

CPSC 304 Project Cover Page

Milestone #: 2

Date: Feb 27, 2024

Group Number: 103

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Junsu An	63647879	anjjunsu	anjjunsu@gmail.com
John Do	56442833	djw0626	djw0626123@gmail.com
Paul Tiberghien	10887602	r1e6f	paultiberghien1@gmail.com

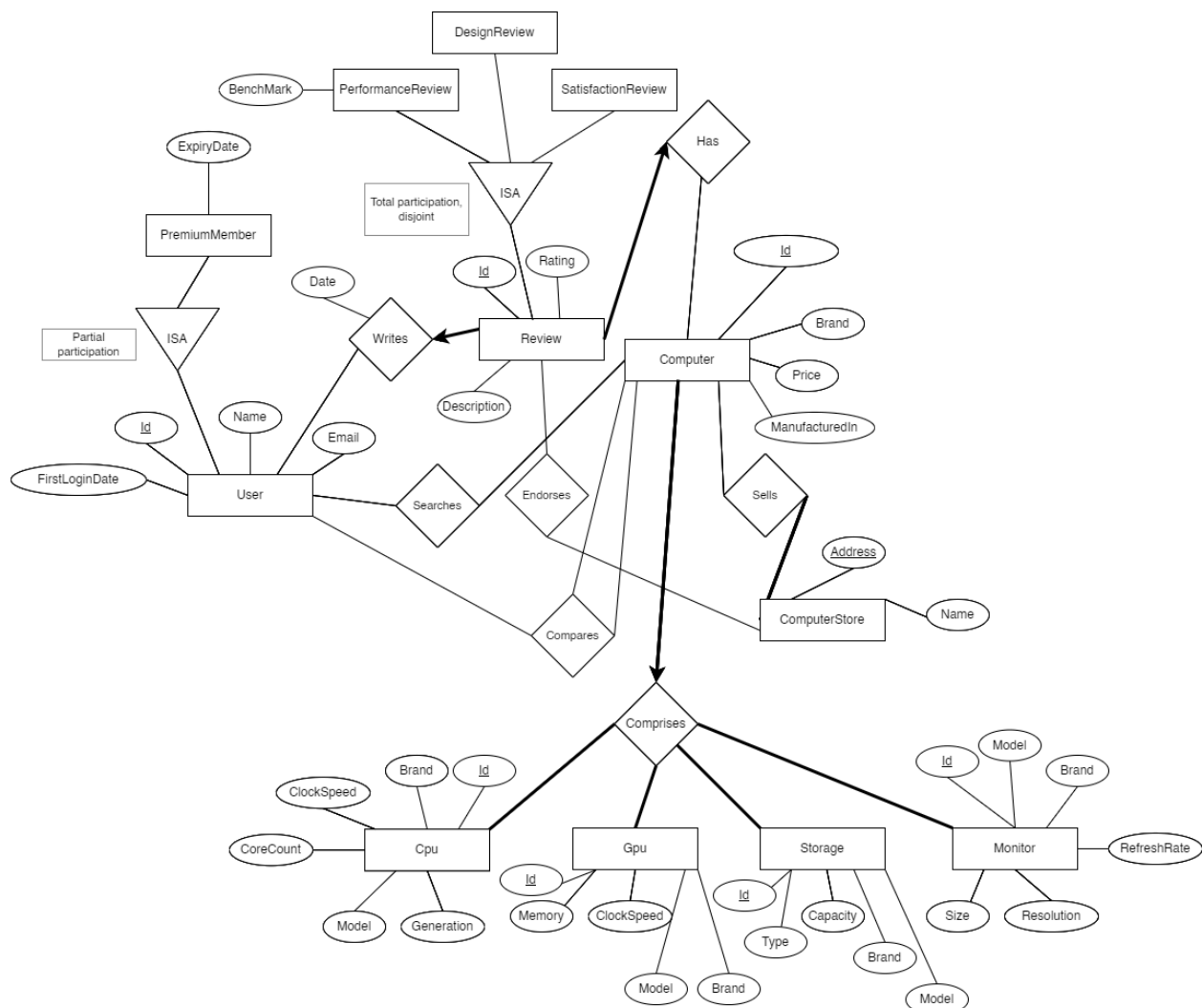
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student Ids are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Summary

Our project allows users to filter, search for, and compare computers/parts from various retailers that meet particular specifications. Stored computer information includes CPU, GPU, storage, and monitor details as well as the location at which it is sold. Our database also allows users to share their experiences regarding the performance, design, and satisfaction of each component to help other users make informed purchases.

