Assessment 2

Database Management Systems

Sreehari P Sreedhar CB.SC.I5DAS20032

In [ ]:
```
import os

try:
    os.remove('../Dumps/Assessment2.db')
except FileNotFoundError:
    pass
```

In [ ]:
```
import sqlite3
```

In [ ]:
```
conn = sqlite3.connect('../Dumps/Assessment2.db')

cur = conn.cursor()
```

In [ ]:
```
createTables = '''
    BEGIN;

    CREATE TABLE SAILORS (
        SID INTEGER PRIMARY KEY,
        SNAME TEXT,
        RATING INTEGER,
        AGE CHAR(2)
    );

    CREATE TABLE BOATS (
        BID INTEGER PRIMARY KEY,
        BNAME TEXT,
        COLOR TEXT
    );

    CREATE TABLE RESERVES (
        SID INTEGER,
```

```
        BID INTEGER,
        DAY DATE,
        PRIMARY KEY (SID, BID, DAY),
        FOREIGN KEY (SID) REFERENCES SAILORS (SID),
        FOREIGN KEY (BID) REFERENCES BOATS (BID)
    );

    COMMIT;
'''
```

In [ ]:
```
cur.executescript(createTables)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
insertValues = '''
    BEGIN;

    INSERT INTO SAILORS VALUES(22,'DUSTIN',7,45);

    INSERT INTO SAILORS VALUES(29,'BRUTUS',1,3);

    INSERT INTO SAILORS VALUES(31,'LUBBER',8,5);

    INSERT INTO SAILORS VALUES(32,'ANDY',8,25);

    INSERT INTO SAILORS VALUES(58,'RUSTY',10,35);

    INSERT INTO SAILORS VALUES(64,'HORATIO',7,35);

    INSERT INTO SAILORS VALUES(71,'ZORBA',10,16);

    INSERT INTO SAILORS VALUES(74,'HORATIO',9,40);

    INSERT INTO SAILORS VALUES(85,'ART',3,25);

    INSERT INTO SAILORS VALUES(95,'BOB',3,63);

    INSERT INTO BOATS VALUES(101,'INTERLAKE','BLUE');

    INSERT INTO BOATS VALUES(102,'INTERLAKE','RED');

    INSERT INTO BOATS VALUES(103,'CLIPPER','GREEN');
```

```
    INSERT INTO BOATS VALUES(104,'MARINE','RED');

    INSERT INTO RESERVES VALUES(22,101,'10-OCT-2022');

    INSERT INTO RESERVES VALUES(22,102,'10-OCT-2022');

    INSERT INTO RESERVES VALUES(22,103,'08-OCT-2022');

    INSERT INTO RESERVES VALUES(22,104,'07-OCT-2022');

    INSERT INTO RESERVES VALUES(31,102,'10-NOV-2022');

    INSERT INTO RESERVES VALUES(31,103,'06-NOV-2022');

    INSERT INTO RESERVES VALUES(31,104,'12-NOV-2022');

    INSERT INTO RESERVES VALUES(64,101,'05-SEP-2022');

    INSERT INTO RESERVES VALUES(64,102,'08-SEP-2022');

    INSERT INTO RESERVES VALUES(74,103,'08-SEP-2022');

    COMMIT;
'''
```

In [ ]:
```
cur.executescript(insertValues)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

Questions:

1. Write down the constraints to Update the Views in SQL.(3m)

> - Only a view based on a single parent table can be updated.
> - The view should share the table's primary key.
> - The view should not have any subquery/filter or group by operation.

2. Write a View to Select the sids of sailors who have reserved a red boat. (2m)

In [ ]:
```
cur.execute(
    '''
    CREATE VIEW REDBOAT AS
    SELECT S.SID, S.SNAME
    FROM SAILORS S INNER JOIN ( BOATS B INNER JOIN RESERVES R ON B.BID = R.BID ) ON S.SID = R.SID
    WHERE B.COLOR = 'RED';
    '''
    )
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
cur.execute('SELECT * FROM REDBOAT;').fetchall()
```

Out[ ]:
```
[(22, 'DUSTIN'),
 (22, 'DUSTIN'),
 (31, 'LUBBER'),
 (31, 'LUBBER'),
 (64, 'HORATIO')]
```

3. Write a View to Select the names of sailors who have reserved atleast one boat. (2m)

In [ ]:
```
cur.execute(
    '''
    CREATE VIEW SAILORRESERVES AS
    SELECT S.SNAME
    FROM SAILORS S INNER JOIN ( BOATS B INNER JOIN RESERVES R ON B.BID = R.BID ) ON S.SID = R.SID
    GROUP BY S.SNAME;
    '''
)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
cur.execute('SELECT * FROM SAILORRESERVES;').fetchall()
```

Out[ ]:
```
[('DUSTIN',), ('HORATIO',), ('LUBBER',)]
```

4. Write a View to Select the names of sailors who have not reserved a red boat. (2m)

In [ ]:
```
cur.execute(
    '''
    CREATE TABLE NOTRESERVED AS
    SELECT S.SID, S.SNAME
    FROM SAILORS S LEFT JOIN ( BOATS B INNER JOIN RESERVES R ON B.BID = R.BID ) ON S.SID = R.SID
    WHERE R.SID IS NULL;
    '''
)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
cur.execute('SELECT * FROM NOTRESERVED;').fetchall()
```

Out[ ]:
```
[(29, 'BRUTUS'),
 (32, 'ANDY'),
 (58, 'RUSTY'),
 (71, 'ZORBA'),
 (85, 'ART'),
 (95, 'BOB')]
```

5. Write a View to Select the name and age of oldest sailor(2m)

In [ ]:
```
cur.execute(
    '''
    CREATE VIEW SAILOROLD AS
    SELECT S.SID, S.SNAME, MAX(S.AGE) AS MAX_AGE
    FROM SAILORS S
    '''
)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
cur.execute('SELECT * FROM SAILOROLD;').fetchall()
```

Out[ ]:
```
[(95, 'BOB', '63')]
```

6. Update the ratings by 3 points for sailors above age 35(2m)

SQL:

```
CREATE VIEW SAILORVIEW AS
SELECT * FROM SAILORS;

UPDATE SAILORVIEW SET RATING = RATING + 3 WHERE AGE > 35;
```

In [ ]:
```
cur.execute('SELECT * FROM SAILORS;').fetchall()
```

Out[ ]:
```
[(22, 'DUSTIN', 7, '45'),
 (29, 'BRUTUS', 1, '3'),
 (31, 'LUBBER', 8, '5'),
 (32, 'ANDY', 8, '25'),
 (58, 'RUSTY', 10, '35'),
 (64, 'HORATIO', 7, '35'),
 (71, 'ZORBA', 10, '16'),
 (74, 'HORATIO', 9, '40'),
 (85, 'ART', 3, '25'),
 (95, 'BOB', 3, '63')]
```

In [ ]:
```
cur.execute(
    '''
    UPDATE SAILORS SET RATING = RATING + 3 WHERE AGE > 35;
    '''
)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
cur.execute('SELECT * FROM SAILORS;').fetchall()
```

Out[ ]:
```
[(22, 'DUSTIN', 10, '45'),
 (29, 'BRUTUS', 1, '3'),
 (31, 'LUBBER', 11, '5'),
 (32, 'ANDY', 8, '25'),
 (58, 'RUSTY', 10, '35'),
 (64, 'HORATIO', 7, '35'),
 (71, 'ZORBA', 10, '16'),
 (74, 'HORATIO', 12, '40'),
 (85, 'ART', 3, '25'),
 (95, 'BOB', 6, '63')]
```

7. Update the date of sid 101 as todays date, use date function.(2m)

SQL:

```
CREATE VIEW RESERVEVIEW AS
SELECT * FROM RESERVES;

UPDATE RESERVEVIEW SET DAY = DATE('NOW') WHERE BID = 101;
```

In [ ]:
```
cur.execute('SELECT * FROM RESERVES;').fetchall()
```

Out[ ]:
```
[(22, 101, '10-OCT-2022'),
 (22, 102, '10-OCT-2022'),
 (22, 103, '08-OCT-2022'),
 (22, 104, '07-OCT-2022'),
 (31, 102, '10-NOV-2022'),
 (31, 103, '06-NOV-2022'),
 (31, 104, '12-NOV-2022'),
 (64, 101, '05-SEP-2022'),
 (64, 102, '08-SEP-2022'),
 (74, 103, '08-SEP-2022')]
```

In [ ]:
```
cur.execute(
    '''
    UPDATE RESERVES SET DAY = DATE('NOW') WHERE BID = 101;
    '''
)
```

Out[ ]:
```
<sqlite3.Cursor at 0x18e30558a40>
```

In [ ]:
```
cur.execute('SELECT * FROM RESERVES;').fetchall()
```

Out[ ]:
```
[(22, 101, '2022-10-11'),
 (22, 102, '10-OCT-2022'),
 (22, 103, '08-OCT-2022'),
 (22, 104, '07-OCT-2022'),
 (31, 102, '10-NOV-2022'),
 (31, 103, '06-NOV-2022'),
 (31, 104, '12-NOV-2022'),
 (64, 101, '2022-10-11'),
 (64, 102, '08-SEP-2022'),
 (74, 103, '08-SEP-2022')]
```

8. Create the following table and insert 5 in each table and to prove the execution of sequence in SQL

Tables:

```
create table author(

id NUMBER(6),

name VARCHAR2(20) NOT NULL,

phone_number VARCHAR2(20)

)
```

Question:

A. Note: author_id should be automatically generated through sequence (2m)
B. Keep the minimum value as 1 and Max value as 5 and repeat the cycle. (2m)
C. Print the next value of the pointer focusing the sequence(1m)

SQL:

```
CREATE SEQUENCE AUTHORSEQ
START WITH 1
INCREMENT BY 1
MAXVALUE 5
CYCLE
CACHE 3;

CREATE TABLE AUTHOR (
    AID INTEGER,
    NAME VARCHAR2(20) NOT NULL,
    PHONE_NUMBER VARCHAR2(20),
    PRIMARY KEY (AID)
);

INSERT INTO AUTHOR VALUES(AUTHORSEQ.NEXTVAL, 'NAME1', 'PHONE_NUMBER1');
INSERT INTO AUTHOR VALUES(AUTHORSEQ.NEXTVAL, 'NAME2', 'PHONE_NUMBER2');
INSERT INTO AUTHOR VALUES(AUTHORSEQ.NEXTVAL, 'NAME3', 'PHONE_NUMBER3');
```

```
INSERT INTO AUTHOR VALUES(AUTHORSEQ.NEXTVAL, 'NAME4', 'PHONE_NUMBER4');
INSERT INTO AUTHOR VALUES(AUTHORSEQ.NEXTVAL, 'NAME5', 'PHONE_NUMBER5');

SELECT AUTHORSEQ.NEXTVAL FROM DUAL;
```

In [ ]:
```
conn.close()
```