Documentation for Library Database and Procedures

Activity tracking is an important part of institution activity. Having efficient and accurate methods of tracking activities such as transactions can help an institution learn about its operational needs, asses productivity and guide its strategic goals. A library like any other institution needs to be able to track patron info, patron activity and the location of library materials. The Library database and accompanying procedures seek to fill this need. The Library database is made up of three tables of which information is drawn to execute four operational procedures. It is the purpose of this library and accompanying procedures to assist the library in tracking day to day activity

Tables

The Library database is made up of three tables(*Table 1)*. The first, Lib_members, contains general information of all library patrons registered for a library card. The first column 'card' is the registered card number associated with the library patron. It is the table's primary key. The table is programmed so every new card registered is automatically assigned a sequential and unique card number, with up to 5 digits. This way no two library card holders will have the same card number and there is no need for library employees to keep track of the last card number issued. This column must have a unique sequential value if other columns in the row are populated. To populate this column the sequence 's_lib_members' was created; every time a new card member is added to the table a new number is generated for the row. The second column in the table, ' mem_name' holds the corresponding full name of the library patron associated with the registered card. The column allows for a max of string size of 30 character to be entered. The column ' phone' contains a twelve-character string representing the provided phone number of the library patron at the time of registration. The column, ' DOB' contains the provided date of birth of the library patron. The last column contains a password of up to 40 characters in length. This password is provided by the library patron at the time of card registration. The password is used by the library patron to access their online library profile.

The Lib_books table contains details about library books in circulation. There are three are four columns. The first 'isnb' contains the 13 character string representing the books isnb number. This is the tables primary key and it must contain a unique value. The 'title' column holds the title of the book, a string up to 40 characters in length. The author column holds the author's full name and can accept a string value of up to 25 characters in length. The genre column is used to identify the genre of the book . It can hold a string up to 20 characters in length.

The Lib_rentals table has six columns with rental details. This table holds these rental details until a book has been returned. If a book is past due it will remain on this table until it is returned. The first column of this table and the primary key, rental_id is an automatic number created each time a book is rented. Each book a patron rents generates a new renatal_id. This is done to avoid checking in all books when only some of the books rented have been returned. The rental_id is generated using the s_lib-rentals sequence which will provide a sequential unique number for this column. The isnb holds the isnb value of the rented book and references the isnb column in table Lib_books. The r_card columns holds the card number of the patron borrowing the book and references the card column in table Lib_members. The rented_date column stores the date in which the book was rented. The due date stores a date 21 days from the rented_date column.

Procedures

The data of each of the three tables will be used to track day to day operations of the library in the way of four procedures: Login, Overview, Rent, Return. Login procedure is used by the user to logon their online library profile. This procedure will first ask the user to enter their card number and password. This info will then be passed the login procedure which will in order check if the card number provide is valid, check if the password is valid and return a completion statement. If the card number entered cannot be matched to a card number in the card column of table Lib_members then the procedure will skip checking if the password is correct to the wrong_user exception which will print out the message, 'Wrong username', to let the user know the card number entered is not valid. If the card number entered is valid the procedure will then check if the password entered matches the password in the mem_pswrd column of table Lib_members. If the password does not match the procedure will go to the wrong_pass exception which will print out a message, ‘Wrong password', that lets the user know the password is invalid. If the password entered matches the password in the mem_pswrd column then the procedure will print, 'Login Successful'.

The Overview SQL procedure will print for either a member or a library employee the name, phone number and dob of the customer whose card number they entered. The user will first be asked to enter a card number and this number will then be passed to the Overview procedure. The procedure will then check if the provided card number exists in the card column of the Lib_member table. If the card number does not exist inn this table, the procedure will skip to the no_data_found exception which will print, 'Enter a valid customer number.', letting the user know the entered card number is invalid. If the card number is valid then procedure selects mem_name, phone, dob from the Lib_members table and sets them to local variables m_name, m_phone, and m_dob, respectively. The values of these variables will then be printed for the user. The Overview Java Procedure has the same function. The biggest difference being that the user is prompted for to enter their DePaul username and password at the beginning of the class in order to create the connection. Afterwards the user is prompted for a card number for which the mem_name, phone, and dob of the Lib_members table are selected and printed.

The procedure rent keeps track of the books being rented. First the library employee is asked to enter the name of the customer renting and the isnb of the book being borrowed. These values are then passed to the rent procedure. The procedure first checks if the entered customer number can be found in the card column of the Lib_members tables. If it is invalid then the procedure skips the rest of the procedure, to the no_data_found exception and prints, ‘Enter a valid customer number and or book isnb.. If the card number is valid then then the procedure checks if the entered isnb is valid. If it not valid then the procedure skips the rest of the procedure to the no_data_found exception and prints, 'Enter a valid customer number and or book isnb. If the isnb is valid then the procedure will print a message with the title name and due date of the book being rented.