Updated Entity-Relationship Diagram



- Removed the Part weak entity along with the ISA relationship associated with it to incorporate TA feedback of lacking entities. Instead, introduced a quinary (5-way) relationship between the Computer, CPU, GPU, Storage, and Monitor entities with

total participation constraints on both sides of the relationship. We show that our database tracks ready-to-use computers, not computer parts. Thus computer parts must be associated with a computer and a computer must consist of the four parts mentioned above.

- The weak entity requirement for the ERD is met by introducing a second ISA entity.
- Added a primary key "ReviewId" to the Review entity.
- Added a ComputerStore entity along with two relationships it participates in, Endorses and Sells, to incorporate TA feedback on lacking relationships.
- Added explicit constraints for both ISA entities.
- Added attribute "AssembledIn" to the Computer entity to introduce meaningful FD between Brand and AssembledIn (Brand -> AssembledIn).
- Changed a few variable names for consistency, including CPU to Cpu and ID to Id.
- Removed entity name in id attributes since attributes are usually denoted in context of entity (EntityName - AttributeName)
- Added a recursive relationship Compares from User between Computers
- Changed the Filter relationship to Search to align with our expectations better

ER Diagram to Schema

Underline represents the primary key

Bold represents a foreign key

Candidate keys indicated as a bulletpoint

- User(Id: INT, Name: VARCHAR(127), Email: VARCHAR(127), FirstLoginDate: DATETIME)
 - Candidate Key(s)
 - (Email)
- PremiumUser(UserId: INT, ExpiryDate: DATETIME)
- PerformanceReview(Id: INT, Rating: INT, Description: VARCHAR(2048), BenchMark: DECIMAL(5, 2), **UserId**: INT, **ComputerId**: INT, Date: DATETIME)
 - UserId, ComputerId, Date are NOT NULL
 - Rating is between 1 and 5
 - BenchMark is between 0.00 and 999.99
 - This captures both Performance Review entity and Writes relationship
- DesignReview(Id: INT, Rating: INT, Description: VARCHAR(2048), **UserId**: INT, **ComputerId**: INT, Date: DATETIME)
 - UserId, ComputerId, Date are NOT NULL
 - Rating is between 1 and 5
 - This captures both Design Review entity and Writes relationship
- SatisfactionReview(Id: INT, Rating: INT, Description: VARCHAR(2048), **UserId**: INT, **ComputerId**: INT, Date: DATETIME)
 - UserId, ComputerId, Date are NOT NULL
 - Rating is between 1 and 5

University of British Columbia, Vancouver

Department of Computer Science

- This captures both Satisfaction Review entity and Writes relationship
- Computer(Id: INT, Brand: VARCHAR(127), Price: DECIMAL(5, 2), AssembledIn: VARCHAR(127), **CpuId**: INT, **GpuId**: INT, **StorageId**: INT, **MonitorId**: INT)
 - CpuId, GpuId, StorageId, MonitorId are NOT NULL
 - Price is between 0.00 and 999.99
 - This captures the Comprises relationship.
- Cpu(Id: INT, Brand: VARCHAR(127), Model: VARCHAR(127), ClockSpeed: DECIMAL(5, 2), CoreCount: INT, Generation: INT)
 - Clock speed (clock rate) is between 0.00 and 100.00 in GHz
 - Candidate Key(s)
 - (Model, ClockSpeed, Generation, CoreCount)
- Gpu(Id: INT, Brand: VARCHAR(127), Model: VARCHAR(127), Memory: INT, ClockSpeed: DECIMAL(3, 2))
 - Memory size in GigaBytes
 - Candidate Key(s)
 - (Model)
- Storage(Id: INT, Brand: VARCHAR(127), Model: VARCHAR(127), Capacity: INT, Type: CHAR(3))
 - Storage type is either HDD or SSD. Hence in CHAR(3) domain
 - Capacity in GigaBytes
 - Candidate Key(s)
 - (Model, Type, Capacity)
- Monitor(Id: INT, Brand: VARCHAR(127), Model: VARCHAR(127), RefreshRate: INT, Size: DECIMAL(3, 1), Resolution: VARCHAR(16))
 - Refresh rate in Hz
 - Resolution expressed as the number of horizontal pixels by the number of vertical pixels on the screen, such as "1920x1080" or "2560x1440".
 - Monitor size is between 0.0 and 99.9 in inches
 - Candidate Key(s)
 - (Model, Size)
- ComputerStore(Address: VARCHAR(255), Name: VARCHAR(255))
- Sells(**StoreAddress**: VARCHAR(255), **ComputerId**: INT)
- Searches(**UserId**: INT, **ComputerId**: INT)
- Compares(**UserId**: INT, **ComputerIdOne**: INT, **ComputerIdTwo**: INT)
- Endorses(**StoreAddress**: VARCHAR(255), **ReviewId**: INT)

