CS 6083: Principles of Database Systems
Homework 2
Professor Phyllis Frankl

Name: Junior Francisco Garcia
NYU ID: jfg388
Date: 2024-02-29

1. Question 1

   (a) SQL queries

      i. Find all Events that are held by the "French Literature Club" and allow more than 10 people to attend.

```
select  *
from (Event natural join HoldsEvent) natural join Club
where cname = 'French Literature Club' and maxpeople > 10;
```

      ii. List students' names along with all the clubs that held events the student has taken in 2023 and spent less than 10$ to register.

```
select distinct sname, cname
from ((((Student natural join Register)
natural join Event)
natural join HoldsEvent)
natural join Club)
where Event.edate between '2023-01-01' and '2023-12-31'
and Register.price < 10;
```

      iii. List students' names along with all the events that they attended as non-member.

```
select distinct sname, ename
from Student natural join Register
natural join Event
where Register.price = Event.nonmemprice;
```

      iv. List students along with all their membership clubs in 2022 Fall whose membership cost is lower than 20$.

```
select sname,cname
from Student natural join Membership natural join Club
where memberfee < 20 and semester = 'Fall' and year = 2022;
```

      v. Output the name of any club that has never held an event with any rating below 4.

```
select distinct cname
from Club natural join HoldsEvent natural join Event natural
    ↪ join Register
where rating >= 4;
```

      vi. Output the name of any student who has attended an event jointly organized by the Chinese and Japanese Student Associations while not being a member of either one.

```
select distinct S.sname
from Student S
join Register R on S.sid = R.sid
where exists (
    -- Check for event organized by the Chinese Student
       ↪ Association
    select 1
    from HoldsEvent HE
    join Club C on HE.cid = C.cid
    where HE.eid = R.eid -- makess ure it matches with the
       ↪ outside query
    and C.cname = 'Chinese Student Association'
)
and exists (
    -- Check for the same event also organized by the Japanese
       ↪  Student Association
    select 1
    from HoldsEvent HE
    join Club C on HE.cid = C.cid
    where HE.eid = R.eid -- makess ure it matches with the
       ↪ outside query
    and C.cname = 'Japanese Student Association'
)
and not exists (
    -- Ensure the student is not a member of either club
    select 1
    from Membership M
    join Club C on M.cid = C.cid
    where M.sid = S.sid -- makess ure it matches with the
       ↪ outside query
    and C.cname in ('Chinese Student Association', 'Japanese
       ↪ Student Association')
);
```

(b) SQL Tables

```
CREATE TABLE Student (
    sid INT PRIMARY KEY,
    sname VARCHAR(255),
    semail VARCHAR(255),
    sphone VARCHAR(20)
);

CREATE TABLE Club (
    cid INT PRIMARY KEY,
    cname VARCHAR(255),
    cdescription TEXT
);

CREATE TABLE Event (
    eid INT PRIMARY KEY,
    ename VARCHAR(255),
    edescription TEXT,
    edate DATE,
    memprice DECIMAL(10, 2),
    nonmemprice DECIMAL(10, 2),
    maxpeople INT
);

CREATE TABLE Membership (
    sid INT,
    cid INT,
    semester VARCHAR(20),
    year INT,
    memberfee DECIMAL(10, 2),
    PRIMARY KEY (sid, cid, semester, year),
    FOREIGN KEY (sid) REFERENCES Student(sid),
    FOREIGN KEY (cid) REFERENCES Club(cid)
);

CREATE TABLE HoldsEvent (
    eid INT,
    cid INT,
    PRIMARY KEY (eid, cid),
    FOREIGN KEY (eid) REFERENCES Event(eid),
    FOREIGN KEY (cid) REFERENCES Club(cid)
);

CREATE TABLE Register (
    eid INT,
    sid INT,
    price DECIMAL(10, 2),
    rating INT CHECK (rating >= 1 AND rating <= 5),
    PRIMARY KEY (eid, sid),
    FOREIGN KEY (eid) REFERENCES Event(eid),
    FOREIGN KEY (sid) REFERENCES Student(sid)
);
```

2. Relational Algebra

   (a) Find all Events that are held by the “French Literature Club” and allow more than 10 people to attend.

$$\pi_{\text{cid}}\left(\sigma_{\text{cname}='FrenchLiteratureClub'\wedge\text{maxpeople}>10}\right.$$

$$\left.(Event \bowtie HoldsEvent \bowtie Club)\right)$$

(b) List students' names along with all the clubs that held events the student has taken in 2023 and spent less than 10$ to register.

$$\pi_{sname,cname}\left(\sigma_{edate\geq'2023-01-01'\wedge edate\leq'2023-12-31'\wedge price<10}\right.$$

$$\left.((Student \bowtie Register) \bowtie Event \bowtie HoldsEvent \bowtie Club)\right)$$

(c) List students' names along with all the events that they attended as non-member.

$$\pi_{sname,ename}\left(\sigma_{price=nonmemprice}\left(Student \bowtie Register \bowtie Event\right)\right)$$

(d) List students along with all their membership clubs in 2022 Fall whose membership cost is lower than 20$.

$$\pi_{sname,cname}\left(\sigma_{memberfee<20\wedge semester='Fall'\wedge year=2022}\left(Student \bowtie Membership \bowtie Club\right)\right)$$

3. Domain Relational Calculus

(a) Find all Events that are held by the "French Literature Club" and allow more than 10 people to attend.

