/Contraintes d'intégrité/Contrôle de données

Partie1 : Types Dates et heures

Principales fonctions pour les dates :

Fonctions pour les dates :		
Fonction	Objectif	Retour
ADDTIME(<i>date1, date2</i>)	Ajoute les deux dates avec <i>date1</i> TIME ou DATETIME et <i>date2</i> TIME.	TIME ou DATETIME
ADDDATE(<i>date, n</i>) CURDATE(), CURRENT_DATE Ou CURRENT_DATE()	Ajoute <i>n</i> jours à une date (heure). Date courante ('YYYY-MM-DD' ou 'YYYYMMDD').	DATE ou DATETIME INT ou DATE
CURTIME(), CURRENT_TIME ou CURRENT_TIME()	Heure courante ('HH:MM:SS' or 'HHMMSS')	INT ou DATE
CURRENT_TIMESTAMP, CURRENT_TIMESTAMP() ou NOW()	Date et heure courantes ('YYYY-MM-DD HH:MM:SS' ou 'YYYYMMDDHHMMSS').	INT ou DATETIME
DATE(<i>date</i>) DATEDIFF(<i>date1, date2</i>) DATE_ADD(<i>date, INTERVAL expr type</i>)	Extrait une date à partir d'une expression de type DATETIME Nombre entier de jours entre les 2 dates. Ajoute un intervalle à une date (heure). <i>expr</i> désigne un intervalle. <i>type</i> indique comment interpréter le format de l'expression (voir tableau ci-contre).	DATE INT DATE ou DATETIME
DATE_FORMAT(<i>date, format</i>)	Présente la date selon un format (voir tableau ci-contre).	VARCHAR
DATE_SUB(<i>date, INTERVAL expr type</i>)	Soustrait un intervalle à une date (heure) Mêmes paramètres que DATE_ADD.	DATE ou DATETIME
DAYNAME(<i>date</i>) DAY(<i>date</i>) ou DAYOFMONTH(<i>date</i>) DAYOFYEAR(<i>date</i>) EXTRACT(<i>type FROM date</i>)	Nom du jour en anglais. Numéro du jour dans le mois (0 à 31). Numéro du jour dans l'année (0 à 366). Extrait une partie d'une date selon un type d'intervalle (comme pour DATE_ADD).	VARCHAR INT INT INT
FROM_DAYS(<i>n</i>)	Retourne une date à partir d'un nombre de jours (le calendrier année 0 débute à <i>n=365</i>).	DATE
FROM_UNIXTIME(<i>unix[, format]</i>)	Retourne une date (heure) à partir d'une estampille Unix (nombre de jours depuis le 1/1/1970). Utilisation possible d'un format.	INT ou DATETIME
HOUR(<i>time</i>) LAST_DAY(<i>date</i>) LOCALTIME, LOCALTIME(), LOCALTIMESTAMP, LOCALTIMESTAMP()	Extrait l'heure d'un temps. Dernier jour du mois d'une date (heure). Synonymes de NOW().	INT DATE
MAKEDATE(<i>annee, njour</i>)	Construit une date à partir d'une année et d'un nombre de jours (>0, si <i>njour</i> >365, l'année s'incrémente automatiquement).	DATE
MAKETIME(<i>heure, minute, seconde</i>) MICROSECOND(<i>date</i>) MINUTE(<i>time</i>) MONTH(<i>date</i>), MONTHNAME(<i>date</i>)	Construit une heure. Extrait les microsecondes d'une dateheure. Extrait les minutes d'un temps. Retourne respectivement le numéro et le nom du mois d'une date-heure.	TIME INT INT INT, VARCHAR
NOW()	Date et heure courantes au format 'YYYYMM-DD HH:MM:SS' ou 'YYYYMMDDHHMMSS'.	DATETIME ou INT
PERIOD_DIFF(<i>int1, int2</i>)	Nombre de mois séparant les deux dates au format 'YMM' ou 'YYYYMM'.	INT
SECOND(<i>time</i>) SEC_TO_TIME(<i>secondes</i>) STR_TO_DATE(<i>c, format</i>)	Extrait les secondes d'un temps. Construit une heure au format 'HH:MM:SS' ou 'HHMMSS'. Construit une date (heure) selon un certain format. C'est l'inverse de DATE_FORMAT().	INT TIME ou INT DATE ou DATETIME Ou TIME
SUBDATE(<i>date, n</i>) SUBTIME(<i>date1, date2</i>) SYSDATE()	Retranche <i>n</i> jours à une date (heure). Retranche <i>date2</i> (TIME) à <i>date1</i> (TIME ou DATETIME). Date et heure courantes au format 'YYYYMM-DD HH:MM:SS' ou 'YYYYMMDDHHMMSS' (différence avec NOW voir chapitre 1).	DATE ou DATETIME TIME ou DATETIME DATETIME ou INT
TIME(<i>datetime</i>)	Extrait le temps d'une date-heure.	TIME
TIMEDIFF(<i>date1, tdate2</i>)	Temps entre 2 temps ou 2 dates ou 2 dates-heure.	TIME
TIMESTAMP(<i>date</i>)	Construit une estampille à partir d'une date(heure).	TIMESTAMP
TIMESTAMPADD(<i>intervalle, int, date</i>)	Ajoute à la date (heure) un intervalle (<i>int</i>) du type FRAC_SECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, ou YEAR.	TIMESTAMP
TIMESTAMPDIFF(<i>intervalle, int, date</i>)	Retranche à la date (heure) un intervalle du type (idem précédent).	TIMESTAMP
TIME_TO_SEC(<i>time</i>) TO_DAYS(<i>date</i>)	Retourne le nombre de secondes équivalent au temps. Retourne un nombre de jours à partir d'une date ('YYYY-MM-DD' ou 'YYYYMMDD'). Inverse de FROM_DAYS().	INT INT
UNIX_TIMESTAMP(<i>date</i>)	Retourne le nombre de secondes depuis le 1/1/1970 jusqu'à la date (heure) passée en paramètre (ou entier au format 'YYYYMMDD' ou 'YYYYMMDD'). Inverse de FROM_UNIXTIME().	INT
UTC_DATE(), UTC_TIME(), UTC_TIMESTAMP()	Retournent respectivement la date, l'heure et l'estampille au méridien de Greenwich.	DATE, TIME, DATETIME
WEEKDAY(<i>date</i>)	Numéro du jour (0 : <i>lundi</i> , 1 : <i>mardi</i> , ... 6 : <i>dimanche</i>) d'une date (heure).	INT
WEEKOFYEAR(<i>date</i>)	Numéro de la semaine en cours (1 à 53).	INT

