|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Sign** |
| **22-47018-1** | **FARJANA YESMIN OPI** |  |

**Instructions:**

* **Make sure to write your Name, ID and Signature on this document.**
* **First write your signature on a paper then take a photo of that signature and use it for signing this document.**
* **After completing the requirements of the PL/SQL Tutorial Supplementary by editing this document, upload this document in the link provided in your VUES Student Account.**

**PL/SQL Tutorial Supplementary**

Belle is held captive by the mighty Beast in his ominous castle. She is forbidden to go outside its gates but is free to roam around inside it. However, the only place Belle seems to enjoy is the library as she loves to read story books. Seeing this, Beast orders that his library should be restructured and in order to do this he has hired your company. Your job is to create a relational database for his Library from the requirements specified below:

RDBMS- Oracle 10g

Language-SQL and PL/SQL

1. Log in as system and create a ***user*** Students having ***password*** ADBMS. The user Students is granted ***unlimited tablespace***. Students is also granted the permission to ***create*** tables, view and sequence.

**Answer:**

CREATE USER Students IDENTIFIED BY ADBMS;

ALTER USER Students DEFAULT TABLESPACE users QUOTA UNLIMITED ON users;

GRANT CREATE SESSION TO Students;

GRANT CREATE TABLE, CREATE VIEW, CREATE SEQUENCE TO Students;

1. After logging in with username and password Students ***run*** the ***queries*** from the ***Supplementary*** ***section*** of ***PL/SQL Tutorial*** to ***create tables*** and ***insert data*** into the created tables.

**Answer:**

**Table Creation:**

**Table name:** author

**Query:**

CREATE TABLE author (

a\_id NUMBER(10) PRIMARY KEY,

a\_name VARCHAR2(20)

);

**Table name:** category

**Query:**

CREATE TABLE category (

c\_id NUMBER(10) PRIMARY KEY,

c\_name VARCHAR2(20)

);

**Table name:** book

**Query:**

CREATE TABLE book (

b\_id NUMBER(10),

b\_name VARCHAR2(20),

isbn VARCHAR2(20),

edition VARCHAR2(20),

c\_id NUMBER(10),

a\_id NUMBER(10),

PRIMARY KEY (b\_id, edition),

FOREIGN KEY (c\_id) REFERENCES category(c\_id),

FOREIGN KEY (a\_id) REFERENCES author(a\_id)

);

**Data Insertion:**

**Table name:** author

INSERT INTO author VALUES (1, 'J.K. Rowling');

INSERT INTO author VALUES (2, 'Stephenie Meyer');

INSERT INTO author VALUES (3, 'Dan Brown');

INSERT INTO author VALUES (4, 'Humayun Ahmed');

INSERT INTO author VALUES (5, 'Zafar Iqbal');

**Table name:** category

INSERT INTO category VALUES (11, 'Fantasy');

INSERT INTO category VALUES (22, 'Romance');

INSERT INTO category VALUES (33, 'Thriller');

INSERT INTO category VALUES (44, 'Anti-logic');

INSERT INTO category VALUES (55, 'Science Fiction');

**Table name:** book

INSERT INTO book VALUES (111, 'HP...Deathly Hallows', '978-3-16-148410-0', '10', 11, 1);

INSERT INTO book VALUES (222, 'Breaking Dawn', '979-3-16-148410-0', '10', 22, 2);

INSERT INTO book VALUES (333, 'Origin', '980-3-16-148410-0', '10', 33, 3);

INSERT INTO book VALUES (444, 'Holud HimuKalo RAB', '981-3-16-148410-0', '10', 44, 4);

INSERT INTO book VALUES (555, 'Obonil', '982-3-16-148410-0', '10', 55, 5);

1. Create a ***function*** that returns the total number of books stored inside the library.

**Answer:**

CREATE OR REPLACE FUNCTION total\_books

RETURN NUMBER IS

total NUMBER;

BEGIN

SELECT COUNT(\*) INTO total FROM book;

RETURN total;

END;

/

DECLARE

total NUMBER;

BEGIN

total := total\_books;

DBMS\_OUTPUT.PUT\_LINE('Total Books: ' || total);

END;

/

1. Beast has ordered that all the books in his library should be of the latest edition. So, create a ***procedure*** to update the value of the edition column of book table from 10 to 20.

**Answer:**

CREATE OR REPLACE PROCEDURE update\_edition IS

BEGIN

UPDATE book SET edition = 20 WHERE edition = 10;

END;

/

BEGIN

update\_edition;

DBMS\_OUTPUT.PUT\_LINE('Edition updated successfully.');

END;

/

1. Create a ***record*** that can output the name of the book whose id is 111.

**Answer:**

DECLARE

book\_rec book%ROWTYPE;

BEGIN

SELECT \* INTO book\_rec FROM book

WHERE b\_id = 111;

DBMS\_OUTPUT.PUT\_LINE('Book Name: ' || book\_rec.b\_name);

END;

/

1. Create a ***record*** that can output the name of all the books inside the library.

**Answer:**

DECLARE

book\_rec book%ROWTYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Names of all the books inside the library:');

FOR rec IN (SELECT \* FROM book) LOOP

book\_rec := rec;

DBMS\_OUTPUT.PUT\_LINE(book\_rec.b\_name);

END LOOP;

END;

/

1. Create a ***cursor*** that can output the id and name of all the categories the books are sorted in.

**Answer:**

DECLARE

CURSOR c\_cat IS SELECT c\_id, c\_name FROM category;

rec\_cat c\_cat%ROWTYPE;

BEGIN

OPEN c\_cat;

LOOP

FETCH c\_cat INTO rec\_cat;

EXIT WHEN c\_cat%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(rec\_cat.c\_id || ' - ' || rec\_cat.c\_name);

END LOOP;

CLOSE c\_cat;

END;

/

1. Create a ***trigger*** in such a way that whenever a new row is inserted into the category table an output ‘New Category Added’ is generated.

**Answer:**

CREATE OR REPLACE TRIGGER trg\_category\_insert

AFTER INSERT ON category

BEGIN

DBMS\_OUTPUT.PUT\_LINE('New Category Added');

END;

/

INSERT INTO category VALUES (66, 'Horror');

1. Create a ***trigger*** in such a way that whenever a row is deleted from the category table an output ‘A Category Deleted’ is generated.

**Answer:**

CREATE OR REPLACE TRIGGER trg\_category\_delete

AFTER DELETE ON category

BEGIN

DBMS\_OUTPUT.PUT\_LINE('A Category Deleted');

END;

/

DELETE FROM category WHERE c\_id = 66;

1. Create a ***package*** that contains a procedure which can display the book name of any book whose id is passed as its parameter.

**Answer:**

**Package Specification:**

CREATE OR REPLACE PACKAGE book\_pkg AS

PROCEDURE show\_book\_name(p\_id IN NUMBER);

END book\_pkg;

/

**Package Body:**

CREATE OR REPLACE PACKAGE BODY book\_pkg AS

PROCEDURE show\_book\_name(p\_id IN NUMBER) IS

v\_name book.b\_name%TYPE;

BEGIN

SELECT b\_name INTO v\_name FROM book WHERE b\_id = p\_id;

DBMS\_OUTPUT.PUT\_LINE('Book Name: ' || v\_name);

END;

END book\_pkg;

/

**Using the Package:**

BEGIN

book\_pkg.show\_book\_name(111);

END;

/