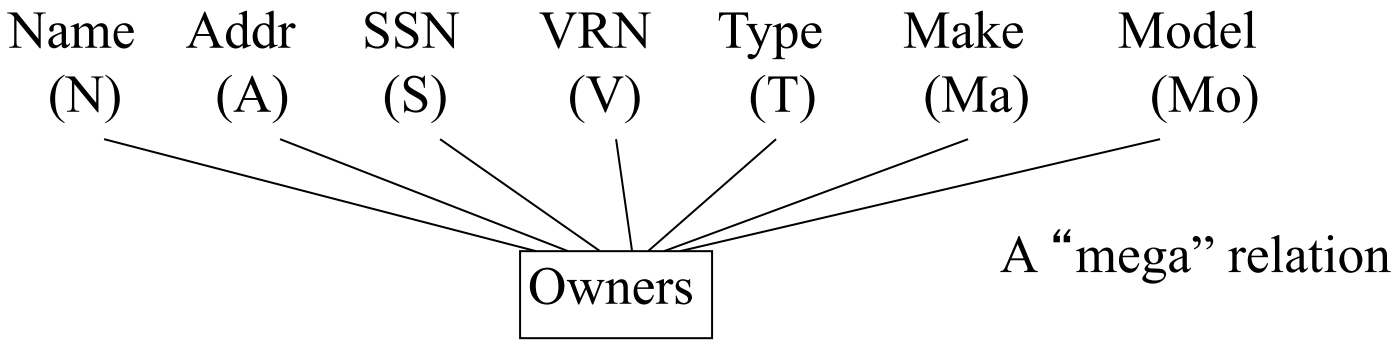
Assume the following relational schema covering vehicle ownership data (forgive lack of key, or assume that all attributes form the key, for now).



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

Name (N)	Addr (A)	SSN (S)	VRN (V)	Type (T)	Make (Ma)	Model (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	*

State the constraint implied by each FD in English.

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.

Which of the FDs above are not needed given (i.e., can be inferred from) the other FDs?

Assume the following functional dependencies apply to the schema

S → N, S → A, ~~V → T, V → Ma,~~ V → Mo, Mo → Ma, Mo → T, V → 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→S)
~~Mary~~	~~LosAngeles~~	~~456~~	~~876~~	~~Car~~	~~Toyota~~	~~Corolla~~	(V→Mo,V→S)
~~Fred~~	~~Nshville~~	~~123~~	~~543~~	~~Car~~	~~Honda~~	~~Accord~~	(S→A)

Give an example FD that can be inferred from the FDs above (other than the ones crossed out).

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

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

Give an example FD that can be inferred from the FDs above: e.g., V→A; V→N, V,Na → S, T, …

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} → (V → Mo) {V, Mo} → (Mo → Ma) {V, Mo, Ma}