Types de données dates et heures :		
Type	Description	Commentaire
DATE	Dates du 1er janvier de l'an 1000 au 31 décembre 9999 après J.-C.	Sur 3 octets. L'affichage est au format 'YYYY-MM-DD'.
DATETIME	Dates et heures (de 0 h de la première date à 23 h 59 minutes 59 secondes de la dernière date).	Sur 8 octets. L'affichage est au format 'YYYY-MM-DD HH:MM:SS'.
YEAR[(2 4)]	Sur 4 positions : de 1901 à 2155 (incluant 0000). Sur 2 positions : de 70 à 69 (désignant 1970 à 2069).	Sur 1 octet : l'année est considérée sur 2 ou 4 positions (4 par défaut). Le format d'affichage est 'YYYY'.
TIME	Heures de -838 h 59 minutes 59 secondes à 838 h 59 minutes 59 secondes.	L'heure au format 'HH:MM:SS' sur 3 octets
TIMESTAMP	Instants du 1er Janvier 1970 0 h 0 minute 0 seconde à l'année 2037.	Estampille sur 4 octets (au format 'YYYY-MM-DD HH:MM:SS') : mise à jour à chaque modification sur la table.

Principaux formats pour les fonctions DATE_FORMAT et STR_TO_DATE :

Format	Description
%a	Nom du jour en anglais abrégé (Sun..Sat)
%b	Nom du mois en anglais abrégé (Jan..Dec)
%c	Mois (0..12)
%e	Jour du mois (0..31)
%f	Microsecondes (000000..999999)
%H	Heures (00..23)
%i	Minutes (00..59)
%j	Jour de l'année (001..366)
%M	Nom du mois en anglais (January..December)
%s	Secondes (00..59)
%T	Time sur 24 heures (hh:mm:ss)
%u	Numéro de semaine (00..53)
%W	Nom du jour en anglais (Sunday..Saturday)
%w	Jour de la semaine (0=Sunday..6=Saturday)
%Y	Année sur 4 positions

Paramètres d'intervalles pour les fonctions DATE_ADD et DATE_SUB :

