João Paulo Oliveira Cruz

Database Systems Management

Practice Exercise 04

As with previous assignments, queries can also be found on https://github.com/jpoCruz/lut-databases/tree/master/databasesSystemsManagemet/week4.

1

a

```
ALTER TABLE CUSM
ADD SignOnFlag varchar(1) CHECK (SignOnFlag = 'Y' or SignOnFlag = 'N');

ALTER TABLE CUSM
ADD SignOnDate date;

ALTER TABLE CUSM
ADD passwrd varchar(10);

UPDATE CUSM SET passwrd = 'apple123' WHERE custid = 1;
UPDATE CUSM SET passwrd = 'qwerty' WHERE custid = 2;
UPDATE CUSM SET passwrd = 'password1' WHERE custid = 3;
```

b

```
CREATE OR REPLACE PROCEDURE passwordChecker(user_id IN NUMBER, user_password IN
VARCHAR)
AS
    v_truePassword varchar(10);
BEGIN

    SELECT passwrd
    INTO v_truePassword
    FROM CUSM
    WHERE CUSM.custid = passwordChecker.user_id;

    IF passwordChecker.user_password = v_truePassword THEN
        UPDATE CUSM SET SignOnFlag = 'Y' WHERE custid = passwordChecker.user_id;
        UPDATE CUSM SET SignOnDate = SYSDATE WHERE custid =
passwordChecker.user_id;
        DBMS_OUTPUT.put_line('True password.');
    ELSE
        UPDATE CUSM SET SignOnFlag = 'N' WHERE custid = passwordChecker.user_id;
        DBMS_OUTPUT.put_line('False password.');
    END IF;

END passwordChecker;
```

2

```
begin
    dbms_errlog.create_error_log(dml_table_name => 'CUSM');

    dbms_errlog.create_error_log(dml_table_name => 'accounts');

    dbms_errlog.create_error_log(dml_table_name => 'TXNM_table');
end;

INSERT INTO CUSM
SELECT *
FROM CUSM LOG errors
INTO err$_CUSM reject LIMIT unlimited;

INSERT INTO accounts
SELECT *
FROM accounts LOG errors
INTO err$_accounts reject LIMIT unlimited;

INSERT INTO TXNM_table
SELECT *
FROM TXNM_table LOG errors
INTO err$_TXNM_table reject LIMIT unlimited;

SELECT * from ERR$_CUSM
SELECT * from ERR$_accounts
SELECT * from ERR$_TXNM_table
```

3

```
CREATE TABLE log (
    logDate DATE,
    logId NUMBER);

CREATE OR REPLACE TRIGGER log_access
AFTER INSERT OR UPDATE OR DELETE on accounts
FOR EACH ROW
BEGIN
    insert into log (logDate, logId)
    values (SYSDATE, :new.CUST_ID);
END;
```

To test this, I had to do one of each operation and then see the log. These are the commands I used:

```
SELECT * FROM accounts;

INSERT INTO accounts(accountUser, interestRate, openingDate, status, balance,
accountType)
VALUES (1, 10, '2015-01-10', 'Active', 1000, 'NOD');

UPDATE accounts SET status = 'Closed' WHERE accountUser = 3;

DELETE FROM accounts WHERE status = 'Closed';

SELECT * FROM accounts;
SELECT * FROM log;
```