School of Information Technologies

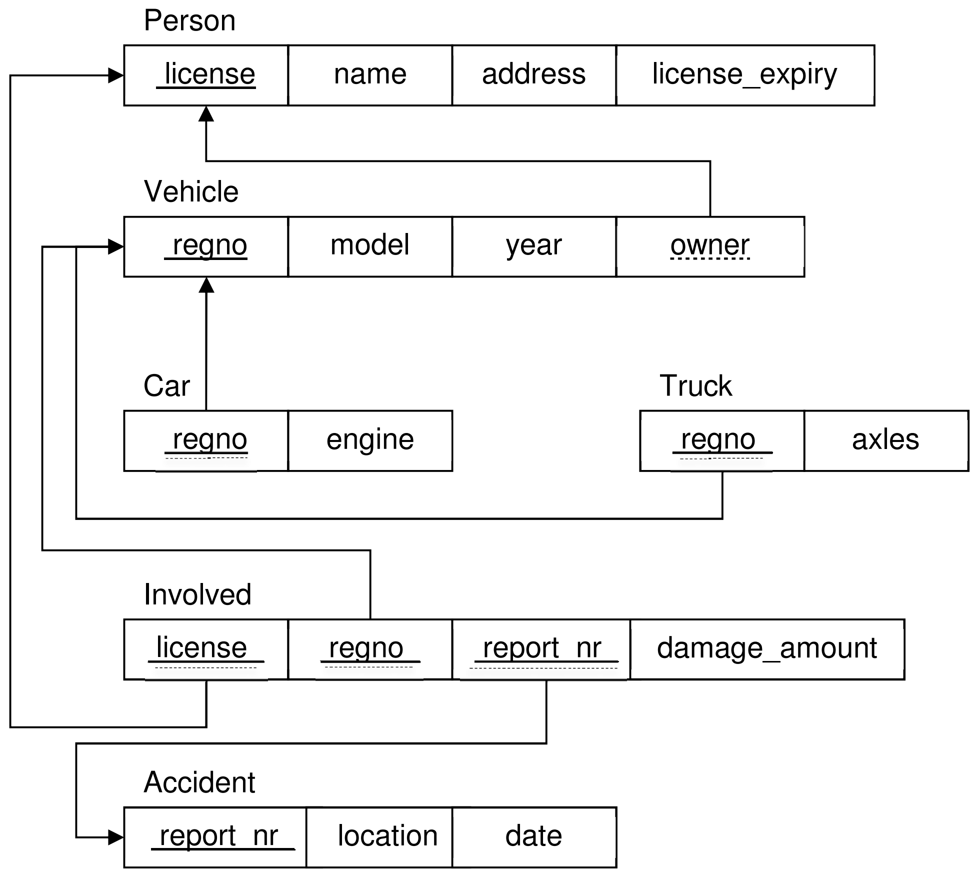
COMP9120 Relational Database Systems

Example solutions: Practice Questions Part 1

Question 1: ER mapping and Data Integrity [30 points]

a) Translate the ER diagram given above into relations using a graphical notation. Clearly indicate all primary and foreign keys.

Solution



b)

b) Give the SQL CREATE TABLE statements needed to create a database for your schema. Include reasonable data types, and other required domain constraints (eg: NOT NULL), primary key and referential integrity (foreign key) constraints.