Paramètre type	Paramètre expr
MICROSECOND	n
SECOND	n
MINUTE	n
HOUR	nn
DAY	nn
WEEK	n
MONTH	nn
YEAR	nnnn
SECOND_MICROSECOND	'ss.microsec'
MINUTE_MICROSECOND	'mi.microsec'
MINUTE_SECOND	'mi:ss'
HOUR_MICROSECOND	'hh.microsec'
HOUR_SECOND	'hh:mi:ss'
HOUR_MINUTE	'hh:mi'
DAY_MICROSECOND	'dd.microsec'
DAY_SECOND	'dd hh:mi:ss'
DAY_MINUTE	'dd hh:mi'
DAY_HOUR	'dd hh'
YEAR_MONTH	'yyyy-mm'

Opérations entre colonnes date-heure :

Opérande 1	Opérateur	Opérande2	Résultat
DATE DATETIME	+ ou -	Interval	DATE DATETIME
DATE DATETIME	+ ou -	INTEGER	DATE DATETIME
TIME	+ ou -	TIME	TIME
TIME	+ ou -	INTEGER	TIME

Fonctions utiles

Fonction	Description
• GET_FORMAT(DATE TIME) DATE, TIME, 'EUR' 'USA' 'JIS' 'ISO' 'INTERNAL')	retourne un format de date (heure).
• TIME(expression)	extrait d'une date (heure) un type TIME.
• TIME_FORMAT(time, format)	convertit un intervalle suivant un certain format.

Note : Les fonctions NOW() et SYSDATE() retournent la date et l'heure courantes. Dans une procédure ou un déclencheur SYSDATE est réévaluée en temps réel, alors que NOW désignera toujours l'instant de début de traitement.

Exercice 1 : (en SQL)

1. Afficher la date dans 31 jours
2. Ajouter et afficher 1 jour et 1 microseconde après le 9/11/2010, 22 heures
3. Afficher la date de RDV dans 4 mois
4. Afficher la date de rendez-vous dans 7 jours, 1 heure et 30 minutes.
5. Afficher la date d'aujourd'hui en anglais
6. Extraire et afficher au format numérique du jour, heures et minutes

Exercice 2 :

Considérons la table suivante :

```
CREATE TABLE Pilote(  
    brevet          VARCHAR(6),  
    nom             VARCHAR(20),  
    dateNaiss       DATETIME,  
    dernierVol      DATE,  
    dateEmbauche    DATETIME,  
    prochainVolControle DATETIME,  
    nombreJoursNaisBoulot INTEGER(5),  
    intervalleNaisBoulot Decimal (20,6),  
    intervalleVolExterieur Decimal (10,6),  
    intervalleEntreVols Decimal (10,6),  
    intervalleEmbaucheControle TIME,  
    NbHvol          Decimal(7,2),  
    grade           VARCHAR(5),  
    compagnie       VARCHAR(4),  
    CONSTRAINT pk_Pilote PRIMARY KEY(brevet));
```

- 1) Insérer le pilote 'Boutef' de la compagnie de code 'AH' et initialiser
 - a. Le brevet à PL-1
 - b. sa date de naissance au 25 mars 1940
 - c. sa date de son dernier vol au 30 octobre 2005
 - d. sa date d'embauche à celle du jour
 - e. et la date de son prochain contrôle en vol au 13 décembre 2010, 15h30
- 2) Sachant que le pilote 'Boutef' est né à 12h35. Modifier sa date de naissance
- 3) Modifier la date d'embauche de 10 après la semaine prochaine.
- 4) Alimenter la colonne 'nombreJoursNaisBoulot' du pilote 'Boutef'. Elle représente le nombre de jours depuis sa date de naissance jusqu'à sa date d'embauche.
- 5) Renseigner les colonnes suivantes à la décimale près :
 - a. intervalleNaisBoulot : Intervalle entre la date de naissance à la date d'embauche
 - b. intervalleEntreVols : Intervalle entre le dernier vol et le prochain vol de contrôle
 - c. intervalleVolExterieu : Intervalle entre le dernier vol et la date d'embauche
- 6) Renseigner la colonne 'intervalleEmbaucheControle' qui représente l'intervalle (en TIME) entre la date d'embauche et le prochain vol de contrôle.

Exercice 3 :

- 1) Afficher le Numéro du jour de l'année
- 2) Afficher les libellés des jours et des mois en anglais par défaut
- 3) Afficher seulement le jour de la date d'embauche
- 4) Afficher le mois de naissance du pilote 'Boutef'
- 5) Afficher le nombre de jour qui se sont écoulé depuis le 1 janvier 1970

Partie 2 : Contraintes d'intégrités et Vues

Rappel :

Dans les bases de données (BD) les contraintes d'intégrité (CI) se déclinent en deux types :

1. Les CI relationnelles : Domaine, Clé primaire et clé étrangère (intégrité de référence)
2. Les CI applicatives (sémantiques), comme par exemple « dans une même journée, l'heure de départ d'un vol doit être inférieure à l'heure d'arrivée »

L'objectif ici est de simuler les CI relationnelles avec le concept de vue.

