Larry’s DVD & Game Rentals

Larry’s DVD & Game Rentals (LDGR), a small DVD / video game rental store, has decided to automate their operations. You have been hired to set up a database to track their inventory, rentals, and customers. The following is a brief description of the design specifications and business rules.

**DVD and Game Rentals**

LDGR maintains a list of customers who can rent items. All inventory items have had a unique barcode id assigned to them. LDGR has several copies of most, but not all, items. When a rental is made, the customer receives an itemized invoice of all rented items. The rental duration is dependent on the item and is 1, 2 or 7 days in length, but is normally 1 day. LDGR wishes to keep historical data on all rentals.

**Additional Information**

1. A customer may have several rentals in one day
2. One rental invoice can have several rental items
3. Rental items consist of DVD’s, video games and video game consoles
4. LDGR can have several copies of a rental item each with its own cost, which may vary depending on when the item was purchased (see sample report below)
5. The description, rental duration and rental rate of an item is the same for all copies of that item
6. The rental rate is for the specific rental duration and **includes** GST
7. The barcode id is a combination of the rental item number and a copy number
8. Barcode ids are never reused even if a rental item is damaged, lost or destroyed
9. Deposits on game console rentals are via credit card imprints and are destroyed when the console is returned (no record of deposits are kept)
10. Overdue charges are calculated on a per day basis using the overdue rental rate
11. LDGR wishes to keep track of whether a late charge owing on a rental has been paid
12. An invoice is printed for each rental (as shown in the example below)
13. Invoice numbers are never reused

###### Sample Rental Invoice

Invoice Number: 5584

Name: Robert Smith Customer Number: 500

Address: Box 333 Phone: 587-986-3201

Leduc, Alberta

Rental Date: September 20, 2020 10:31 pm

Expected

Barcode Title Return Date Duration Rental Price

2210-004 Die Hard IX September 27, 2020 7 $0.99

2220-001 Lord of the Rings I September 27, 2020 7 $0.99

4500-003 Lord of the Rings III September 21, 2020 1 $4.97

Total $6.95

###### Sample Inventory Report

Rental Number: 4500 Description: Lord of the Rings III

Rental Duration: 1 day Rental Rate: 4.97

Copies:

Barcode Purchase Cost

4500-001 $15.00

4500-002 $15.00

4500-003 $11.99

4500-004 $12.00

4500-005 $15.00

**Part A (Due Sep 25th @ 05:00 p.m.)**

1. Create the following tables and corresponding constraints using **only** the Create table statement. Ensure that table and column names are **identical** to what is specified below and do **NOT** use double quotes around your table or column names. All constraints must be **explicitly** named, and you must be consistent with your constraint names. **22 marks**

|  |  |  |  |
| --- | --- | --- | --- |
| Customer | **Customer\_Number** **PK** | Number (6) | Not Null |
|  | First\_Name | Varchar2 (25) | Not Null |
|  | Last\_Name | Varchar2 (30) | Not Null |
|  | Area\_Code | Number (3) | Null |
|  | Phone\_Number | Number (7) | Null |
|  | Street\_Address | Varchar2 (35) | Null |
|  | City | Varchar2 (25) | Null |
|  | Province | Char (2) | Null |
|  | Postal\_Code | Char (6) | Null |
|  | Email\_Address | Varchar2 (75) | Not Null |
|  | Membership\_Date | Date | Not Null |
|  | Current\_Status\_YN | Char (1) | Not Null |
|  |  |  |  |
| Rental\_Invoice\_Detail | ***Invoice\_Number* PK**/FK | Number (5) | Not Null |
|  | ***Bar\_Code* PK**/FK | Char (8) | Not Null |
|  | Date\_Returned | Date | Null |
|  | Rental\_Duration | Number (2) | Not Null |
|  | Original\_Rental\_Rate | Number (4,2) | Not Null |
|  | Overdue\_Paid\_YN | Char (1) | Null |
|  | Overdue\_Rental\_Rate | Number (4,2) | Null |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Rental\_Invoice | **Invoice\_Number** **PK** | Number (5) | Not Null |
|  | *Customer\_Number**FK* | Number (6) | Not Null |
|  | Invoice\_Date | Date | Not Null |
|  |  |  |  |
| Rental\_Item\_Copy | Bar\_Code PK | Char (8) | Not Null |
|  | *Rental\_Item\_Number**FK* | Number (4) | Not Null |
|  | Purchase\_Cost | Number (6,2) | Null |
|  |  |  |  |
| Rental\_Item | Rental\_Item\_Number PK | Number (4) | Not Null |
|  | *Supplier\_Number**FK* | Number (4) | Not Null |
|  | Description | Varchar2 (80) | Not Null |
|  | Rental\_Duration | Number (2) | Not Null |
|  | Rental\_Rate | Number (4,2) | Not Null |
|  | Overdue\_Rental\_Rate | Number (4,2) | Not Null |
|  |  |  |  |
| Special\_Order\_Detail | *Order\_Number* PK/FK | Number (8) | Not Null |
|  | ***Item\_Number***  **PK**/FK | Number (6) | Not Null |
|  | Quantity | Number (5) | Not Null |
|  | Sales\_Price | Number (7,2) | Not Null |
|  |  |  |  |
| Special\_Order\_Item | Item\_Number PK | Number (6) | Not Null |
|  | Description | Varchar2 (80) | Not Null |
|  | Item\_Cost | Number (7,2) | Null |
|  |  |  |  |
| Special\_Order | Order\_Number PK | Number (8) | Not Null |
|  | *Customer\_Number**FK* | Number (6) | Not Null |
|  | Order\_Date | Date | Not Null |
|  | Received\_Date | Date | Null |
|  | Picked\_Up\_Date | Date | Null |
|  |  |  |  |
| Supplier | **Supplier\_Number PK** | Number (4) | Not Null |
|  | Supplier\_Name | Varchar2 (50) | Not Null |
|  | Active\_YN | Char (1) | Not Null |