Solution

```
CREATE TABLE Person (
 license INTEGER,
 name VARCHAR(20),
 address VARCHAR(20),
 license_expiry DATE,
 CONSTRAINT Person_PK PRIMARY KEY (license)
);

CREATE TABLE Vehicle (
 regno VARCHAR(10),
 model VARCHAR(20),
 year SMALLINT,
 owner INTEGER NOT NULL,
 CONSTRAINT Vehicle_PK PRIMARY KEY (regno),
 CONSTRAINT Vehicle_FK FOREIGN KEY(owner)
REFERENCES Person
);

CREATE TABLE Car (
 regno VARCHAR(10),
 engine VARCHAR(20),
 CONSTRAINT Car_PK PRIMARY KEY (regno),
 CONSTRAINT Car_FK FOREIGN KEY (regno) REFERENCES
Vehicle
);

CREATE TABLE Truck (
 regno VARCHAR(10),
 axles INTEGER,
 CONSTRAINT Truck_PK PRIMARY KEY (regno),
 CONSTRAINT Truck_FK FOREIGN KEY (regno) REFERENCES
Vehicle
);

CREATE TABLE Accident (
 report_nr INTEGER,
 location VARCHAR(20),
 acc_date DATE,
 CONSTRAINT Accident_PK PRIMARY KEY (report_nr)
);
```

```
CREATE TABLE Involved (
 driver INTEGER,
 regno VARCHAR(10),
 report_nr INTEGER,
 damage_amount DECIMAL(10,2),
 CONSTRAINT Involved_FK1 FOREIGN KEY (driver)
REFERENCES Person,
 CONSTRAINT Involved_FK2 FOREIGN KEY (regno)
REFERENCES Vehicle,
 CONSTRAINT Involved_FK3 FOREIGN KEY(report_nr)
REFERENCES Accident,
 CONSTRAINT Involved_PK PRIMARY KEY(driver, regno,
report_nr)
);
```

c) Write the appropriate alter table SQL code to add a table check constraint that evaluates whether a truck has 2 or more axles.

Solution

```
ALTER TABLE Truck ADD CONSTRAINT AxlesChk CHECK (
Axles >= 2 );
```

d) Write an assertion that checks if all driver licenses are valid until at least January 1, 2015.

Solution

```
CREATE ASSERTION AssertLicenses
 CHECK (
     NOT EXISTS (
         SELECT license_expiry
         FROM Person
         WHERE license_expiry < '1/JAN/2015'
     )
 );
```

e) Briefly describe one possible advantage and one possible disadvantage of SQL assertions relative to a table check constraints.

Solution

Possible advantage: Unlike table check constraints, assertions check if a condition is violated whenever any of the tables mentioned in the condition are modified.

Possible disadvantage: Not adopted by major DBMS'; can incur significant performance overheads.

Question 2: Normal Forms [26 points]

a) Give an interpretation in plain English for each of the functional dependencies.

Solution

PatID → Age, City, Zip

Each patient can have at most one age, city and zip code

Zip → City

Each Zip can describe at most one city.

ProvNo → ProvSpeciality

Each doctor can have at most one speciality.

VisitID → PatID, VisitDate, Age, City, Zip

Each visit is for at most one patient, it can take place on at most one day, in at most one city, in at most one zip code, and for at most one age.

VisitID, ProNo → Diagnosis

Several doctors can attend the same visit, but each doctor can give at most one diagnosis for each visit.

Note: Statements like “Zip is functional dependent on City” get no marks. Some interpretation of the meaning is required.

b) Identify an insert anomaly, delete anomaly, and update anomaly.

Some examples of anomalies

Insert anomaly: We cannot insert a new ProvNo to describe a doctor’s speciality unless we also include a VisitID (i.e., we can’t record details about a doctor unless they’ve made a visit).

Delete anomaly: If we delete the only record of a visit made by a doctor, we lose the information about their speciality.

Update anomaly: If a patient’s address changes, we must update this address for every row in which that patient appears.