→ (Mo → T) {V, Mo, Ma, T} → (V → S) {V, Mo, Ma, T, S}

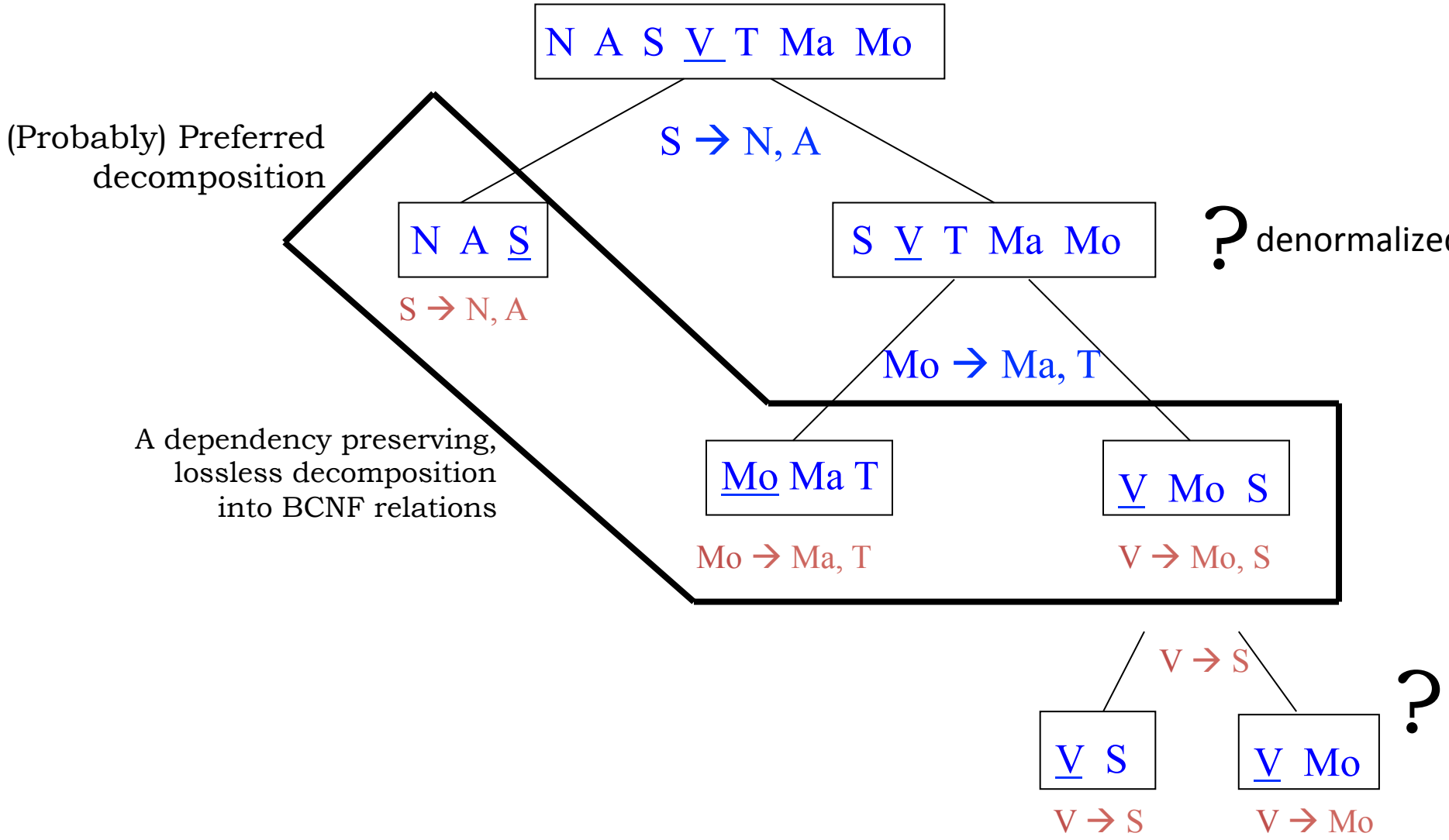
Are there any other (minimal) keys?

→ (S → N) {V, Mo, Ma, T, S, N} → (S → A) {V, Mo, Ma, T, S, N, A}

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

Importance of lossless decomposition: obvious, don't want to lose information

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

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

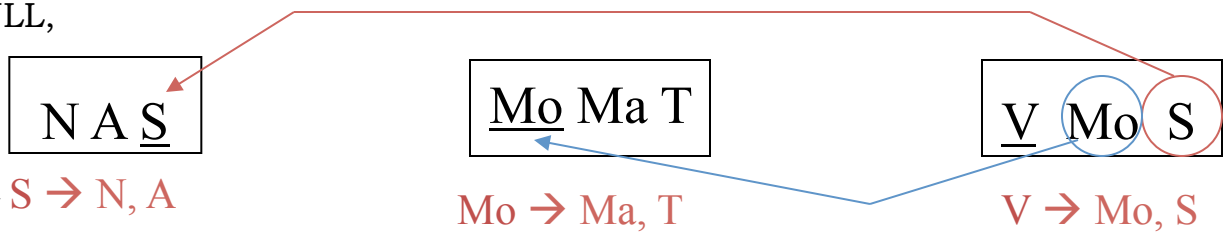
S	V	Mo
123	987	Ranger
234	876	Camry
345	765	Hog
123	654	Bug

