BUILD-A-COMPUTER WORKSHOP

IST 659: PROJECT FINAL REPORT

JAMES EAKINS

Professor Yang Wang

SYRACUSE UNIVERSITY

CONTENTS

PROJECT SUMMARY ------------------------------------------------------------------3

ENTITY AND ATTRIBUTE TABLE ----------------------------------------------------4

ENTITY-RELATIONSHIP DIAGRAM ----------------------------------------------- 10

CREATION OF TABLES ----------------------------------------------------------------7

MAJOR DATA QUESTIONS ---------------------------------------------------------21

INTERFACES ---------------------------------------------------------------------------26

REPORT --------------------------------------------------------------------------------28

SECTION I: Project Summary

This project focuses on designing a database for computer enthusiasts who want to build their own computers but unsure of what they need to accomplish the task. There are at least 8 basic compartments that are required to build a computer and 2 categories of compartment factors to consider before purchasing core parts. There are many options out there to build a perfect computer, but always the budget and professional knowledge needed If someone wants to meet certain criteria of software such as games or video editing program requirements.

The data that will be collected for this system is the item records of 8 components from vendors and manufacturers, and the details of each compartment to give more information to users about the compartments. Moreover, the details will include the form factors and socket types to make sure the pieces go with another, or otherwise, a build can fail. The system will have the website information along with the component information to help the users to directly connect to the vendors where they can buy the new components.

The system will have most of the products that are out there stored in the database to serve the purpose of the system, but it will be available to users to input more items that are newly leased from the manufacturers. When the user input new product to the system, they will require to put details like what type of compartment it is, who manufactured it, the form factor if applicable, the socket type if applicable, and more details if available. This will be an interactive system between users to help each other, but at the same time, if they want to present their own build or suggest any unique build, they can enter it separately to help other users.

The flow of the data will be somewhat slow, and the amount of new data will be entered in limited since there will be the only handful of new compartments and accessories will be released every month or year from the manufacturer. The compartments will have different form factors and socket types that work with other parts, so it is important to have those two as required on the most of compartment to prevent any unfortunate incidents where users buy the compartments that don’t work with each other, then it will upset the users.

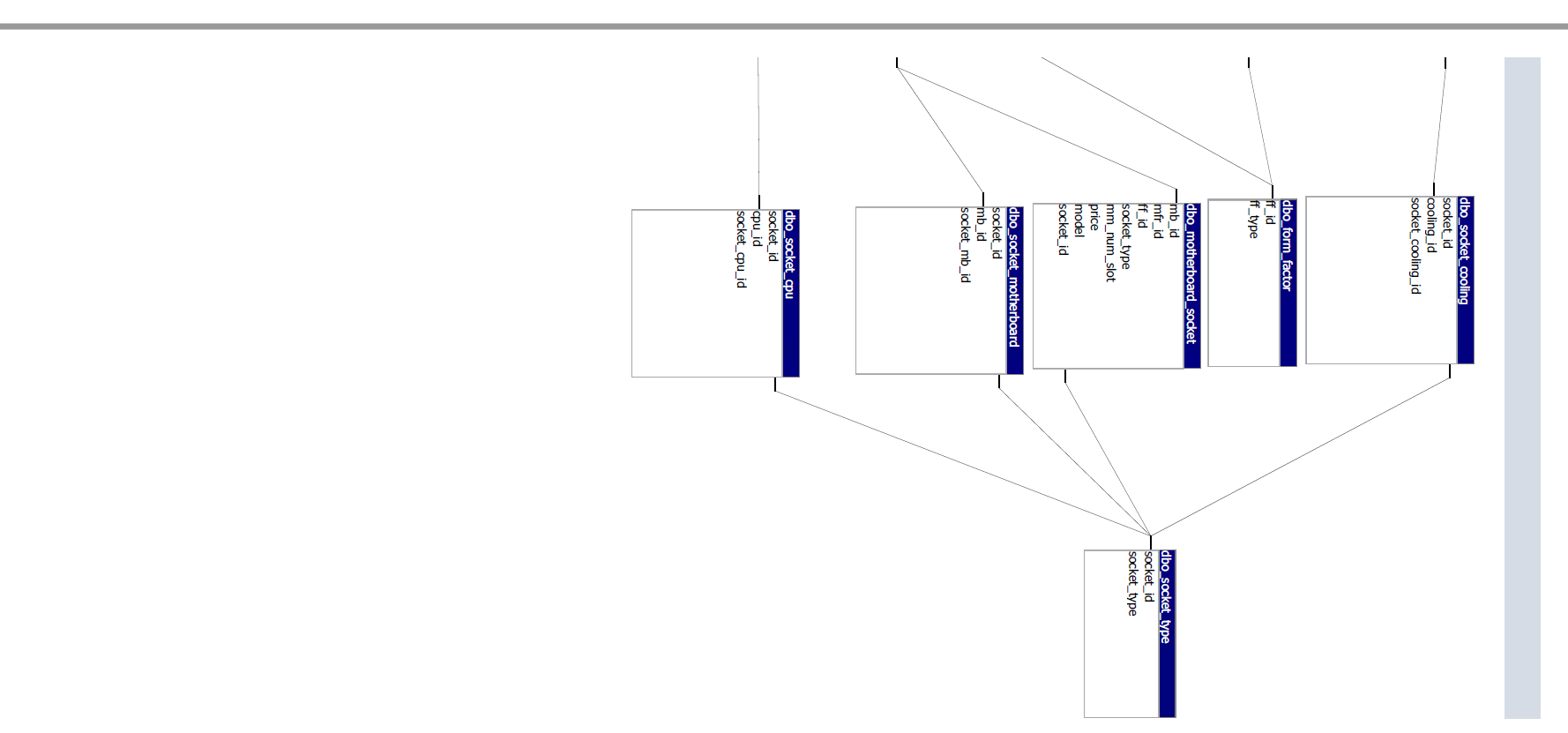
The users will have access to these data through the Microsoft Access where they can view and use the tool to add more data to the system.