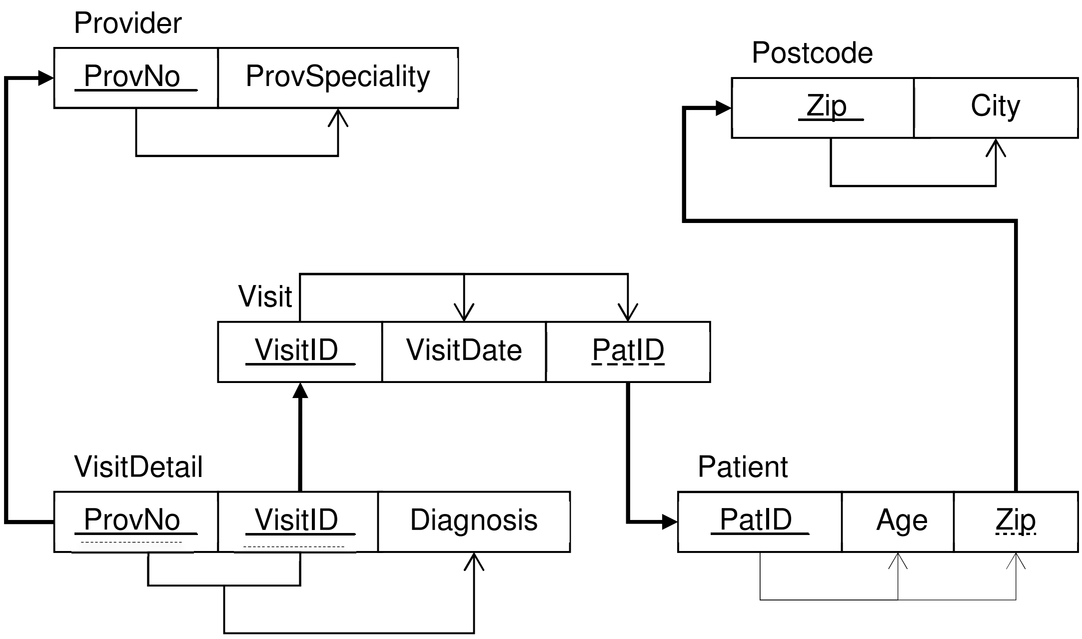
c) Identify all candidate keys.

Solution

Candidate key is (VisitID, ProvNo)

d) Construct a set of BCNF relations decomposed from the original relation. Use the graphical notation described at the beginning of this paper to clearly indicate all primary and foreign keys.

Solution: Note: Here I show links for foreign keys in thick arrows, but I also show dependencies in thin arrows for your convenience. For each decomposition you should consider the lossless join and BCNF rules. The FD decomposition rule can help here.



Question 3: SQL and Relational Algebra [24 points]

a) Write an SQL query to find the count and the average screensize of televisions costing more than $2000.

Example Solution

```
SELECT COUNT(*), AVG(screensize)
FROM Television t INNER JOIN Device p ON (t.dev=p.dev)
WHERE p.price > 2000;
```

b) Assume the following example database instance for the above schema:

```
Device('Alex', 'T20FV300', 300, 1)
Device('Alex', 'T34HS510', 2000, 2)
Device('Bundig', 'B45', 3899, 1)
Device('ABC', 'ABC tba', 2500, 1)

Television('T20FV300', 'trinitron', 20, FALSE)
Television('T34HS510', 'tube', 34, TRUE)
Television('B45', 'Plasma', 42, TRUE)
Television('ABC tba', 'Plasma', null, null)
```

What will be the result of your SQL query from the previous subquestion (a) on this example database?

Solution

Expected result is (2, 42)

c) Write an SQL query for “what are the cheapest (lowest price) DVDPlayers made by ’TEC’ and what do they cost?”

Example Solution

```
SELECT p.dev, p.price
FROM Device p INNER JOIN DVDPlayer d ON(p.dev=d.dev)
WHERE maker = ‘TEC’
AND p.price <= ALL (
SELECT price
 FROM Device ip INNER JOIN DVDPlayer id ON (ip.dev=id.dev)
 WHERE maker = ‘TEC’ );
```

d) Consider the following SQL query:

```
SELECT p.dev, c.max_consumption
FROM Device p, Projector r, PowerClass c
WHERE p.maker = ’Alex’ AND
p.dev = r.dev AND
p.powerClass = c.powerClass
```

Write down an equivalent expression in relational algebra.

Example Solution

$\pi_{dev,\text{max\_}consumption}\sigma_{maker=\text{"}Alex\text{"}}(\text{PowerClass} \bowtie (Device \bowtie Projector))$

e) Write an SQL for the following: For each manufacturer (maker), find the price range (lowest and highest price) and the average price of televisions and projectors made by them. The result shall show the makers, the corresponding price ranges (i.e., minimum and maximum) and average of the prices. Results should only be shown for manufacturers where at least 3 devices are considered in calculating the results.

Example Solution

```
SELECT maker, MIN(price), MAX(price), AVG(price)
FROM Device NATURAL JOIN (
    SELECT dev
    FROM Television
    UNION
    SELECT dev
    FROM Projector)
GROUP BY maker
HAVING COUNT(*) >= 3;
```