University of British Columbia, Vancouver
Department of Computer Science

Project Summary

Our project is about modeling PC components to allow users to understand how different parts of a PC work together and what is required to create your own PC. Using this database, users will be able to look up the price of different PC parts and customize their own PC. The final build of the PC will indicate whether all the parts are compatible with each other as well as the final price of the PC.

ER Diagram Modifications

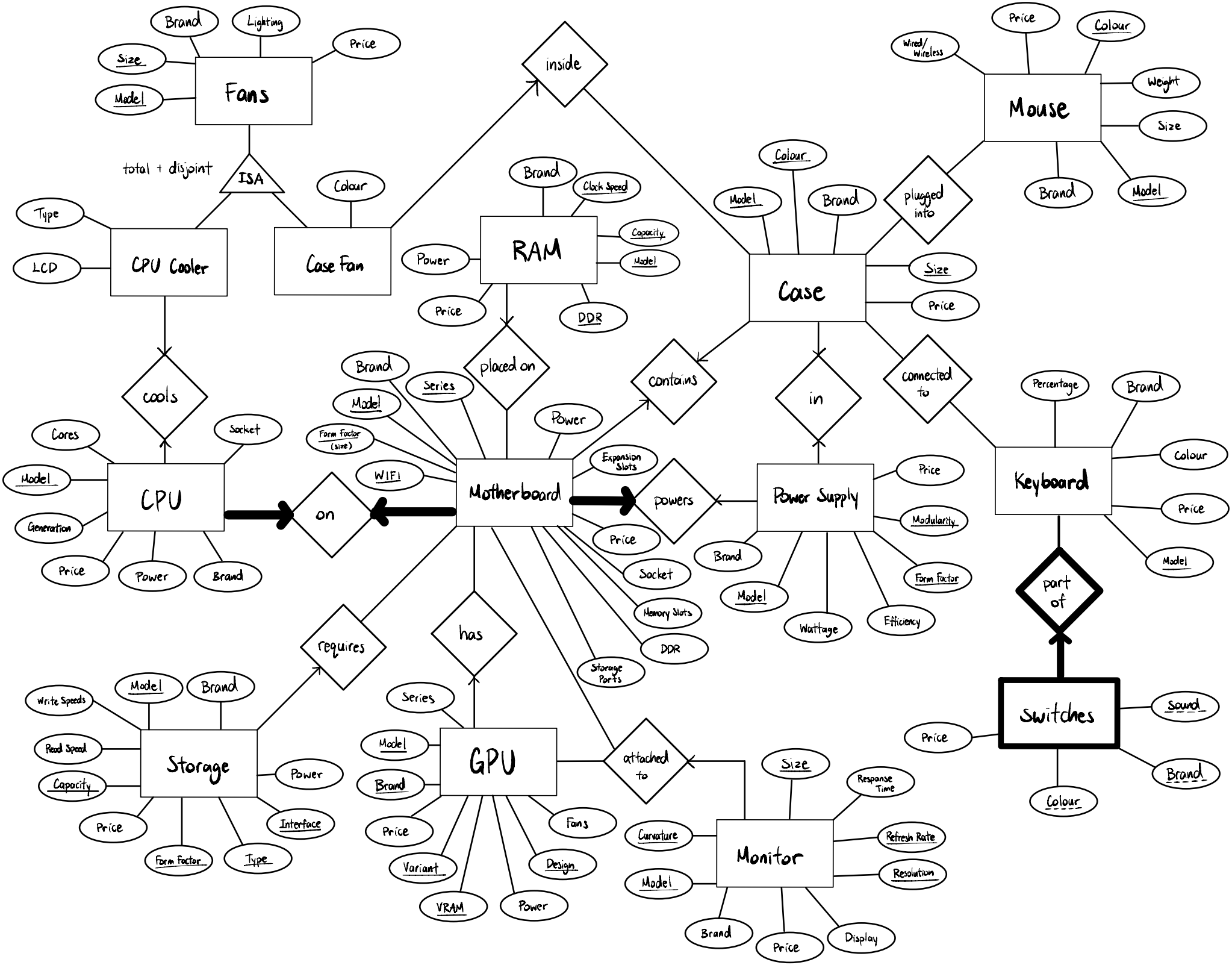
Based on feedback from milestone 1, we have made changes to these constraints in the ER diagram:

1. Relationship “plugged into” is now many-to-many as many mouse can be plugged into a single case and many cases can be connected to the same mouse
2. The relationship “connected to” is now many-to-many as many keyboards can be plugged into a single case and many cases can be connected to the same keyboard
3. The relationship “has” is now many-to-one as a motherboard can have multiple GPUs but a GPU can only be on one motherboard.

We have also deleted the ISA hierarchy of the entity set Storage since it does not provide any additional attributes so it is not valuable.

Many changes to the attributes and keys for entities from the feedback given in milestone 1 (Note brand determines model for all entities except for GPU):

1. CPU entity: model (Core i7-12700K, Ryzen 7 7800X3D, etc.) is now the key
2. Fans entity: model, size, lighting are now the key
    a. CPU Cooler entity: key is the same as Fans entity key
    b. Case fan entity: key is the same as Fans entity key
3. Storage entity: model, capacity, form factor, type, interface are now the key
4. GPU entity: model, brand, variant, VRAM, design are now the key
5. Motherboard entity: model, series, form factor (size), wifi are now the key
6. RAM entity: model, clock speed, capacity, DDR are now the key
7. Power supply entity: model (has wattage), form factor, modularity are the key
8. Case entity: model, colour, size are now the key
9. Monitor entity: model, curvature, size, refresh rate, resolution are now the key
10. Keyboard entity: model is now the key
    a. Switches entity: sound, brand, colour are part of the key
11. Mouse entity: model, colour are now the key


Brand
Lighting
Size
Price
Model
Fans
total + disjoint
ISA
Type
LCD
CPU Cooler
Colour
Case Fan
inside
Brand
Clock Speed
Power
RAM
Capacity
Model
Price
DDR
Price
Colour
Wired/ Wireless
Weight
Mouse
Size
Brand
Model
Colour
Model
Brand
plugged into
Size
Case
Price
cools
placed on
contains
in
connected to
Brand
Series
Model
Form Factor (size)
Power
WIFI
Expansion Slots
Cores
Socket
Model
CPU
Generation
Price
Power
Brand
on
Motherboard
powers
Price
Socket
Memory Slots
DDR
Storage Ports
Power Supply
Price
Modularity
Brand
Model
Wattage
Efficiency
Form Factor
Percentage
Brand
Colour
Price
Model
Keyboard
part of
requires
has
attached to
Write Speeds
Model
Brand
Read Speed
Capacity
Storage
Power
Price
Interface
Form Factor
Type
Series
Model
Brand
GPU
Price
Fans
Variant
Design
VRAM
Power
Size
Response Time
Curvature
Refresh Rate
Model
Monitor
Resolution
Brand
Price
Display
Price
Switches
Sound
Brand
Colour

University of British Columbia, Vancouver

Department of Computer Science

Schema

- Primary Keys are underlined, foreign keys are **bolded**

CPUCooler_Cools(Model: VARCHAR, Size: INT, Lighting: VARCHAR, Brand: VARCHAR, Price: NUMBER, LCD: VARCHAR, Type: VARCHAR, **CPU_Model**: VARCHAR)

- Primary key: {Model, Size}
- Candidate keys: {Model, Size}, {CPU_Model}
- Unique: CPU_Model

CaseFan_Inside(Model: VARCHAR, Size: INT, Lighting: VARCHAR, Brand: VARCHAR, Price: NUMBER, Colour: VARCHAR, **Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR)

- Primary key: {Model, Size}
- Candidate keys: {Model, Size}
- Foreign keys: {Case_Model, Case_Colour, Case_Size}

CPU_On(Model: VARCHAR, Cores: INT, Socket: VARCHAR, Generation: INT, Price: NUMBER, Power: INT, Brand: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model}
- Candidate keys: {Model}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Not null: MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- Unique: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

Motherboard_Powers(Model: VARCHAR, Form_Factor: VARCHAR, Series: VARCHAR, WIFI: VARCHAR, Brand: VARCHAR, Power: INT, Expansion_Slots: VARCHAR, Price: NUMBER, Socket: VARCHAR, Memory_Slots: INT, DDR: VARCHAR, Storage_Ports: INT, **CPU_Model**: VARCHAR, **PSU_Modularity**: VARCHAR, **PSU_Form_Factor**: VARCHAR, **PSU_Model**: VARCHAR)