Additional notes:

Primary keys are **bolded** and are denoted by PK

Foreign keys are *italicized* and are denoted by FK

All date fields will default to the current date

Picked\_Up\_Date must not be before the Received\_Date

All monetary fields must be greater than zero

Quantity fields cannot be negative

The rental rates will default to 4.40

The overdue rental rates will default to 3.83

Area code must be three digits and will default to 780

Province must be two upper-case characters and will default to Alberta

All YN fields must only accept the upper case value ‘Y’ (Yes) or ’N’ (No)

Current\_Status\_YN and Active\_YN will default to Y

Overdue\_Paid\_YN will default to N

Bar\_Code must be in the format mnnn-zzz (m represents a number from 1 to 9, n represents a number from 0 to 9, zzz represents a number from 001 to 999)

1. Create the following sequences. Ensure that the sequence names are **identical** to what is specified below. **1 mark**

|  |  |  |
| --- | --- | --- |
| Sequence Name | Initial Value | Increment |
| Seq\_Customer | 750 | 10 |
| Seq\_Invoice | 25700 | 1 |
| Seq\_Item | 9000 | 100 |
| Seq\_Order | 500 | 5 |

You can load your tables with the “Data File” which is/will be available from Moodle.

**Part A Lab Submission is to include the following:**

* Ensure your tables and sequences are created within the Oracle DMIT environment (not just on your laptop or home computer)
  + If you have trouble with a particular constraint, **please** contact your instructor
  + Please do not change anything with respect to your tables and sequences after the due date/time until your lab has been **marked and returned**
* An emailed discussion about part A of the lab including:
  + How long the lab took to complete (split between in class and out of class time)
  + Any significant problems that were encountered
  + Whether or not there was enough material provided in the lectures to complete the lab
  + What, if any, changes or improvements could be made to the lab
* One electronic non-compressed text file named Lab1A\_YourLastName\_YourFirstName.sql will be submitted to Moodle
* If you need to re-submit your lab after you have submitted but prior to the due date/time, email your instructor a request to adjust your submission to Draft Mode, then you can submit the revised file
* Do not include your password, the lab specification or the “Data File” in your lab submission
* Any additional requirements as specified by your instructor

**Part B (Due Oct 23rd @ 05:00 p.m.)**

NOTE: A rental is deemed to be **outstanding** if it was returned after the expected due date and the overdue charge has not been paid. A rental is deemed to be **overdue** if it has not been returned and it is past the expected due date.

1. Write a **procedure** called **PR\_Q3** that will be used to insert a record into the rental invoice table. The customer first and last names (in that order) will be specified as parameters. The procedure must work regardless of the case that the names are entered in. The procedure will ensure that appropriate error messages are displayed for the scenarios listed in test cases a, b and c. Use the sequence Seq\_Invoice as the invoice number and the current date and time as the invoice date. Use exception handling in your solution.

For testing purposes, please add "authid current\_user" to your procedure.

