Presentation on Functional Dependencies CS x265

This presentation assumes that you have previously viewed
Watch videos (90 min) and answer questions from DB8
Relational Design Theory
(https://class.stanford.edu/courses/DB/RD/SelfPaced/courseware/ch-relational_design_theory/)
and/or read Chapter 3 of Ullman and Widom, Introduction to Database Management Systems

and watched Doug's first lecture on functional dependencies

**Quiz Q-w10**

**QUESTION 1**

Consider the relation R(A, B, C, D) with functional dependencies (FDs)

A —> B, C

A, C —> D

C, D —> A, B

Which of the following sets of FDs is a minimal set that is informationally equivalent to the set of FDs given above.

The key characteristics are "minimal set" and "informationally equivalent" (or just equivalent), where two sets of FDs are equivalent if

- the closures of their FDs are the same

For example, {A→B, B→ C, C→A} is a minimal set (confirm)

- {A→B, B→ C, C→A} has A→C, B→A, and C→B in its FD closure
  - { A→B, B→ C , C→A, A→C, B→A, C→B, … A,C→B, … A,B,C→A}.

- {A→B, B→ A, B→C, C→B} is also a minimal set, and informationally equivalent to the first

  1 2
  - { A→B, B→ C , C→A, A→C, B→A, C→B, … A,C→B, … A,B,C→A}

- No proper subset of either {A→B, B→ C, C→A} and {A→B, B→ A, B→C, C→B} has the same FD closure.

- Remember that a minimal set is one for which there is no proper subset that is informationally equivalent. Minimality is not judged as a set with the minimal cardinality.

What is a minimal equivalent set for the given FDs of Q-w10?

Start with the set I gave. In this illustration I use the ordering in which I gave the FDs.

A —> B, C (or A→ B and A → C)

A, C —> D

C, D —> A, B (or C,D → A and C,D → B)

Step 1: Are the left hand sides of any FD redundant?

Yes, A, C → D can be simplified to A → D, since A→C (if A, C determines D, but A determines C, then A all by itself determines D).

So, **A,C → D can be replaced by A → D**, and this can be combined with the first FD in the list, yielding A → B, C, D

The left-hand side of C,D → A,B **cannot** be simplified.

**Step 2: Can any FD be eliminated?**

A —> B, C, D (or A→ B; A→ C; A→ D )

C, D —> A, B (or C,D → A and C,D → B)

Go through them in given order – it can be in any order, but different orderings of the FDs can lead to different minimal sets. Choice 2 of Q-w10, which was the only correct choice, results from considering the FDs in **RIGHT-TO-LEFT** order:

A→B; A→C; A→D; C,D → A; C,D→ B, so that C,D → B is the first to be considered.

Pretend C,D → B doesn't exist and take the attribute closure of {C, D}.

C,D → A allows us to add A to the closure: {A, C, D}.
A→ B allows us to add B to the closure: {A,B,C,D}.
So, the attribute closure of {C,D} includes B, even without explicitly giving C,D → B.
**Eliminate C,D → B as redundant.**

No other FDs can be eliminated as redundant. So, a minimal set of the FDs that is informationally equivalent to those given at the start is

**A —> B, C, D**
**C, D —> A** (choice 2 of Q-w10)

Consider the relation R(A, B, C, D) with functional dependencies (FDs)

A —> B, C

A, C —> D

C, D —> A, B

All choices for Q-w10 question 1

1.

**A —> B, C, D**

**C, D —> B**

*This choice follows if C,D→A is redundant in the list above, but if you take the attribute closure of {C,D} without the C,D→A, you will NOT find A in the closure. Not informationally equivalent*

**2. correct**

**A —> B, C, D**

**C, D —> A**

All choices for Q-w10 question 1

3.

**A —> B, C**

**A, C —> D**

**C, D —> A, B**

*not minimal, as illustrated above*

4.

**A —> B, C**

**A, C —> D**

**C, D —> A**

*not minimal, as illustrated above*

Consider the relation R(A, B, C, D) with functional dependencies (FDs)

A —> B, C

A, C —> D

C, D —> A, B

All choices for Q-w10 question 1

5.