SECTION II: Entities and Attributes:

|  |  |
| --- | --- |
| Data Object | Description |
| **1.CPU** | One of the compartments of the computer build. Central processing unit. |
| A.cpu\_id | Primary key/unique identifier of the CPU |
| B.mfr\_id | Foreign key from Manufacturer ID |
| C.numb\_core | Number of cores from the CPU |
| D.int\_graphic | Integrated graphic of the CPU |
| E.hyper\_threading | Hyper threading capability |
| F.cpu\_name | Name of the cpu |
| G.num\_thread | Number of thread |
| H.price | Price of the CPU Unit |
| I.clock\_speed | Clock speed of the CPU |
| J.cpu\_model | CPU Model from the manufacture |
| K.cpu\_series | CPU model number from the manufacture |
| L.cpu\_series\_letter | The letter comes with the model number |
| **2.Mother\_Board** | Mother board is another compartment of the computer. |
| A.mb\_id | Primary key/unique identifier of the mother\_board |
| B.mfr\_id | Foreign key from Manufacturer ID |
| C.ff\_id | Foreign key from Form Factor ID |
| D.socket\_type | Socket type for the CPU |
| E.price | Motherboard price |
| F.mm\_num\_slot | Number of memory slot |
| G.model | Name of the motherboard model |
| **3.Memory** | Memory is another compartment of the computer |
| A.mm\_id | Primary/unique identifier of the memory |
| B.mfr\_id | Foreign key from Manufacturer ID |
| C.size | Size of the memory which count in Gigabytes |
| D.Speed | Speed of the memory |
| E.price | Price of the memory |
| F.model | Name of the model of the memory |
| **4.Storage** | Storage is another compartment of the computer |
| A.storage\_id | Primary/unique identifier of the storage |
| B.mfr\_id | Foreign key from Manufacturer ID |
| C.type | Identifies which type of storage it is |
| D.size | Identifies the size of the storage |
| **5.video\_card** | Video card is another compartment of the computer |
| A.videocard\_id | Primary/unique identifier of video card |
| B.mfr\_id | Foreign key from Manufacturer ID |
| C.mm\_size | Memory size of the video card |
| D.mm\_type | Memory type of the video card |
| E.numb\_cool | Number of cooling fan on Video Card |
| F.interface | Type of interface that video card uses |
| G.max\_length | Length of the card |
| H.web\_id | Foreign key from Website ID |
| **6.power\_supply** | A power supply is another compartment of the computer |
| A.ps\_id | Primary/Unique identifier of power\_supply |
| B.max\_pow | Indicates how much power it can handle |
| C.energy\_eff | An indicator of whether it is energy efficient or not |
| D.mfr\_id | Foreign key from Manufacturer ID |
| E.ff\_id | Foreign key from Form Factor ID |
| F.web\_id | Foreign key from Website ID |
| **7.computer\_case** | A computer case is another compartment of the computer |
| A.case\_id | Primary/Unique identifier of a computer case |
| B.mfr\_id | Foreign key from Manufacturer ID |
| C.ff\_id | Foreign key from Form Factor ID |
| D.web\_id | Foreign key from Website ID |
| **8.cooling** | Cooling is another compartment of the computer |
| A.cooling\_id | Primary/Unique identifier of the cooling |
| B.ff\_id | Foreign key from Form Factor ID |
| C.mfr\_id | Foreign key from Manufacturer ID |
| D.fan\_size | An indicator of the size of fans |
| E.web\_id | Foreign key from Website ID |
| **9.form\_factor** | Form Factor is a unique distinguisher of the compartments to make sure they fit together |
| A.ff\_id | Primary/Unique identifier of the form factor |
| B.type | A type of form factor is identified by this attribute. Each type is standardized. |
| **10.manufacturer** | List of the all manufacturer |
| A.mfr\_id | Primary/Unique identifier of the manufacturer |
| B.name | Name of the manufacturer |
| **12.games** | List of all games |
| A.game\_id | Primary/Unique identifier of games |
| B.names | Name of all the games |
| **13.vid\_edit\_sw** | List of all video editing software |
| A.vid\_sw\_id | Primary/Unique identifier of video editing software |
| B.name | Name of all the video editing software |
| **14.other\_sw** | List of all other software |
| A.other\_sw\_id | Primary/Unique identifier of other software |
| C.name | Name of all the other software |
| **16.customer** | List of customers |
| A.user\_id | Primary/Unique key of the customer |
| B.first\_name | First name |
| C.last\_name | Last name |
| D.phone | Phone number |
| E.address\_1 | Address line 1 |
| F.address\_2 | Address line 2 |
| G.city | City |
| H.state | State |
| I.zip | Zip |
| J.country | Country |
| **17.Build\_customer** | A connection between build and customer |
| A.user\_id | Primary/Foreign key from Customer table |
| B.build\_id | Primary/Foreign key from Build table |
| A.build\_id | Primary/Unique identifier of the manufacturer |
| B.game\_id | Foreign key from Game table, not required |
| C.vid\_sw\_id | Foreign key from Video software table, not required |
| D.other\_sw\_id | Foreign key from other\_software table, not required |
| E.cpu\_id | Foreign key from CPU table, required |
| F.mb\_id | Foreign key from Motherboard table, required |
| G.mm\_id | Foreign key from Memory table, required |
| H.video\_card\_id | Foreign key from Video Card table, required |
| I.ps\_id | Foreign key from Power Supply table, required |
| J.storage\_id | Foreign key from Storage table, required |

