****

**COMP741 – Applications Development**

**Library Management System**

**Diploma in Software Development Level 7**

**Rodrigo Massami Tamura Adati 17018158c@cornell.ac.nz**

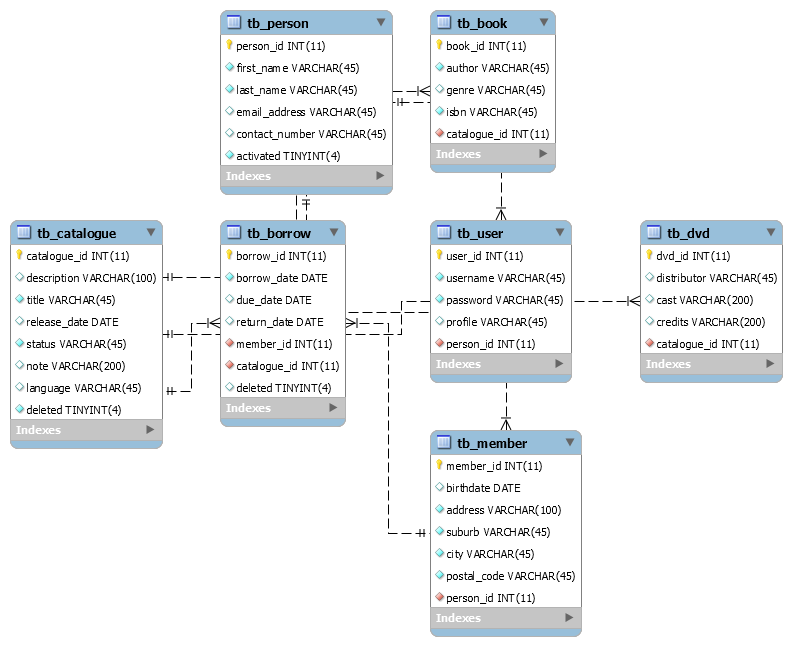
**Romeric Heredia** [**17018054c@cornell.ac.nz**](mailto:17018054c@cornell.ac.nz)

Contents

[ERD Diagram 3](#_Toc491926821)

[Stored Procedures 4](#_Toc491926822)

# ERD Diagram



# Stored Procedures

|  |  |
| --- | --- |
| Name | sp\_addBook |
| Description | Insert a new book item |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_addBook`(author varchar(45), genre varchar(45), isbn varchar(45), title varchar(45), description varchar(100), release\_date date, status varchar(45), note varchar(45), language varchar(45))  BEGIN  DECLARE pk INT;  insert into tb\_catalogue(description, title, release\_date, status, note, language, deleted) values (description, title, release\_date, status, note, language, 0);  select last\_insert\_id() into pk;  insert into tb\_book(author, genre, isbn, catalogue\_id) values (author, genre, isbn, pk);  END |

|  |  |
| --- | --- |
| Name | sp\_addBorrow |
| Description | Insert borrow information about the items borrowed by the members |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_addBorrow`(borrow Date, due Date, retDate Date, memberId int, catalogueId int)  BEGIN  insert into tb\_borrow(borrow\_date, due\_date, return\_date, member\_id, catalogue\_id, deleted) values (borrow, due, retDate, memberId, catalogueId, 0);  update tb\_catalogue  set status = 'Borrowed'  where catalogue\_id = catalogueId;  END |

|  |  |
| --- | --- |
| Name | sp\_addDVD |
| Description | Insert a new DVD item |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_addDvd`(distributor varchar(45), cast varchar(200), credits varchar(200),title varchar(45), description varchar(100), release\_date date, status varchar(45), note varchar(45), language varchar(45))  BEGIN  DECLARE pk INT;  insert into tb\_catalogue(description, title, release\_date, status, note, language, deleted) values (description, title, release\_date, status, note, language,0);  select last\_insert\_id() into pk;  insert into tb\_dvd(distributor, cast, credits, catalogue\_id) values (distributor, cast, credits, pk);  END |

|  |  |
| --- | --- |
| Name | sp\_addMember |
| Description | Insert a new member |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_addMember`(first\_name varchar(100), last\_name varchar(100), email\_address varchar(45), contact\_number varchar(45), birthdate Date, address varchar(100), suburb varchar(100), city varchar(45), postal\_code varchar(45))  BEGIN  DECLARE pk INT;  insert into tb\_person(first\_name, last\_name, email\_address, contact\_number, activated) values (first\_name, last\_name, email\_address, contact\_number, 0);  select last\_insert\_id() into pk;  insert into tb\_member(birthdate, address, suburb, city, postal\_code, person\_id) values (birthdate, address, suburb, city, postal\_code, pk);  END |

|  |  |
| --- | --- |
| Name | sp\_addUser |
| Description | Insert a new user |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_addUser`(first\_name varchar(100), last\_name varchar(100), email\_address varchar(45), contact\_number varchar(45), username varchar(45), password varchar(45))  BEGIN  DECLARE pk INT;  insert into tb\_person(first\_name, last\_name, email\_address, contact\_number, activated) values (first\_name, last\_name, email\_address, contact\_number, 0);  select last\_insert\_id() into pk;  insert into tb\_user(username, password, person\_id) values (username, password, pk);  END |