**A —> B, C**

**A, C —> D**

**C, D —> B**

*not minimal and not equivalent*

Consider the relation R(A, B, C, D) with functional dependencies (FDs)

A —> B, C

A, C —> D

C, D —> A, B

**QUESTION 2 of Q-w10**

Consider the relation R(A, B, C, D) with functional dependencies (FDs)

B, C —> A

A, D —> C

Select all true statements

1. **R(A, B, C, D) is in BCNF**

*no, both FDs that were given have left hand sides that are NOT keys of R; both violate the BCNF condition*

**2. Each of R1(A, B, C) and R2(B, C, D) is in BCNF, where R1 and R2 are a decomposition of R**

*yes, B,C→ is assignable to R1 and B,C is a key of R1; no FD is assignable to R2. This no FD violates BCNF condition for either relation*

3. **Relation R has exactly one minimal key**

*no, R has two minimal keys*

B and D do not appear on the right-hand side of any FD; they cannot be inferred from anything else; they must be part of any key

The attribute closure of {B,D} is {B,D}, so not a key by itself

{A,B,D} is a key   and   {B,C,D} is a key

4. **The decomposition of R into R3(A, B, C) and R4(A, C, D) is lossless**

no, A,C is a basis for natural join, but not without losing information (A,C determines nothing else, and under-constrains join). A lossless decomposition of a relation R, with tuples T, is a set of smaller relations for which a natural join of those smaller relations gives back R (both the relational schema and its tuples Consider R = A B C D with two tuples (and consistent with the given FDs, B,C→A and A,D→C)

1 3 2 4

1 5 2 6

Represent these as R3 and R4

A B C and A C D

1 3 2 1 2 4

1 5 2 1 2 6

Do a natural join (on A and C), and get back

A B C D

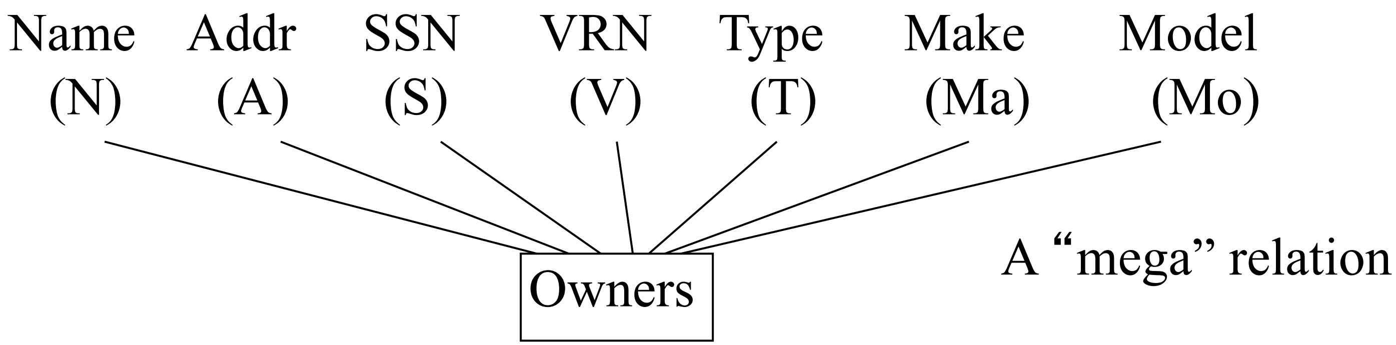
1 3 2 4

1 3 2 6

1 5 2 4

1 5 2 6

Assume the following relational schema covering vehicle ownership data (forgive lack of key, or assume that all attributes form the key, for now).




Individual persons, uniquely identified by SSN, are stored with their Name and Addr(ess), and are stored with information of the vehicles they own, where each vehicle is uniquely Identified by a Vehicle Registration Number (VRN), its Type (auto, truck, motorcycle), Manufacturer (aka Make), and Model. A sample of a database fitting this schema is below.

| 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 |

**Given your domain knowledge of vehicle ownership relationships in the real world, list functional dependencies that you believe should be asserted as true/required of this relational schema (and be enforced in the database).**

Assume the following functional dependencies apply to the schema

$S \rightarrow N$, $S \rightarrow A$, $V \rightarrow T$, $V \rightarrow Ma$, $V \rightarrow Mo$, $Mo \rightarrow Ma$, $Mo \rightarrow T$, $V \rightarrow S$

| Name | Addr | SSN | VRN | Type | Make | Model |
|---|---|---|---|---|---|---|
| (N) | (A) | (S) | (V) | (T) | (Ma) | (Mo) |