SECTION III: ENTITY-RELATIONSHIP DIAGRAM

A close up of a map

Description generated with high confidence

SECTION IV: CREATION OF TABLES

CREATE TABLE manufacture(

mfr\_id INTEGER NOT NULL PRIMARY KEY,

name VARCHAR(255) NOT NULL

)

CREATE TABLE form\_factor(

ff\_id INTEGER NOT NULL PRIMARY KEY,

ff\_type VARCHAR (30) NOT NULL

)

CREATE TABLE cpu (

cpu\_id INTEGER NOT NULL PRIMARY KEY,

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

numb\_core INTEGER NOT NULL,

int\_graphic TEXT NULL,

hyper\_threading BIT NOT NULL

)

ALTER TABLE cpu ADD socket\_type VARCHAR(50) NOT NULL;

ALTER TABLE cpu ADD cpu\_name VARCHAR(50) NULL;

ALTER TABLE cpu ADD numb\_thread INTEGER NULL;

ALTER TABLE cpu ADD price VARCHAR(30) NULL;

ALTER TABLE cpu ADD clock\_speed INTEGER NULL;

ALTER TABLE cpu ALTER COLUMN clock\_speed FLOAT;

CREATE TABLE mother\_board(

mb\_id INTEGER NOT NULL PRIMARY KEY,

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

ff\_id INTEGER NOT NULL FOREIGN KEY REFERENCES form\_factor (ff\_id),

socket\_type VARCHAR (50) NOT NULL,

mm\_std VARCHAR (100) NOT NULL,

mm\_num\_slot INTEGER NOT NULL

)

ALTER TABLE mother\_board ADD price VARCHAR(30) NULL;

ALTER TABLE mother\_board ALTER COLUMN mm\_std VARCHAR (100) NULL;

ALTER TABLE mother\_board DROP COLUMN mm\_std;

ALTER TABLE mother\_board ADD model VARCHAR (255) NULL;

CREATE TABLE memory(

mm\_id INTEGER NOT NULL PRIMARY KEY,

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

size INTEGER NOT NULL,

speed INTEGER NOT NULL

)

ALTER TABLE memory ADD price VARCHAR(30) NOT NULL;

ALTER TABLE memory ADD model VARCHAR(255) NULL;

CREATE TABLE storage(

storage\_id INTEGER NOT NULL PRIMARY KEY,

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

storage\_type INTEGER NOT NULL

)

CREATE TABLE video\_card(

video\_card\_id INTEGER NOT NULL PRIMARY KEY,

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

ff\_id INTEGER NOT NULL FOREIGN KEY REFERENCES form\_factor (ff\_id),

mm\_size INTEGER NOT NULL,

mm\_type VARCHAR(30) NOT NULL,

numb\_cool INTEGER NOT NULL,

interface VARCHAR(50) NOT NULL,

max\_length INTEGER NOT NULL,

vr\_ready BIT NOT NULL,

core\_clock INTEGER NOT NULL

)

ALTER TABLE video\_card ADD price VARCHAR(30) not null;

ALTER TABLE video\_card ALTER COLUMN price FLOAT;

ALTER TABLE video\_card ADD model VARCHAR(255) not null;

ALTER TABLE video\_card DROP COLUMN vr\_ready;

ALTER TABLE video\_card ADD series INTEGER;

ALTER TABLE video\_card ADD chipset\_mfr VARCHAR(50);

ALTER TABLE video\_card ADD series\_name VARCHAR(50);

CREATE TABLE power\_supply(

ps\_id INTEGER NOT NULL PRIMARY KEY,

max\_pow INTEGER NOT NULL,

energy\_eff BIT NOT NULL,

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

ff\_id INTEGER NOT NULL FOREIGN KEY REFERENCES form\_factor (ff\_id)

)

CREATE TABLE computer\_case(

case\_id INTEGER NOT NULL PRIMARY KEY,

ff\_id INTEGER NOT NULL FOREIGN KEY REFERENCES form\_factor(ff\_id),

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture(mfr\_id),

price INTEGER NOT NULL

)

ALTER TABLE computer\_case ADD model VARCHAR(255);

CREATE TABLE cooling (

cooling\_id INTEGER NOT NULL PRIMARY KEY,

ff\_id INTEGER NOT NULL FOREIGN KEY REFERENCES form\_factor (ff\_id),

mfr\_id INTEGER NOT NULL FOREIGN KEY REFERENCES manufacture (mfr\_id),

fan\_size INTEGER NOT NULL

)

CREATE TABLE games (

game\_id INTEGER NOT NULL PRIMARY KEY,

name VARCHAR (255) NOT NULL

)

CREATE TABLE other\_sw(

other\_sw\_id INTEGER NOT NULL PRIMARY KEY,

name VARCHAR (255) NOT NULL

)