**Procedure Code 6 marks, Test cases 2 marks**

Test Cases: Demonstrate your procedure with the following scenarios:

* 1. If there are two customers with the same first name / last name combination (irrespective of case), discontinue processing and display a descriptive error message
  2. If the customer specified doesn’t exist, discontinue processing and display a descriptive error message
  3. If the customer exists and their current status is set to ‘N’, discontinue processing, and display a descriptive error message
  4. If no errors occur, insert a complete record into the rental invoice table

Each test case will consist of three steps:

1. A select **and** display of **appropriate** data from the appropriate table(s) before the procedure is executed
2. Execute your procedure (be sure to include your execute statement)
3. An error message (if applicable) **and** a select **and** display of **appropriate** data from the appropriate table(s) after the procedure is executed
4. Write an overloaded **function** (**FN\_Q4**) inside a **package** (**PKG\_Q4**) that returns a list of the outstanding rental invoice numbers for a specific customer. This function is passed either: i) a customer first and last name (in that order); or ii) a customer number. The package will ensure that appropriate messages are displayed for the scenarios listed in test cases a, b, c and d. Input case is not guaranteed. Use at least one **explicit** cursor in each function and each function must stand alone (i.e. a function must not call the other function).

For testing purposes, please add "authid current\_user" to your **package specification only**.

**Function/Package Code 10 marks, Test cases 4 marks**

Test Cases: Demonstrate your overloaded function with the following scenarios using both sets of parameters:

* 1. If the customer does not exist, return an appropriate message including the customer number or first and last names (i.e. input parameters)
  2. If the customer first name / last name combination is not unique, return an appropriate message including the customer first and last names (parameter set “i” only)
  3. If the customer exists but has no outstanding rentals, return an appropriate message including the customer number, first and last names
  4. If the customer exists and has outstanding rentals, return a message including the customer number, first and last names followed by a list of their outstanding rental invoice numbers

Each test case will consist of two steps:

1. A select **and** display of the **appropriate** data from the appropriate table(s)
2. Call your function and display the results (ensure your call statement is included in your lab submission).
3. Write a **trigger** called **TR\_Q5** that will prevent a customer from renting anything new if they have more than 5 overdue rentals **or** more than 3 outstanding rentals.

**Trigger Code 6 Marks, Test Cases 2 marks**

Test Cases: Test cases displaying both successful **and** unsuccessful completions are required. Each test case will consist of three steps:

1. A select **and** display of the **appropriate** data from the appropriate table(s) before the code is executed
2. Execute your DML statement (ensure your DML statement is included in your lab submission)
3. An error message (if applicable) **and** a select **and** display of the **appropriate** data from the appropriate table(s) after the code is executed

***Part B Lab Submission* *is to include the following:***

* Ensure your procedure, package and trigger are created within the Oracle DMIT environment (not just on your laptop or home computer)
  + If you have trouble with a particular question, **please** contact your instructor
  + Please do not change anything with respect to your procedure, package and trigger after the due date/time until your lab has been **marked and returned**
* A short emailed discussion about the lab including:
  + What you liked and/or disliked about the lab
  + How long the lab took to complete (split between in class and out of class time)
  + Any significant problems that were encountered
  + Whether or not there was enough material provided in the lectures to complete the lab
  + What, if any, changes or improvements could be made to the lab
* Two electronic files will be submitted to Moodle:
  + Your source code as a single text file named Lab1B\_Source\_LastName\_FirstName.sql
  + Your test cases as a single text file named Lab1B\_TestCases\_LastName\_FirstName.sql
  + Compressed files (i.e. zip, rar, etc.) or folders will NOT be accepted
* If you need to re-submit your lab after you have submitted but prior to the due date/time, email your instructor a request to adjust your submission to Draft Mode, then you can submit the revised file
* Ensure there are no rollback, or commit, statements in your code!
* Do not include your password, the lab specification or the “Data File” in your lab submission
* **Select \* is NOT acceptable anywhere in your lab (including test cases and/or commented out code)**
* Any additional requirements as specified by your instructor

***Lab Expectations:***

* All code **MUST** be executable within the Oracle DMIT environment
* Tables, columns, sequences, procedures, functions, packages and triggers **MUST** be named exactly as indicated
* All cursors must be explicitly declared, opened, retrieved from and closed
* All code will be documented, where appropriate, and will follow a convention such as standards.doc (available from Moodle)
* Anything additional as specified by your instructor