EJERCICIOS PL EN SAKILA

1. **Escribe una función que dado un código de alquiler (rental\_id) devuelva el título que corresponde a la película alquilada.**

**CREATE OR REPLACE FUNCTION TITULO\_PELI\_ALQ\_FUNC (CODIGO RENTAL.RENTAL\_ID%TYPE)**

**RETURN VARCHAR2**

**IS**

**CONTADOR NUMBER(1);**

**TITULO FILM.TITLE%TYPE;**

**BEGIN**

**SELECT COUNT(\*) INTO CONTADOR**

**FROM RENTAL**

**WHERE RENTAL\_ID=CODIGO;**

**IF CONTADOR=0 THEN**

**RETURN 0;**

**ELSE**

**SELECT TITLE INTO TITULO**

**FROM FILM JOIN INVENTORY USING (FILM\_ID)**

**JOIN RENTAL USING (INVENTORY\_ID)**

**WHERE RENTAL\_ID=CODIGO;**

**RETURN TITULO;**

**END IF;**

**END;**

**SELECT TITULO\_PELI\_ALQ\_FUNC(1)**

**FROM DUAL**

1. **Escribe una función que dado el nombre de un cliente devuelva su código.**

**CREATE OR REPLACE FUNCTION CODIGO\_CLIENTE\_FUNC (NOMBRE CUSTOMER.FIRST\_NAME%TYPE, APELLIDO CUSTOMER.LAST\_NAME%TYPE)**

**RETURN NUMBER**

**IS**

**CONTADOR NUMBER(1);**

**CODIGO FILM.TITLE%TYPE;**

**BEGIN**

**SELECT COUNT(\*) INTO CONTADOR**

**FROM CUSTOMER**

**WHERE FIRST\_NAME=NOMBRE AND LAST\_NAME=APELLIDO;**

**IF CONTADOR=0 THEN**

**RETURN 0;**

**ELSE**

**SELECT CUSTOMER\_ID INTO CODIGO**

**FROM CUSTOMER**

**WHERE FIRST\_NAME=NOMBRE AND LAST\_NAME=APELLIDO;**

**RETURN CODIGO;**

**END IF;**

**END;**

**SELECT CODIGO\_CLIENTE\_FUNC('MARY', 'SMITH')**

**FROM DUAL**

1. **Escribe un procedimiento que dado el nombre de un cliente muestre en pantalla un listado con los títulos de películas que ha alquilado. Emite también al final un mensaje que indique el número total de películas alquiladas. (NOTA: Puedes utilizar las funciones de los ejercicios anteriores).**

**CREATE OR REPLACE PROCEDURE EJER3 ( NOMBRE CUSTOMER.FIRST\_NAME%TYPE,**

**APELLIDO CUSTOMER.LAST\_NAME%TYPE) IS**

**CURSOR PELICULAS IS**

**SELECT TITLE**

**FROM FILM JOIN INVENTORY USING (FILM\_ID)**

**JOIN RENTAL USING(INVENTORY\_ID)**

**JOIN CUSTOMER USING (CUSTOMER\_ID)**

**WHERE FIRST\_NAME=NOMBRE AND LAST\_NAME=APELLIDO;**

**CONTADOR NUMBER(2):=0;**

**BEGIN**

**FOR FILA IN PELICULAS LOOP**

**DBMS\_OUTPUT.PUT\_LINE(FILA.TITLE);**

**CONTADOR:=CONTADOR+1;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE('TOTAL DE PELÍCULAS ALQUILADAS: '||CONTADOR);**

**END;**

**SET SERVEROUTPUT ON**

**EXECUTE EJER3('MARY', 'SMITH');**

**CREATE OR REPLACE PROCEDURE EJER3\_V2 ( NOMBRE CUSTOMER.FIRST\_NAME%TYPE,**

**APELLIDO CUSTOMER.LAST\_NAME%TYPE) IS**

**CURSOR PELICULAS IS**

**SELECT FUNC1(RENTAL\_ID) PELICULA**

**FROM RENTAL**

**WHERE CUSTOMER\_ID=FUNC2(NOMBRE, APELLIDO);**

**CONTADOR NUMBER(2):=0;**

**BEGIN**

**FOR FILA IN PELICULAS LOOP**

**DBMS\_OUTPUT.PUT\_LINE(FILA.PELICULA);**

**CONTADOR:=CONTADOR+1;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE('TOTAL DE PELÍCULAS ALQUILADAS: '||CONTADOR);**

**END;**

**SET SERVEROUTPUT ON**

**EXECUTE EJER3\_V2('MARY', 'SMITH');**

1. **Escribe un procedimiento que dado un título de película, muestre un listado de actores que participan en ella.**

**CREATE OR REPLACE PROCEDURE EJER4 ( TITULO FILM.TITLE%TYPE) IS**

**CURSOR ACTORES IS**

**SELECT FIRST\_NAME, LAST\_NAME**

**FROM ACTORES JOIN FILM\_ACTOR USING (ACTOR\_ID)**

**JOIN FILM USING (FILM\_ID)**

**WHERE TITLE=TITULO;**

**BEGIN**

**FOR REG IN ACTORES LOOP**

**DBMS\_OUTPUT.PUT\_LINE(REG.FIRST\_NAME||' '||REG.LAST\_NAME);**

**END LOOP;**

**END;**

**EXECUTE EJER4('ACADEMY DINOSAUR');**

**PREXAMEN:**

**CREATE OR REPLACE PROCEDURE EJER4 ( TITULO FILM.TITLE%TYPE) IS**

**CURSOR ACTORES IS**

**SELECT FIRST\_NAME, LAST\_NAME**

**FROM ACTORES JOIN FILM\_ACTOR USING (ACTOR\_ID)**

**JOIN FILM USING (FILM\_ID)**

**WHERE TITLE=TITULO;**

**BEGIN**

**FOR REG IN ACTORES LOOP**

**DBMS\_OUTPUT.PUT\_LINE(REG.FIRST\_NAME||' '||REG.LAST\_NAME);**

**END LOOP;**

**END;**

**EXECUTE EJER4('ACADEMY DINOSAUR');**

**UPDATE ACTORES**

**SET NUMPELIS = (SELECT COUNT(\*)**

**FROM FILM\_ACTOR**

**WHERE ACTOR\_ID= ACTORES.ACTOR\_ID);**

**CREATE OR REPLACE TRIGGER T2 AFTER INSERT OR DELETE ON FILM\_ACTOR FOR EACH ROW**

**BEGIN**

**IF DELETING THEN**

**UPDATE ACTOR SET NUMPELIS=NUMPLEIS-1**

**WHERE ACTOR\_ID=:OLD.ACTOR\_ID;**

**ELSE**

**UPDATE ACTOR SET NUMPELIS=NUMPLEIS+1**

**WHERE ACTOR\_ID=:NEW.ACTOR\_ID;**

**END IF;**

**END;**

**CREATE OR REPLACE TRIGGER T3 BEFORE INSERT OR UPDATE ON FILM FOR EACH ROW**

**WHEN ( NEW.REPLACEMENT\_COST > NEW.RENTAL\_RATE \*20)**

**BEGIN**

**RAISE\_APPLICATION\_ERROR(-20100, 'NO SE PUEDA SUPERAR 20 ...');**

**END;**