CSC 352 / 452: Database Programming

Project  
(270 Points)

**Due on Friday, 8/18/2017 at midnight**

**Late projects will not be accepted or graded.**

* **This is NOT a group project. You may not consult in any form with any other person while doing this project.**
* **You must submit the procedures in Part 2 and the package in Part 3 separately. You will lose 245 points if you only submit the package in Part 3.**
* **If there is a syntax error anywhere in your program, you will receive 0 points for the program.**
* **Please submit a text file containing all your answers to D2L by the due date. All other file types (e.g., DOC, DOCX, RTF, PDF, JPG, or ZIP) will be rejected. In D2L, only the most recent submission is kept.**
* **Please review your assignment file before submitting it to make sure you have the correct one. It is your responsibility to upload the correct assignment file.**
* **Do not wait until the last day to submit your file! Submit ahead in case of unforeseen circumstances such as a computer crash or illness. There are no exceptions to the deadline.**

Your task is to build a PL/SQL-based application to insert and update records in a video rental store database and generate some reports.

The database consists of only the following essential tables.

* TAB\_CUSTOMER(**CUSTOMER\_ID,** CUSTOMER\_NAME, CUSTOMER\_STATUS, EMAIL,

PHONE\_NUMBER, REGISTRATION\_DATE, LAST\_UPDATE\_DATE);

* TAB\_VIDEO(**VIDEO\_ID**, VIDEO\_NAME, VIDEO\_FORMAT, PUBLISH\_DATE,

MAXIMUM\_RENTAL\_DAYS);

* TAB\_VIDEO\_COPY(**COPY\_ID**, VIDEO\_ID\*, COPY\_STATUS, NOTE);
* TAB\_VIDEO\_RENTAL (**CUSTOMER\_ID\*, COPY\_ID\*, CHECKOUT\_DATE,**

DUE\_DATE, RETURN\_DATE);

The primary keys are marked in red and the foreign keys are marked with asterisks.

TAB\_CUSTOMER(CUSTOMER\_STATUS): AC – Active, IN – Inactive

TAB\_VIDEO\_COPY(COPY\_STATUS): A – Available, R – Rented, D – Damaged

Each video in the TAB\_VIDEO table has at least one video copy in the TAB\_VIDEO\_COPY table.

**Part 1)**

Create and populate the TAB\_CUSTOMER, TAB\_VIDEO, TAB\_VIDEO\_COPY, and TAB\_VIDEO\_RENTAL tables by using the following SQL statements.

CREATE TABLE tab\_customer

( CUSTOMER\_ID VARCHAR2(15) PRIMARY KEY,

CUSTOMER\_NAME VARCHAR2(30) NOT NULL,

CUSTOMER\_STATUS VARCHAR2(2) NOT NULL CONSTRAINT ck\_custatus

CHECK (CUSTOMER\_STATUS IN ('AC', 'IN')),

EMAIL VARCHAR2(50) NOT NULL,

PHONE\_NUMBER VARCHAR2(15) NOT NULL,

REGISTRATION\_DATE DATE NOT NULL,

LAST\_UPDATE\_DATE DATE NOT NULL);

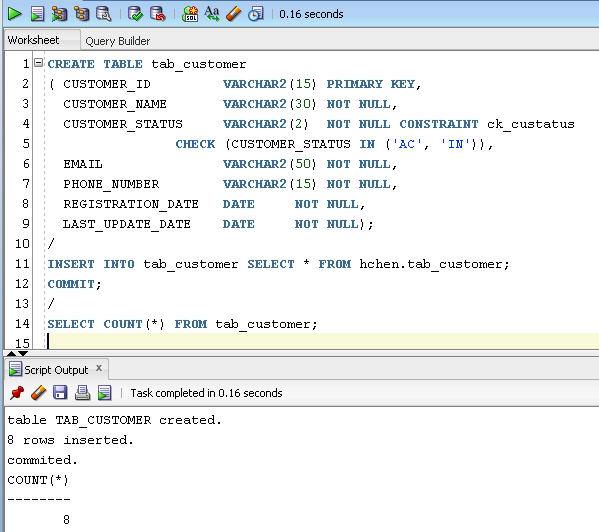
/

INSERT INTO tab\_customer SELECT \* FROM **hchen**.tab\_customer;

COMMIT;

/

SELECT COUNT(\*) FROM tab\_customer;



Please make sure that there are 8 rows in your TAB\_CUSTOMER table.

