

2SGBD

Library Management System - Constraints, Event Triggers, and Audit

Documentation Technique



Réalisé par :

Mohamed Marwen Meddeb – marwen.meddeb@supinfo.com

Table des matières

I.	Part A.....	3
1.	Entités.....	3
2.	Creation des tables.....	4
3.	Constraints	5
II.	Part B	5
1.	Event Trigger for Borrowing:.....	5
2.	Event Trigger for Returning:.....	5
3.	Test:	6
III.	Part C	7
1.	Event Trigger d'audit	7
2.	Event Trigger pour empêcher les opérations sur la table "authors" pendant les jours de semaine :	8
3.	Déclencheur pour capturer l'adresse IP lors des opérations sur la table "users" :.....	9
4.	Déclencheur pour appliquer la contrainte de dates d'emprunt et de retour dans la table "borrowings" :	10
5.	Tests :.....	10

I. Part A

1. Entités

Books:

- book_id (primary key)
- title
- author_id (foreign key referencing Authors table)
- availability_status

Authors:

- author_id (primary key)
- author_name

Users:

- user_id (primary key)
- user_name

Borrowings:

- borrowing_id (primary key)
- book_id (foreign key referencing Books table)
- user_id (foreign key referencing Users table)
- borrow_date
- return_date

2. Creation des tables

-- Create Authors table

```
CREATE TABLE Authors (  
  author_id INT PRIMARY KEY,  
  author_name VARCHAR(100)  
);
```

-- Create Books table

```
CREATE TABLE Books (  
  book_id INT PRIMARY KEY,  
  title VARCHAR(100),  
  author_id INT,  
  availability_status VARCHAR(20),  
  FOREIGN KEY (author_id) REFERENCES Authors(author_id)  
);
```

-- Create Users table

```
CREATE TABLE Users (  
  user_id INT PRIMARY KEY,  
  user_name VARCHAR(100)  
);
```

-- Create Borrowings table

```
CREATE TABLE Borrowings (  
  borrowing_id INT PRIMARY KEY,  
  book_id INT,  
  user_id INT,
```

```
borrow_date DATE,  
return_date DATE,  
FOREIGN KEY (book_id) REFERENCES Books(book_id),  
FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

3. Constraints

- ALTER TABLE Borrowings ADD CONSTRAINT CHK_Borrowings_Dates CHECK (borrow_date < return_date);

II. Part B

1. Event Trigger pour les emprunts:

```
CREATE OR REPLACE TRIGGER after_borrow_trigger  
AFTER INSERT ON Borrowings  
FOR EACH ROW  
BEGIN  
-- Mettre à jour le statut à 'Unavailable'  
UPDATE Books  
SET availability_status = 'Unavailable'  
WHERE book_id = :NEW.book_id;  
END;  
/
```

2. Event Trigger pour le retour:

```
CREATE OR REPLACE TRIGGER after_return_trigger
```

```

AFTER UPDATE ON Borrowings
FOR EACH ROW
BEGIN
    -- Vérifier si le livre a été retourné (return_date n'est pas null)
    IF :NEW.return_date IS NOT NULL THEN
        -- Mettre à jour le statut à 'Available'
        UPDATE Books
            SET availability_status = 'Available'
            WHERE book_id = :NEW.book_id;
    END IF;
END;
/

```

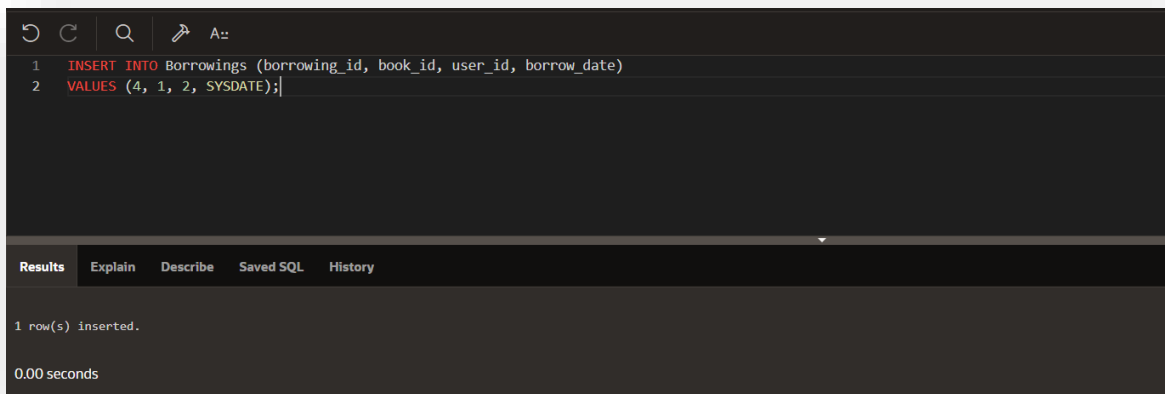
3. Test:

- Effectuer une opération d'emprunt :

