

Department of Computer Science

CS220: Database Systems

Class: BESE-12 B

Imaan Ibrar

373982

Lab12:

Open Ended Lab

Date: October 24, 2022

Time: 10:00-01:00 & 02:00 – 05:00

Instructor: Dr. Bilal Ali Rizvi

Lab Engineer: Hada iq Ahmad

Open Ended Lab

BESE-12 B

Scenario

With the recent increase in Shahid's salary, he purchased Honda Civic 2022 model and was driving around the main shopping mall located in Isra Apartments at H-12, Islamabad. All of a sudden, another driver Awaiz driving Camry 2015 with his boss Sohail bumped into his car and caused a damaged to cars back lights, bumper and body of Rs. 150,000. Sohail immediately called Hamza who owns the 40% of Camry alongwith Corolla GLi that he bought in 2018 and informed him about the damages brought by the accident.

The junior DB analyst has analyzed the given scenario and design the schema given below. Consider the above mentioned scenario and designed schema, recommend appropriate changes with proper reasons.

The designed relational schema is as follows:

```
Persons(pid:number, name:string, street: string, postcode: string)
Cars(cid:number, model:number, make:string)
OwnedBy(pid:number, cid:number)
AccidentReports(rid:number, damage:number, details:string)
ParticipatedIn(pid:number, rid:number, cid:number)
```

where the underlined attributes represent the primary keys of the associated relation. The table OwnedBy implements a relationship between persons and cars using foreign keys. The table ParticipatedIn implements a relationship between persons, accident reports, and cars, where tuple (p, r, c) indicates that the person p was the driver of the car c associated with the accident report r.

Requirements :

1. Add the attributes in the schema to make it more comprehensive.
Here are some additional attributes that could be added to the schema to make it more comprehensive:
Persons: age (integer), gender (string), phone number (string)
Cars: year (integer), color (string), purchase date (date)
AccidentReports: date (date), time (time), location (string)
ParticipatedIn: role (string, e.g. "driver", "passenger")
2. Convert the schema into executable relations

```
CREATE TABLE Persons (  
    pid INTEGER PRIMARY KEY,  
    name VARCHAR(255),
```

```

street VARCHAR(255),
postcode VARCHAR(255),
age INTEGER,
gender VARCHAR(255),
phone_number VARCHAR(255)
);

CREATE TABLE Cars (
  cid INTEGER PRIMARY KEY,
  model VARCHAR(255),
  make VARCHAR(255),
  year INTEGER,
  color VARCHAR(255),
  purchase_date DATE
);

CREATE TABLE OwnedBy (
  pid INTEGER,
  cid INTEGER,
  PRIMARY KEY (pid, cid),
  FOREIGN KEY (pid) REFERENCES Persons(pid),
  FOREIGN KEY (cid) REFERENCES Cars(cid)
);

CREATE TABLE AccidentReports (
  rid INTEGER PRIMARY KEY,
  damage INTEGER,
  details VARCHAR(255),
  date DATE,
  time TIME,
  location VARCHAR(255)
);

CREATE TABLE ParticipatedIn (
  role varchar(255),
  pid INTEGER,
  rid INTEGER,
  cid INTEGER,
  PRIMARY KEY (pid, rid, cid),
  FOREIGN KEY (pid) REFERENCES Persons(pid),
  FOREIGN KEY (rid) REFERENCES AccidentReports(rid),
  FOREIGN KEY (cid) REFERENCES Cars(cid)
);

```

3. Consider the following scenario, populate the relations accordingly and express the solutions of the following requirements:

- (a) Write an SQL query to return those pid's of persons driving in at least one accident, with no duplicates.

To return those pid's of persons driving in at least one accident, with no duplicates:

```
SELECT DISTINCT pid
FROM ParticipatedIn
WHERE role = 'driver';
```

- (b) Write an SQL query to return all tuples (pid, c), where c is the number of cars owned by person pid (records where c = 0 do not have to be generated).

To return all tuples (pid, c), where c is the number of cars owned by person pid (records where c = 0 do not have to be generated):

```
SELECT pid, COUNT(cid) AS c
FROM OwnedBy
GROUP BY pid;
```

- (c) Write an SQL query to return all tuples (cid, c), where c is the number of persons owning car cid (records where c = 0 do not have to be generated).

To return all tuples (cid, c), where c is the number of persons owning car cid (records where c = 0 do not have to be generated):

```
SELECT cid, COUNT(pid) AS c
FROM OwnedBy
GROUP BY cid;
```

- (d) Write a (nested) SQL query to return all tuples (pid, rid) where pid was driving in the accident reported in rid, but the car driven by pid is not owned by pid.

To return all tuples (pid, rid) where pid was driving in the accident reported in rid, but the car driven by pid is not owned by pid:

```
SELECT pid, rid
FROM ParticipatedIn
WHERE role = 'driver' AND cid NOT IN (SELECT cid FROM OwnedBy WHERE pid = Particip
```

- (e) Write an SQL query to return all tuples (rid, c), where c is the number of drivers involved in the accident reported in by rid (records where c = 0 do not have to be generated).

To return all tuples (rid, c), where c is the number of drivers involved in the accident reported in by rid (records where c = 0 do not have to be generated):

```
SELECT rid, COUNT(pid) AS c
FROM ParticipatedIn
WHERE role = 'driver'
GROUP BY rid;
```