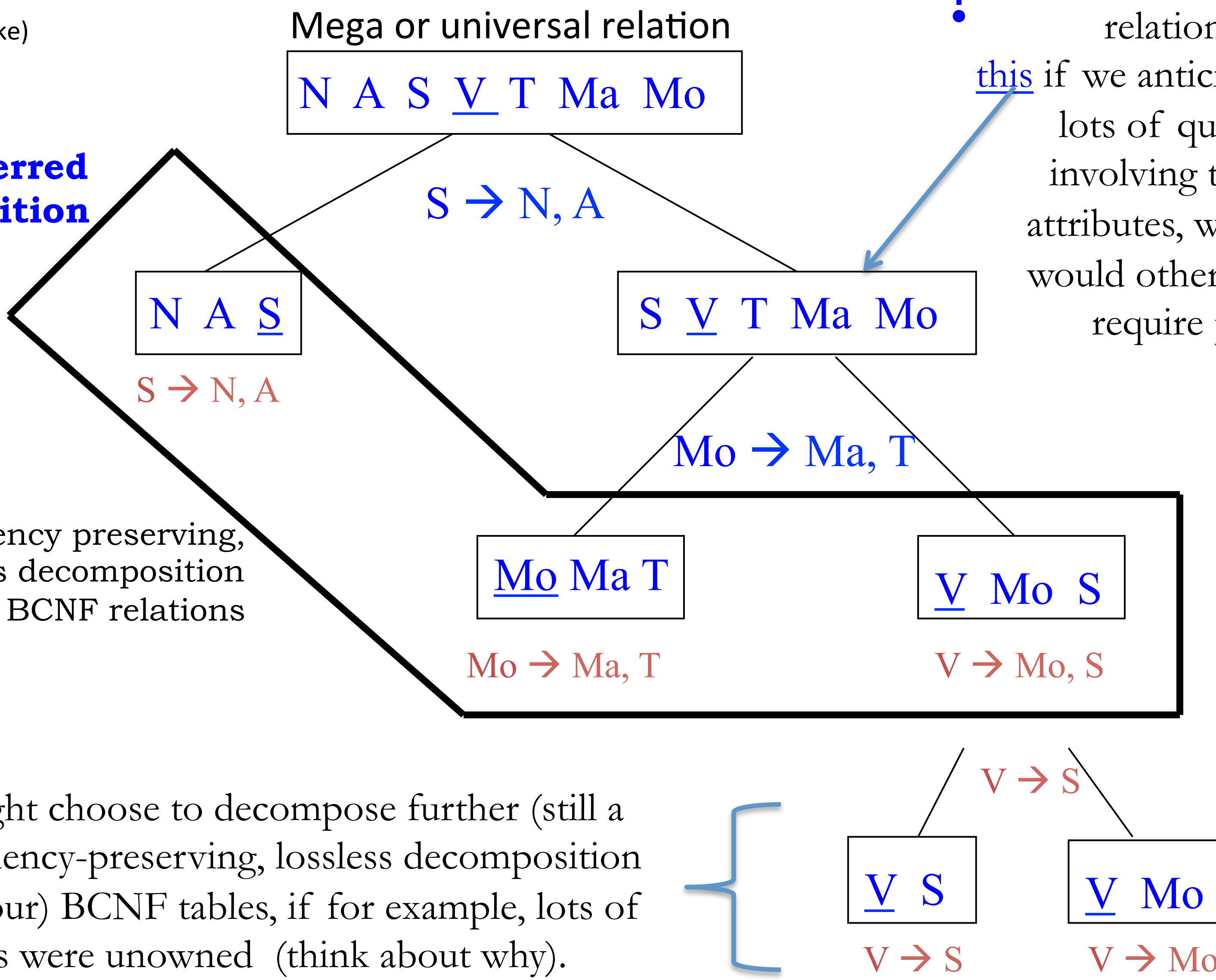
N: Name
A: Address
S: SSN
V: VRN
T: Type
Ma: Manufacturer (or Make)
Mo: Model

Review

(Probably) Preferred decomposition



? We might choose a denormalized relation like this if we anticipate lots of queries involving these attributes, which would otherwise require joins

A dependency preserving, lossless decomposition into BCNF relations

? We might choose to decompose further (still a dependency-preserving, lossless decomposition into (four) BCNF tables, if for example, lots of vehicles were unowned (think about why).

