Andrew DiCarlo
ebq929
CS 3743 - 002

```
CREATE TABLE BRANCHES(BID CHAR(6) PRIMARY KEY,
                      BNAME VARCHAR(64) UNIQUE NOT NULL,
                      BADDRESS VARCHAR(512),
                      BPHONE NUMERIC(10),
                      BFAX NUMERIC(10)
                      );
CREATE TABLE STORES_B(ISBN CHAR(10),
                      BINDEX INT,
                      BID CHAR(6),
                      PRIMARY KEY(ISBN, BINDEX),
                      FOREIGN KEY (BID) REFERENCES BRANCHES(BID),
                      FOREIGN KEY (ISBN) REFERENCES BOOKS(ISBN)
                      );
CREATE TABLE BOOKS(ISBN CHAR(10) PRIMARY KEY,
                      BTITLE VARCHAR(512),
                      BAUTHOR VARCHAR(64),
                      DDS CHAR(14),
                      BSUBJECT CHAR(32),
                      BGENRE CHAR(24),
                      BLANGUAGE VARCHAR(64),
                      BDESC VARCHAR(4096),
                      BPUB VARCHAR(64),
                      BYEAR DATE
                      );
CREATE TABLE PERIODICALS(PNAME VARCHAR(64),
                      PEDITION VARCHAR(64),
                      PGENRE CHAR(24),
                      PTYPE CHAR(16),
                      DDS CHAR(14),
                      PSUBJECT CHAR(32),
                      PRIMARY KEY(PNAME, PEDITION)
                      );
CREATE TABLE STORES_P(PNAME VARCHAR(64),
                      PEDITION VARCHAR(64),
                      BID CHAR(6),
                      FOREIGN KEY (PNAME, PEDITION) REFERENCES
                      PERIODICALS (PNAME, PEDITION),
                      FOREIGN KEY (BID) REFERENCES BRANCHES(BID),
                      PRIMARY KEY(PNAME, PEDITION, BID)
                      );
CREATE TABLE CUSTOMERS(UNAME VARCHAR(32) PRIMARY KEY,
                      CARD NUMERIC(12) UNIQUE NOT NULL,
                      EMAIL VARCHAR(256),
                      CADDRESS VARCHAR(512),
                      CDOB DATE,
                      CPHONE NUMERIC(10),
                      CSTANDING CHAR(7) CHECK(CSTANDING='GOOD' OR
                      CSTANDING='OVERDUE') DEFAULT 'GOOD',
```

```
                    CFINES NUMERIC(9, 2) CHECK(CFINES >= 0) DEFAULT 0
                    );
CREATE TABLE CHECKED_OUT(UNAME VARCHAR(32),
                    ISBN CHAR(10),
                    BINDEX INT,
                    RENEWALS INT DEFAULT 2,
                    DUE DATE NOT NULL,
                    PRIMARY KEY (ISBN, BINDEX),
                    FOREIGN KEY (ISBN, BINDEX) REFERENCES
                    STORES_B(ISBN, BINDEX),
                    FOREIGN KEY (UNAME) REFERENCES CUSTOMERS(UNAME)
                    );
```

- Get list of all branch names
  - SELECT BNAME FROM BRANCHES;
- Lookup all information about a branch using the BID, which will be substituted in for {ID_TO_FIND}
  - SELECT *
    FROM BRANCHES
    WHERE BID='{ID_TO_FIND}';
- Search for all books with a title that matches the keyword and sort by title. Sort order can be modified by changing the column to order by.
  - SELECT * FROM BOOKS WHERE BTITLE LIKE '%KEYWORD%' ORDER BY BTITLE;
- Get the names of all branches that have a copy of a particular book with an ISBN that matches {ISBN}.
  - SELECT B.BNAME
    FROM BRANCHES B
    WHERE EXISTS (
        SELECT *
        FROM STORES_B S
        WHERE S.ISBN='{ISBN}' AND S.BID=B.BID
    );
- Get the names of branches that have a copy of the book with an ISBN matching {ISBN}, along with the number of copies of that book at that location.
  - SELECT B.BNAME, COUNT(*)
    FROM (
      SELECT *
      FROM STORES_B S JOIN BOOKS B ON S.ISBN = B.ISBN
) AS B_TO_P JOIN BRANCHES B ON B.BID=B_TO_P.BID
    WHERE B_TO_P.ISBN='{ISBN}'
    GROUP BY B.BID, B_TO_P.ISBN;
- Get a list of all different periodical names, sorted by name.
  - SELECT DISTINCT PNAME
    FROM PERIODICALS
    ORDER BY PNAME;
- Get all available editions of a periodical identified by {PERIODICAL_NAME}.
  - SELECT PEDITION
    FROM PERIODICALS
    WHERE PNAME='{PERIODICAL_NAME}';
- Get the names of all branches that have a particular periodical identified by {PNAME} and {PEDITION}
  - SELECT B.BNAME
    FROM BRANCHES B
    WHERE EXISTS (
        SELECT *
        FROM STORES_P S
        WHERE S.PNAME='{PNAME}' AND S.PEDITION='{PEDITION}' AND S.BID=B.BID
    );
- Get the names of all periodicals that are available at a particular branch identified by {BID}.