|  |  |
| --- | --- |
| Name | sp\_checkPassword |
| Description | Check if the user password is correct |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_checkPassword`(usern varchar(45), pass varchar(45))  BEGIN  select \* from tb\_user  where username = usern and password = pass;  END |

|  |  |
| --- | --- |
| Name | sp\_deactivateMember |
| Description | Deactivate the selected member from the application |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_deactivateMember`(id int)  BEGIN  Update tb\_person  set activated = 1  where person\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_deleteBook |
| Description | Logical delete of the book item |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_deleteBook`(id int)  BEGIN  Update tb\_catalogue  set deleted = 1  where catalogue\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_deleteDVD |
| Description | Logical delete of dvd item |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_deleteDvd`(id int)  BEGIN  Update tb\_catalogue  set deleted = 1  where catalogue\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_resetPassword |
| Description | Set a default password for the selected user |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_resetPassword`(id int)  BEGIN  Update tb\_user  set password = '123'  where person\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_returnItem |
| Description | Return the selected item |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_returnItem`(borrow Date, due Date, retDate Date, memberId int, catalogueId int)  BEGIN  insert into tb\_borrow(borrow\_date, due\_date, return\_date, member\_id, catalogue\_id, deleted)  values (borrow, due, retDate, memberId, catalogueId, 0);  update tb\_catalogue  set status = 'Available'  where catalogue\_id = catalogueId;  END |

|  |  |
| --- | --- |
| Name | sp\_searchAvailableBook |
| Description | Get the list of available books |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchAvailableBook`()  BEGIN  select b.\*, c.\*  from tb\_book b inner join tb\_catalogue c  on b.catalogue\_id = c.catalogue\_id  where c.deleted = 0 and c.status = 'Available';  END |

|  |  |
| --- | --- |
| Name | sp\_searchAvailableBookByTitle |
| Description | Get the list of available books filtered by the title |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchAvailableBookByTitle`(title varchar(45))  BEGIN  select b.\*, c.\*  from tb\_book b inner join tb\_catalogue c  on b.catalogue\_id = c.id  where c.title like CONCAT(title,'%') and  c.deleted = 0 and c.status = 'Available';  END |

|  |  |
| --- | --- |
| Name | sp\_searchAvailableDvd |
| Description | Get the list of available dvds |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchAvailableDvd`()  BEGIN  select d.\*, c.\*  from tb\_dvd d inner join tb\_catalogue c  on d.catalogue\_id = c.catalogue\_id  where c.deleted = 0 and c.status = 'Available';  END |

|  |  |
| --- | --- |
| Name | sp\_searchAvailableDvdByTitle |
| Description | Get the list of available dvds filtered by the title |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchAvailableDvdByTitle`(title varchar(45))  BEGIN  select d.\*, c.\*  from tb\_dvd d inner join tb\_catalogue c  on d.catalogue\_id = c.catalogue\_id  where c.title like CONCAT(title,'%') and  c.deleted = 0 and c.status = 'Available';  END |

|  |  |
| --- | --- |
| Name | sp\_searchBook |
| Description | Get the list of books |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchBook`()  BEGIN  select b.\*, c.\*  from tb\_book b inner join tb\_catalogue c  on b.catalogue\_id = c.catalogue\_id  where c.deleted = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchBookByTitle |
| Description | Get the list of books filtered by title |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchBookByTitle`(title varchar(45))  BEGIN  select b.\*, c.\*  from tb\_book b inner join tb\_catalogue c  on b.catalogue\_id = c.id  where c.title like CONCAT(title,'%') and  c.deleted = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchBorrowedItem |
| Description | Get the list of borrowed items |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchBorrowedItem`(name varchar(45))  BEGIN  select distinct tb\_catalogue.\* from tb\_catalogue  inner join tb\_borrow  on tb\_catalogue.catalogue\_id = tb\_borrow.catalogue\_id  inner join tb\_member  on tb\_member.member\_id = tb\_borrow.member\_id  inner join tb\_person  on tb\_person.person\_id = tb\_member.person\_id  where concat(tb\_person.first\_name, ' ' , tb\_person.last\_name) = name;  END |

