

# CPSC 304 Project Cover Page

Milestone #: 2

Date: 10/14/22

Group Number: 94

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alexander Liteplo	14470900	x5f6h	alexanderliteplo@gmail.com
Bhairaw Aryan	81199119	k0i3b	baryan01@student.ubc.ca
Lucas Moynier	94057809	s2x2b	lmoynier@student.ubc.ca

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

## Deliverables

Each group must provide the following as a single PDF file:

1. A completed cover page (template on Canvas)
2. The ER diagram you are basing your item #3 (below) on. This ER diagram may be the same as your milestone 1 submission or it might be different. If you have made changes from the version submitted in milestone 1, attach a note indicating what changes have been made and why.
  - Made the product accumulates review relationships one-to-many. A review can be associated with a singular product but a product can have many reviews
  - Removed attributes name and dob from user table
  - Added total participation on the review side for the relationship since a review must be associated with a product.
  - Added date attribute for global inventory because we want to record the dates people are making searches.
  - Added algorithm attribute for AI review because we may use different ML algorithms to review products.
  - Made the manage relationship bold because it is part of a weak-entity
  - Added UserInterface Entity to add theme features to our website
3. The schema derived from your ER diagram (above). For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward. For each table:
  - a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...))
  - b. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints that the table must maintain.
4. Functional Dependencies (FDs)
  - a. Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key).

Note: In your list of FDs, there must be some kind of valid FD other than those identified by a PK or CK. If you observe that no relations have FDs other than

the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process.

## 5. Normalization

a. Normalize each of your tables to be in 3NF or BCNF. Give the list of tables, their primary keys, their candidate keys, and their foreign keys after normalization.

You should show the steps taken for the decomposition. Should there be errors, and no work is shown, no partial credit can be awarded without steps shown. The format should be the same as Step 3, with tables listed similar to Table1(attr1:domain1, attr2:domain2, ...). ALL Tables must be listed, not only the ones post normalization.

6. The SQL DDL statements required to create all the tables from item #5. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc.

7. INSERT statements to populate each table with at least 5 tuples. You will likely want to have more than 5 tuples so that you can have meaningful queries later on.

**NOTE: PRODUCT GLOBAL INVENTORY MUST BE ONE TO MANY**

## **3. Schema:**

Total participation for Users and credentials so we combine the tables

**User** (address, username, Login ID(UNIQUE), password, **globalInventoryID**, **advertisementID**, postal code, address, city, country)

- PK = username
- CK = username, Login ID, password
- FK = globalInventoryID references GlobalInventory
  - advertisementID references Advertisement

**Advertisement** (ID, brand)

- PK = ID
- CK = ID

**Cart**(cartID, **username**)

- PK = (cartID, username)
- CK = (cartID, username)

**CartContainsProduct**(**cartID**, **username**, productID)

- PK = cartID, username, productID
- CK = cartID, username, productID
- FK = (cartID, username) references Cart  
productID references Products

**Product**(productID, price, name, **globalInventoryID**, company)

- PK = productID
- FK = globalInventoryID references GlobalInventory
- CK = productID

**GlobalInventory**(globalInventoryID, search, date)

- PK = globalInventoryID
- CK = globalInventoryID

**AIReview**(dateGenerated, Algorithm, reviewID, score, **productID**)

- PK = reviewID
- CK = reviewID
- FK = ProductID references product

**AmazonReview**(reviewID, Score, ReviewerName, dateReviewed, **productID**)

- PK = reviewID
- CK = reviewID
- FK = productID references product

**UserInterface**(EnumCode, RGB value, Theme)

- PK = EnumCode
- CK = EnumCode, RGB value

## 4. Functional Dependencies:

User:

PostalCode -> City

Address, Postal Code -> country, City

UserInterface:

Theme -> RGB value

EnumCode->Theme, RGB value (this is trivial but we want to show)

## 5. Normalization

### Using 3NF synthesis

User (address, username, Login ID(UNIQUE), password, **globalInventoryID**, **advertisementID**, postalCode, address, city, country)

Step 1. (minimal cover)

PostalCode -> City

~~Address, Postal code -> City~~

Address, Postal Code -> Country

Username -> address

Username -> password

Username -> LOGIN ID

Username -> globalInventoryID

Username -> advertisementID

Username -> postalCode

~~Username -> City~~

~~Username -> Country~~

LOGIN ID, password, globalInventoryID, advertisementID, postal code, address, city

UserR1(PostalCode, City)

CK: PostalCode

UserR2(Address, Postal Code, Country)

CK: Address, Postal Code, Country

UserR3(Username, Address)

CK: Username, Address

UserR4(Username, password)

CK: Username, password

UserR5(Username, LOGINID)

CK: Username, LOGINID

UserR6(Username, **globalInventoryID**, **advertisementID**)

CK: Username

FK: globalInventoryID references GlobalInventory

advertisementID references Advertisement

UserR7(Username, postal code)

CK: Username

### **Using 3NF synthesis**

UserInterface (EnumCode, RGB value, Theme)

FDs:

(normal form)

Theme-> RGB value

EnumCode->Theme

EnumCode->RGB value

We now remove EnumCode->RGB value because it's redundant.