Functional Dependencies

FDs:

1. User
 - a. Id -> Name, Email, FirstLoginDate
 - b. Email -> Id, Name, FirstLoginDate
2. PremiumUser

University of British Columbia, Vancouver

Department of Computer Science

- a. Id -> Name, Email, FirstLoginDate, ExpiryDate
 - b. Email -> Id, Name, FirstLoginDate, ExpiryDate
- 3. PerformanceReview
 - a. Id -> Rating, Description, BenchMark, UserId, ComputerId, Date
- 4. DesignReview
 - a. Id -> Rating, Description, UserId, ComputerId, Date
- 5. SatisfactionReview
 - a. Id -> Rating, Description, UserId, ComputerId, Date
- 6. Computer
 - a. Id -> Brand, Price, AssembledIn, CpuId, GpuId, StorageId, MonitorId
 - b. Brand -> AssembledIn
- 7. ComputerStore
 - a. Address -> Name
- 8. Cpu
 - a. Id -> Brand, Model, ClockSpeed, CoreCount, Generation
 - b. Model -> Brand
 - c. Model, ClockSpeed, Generation, CoreCount -> Brand, Id
- 9. Gpu
 - a. Id -> Brand, Model, Memory, ClockSpeed
 - b. Model -> Memory, Brand, ClockSpeed, Id
- 10. Storage
 - a. Id -> Type, Capacity, Model, Brand
 - b. Model, Type, Capacity -> Id, Brand
- 11. Monitor
 - a. Id -> Model, Brand, Resolution, Size, RefreshRate
 - b. Model, Size -> Id, Brand, RefreshRate

Normalization

We normalize our schemas in BCNF. We only show normalizations of tables that are not already in BCNF.

1. Computer

- a. Computer is not in BCNF because in the FD Brand -> AssembledIn, Brand isn't a superkey for Computer. Thus, we decompose Computer.
- b. Computer1(Brand, AssembledIn), Computer2(Id, Brand, Price, **CpuId**, **GpuId**, **StorageId**, **MonitorId**)
- c. We now rename Computer1 to BrandAssembles and Computer2 to Computer to better reflect its purpose.
- d. Final tables
 - i. BrandAssembles(Brand, AssembledIn)
 - ii. Computer(Id, Brand, Price, **CpuId**, **GpuId**, **StorageId**, **MonitorId**)

2. CPU

- a. CPU is not in BCNF because, in the FD Model -> Brand, Model isn't a superkey for CPU
- b. CPU1(Id, Model, ClockSpeed, CoreCount, Generation), CPU2(Model, Brand)
- c. We now rename CPU1 to CPU and CPU2 to CPUBrand to better reflect its purpose
- d. Final tables
 - i. CPU(Id, Model, ClockSpeed, CoreCount, Generation)
 - 1. Candidate keys
 - a. (Model, ClockSpeed, Generation, CoreCount)
 - ii. CPUBrand(Model, Brand)