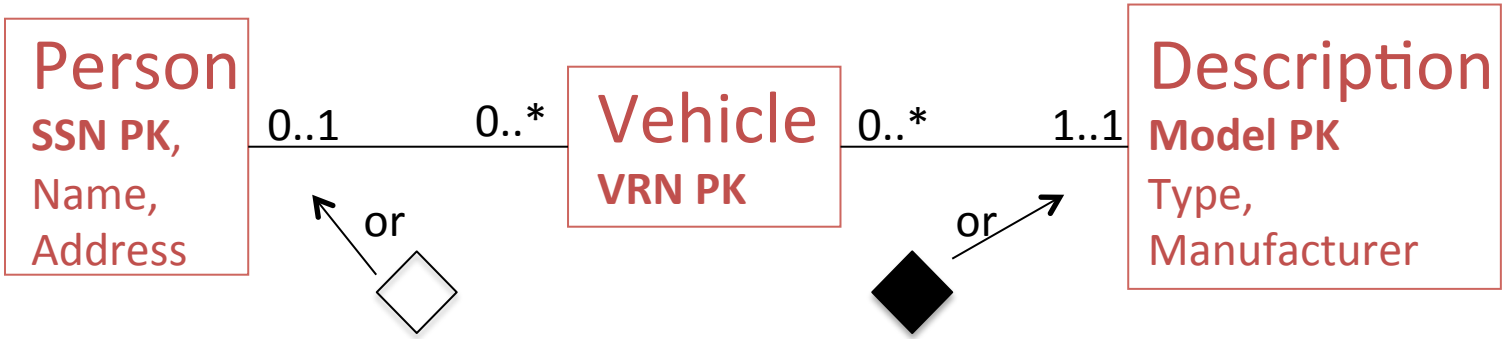
```
CREATE TABLE Person (
      Name VARCHAR(60) NOT NULL,
      Address VARCHAR(120) NOT NULL,
      SSN INTEGER PRIMARY KEY
);
```

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);

Study exercises

1. What are the limitations of the previous database design of vehicle/owner relationships?

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

A → B , BCD → E , E → 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 attribute F

3. Suppose that you have a relation Q (A , B, C, D, E) with only one FD AB → CDE . 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 AB → CD and D → EF. 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.

Study exercises

1. What are the limitations of the previous database design of vehicle/owner relationships?

I think that the biggest limitation is that the DB design 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.

There are other limitations – perhaps we would want to allow multiple simultaneous owners as well, for example

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

A → B , BCD → E , E → 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 attribute F? **8**

(At most 2 values for A) AND (A→B) means **(At most 2 values for B)**

(**At most 2 values for B)** AND (At most 2 values for C) AND (At most 2 values for D) means **(At most 8 combinations of B,C,D composite values)**

(At most 8 combinations of B,C,D composite values) AND (BCD→E) means **(At most 8 values for E)**

(At most 8 values for E) AND (E→F) means **(At most 8 values for F)**

3. Suppose that you have a relation Q (A , B, C, D, E) with only one FD AB → CDE . 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).

AB must be part of any key for Q, since A and B do not appear on right-hand side Of any FD.