CREATE TABLE tab\_video

( VIDEO\_ID VARCHAR2(15) PRIMARY KEY,

VIDEO\_NAME VARCHAR2(100) NOT NULL,

VIDEO\_FORMAT VARCHAR2(100) NOT NULL,

PUBLISH\_DATE DATE NOT NULL,

MAXIMUM\_RENTAL\_DAYS NUMBER(3) NOT NULL

);

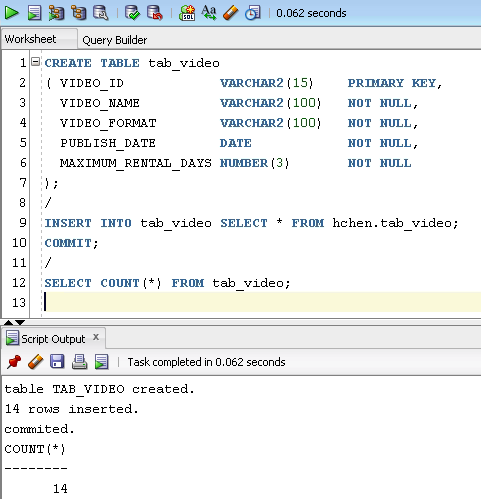
/

INSERT INTO tab\_video SELECT \* FROM **hchen**.tab\_video;

COMMIT;

/

SELECT COUNT(\*) FROM tab\_video;



Please make sure that there are 14 rows in your TAB\_VIDEO table.

CREATE TABLE tab\_video\_copy

( COPY\_ID VARCHAR2(15) PRIMARY KEY,

VIDEO\_ID VARCHAR2(15) NOT NULL REFERENCES tab\_video (VIDEO\_ID),

COPY\_STATUS CHAR NOT NULL CONSTRAINT ck\_cp\_status

CHECK (COPY\_STATUS in ('A', 'R', 'D')),

NOTE VARCHAR2(200));

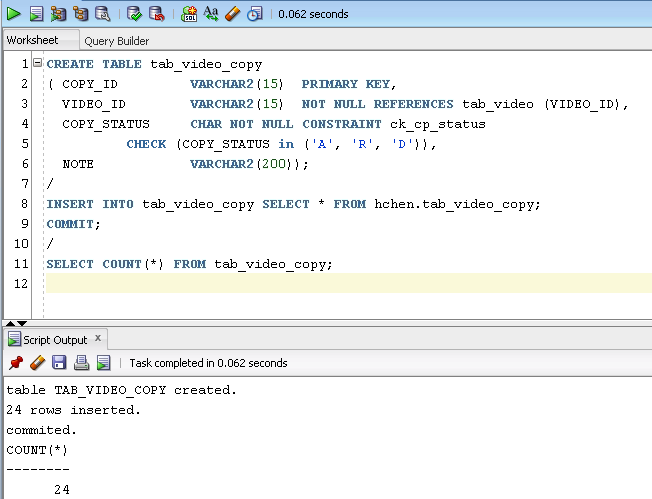
/

INSERT INTO tab\_video\_copy SELECT \* FROM **hchen**.tab\_video\_copy;

COMMIT;

/

SELECT COUNT(\*) FROM tab\_video\_copy;



Please make sure that there are 24 rows in your TAB\_VIDEO\_COPY table.

CREATE TABLE tab\_video\_rental

( CUSTOMER\_ID VARCHAR2(15) REFERENCES tab\_customer (CUSTOMER\_ID),

COPY\_ID VARCHAR2(15) REFERENCES tab\_video\_copy (COPY\_ID),

CHECKOUT\_DATE DATE NOT NULL,

DUE\_DATE DATE NOT NULL,

RETURN\_DATE DATE,

NOTE VARCHAR2(200),

CONSTRAINT pk\_rental PRIMARY KEY

(CUSTOMER\_ID, COPY\_ID, CHECKOUT\_DATE));