$$\begin{aligned}
\{\langle eid\rangle \mid \exists cid, cname, maxpeople\, (&\langle eid, cid\rangle \in \text{HoldsEvent} \land \\
&\langle cid, cname, cdescription\rangle \in \text{Club} \land \\
&\langle eid, ename, edescription, edate, memprice, nonmemprice, maxpeople\rangle \in \text{Event} \land \\
&cname = \text{’French Literature Club’} \land maxpeople > 10)\}
\end{aligned}$$

(b) List students' names along with all the clubs that held events the student has taken in 2023 and spent less than 10$ to register.

$$\begin{aligned}
\{\langle sname, cname\rangle \mid \exists sid, eid, cid, price, edate\, (&\langle sid, sname\rangle \in \text{Student} \land \\
&\langle sid, eid, price\rangle \in \text{Register} \land price < 10 \land \\
&\langle eid, edate\rangle \in \text{Event} \land \text{’2023-01-01’} \leq edate \leq \text{’2023-12-31’} \land \\
&\langle eid, cid\rangle \in \text{HoldsEvent} \land \\
&\langle cid, cname\rangle \in \text{Club})\}
\end{aligned}$$

(c) List students' names along with all the events that they attended as non-member.

$$\begin{aligned}
\{\langle sname, ename\rangle \mid \exists & sid, eid, ename, price, nonmemprice \\
&(\langle sid, sname\rangle \in \text{Student} \land \\
&\langle sid, eid, price\rangle \in \text{Register} \land \\
&\langle eid, ename, nonmemprice\rangle \in \text{Event} \land \\
&price = nonmemprice)\}
\end{aligned}$$

(d) List students along with all their membership clubs in 2022 Fall whose membership cost is lower than 20$.

$$\begin{aligned}
\{\langle sname, cname\rangle \mid \exists & sid, cid, memberfee, semester, year \\
&(\langle sid, sname\rangle \in \text{Student} \land \\
&\langle sid, cid, memberfee, semester, year\rangle \in \text{Membership} \land \\
&\langle cid, cname\rangle \in \text{Club} \land \\
&memberfee < 20 \land semester = \text{’Fall’} \land year = 2022)\}
\end{aligned}$$

4. Theme Park Database Queries

Table Queries (for reference)

```sql
CREATE TABLE Tourist (
tid VARCHAR(7) PRIMARY KEY,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
DOB date,
email VARCHAR(100) NOT NULL
);
CREATE TABLE ThemePark (
ParkTitle VARCHAR(50) NOT NULL,
Addr VARCHAR(50) NOT NULL,
Rating FLOAT NOT NULL,
PRIMARY KEY (ParkTitle, Addr)
);
CREATE TABLE Ride (
RideTitle VARCHAR(50) PRIMARY KEY,
Avg_wt TIME NOT NULL,
MinHeightRestriction INT
);
CREATE TABLE Visit (
tid VARCHAR(7),
ParkTitle VARCHAR(50),
Addr VARCHAR(50),
Stars INT NOT NULL,
FOREIGN KEY (tid) REFERENCES Tourist (tid),
FOREIGN KEY (ParkTitle, Addr) REFERENCES ThemePark (ParkTitle, Addr)
);
CREATE TABLE Favorite (
tid VARCHAR(7),
RideTitle VARCHAR(50),
FOREIGN KEY (tid) REFERENCES Tourist (tid),
FOREIGN KEY (RideTitle) REFERENCES Ride (RideTitle)
);
CREATE TABLE Locate (
RideTitle VARCHAR(50),
ParkTitle VARCHAR(50),
Addr VARCHAR(50),
IP VARCHAR(50) NOT NULL,
FOREIGN KEY (RideTitle) REFERENCES Ride (RideTitle),
FOREIGN KEY (ParkTitle, Addr) REFERENCES ThemePark (ParkTitle, Addr)
);
```

(a) List all Rides in the Theme Park with the title MagicKingdomPark located at

Bay Lake.

```
SELECT RideTitle
FROM Ride
NATURAL JOIN Locate
WHERE ParkTitle = 'MagicKingdomPark' AND Addr = 'Bay Lake';
```

(b) List all Tourists who were born in 2001.

```
SELECT first_name, last_name
FROM Tourist
WHERE YEAR(DOB) = 2001;
```

(c) List the names and emails of Tourists who have visited Disney’sHollywoodStudios (the string here is exactly the value in the test database) located at Bay Lake. **Note that there is an error with handling the string literal inside this query on LaTeX verbatim. Assume the string literal is ’Disney”sHollywoodStudios’**.

```
SELECT t.first_name, t.last_name, t.Email
FROM Tourist t
NATURAL JOIN Visit v
WHERE v.ParkTitle = 'Disney''sHollywoodStudios' AND v.Addr = 'Bay
   ↪  Lake';
```

(d) List the names of Tourists whose favorite rides have a waiting time longer than 30 minutes.

```
SELECT T.first_name, T.last_name
FROM Tourist T
NATURAL JOIN Favorite F
NATURAL JOIN Ride R
WHERE R.Avg_wt > 30;
```

(e) List the names of Tourists who give a non-5 star to the Theme Park where their favorite rides are located.

```
SELECT T.first_name, T.last_name
FROM Tourist T
NATURAL JOIN Favorite F
NATURAL JOIN ThemePark TP
NATURAL JOIN Visit V
WHERE V.Stars != 5;
```