```

INSERT INTO Borrowings (borrowing_id, book_id, user_id, borrow_date)
VALUES (4, 1, 2, SYSDATE);

```



The screenshot shows a SQL IDE interface. The top bar contains icons for undo, redo, search, and a toolbar with 'A::'. The main editor area displays two lines of SQL code: `1 INSERT INTO Borrowings (borrowing_id, book_id, user_id, borrow_date)` and `2 VALUES (4, 1, 2, SYSDATE);`. Below the editor, there is a tabbed interface with 'Results' selected. The 'Results' tab shows the output: `1 row(s) inserted.` and `0.00 seconds`.

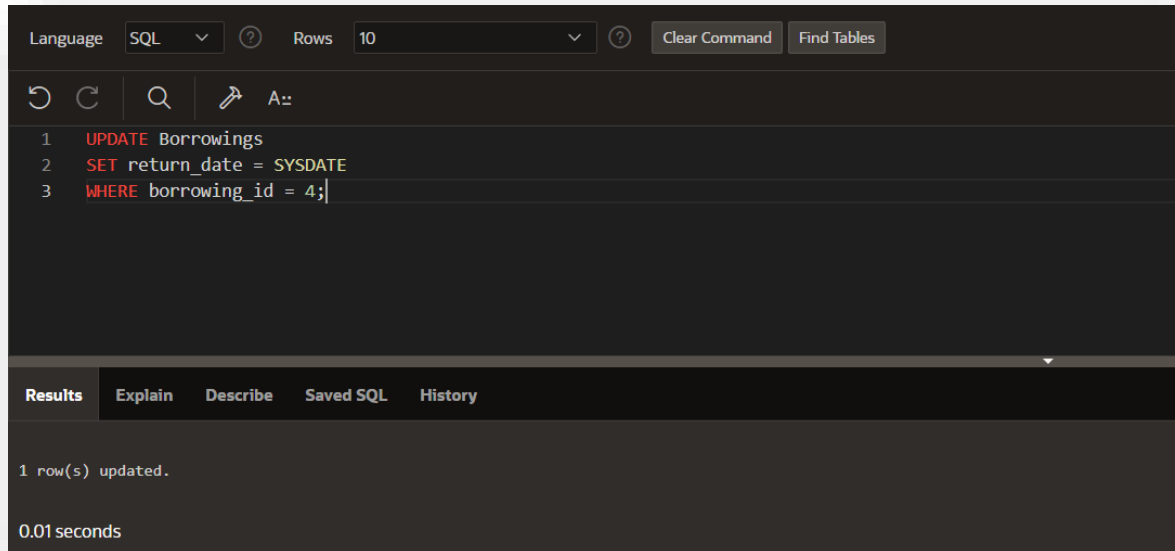
Cela insérera une nouvelle ligne dans la table "Borrowings" et déclenchera le déclencheur "after_borrow_trigger". Le statut de disponibilité du livre avec l'identifiant "book_id" 1 sera mis à jour en "Unavailable" dans la table "Books".

- Effectuer une opération de retour :

UPDATE Borrowings

SET return_date = SYSDATE

WHERE borrowing_id = 4;



Cela mettra à jour la date de retour de l'emprunt avec l'identifiant "borrowing_id" 4 et déclenchera le déclencheur "after_return_trigger". Étant donné que la date de retour n'est pas nulle, le statut de disponibilité du livre avec l'identifiant "book_id" 1 sera mis à jour en "Available" dans la table "Books".

The screenshot shows a SQL IDE interface with a table of results. The table has two columns: 'BOOK_ID' and 'AVAILABILITY_STATUS'. The results are as follows:

BOOK_ID	AVAILABILITY_STATUS
2	Unavailable
1	Available
3	Unavailable

At the bottom of the table, it says '3 rows returned in 0.01 seconds' and there is a 'Download' button.

III. Part C

1. Event Trigger d'audit

- Création de la table d'audit

CREATE TABLE book_audit (

audit_id INT PRIMARY KEY,

book_id INT,

```
action VARCHAR2(10),  
audit_date TIMESTAMP  
);
```

- Création du déclencheur pour l'insertion, la mise à jour et la suppression des livres