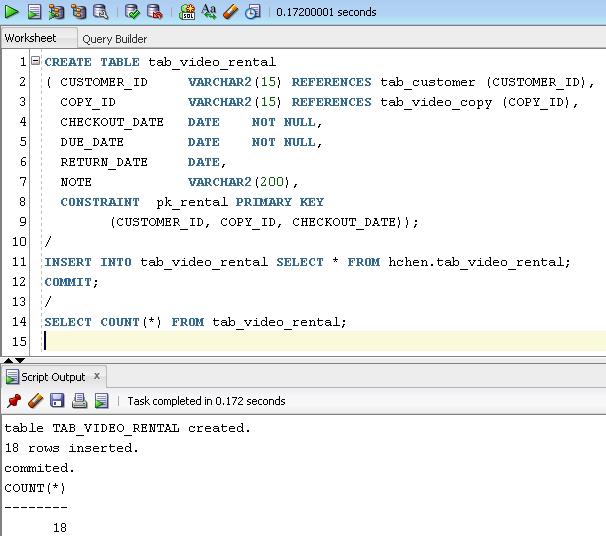
/

INSERT INTO tab\_video\_rental SELECT \* FROM **hchen**.tab\_video\_rental;

COMMIT;

/

SELECT COUNT(\*) FROM tab\_video\_rental;



Please make sure that there are 18 rows in your TAB\_VIDEO\_RENTAL table.

**Part 2) [245 Points]**

* You are not allowed to create temporary tables/views/triggers.
* You can only use the TAB\_CUSTOMER, TAB\_VIDEO, TAB\_VIDEO\_COPY, and TAB\_VIDEO\_RENTAL tables. You will receive 0 points if you use a different table (e.g., different table names, column names, or data types).
* You cannot change the procedure headers. You will receive 0 points if you use a different procedure header (e.g., different procedure names, parameter names, data types, or default values).

1) [5 Points] **– proc\_customer\_registration()**

Create a procedure called proc\_customer\_registrationto add a new customer to the TAB\_CUSTOMER table.

**(You may use my example in your project.)**

The procedure header is

CREATE OR REPLACE PROCEDURE proc\_customer\_registration

(

in\_customer\_id VARCHAR2,

in\_customer\_name VARCHAR2,

in\_customer\_status CHAR,

in\_email VARCHAR2,

in\_phone\_number VARCHAR2,

in\_registration\_date DATE

)

Consider the following special cases:

* The in\_customer\_id is invalid.
* The in\_customer\_status is invalid.
* The in\_customer\_name is empty.
* The in\_phone\_number is empty.
* The in\_email is empty.
* The value of in\_registration\_date is greater than the current date.

**Example**

***CREATE OR REPLACE PROCEDURE proc\_customer\_registration***

***(***

***in\_customer\_id VARCHAR2,***

***in\_customer\_name VARCHAR2,***

***in\_customer\_status CHAR,***

***in\_email VARCHAR2,***

***in\_phone\_number VARCHAR2,***

***in\_registration\_date DATE***

***)***

***IS***

***v\_count NUMBER;***

***BEGIN***

***IF LENGTH(TRIM(in\_customer\_id)) > 15 THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_customer\_id!');***

***RETURN;***

***END IF;***

***SELECT COUNT(\*)***

***INTO v\_count***

***FROM tab\_customer***

***WHERE customer\_id = in\_customer\_id;***

***IF v\_count != 0 THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_customer\_id!');***

***RETURN;***

***END IF;***

***IF TRIM(in\_customer\_name) is NULL OR***

***LENGTH(TRIM(in\_customer\_name)) > 30 THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_customer\_name!');***

***RETURN;***

***END IF;***

***IF UPPER(TRIM(in\_customer\_status)) != 'AC' AND***

***UPPER(TRIM(in\_customer\_status)) != 'IN' THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_customer\_status!');***

***RETURN;***

***END IF;***

***IF TRIM(in\_email) is NULL OR LENGTH(TRIM(in\_email)) > 50 THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_email!');***