**State the constraint implied by each FD in English.**

| 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 | |
| Sri | NewYork | 234 | 654 | Car | VW | Bug | * |
| Mary | LosAngeles | 456 | 876 | Car | Toyota | Corolla | * |
| Fred | Nshville | 123 | 543 | Car | Honda | Accord | * |

**Each of the final three rows cause a violation of at least one FD (given the earlier rows). Identify the FDs that are violated in each case.**

Assume the following functional dependencies apply to the schema

$S \rightarrow N$, $S \rightarrow A$, $V \rightarrow T$, $V \rightarrow Ma$, $V \rightarrow Mo$, $Mo \rightarrow Ma$, $Mo \rightarrow T$, $V \rightarrow S$

| Name (N) | Addr (A) | SSN (S) | VRN (V) | Type (T) | Make (Ma) | Model (Mo) |
|---|---|---|---|---|---|---|

Each vehicle associated with one owner (1..1) or at most one (0..1). Nothing about these FDs implies that a Person need be an owner (0..*)

| 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 | |
| ~~Sri~~ | ~~NewYork~~ | ~~234~~ | ~~654~~ | ~~Car~~ | ~~VW~~ | ~~Bug~~ | ($V \rightarrow S$) |
| ~~Mary~~ | ~~LosAngeles~~ | ~~456~~ | ~~876~~ | ~~Car~~ | ~~Toyota~~ | ~~Corolla~~ | ( $V \rightarrow Mo, V \rightarrow S$) |
| ~~Fred~~ | ~~Nshville~~ | ~~123~~ | ~~543~~ | ~~Car~~ | ~~Honda~~ | ~~Accord~~ | ($S \rightarrow A$) |

**Of the FDs given, which are redundant (and not needed because the can be inferred from the remaining FDs)?**

**Give the key(s) of the relation above, as dictated by the FDs.**

Are the FDs minimal? Can we infer an FD from the other FDs?

S $\rightarrow$ N, S $\rightarrow$ A, V $\rightarrow$ T, V $\rightarrow$ Ma, V $\rightarrow$ Mo, Mo $\rightarrow$ Ma, Mo $\rightarrow$ T, V $\rightarrow$ S

From V $\rightarrow$ Mo and Mo $\rightarrow$ Ma we know V $\rightarrow$ Ma (V $\rightarrow$ Ma not needed)
From V $\rightarrow$ Mo and Mo $\rightarrow$ T we know V $\rightarrow$ T (V $\rightarrow$ T not needed)

A minimal set of FDs (not necessarily unique set):

S $\rightarrow$ N, S $\rightarrow$ A, ~~V $\rightarrow$ T,~~ ~~V $\rightarrow$ Ma,~~ V $\rightarrow$ Mo, Mo $\rightarrow$ Ma, Mo $\rightarrow$ T, V $\rightarrow$ S

Systematic algorithm for determining minimal set of FDs (and …) uses the *Attribute Closure algorithm* (if you know attribute A, what other attributes, B, can be determined)

*Extend the set using any FD with a LHS that is member of current set*

Attribute closure of N is {N} N not on LHS of any FD
Attribute closure of A is {A}
Attribute closure of S is {S} $\xrightarrow{S \rightarrow N}$ {S, N} $\xrightarrow{S \rightarrow A}$ {S, N, A}
Attribute closure of T is {T}
Attribute closure of Ma is {Ma}
Attribute closure of Mo is {Mo} $\xrightarrow{Mo \rightarrow T}$ {Mo, T} $\xrightarrow{Mo \rightarrow Ma}$ {Mo, T, Ma}

$S \rightarrow N$, $S \rightarrow A$, $V \rightarrow T$, $V \rightarrow Ma$, $V \rightarrow Mo$, $Mo \rightarrow Ma$, $Mo \rightarrow T$, $V \rightarrow S$

Attribute closure of V is $\{V\} \xrightarrow{V \rightarrow Mo} \{V, Mo\} \xrightarrow{Mo \rightarrow Ma} \{V, Mo, Ma\}$

$\xrightarrow{Mo \rightarrow T} \{V, Mo, Ma, T\} \xrightarrow{V \rightarrow S} \{V, Mo, Ma, T, S\}$

$\xrightarrow{S \rightarrow N} \{V, Mo, Ma, T, S, N\} \xrightarrow{S \rightarrow A} \underline{\{V, Mo, Ma, T, S, N, A\}}$

Note that the attribute closure of V includes ALL attributes. **V is a key.**

In general, a key is any minimal set of attributes with attribute closures whose union includes all attributes. Only the attribute closure of V contains V, so V is the only key (though there are many super keys).