CREATE TABLE vid\_edit\_sw(

vid\_edit\_sw\_id INTEGER NOT NULL PRIMARY KEY,

name VARCHAR (255) NOT NULL

)

CREATE TABLE customers(

customer\_id INTEGER NOT NULL PRIMARY KEY,

first\_name VARCHAR (255) NOT NULL,

last\_name VARCHAR (255) NOT NULL,

phone VARCHAR (50) NULL,

address\_1 VARCHAR (255) NOT NULL,

address\_2 VARCHAR (255) NOT NULL,

city VARCHAR (50) NOT NULL,

states VARCHAR (50) NOT NULL,

country VARCHAR (50) NOT NULL,

build\_id INTEGER NULL FOREIGN KEY REFERENCES build (build\_id)

)

ALTER TABLE customers ADD customer\_id INT IDENTITY(1,1) PRIMARY KEY

CREATE TABLE build\_customer(

customer\_id INTEGER NOT NULL PRIMARY KEY,

)

ALTER TABLE build\_customer ADD build\_id INT IDENTITY(1,1) PRIMARY KEY

ALTER TABLE build\_customer ADD cpu\_id INT FOREIGN KEY REFERENCES cpu(cpu\_id)

ALTER TABLE build\_customer ADD mb\_id INT FOREIGN KEY REFERENCES mother\_board(mb\_id)

ALTER TABLE build\_customer ADD mm\_id INT FOREIGN KEY REFERENCES memory(mm\_id)

ALTER TABLE build\_customer ADD cooling\_id INT FOREIGN KEY REFERENCES cooling(cooling\_id)

ALTER TABLE build\_customer ADD ps\_id INT FOREIGN KEY REFERENCES power\_supply(ps\_id)

ALTER TABLE build\_customer ADD storage\_id INT FOREIGN KEY REFERENCES storage(storage\_id)

ALTER TABLE build\_customer ADD video\_card\_id INT FOREIGN KEY REFERENCES video\_card(video\_card\_id)

CREATE TABLE sys\_req (

sys\_req\_id INTEGER IDENTITY(1,1) PRIMARY KEY,

cpu\_model VARCHAR (50) NULL,

cpu\_series INTEGER NOT NULL,

cpu\_series\_letter VARCHAR (10) NULL,

memory\_size INTEGER NOT NULL,

gcard\_opt1\_model VARCHAR (50) NULL,

gcard\_opt1\_msize INTEGER NULL,

gcard\_opt1\_series INTEGER NULL,

gcard\_opt2\_model VARCHAR (50) NULL,

gcard\_opt2\_msize INTEGER NULL,

gcard\_opt2\_series INTEGER NULL,

storage INTEGER NULL,

game\_id INTEGER NULL FOREIGN KEY REFERENCES games(game\_id),

vid\_edit\_sw\_id INTEGER NULL FOREIGN KEY REFERENCES vid\_edit\_sw (vid\_edit\_sw\_id),

other\_sw\_id INTEGER NULL FOREIGN KEY REFERENCES other\_sw(other\_sw\_id)

)

ALTER TABLE sys\_req ADD cpu\_clock\_speed FLOAT NULL;

ALTER TABLE cpu ADD cpu\_model VARCHAR (50)

ALTER TABLE cpu ALTER COLUMN cpu\_series INTEGER

ALTER TABLE cpu ADD cpu\_series\_letter VARCHAR (10)

ALTER TABLE sys\_req ADD cpu\_2\_model VARCHAR (50)

ALTER TABLE sys\_req ADD cpu\_2\_series INTEGER

ALTER TABLE sys\_req ADD cpu\_2\_series\_letter VARCHAR(10)

SECTION V: INSERTING DATA INTO THE TABLES

-- INSERT DATA INTO MANUFACTURE TABLE --

INSERT INTO

manufacture(mfr\_id, name)

VALUES(1, 'Intel')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(2, 'AMD')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(3, 'G.SKILL')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(4, 'Corsair')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(5, 'Crucial')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(6, 'Samsung')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(7, 'ASUS')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(8, 'ASRock')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(9, 'GIGABYTE')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(10, 'MSI')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(11, 'EVGA')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(12, 'DELL')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(13, 'HP')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(14, 'SEAGATE')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(15, 'WESTERN DIGITAL')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(16, 'IBM')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(17, 'ROSEWILL')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(18, 'THERMALTAKE')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(19, 'FRACTAL DESIGN')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(20, 'COOLER MASTER')

INSERT INTO

manufacture(mfr\_id, name)

VALUES(21, 'DIYPC')

SELECT \* FROM manufacture

-- INSERT DATA INTO FORM FACTOR TABLE --

INSERT INTO

form\_factor(ff\_id, ff\_type)

VALUES(1, 'ATX')

INSERT INTO

form\_factor(ff\_id, ff\_type)

VALUES(2, 'EXTENDED ATX')

INSERT INTO

form\_factor(ff\_id, ff\_type)

VALUES(3, 'MICRO ATX')

INSERT INTO

form\_factor(ff\_id, ff\_type)

VALUES(4, 'MINI ITX')

INSERT INTO

form\_factor(ff\_id, ff\_type)

VALUES(5, 'FLEX ATX')

SELECT \* FROM form\_factor

-- INSERT DATA INTO CPU TABLE --

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type)

VALUES(1, 1, 8, 'Intel UHD Graphics', 'True', 'LGA 1151')

UPDATE cpu SET cpu\_name='INTEL CORE I9-9900K' WHERE cpu\_id='1';