SQL DDL Statements - Create Tables

```
CREATE TABLE User (  
  Id INT PRIMARY KEY,  
  Name VARCHAR(127),  
  Email VARCHAR(127),  
  FirstLoginDate DATETIME,  
);
```

```
CREATE TABLE PremiumUser (  
  UserId INT PRIMARY KEY,  
  ExpiryDate DATETIME,  
  FOREIGN KEY (UserId) REFERENCES User(Id)  
);
```

```
CREATE TABLE Computer (  
  ComputerId INT PRIMARY KEY,  
  Brand VARCHAR(127),  
  Price DECIMAL(5, 2),  
  cpuId INT,  
  gpuId INT,  
  storageId INT,  
  MonitorId INT,  
  FOREIGN KEY (cpuId) REFERENCES Cpu(Id),  
  FOREIGN KEY (gpuId) REFERENCES Gpu(Id),  
  FOREIGN KEY (storageId) REFERENCES Storage(Id),  
  FOREIGN KEY (MonitorId) REFERENCES Monitor(Id)
```

University of British Columbia, Vancouver

Department of Computer Science

```
);
```

```
CREATE TABLE BrandAssemblies (  
  Brand VARCHAR(127) PRIMARY KEY,  
  AssembledIn VARCHAR (127)  
);
```

```
CREATE TABLE Cpu (  
  Id INT PRIMARY KEY,  
  Model INT,  
  ClockSpeed DECIMAL(3, 2),  
  CoreCount INT,  
  Generation INT  
);
```

```
CREATE TABLE CpuBrand (  
  Model VARCHAR(127) PRIMARY KEY,  
  Brand VARCHAR(127)  
);
```

```
CREATE TABLE Gpu (  
  Id INT PRIMARY KEY,  
  Brand VARCHAR(127),  
  Model VARCHAR(127),  
  Memory INT,  
  ClockSpeed DECIMAL(3, 2)  
);
```

```
CREATE TABLE Storage (  
  Id INT PRIMARY KEY,  
  Model VARCHAR(127),  
  Brand VARCHAR(127),  
  Capacity INT,  
  Type CHAR(3)  
);
```

```
CREATE TABLE Monitor (  
  Id INT PRIMARY KEY,  
  Brand VARCHAR(127),  
  Model VARCHAR(127),  
  RefreshRate INT,  
  Size DECIMAL(2, 1),
```

University of British Columbia, Vancouver

Department of Computer Science

```
Resolution VARCHAR(15),  
RefreshRate INT  
);
```

```
CREATE TABLE PerformanceReview (  
ReviewId INT PRIMARY KEY,  
Rating INT,  
Description VARCHAR(2048),  
BenchMark DECIMAL(5, 2),  
UserId INT,  
ComputerId INT,  
Date DATETIME,  
FOREIGN KEY (UserId) REFERENCES User(Id),  
FOREIGN KEY (ComputerId) REFERENCES Computer(Id)  
-- Constraints for Rating and BenchMark should be enforced by the application logic  
);
```

```
CREATE TABLE DesignReview (  
ReviewId INT PRIMARY KEY,  
Rating INT,  
Description VARCHAR(255),  
UserId INT,  
ComputerId INT,  
Date DATETIME,  
FOREIGN KEY (UserId) REFERENCES User(Id),  
FOREIGN KEY (ComputerId) REFERENCES Computer(Id)  
-- Constraints for Rating should be enforced by the application logic  
);
```

```
CREATE TABLE SatisfactionReview (  
ReviewId INT PRIMARY KEY,  
Rating INT,  
Description VARCHAR(255),  
UserId INT,  
ComputerId INT,  
Date DATETIME,  
FOREIGN KEY (UserId) REFERENCES User(Id),  
FOREIGN KEY (ComputerId) REFERENCES Computer(Id)  
-- Constraints for Rating should be enforced by the application logic  
);
```


University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE ComputerStore (  
  Address VARCHAR(255) PRIMARY KEY,  
  Name VARCHAR(255)  
);
```

```
CREATE TABLE Sells (  
  StoreAddress VARCHAR(255),  
  ComputerId INT,  
  PRIMARY KEY (StoreAddress, ComputerId),  
  FOREIGN KEY (StoreAddress) REFERENCES ComputerStore(Address),  
  FOREIGN KEY (ComputerId) REFERENCES Computer(Id)  
);
```