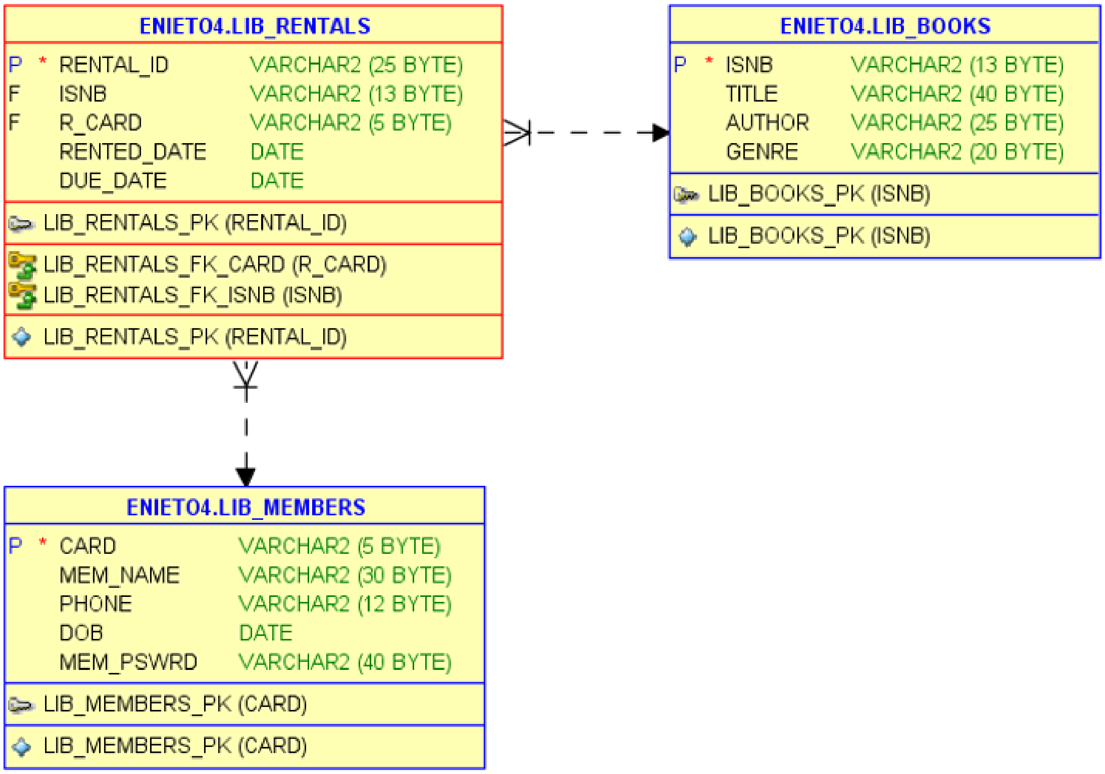
The procedure return keeps track of returned books and fines when applicable. When a book is returned on time the corresponding rental id is removed from the Lib_rentals table. When a book is returned past the due date then a fee is calculated, and the rental id remains in the Lib_rentals table. First the user is asked for a card number and isnb. Similar to the rent procedure the return procedure checks if both entries are valid. Using these two inputs the procedure gets the rental ID to check the due date. If the book is being returned past the due date, the procedure calculates the number of days late the book is and then multiplies this number by $.30. So for every day the book is late the patron is charged $.30. the procedure then prints a statement for the user letting them know how many days late the book is and how much is due in fees. If the book is not late the procedure prints the message, 'Book has been returned. Thank you.', and the procedure is ended.

Conclusion

The library database developed using the Oracle Database 11g by J. Price(2008) and the website Geeks for geeks and can be used by a library to track patron activity and library material. The Login procedure helps a library patron login to their online profile. The Overview procedure helps the patron or library employee view a patron's general information. The Rent procedure allows an employee to check out a book for a patron and track this information on the Lib_rentals table. Using this same table The Return procedure removes information from the table once a library book is returned on time; it also uses the table to calculate a late fee if the material is returned past the due date. These four procedures in combination with the three library database tables can be used to track library day to day activity. By tracking this information the library will be able to learn about its operational needs, asses productivity and guide its strategic goals.

Appendix

Table 1: Library Database Model



ENIETO4.LIB_RENTALS

P	*	RENTAL_ID	VARCHAR2 (25 BYTE)
F		ISNB	VARCHAR2 (13 BYTE)
F		R_CARD	VARCHAR2 (5 BYTE)
		RENTED_DATE	DATE
		DUE_DATE	DATE

LIB_RENTALS_PK (RENTAL_ID)

LIB_RENTALS_FK_CARD (R_CARD)
LIB_RENTALS_FK_ISNB (ISNB)

LIB_RENTALS_PK (RENTAL_ID)

ENIETO4.LIB_BOOKS

P	*	ISNB	VARCHAR2 (13 BYTE)
		TITLE	VARCHAR2 (40 BYTE)
		AUTHOR	VARCHAR2 (25 BYTE)
		GENRE	VARCHAR2 (20 BYTE)

LIB_BOOKS_PK (ISNB)

LIB_BOOKS_PK (ISNB)

ENIETO4.LIB_MEMBERS

P	*	CARD	VARCHAR2 (5 BYTE)
		MEM_NAME	VARCHAR2 (30 BYTE)
		PHONE	VARCHAR2 (12 BYTE)
		DOB	DATE
		MEM_PSWRD	VARCHAR2 (40 BYTE)

LIB_MEMBERS_PK (CARD)

LIB_MEMBERS_PK (CARD)

Table 1: This figure decpics the Lib_rentals, Lib_books, and Lib_members tables and how they are connected in the Library database.

References

Price, J. (2008). *Oracle database 11g Sql*.

Techonthenet.com. Oracle / PLSQL: Insert a date/time value into an Oracle table. Retrieved from https://www.techonthenet.com/oracle/questions/insert_date.php