If V were not an attribute (and all FDs involving V were removed), then S, Mo would be the only key. In general, however, there may be more than one Key.

S → N, S → A, V → T, V → Ma, V → Mo, Mo → Ma, Mo → T, V → S

Attribute closure of V is $\{V\} \xrightarrow{V \to Mo} \{V, Mo\} \xrightarrow{Mo \to Ma} \{V, Mo, Ma\}$

$\xrightarrow{Mo \to T} \{V, Mo, Ma, T\} \xrightarrow{V \to S} \{V, Mo, Ma, T, S\}$

$\xrightarrow{S \to N} \{V, Mo, Ma, T, S, N\} \xrightarrow{S \to A} \underline{\{V, Mo, Ma, T, S, N, A\}}$

***Incomplete*** Alg. for determining minimal FD set

Give an ordering of FDs (different orderings may lead to different minimal FD sets)

For each FD, A → B,

does attribute closure of A include B when A → B is excluded from derivation?

if so, eliminate A → B and continue

S → N,   S → A,  V → T,  V → Ma,  V → Mo,  Mo → Ma,  Mo → T,  V → S

***Incomplete*** Alg. for determining minimal FD set

Give an ordering of FDs  (different orderings may lead to different minimal FD sets)

For each FD, A → B,

does attribute closure of A include B when A → B is excluded from derivation?

if so, eliminate A → B and continue

S → N,   S → A,  V → T,  V → Ma,  V → Mo,  Mo → Ma,  Mo → T,  V → S

Attribute closure of S without S→N is {S, A}: have to keep S→N
Attribute closure of S without S→A is {S, N}: have to keep S→A
Attribute closure of V without V→T is still {V, Mo, Ma, T, S, N, A}: can remove V→T
Attribute closure of V without V→Ma (and without V→T) is still {V, Mo, Ma, T, S, N, A}: can remove V→Ma
Confirm that we would have to keep V → Mo,  Mo → Ma,  Mo → T,  V → S

Consider FDs of form X → B, where X is a set of attributes and B is a single attribute. For example, we might assert N,A → S and V, Mo → Ma in addition to the other FDs asserted previously (but excluding V → Ma and Mo → Ma).

S → N, S → A, V → T, V → Ma, V → Mo, Mo → Ma, Mo → T, V → S, N,A → S, V, Mo → Ma

<u>**Complete** Alg. For determining minimal FD set</u>

For each FD, X → B, with non-unit LHS, X

if any proper subset of X, Y, determines remainder of X (X – Y), then replace X by Y (replace X → B by Y → B)

*V, Mo → Ma replaced by V → Ma*

Give an ordering of the FDs (different orderings may result in different minimal FD sets)

For each FD, Z → B,

does *generalized* attribute closure of Z include B when Z → B is excluded from derivation? (e.g., generalized attribute closure of {N,A} is <u>{N,A, S}</u>)

if so, eliminate Z → B from set of FDs and continue *(V → T eliminated; V → Ma not)*

S → N, S → A, V → Mo, Mo → Ma, Mo → T, V → S **(or S → N, A ; Mo → Ma, T ; V → Mo, S)**

| 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 |

**Show a decomposition of the mega relation into BCNF relations using the FDs above**

Give the key(s) of the relation above, as dictated by the FDs.

V is NOT on the right-hand side (RHS) of any FD. Thus, V must be part of any key (i.e., the only way to infer V is to be given V).