From this we create UserInteface1(Theme, rgbValue) UserInterface2(EnumCode, Theme)

A relationship that contains all the attributes of the key exists (UserInterface2), so we are now successfully in 3NF.

## **6. SQL DDL statements:**

```
CREATE TABLE Advertisement(  
    id INT,  
    brand VARCHAR(255),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE CartContainsProduct(  
    cartID INT,  
    username VARCHAR(255),  
    productID INT,  
    PRIMARY KEY (cartID , username) ,  
    FOREIGN KEY (cartID, username) REFERENCES Cart,  
    FOREIGN KEY (productID) REFERENCES Products  
);
```

```
CREATE TABLE GlobalInventory(  
    globalInventoryID INT,  
    search VARCHAR(255),  
    date DATETIME,  
    PRIMARY KEY (globalInventoryID)  
);
```

```
CREATE TABLE AmazonReview(  
    reviewID INT,  
    Score FLOAT,  
    ReviewerName VARCHAR(255),  
    dateReviewed DATETIME,  
    productID DECIMAL NOT NULL,  
    FOREIGN KEY productID REFERENCES Product,  
    ON DELETE NO ACTION,  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE Cart(  
    cartID Integer,  
    username VARCHAR(255),  
    PRIMARY KEY (cartID, username),  
    FOREIGN KEY username REFERENCES User,  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE Product(  
    productID Integer PRIMARY KEY,  
    price FLOAT,  
    name VARCHAR(255),  
    globalInventoryID Integer NOT NULL,  
    company VARCHAR(255),  
    FOREIGN KEY globalInventoryID REFERENCES GlobalInventory,  
    ON DELETE NO ACTION,  
    ON UPDATE CASCADE  
);
```



```
CREATE TABLE AIReview(  
    reviewID Integer PRIMARY KEY,  
    dateGenerated DATETIME,  
    algorithm VARCHAR(255),  
    score FLOAT,  
    productID DECIMAL NOT NULL,  
    FOREIGN KEY productID REFERENCES Product,  
    ON DELETE NO ACTION,  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE UserInterface1(  
    theme VARCHAR(255) PRIMARY KEY,  
    rgbValue INTEGER NOT NULL  
);
```

```
CREATE TABLE UserInterface2(  
    enumCode Integer PRIMARY KEY,  
    theme VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE UserR1(  
    postalCode VARCHAR(6),  
    city VARCHAR(255),  
    PRIMARY KEY(postalCode)  
);
```

```
CREATE TABLE UserR2(  
    address VARCHAR(255),  
    postalCode VARCHAR(6),  
    country VARCHAR(255),  
    PRIMARY KEY(address),  
    PRIMARY KEY(postalCode)  
);
```

```
CREATE TABLE UserR3(  
    userName VARCHAR(255),  
    address VARCHAR(6),  
    PRIMARY KEY(userName)  
);
```

```
CREATE TABLE UserR4(  
    userName VARCHAR(255),  
    password VARCHAR(255),  
    PRIMARY KEY(userName)  
);
```

```
CREATE TABLE UserR5(  
    userName VARCHAR(255),  
    loginID VARCHAR(255) NOT NULL UNIQUE,  
    PRIMARY KEY(userName)  
);
```

```
CREATE TABLE UserR6(  
    userName VARCHAR(255),  
    globalInventoryID INT,  
    advertisementID INT,  
    PRIMARY KEY(userName),  
    FOREIGN KEY(globalInventoryID) REFERENCES (GlobalInventory),  
    FOREIGN KEY(advertisementID) REFERENCES (Advertisement)  
);
```

```
CREATE TABLE UserR7(  
    userName VARCHAR(255),  
    postalCode VARCHAR(6),  
    PRIMARY KEY(userName)  
);
```