***RETURN;***

***END IF;***

***IF TRIM(in\_phone\_number) is NULL OR***

***LENGTH(TRIM(in\_phone\_number)) > 50 THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_phone\_number!');***

***RETURN;***

***END IF;***

***IF in\_registration\_date IS NULL OR***

***TO\_CHAR(in\_registration\_date, 'yyyymmdd') < '20000101' OR***

***TO\_CHAR(in\_registration\_date, 'yyyymmdd') >***

***TO\_CHAR(sysdate, 'yyyymmdd') THEN***

***DBMS\_OUTPUT.PUT\_LINE('Invalid in\_registration\_date!');***

***RETURN;***

***END IF;***

***INSERT INTO tab\_customer***

***VALUES(TRIM(in\_customer\_id), UPPER(TRIM(in\_customer\_name)),***

***UPPER(TRIM(in\_customer\_status)), TRIM(in\_email),***

***TRIM(in\_phone\_number), in\_registration\_date, sysdate);***

***COMMIT;***

***DBMS\_OUTPUT.PUT\_LINE***

***(INITCAP(in\_customer\_name) ||***

***' has been added into the tab\_customer table.');***

***EXCEPTION***

***WHEN OTHERS THEN***

***DBMS\_OUTPUT.PUT\_LINE('My exception: ' ||***

***TO\_CHAR(SQLCODE) || ' ' || SQLERRM);***

***END;***

**Testing the procedure**

* EXEC proc\_customer\_registration('C4000', 'Adams', 'AC', 'adams\_1@yahoo.com', '3123621111', '02-FEB-2017')

Dbms Output: Adams has been added into the tab\_customer table.

* EXEC proc\_customer\_registration('C5010', ' ', 'AC', 'ford1@yahoo.com', '3123622222', '02-FEB-2017')

Dbms Output: Invalid in\_customer\_name!

* ……

2) [CSC 352 - 30 points | CSC 452 - 20 points] **– proc\_change\_customer\_status()**

Create a procedure called proc\_change\_customer\_status to change the status of an existing customer.

The procedure header is

CREATE OR REPLACE PROCEDURE proc\_change\_customer\_status

(

in\_customer\_id VARCHAR2,

in\_new\_customer\_status VARCHAR2

)

You need to consider the following cases:

* The value of in\_customer\_id is not in the CUSTOMER\_ID column of the TAB\_CUSTOMER table.

Dbms Output: Invalid in\_customer\_id!

* The value of UPPER(in\_new\_customer\_status) is not in ('AC', 'IN').

Dbms Output: Invalid in\_new\_customer\_status!

(The value of in\_new\_customer\_status is NOT case-sensitive.)

* The value of in\_customer\_id is in the CUSTOMER\_ID column of the TAB\_CUSTOMER table and the value of UPPER(in\_new\_customer\_status) is in ('AC', 'IN').
* Update the TAB\_CUSTOMER table (the CUSTOMER\_STATUS and LAST\_UPDATE\_DATE columns).

Dbms Output: The status of the customer has been updated.

3) [CSC 352 - 90 points | CSC 452 - 70 points] **– proc\_video\_search()**

Create a procedure called proc\_video\_search to search a video and display the name, copy ID, format, and status of the video’s copies. In addition, the checkout dates and due dates are also displayed for unreturned copies. The damaged copies (COPY\_STATUS = 'D') are excluded in your output. Sort your output by the video name (NAME) and then the copy ID (COPY\_ID).

The video name is not case-sensitive (e.g., Ocean = ocean). The format of a video is not case-sensitive (e.g., DVD = dvd).

The procedure header is

CREATE OR REPLACE PROCEDURE proc\_video\_search

(

in\_video\_name VARCHAR2,

in\_video\_format VARCHAR2 DEFAULT NULL

)

Hint: WHERE UPPER(video\_name) like '%' || UPPER(in\_video\_name) || '%';

**Testing the procedure**

(If your output does not match mine EXACTLY, you will lose points.)