|  |  |
| --- | --- |
| Name | sp\_searchDvd |
| Description | Get the list of dvds |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchDvd`()  BEGIN  select d.\*, c.\*  from tb\_dvd d inner join tb\_catalogue c  on d.catalogue\_id = c.catalogue\_id  where c.deleted = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchDvdByTitle |
| Description | Get the list of dvds filtered by title |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchDvdByTitle`(title varchar(45))  BEGIN  select d.\*, c.\*  from tb\_dvd d inner join tb\_catalogue c  on d.catalogue\_id = c.catalogue\_id  where c.title like CONCAT(title,'%') and  c.deleted = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchMember |
| Description | Get the list of members |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchMember`()  BEGIN  select p.\*, m.\*  from tb\_member m inner join tb\_person p  on p.person\_id = m.person\_id  where p.activated = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchMemberByName |
| Description | Get the list of members filtered by name |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchMemberByName`(name varchar(100))  BEGIN  select p.\*, m.\*  from tb\_member m inner join tb\_person p  on p.person\_id = m.person\_id  where p.first\_name like CONCAT(name,'%') or p.last\_name like CONCAT(name,'%') and p.activated = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchMemberId |
| Description | Get the id of the member |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchMemberId`(name varchar(100))  BEGIN  select m.member\_id from tb\_person p inner join tb\_member m  on p.person\_id = m.person\_id  where concat(p.first\_name, ' ' , p.last\_name) = name;  END |

|  |  |
| --- | --- |
| Name | sp\_searchPerson |
| Description | Get the list of persons that are active |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchPerson`()  BEGIN  select p.person\_id, Concat(p.first\_name, ' ', p.last\_name) as name  from tb\_member m inner join tb\_person p  on p.person\_id = m.person\_id  where p.activated = 0;  END |

|  |  |
| --- | --- |
| Name | sp\_searchTransactionHistory |
| Description | Get the list of transaction history of the member |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchTransactionHistory`(name varchar(100))  BEGIN  select c.title, b.due\_date, b.borrow\_date , concat(p.first\_name,' ',p.last\_name) as name  from tb\_borrow b  inner join tb\_catalogue c  on b.catalogue\_id = c.catalogue\_id  inner join tb\_member m  on b.member\_id = m.member\_id  inner join tb\_person p  on m.person\_id = p.person\_id  where concat(p.first\_name, ' ', p.last\_name)= name;  END |

|  |  |
| --- | --- |
| Name | sp\_searchUser |
| Description | Get the information of the active user |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchUser`()  BEGIN  select p.\*, u.\*  from tb\_user u inner join tb\_person p  on p.person\_id = u.person\_id  where p.activated = 0 and u.profile = 'user';  END |

|  |  |
| --- | --- |
| Name | sp\_searchUserByName |
| Description | Get the activate users filtered by name |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_searchUserByName`(name varchar(100))  BEGIN  select p.\*, u.\*  from tb\_user u inner join tb\_person p  on p.person\_id = u.person\_id  where (p.first\_name like CONCAT(name,'%') or p.last\_name like CONCAT(name,'%')) and p.activated = 0 and u.profile = 'user';  END |

|  |  |
| --- | --- |
| Name | sp\_updateBook |
| Description | Update the information of the selected book |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_updateBook`(id int, author varchar(45), genre varchar(45), isbn varchar(45), title varchar(45), description varchar(100), release\_date date, status varchar(45), note varchar(45), language varchar(45))  BEGIN  update tb\_book  set author = author,  genre = genre,  isbn = isbn  where catalogue\_id = id;    update tb\_catalogue  set description = description,  title = title,  release\_date = release\_date,  status = status,  note = note,  language = language  where catalogue\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_updateDvd |
| Description | Update the information of the selected Dvd |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_updateDvd`(id int, distributor varchar(45), cast varchar(45), credits varchar(45), title varchar(45), description varchar(100), release\_date date, status varchar(45), note varchar(45), language varchar(45))  BEGIN  update tb\_dvd  set distributor = distributor,  cast = cast,  credits = credits  where catalogue\_id = id;    update tb\_catalogue  set description = description,  title = title,  release\_date = release\_date,  status = status,  note = note,  language = language  where catalogue\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_updateMember |
| Description | Update the information of the selected member |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_updateMember`(id int, first\_name varchar(100), last\_name varchar(100), email\_address varchar(45), contact\_number varchar(45), birthdate Date, address varchar(100), suburb varchar(100), city varchar(45), postal\_code varchar(45))  BEGIN  update tb\_person  set first\_name = first\_name,  last\_name = last\_name,  email\_address = email\_address ,  contact\_number = contact\_number  where person\_id = id;    update tb\_member  set birthdate = birthdate,  address = address,  suburb = suburb,  city = city,  postal\_code = postal\_code  where person\_id = id;  END |

|  |  |
| --- | --- |
| Name | sp\_updatePassword |
| Description | Update the password of the member |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_updatePassword`(usern varchar(45), pass varchar(45), newPassword varchar(45))  BEGIN  update tb\_user  set password = newPassword  where username = usern and password = pass;  END |

|  |  |
| --- | --- |
| Name | sp\_updateUser |
| Description | Update the user information of the selected user |
| Script | CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_updateUser`(id int, first\_name varchar(100), last\_name varchar(100), email\_address varchar(45), contact\_number varchar(45), username varchar(100))  BEGIN  update tb\_person  set first\_name = first\_name,  last\_name = last\_name,  email\_address = email\_address,  contact\_number = contact\_number  where person\_id = id;    update tb\_user  set username = username  where person\_id = id;  END |