## **7. INSERT statements:**

```
INSERT INTO Advertisement VALUES (1, 'Apple');
INSERT INTO Advertisement VALUES (2, 'Samsung');
INSERT INTO Advertisement VALUES (3, 'Google');
INSERT INTO Advertisement VALUES (4, 'Microsoft');
INSERT INTO Advertisement VALUES (5, 'Amazon');
```

```
INSERT INTO Cart(cartID, username) VALUES (1, 'sickduck');
INSERT INTO Cart(cartID, username) VALUES (2, 'noobmaster68');
INSERT INTO Cart(cartID, username) VALUES (3, 'xanderminer');
INSERT INTO Cart(cartID, username) VALUES (4, 'tryhardallday');
INSERT INTO Cart(cartID, username) VALUES (5, 'shadowblade');
```

```
INSERT INTO Product(productID, price, name, globalInventoryID, company)
VALUES (1, 420, "testproduct1", 1, "testcompany1");
INSERT INTO Product(productID, price, name, globalInventoryID, company)
VALUES (2, 420, "testproduct2", 2, "testcompany2");
INSERT INTO Product(productID, price, name, globalInventoryID, company)
VALUES (3, 420, "testproduct3", 3, "testcompany3");
INSERT INTO Product(productID, price, name, globalInventoryID, company)
VALUES (4, 420, "testproduct4", 4, "testcompany4");
INSERT INTO Product(productID, price, name, globalInventoryID, company)
VALUES (5, 420, "testproduct5", 5, "testcompany5");
```

```
INSERT INTO CartContainsProduct(cartID, username, productID ) VALUES(1,
"shadowblade", 1);
INSERT INTO CartContainsProduct(cartID, username, productID ) VALUES(2,
"sickduck", 2);
INSERT INTO CartContainsProduct(cartID, username, productID ) VALUES(3,
"noobmaster68", 3);
INSERT INTO CartContainsProduct(cartID, username, productID ) VALUES(4,
"xanderminer", 4);
INSERT INTO CartContainsProduct(cartID, username, productID ) VALUES(5,
"tryhardallday", 5);
```

**University of British Columbia, Vancouver**  
**Department of Computer Science**

---

```
INSERT INTO GlobalInventory(globalInventoryID, search, date ) VALUES(1,
"shoes", 2018-01-01);
INSERT INTO GlobalInventory(globalInventoryID, search, date ) VALUES(2,
"gpus", 2018-01-01);
INSERT INTO GlobalInventory(globalInventoryID, search, date ) VALUES(3,
"shirt", 2018-01-01);
INSERT INTO GlobalInventory(globalInventoryID, search, date ) VALUES(4,
"chair", 2018-01-01);
INSERT INTO GlobalInventory(globalInventoryID, search, date ) VALUES(5,
"fan", 2018-01-01);
```

```
INSERT INTO AmazonReview(reviewID, Score, ReviewerName, dateReviewed,
productID ) VALUES(1, 5, "shadowblade", 2018-01-01, 1);
INSERT INTO AmazonReview(reviewID, Score, ReviewerName, dateReviewed,
productID ) VALUES(2, 4, "sickduck", 2018-01-01, 2);
INSERT INTO AmazonReview(reviewID, Score, ReviewerName, dateReviewed,
productID ) VALUES(3, 3, "noobmaster68", 2018-01-01, 3);
INSERT INTO AmazonReview(reviewID, Score, ReviewerName, dateReviewed,
productID ) VALUES(4, 2, "xanderminer", 2018-01-01, 4);
INSERT INTO AmazonReview(reviewID, Score, ReviewerName, dateReviewed,
productID ) VALUES(5, 1, "tryhardallday", 2018-01-01, 5);
```

```
INSERT INTO AIReview(reviewID, dateGenerated, algorithm, score,
productID ) VALUES(1, 2018-01-01, "testalgo", 5, 1);
INSERT INTO AIReview(reviewID, dateGenerated, algorithm, score,
productID ) VALUES(2, 2018-01-01, "testalgo", 4, 2);
INSERT INTO AIReview(reviewID, dateGenerated, algorithm, score,
productID ) VALUES(3, 2018-01-01, "testalgo", 3, 3);
INSERT INTO AIReview(reviewID, dateGenerated, algorithm, score,
productID ) VALUES(4, 2018-01-01, "testalgo", 2, 4);
INSERT INTO AIReview(reviewID, dateGenerated, algorithm, score,
productID ) VALUES(5, 2018-01-01, "testalgo", 1, 5);
```

```
INSERT INTO UserInterface1(theme, rgbValue ) VALUES("darktheme", 123456);
INSERT INTO UserInterface1(theme, rgbValue ) VALUES("lighttheme",
654321);
INSERT INTO UserInterface1(theme, rgbValue ) VALUES("greentheme",
123654);
INSERT INTO UserInterface1(theme, rgbValue ) VALUES("redtheme", 456123);
```