- Primary key: {Model, Form_Factor, Series, WIFI}
- Candidate keys: {Model, Form_Factor, Series, WIFl}, {CPU_Model}, {PSU_Modularity, PSU_Brand, PSU_Form_Factor, PSU_Model}
- Foreign keys: {CPU_Model}, {PSU_Modularity, PSU_Form_Factor, PSU_Model}
- Not null: CPU_Model, PSU_Modularity, PSU_Form_Factor, PSU_Model
- Unique: {CPU_Model}, {PSU_Modularity, PSU_Form_Factor, PSU_Model}

Storage_Requires(Model: VARCHAR, Capacity: INT, Interface: VARCHAR, Type: VARCHAR, Form_Factor: VARCHAR, Brand: VARCHAR, Read_Speed: INT, Write_Speeds: INT, Power: INT, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Capacity, Interface, Type, Form_Factor}
- Candidate keys: {Model, Capacity, Interface, Type, Form_Factor}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

GPU_Has(Model: VARCHAR, Brand: VARCHAR, Variant: VARCHAR, VRAM: INT, Design: VARCHAR, Fans: INT, Power: INT, Series: VARCHAR, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Brand, Variant, VRAM, Design}
- Candidate keys: {Model, Brand, Variant, VRAM, Design}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

Monitor_Attached_To(Model: VARCHAR, Curvature: VARCHAR, Resolution: VARCHAR, Refresh_Rate: INT, Size: NUMBER, Response_Time: NUMBER, Brand: VARCHAR, Price: NUMBER, Display: VARCHAR, **GPU_Model**: VARCHAR, **GPU_Brand**: VARCHAR, **GPU_Variant**: VARCHAR, **GPU_VRAM**: INT, **GPU_Design**: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Curvature, Resolution, Refresh_Rate, Size}
- Candidate keys: {Model, Curvature, Resolution, Refresh_Rate, Size}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}, {GPU_Model, GPU_Brand, GPU_Variant, GPU_VRAM, GPU_Design}

RAM_Placed_On(Model: VARCHAR, DDR: VARCHAR, Capacity: INT, Clock_Speed: INT, Power: INT, Price: NUMBER, Brand: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, DDR, Capacity, Clock_Speed}
- Candidate keys: {Model, DDR, Capacity, Clock_Speed}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

PowerSupply_In(Model: VARCHAR, Brand: VARCHAR, Efficiency: VARCHAR, Form_Factor: VARCHAR, Modularity: VARCHAR, Wattage: INT, Price: NUMBER, **Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR)

- Primary key: {Model, Form_Factor, Modularity}
- Candidate keys: {Model, Form_Factor, Modularity}, {Case_Model, Case_Colour, Case_Size}
- Foreign keys: {Case_Model, Case_Colour, Case_Size}

- Unique: {Case_Model, Case_Colour, Case_Size}

Case_Contains(Model: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Size: VARCHAR, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Colour, Size}
- Candidate keys: {Model, Colour, Size}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Unique: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

Mouse(Model: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Size: VARCHAR: Weight: INT, Price: NUMBER, Wired_Wireless: VARCHAR)

- Primary key: {Model, Colour}
- Candidate keys: {Model, Colour}

Plugged_Into(**Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR, **Mouse_Model**: VARCHAR, **Mouse_Colour**: VARCHAR)

- Primary Key: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}
- Candidate Key: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}
- Foreign Key: {Case_Model, Case_Colour, Case_Size}, {Mouse_Model, Mouse_Colour}

Keyboard(Model: VARCHAR, Percentage: NUMBER, Brand: VARCHAR, Colour: VARCHAR, Price: NUMBER)

- Primary Key: {Model}
- Candidate Keys: {Model}

Switches(**Model**: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Sound: VARCHAR, Price: NUMBER)

- Primary Key: {Model, Brand, Colour, Sound}
- Candidate Keys: {Model, Brand, Colour, Sound}

Connected_To(**Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR, **Keyboard_Model**: VARCHAR)

- Primary Key: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}
- Candidate Key: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}
- Foreign Key: {Case_Model, Case_Colour, Case_Size}, {Keyboard_Model}

University of British Columbia, Vancouver
Department of Computer Science

Functional Dependencies & Decomposition:

Below are all the functional dependencies list for the table and the steps to decompose it into BCNF.

CPUCooler_Cools(Model, Size, Lighting, Brand, Price, LCD, Type, **CPU_Model**)

- Model, Size → Brand, Price, LCD, Type, CPU_Model
- CPU_Model → Model, Size, Lighting, Brand, Price, LCD, Type
- Model → Lighting, Brand, Type, LCD
- Size, Lighting, Brand, Type → Price

CaseFan_Inside(Model, Size, Lighting, Brand, Price, Colour, **Case_Model**, **Case_Colour**, **Case_Size**)

- Model, Size → Brand, Price, Colour, Case_Model, Case_Colour, Case_Size
- Model → Brand, Lighting
- Model, Colour, Size → Price

CPU_On(Model, Cores, Socket, Generation, Price, Power, Brand, **MB_Model**, **MB_Form_Factor**, **MB_Series**, **MB_WIFI**)

- Model → Cores, Socket, Generation, Price, Power, Brand, MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- MB_Model, MB_Form_Factor, MB_Series, MB_WIFI → Cores, Socket, Generation, Price, Power, Brand, Model
- Cores, Generation → Power

Motherboard_Powers(Model, Form_Factor, Series, WIFI, Brand, Power, Expansion_Slots, Price, Socket, Memory_Slots, DDR, Storage_Ports, **CPU_Model**, **PSU_Modularity**, **PSU_Form_Factor**, **PSU_Model**)

- Model, Form_Factor, Series, WIFI → Brand, Power, Expansion_Slots, Price, Socket, Memory_Slots, DDR, Storage_Ports, CPU_Model, PSU_Modularity, PSU_Form_Factor, PSU_Model
- CPU_Model → Model, Form_Factor, Series, WIFI, Brand, Power, Expansion_Slots, Price, Socket, Memory_Slots, DDR, Storage_Ports, PSU_Modularity, PSU_Form_Factor, PSU_Model
- PSU_Modularity, PSU_Form_Factor, PSU_Model → Model, Form_Factor, Series, WIFI, Brand, Power, Expansion_Slots, Price, Socket, Memory_Slots, DDR, Storage_Ports, CPU_Model
- CPU_Model → Socket

University of British Columbia, Vancouver

Department of Computer Science

Storage_Requires(Model, Capacity, Interface, Type, Form_Factor, Brand, Read_Speeds, Write_Speeds, Power, Price, **MB_Model**, **MB_Form_Factor**, **MB_Series**, **MB_WIFI**)

- Model, Capacity, Interface, Type, Form_Factor → Brand, Read_Speeds, Write_Speeds, Power, Price, MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- Model, Capacity, Type → Price
- Model → Read_Speeds, Write_Speeds, Brand

GPU_Has(Model, Brand,Variant, VRAM, Design, Fans, Power, Series, Price, **MB_Model**, **MB_Form_Factor**, **MB_Series**, **MB_WIFI**)

- Model, Brand, Variant, VRAM, Design → Fans, Power, Series, Price, MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- Model → Series, Fans
- Model, Brand, Variant, Fans, Design, VRAM, Series → Price
- Model, VRAM, Fans → Power

Monitor_Attached_To(Model, Curvature, Resolution, Refresh_Rate, Size, Response_Time, Brand, Price, Display, **GPU_Model**, **GPU_Brand**, **GPU_Variant**, **GPU_VRAM**, **GPU_Design**, **MB_Model**, **MB_Form_Factor**, **MB_Series**, **MB_WIFI**)

- Model, Curvature, Resolution, Refresh_Rate, Size → Response_Time, Brand, Price, Display, GPU_Model, GPU_Brand, GPU_Varient, GPU_VRAM, GPU_Design, MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- Model → Brand, Display

RAM_Placed_On(Model, DDR, Capacity, Clock Speed, Power, Price, Brand, **MB_Model**, **MB_Form_Factor**, **MB_Series**, **MB_WIFI**):

- Model, DDR, Capacity, Clock Speed → Power, Price, Brand, MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- Model → Brand
- DDR, Capacity → Power

PowerSupply_In(Model, Brand, Efficiency, Form_Factor, Modularity, Wattage, Price, **Case_Model**, **Case_Brand**, **Case_Colour**, **Case_Size**):

- Model, Form_Factor, Modularity → Brand, Wattage, Price, Case_Model, Case_Brand, Case_Colour, Case_Size
- Case_Model, Case_Colour, Case_Size → Model, Efficiency, Form_Factor, Modularity, Brand, Wattage, Price
- Model → Brand, Wattage, Efficiency

