Carl Hartry Jr
Project 2(Beer.db)
4710 Databases
Professor Hong
11/13/2023

Database Project Report

Introduction

The purpose of this project is to design and create a SQLite3 database for managing information related to bars, beers, brewers, and drinkers. The project includes the creation of tables for independent entities and bridge entities, as well as the population of data into the database.

Database Schema

I. BAR
- Attributes: BAR_ID (Primary Key), NAME, ADDRESS
- Constraints: NAME and ADDRESS are unique.

II. BEER
- Attributes: BEER_ID (Primary Key), NAME
- Constraint: NAME is unique.

III. BREWER
- Attributes: BREWER_ID (Primary Key), NAME, ADDRESS
- Constraint: NAME is unique

IV. DRINKER
- Attributes: DRINKER_ID (Primary Key), FNAME, LNAME, ADDRESS
- Constraint: A combination of FNAME and LNAME is unique.

The bridge entities include SERVES, BREWS, VISITS, and LIKES, which establish relationships between the independent entities.

Data Population

I have populated the database with sample data for the entities.
Independent:
BAR - includes three bars with their names and addresses.
BEER - includes four beer names.
BREWER - includes information about three brewers and their location of operations.
DRINKER - includes details of four individuals, including their first names, last names, and addresses.

Dependent(Bridge):
BREWS - specifying which brewer produces which beer.
VISITS - indicating how often a person visits a particular bar.

SERVES - showing the price of a specific beer at a particular bar.
LIKES - revealing which beers a person likes

Scripts

I have created two scripts to complete two independent actions.
INIT.SQL - To initialize and populate the Database.
TEST.SQL - To test accuracy and generate sample data.

Actual Schema from Terminal

```
CREATE TABLE BAR(
    BAR_ID INTEGER PRIMARY KEY,
    NAME TEXT NOT NULL,
    ADDRESS TEXT NOT NULL,
    UNIQUE(NAME),
    UNIQUE(ADDRESS)
);
CREATE TABLE BEER(
    BEER_ID INTEGER PRIMARY KEY,
    NAME TEXT NOT NULL,
    UNIQUE(NAME)
);
CREATE TABLE BREWER(
    BREWER_ID INTEGER PRIMARY KEY,
    NAME TEXT NOT NULL,
    ADDRESS TEXT NOT NULL,
    UNIQUE(NAME)
);
CREATE TABLE DRINKER(
    DRINKER_ID INTEGER PRIMARY KEY,
    FNAME TEXT NOT NULL,
    LNAME TEXT NOT NULL,
    ADDRESS TEXT,
    UNIQUE(FNAME,LNAME)
);
CREATE TABLE SERVES(
    SERVES_ID INEGER PRIMARY KEY,
    BEER_SOLD INTEGER,
    BAR_SELLING INTEGER,
    PRICE NUMERIC(3,2) NOT NULL,
    FOREIGN KEY (BEER_SOLD) REFERENCES BEER(BEER_ID) ON DELETE
CASCADE,
    FOREIGN KEY (BAR_SELLING) REFERENCES BAR(BAR_ID) ON DELETE
CASCADE,
    CHECK(PRICE >= 0.0)
);
CREATE TABLE BREWS(
    BREWS_ID INTEGER PRIMARY KEY,
    PRODUCT INEGER,
    VENDER INTEGER,
    FOREIGN KEY (VENDER) REFERENCES BREWER(BREWER_ID) ON DELETE
CASCADE,
    FOREIGN KEY (PRODUCT) REFERENCES BEER(BEER_ID) ON DELETE CASCADE
);
CREATE TABLE VISITS(
    VISITS_ID INEGER PRIMARY KEY,
    PERSON INTEGER,
    BAR_VISITED INTEGER,
    TIMES_A_WEEK INTEGER NOT NULL,
    FOREIGN KEY (PERSON) REFERENCES DRINKER(DRINKER_ID) ON DELETE
CASCADE,
    FOREIGN KEY (BAR_VISITED) REFERENCES BAR(BAR_ID) ON DELETE
CASCADE,
    CHECK(TIMES_A_WEEK > 0)
);
CREATE TABLE LIKES(
    PREFERS_ID INEGER PRIMARY KEY,
    PERSON INEGER,
    BEER_CHOICE INTEGER,
    FOREIGN KEY (PERSON) REFERENCES DRINKER(DRINKER_ID) ON DELETE
CASCADE,
    FOREIGN KEY (BEER_CHOICE) REFERENCES BEER(BEER_ID) ON DELETE
CASCADE
);
```