```
CREATE OR REPLACE TRIGGER book_audit_trigger  
AFTER INSERT OR UPDATE OR DELETE ON Books  
FOR EACH ROW  
DECLARE  
    v_action VARCHAR2(10);  
BEGIN  
    IF INSERTING THEN  
        v_action := 'insert';  
    ELSIF UPDATING THEN  
        v_action := 'update';  
    ELSIF DELETING THEN  
        v_action := 'delete';  
    END IF;  
    INSERT INTO book_audit (audit_id, book_id, action, audit_date)  
    VALUES (book_audit_seq.NEXTVAL, :NEW.book_id, v_action, SYSTIMESTAMP);  
END;  
/
```

2. Event Trigger pour empêcher les opérations sur la table "authors" pendant les jours de semaine :

- Création du déclencheur pour prévenir les opérations sur la table "authors" pendant les jours de semaine

```
CREATE OR REPLACE TRIGGER prevent_authors_operations_trigger  
BEFORE UPDATE OR DELETE ON Authors  
FOR EACH ROW
```



```

DECLARE

v_day_of_week VARCHAR2(20);

BEGIN

v_day_of_week := TO_CHAR(SYSDATE, 'DAY');


IF v_day_of_week IN ('MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY') THEN

    RAISE_APPLICATION_ERROR(-20001, 'Les opérations sur la table "auteurs" ne sont pas
    autorisées pendant les jours de semaine.');
```

END IF;

```

END;

/
```

3. Déclencheur pour capturer l'adresse IP lors des opérations sur la table "users" :

- Création de la table de journalisation :

```

CREATE TABLE audit_log (

log_id INT PRIMARY KEY,

action VARCHAR2(10),

ip_address VARCHAR2(50),

log_date TIMESTAMP

);
```

- Création du déclencheur pour capturer l'adresse IP lors des opérations sur la table "users":

```

CREATE OR REPLACE TRIGGER capture_user_operations_trigger

AFTER UPDATE OR DELETE ON Users

FOR EACH ROW

DECLARE

v_action VARCHAR2(10);

v_ip_address VARCHAR2(50);

BEGIN

IF UPDATING THEN

    v_action := 'update';

ELSIF DELETING THEN
```

```

        v_action := 'delete';
    END IF;

    v_ip_address := SYS_CONTEXT('USERENV', 'IP_ADDRESS');

    INSERT INTO audit_log (log_id, action, ip_address, log_date)
    VALUES (audit_log_seq.NEXTVAL, v_action, v_ip_address, SYSTIMESTAMP);
END;
/

```

4. Déclencheur pour appliquer la contrainte de dates d'emprunt et de retour dans la table "borrowings" :

- Création du déclencheur pour la contrainte de dates d'emprunt et de retour :