N	A	S	V	T	Ma	Mo
Fred	Nashville	123	987	Truck	Ford	Ranger
Sri	NewYork	234	876	Car	Toyota	Camry
Gabriel	Nashville	345	765	MotorCy	Harley	Hog
Fred	Nashville	123	654	Car	VW	Bug

N	A	S
Fred	Nashville	123
Sri	NewYork	234
Gabriel	Nashville	345

T	Ma	Mo
Truck	Ford	Ranger
Car	Toyota	Camry
MotorCy	Harley	Hog
Car	VW	Bug

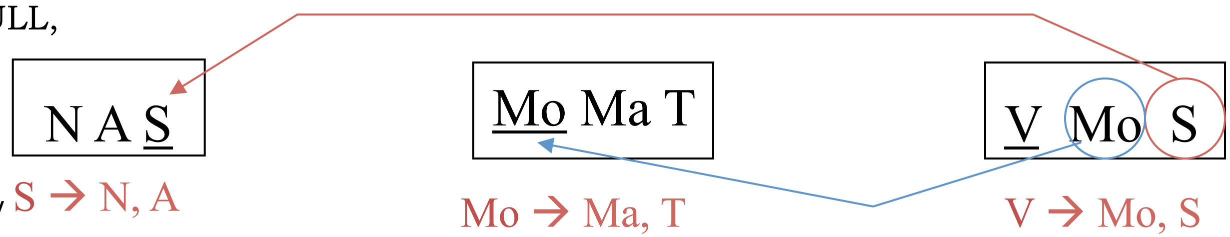
S	V	Mo
123	987	Ranger
234	876	Camry
345	765	Hog
123	654	Bug

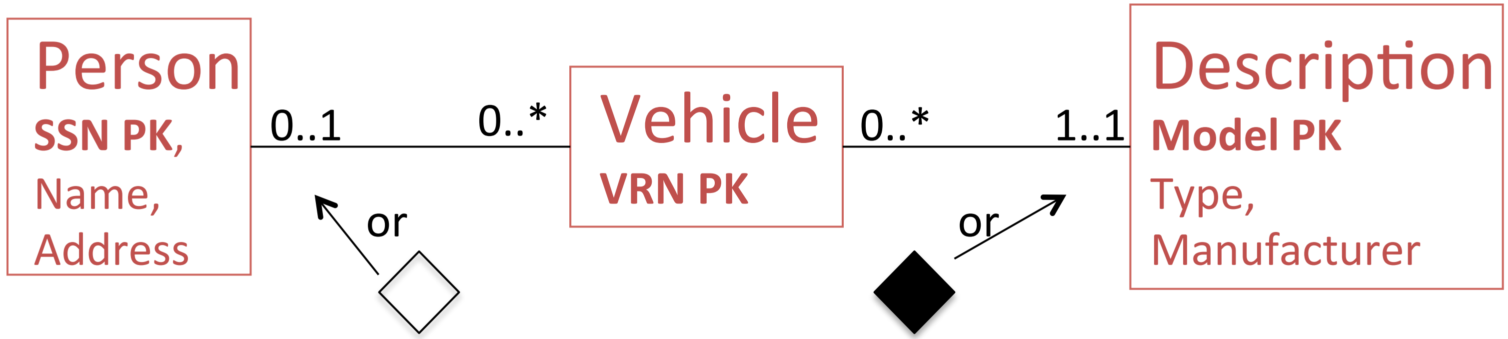
An example DB instance using the preferred decomposition

```
CREATE TABLE Person (
        Name VARCHAR(60) NOT NULL,
        Address VARCHAR(120) NOT NULL,
        SSN INTEGER PRIMARY KEY
);

CREATE TABLE Description (
        Model CHAR(20) PRIMARY KEY,
        Manufacturer CHAR(20) NOT NULL,
        Type CHAR(10)
);

CREATE TABLE Vehicle (
        SSN INTEGER,  /* NOT NULL? */
        VRN INTEGER PRIMARY KEY,
        Model CHAR(10) NOT NULL,
        FOREIGN KEY (SSN) REFERENCES Person ON DELETE NO ACTION ON UPDATE CASCADE
        FOREIGN KEY (Model) REFERENCES  Description ON DELETE NO ACTION ON UPDATE CASCADE
);
```






```
CREATE TABLE Person (
      Name VARCHAR(60) NOT NULL,
      Address VARCHAR(120) NOT NULL,
      SSN INTEGER PRIMARY KEY);

CREATE TABLE Description (
      Model CHAR(20) PRIMARY KEY,
      Manufacturer CHAR(20) NOT NULL,
      Type CHAR(10));

CREATE TABLE Vehicle (
      SSN INTEGER,  /* NOT NULL? */
      VRN INTEGER PRIMARY KEY,
      Model CHAR(10) NOT NULL,
      FOREIGN KEY (SSN) REFERENCES Person ON DELETE NO ACTION ON UPDATE CASCADE
      FOREIGN KEY (Model) REFERENCES  Description ON DELETE NO ACTION ON UPDATE CASCADE);
```