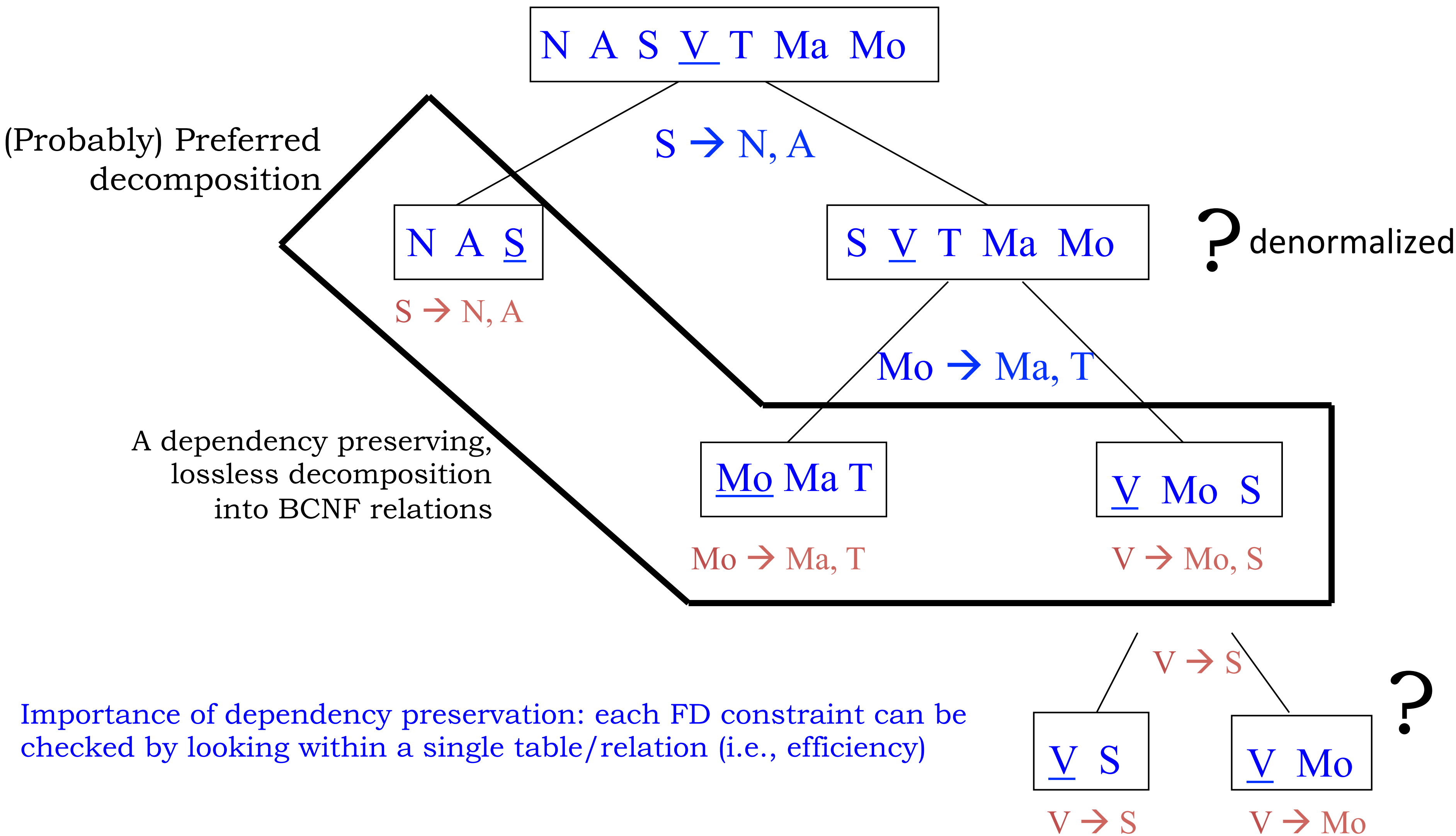
Is V alone a key? Yes. Attribute closure of V is {V} $\xrightarrow{V \to Mo}$ {V, Mo} $\xrightarrow{Mo \to Ma}$ {V, Mo, Ma}

$\xrightarrow{Mo \to T}$ {V, Mo, Ma, T} $\xrightarrow{V \to S}$ {V, Mo, Ma, T, S}

$\xrightarrow{S \to N}$ {V, Mo, Ma, T, S, N} $\xrightarrow{S \to A}$ <u>{V, Mo, Ma, T, S, N, A}</u>

Are there any other (minimal) keys?

A dependency preserving decomposition: every FD of a minimal set of is assignable to a relation in the decomposition




Importance of dependency preservation: each FD constraint can be checked by looking within a single table/relation (i.e., efficiency)

**Write CREATE TABLE statements for each of the three relations in the preferred decomposition**

Dependency preservation can be subtle

For example, consider R(A, B, C) and FDs $A \rightarrow B$ $B \rightarrow C$ $C \rightarrow A$

Is R1(A,B) and R2(B,C) a dependency preserving decomposition of R(A, B, C)?