UPDATE cpu SET int\_graphic='INTEL UHD GRAPHICS 630' WHERE cpu\_id='1';

UPDATE cpu SET numb\_thread = 16 WHERE cpu\_id = '1';

UPDATE cpu SET price = '524.99' WHERE cpu\_id = '1';

UPDATE cpu SET clock\_speed= 3.6 WHERE cpu\_id = '1';

UPDATE cpu SET cpu\_model = 'I9' WHERE cpu\_id = '1';

UPDATE cpu SET cpu\_series = 9900 WHERE cpu\_id = '1';

UPDATE cpu SET cpu\_series\_letter = 'K' WHERE cpu\_id = '1';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name)

VALUES(2,2, 8, NULL, 'TRUE', 'SOCKET AM4','AMD RYZEN 7 2700X')

UPDATE cpu SET numb\_thread = 16 WHERE cpu\_id = '2';

UPDATE cpu SET price = '294.99' WHERE cpu\_id = '2';

UPDATE cpu SET hyper\_threading = 'FALSE' WHERE cpu\_id = '2';

UPDATE cpu SET clock\_speed= 3.7 WHERE cpu\_id = '2';

UPDATE cpu SET cpu\_model = 'RYZEN 7' WHERE cpu\_id = '2';

UPDATE cpu SET cpu\_series = 2700 WHERE cpu\_id = '2';

UPDATE cpu SET cpu\_series\_letter = 'X' WHERE cpu\_id = '2';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name)

VALUES (3, 1, 8, 'INTEL UHD GRAPHICS 630', 'TRUE', 'LGA 1151', 'INTEL I7-9700K')

UPDATE cpu SET numb\_thread = 8 WHERE cpu\_id = '3';

UPDATE cpu SET price = '409.99' WHERE cpu\_id = '3';

UPDATE cpu SET cpu\_name = 'INTEL CORE I7-9700K' WHERE cpu\_id='3';

UPDATE cpu SET clock\_speed= 3.6 WHERE cpu\_id = '3';

UPDATE cpu SET cpu\_model = 'I7' WHERE cpu\_id = '3';

UPDATE cpu SET cpu\_series = 9700 WHERE cpu\_id = '3';

UPDATE cpu SET cpu\_series\_letter = 'K' WHERE cpu\_id = '3';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name, numb\_thread, price)

VALUES (4,2, 6, NULL, 'FALSE', 'SOCKET AM3+', 'AMD FX-6300', 6,'68.19' )

UPDATE cpu SET clock\_speed= 3.5 WHERE cpu\_id = '4';

UPDATE cpu SET cpu\_model = 'FX' WHERE cpu\_id = '4';

UPDATE cpu SET cpu\_series = 6300 WHERE cpu\_id = '4';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name, numb\_thread, price)

VALUES (5,2,6, NULL, 'FALSE', 'SOCKET AM4', 'AMD RYZEN 5 2600', 6, '189.99')

UPDATE cpu SET clock\_speed= 3.4 WHERE cpu\_id = '5';

UPDATE cpu SET cpu\_model = 'RYZEN 5' WHERE cpu\_id = '5';

UPDATE cpu SET cpu\_series = 2600 WHERE cpu\_id = '5';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name, numb\_thread, price)

VALUES (6, 1, 6, 'INTEL UHD GRAPHICS 630', 'FALSE', 'LGA 1151', 'INTEL CORE I5-9600K', 6, '264.99' )

UPDATE cpu SET clock\_speed= 3.7 WHERE cpu\_id = '6';

UPDATE cpu SET cpu\_model = 'I5' WHERE cpu\_id = '6';

UPDATE cpu SET cpu\_series = 9600 WHERE cpu\_id = '6';

UPDATE cpu SET cpu\_series\_letter = 'K' WHERE cpu\_id = '6';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name, numb\_thread, price, clock\_speed)

VALUES (7, 1, 2, 'INTEL HD GRAPHICS 610', 'FALSE', NULL, 'LGA 1151', 'INTEL CELERON-G3950', 2, '65.99', 3.0)

UPDATE cpu SET cpu\_model = 'CELERON' WHERE cpu\_id = '7';

UPDATE cpu SET cpu\_series = 3950 WHERE cpu\_id = '7';

UPDATE cpu SET cpu\_series\_letter = 'G' WHERE cpu\_id = '7';

INSERT INTO

cpu (cpu\_id, mfr\_id, numb\_core, int\_graphic, hyper\_threading, socket\_type, cpu\_name, numb\_thread, price, clock\_speed)

VALUES (8, 2, 2, NULL, 'FALSE', NULL, 'SOCKET AM4', 'AMD AHTHLON-200GE', 4, '59.99', 3.2)

UPDATE cpu SET cpu\_model = 'AHTHLON' WHERE cpu\_id = '8';

UPDATE cpu SET cpu\_series = 200 WHERE cpu\_id = '8';

UPDATE cpu SET cpu\_series\_letter = 'GE' WHERE cpu\_id = '8';

SELECT \* FROM cpu

-- INSERT DATA INTO MOTHER BOARD TABLE --

INSERT INTO

mother\_board(mb\_id, mfr\_id, ff\_id, socket\_type, mm\_num\_slot, price)

VALUES(1, 9, 1, 'LGA 1151', 2, 139.99, NULL )

UPDATE mother\_board SET model = 'H370N' WHERE mb\_id=1

INSERT INTO

mother\_board(mb\_id, mfr\_id, ff\_id, socket\_type, mm\_num\_slot, price, model)