* EXEC proc\_video\_search('ocean')

Dbms Output:

\*\*\*\*\* 0 results found for ocean. \*\*\*\*\*

* EXEC proc\_video\_search('PRETTY WOMAN', 'Blu-Ray')

Dbms Output:

\*\*\*\*\* 0 results found for PRETTY WOMAN (Blu-Ray). \*\*\*\*\*

* EXEC proc\_video\_search('Pretty Woman')

Dbms Output:

\*\*\*\*\* 3 results found for Pretty Woman. (Available copies: 3) \*\*\*\*\*

VIDEO NAME COPY ID VIDEO FORMAT COPY STATUS CHECKOUT DATE DUE DATE

----------------------------------------------------------------------------------------------------

PRETTY WOMAN VC6000 VHS TAPE Available

PRETTY WOMAN VC6001 VHS TAPE Available

PRETTY WOMAN VC6015 DVD Available

* EXEC proc\_video\_search('Another')

Dbms Output:

\*\*\*\*\* 4 results found for Another. (Available copies: 2) \*\*\*\*\*

VIDEO NAME COPY ID VIDEO FORMAT COPY STATUS CHECKOUT DATE DUE DATE

----------------------------------------------------------------------------------------------------

DIE ANOTHER DAY VC6010 VHS TAPE Available

DIE ANOTHER DAY VC6011 VHS TAPE Rented 20-APR-2016 04-MAY-2016

DIE ANOTHER DAY VC6014 DVD Available

DIE ANOTHER DAY VC6016 BLU-RAY Rented 01-OCT-2016 04-OCT-2016

* EXEC proc\_video\_search('ANOTHER', 'Dvd')

Dbms Output:

\*\*\*\*\* 1 result found for ANOTHER. (Available copies: 1) \*\*\*\*\*

VIDEO NAME COPY ID VIDEO FORMAT COPY STATUS CHECKOUT DATE DUE DATE

----------------------------------------------------------------------------------------------------

DIE ANOTHER DAY VC6014 DVD Available

* EXEC proc\_video\_search('Story')

Dbms Output:

\*\*\*\*\* 7 results found for Story. (Available copies: 4) \*\*\*\*\*

VIDEO NAME COPY ID VIDEO FORMAT COPY STATUS CHECKOUT DATE DUE DATE

----------------------------------------------------------------------------------------------------

TOY STORY VC6002 VHS TAPE Rented 09-APR-2016 23-APR-2016

TOY STORY VC6003 VHS TAPE Available

TOY STORY VC6017 DVD Rented 01-OCT-2016 08-OCT-2016

TOY STORY 2 VC6009 VHS TAPE Available

TOY STORY 2 VC6018 DVD Rented 28-APR-2016 05-MAY-2016

TOY STORY 2 VC6019 DVD Available

TOY STORY 2 VC6020 BLU-RAY Available

4) [CSC 352 - 75 points | CSC 452 - 65 Points] **–** **proc\_video\_checkout()**

Create a procedure called proc\_video\_checkout to record a new rental. When the video is successfully checked out, you need to insert a new row (record) into the TAB\_VIDEO\_RENTAL table and update the corresponding row (record) in the TAB\_VIDEO\_COPY table. Otherwise, the action is denied.

The procedure header is

CREATE OR REPLACE PROCEDURE proc\_video\_checkout

(

in\_customer\_id VARCHAR2,

in\_copy\_id VARCHAR2,

in\_video\_checkout\_date DATE

)

An inactive customer (customer\_status = 'IN') is not allowed to make a rental.

Consider the following special cases:

* The value of in\_customer\_id is not in the CUSTOMER\_ID column of the TAB\_CUSTOMER table.
* The status of the customer is not 'AC'.
* The copy is not available (COPY\_STATUS = 'R' or 'D').
* The value of in\_video\_checkout\_date is greater than the current date.
* Checkout periods are determined by the values in the MAXIMUM\_RENTAL\_DAYS column. The due date is in\_video\_checkout\_date plus the corresponding MAXIMUM\_RENTAL\_DAYS. Hard coding (e.g., 3, 7, or 14) is not allowed.
* A customer may have up to **three (3)** copies checked out at any one time.

(For example, Tom has three copies checked out; he cannot rent a copy before he returns one of the three copies he checked out.)

* (**CSC 452 only**) Before a customer returns a copy, he/she cannot rent a second copy of the same video (**VIDEO\_ID**).