**University of British Columbia, Vancouver**  
**Department of Computer Science**

---

```
INSERT INTO UserInterface1(theme, rgbValue ) VALUES("bluetheme", 789456);
```

```
INSERT INTO UserInterface2(enumCode, theme ) VALUES(1, "darktheme");
INSERT INTO UserInterface2(enumCode, theme ) VALUES(2, "lighttheme");
INSERT INTO UserInterface2(enumCode, theme ) VALUES(3, "greentheme");
INSERT INTO UserInterface2(enumCode, theme ) VALUES(4, "redtheme");
INSERT INTO UserInterface2(enumCode, theme ) VALUES(5, "bluetheme");
```

```
INSERT INTO UserR1(postalCode, city) VALUES("123456", "testcity1");
INSERT INTO UserR1(postalCode, city) VALUES("654321", "testcity2");
INSERT INTO UserR1(postalCode, city) VALUES("123654", "testcity3");
INSERT INTO UserR1(postalCode, city) VALUES("456123", "testcity4");
INSERT INTO UserR1(postalCode, city) VALUES("789456", "testcity5");
```

```
INSERT INTO UserR2(address, postalCode, country) VALUES("testaddress1",
"123456", "testcountry1");
INSERT INTO UserR2(address, postalCode, country) VALUES("testaddress2",
"654321", "testcountry2");
INSERT INTO UserR2(address, postalCode, country) VALUES("testaddress3",
"123654", "testcountry3");
INSERT INTO UserR2(address, postalCode, country) VALUES("testaddress4",
"456123", "testcountry4");
INSERT INTO UserR2(address, postalCode, country) VALUES("testaddress5",
"789456", "testcountry5");
```

```
INSERT INTO UserR3(userName, address) VALUES("shadowblade",
"testaddress1");
INSERT INTO UserR3(userName, address) VALUES("sickduck", "testaddress2");
INSERT INTO UserR3(userName, address) VALUES("noobmaster68",
"testaddress3");
INSERT INTO UserR3(userName, address) VALUES("xanderminer",
"testaddress4");
INSERT INTO UserR3(userName, address) VALUES("tryhardallday",
"testaddress5");
```

```
INSERT INTO UserR4(userName, password) VALUES("shadowblade",
"testpassword1");
```

## University of British Columbia, Vancouver

### Department of Computer Science

---

```
INSERT INTO UserR4 (userName, password) VALUES ("sickduck",
"testpassword2");
INSERT INTO UserR4 (userName, password) VALUES ("noobmaster68",
"testpassword3");
INSERT INTO UserR4 (userName, password) VALUES ("xanderminer",
"testpassword4");
INSERT INTO UserR4 (userName, password) VALUES ("tryhardallday",
"testpassword5");

INSERT INTO UserR5 (userName, loginID) VALUES ("shadowblade",
"testloginID1");
INSERT INTO UserR5 (userName, loginID) VALUES ("sickduck", "testloginID2");
INSERT INTO UserR5 (userName, loginID) VALUES ("noobmaster68",
"testloginID3");
INSERT INTO UserR5 (userName, loginID) VALUES ("xanderminer",
"testloginID4");
INSERT INTO UserR5 (userName, loginID) VALUES ("tryhardallday",
"testloginID5");

INSERT INTO UserR6 (userName, globalInventoryID, advertisementID)
VALUES ("shadowblade", 1, 1);
INSERT INTO UserR6 (userName, globalInventoryID, advertisementID)
VALUES ("sickduck", 2, 2);
INSERT INTO UserR6 (userName, globalInventoryID, advertisementID)
VALUES ("noobmaster68", 3, 3);
INSERT INTO UserR6 (userName, globalInventoryID, advertisementID)
VALUES ("xanderminer", 4, 4);
INSERT INTO UserR6 (userName, globalInventoryID, advertisementID)
VALUES ("tryhardallday", 5, 5);

INSERT INTO UserR7 (userName, postalCode) VALUES ("shadowblade", "123456");
INSERT INTO UserR7 (userName, postalCode) VALUES ("sickduck", "654321");
INSERT INTO UserR7 (userName, postalCode) VALUES ("noobmaster68", "123654");
INSERT INTO UserR7 (userName, postalCode) VALUES ("xanderminer", "456123");
INSERT INTO UserR7 (userName, postalCode) VALUES ("tryhardallday",
"789456");
```