A functional dependency can correspond to either a 0..1 constraint, as in VRN → SSN (above, left) or a 1..1 constraint, as in VRN → Model (above, right). In either case, VRN determines the right hand side values (which can be NULL in the case of 0..1)

Remember, if you were to design a sizable database (sizable in terms of number of relations, you would probably start with a UML diagram (design in the large), but might descide on functional dependencies to refine the database at a smaller scale.

Assignment A-w10 questions – post a single PDF to Brightspace

1. The database design of the previous page only allows storage of one recorded owner (perhaps the current owner) over the DB-lifetime of a vehicle. A DMV would probably want to store records of all past owners of a vehicle, probably with the dates of ownership. Give set of functional dependencies for the following universal relation, where StartDate and EndDate indicate the interval that a particular person owns a particular vehicle. Two people cannot be recorded as owning the same vehicle during overlapping intervals.

Name Address SSN StartDate EndDate VRN Type Make Model

2. Suppose you have a relation P (A, B, C, D, E, F), with functional dependencies (FDs)

$A \rightarrow B$, $BCD \rightarrow E$, $E \rightarrow F$

Suppose there are at most 2 different possible values for each of attributes A, C, and D. What is the maximum number of different values for each other attribute?

Maximum number of values for B:

Maximum number of values for E:

Maximum number of values for F:

3. Suppose that you have a relation Q (A , B, C, D, E) with only one FD A,B → C,D,E . Decompose Q into a set of relations, EACH of which is in BCNF, or state that Q is already in BCNF (and in either case, explain your answer, and in doing so, identify the key for each relation).

4. Suppose that you have a relation R (A, B, C, D, E, F) with FDs A,B → C,D and D → E,F. Decompose R into a set of relations, EACH of which is in BCNF, or state that R is already in BCNF (and in either case, explain your answer, and in doing so, identify the key for each relation).