For example, Mary has the copy with ID 'VC6000' (COPY\_ID = '**VC6000**') checked out. She cannot rent the copy with ID 'VC6001' (COPY\_ID = '**VC6001**') before she returns the copy with ID '**VC6000**'. The reason is that both copies have the same VIDEO\_ID (VIDEO\_ID = 'V1000'), ('V1000','PRETTY WOMAN', 'VHS TAPE').

**You need to create/run some test cases.**

5) [CSC 352 - 45 points | CSC 452 - 40 points] **–** **proc\_video\_return()**

Create a procedure called proc\_video\_return to change the rental status for that returned copy. When the copy is successfully returned, you need to update the corresponding rows (records) in the TAB\_VIDEO\_RENTAL and TAB\_VIDEO\_COPY tables. Otherwise, the action is denied.

The procedure header is

CREATE OR REPLACE PROCEDURE proc\_video\_return

(

in\_copy\_id VARCHAR2,

in\_video\_return\_date DATE

)

Consider the following special cases:

* The value of in\_copy\_id does not exist in the corresponding column of the TAB\_VIDEO\_COPY table.
* The status of that copy is not “R” (COPY\_STATUS != 'R').
* The value of in\_video\_return\_date is greater than the current date.

**You need to create/run some test cases.**

6) [CSC 452 only - 45 points] – **proc\_unreturned\_video()**

Create a procedure called proc\_unreturned\_video to retrieve all the copies that a customer hasn't returned. The output should include the customer's ID, name, status, email, phone number, first checkout date, last checkout date, the number of unreturned copies, video name, copy ID, video ID, video format, checkout date, and due date of the rentals. Sort the data by due date and then the video name. The procedure header is

CREATE OR REPLACE PROCEDURE proc\_unreturned\_video

(

in\_customer\_id VARCHAR2

)

**Testing the procedure**

(If your output does not match mine EXACTLY, you will lose points.)

* EXEC proc\_unreturned\_video('90')

Dbms Output:

The customer (id = 90) is not in the tab\_customer table.

* EXEC proc\_unreturned\_video('C2004')

Dbms Output:

----------------------------------------

Customer ID: C2004

Customer Name: JOHNSON

Customer Status: AC

Email: jo12@gmail.com

Phone Number: 7732015678

First Checkout Date: 01-MAY-2015

Last Checkout Date: 01-MAY-2015

----------------------------------------

Number of Unreturned Videos: 0

----------------------------------------

* EXEC proc\_unreturned\_video('C2008')

Dbms Output:

----------------------------------------

Customer ID: C2008

Customer Name: SCOTT

Customer Status: IN

Email: scott@hotmail.com

Phone Number: 7701239876

First Checkout Date: N/A

Last Checkout Date: N/A

----------------------------------------

Number of Unreturned Videos: 0

----------------------------------------

* EXEC proc\_unreturned\_video('C2002')

Dbms Output:

----------------------------------------

Customer ID: C2002

Customer Name: JONES

Customer Status: AC

Email: jones@hotmail.com

Phone Number: 6302077890

First Checkout Date: 04-MAR-2015

Last Checkout Date: 01-OCT-2016

----------------------------------------

Number of Unreturned Videos: 3

----------------------------------------

Copy ID: VC6007

Video ID: V1040

Video Name: TARZAN

Format: VHS TAPE

Checkout Date: 22-APR-2016

Due Date: 06-MAY-2016

----------------------------------------

Copy ID: VC6016

Video ID: V1100

Video Name: DIE ANOTHER DAY

Format: BLU-RAY

Checkout Date: 01-OCT-2016

Due Date: 04-OCT-2016

----------------------------------------

Copy ID: VC6017

Video ID: V1110

Video Name: TOY STORY

Format: DVD

Checkout Date: 01-OCT-2016

Due Date: 08-OCT-2016

----------------------------------------

**Part 3) [25 Points]**

Package **video\_rental\_pkg**

Group all the subprograms (proc\_customer\_registration, proc\_change\_customer\_status, proc\_video\_search, proc\_video\_checkout …) in Part 2 together in a package (package specification and package body) called video\_rental\_pkg.

5 points - package specification

20 points - package body

Please test your package.

**You must submit Part 2 (all procedures) and Part 3 (package specification and package body) separately. You will lose 245 points if you only submit the package in Part 3.**

Please submit a text file containing all the source codes to D2L by the due date.