KEY for Q is AB, since all attributes (ABCDE) determined by AB alone

ALL FDs asserted of Q have a left-hand side that is a key of Q; Q is already in BCNF

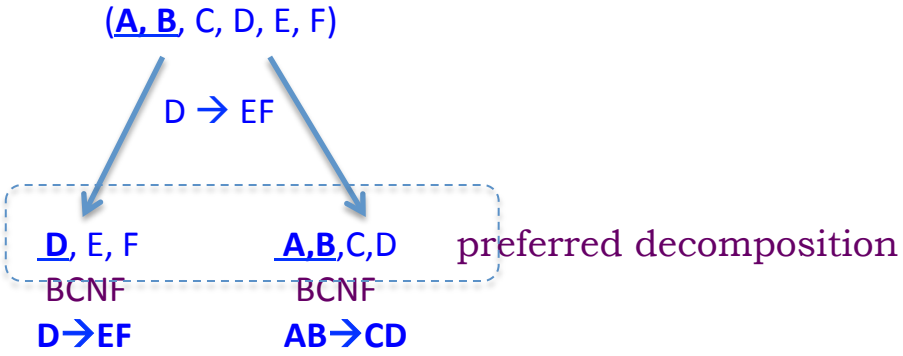
4. Suppose that you have a relation R (A, B, C, D, E, F) with FDs AB → CD and D → EF. Decompose R into a set of relations, EACH of which is in BCNF, or state that R is already in BCNF (and in either case, explain your answer, and in doing so, identify the key for each relation).

AB must be part of any key for R, since A and B do not appear on right-hand side of any FD.

KEY for R is AB, since all attributes (ABCDEF) determined by AB alone

R is NOT in BCNF since D→EF violates BCNF constraint (left-hand side of D→EF is Not a key of R)

Decompose R using an offending FD



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)

Of course, FDs with ISBN as left-hand side LHS

Also,

AmazonPrice , SavingsInPrice → ListPrice
ListPrice, SavingsInPrice → AmazonPrice
AmazonPrice , ListPrice → SavingsInPrice

Title, Format, PublicationDate, PublisherName → ISBN (????)

```
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,   /* redundant
   AveShipLag          INTEGER,
   AveCustRating       REAL,
   SalesRank           INTEGER,
   CoverArt            FILE,
   Format              CHAR[4] NOT NULL,
   CopiesInStock       INTEGER,
   PublisherName       CHAR[120] NOT NULL,  //Remove NOT NULL if you want 0 or 1
   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.

b) Give all keys for the relation

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.

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 (see (a) below)
~~**5)** Dorm, Floor#, Date, Time, Temp → Kwatts~~ (see (a) below)
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.

Can eliminate 4 or 5, but not both – I expect you to have found this. Can also eliminate Temp from 4 (or 5) because of 2

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

Date, Time must be part of any key (they are not on RHS of any FD; but attribute closure of Date,Time = {Date, Time, Temp, Humidity, Weekday?} so Date and Time alone not a key

Date,Time, SensorID is a key
Date,Time,Dorm, Floor is a 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.

1) Do, F → O
~~4') S, Da, Ti → K~~
6) S → Do, F

2) Da, Ti → Te, Hu
5') Do, Fl, Da, Ti → K
7) Do, F → S

3) Da → W

Note that this is an alternative Minimal set to that identified in 6 (a) answer

Keys: SensorId, Date, Time (S, Da, Ti)
Dorm, Floor#, Date, Time (Do, F, Da, Ti)

All FDs from this minimal set, except 5' (and 4', if it were in the min set), violate BCNF

Kwatts, Dorm, Floor#, Date, Time, Temp, Humidity, Occupancy, Weekday?, SensorID
K Do F Da Ti Te H O W S

Do, F → O, S

Do, F, O, S

why doesn't FD 6 violate BCNF?

Because this table covers S→DoF too, and S is a key of the table too. In an SQL CREATE TABLE statement, we might declare Do,F as PRIMARY KEY and S as UNIQUE, or vice versa

K, Do, F, Da, Ti, Te, H, W (FDs 2, 3 violate BCNF)

Da, Ti → Te, Hu

Da, Ti, Te, Hu

K, Do, F, Da, Ti, W (FD 3 violates BCNF)

Da → W

Da, W

K, Do, F, Da, Ti

Other answers possible