5. For the Book table (from the Book-seller database), give (a) all the FDs that you believe are enforced by the table definition, and (b) any FDs that you think should be enforced, but aren't currently. (see Book CREATE TABLE statement on next page, for convenience)

(a)

(b)

```
CREATE TABLE Book (
    Isbn                    INTEGER,
    Title                   CHAR[120] NOT NULL,
    Synopsis                CHAR[500],
    ListPrice               CURRENCY NOT NULL,
    AmazonPrice             CURRENCY NOT NULL,
    SavingsInPrice          CURRENCY NOT NULL,
    AveShipLag              INTEGER,
    AveCustRating           REAL,
    SalesRank               INTEGER,
    CoverArt                FILE,
    Format                  CHAR[4] NOT NULL,
    CopiesInStock           INTEGER,
    PublisherName           CHAR[120] NOT NULL,
    PublicationDate         DATE NOT NULL,
    PublisherComment  CHAR[500],
    PublicationCommentDate DATE,
    PRIMARY KEY (Isbn)
    FOREIGN KEY (PublisherName) REFERENCES Publisher,
        ON DELETE NO ACTION,  ON UPDATE CASCADE,
    CHECK (Format = ‘hard’ OR Format = ‘soft’ OR Format = ‘audi’
                                OR Format = ‘cd’ OR Format = ‘digital’)
         // alternatively, CHECK (Format IN (‘hard’, ‘soft’, ‘audi’, ‘cd’, ‘digital’))
    CHECK (AmazonPrice + SavingsInPrice = ListPrice)   )
```

6. Consider the relation

Kwatts, Dorm, Floor#, Date, Time, Temp, Humidity, Occupancy, Weekday?, SensorID

and FDs that are asserted as true of the relation

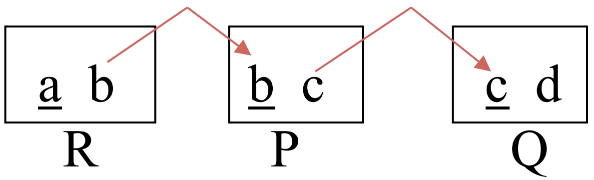
1) Dorm, Floor# → Occupancy
2) Date, Time → Temp, Humidity
3) Date → Weekday?
4) SensorID, Date, Time, Temp → Kwatts
5) Dorm, Floor#, Date, Time, Temp → Kwatts
6) SensorId → Dorm, Floor#
7) Dorm, Floor# → SensorId

a) Give a minimal FD set (remove any FDs that need not be explicitly stated, but that are implied by the remaining FDs). If there is more than one such set, just give one of them. You can cross out any FDs above, or indicate changes in space provided here.

b) Give all keys for the relation (when I use the term "key", I always mean minimal key

6 c) Give a decomposition of the relation into BCNF tables. If the relation is already in BCNF then state so. If there is no dependency-preserving decomposition into BCNF tables then state so.

7. A colleague brings you three table definitions, summarized by these relational schema (R, P, Q), with '<u>a</u>' as a primary key for table R, '<u>b</u>' the primary key for table P, and '<u>c</u>' the primary key for table Q. '<u>b</u>' is a foreign key from R to P, and '<u>c</u>' is a foreign key from P to Q.



In addition to the table definitions, your colleague gives you this assertion, intended to enforce the FD Q.d → R.a.

```
CREATE ASSERTION AsPerD
CHECK (NOT EXISTS (SELECT *
                   FROM (SELECT COUNT (DISTINCT R.a) AS cnt
                         FROM R, P, Q
                         WHERE R.b = P.b AND P.c = Q.c
                         GROUP BY Q.d, R.a) AS Temp
                   WHERE Temp.cnt > 1))
```

(a) Ignoring for the moment that your colleague requires a course in DB design, you recognize that the assertion is incorrect, but that you can correct it by making ONE simple STRIKETHROUGH. Put a line through that part of the assertion definition so that the corrected version properly enforces the FD, d → a.

(b) After you explain your fix, your colleague leaves, and you replace your colleague's three tables and one (corrected) assertion by ONE table definition that enforces all the constraints encoded in the original three tables and one assertion. Give the definition for this one table as a CREATE TABLE statement. List all attributes, and show other constraints, but do not worry about the types of the attributes.