Unit 5 Assignment Solution

The following commands create the librarian table. Not null constraints are applied by adding the not null clause to the column name, the unique constraint is applied by adding the unique clause to the column name. The primary key is identified by adding a primay key constraint.

```
create table librarian (
        librarianid integer not null,
        name char(40) not null,
        phonenumber char(20),
        supervisor integer
);

alter table librarian add unique (librarianid);

alter table librarian add primary key (librarianid);
```

The following commands create the borrower table. In this table we show an example of using a sequence implemented with the 'IDENTITY' clause to automatically generate a library card number. The Not null constraint is applied by adding the not null clause to the column name, I did not implement the unique constraint because the primary key of librarycard number is being automatically generated which by default is unique.

```
create table borrower (
        librarycard integer GENERATED BY DEFAULT AS IDENTITY,
        name char(40) not null,
        address char(40),
        postalcode char(20),
        phonenumber char(20),
        membershipdate date
);
```

The following commands create the book table. Not null constraints are applied by adding the not null clause to the column name, the unique constraint is applied by adding the unique clause to the column name. The primary key is identified by adding a primay key constraint.

```
create table book (
        title varchar(60) not null,
        author varchar(40),
        cost decimal(10,2),
        isbn_number integer not null
);
```

```
alter table book add unique (isbn_number);

alter table book add primary key (isbn_number);
```

The following commands create the bookcopy table. Not null constraints are applied by adding the not null clause to the column name. Both the primary key and unique constraints are applied with constraint clauses that are added with the alter statement. Referential integrity with the book table is enforced by adding a FOREIGN KEY constraint clause using the alter table statement. Note the use of ON DELETE cascade clause which means that when the table for which the foreign key is primary key deletes a row, the delete is cascaded to referenced tables.

```
create table bookcopy (
        isbn_number     integer not null,
        sequence        integer not null,
        publicationdate date
);

alter table bookcopy add primary key (isbn_number, sequence);

alter table bookcopy add CONSTRAINT  bookcopy_uniq UNIQUE (ISBN_number, sequence);

alter table bookcopy add CONSTRAINT book_fkey FOREIGN KEY (isbn_number) REFERENCES  book
(isbn_number) ON DELETE CASCADE;
```

The following commands create the booklended table. Not null constraints are applied by adding the not null clause to the column name. The primary key and unique constraints are applied with constraint clauses. Referential integrity via the foreign keys is implemented using a FOREIGN KEY constraint for each of the referenced tables including borrower, librarian, and bookcopy tables.

```
create  table  booklended (
        librarycard integer not null,
        checkoutdate date not null,
        returndate date,
        ISBN_number integer not null,
        sequence  integer  not null,
        librarianid integer not null
);

alter table booklended add PRIMARY KEY (ISBN_number, sequence, librarycard,  checkoutdate);

alter table booklended add CONSTRAINT  booklended_uniq UNIQUE (ISBN_number, sequence,
librarycard, checkoutdate);
```

```
alter table booklended add CONSTRAINT borrower_fkey FOREIGN KEY (librarycard) REFERENCES
borrower (librarycard) ON DELETE CASCADE;

alter table booklended add CONSTRAINT librarian_fkey FOREIGN KEY (librarianid) REFERENCES librarian
(librarianid) ON DELETE CASCADE;

alter table booklended add CONSTRAINT bookcopy_fkey FOREIGN KEY (isbn_number, sequence)
REFERENCES bookcopy (isbn_number, sequence);
```