```
CREATE TABLE Searches(  
  UserId INT,  
  ComputerId INT,  
  PRIMARY KEY (UserId, ComputerId),  
  FOREIGN KEY (UserId) REFERENCES User(Id),  
  FOREIGN KEY (ComputerId) REFERENCES Computer(Id)  
);
```

```
CREATE TABLE Compares (  
  UserId INT,  
  ComputerIdOne INT,  
  ComputerIdTwo INT,  
  PRIMARY KEY (UserId, ComputerIdOne, ComputerIdTwo),  
  FOREIGN KEY (UserId) REFERENCES User(Id),  
  FOREIGN KEY (ComputerIdOne) REFERENCES Computer(Id),  
  FOREIGN KEY (ComputerIdTwo) REFERENCES Computer(Id)  
);
```

```
CREATE TABLE Endorses (  
  ReviewId INT,  
  StoreAddress VARCHAR(255),  
  FOREIGN KEY (ReviewId) REFERENCES PerformanceReview(Id),  
  FOREIGN KEY (StoreAddress) REFERENCES ComputerStore(Address)  
);
```

INSERT Statements (Population)

```
INSERT INTO User (Id, Name, Email, FirstLoginDate) VALUES
(1, 'John Doe', 'john.doe@example.com', '2024-03-01 09:00:00');
(2, 'Jane Smith', 'jane.smith@example.com', '2024-02-28 09:00:00'),
(3, 'Alex Johnson', 'alex.johnson@example.com', '2024-01-15 09:00:00'),
(4, 'Maria Garcia', 'maria.garcia@example.com', '2023-12-10 09:00:00');
(5, 'Paul Maximus', 'paul.maximus@example.com', '2023-12-15 09:00:00');
```

```
INSERT INTO PremiumUser(UserId, ExpiryDate) VALUES
(1, '2024-03-01 09:00:00')
(2, '2025-02-28 09:00:00'),
(3, '2025-01-15 09:00:00'),
(4, '2024-12-10 09:00:00');
(5, '2025-12-15 09:00:00');
```

```
INSERT INTO Cpu (Id, Model, ClockSpeed, CoreCount, Generation) VALUES
(1, 'Core i7', 3.8, 8, 10),
(2, 'Core i5', 2.9, 4, 10),
(3, 'Ryzen 7', 3.6, 8, 4),
(4, 'Ryzen 5', 3.4, 6, 4),
(5, 'Core i9', 4.0, 8, 11);
```

```
INSERT INTO CpuBrand(Model, Brand) VALUES
('Core i7', 'Intel'),
('Core i5', 'Intel'),
('Ryzen 7', 'AMD'),
('Ryzen 5', 'AMD'),
('Core i9', 'Intel');
```

```
INSERT INTO Gpu (Id, Brand, Model, Memory, ClockSpeed) VALUES
(1, 'NVIDIA', 'RTX 3080', 10, 1.7),
(2, 'NVIDIA', 'RTX 3070', 8, 1.5),
(3, 'AMD', 'Radeon RX 6800', 16, 1.8),
(4, 'AMD', 'Radeon RX 6700 XT', 12, 1.6),
(5, 'NVIDIA', 'RTX 3060', 12, 1.4);
```

```
INSERT INTO Storage (Id, Model, Brand, Capacity, Type) VALUES
(1, 'ModelX1', 'BrandX', 256, 'SSD'),
(2, 'ModelX2', 'BrandX', 512, 'SSD'),
(3, 'ModelY1', 'BrandY', 1024, 'HDD'),
```

University of British Columbia, Vancouver

Department of Computer Science

```
(4, 'ModelY2', 'BrandY', 2048, 'HDD'),  
(5, 'ModelZ', 'BrandZ', 512, 'SSD');
```

```
INSERT INTO Monitor (Id, Brand, Model, Size, Resolution, RefreshRate) VALUES  
(1, 'BrandM', 'ModelM', 27.0, '2560x1440', 144),  
(2, 'BrandM', 'ModelN', 24.0, '1920x1080', 60),  
(3, 'BrandO', 'ModelO', 21.5, '1920x1080', 75),  
(4, 'BrandP', 'ModelP', 24.0, '2560x1440', 120),  
(5, 'BrandQ', 'ModelQ', 27.0, '1920x1080', 240);
```