Etant donnée le schéma relationnel « AIRBASE » suivant :

Pilote (plnum, plnom, plprenom, ville, salaire)

Avion (avnum, avnom, capacité, localisation)

Vol (volnum, plnum, avnum, villedep, villearr, heuredep, heurearr)

Exercice 1 : Simulation de la contrainte de domaine.

Dans la BD « AIRBASE » :

- La colonne 'Avion.avnom' prend ses valeurs dans la liste : A300, A310, A320, B707, B727, B747
- La colonne 'Avion.capacité' prend ses valeurs dans l'intervalle 100 à 400

Utiliser la notion de vue pour écrire en SQL la commande permettant de garantir la cohérence des données en cas d'insertion dans la table Avion. Expliquer

Exercice 2 : Simulation de l'intégrité référentielle en insertion dans Vol

Pour assurer la cohérence de la table Vol dans le cas d'insertion, il faut garantir la vérification des contraintes de référence des colonnes plnum et avnum de la table Vol sur respectivement les colonnes Pilote.plnum et Avion.avnum.

Proposer en SQL une solution garantissant cette cohérence en utilisant la notion de vue

Exercice 3 : Simulation de l'intégrité référentielle en suppression dans Pilote

Pour assurer la cohérence de la table Pilote en cas de suppression, il faut garantir la vérification de la contrainte de référence de la colonne Vol.plnum vers la colonne Pilote.plnum.

Proposer en SQL une solution garantissant cette cohérence en cas de suppression d'un pilote.

Exercice 4 : Simulation de l'intégrité référentielle en suppression dans Avion

Même question que dans l'exercice 3 concernant la table Avion.

Partie 3 : Confidentialité (Vues et privilèges)

Rappel :

La sécurité est liée à l'intégrité et la confidentialité. Dans la partie précédente on a traité l'intégrité relationnelle. La confidentialité permet de s'assurer que chaque utilisateur effectue des accès licites aux données. Les mécanismes de contrôle permettant cela sont les vues et l'opérateur d'octroi de privilèges (GRANT). Pour exécuter un ordre SQL, l'utilisateur doit en détenir le privilège.

L'accord de privilèges se fait par la commande GRANT qui a un double rôle :

- Ajouter des utilisateurs,
- Accorder des privilèges sur les des tables ou sur des vues.

La syntaxe de cette commande est :