VALUES(2, 7, 1, 'LGA 1151', 4, 189.99, NULL, 'Prime Z390-A' )

INSERT INTO

mother\_board(mb\_id, mfr\_id, ff\_id, socket\_type, mm\_num\_slot, price, model)

VALUES(3, 7, 1, 'LGA 1151', 4, 254.71, NULL, 'ROG Strix Z390-E Gaming')

INSERT INTO

mother\_board(mb\_id, mfr\_id, ff\_id, socket\_type, mm\_num\_slot, price, model)

VALUES(4, 10, 1, 'AM4', 4, 139.99, NULL, 'X470 GAMING PLUS')

INSERT INTO

mother\_board(mb\_id, mfr\_id, ff\_id, socket\_type, mm\_num\_slot, price, model)

VALUES(6, 7, 1, 'LGA 1151', 4, 109.99, NULL, 'TUF B360-PLUS GAMING')

INSERT INTO

mother\_board(mb\_id, mfr\_id, ff\_id, socket\_type, mm\_num\_slot, price, model)

VALUES(7, 9, 1, 'AMD3+', 4, 74.99, NULL, 'GA-970A-DS3P')

-- INSERT DATA INTO MEMORY --

INSERT INTO

memory(mm\_id, mfr\_id, size, speed, price, model)

VALUES(1,4,16,3000, 119.99,'Vengeance RGB DRAM 16GB');

INSERT INTO

memory(mm\_id, mfr\_id, size, speed, price, model)

VALUES(2, 3, 16, 2400, 89.99, 'Ripjaws V Series 16GB');

INSERT INTO

memory(mm\_id, mfr\_id, size, speed, price, model)

VALUES(3, 4, 8, 1600, 51.99, 'Vengeance 8GB');

INSERT INTO

memory(mm\_id, mfr\_id, size, speed, price, model)

VALUES (4, 6, 4, 1600, 29.52, 'M378B5173DB0-CK0')

INSERT INTO

memory(mm\_id, mfr\_id, size, speed, price, model)

VALUES(5, 3, 64, 3200, 849.99, 'Trident Z RGB DC Series 64GB')

-- Video Card -- (2080, 2060, AMD Radeon (2 of them), one cheap one)

INSERT INTO

video\_card(video\_card\_id, mfr\_id, mm\_size, mm\_type, numb\_cool, interface, max\_length, core\_clock, price, model)

VALUES(1,9, 8, 'GDDR6', 1, '254-bit',272, 1785, 739.99, 'GV-N2080TURBO OC-8GC' );

INSERT INTO

video\_card(video\_card\_id, mfr\_id, mm\_size, mm\_type, numb\_cool, interface, max\_length, core\_clock, price, model)

VALUES(2,10,6,'GDDR6', 2, '192-bit', 247, 1830,389.99,'RTX 2060 GAMING Z 6G' );

INSERT INTO

video\_card(video\_card\_id, mfr\_id, mm\_size, mm\_type, numb\_cool, interface, max\_length, core\_clock, price, model)

VALUES(3,10,8, 'GDDR5', 2, '512-bit', 278, 1060,499.00, 'R9 390 GAMING 8G' );

INSERT INTO

video\_card(video\_card\_id, mfr\_id, mm\_size, mm\_type, numb\_cool, interface, max\_length, core\_clock, price, model)

VALUES(4,7,8,'HBM2', 3, '2048-bit', 298, 1573, 379.99, 'AREZ-STRIX-RXVEGA56-O8G-GAMING' );

INSERT INTO

video\_card(video\_card\_id, mfr\_id, mm\_size, mm\_type, numb\_cool, interface, max\_length, core\_clock, price, model)

VALUES(5,10, 1, 'DDR3', 0, '64-bit', 146,954, 39.99, 'GT 710 1GD3H LP' );

-- INSERT DATA INTO GAMES TABLE --

INSERT INTO

games (game\_id, model)

VALUES(1, 'Player Unknowns Battle Ground')

INSERT INTO

games (game\_id, model)

VALUES(2, 'Far Cry New Dawn')

INSERT INTO

games ( game\_id, model)

VALUES(3, 'Grand Theft Auto V')

INSERT INTO

games (game\_id, model)

VALUES (4, 'BioShock')

INSERT INTO

games (game\_id, model)

VALUES (5, 'Overwatch')

-- INSERT DATA INTO VIDEO EDITING SOFTWARE TABLE --

SELECT \* FROM vid\_edit\_sw

INSERT INTO

vid\_edit\_sw(vid\_edit\_sw\_id, vid\_edit\_sw\_name )

VALUES(1,'Adobe Premiere Pro')

INSERT INTO

vid\_edit\_sw(vid\_edit\_sw\_id, vid\_edit\_sw\_name )

VALUES(2,'Vegas Pro')

INSERT INTO

vid\_edit\_sw(vid\_edit\_sw\_id, vid\_edit\_sw\_name )

VALUES(3,'VideoPad')

INSERT INTO

vid\_edit\_sw(vid\_edit\_sw\_id, vid\_edit\_sw\_name )

VALUES(4,'PowerDirector')

INSERT INTO

vid\_edit\_sw(vid\_edit\_sw\_id, vid\_edit\_sw\_name )

VALUES(5,'Windows Movie Maker')

-- INSERT DATA INTO OTHER SOFTWARE TABLE --

INSERT INTO

other\_sw(other\_sw\_id, sw\_name)