```
INSERT INTO Computer (Id, Brand, Price, CpuId, GpuId, StorageId, MonitorId) VALUES  
(1, 'Dell', 3200.00, 1, 2, 2, 3),  
(2, 'Razor', 2100.00, 3, 3, 3, 1),  
(3, 'Apple', 3499.99, 2, 1, 1, 3),  
(4, 'HP', 1500.00, 4, 4, 4, 4),  
(5, 'Lenovo', 18000, 5, 5, 5, 5);
```

```
INSERT INTO BrandAssembles (Brand, AssembledIn) VALUES  
( 'Dell', 'China'),  
( 'Apple', 'USA'),  
( 'Razor', 'Singapore'),  
( 'HP', 'Japan'),  
( 'Lenovo', 'Taiwan');
```

```
INSERT INTO PerformanceReview (Id, Rating, Description, BenchMark, UserId,  
ComputerId, Date) VALUES  
(1, 5, 'Excellent performance and value.', 10000.00, 1, 1, '2024-03-01 09:00:00'),  
(2, 4, 'Good performance for the price.', 8500.00, 2, 2, '2024-03-02 09:30:00'),  
(3, 3, 'Average performance, decent for everyday tasks.', 7000.00, 3, 3,  
'2024-03-03 10:00:00'),  
(4, 2, 'Below average performance, struggles with heavy tasks.', 5500.00, 4, 4,  
'2024-03-04 10:30:00'),  
(5, 1, 'Poor performance, not recommended for modern applications.', 4000.00, 5, 5,  
'2024-03-05 11:00:00');
```

```
INSERT INTO DesignReview (Id, Rating, Description, UserId, ComputerId, Date) VALUES  
(6, 4, 'Sleek and modern design.', 1, 2, '2024-03-06 10:00:00'),  
(7, 5, 'Outstanding design and aesthetics.', 2, 3, '2024-03-07 11:00:00'),  
(8, 3, 'Functional design, but nothing special.', 3, 4, '2024-03-08 12:00:00'),  
(9, 2, 'Design is a bit outdated.', 4, 1, '2024-03-09 13:00:00'),  
(10, 1, 'Unappealing design and poor ergonomics.', 5, 2, '2024-03-10 14:00:00');
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO SatisfactionReview (Id, Rating, Description, UserId, ComputerId,
writtenDate) VALUES
(11, 4, 'Quite satisfied with the purchase overall.', 1, 1, '2024-03-11 15:00:00'),
(12, 3, 'Satisfied, but there are some issues.', 2, 2, '2024-03-12 16:00:00'),
(13, 5, 'Extremely satisfied, exceeded expectations.', 3, 3, '2024-03-13
17:00:00'),
(14, 2, 'Not satisfied, many problems encountered.', 4, 4, '2024-03-14 18:00:00'),
(15, 1, 'Very unsatisfied, would not recommend.', 5, 5, '2024-03-15 19:00:00');
```

```
INSERT INTO ComputerStore (Address, Name) VALUES
('123 Tech Road', 'Tech Store A'),
('456 Innovation Ave', 'Tech Store B'),
('789 Computing Blvd', 'Tech Store C'),
('101 Data Dr', 'Tech Store D'),
('202 Info Way', 'Tech Store E');
```

```
INSERT INTO Sells (Address, ComputerId) VALUES
('123 Tech Road', 1),
('456 Innovation Ave', 2),
('789 Computing Blvd', 3),
('101 Data Dr', 4),
('202 Info Way', 5);
```

```
INSERT INTO Searches (UserId, ComputerId) VALUES
(1, 2),
(2, 3),
(3, 4),
(4, 5),
(5, 1);
```

```
INSERT INTO Compares (UserId, ComputerIdOne, ComputerIdTwo) VALUES
(1, 3, 2),
(2, 4, 3),
(3, 5, 4),
(4, 1, 3),
(5, 2, 4);
```

```
INSERT INTO Endorses (ReviewId, StoreAddress) VALUES
(1, '123 Tech Road'),
(2, '456 Innovation Ave'),
(3, '789 Computing Blvd'),
```

University of British Columbia, Vancouver
Department of Computer Science

```
(4, '101 Data Dr'),  
(5, '202 Info Way');
```