```

CREATE OR REPLACE TRIGGER enforce_borrow_return_dates_trigger
BEFORE INSERT OR UPDATE ON Borrowings
FOR EACH ROW
BEGIN
    IF :NEW.borrow_date > :NEW.return_date THEN
        RAISE_APPLICATION_ERROR(-20002, 'La date d'emprunt ne peut pas être postérieure à la date de retour.');
```

5. Tests :

BOOKS												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
Create	Drop	Enable	Disable									
Trigger Name		Trigger Type		Triggering Event			Status					
BOOK_AUDIT_TRIGGER		AFTER EACH ROW		INSERT OR UPDATE OR DELETE			ENABLED					

AUTHORS										
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
<div>CreateDropEnableDisable</div>										
Trigger Name				Trigger Type		Triggering Event		Status		
PREVENT_AUTHORS_OPERATIONS_TRIGGER				BEFORE EACH ROW		UPDATE OR DELETE		ENABLED		

USERS

Table

Data

Indexes

Model

Constraints

Grants

Statistics

UI Defaults

Triggers

Dependencies

SQL

Create

Drop

Enable

Disable

Trigger Name	Trigger Type	Triggering Event	Status
CAPTURE_USER_OPERATIONS_TRIGGER	AFTER EACH ROW	UPDATE OR DELETE	ENABLED

BORROWINGS

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
<div>CreateDropEnableDisable</div>										
Trigger Name				Trigger Type		Triggering Event		Status		
AFTER_BORROW_TRIGGER				AFTER EACH ROW		INSERT		ENABLED		
AFTER_RETURN_TRIGGER				AFTER EACH ROW		UPDATE		ENABLED		
ENFORCE_BORROW_RETURN_DATES_TRIGGER				BEFORE EACH ROW		INSERT OR UPDATE		ENABLED		

- PL/SQL block pour tester les déclencheurs :

BEGIN

-- Test du déclencheur book_audit_trigger

INSERT INTO Books (book_id, title, author_id, availability_status)

VALUES (4, 'New Book', 1, 'Available');

UPDATE Books SET author_id = [nouvelle_valeur] WHERE book_id = 3;

UPDATE Books SET title = 'Updated Book' WHERE book_id = 1;

DELETE FROM Books WHERE book_id = 3;

-- Vérification des enregistrements dans la table book_audit

-- Test du déclencheur prevent_authors_operations_trigger

UPDATE Authors SET author_name = 'New Name' WHERE author_id = 1; -- Essayer cela pendant un jour de semaine

-- Test du déclencheur capture_user_operations_trigger

UPDATE Users SET user_name = 'New Name' WHERE user_id = 1;

-- Vérification des enregistrements dans la table audit_log

-- Test du déclencheur enforce_borrow_return_dates_trigger

INSERT INTO Borrowings (borrowing_id, book_id, user_id, borrow_date, return_date)

VALUES (4, 2, 3, SYSDATE, SYSDATE - 1); -- Essayer cela avec une date d'emprunt supérieure à la date de retour

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

END;

Results

Explain

Describe

Saved SQL

History

Error: ORA-02292: integrity constraint (FR_A580_PLSQL_S13.SYS_C001159401) violated - child record found

1 row(s) updated.

0.02 seconds

Résultat :

Books :

Tables

Q

AUDIT_LOG

AUTHORS

BOOKS

BOOK_AUDIT

BORROWINGS

USERS

BOOKS

Table

Data

Indexes

Model

Constraints

Grants

Statistics

UI Defaults

Triggers

Dependencies

SQL

REST

Sample Queries

Query

Count Rows

Insert Row

Load Data

EDIT	BOOK_ID	TITLE	AUTHOR_ID	AVAILABILITY_STATUS
	2	Dragon	2	Unavailable
	4	New Book	1	Available
	1	Updated Book	1	Available
	3	Technology	3	Unavailable

Download

row(s) 1 - 4 of 4

Book_audit :

Tables

Q

AUDIT_LOG

AUTHORS

BOOKS

BOOK_AUDIT

BORROWINGS

USERS

BOOK_AUDIT

Table

Data

Indexes

Model

Constraints

Grants

Statistics

UI Defaults

Triggers

Dependencies

SQL

REST

Sample Queries

Query

Count Rows

Insert Row

Load Data

EDIT	AUDIT_ID	BOOK_ID	ACTION	AUDIT_DATE
	1	4	INSERT	21-JUN-23 12:56:16.000000 PM
	2	1	UPDATE	21-JUN-23 12:56:16.000000 PM

Download

row(s) 1 - 2 of 2

Borrowings :

Tables

Q

AUDIT_LOG

AUTHORS

BOOKS

BOOK_AUDIT

BORROWINGS

USERS

BORROWINGS

Table

Data

Indexes

Model

Constraints

Grants

Statistics

UI Defaults

Triggers

Dependencies

SQL

REST

Sample Queries

Query

Count Rows

Insert Row

Load Data

EDIT	BORROWING_ID	BOOK_ID	USER_ID	BORROW_DATE	RETURN_DATE
	1	1	1	21-Jun-2023	-
	4	1	2	21-Jun-2023	21-Jun-2023
	2	2	2	21-Jun-2023	-
	3	3	3	21-Jun-2023	-

Download

row(s) 1 - 4 of 4

- Insérer un enregistrement dont la date d'emprunt est postérieure à la date de retour :

Language SQL Rows 10 Clear Command Find Tables

↶ ↷ 🔍 ↵ A::

```
1 INSERT INTO Borrowings (borrowing_id, book_id, user_id, borrow_date, return_date)
2 VALUES (4, 1, 1, SYSDATE, SYSDATE - 1);
```

Results Explain Describe Saved SQL History

ORA-20002: La date d'emprunt ne peut pas être postérieure à la date de retour.
ORA-06512: at "FR_A580_PLSQL_S13.ENFORCE_BORROW_RETURN_DATES_TRIGGER", line 3
ORA-04088: error during execution of trigger
'FR_A580_PLSQL_S13.ENFORCE_BORROW_RETURN_DATES_TRIGGER'
ORA-06512: at "SYS.DBMS_SQL", line 1721

```
1. INSERT INTO Borrowings (borrowing_id, book_id, user_id, borrow_date,
2. VALUES (4, 1, 1, SYSDATE, SYSDATE - 1);
```

Audit_Log :

Language SQL Rows 10 Clear Command Find Tables

↶ ↷ 🔍 ↵ A::

```
1 -- Update an existing user
2 UPDATE Users SET user_name = 'Adam Smith' WHERE user_id = 1;
3
4 -- Delete a user
5 DELETE FROM Users WHERE user_id = 2;
```

AUDIT_LOG + ▾

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL REST Sample Queries

Query Count Rows Insert Row Load Data

EDIT	LOG_ID	ACTION	IP_ADDRESS	LOG_DATE
	4	delete	100.120.126.34	21-JUN-23 01:25:20:160084 PM
	1	update	100.120.126.34	21-JUN-23 01:09:21:032521 PM

Download

row(s) 1 - 2 of 2