$A \rightarrow B$ is assignable to R1(A,B)

$B \rightarrow C$ is assignable to R2(B,C)

But $C \rightarrow A$ is not assignable to either R1 or R2

So, initial thought is that R1 and R2 are not a dependency preserving decomposition

But it is dependency preserving

Dependency preservation can be subtle

For example, consider R(A, B, C) and FDs  A→B   B→C   C→A

Is R1(A,B) and R2(B,C) a dependency preserving decomposition of R(A, B, C)?

A→B is assignable to R1(A,B)

B→C is assignable to R2(B,C)

But C→A is not assignable to either R1 or R2

So, initial thought is that R1 and R2 are not a dependency preserving decomposition

But it is dependency preserving

Consider that A→B   B→A   B→C   C→B  is an alternative minimal set that is equivalent.
All four FDs are assignable to either R1 or R2

| 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 |

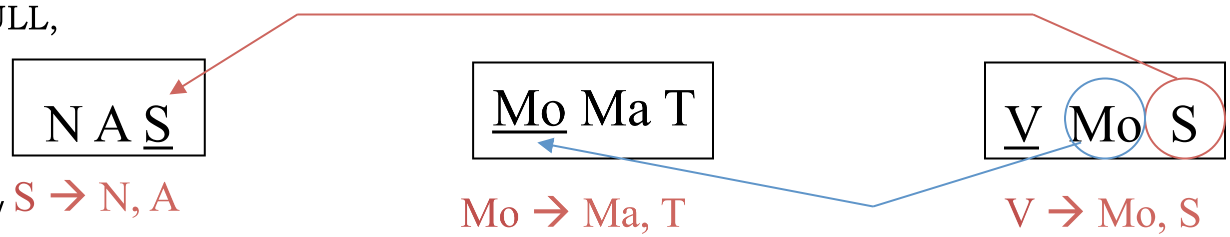
| S | V | Mo |
|---|---|---|
| 123 | 987 | Ranger |
| 234 | 876 | Camry |
| 345 | 765 | Hog |
| 123 | 654 | Bug |

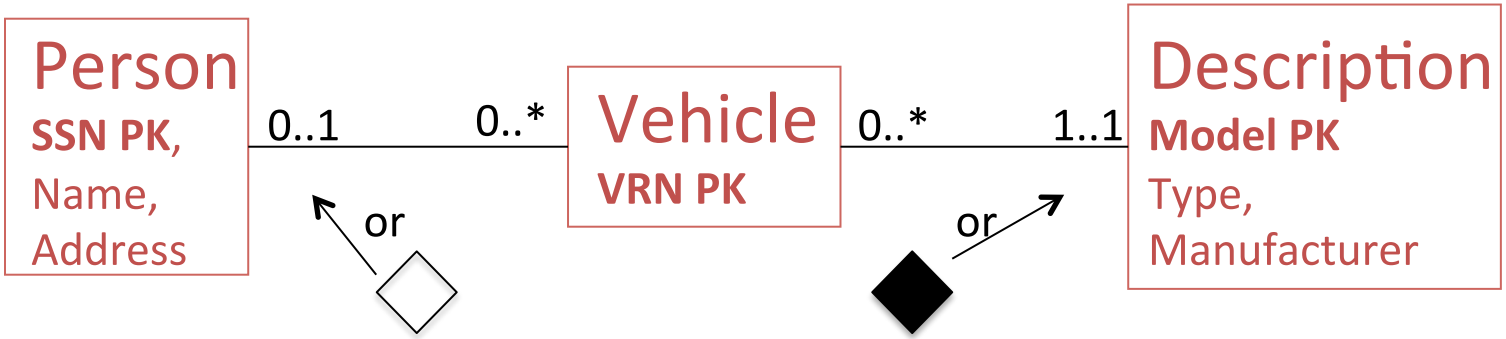
```
CREATE TABLE Person (
    Name VARCHAR(60) NOT NULL,
    Address VARCHAR(120) NOT NULL,
    SSN INTEGER PRIMARY KEY
);
```

**Give a UML diagram that is consistent with these Table definitions.**

```
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)