University of British Columbia, Vancouver
Department of Computer Science

Case_Contains(Model, Brand, Colour, Size, Price, **MB_Model**, **MB_Form_Factor**, **MB_Series**, **MB_WIFI**):

- Model, Colour, Size → Brand, Price, MB_Model, MB_Form_Factor, MB_Series, MB_WIFI
- MB_Model, MB_Form_Factor, MB_Series, MB_WIFI → Model, Brand, Colour, Size, Price
- Model → Brand

Mouse(Model, Brand, Colour, Size, Weight, Price, Wired_Wireless):

- Model, Colour → Brand, Size, Weight, Price, Wired_Wireless
- Model → Brand

Plugged_Into(**Case_Model**, **Case_Colour**, **Case_Size**, **Mouse_Name**, **Mouse_Colour**):

- Case_Model, Case_Colour, Case_Size, Mouse_Name, Mouse_Colour → Case_Model, Case_Colour, Case_Size, Mouse_Name, Mouse_Colour

Keyboard(Model, Percentage, Brand, Colour, Price):

- Model → Brand, Colour, Price, Percentage

Switches(**Model**, Brand, Colour, Sound, Price):

- Model, Brand, Colour → Price

Connected_To(**Case_Model**, **Case_Colour**, **Case_Size**, **Keyboard_Model**) :

- Case_Model, Case_Colour, Case_Size, Keyboard_Model → Case_Model, Case_Colour, Case_Size, Keyboard_Model

University of British Columbia, Vancouver
Department of Computer Science

Decomposition

- Tables in Red are in BCNF

CPUCooler_Cools(Model: VARCHAR, Size: INT, Lighting: VARCHAR, Brand: VARCHAR, Price: NUMBER, LCD: VARCHAR, Type: VARCHAR, **CPU_Model**: VARCHAR)

- Primary key: {Model, Size}
- Candidate keys: {Model, Size}, {CPU_Model}
- Model ⟶ Lighting, Brand, Type, LCD
- Size, Lighting, Brand, Type ⟶ Price

Decomposing on Model ⟶ Lighting, Brand, Type, LCD gives us
R_1(Model: VARCHAR, Size: INT, Price: NUMBER, **CPU_Model**: VARCHAR)

- PK: {Model, Size}
- CK: {Model, Size}, {CPU_Model}
- FK: {CPU_Model}

R_2(Model: VARCHAR, Lighting: VARCHAR, Brand: VARCHAR, Type: VARCHAR, LCD: VARCHAR)

- PK: {Model}
- CK: {Model}

CaseFan_Inside(Model: VARCHAR, Size: INT, Lighting: VARCHAR, Brand: VARCHAR, Price: NUMBER, Colour: VARCHAR, **Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, Case_Brand: VARCHAR, **Case_Size**: VARCHAR)

- Primary key: {Model, Size}
- Candidate keys: {Model, Size}
- Foreign keys: {Case_Model, Case_Colour, Case_Size}
- Model ⟶ Brand, Lighting
- Model, Colour, Size ⟶ Price This is a super key so in BCNF

Decomposing on Model ⟶ Brand, Lighting gives us
R_1(Model: VARCHAR, Size: INT, Price: NUMBER, Colour: VARCHAR, **Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR)

- PK: {Model, Size}
- CK: {Model, Size}, {Case_Model, Case_Colour, Case_Size}
- FK: {Case_Model, Case_Colour, Case_Size}

R_2(Model: VARCHAR, Brand: VARCHAR, Lighting: VARCHAR)

- PK: {Model}
- CK: {Model}

CPU_On(Model: VARCHAR, Cores: INT, Socket: VARCHAR, Generation: INT, Price: NUMBER, Power: INT, Brand: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model}
- Candidate keys: {Model}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Cores, Generation ⟶ Power

Decomposing on Cores, Generation⟶ Power gives us

R_1(Cores: VARCHAR, Generation: INT, Power: INT)

- PK: {Cores, Generation}
- CK: {Cores, Generation}

R_2(Cores: VARCHAR, Generation: INT, Model:VARCHAR , Socket: VARCHAR, Price: Number, Brand: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model}
- CK: {Model}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

Motherboard_Powers(Model: VARCHAR, Form_Factor: VARCHAR, Series: VARCHAR, WIFI: VARCHAR, Brand: VARCHAR, Power: INT, Expansion_Slots: VARCHAR, Price: NUMBER, Socket: VARCHAR, Memory_Slots: INT, DDR: VARCHAR, Storage_Ports: INT, **CPU_Model**: VARCHAR, **PSU_Modularity**: VARCHAR, **PSU_Form_Factor**: VARCHAR, **PSU_Model**: VARCHAR)

- Primary key: {Model, Form_Factor, Series, WIFI}
- Candidate keys: {Model, Form_Factor, Series, WIFI}, {CPU_Model}, {PSU_Modularity, PSU_Brand, PSU_Form_Factor, PSU_Model}
- Foreign keys: {CPU_Model}, {PSU_Modularity, PSU_Form_Factor, PSU_Model}
- CPU_Model ⟶ Socket

Decomposing on CPU_Model → Socket gives us
R_1(Model: VARCHAR, Form_Factor: VARCHAR, Series: VARCHAR, WIFI: VARCHAR, Brand: VARCHAR, Power: INT, Expansion_Slots: VARCHAR, Price: NUMBER, Memory_Slots: INT, DDR: VARCHAR, Storage_Ports: INT, **CPU_Model**: VARCHAR, **PSU_Modularity**: VARCHAR, **PSU_Form_Factor**: VARCHAR, **PSU_Model**: VARCHAR)

- PK {Model, Form_Factor, Series, WIFI}
- CK: {Model, Form_Factor, Series, WIFI}, {CPU_Model}, {PSU_Modularity, PSU_Form_Factor, PSU_Model}
- FK: {CPU_Model}, {PSU_Modularity, PSU_Form_Factor, PSU_Model}

R_2(CPU_Model: VARCHAR, Socket: VARCHAR)

- PK: {CPU_Model}
- CK: {CPU_Model}

Storage_Requires(Model: VARCHAR, Capacity: INT, Interface: VARCHAR, Type: VARCHAR, Form_Factor: VARCHAR, Brand: VARCHAR, Read_Speed: INT, Write_Speeds: INT, Power: INT, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Capacity, Interface, Type, Form_Factor}
- Candidate keys: {Model, Capacity, Interface, Type, Form_Factor}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Model, Capacity, Type → Price
- Model → Read_Speeds, Write_Speeds, Brand

Decomposing on Model → Read_Speeds, Write_Speeds, Brand gives us
R_1(Model: VARCHAR, Capacity: INT, Interface: VARCHAR, Type: VARCHAR, Form_Factor: VARCHAR, Power: INT, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, Capacity, Interface, Type, Form_Factor}
- CK: {Model, Capacity, Interface, Type, Form_Factor}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

R_2(Model: VARCHAR, Read_Speeds: INT, Write_Speeds: INT, Brand: VARCHAR)

- PK: {Model}
- CK: {Model}

Decomposing on Model, Capacity, Type → Price gives us
R_3(Model: VARCHAR, Capacity: INT, Interface: VARCHAR, Type: VARCHAR, Form_Factor: VARCHAR, Power: INT, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, Capacity, Interface, Type, Form_Factor}
- CK: {Model, Capacity, Interface, Type, Form_Factor}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

R_4(Model: VARCHAR, Capacity: INT, Type: VARCHAR, Price: NUMBER)

- PK: {Model, Capacity, Type}
- CK: {Model, Capacity, Type}

GPU_Has(Model: VARCHAR, Brand: VARCHAR, Variant: VARCHAR, VRAM: INT, Design: VARCHAR, Fans: INT, Power: INT, Series: VARCHAR, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Brand, Variant, VRAM, Design}
- Candidate keys: {Model, Brand, Variant, VRAM, Design}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Model → Series, Fans
- Model, Brand, Variant, Fans, Design, VRAM, Series → Price
- Model, VRAM, Fans → Power

Decomposing Model → Series, Fans gives us
R_1(Model: VARCHAR, Fans: INT, Series: VARCHAR)

- PK: {Model}
- CK: {Model}

R_2(Model: VARCHAR, Brand: VARCHAR, Variant: VARCHAR, VRAM: INT, Design: VARCHAR, Power: INT, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, Brand, Variant, VRAM, Design}
- CK: {Model, Brand, Variant, VRAM, Design}{MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

Decomposing Model, Brand, Variant, Fans, Design, VRAM, Series → Price gives us
R_3(Model: VARCHAR, Brand: VARCHAR, Variant: VARCHAR, VRAM: INT, Design: VARCHAR, Fans: INT, Series: VARCHAR, Price: NUMBER)

- PK: {Model, Brand, Variant, VRAM, Design}
- CK: {Model, Brand, Variant, VRAM, Design}

R_4(Model: VARCHAR, VRAM: INT, Fans: INT, Brand: VARCHAR, Variant: VARCHAR, Design: VARCHAR, Power: INT, Series: VARCHAR,**MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, Brand, Variant, VRAM, Design}
- CK: {Model, Brand, Variant, VRAM, Design}{MB_Model, MB_Form_Factor, MB_Series, MB_WIFI)
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI)

Decomposing Model, VRAM, Fans → Power gives us
R_5(Model: VARCHAR, VRAM: INT, Fans: INT, Power: INT)

- PK: {Model, VRAM}
- CK: {Model, VRAM}

R_6(Model: VARCHAR, VRAM: INT, Fans: INT, Brand: VARCHAR, Variant: VARCHAR, Design: VARCHAR, Series: VARCHAR,**MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, Brand, Variant, VRAM, Design}
- CK: {Model, Brand, Variant, VRAM, Design}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

Monitor_Attached_To(Model: VARCHAR, Curvature: VARCHAR, Resolution: VARCHAR, Refresh_Rate: INT, Size: NUMBER, Response_Time: NUMBER, Brand: VARCHAR, Price: NUMBER, Display: VARCHAR, **GPU_Model**: VARCHAR, **GPU_Brand**: VARCHAR, **GPU_Variant**: VARCHAR, **GPU_VRAM**: INT, **GPU_Design**: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Curvature, Resolution, Refresh_Rate, Size}
- Candidate keys: {Model, Curvature, Resolution, Refresh_Rate, Size}

- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}, {GPU_Model, GPU_Brand, GPU_Variant, GPU_VRAM, GPU_Design}
- Model → Brand, Display

Decomposing on Model → Brand, Display gives us
R_1(Model: VARCHAR, Curvature: VARCHAR, Resolution: VARCHAR, Refresh_Rate: INT, Size: NUMBER, Response_Time: NUMBER, Price: NUMBER, **GPU_Model**: VARCHAR, **GPU_Brand**: VARCHAR, **GPU_Variant**: VARCHAR, **GPU_VRAM**: INT, **GPU_Design**: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, Curvature, Resolution, Refresh_Rate, Size}
- CK: {Model, Curvature, Resolution, Refresh_Rate, Size}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}, {GPU_Model, GPU_Brand, GPU_Variant, GPU_VRAM, GPU_Design}

R_2(Model: VARCHAR, Brand: VARCHAR, Display: VARCHAR)

- PK: {Model}
- CK: {Model}

RAM_Placed_On(Model: VARCHAR, DDR: VARCHAR, Capacity: INT, Clock_Speed: INT, Power: INT, Price: NUMBER, Brand: VARCHAR, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, DDR, Capacity, Clock_Speed}
- Candidate keys: {Model, DDR, Capacity, Clock_Speed}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Model → Brand
- DDR, Capacity → Power

Decomposing on Model → Brand gives us
R_1(Model: VARCHAR, DDR: VARCHAR, Capacity: INT, Clock_Speed: INT, Power: INT, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, DDR, Capacity, Clock_Speed}
- CK: {Model, DDR, Capacity, Clock_Speed}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

R_2(Model: VARCHAR, Brand: VARCHAR)

- PK: {Model}
- CK: {Model}

Decomposing on DDR, Capacity → Power gives us
R_3(Model: VARCHAR, DDR: VARCHAR, Capacity: INT, Clock_Speed: INT, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- PK: {Model, DDR, Capacity, Clock_Speed}
- CK: {Model, DDR, Capacity, Clock_Speed}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- FK: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

R_4(DDR: VARCHAR, Capacity: INT, Power: INT)

- PK: {DDR, Capacity}
- CK: {DDR, Capacity}

PowerSupply_In(Model: VARCHAR, Brand: VARCHAR, Efficiency: VARCHAR, Form_Factor: VARCHAR, Modularity: VARCHAR, Wattage: INT, Price: NUMBER, **Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR)

- Primary key: {Model, Form_Factor, Modularity}
- Candidate keys: {Model, Form_Factor, Modularity}, {Case_Model, Case_Colour, Case_Size}
- Foreign keys: {Case_Model, Case_Colour, Case_Size}
- Model → Brand, Wattage, Efficiency

Decomposing Model → Brand, Wattage, Efficiency gives us
R_1(Model: VARCHAR, Brand: VARCHAR, Efficiency: VARCHAR, Wattage: INT)

- PK: {Model}
- CK:{Model}

R_2(Model: VARCHAR, Modularity: VARCHAR, Form_Factor: VARCHAR, Price: NUMBER, **Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR)

- PK: {Model, Modularity, Form_Factor}
- CK: {Model, Modularity, Form_Factor}, {Case_Model, Case_Colour, Case_Size}
- FK: {Case_Model, Case_Colour, Case_Size}

Case_Contains(Model: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Size: VARCHAR, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Colour, Size}
- Candidate keys: {Model, Colour, Size}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Model → Brand

Decomposing on Model → Brand gives us

R_1(Model: VARCHAR, Colour: VARCHAR, Size: VARCHAR, Price: NUMBER, **MB_Model**: VARCHAR, **MB_Form_Factor**: VARCHAR, **MB_Series**: VARCHAR, **MB_WIFI**: VARCHAR)

- Primary key: {Model, Colour, Size}
- Candidate keys: {Model, Colour, Size}, {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}
- Foreign keys: {MB_Model, MB_Form_Factor, MB_Series, MB_WIFI}

R_2(Model: VARCHAR, Brand: VARCHAR)

- PK: {Model}
- CK: {Model}

Mouse(Model: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Size: VARCHAR: Weight: INT, Price: NUMBER, Wired_Wireless: VARCHAR)

- Primary key: {Model, Colour}
- Candidate keys: {Model, Colour}
- Model → Brand

Decomposing on Model → Brand gives us

R_1(Model: VARCHAR, Colour: VARCHAR, Size: VARCHAR: Weight: INT, Price: NUMBER, Wired_Wireless: VARCHAR)

- PK: {Model, Colour}
- CK: {Model, Colour}

R_2(Model: VARCHAR, Brand: VARCHAR)

- PK: {Model}
- CK: {Model}

Plugged_Into(**Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR, **Mouse_Model**: VARCHAR, **Mouse_Colour**: VARCHAR)

- Primary Key: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}
- Candidate Key: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}
- Foreign Key: {Case_Model, Case_Colour, Case_Size}, {Mouse_Model, Mouse_Colour}

This is already in BCNF as it has no functional dependencies

R(**Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR, **Mouse_Model**: VARCHAR, **Mouse_Colour**: VARCHAR)

- PK: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}
- CK: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}
- FK: {Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour}

Keyboard(Model: VARCHAR, Percentage: NUMBER, Brand: VARCHAR, Colour: VARCHAR, Price: NUMBER)

- Primary Key: {Model}
- Candidate Keys: {Model}
- Model ⟶ Brand, Colour, Price, Percentage

This is already in BCNF as Model is a superkey (specifically it is a key)

R(Model: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Price: NUMBER, Percentage: NUMBER)

- PK: {Model}
- CK: {Model}

Switches(**Model**: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Sound: VARCHAR, Price: NUMBER)

- Primary Key: {Model, Brand, Colour, Sound}
- Candidate Keys: {Model, Brand, Colour, Sound}
- Model, Brand, Colour ⟶ Price

Decomposing on Model, Brand, Colour → Price gives us
R_1(**Model**: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Sound: VARCHAR)

- PK: {Model, Brand, Colour, Sound}
- CK: {Model, Brand, Colour, Sound}

R_2(**Model**: VARCHAR, Brand: VARCHAR, Colour: VARCHAR, Price: NUMBER)

- PK: {Model, Brand, Colour}
- CK: {Model, Brand, Colour}

Connected_To(**Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR, **Keyboard_Model**: VARCHAR)

- Primary Key: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}
- Candidate Key: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}
- Foreign Key: {Case_Model, Case_Colour, Case_Size}, {Keyboard_Model}

This is already in BCNF as it has no functional dependencies

R(**Case_Model**: VARCHAR, **Case_Colour**: VARCHAR, **Case_Size**: VARCHAR, **Keyboard_Model**: VARCHAR)

- PK: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}
- CK: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}
- FK: {Case_Model, Case_Colour, Case_Size, Keyboard_Model}

Creating & Populating Tables

Normalized CPUCooler_Cools

```
CREATE TABLE CPUCooler_On_Model(
 Model          VARCHAR              PRIMARY KEY,
 Lighting       VARCHAR,
 Brand          VARCHAR,
 Type           VARCHAR,
 LCD            VARCHAR
);

INSERT INTO CPUCooler_On_Model  VALUES('DeepCool GAMMAXX AG400 BK ARGB', 'RGB', 'Deepcool', 'Air', null);
INSERT INTO CPUCooler_On_Model  VALUES('DeepCool GAMMAXX AG400', null, 'Deepcool', 'Air', null);
INSERT INTO CPUCooler_On_Model  VALUES('DeepCool GAMMAXX AG400 WH ARGB', RGB, 'Deepcool', 'Air',  null);
INSERT INTO CPUCooler_On_Model  VALUES('DeepCool GAMMAXX CT', 'RGB', 'Deepcool', 'Air', null);
INSERT INTO CPUCooler_On_Model  VALUES('Corsair iCUE H150i Elite CAPELLIX XT', 'RGB', 'Corsair', 'Liquid', null);

CREATE TABLE CPUCooler_On(
 Model          VARCHAR,
 Size           INT,
 Price          NUMBER,
 CPU_Model      VARCHAR              UNIQUE,
 PRIMARY KEY (Model, Size),
 FOREIGN KEY(CPU_Model) REFERENCES CPU_On ON DELETE CASCADE
);

INSERT INTO CPUCooler_On  VALUES('DeepCool GAMMAXX AG400 BK ARGB', 120, 39.99, null);
INSERT INTO CPUCooler_On  VALUES('DeepCool GAMMAXX AG400', 120, 34.99, null);
INSERT INTO CPUCooler_On  VALUES('DeepCool GAMMAXX AG400 WH ARGB', 120, 39.99, null);
INSERT INTO CPUCooler_On  VALUES('DeepCool GAMMAXX CT', 120, 49.99, null);
INSERT INTO CPUCooler_On  VALUES('Corsair iCUE H150i Elite CAPELLIX XT', 360, 284.99, null);
```

Normalized CaseFan_Inside

```
CREATE TABLE CaseFan_Inside_Model(
  Model          VARCHAR              PRIMARY KEY,
  Brand          VARCHAR,
  Lighting       VARCHAR
);

INSERT INTO CaseFan_Inside_Model('NF-P12 Redux', 'Noctua', 'none');
INSERT INTO CaseFan_Inside_Model('NF-A20 PWM', 'Noctua', 'none');
INSERT INTO CaseFan_Inside_Model('QL120', 'Corsair', 'RGB');
INSERT INTO CaseFan_Inside_Model('LL120', 'Corsair', 'RGB');
INSERT INTO CaseFan_Inside_Model('P12', 'ARCTIC', 'none');

CREATE TABLE CaseFan_Inside(
  Model          VARCHAR,
  Size           INT,
  Price          NUMBER,
  Colour         VARCHAR,
  Case_Model     VARCHAR,
  Case_Colour    VARCHAR,
  Case_Size      VARCHAR,
  PRIMARY KEY (Model, Size),
  UNIQUE (Case_Model, Case_Colour, Case_Size)
  FOREIGN KEY(Case_Model, Case_Colour, Case_Size) REFERENCES Case_Contains ON DELETE CASCADE
);

INSERT INTO CaseFan_Inside('NF-P12 Redux', 120, 15.95, 'grey', null, null, null)
INSERT INTO CaseFan_Inside('NF-A20 PWM', 200, 36.95, 'black', null, null, null)
INSERT INTO CaseFan_Inside('QL120', 120, 35.99,'black', null, null, null)
INSERT INTO CaseFan_Inside('LL120', 120, 25.99, 'black',  null, null, null)
INSERT INTO CaseFan_Inside('P12', 120, 8.45, 'black', null, null, null)
```

Normalized CPU_On

```
CREATE TABLE CPU_On_Power(
 Cores                  VARCHAR,
 Generation             INT,
 Power                  INT,
);

INSERT INTO CPU_On_Power(6, 5000, 65);
INSERT INTO CPU_On_Power(8, 7000, 120);
INSERT INTO CPU_On_Power(16, 13, 125);
INSERT INTO CPU_On_Power(24, 13, 125);
INSERT INTO CPU_On_Power(6, 7000, 105);

CREATE TABLE CPU_On(
 Model                  VARCHAR         PRIMARY KEY,
 Cores                  VARCHAR,
 Generation             INT,
 Socket                 VARCHAR,
 Price                  Number,
 Brand                  VARCHAR,
 MB_Model               VARCHAR,
 MB_Form_Factor         VARCHAR,
 MB_Series              VARCHAR,
 MB_WIFI                VARCHAR,
FOREIGN KEY(MB_Model, MB_Form_Factor, MB_Series, MB_WIFI) REFERENCES Motherboard_Powers
on DELETE CASCADE
);

INSERT INTO CPU_On('Ryzen 5 5600X', 6, 5000, 'AM4', 160, 'AMD', null, null, null, null);
INSERT INTO CPU_On('Ryzen 7 7800X3D', 8, 7000, 'AM5', 370, 'AMD', null, null, null, null);
INSERT INTO CPU_On('Ryzen 7 5800X', 8, 5000, 'AM4', 160, 'Intel', null, null, null, null);
INSERT INTO CPU_On('Intel Core i7-13700K', 16, 13, 'LGA 1700', 360, 'AMD', null, null, null, null);
INSERT INTO CPU_On('Intel Core i9-13900K', 24, 13, 'LGA 1700', 570, 'AMD', null, null, null, null);
```

Normalized Motherboard_Powers

```
CREATE TABLE Motherboard_Powers_Socket(
  CPU_Model   VARCHAR              PRIMARY KEY,
  Socket      VARCHAR
 );

INSERT INTO Motherboard_Powers_Socket  VALUES('AMD Ryzen 5 5600X 6-core', 'Socket AM4');
INSERT INTO Motherboard_Powers_Socket  VALUES('AMD Ryzen 5 5500 6-core', 'Socket AM4');
INSERT INTO Motherboard_Powers_Socket  VALUES('AMD Ryzen 5 4600G', 'Socket AM4');
INSERT INTO Motherboard_Powers_Socket  VALUES('Intel Core i5-12600K', 'LGA 1700');
INSERT INTO Motherboard_Powers_Socket  VALUES('AMD Ryzen 7 5700X', 'Socket AM4');

CREATE TABLE Motherboard_Powers(
  Model             VARCHAR,
  Form_Factor       VARCHAR,
  Series            VARCHAR,
  WIFI              VARCHAR,
  Brand             VARCHAR,
  Power             INT,
  Expansion_slots   VARCHAR,
  Price             NUMBER,
  Memory_Slots      INT,
  DDR               VARCHAR,
  Storage_Ports     INT,
  CPU_Model         VARCHAR     NOT NULL,
  PSU_Modularity    VARCHAR     NOT NULL,
  PSU_Form_Factor   VARCHAR     NOT NULL,
  PSU_Model         VARCHAR     NOT NULL,
  PRIMARY KEY (Model, Form_Factor, Series, WIFI),
  UNIQUE (CPU_Model),
  UNIQUE (PSU_Modularity, PSU_Form_Factor, PSU_Model),
  FOREIGN KEY(CPU_Model) REFERENCES CPU_On ON DELETE CASCADE,
  FOREIGN KEY(PSU_Modularity, PSU_Form_Factor, PSU_Model) REFERENCES PowerSupply_In ON
DELETE CASCADE
 );
```

```
INSERT INTO Motherboard_Powers  VALUES('Pro Z690-A', 'ATX', 'Pro Z Series', 'No', 'MSI', 50, null,
299.99, 4, 'DDR4', 3, 'Intel Core i5-12600K',  'Full', 'ATX', 'CP-92020200-NA' );
INSERT INTO Motherboard_Powers  VALUES('Prime B660M-A D4', 'mATX', 'Prime B660M-A D4', 'No',
'ASUS', 50, null, 169.99, 2, 'DDR4', 3, 'Intel Core i5-12600K',  'Full', 'ATX', 'CP-92020200-NA' );
INSERT INTO Motherboard_Powers  VALUES('MAG Z690 TOMAHAWK WIFI', 'ATX', 'Tomahawk Series',
'MSI', 80, null, 299.99, 4, 'DDR5', 5, 'Intel Core i5-12600K',  'Full', 'ATX', 'CP-92020200-NA' );
INSERT INTO Motherboard_Powers  VALUES('MSI Z690I UNIFY', 'Mini ITX', 'Z Series', 'MSI', 60, null,
519.99, 5, 'DDR5', 5, 'Intel Core i5-12600K',  'Full', 'ATX', 'CP-92020200-NA' );
INSERT INTO Motherboard_Powers  VALUES('Pro Z690-P', 'ATX', 'Pro Z Series', 'MSI', 50, null, 277.99, 4,
'DDR4', 4, 'Intel Core i5-12600K',  'Full', 'ATX', 'CP-92020200-NA' );
```

Normalized Storage_Requires

```
CREATE TABLE Storage_Requires_RWSpeed(
  Model              VARCHAR          PRIMARY KEY,
  Read_Speeds        INT,
  Write_Speeds       INT,
  Brand              VARCHAR
);

INSERT INTO Storage_Requires_RWSpeed('IronWolf Pro 12', 250, 250, 'Seagate');
INSERT INTO Storage_Requires_RWSpeed('WD Red Plus', 160, 120, 'Western Digital');
INSERT INTO Storage_Requires_RWSpeed('870 EVO SATA III', 560, 530, 'Samsung');
INSERT INTO Storage_Requires_RWSpeed('WD Blue SA510', 560, 540, 'Western Digital');
INSERT INTO Storage_Requires_RWSpeed('WD_Black SN770', 5150, 4850, 'Western Digital');

CREATE TABLE Storage_Requires(
  Model              VARCHAR,
  Capacity           INT,
  Interface          VARCHAR,
  Type               VARCHAR,
  Form_Factor        VARCHAR,
  Power              INT,
  MB_Model           VARCHAR,
  MB_Form_Factor     VARCHAR,
```

```
 MB_Series          VARCHAR,
 MB_WIFI            VARCHAR,
 PRIMARY KEY (Model, Capacity, Interface, Type, Form_Factor),
 FOREIGN KEY (MB_Model, MB_Form_Factor, MB_Series, MB_WIFI) REFERENCES
Motherboard_Powers ON DELETE CASCADE
);

INSERT INTO Storage_Requires('IronWolf Pro 12', 12, 'SATA', 'HDD', '3.5 Inch', 9, null, null, null null);
INSERT INTO Storage_Requires('WD Read Plus', 6, 'SATA', 'HDD', '3.5 Inch', 9, null, null, null null);
INSERT INTO Storage_Requires('870 EVO SATA III', 1, 'SATA', 'SSD', '2.5 Inch', 16, null, null, null null);
INSERT INTO Storage_Requires('WD Blue SA510', 2, 'SATA', 'SSD', '2.5 Inch', 16, null, null, null null);
INSERT INTO Storage_Requires('WD_Black SN770', 2, 'NVMe', 'SSD', 'M.2', 1, null, null, null null);

CREATE TABLE Storage_Requires_Price(
 Model              VARCHAR,
 Capacity           INT,
 Type               VARCHAR,
 Price              NUMBER,
 PRIMARY KEY (Model, Capacity, Type)
);

INSERT INTO Storage_Requires_Price('IronWolf Pro 12', 12, 'HDD', 269.99);
INSERT INTO Storage_Requires_Price('WD Red Plus', 6, 'HDD', 149.99);
INSERT INTO Storage_Requires_Price('870 EVO SATA III', 1, 'SSD', 59.99);
INSERT INTO Storage_Requires_Price('WD Blue SA510', 2, 'SSD', 129.99);
INSERT INTO Storage_Requires_Price('WD_Black SN770', 2, 'SSD', 99.99);
```

Normalized GPU_Has

```
CREATE TABLE GPU_Has_Model(
 Model      VARCHAR          PRIMARY KEY,
 Fans       INT
);

INSERT INTO GPU_Has_Model('RTX 2060', 3);
INSERT INTO GPU_Has_Model('RTX 2060, 2);
```

```
INSERT INTO GPU_Has_Model('GTX 1050', 3);
INSERT INTO GPU_Has_Model('GTX 980', 2);
INSERT INTO GPU_Has_Model('RX 6800', 3);
INSERT INTO GPU_Has_Model('GTX 1050', 2);

CREATE TABLE GPU_Has_Price(
 Model          VARCHAR,
 Brand          VARCHAR,
 Variant        VARCHAR,
 VRAM           INT,
 Design         VARCHAR,
 Fans           INT,
 Series         VARCHAR,
 Price          NUMBER,
PRIMARY KEY(Model, Brand, Variant, VRAM, Design)
);

INSERT INTO GPU_Has_Price('RTX 2060', 'MSI', 'OC', 6, 'Black', 3, 'MSI GeForce RTX 2060',  650);
INSERT INTO GPU_Has_Price('RTX 2060', 'ASUS', 'OC', 12, 'Black', 3, 'Dual NVIDIA GeForce RTX 2060',
500);
INSERT INTO GPU_Has_Price('GTX 1050', 'ASUS', 'TI', 4, 'Black', 1, 'ASUS PH-GTX 1050',  230);
INSERT INTO GPU_Has_Price('GTX 1050', 'MSI', 'OC TI', 4, 'Black', 2, 'MSI GTX 1050',  450);
INSERT INTO GPU_Has_Price('GTX 1050', 'MSI', 'TI', 4, Red, 2, 'MSI GTX 1050',  380);

CREATE TABLE GPU_Has_Power(
 Model          VARCHAR
 VRAM           INT
 Fans           INT
 Power          INT,
 PRIMARY KEY(Model, VRAM)
);

INSERT INTO GPU_Has_Power('RTX 3060', 8, 3, 170);
INSERT INTO GPU_Has_Power('GTX 1060', 6, 3, 100);
INSERT INTO GPU_Has_Power('RTX 2060', 16, 3, 190);
INSERT INTO GPU_Has_Power('RTX 4060', 16, 3, 160);
INSERT INTO GPU_Has_Power('GTX 1050', 8, 3, 75);
```

```
CREATE TABLE GPU_Has(
 Model                VARCHAR,
 VRAM:                INT,
 Fans:                INT,
 Brand:               VARCHAR,
 Variant:             VARCHAR,
 Design:              VARCHAR,
 Series:              VARCHAR,
 MB_Model:            VARCHAR,
 MB_Form_Factor:      VARCHAR,
 MB_Series:           VARCHAR,
 MB_WIFI:             VARCHAR,
 PRIMARY KEY(Model, VRAM, Brand, Variant, Design)
 FOREIGN KEY(MB_Model, MB_Form_Factor, MB_Series, MB_WIFI) REFERENCES Motherboard_Powers
ON DELETE CASCADE
);

INSERT INTO GPU_Has('RTX 3060', 16, 3, 'ASUS', 'Ti', 'Black',  'ASUS TUF GeForce RTX 3060' null, null,
null, null);
INSERT INTO GPU_Has('RTX 3050', 16, 3, 'ASUS', 'OC Ti', 'Black',  'ASUS TUF GeForce RTX 3050' null, null,
null, null);
INSERT INTO GPU_Has('RTX 4060', 16, 3, 'MSI', 'Ti', 'Black',  'MSI GeForce RTX 4060' null, null, null, null);
INSERT INTO GPU_Has('GTX 1050', 16, 3, 'MSI', 'Ti', 'Black',  'MSI GeForce RTX 1050' null, null, null, null);
INSERT INTO GPU_Has('GTX 1050', 8, 2, 'MSI', 'OC', 'Black',  'MSI GTX 1050' null, null, null, null);
```

Normalized Monitor_Attached_To

```
CREATE TABLE Monitor_Attached_To_Model(
  Model        VARCHAR              PRIMARY KEY,
  Brand        VARCHAR,
  Display      VARCHAR
 );

INSERT INTO Monitor_Attached_To_Model  VALUES('LG 24MP40A-C Full HD LCD', 'LG', 'LCD');
INSERT INTO Monitor_Attached_To_Model  VALUES('Samsung LS24R350FZNXZA LED-Lit Monitor',
'Samsung', 'LED');
```

```
INSERT INTO Monitor_Attached_To_Model  VALUES('Samsung LS24T350FHNXZA LED-Lit Monitor',
'Samsung', 'LED');
INSERT INTO Monitor_Attached_To_Model  VALUES('Alienware AW3423DWF QD-OLED LED',
'Alienware', 'OLED');
INSERT INTO Monitor_Attached_To_Model  VALUES('Samsung LS32B200NWNXGO LED Monitor',
'Samsung', 'LED');

CREATE TABLE Monitor_Attached_To(
  Model              VARCHAR,
  Curvature          VARCHAR,
  Resolution         VARCHAR,
  Refresh_Rate       INT,
  Size               NUMBER,
  Response_Time      NUMBER,
  Price              NUMBER,
  GPU_Model          VARCHAR,
  GPU_Brand          VARCHAR,
  GPU_Variant        VARCHAR,
  GPU_VRAM           INT,
  GPU_Design         VARCHAR,
  GPU_Brand          VARCHAR,
  GPU_Variant        VARCHAR,
  MB_Model           VARCHAR,
  MB_Form_Factor     VARCHAR,
  MB_Series          VARCHAR,
  MB_WIFI            VARCHAR,
  PRIMARY KEY (Model, Curvature, Resolution, Refresh_Rate, Size),
  FOREIGN KEY(MB_Model, MB_Form_Factor, MB_Series, MB_WIF) REFERENCES Motherboard_Powers
ON DELETE CASCADE,
  FOREIGN KEY(GPU_Model, GPU_Brand, GPU_Variant, GPU_VRAM, GPU_Design) REFERENCES
GPU_Has ON DELETE CASCADE
 );

INSERT INTO Monitor_Attached_To  VALUES('Alienware AW3423DWF QD-OLED LED', '1800R', '3440 x
1440', 165, 34, 0.1, 999.99, null, null, null, null, null, null, null, null, null, null);
INSERT INTO Monitor_Attached_To  VALUES('Samsung LS32B200NWNXGO LED Monitor', '4000R', '1920
x 1080', 75, 32, 8, 199.99, null, null, null, null, null, null, null, null, null, null);
```

```
INSERT INTO Monitor_Attached_To  VALUES('Samsung LS24T350FHNXZA LED-Lit Monitor', '4000R', '1920 x 1080', 75, 24, 5, 128.00, null, null, null, null, null, null, null, null, null, null, null);
INSERT INTO Monitor_Attached_To  VALUES('Samsung LS24R350FZNXZA LED-Lit Monitor', '4000R', '1920 x 1080', 75, 24, 5, 128.00, null, null, null, null, null, null, null, null, null, null, null);
INSERT INTO Monitor_Attached_To  VALUES('LG 24MP40A-C Full HD LCD', '4000R', '1920 x 1080', 75, 23.8, 5, 118.00, null, null, null, null, null, null, null, null, null, null, null);
```

Normalized RAM_Placed_On

```
CREATE TABLE RAM_Placed_On_Brand(
  Model          VARCHAR        PRIMARY KEY,
  Brand          VARCHAR
);

INSERT INTO RAM_Placed_On_Brand('VENGEANCE LPX', 'Corsair');
INSERT INTO RAM_Placed_On_Brand('VENGEANCE RGB PRO SL', 'Corsair');
INSERT INTO RAM_Placed_On_Brand('VENGEANCE LPX PRO', 'Corsair');
INSERT INTO RAM_Placed_On_Brand('VENGEANCE DDR5', 'Corsair');
INSERT INTO RAM_Placed_On_Brand('VENGEANCE SODIMM', 'Corsair');

CREATE TABLE RAM_Placed_On(
  Model                VARCHAR,
  DDR                  VARCHAR,
  Capacity             INT,
  Clock_Speed          INT,
  Price                NUMBER,
  MB_Model             VARCHAR,
  MB_Form_Factor       VARCHAR,
  MB_Series            VARCHAR,
  MB_WIFI              VARCHAR,
  PRIMARY KEY (Model, DDR, Capacity, Clock_Speed),
  FOREIGN KEY (MB_Model, MB_Form_Factor, MB_Series, MB_WIF) REFERENCES Motherboard_Powers
ON DELETE CASCADE
);
```

```
INSERT INTO RAM_Placed_On('VENGEANCE LPX', 'DDR4', 32, 3200, 114.00, null, null, null, null);
INSERT INTO RAM_Placed_On('VENGEANCE RGB PRO SL', 'DDR4', 32, 3600, 134.99, null, null, null, null);
INSERT INTO RAM_Placed_On('VENGEANCE RGB PRO', 'DDR4', 32, 3600, 79.99, null, null, null, null);
INSERT INTO RAM_Placed_On('VENGEANCE DDR5', 'DDR5', 32, 5600, 94.66, null, null, null, null);
INSERT INTO RAM_Placed_On('VENGEANCE SODIMM', 'DDR5', 32, 4800, 94.99, null, null, null, null);

CREATE TABLE RAM_Placed_On_Power(
  DDR                VARCHAR,
  Capacity           INT,
  Power              INT
  PRIMARY KEY (DDR, Capacity)
);

INSERT INTO RAM_Placed_On_Power('DDR4', 32, 12);
INSERT INTO RAM_Placed_On_Power('DDR4', 32, 12);
INSERT INTO RAM_Placed_On_Power('DDR4', 32, 12);
INSERT INTO RAM_Placed_On_Power('DDR5', 32, 16);
INSERT INTO RAM_Placed_On_Power('DDR5', 32, 16);
```

Normalized PowerSupply_In

```
CREATE TABLE PowerSupply_In_Model(
 Model              VARCHAR,
 Brand              VARCHAR,
 Efficiency         VARCHAR,
 Wattage            INT,
PRIMARY KEY(Model, Form_Factor)
);

INSERT INTO PowerSupply_In_Model('RM750e', 'CORSAIR', '80+ Gold', 750);
INSERT INTO PowerSupply_In_Model('RM1000x', 'CORSAIR', '80+ Gold', 1000);
INSERT INTO PowerSupply_In_Model('Toughpower GX2', 'Thermaltake', '80+ Gold', 600);
INSERT INTO PowerSupply_In_Model('SF750', 'CORSAIR', '80+ Platinum', 1200);
INSERT INTO PowerSupply_In_Model('CX650M', 'CORSAIR', '80+ Bronze', 650);
```

```
CREATE TABLE PowerSupply_in(
 Model              VARCHAR,
 Modularity         VARCHAR,
 Form_Factor        VARCHAR,
 Price              NUMBER,
 Case_Model         VARCHAR,
 Case_Colour        VARCHAR,
 Case_Size          VARCHAR,
PRIMARY KEY(Model, Modularity, Form_Factor),
FOREIGN KEY(Case_Model, Case_Colour, Case_Size) REFERENCES Case_Contains ON DELETE CASCADE
);

INSERT INTO PowerSupply_In('RM750e', Full, 'ATX', 100);
INSERT INTO PowerSupply_In('RM1000x', Full, 'ATX', 170);
INSERT INTO PowerSupply_In('ToughPower GX2', None, 'ATX', 65);
INSERT INTO PowerSupply_In('CX650M', Semi, 'ATX', 70);
INSERT INTO PowerSupply_In('RM1000e', Full, 'ATX', 160);
```

Normalized Case_Contains

```
CREATE TABLE Case_Contains(
  Model             VARCHAR,
  Colour            VARCHAR,
  Size              VARCHAR,
  Price             NUMBER,
  MB_Model          VARCHAR,
  MB_Form_Factor    VARCHAR,
  MB_Series         VARCHAR,
  MB_WIFI           VARCHAR,
  PRIMARY KEY (Model, Colour, Size),
  UNIQUE (MB_Model, MB_Form_Factor, MB_Series, MB_WIFI),
  FOREIGN KEY(MB_Model, MB_Form_Factor, MB_Series, MB_WIFI) REFERENCES Motherboard_Powers
ON DELETE CASCADE,
 );
```

```
INSERT INTO Case_Contains  VALUES('Corsair 4000D CC-9011200-WW', 'Black', 'Mid-Tower', 119.98, null, null, null, null);
INSERT INTO Case_Contains  VALUES('Corsair 4000D CC-9011201-WW', 'White', 'Mid-Tower', 119.99, null, null, null, null);
INSERT INTO Case_Contains  VALUES('Corsair iCUE 4000X RGB CC-9011204-WW', 'Black', 'Mid-Tower', 159.99, null, null, null, null);
INSERT INTO Case_Contains  VALUES('Corsair iCUE 4000X RGB CC-9011205-WW', 'White', 'Mid-Tower', 189.99, null, null, null, null);
INSERT INTO Case_Contains  VALUES('Corsair 5000D CC-9011210-WW', 'Black', 'Mid-Tower', 204.99, null, null, null, null);

CREATE TABLE Case_Contains_Brand(
  Model          VARCHAR                PRIMARY KEY,
  Brand          VARCHAR
 );

INSERT INTO Case_Contains_Brand  VALUES('Corsair 4000D CC-9011200-WW', 'Corsair');
INSERT INTO Case_Contains_Brand  VALUES('Corsair 4000D CC-9011201-WW', 'Corsair');
INSERT INTO Case_Contains_Brand  VALUES('Corsair iCUE 4000X RGB CC-9011204-WW', 'Corsair');
INSERT INTO Case_Contains_Brand  VALUES('Corsair iCUE 4000X RGB CC-9011205-WW', 'Corsair');
INSERT INTO Case_Contains_Brand  VALUES('Corsair 5000D CC-9011210-WW', 'Corsair');
```

Normalized Mouse

```
CREATE TABLE Mouse(
  Model                VARCHAR,
  Colour               VARCHAR,
  Size                 VARCHAR,
  Weight               INT,
  Price                NUMBER,
  Wired_Wireless       VARCHAR,
  PRIMARY KEY (Model, Colour)
);

INSERT INTO Mouse('G305 LIGHTSPEED', 'Black', 'Standard', 99, 49.99, 'Wireless');
```

```
INSERT INTO Mouse(‘G502 HERO, ‘Black’, ‘Standard’, 89, 79.99, ‘Wired’);
INSERT INTO Mouse(‘Basilisk V3’, ‘Black’, ‘Ergonomic’, 128, 69.99, ‘Wired’);
INSERT INTO Mouse(‘DeathAdder’, ‘Black’, ‘Standard’, 96, 29.99, ‘Wired’);
INSERT INTO Mouse(‘DeathAdder V2 Pro’, ‘Black’, ‘Standard’, 88, 129.99, ‘Wireless’);

CREATE TABLE Mouse_Brand(
  Model          VARCHAR       PRIMARY KEY,
  Brand          VARCHAR
);

INSERT INTO Mouse_Brand(‘G305 LIGHTSPEED’, ‘Logitech’);
INSERT INTO Mouse_Brand(‘G502 HERO’, ‘Logitech’);
INSERT INTO Mouse_Brand(‘Basilisk V3’, ‘Razer’);
INSERT INTO Mouse_Brand(‘DeathAdder’, ‘Razer’);
INSERT INTO Mouse_Brand(‘DeathAdder V2 Pro’, ‘Razer’);
```

Normalized Plugged_Into

```
CREATE TABLE Plugged_Into(
 Case_Model                VARCHAR
 Case_Colour               VARCHAR
 Case_Size                 VARCHAR
 Mouse_Model               VARCHAR
 Mouse_Colour              VARCHAR
 PRIMARY KEY(Case_Model, Case_Colour, Case_Size, Mouse_Model, Mouse_Colour)
 FOREIGN KEY(Case_Model, Case_Colour, Case_Size) REFERENCES Case_Contains ON DELETE CASCADE,
 FOREIGN KEY(Mouse_Model, Mouse_Colour) REFERENCES Mouse ON DELETE CASCADE,
);

INSERT INTO Plugged_Into(‘Corsair 4000D CC-9011200-WW', ‘Black’, ‘Mid-Tower’, ‘G305 LIGHTSPEED’,
‘Black’);
INSERT INTO Plugged_Into(‘Corsair 4000D CC-9011201-WW', ‘White’, ‘Mid-Tower’, ‘G502 HERO’,
‘Black’);
INSERT INTO Plugged_Into(‘Corsair iCUE 4000X RGB CC-9011204-WW', ‘Black’, ‘Mid-Tower’, ‘Basilisk
V3’, ‘Black’);
```

INSERT INTO Plugged_Into(‘Corsair iCUE 4000X RGB CC-9011205-WW', ‘White’, ‘Mid-Tower’, ‘DeathAdder’, ‘Black’);
INSERT INTO Plugged_Into(‘Corsair 5000D CC-9011210-WW', ‘Black’, ‘Mid-Tower’, ‘DeathAdder V2 Pro’, ‘Black’);

Normalized Keyboard

```
CREATE TABLE Keyboard(
  Model          VARCHAR              PRIMARY KEY,
  Brand          VARCHAR,
  Colour         VARCHAR,
  Price          NUMBER,
  Percentage     NUMBER
);

INSERT INTO Keyboard_Brand  VALUES(‘Corsair K55 Pro Lite', ‘Corsair’, ‘Black’, 100.00, 74.95);
INSERT INTO Keyboard_Brand  VALUES(‘Corsair K70 Pro Mini Wireless', ‘Corsair’, ‘Black’, 60.00, 259.99);
INSERT INTO Keyboard_Brand  VALUES(‘Corsair K100 Air Wireless RGB Ultra-Thin', ‘Corsair’, ‘Black’, 100.00, 399.99);
INSERT INTO Keyboard_Brand  VALUES(‘Corsair K100 RGB Optical-Mechanical', ‘Corsair’, ‘Black’, 100.00, 349.99);
INSERT INTO Keyboard_Brand  VALUES(‘Corsair K70 Core RGB Mechanical', ‘Corsair’, ‘Black’, 100.00, 149.99);
```

Normalized Switches

```
CREATE TABLE Switches(
  Model          VARCHAR,
  Brand          VARCHAR,
  Colour         VARCHAR,
  Sound          VARCHAR,
  PRIMARY KEY (Model, Brand, Colour, Sound)
);
```

```
INSERT INTO Switches(‘Oil King Axis Pre Lubricated’, ‘Gateron’, ‘Black’, ‘Linear’);
INSERT INTO Switches(‘Panda Switch Lubed’, ‘Glorious Gaming’, ‘Orange’, ‘Tactile’);
INSERT INTO Switches(‘Panda Switch Unlubed’, ‘Glorious Gaming’, ‘Orange’, ‘Tactile’);
INSERT INTO Switches(‘Wisteria 39gf Linear, ‘EPOMAKER’, ‘Wisteria’, ‘Linear’);
INSERT INTO Switches(‘Dawn Pink 38gf Linear, ‘EPOMAKER, ‘Dawn Pink’, ‘Linear’);

CREATE TABLE Switches_Price(
  Model         VARCHAR,
  Brand         VARCHAR,
  Colour        VARCHAR,
  Price         NUMBER,
  PRIMARY KEY (Model, Brand, Colour)
);

INSERT INTO Switches_Price(‘Oil King Axis Pre Lubricated’, ‘Gateron’, ‘Black’, 28.90);
INSERT INTO Switches_Price(‘Panda Switch Lubed’, ‘Glorious Gaming’, ‘Orange’, 54.99);
INSERT INTO Switches_Price(‘Panda Switch Unlubed’, ‘Glorious Gaming’, ‘Orange’, 24.99);
INSERT INTO Switches_Price(‘Wisteria 39gf Linear, ‘EPOMAKER’, ‘Wisteria’, 11.99);
INSERT INTO Switches_Price(‘Dawn Pink 38gf Linear, ‘EPOMAKER, ‘Dawn Pink’, 11.99);
```

Normalized Connected_To

```
CREATE TABLE Connected_To(
 Case_Model                VARCHAR,
 Case_Colour               VARCHAR,
 Case_Size                 VARCHAR,
 Keyboard_Model            VARCHAR,
PRIMARY KEY(Case_Model, Case_Colour, Case_Size, Keyboard_Model),
 FOREIGN KEY(Case_Model, Case_Colour, Case_Size) REFERENCES Case_Contains ON DELETE CASCADE,
 FOREIGN KEY(Keyboard_Model) REFERENCES Keyboard ON DELETE CASCADE
);

INSERT INTO Connected_To(‘Corsair 4000D CC-9011200-WW', ‘Black’, ‘Mid-Tower’, ‘Corsair K55 Pro
Lite');
```

INSERT INTO Connected_To(‘Corsair 4000D CC-9011201-WW', ‘White’, ‘Mid-Tower’, ‘Corsair K70 Pro Mini Wireless');
INSERT INTO Connected_To(‘Corsair iCUE 4000X RGB CC-9011204-WW', ‘Black’, ‘Mid-Tower’, ‘Corsair K100 Air Wireless RGB Ultra-Thin');
INSERT INTO Connected_To(‘Corsair iCUE 4000X RGB CC-9011205-WW', ‘White’, ‘Mid-Tower’, ‘Corsair K100 RGB Optical-Mechanical');
INSERT INTO Connected_To(‘Corsair 5000D CC-9011210-WW', ‘Black’, ‘Mid-Tower’, ‘Corsair K70 Core RGB Mechanical');