Syntaxe des commandes GRANT et REVOKE:	Privilèges principaux pour GRANT et REVOKE:																																						
<pre>GRANT privilège [(col1 [, col2...])] [,privilège2 ...] ON [{TABLE FUNCTION PROCEDURE}] {nomTable * *.* nomBase.*} TO utilisateur [IDENTIFIED BY [PASSWORD] 'password'] [,utilisateur2 ...] [WITH [GRANT OPTION] [MAX_QUERIES_PER_HOUR nb] [MAX_UPDATES_PER_HOUR nb2] [MAX_CONNECTIONS_PER_HOUR nb3] [MAX_USER_CONNECTIONS nb4]];</pre> <p>Où :</p> <p>privilège : description du privilège (ex : SELECT, DELETE, etc.),</p> <p>col : précise la ou les colonnes sur lesquelles se portent les privilèges SELECT, INSERT ou UPDATE</p> <p>GRANT OPTION : permet de donner le droit de retransmettre les privilèges reçus à une tierce personne.</p> <pre>REVOKE privilège [(col1 [, col2...])] [,privilège2 ...] ON [{TABLE FUNCTION PROCEDURE}] {nomTable * *.* nomBase.*} FROM utilisateur [,utilisateur2 ...];</pre>	<table><thead><tr><th>privilège</th><th>Commentaire</th></tr></thead><tbody><tr><td>ALL (PRIVILEGES)</td><td>Tous les privilèges.</td></tr><tr><td>ALTER</td><td>Modification de base/table.</td></tr><tr><td>ALTER ROUTINE</td><td>Modification de procédure.</td></tr><tr><td>CREATE</td><td>Création de base/table.</td></tr><tr><td>CREATE ROUTINE</td><td>Création de procédure.</td></tr><tr><td>CREATE USER</td><td>Création d'utilisateur.</td></tr><tr><td>CREATE VIEW</td><td>Création de vue.</td></tr><tr><td>DELETE</td><td>Suppression de données de table.</td></tr><tr><td>DROP</td><td>Suppression de base/table.</td></tr><tr><td>EXECUTE</td><td>Exécution de procédure.</td></tr><tr><td>INDEX</td><td>Création/Suppression d'index.</td></tr><tr><td>INSERT</td><td>Insertion de données de table.</td></tr><tr><td>SELECT</td><td>Extraction de données de table.</td></tr><tr><td>SHOW DATABASES</td><td>Lister les bases.</td></tr><tr><td>SHOW VIEW</td><td>Lister les vues d'une base.</td></tr><tr><td>SUPER</td><td>Gestion des déclencheurs.</td></tr><tr><td>UPDATE</td><td>Modification de données de table.</td></tr><tr><td>USAGE</td><td>Synonyme de « sans privilège », USAGE est utilisé pour conserver les privilèges précédemment définis tout en les restreignant avec des options.</td></tr></tbody></table>	privilège	Commentaire	ALL (PRIVILEGES)	Tous les privilèges.	ALTER	Modification de base/table.	ALTER ROUTINE	Modification de procédure.	CREATE	Création de base/table.	CREATE ROUTINE	Création de procédure.	CREATE USER	Création d'utilisateur.	CREATE VIEW	Création de vue.	DELETE	Suppression de données de table.	DROP	Suppression de base/table.	EXECUTE	Exécution de procédure.	INDEX	Création/Suppression d'index.	INSERT	Insertion de données de table.	SELECT	Extraction de données de table.	SHOW DATABASES	Lister les bases.	SHOW VIEW	Lister les vues d'une base.	SUPER	Gestion des déclencheurs.	UPDATE	Modification de données de table.	USAGE	Synonyme de « sans privilège », USAGE est utilisé pour conserver les privilèges précédemment définis tout en les restreignant avec des options.
privilège	Commentaire																																						
ALL (PRIVILEGES)	Tous les privilèges.																																						
ALTER	Modification de base/table.																																						
ALTER ROUTINE	Modification de procédure.																																						
CREATE	Création de base/table.																																						
CREATE ROUTINE	Création de procédure.																																						
CREATE USER	Création d'utilisateur.																																						
CREATE VIEW	Création de vue.																																						
DELETE	Suppression de données de table.																																						
DROP	Suppression de base/table.																																						
EXECUTE	Exécution de procédure.																																						
INDEX	Création/Suppression d'index.																																						
INSERT	Insertion de données de table.																																						
SELECT	Extraction de données de table.																																						
SHOW DATABASES	Lister les bases.																																						
SHOW VIEW	Lister les vues d'une base.																																						
SUPER	Gestion des déclencheurs.																																						
UPDATE	Modification de données de table.																																						
USAGE	Synonyme de « sans privilège », USAGE est utilisé pour conserver les privilèges précédemment définis tout en les restreignant avec des options.																																						

Considérons le schéma de relations 'AIRBASE' :

Exercice 1 :

1. Ajouter un nouvel utilisateur « BOUTEF », qui doit avoir le mot de passe « passe », avec la permission de créer des bases et des tables
2. Ajouter le droit de modification de la colonne plnum de la table Pilote à l'utilisateur « BOUTEF » et donner lui le droit de transmettre se privilège à « OUYAHIA » (donner les deux ordres)
3. Ajouter à « BOUTEF » le privilège de créer ou supprimer des tables dans la base BD1.
4. Supprimer à 'BOUTEF' les droits de modifier la structure, d'insérer et modifier la colonne 'Salaire' de la table 'Pilote' contenue dans la base 'AIRBASE'.

Exercice 2 : (En Sql)

Comment peut-on masquer le salaire de la table Pilote à l'utilisateur 'TOTO'? De plus, on souhaite limiter ses accès aux pilotes d'ALGER. On supposera qu'il possède actuellement du privilège SELECT sur cette table.

Exercice 3 :

Considérant la table pilote de l'exercice 2 (partie1). Et les règles de gestions suivantes :

- ne peut être commandant de bord (CDB) qu'à la condition qu'il ait entre 1 000 et 4 000 heures de vol ;
- ne peut être copilote (COP) qu'à la condition qu'il ait entre 100 et 1 000 heures de vol ;
- ne peut être instructeur (INST) qu'à partir de 3 000 heures de vol.

Proposer deux solutions permettant de prendre en compte ces règles dans la BD.

Exercice 4 :

1. Restreindre l'accès à la table Vol entre 10h00 et 18h00.
2. Restreindre l'accès à la table Pilote à l'utilisateur 'TOTO' en utilisant la notion de vue