VALUES(1,'Microsoft Office Suits')

INSERT INTO

other\_sw(other\_sw\_id, sw\_name)

VALUES(2,'RStudio')

INSERT INTO

other\_sw(other\_sw\_id, sw\_name)

VALUES(3,'Ubuntu')

INSERT INTO

other\_sw(other\_sw\_id, sw\_name)

VALUES(4,'AutoCAD')

INSERT INTO

other\_sw(other\_sw\_id, sw\_name)

VALUES(5,'Mathcad')

-- INSERT DATA INTO CUSTOMER TABLE--

SELECT \* FROM customers

INSERT INTO

customers(customer\_id, first\_name, last\_name, phone, address\_1, city, states, country)

VALUES(1, 'Harvey', 'Spector', '315-123-4567', '123 Main st', 'Syracuse', 'NY', 'USA' )

INSERT INTO

customers(customer\_id, first\_name, last\_name, phone, address\_1, address\_2, city, states, country)

VALUES(2,'Mike', 'Ross', '716-476-8569', '244 Apple st', 'apt 3', 'Rochester', 'NY', 'USA' )

INSERT INTO

customers(customer\_id, first\_name, last\_name, phone, address\_1, city, states, country)

VALUES(3, 'Trevor', 'Pearson', '856-112-4587', '5 Cocacola st', 'Houston', 'TX', 'USA' )

INSERT INTO

customers(customer\_id, first\_name, last\_name, phone, address\_1, address\_2, city, states, country)

VALUES(4,'Louis', 'Litt', '546-188-4896', '9001 redbull rd', 'apt# 8', 'Miami', 'FL', 'USA' )

INSERT INTO

customers(customer\_id, first\_name, last\_name, phone, address\_1, city, states, country)

VALUES(5,'Iron', 'Man', '222-455-3325', '111 Avengers Rd', 'New York City', 'NY', 'USA' )

INSERT INTO

sys\_req(cpu\_model, cpu\_series, cpu\_series\_letter, memory\_size, gcard\_opt1\_model, gcard\_opt1\_msize, gcard\_opt1\_series, gcard\_opt2\_model, gcard\_opt2\_msize, gcard\_opt2\_series, storage, game\_id)

VALUES('I5', 4430, NULL, 8, 'GTX', 2, 960, 'R7', 2, 370, 30, 1)

UPDATE sys\_req SET cpu\_clock\_speed = 3.2 WHERE sys\_req\_id = 1

UPDATE sys\_req SET cpu\_2\_model = 'FX', cpu\_2\_series = 6300 WHERE sys\_req\_id = 1

INSERT INTO

sys\_req(cpu\_model, cpu\_series, cpu\_series\_letter, memory\_size, gcard\_opt1\_model, gcard\_opt1\_msize, gcard\_opt1\_series, gcard\_opt2\_model, gcard\_opt2\_msize, gcard\_opt2\_series, storage, game\_id, cpu\_2\_model, cpu\_2\_series)

VALUES('I7', 4790, NULL, 8, 'GTX', 2, 960, 'R7', 2, 370, 30, 1, 'RYZEN 5', 1600)

UPDATE sys\_req SET cpu\_clock\_speed = 3.6 WHERE sys\_req\_id = 2

UPDATE sys\_req SET game\_id= 2 WHERE sys\_req\_id = 2

UPDATE video\_card SET series = 2080 WHERE video\_card\_id = 1

UPDATE video\_card SET chipset\_mfr = 'NVIDIA' WHERE video\_card\_id = 1

UPDATE video\_card SET series\_name = 'RTX' WHERE video\_card\_id=1

UPDATE video\_card SET series = 2060 WHERE video\_card\_id = 2

UPDATE video\_card SET chipset\_mfr = 'NVIDIA' WHERE video\_card\_id = 2

UPDATE video\_card SET series\_name = 'GTX' WHERE video\_card\_id=2

UPDATE video\_card SET series = 390 WHERE video\_card\_id = 3

UPDATE video\_card SET chipset\_mfr = 'RADEON' WHERE video\_card\_id = 3

UPDATE video\_card SET series\_name = 'R9' WHERE video\_card\_id=3

UPDATE video\_card SET series = 56 WHERE video\_card\_id = 4

UPDATE video\_card SET chipset\_mfr = 'RADEON' WHERE video\_card\_id = 4

UPDATE video\_card SET series\_name = 'RX VEGA' WHERE video\_card\_id= 4

UPDATE video\_card SET series = 710 WHERE video\_card\_id = 5

UPDATE video\_card SET chipset\_mfr = 'NVIDIA' WHERE video\_card\_id = 5

UPDATE video\_card SET series\_name = 'GT' WHERE video\_card\_id= 5

SELECT \* FROM mother\_board

SELECT \* FROM memory

SELECT \* FROM video\_card

SELECT \* FROM power\_supply

SELECT \* FROM form\_factor

SELECT \* FROM computer\_case

SELECT \* FROM manufacture

SELECT \* FROM cooling

SELECT \* FROM cpu

SELECT \* FROM socket\_type

SELECT \* FROM build\_customer

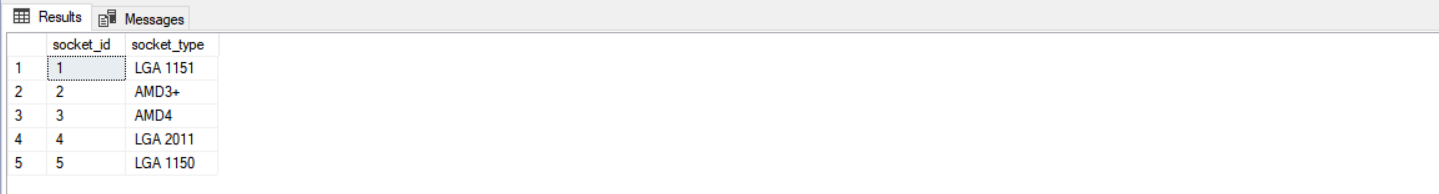
select \* from cm\_socket

select \* from sys\_req

A screenshot of a computer

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceA screenshot of a computer