    - SELECT DISTINCT PNAME
      FROM STORES_P
      WHERE BID='{BID}';
- Update the fines due by a particular customer. This is meant to be run by the system every day, and adds $3 to the customers overall fines due for every book checked out by that customer that is past the due date.
    - UPDATE CUSTOMERS
      SET CFINES = CFINES + 3 * (
        SELECT COUNT(C1.UNAME) COUNTS
        FROM CUSTOMERS C1 JOIN (
          SELECT *
          FROM CHECKED_OUT
          WHERE DUE < DATE('NOW', 'LOCALTIME')
) CO
        ON C1.UNAME=CO.UNAME GROUP BY C1.UNAME
      );
- Allow a customer to pay fines by subtracting {AMOUNT} from the account with the username {USERNAME}.
    - UPDATE CUSTOMERS
      SET CFINE = CFINES - {AMOUNT}
      WHERE UNAME='{USERNAME}';
- Update the standing of a customer based on whether they have books that are past due. Similar to how CFINES are handled, this is meant to be run on a regular schedule.
    - UPDATE CUSTOMERS
      SET CSTANDING = CASE
        WHEN UNAME IN (
          SELECT DISTINCT C1.UNAME
          FROM CUSTOMERS C1 INNER JOIN (
            SELECT *
            FROM CHECKED_OUT
            WHERE DUE < DATE('NOW', 'LOCALTIME')
) CO ON C1.UNAME=CO.UNAME) THEN 'OVERDUE'
        ELSE 'GOOD'
        END;
- Get information about all books checked out by the user with {USERNAME}.
    - SELECT *
      FROM (CHECKED_OUT C JOIN BOOKS B ON C.ISBN=B.ISBN)
      WHERE C.UNAME='{USERNAME}';
- Renew one book checked out by the user {USERNAME}, identified by ISBN contained in {ISBN}. This is done by finding all rows in CHECKED_OUT that match the conditions and adding 14 days to them and decrementing the number of renewals.
    - UPDATE CHECKED_OUT
      SET DUE=DATE(DUE, '+14 DAYS'),
      RENEWALS = RENEWALS – 1
      WHERE RENEWALS > 0 AND UNAME='{USERNAME}' AND ISBN ='{ISBN}';
- Renew all books checked out by the user {USERNAME}. Like above, this is done by adding 14 days to the due date and decrementing the number of renewals.
    - UPDATE CHECKED_OUT

SET DUE=DATE(DUE, '+14 DAYS'), RENEWALS = RENEWALS - 1
WHERE RENEWALS > 0 AND UNAME='{USERNAME}';

- Get the name and edition of every periodical available by branch, which is referenced by name.
  - SELECT B.BNAME, B_TO_P.PNAME, B_TO_P.PEDITION
    FROM (
        SELECT *
        FROM STORES_P S JOIN PERIODICALS P
        ON S.PNAME = P.PNAME AND S.PEDITION = P.PEDITION
) AS B_TO_P JOIN BRANCHES B ON B.BID=B_TO_P.BID;
- Get the information for the customer with the username that matches {USERNAME}
  - SELECT * FROM CUSTOMERS WHERE UNAME={USERNAME};
- Update a customer’s information.
  - UPDATE CUSTOMERS
    SET CARD={NEW_CARD}, EMAIL='{NEW_EMAIL}',
    CADDRESS='{NEW_ADD}', CDOB='{NEW_DOB}', CPHONE={NEW_PHONE}
    WHERE UNAME=’{USER}’;