Description generated with very high confidenceA screenshot of a computer

Description generated with very high confidence

SECTION VI: MAJOR DATA QUESTIONS

-- List of cpu and motherboard that are compatiable -------------------

CREATE VIEW cm\_socket AS

SELECT c.cpu\_name, m.model,m.mm\_num\_slot,c.socket\_type, (c.price+m.price) AS price

from cpu\_socket as c

INNER JOIN motherboard\_socket as m ON c.socket\_id = m.socket\_id

DROP VIEW cm\_socket

SELECT \* FROM cm\_socket

SELECT \* FROM cpu\_socket

SELECT \* FROM motherboard\_socket

A screenshot of a computer

Description generated with very high confidenceA screenshot of a computer

Description generated with very high confidence

-- Cheapest combo (cpu/motherboard/videocard)

SELECT DISTINCT TOP 1 c.\*, v.model, (c.price+v.price) as total

FROM cm\_socket as c, video\_card as v

ORDER BY total ASC

A screenshot of a social media post

Description generated with very high confidence

-- Most expensive combo (cpu/motherboard/videocard)

SELECT DISTINCT TOP 1 c.\*, v.model, (c.price+v.price) as total

FROM cm\_socket as c, video\_card as v

ORDER BY total DESC

A screenshot of a cell phone

Description generated with high confidence

-- List of compartments that are compatible with game\_1-------------------

SELECT DISTINCT dbo\_cpu.cpu\_name, dbo\_mother\_board.model, dbo\_memory.model, dbo\_power\_supply.model, dbo\_cooling.model, dbo\_video\_card.model, dbo\_games.model, dbo\_cpu.price+dbo\_mother\_board.price+ dbo\_memory.price+ dbo\_power\_supply.price+ dbo\_cooling.price+ dbo\_video\_card.price AS Total

FROM dbo\_cpu, dbo\_mother\_board, dbo\_memory, dbo\_power\_supply, dbo\_cooling, dbo\_video\_card, dbo\_sys\_req INNER JOIN dbo\_games ON dbo\_games.game\_id = dbo\_sys\_req.game\_id

WHERE dbo\_games.game\_id = 1 AND dbo\_cpu.clock\_speed >= dbo\_sys\_req.cpu\_clock\_speed AND dbo\_memory.size >= dbo\_sys\_req.memory\_size

ORDER BY dbo\_cpu.price+dbo\_mother\_board.price+ dbo\_memory.price+ dbo\_power\_supply.price+ dbo\_cooling.price+ dbo\_video\_card.price;

A close up of text on a white background

Description generated with high confidence

-- List of compartments that are compatible with game\_2-------------------

SELECT DISTINCT dbo\_cpu.cpu\_name, dbo\_mother\_board.model, dbo\_memory.model, dbo\_power\_supply.model, dbo\_cooling.model, dbo\_video\_card.model, dbo\_games.model, dbo\_cpu.price+dbo\_mother\_board.price+ dbo\_memory.price+ dbo\_power\_supply.price+ dbo\_cooling.price+ dbo\_video\_card.price AS Total

FROM dbo\_cpu, dbo\_mother\_board, dbo\_memory, dbo\_power\_supply, dbo\_cooling, dbo\_video\_card, dbo\_sys\_req INNER JOIN dbo\_games ON dbo\_games.game\_id = dbo\_sys\_req.game\_id

WHERE dbo\_games.game\_id = 2 AND dbo\_cpu.clock\_speed >= dbo\_sys\_req.cpu\_clock\_speed AND dbo\_memory.size >= dbo\_sys\_req.memory\_size

ORDER BY dbo\_cpu.price+dbo\_mother\_board.price+ dbo\_memory.price+ dbo\_power\_supply.price+ dbo\_cooling.price+ dbo\_video\_card.price;

A close up of text on a white background

Description generated with very high confidence

SECTION VII: INTERFACES

1. MAIN SCREEN

A screenshot of a cell phone

Description generated with very high confidence

USERS CHOOSE WHETHER IF THEY WANT TO REGISTER, CHECK PRE-BUILT OR ADD NEW BUILD

A screenshot of a cell phone

Description generated with very high confidence

USERS CAN REGISTER THEIR INFORMATION INTO THE SYSTEM. AFTER THEY ARE DONE, THEY CAN GO BACK TO HOME.

A screenshot of a social media post

Description generated with very high confidence

THIS IS THE SECOND SCREEN TO SEE WHAT ARE THE PRE-BUILT CONFIGURATIONS AVAILABLE.

A screenshot of a cell phone

Description generated with very high confidence

LASTLY, USERS CAN ENTER THEIR OWN BUILD BY USING EXISTING INVENTORY IN THE SYSTEM

SECTION VIII: REPORT

A screenshot of a social media post

Description generated with very high confidence

A screenshot of a social media post

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidence