CPSC 304 Project Cover Page

Milestone #: 2

Date: 20 of October

Group Number: 43

Name	Student Number	CS Alias (Userid)	Preferred email Address
Nicolas Rubiano Avila	32234049	L8l3a	n.rubiano01@gmail.com
Song Shi	91903989	w7v3o	771plant@gmail.com
Jason Xu	36612943	v4j8y	Jasonxuluyang@gmail.com

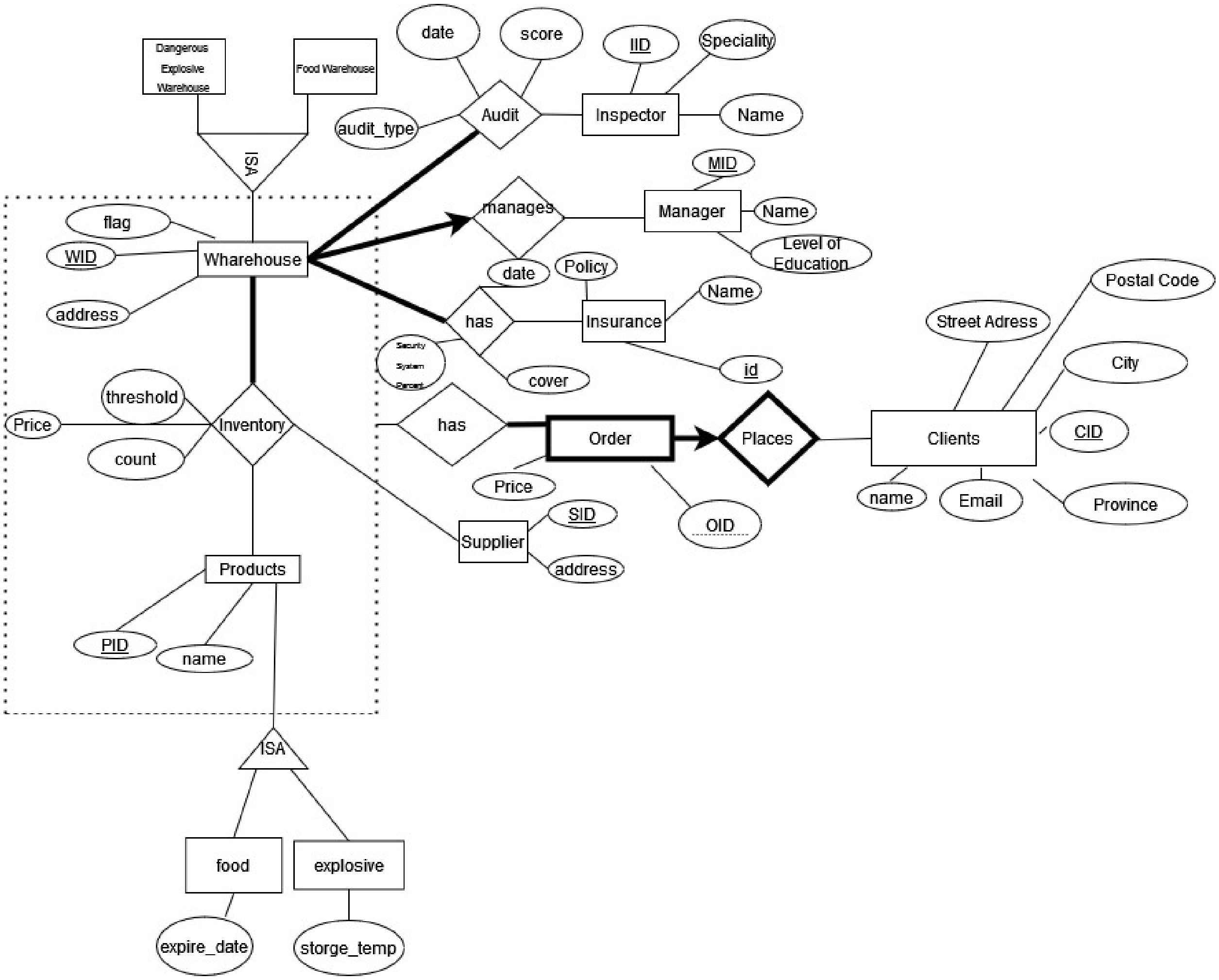
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 2, the main purpose of this page is for you to let us know your email address, 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

Section 2:

Our project is based on having a database that can keep track of amazon warehouses around canada, where each warehouse can have unique suppliers from the same products, and in which clients can order their favorite products online sometimes from different warehouses.

Section 3:



Our mentor TA suggested modifying the ternary relationship between supplier, warehouse, and products based on the fact that supplier was also connected to products to a relationship of supplies, in this case we decided to change by eliminating this relationship altogether as it as he said creates redundant information. Apart from that we made some changes to the statistics to represent the aggregation of order over the ternary relationship and specify far more the client address as is the piece of information that most importantly represents where the product is going to be sended. Some attributes have changed for some entities as we considered there were many addresses that were not that meaningful to be represented or part of the diagram, for instance Inspector now has a speciality rather than just an address which makes the information more useful and eliminates unnecessary information

Section 4.

1. Inventory(WID(FK, PK): int, PID(FK, PK): int, SID(FK, PK): int, threshold: int, count: int, price:int)
2. warehouse(WID(PK): int, MID(FK): int, address: varchar(255), flag: int) //comment: flag 0,1,2 to identify type of warehouse
3. product(PID(PK): int,name: varchar(255))
4. food(PID(FK,PK): int, shelf_life: int)
5. explosive(PID(FK,PK): int, storge_temp: int)
6. supplier(SID(PK): int, address: varchar(255))
7. client(CID(PK): int, name: varchar(255), street_address: varchar(255), postal_code: varchar(255), city:varchar(255), province:varchar(250), email:varchar(250))
8. Order(OID(PK): int, CID(FK,PK): int, price: int)
9. ItemsOrder(OID(PK, FK);int, PID(FK):int, WID:int,SID(PK, FK):int,CID(PK, FK):int quantity: int)
10. Inspector(InID(PK): int, name: varchar(255), scpeciliaty: int)//comment: flag 0,1,2… to identify type of inspector
11. Manager(MID(PK): int, name: varchar(255),LevelofEducation: int)
12. Insurance(IID(PK): int, name:varchar(255), policy: text)
13. InsurancePerWarehouses(IID(FK,PK):int, WID(PK, FK):int, date:date(PK), cover:in, SecuritySysemPercent: int)
14. Audit(WID(FK, PK): int, IID(FK,PK): int, AUDIT_TYPE(PK): int, date: date(PK), score: int)

Section 5.

1）WID, PID, SID ⟶ threshold, count, price

2）WID ⟶ address, flag, MID AND address ⟶ WID

3）PID ⟶ name

4）PID ⟶ shelf_life

5）PID ⟶ storge_temp

6）SID ->address

7）CID -> street address, postal code, city, province email, name

Postal code -> city, province

City -> Province

Postal code -> city

8） OID, CID -> price

9） OID, PID, WID, CID SID -> quantity

10）IID -> Speciality, name

11) MID -> name LevelofEducation

12) IID-> name policy

13) IID, WID-> date cover

SecuritySysemPercent-> cover

14) WID, IID, AUDIT_TYPE -> date score

Section 6.

In this case some of the relationship do not happen to be in 3NF or BCNF in so we are going to tackle the one that do not happen to fall all on this and do the decomposition

7）client(CID(PK): int, name: varchar(255), street_address: varchar(255), postal_code: varchar(255), city:varchar(255), province:varchar(250), email:varchar(250))

CID -> street address, postal code, city, province email, name

Postal code -> city, province

City -> Province

The only candidate key for this relation is {CID}

{CID}+ = {street address, postal code, city, province email, name, CID}

{Postal Code}+ = {Postal Code, city, province}

{City}+ = {City, Province}

In this case, Postal code -> city, province, City -> Province, Postal code -> city
Is clearly violating the criteria for BCNF because the determinant for all of those functional dependencies is not a superkey based on the closure of all of those determinants shown above. Moreover, the relation is also not in 3NF because for all of those functional dependencies mentioned above that do not satisfy BCNF, their RHS, {city, province} in "Postal code -> city, province", { province} in
" City -> Province", and {city} in " Postal code -> city", all are not a subset of the candidate key {CID}, therefore the relation is also not in 3NF.

Decompose into BCNF:

Clients(CID,street address, postal code, city, province email, name, CID)

Decompose using "Postal code -> city, province"

R1 = {Postal code, city, province}, R2 = {street address, postal code, email, name, CID}

R1 is not in BCNF because City still determines province in R1.

->**decompose R1 using City -> Province**

R3 = {City, Province}, R4 = {Postal Code, City}

-> **R3 and R4 is in BCNF because all relations that have two attributes are automatically in BCNF**

-> R2 is in BCNF because all of the fds either don't hold or have determinant as the superkey for R2.

-> therefore, our final solution is

R2 ={street address, postal code, email, name, CID}

R3 = {City, Province}

R4 = {Postal Code, City}

Clients(CID(PK): int, name: varchar(255), street_address: varchar(255), postal_code: varchar(255), email:varchar(250))
Clients3(postal_code(PK, FK): varchar(255), city:varchar(255))
Client2(city(PK, FK):varchar(255), province:varchar(250))

13) IID WID-> date cover SecuritySysemPercent
SecuritySysemPercent-> cover
In this case SecuritySysemPercent-> cover is a non-trivial relationship where SecuritySysemPercent is not a primary key so it is not in BCNF.
The key in this relationship is IID WID as shown by the cover
{IID, WID}+ ={IID, WID, data, cover, SecuritySysemPercent}

This decompose: into
R1(WID, IID, Date, SecuritySysemPercent)
R2(SecuritySysemPercent, Cover)
So for that we end with two tables
Insurances(WID(FK):int, IID(FK):int, date:date, SecuritySysemPercent: int)
Covers(SecuritySysemPercent(FK), Cover:int)

So with all thai the resulting tables are:

1. Inventory(WID(FK, PK): int, PID(FK, PK): int, SID(FK, PK): int, threshold: int, count: int, price:int)
2. warehouse(WID(PK): int, MID(FK): int, address: varchar(255), flag: int) //comment: flag 0,1,2 to identify type of warehouse
3. product(PID(PK): int,name: varchar(255))
4. food(PID(FK,PK): int, shelf_life: int)
5. explosive(PID(FK,PK): int, storge_temp: int)
6. supplier(SID(PK): int, address: varchar(255))
7. Clients(CID(PK): int, name: varchar(255), street_address: varchar(255), postal_code: varchar(255), email:varchar(250))
8. Clients3(postal_code(PK, FK): varchar(255), city:varchar(255))

9. Client2(city(PK, FK):varchar(255), province:varchar(250))
10. Order(OID(PK): int, CID(FK,PK): int, price: int)
11. ItemsOrder(OID(PK, FK);int, PID(FK):int, WID:int,SID(PK, FK):int,CID(PK, FK):int quantity: int)
12. Inspector(InID(PK): int, name: varchar(255), scpeciliaty: int)//comment: flag 0,1,2… to identify type of inspector
13. Manager(MID(PK): int, name: varchar(255),LevelofEducation: int)
14. Insurance(IID(PK): int, name:varchar(255), policy: text)
15. Insurances(WID(FK):int, IID(FK):int, date:date, SecuritySysemPercent: int)
16. Covers(SecuritySysemPercent(FK), Cover:int)
17. Audit(WID(FK, PK): int, IID(FK,PK): int, AUDIT_TYPE(PK): int, date: date(PK), score: int)

Section 7:

```
CREATE TABLE manager (
 MID int PRIMARY KEY,
 name varchar(255) NOT NULL,
 level_edu int NOT NULL,
 CONSTRAINT edu check (level_edu >= 0 AND level_edu <=4)
 );

CREATE TABLE warehouse (
 WID int PRIMARY KEY,
 address varchar(255) NOT NULL,
 MID int,
 flag int,
 CONSTRAINT MID_con FOREIGN KEY (MID)
  REFERENCES manager(MID),
 CONSTRAINT warehouse_type check (flag >=0 and flag <=2)
 );

CREATE TABLE product (
 PID int PRIMARY KEY,
 name varchar(255) NOT NULL
 );

CREATE TABLE supplier (
 SID int PRIMARY KEY,
 address varchar(255) NOT NULL
 );

CREATE TABLE inventory (
 WID int,
 PID int,
 SID int,
 threshold int NOT NULL,
 item_count int NOT NULL,
 price decimal(15,2),
 CONSTRAINT price check (price >0),
 CONSTRAINT wid FOREIGN KEY (WID)
  REFERENCES warehouse(WID),
 CONSTRAINT pid FOREIGN KEY (PID)
  REFERENCES product(PID),
 CONSTRAINT sid_1 FOREIGN KEY (SID)
  REFERENCES supplier(SID),
 CONSTRAINT pk PRIMARY KEY (WID,PID)
 );

CREATE TABLE food (
 PID int PRIMARY KEY,
 shelf_life int NOT NULL,
 CONSTRAINT shelf_life check (shelf_life >0),
 CONSTRAINT pid_2 FOREIGN KEY (PID)
```

```
    REFERENCES product(PID)
  );

CREATE TABLE explosive (
  PID int PRIMARY KEY,
  storage_temp int NOT NULL,
  CONSTRAINT pid_3 FOREIGN KEY (PID)
    REFERENCES product(PID)
  );

CREATE TABLE client (
  CID int PRIMARY KEY,
  name varchar(255) NOT NULL,
  street_address varchar(255) NOT NULL,
  postal_code varchar(255) NOT NULL UNIQUE,
  email varchar(255) NOT NULL,
  city varchar(255) NOT NULL,
  province varchar(255) NOT NULL
  );

  CREATE TABLE Client3 (
  postal_code varchar(255) PRIMARY KEY,
  city varchar(255) NOT NULL UNIQUE,
 FOREIGN KEY (postal_code) REFERENCES client(postal_code)
);

CREATE TABLE Client2 (
  city varchar(255) PRIMARY KEY,
  province varchar(255) NOT NULL,
  FOREIGN KEY (city) REFERENCES Client3(city)
);
```

```
CREATE TABLE orders (
  OID int,
  CID int,
  price int NOT NULL,
  CONSTRAINT cid_1 FOREIGN KEY (CID)
    REFERENCES client(CID),
  CONSTRAINT pk_1 PRIMARY KEY (OID,CID),
  CONSTRAINT price check (counts > 0)
  );

CREATE TABLE inspector (
  InID int PRIMARY KEY,
  name varchar(255) NOT NULL,
  scpeciliaty int NOT NULL,
  CONSTRAINT scpeciliaty check (scpeciliaty >=0 and scpeciliaty <=2)
  );

CREATE TABLE ItemsOrder (
  OID int,
  CID int,
  WID int,
  PID int,
  SID int,
  quantity int,
  CONSTRAINT cid_9 FOREIGN KEY (CID)
    REFERENCES client(CID),
  CONSTRAINT wid_6 FOREIGN KEY (WID)
    REFERENCES warehouse(WID),
  CONSTRAINT sid_11 FOREIGN KEY (SID)
    REFERENCES supplier(SID),
  CONSTRAINT pid_6 FOREIGN KEY (PID)
    REFERENCES product(PID),
  CONSTRAINT pk_1 PRIMARY KEY (OID,CID),
  CONSTRAINT nums check (quantity > 0)
  );

CREATE TABLE insurance (
  IID int PRIMARY KEY,
  policy text,
  name varchar(255)
  );

create TABLE Insurances(
WID int,
IID int,
dates date,
SecuritySysemPercent int NOT NULL UNIQUE,
constraint wid_8 FOREIGN KEY (WID)
references warehouse(WID),
constraint idd_8 FOREIGN KEY (IID)
references insurance(IID),
  Primary KEY( dates, IID, WID),
CONSTRAINT SecuritySysemPercent check (SecuritySysemPercent >=0 and
SecuritySysemPercent <=100)
);

create TABLE covers(
  SecuritySysemPercent int,
  Cover int,
  constraint SecuritySysemPercent foreign key(SecuritySysemPercent)
   references insurances(SecuritySysemPercent),
   PRIMARY KEY(SecuritySysemPercent),
CONSTRAINT Cover check (Cover >=0 and Cover <=100)
);


CREATE TABLE audit (
  WID int,
  InID int,
  audit_type int,
  audit_date date,
  score int,
  CONSTRAINT pk_2 PRIMARY KEY (WID,InID,audit_type,audit_date),
  CONSTRAINT InID FOREIGN KEY (InID)
    REFERENCES inspector(InID),
  CONSTRAINT score check (score>=0 and score<=100)
  );
```

Section 8.

```
INSERT INTO manager (MID, name, level_edu)
VALUES
  (1, 'John Doe', 3),
  (2, 'Alice Smith', 4),
  (3, 'Bob Johnson', 2),
  (4, 'Eva Williams', 1),
  (5, 'Jane Doe', 1);

INSERT INTO warehouse (WID, address, MID, flag)
```

```
VALUES (1, '123 Main St', 1, 0),
       (2, 'UBC Main St', 2, 2),
       (3, '246 Hillcrest Ave', 3, 0),
       (4, 'Granville st', 4, 1),
       (5, 'XXX Mall', 5, 2);

INSERT INTO product (PID, name)
VALUES (1, 'coke'),
       (2, 'instant noodle'),
       (3, 'chips'),
       (4, 'juice'),
       (5, 'laptop'),
       (6, 'beef'),
       (7, 'alcohol'),
       (8, 'match'),
       (9, 'lighter'),
       (10, 'hydrogen tank');


INSERT INTO supplier (SID, address)
VALUES (1, '123 Main St'),
       (2, 'UBC Main St'),
       (3, '246 Hillcrest Ave'),
       (4, 'Granville St'),
       (5, 'XXX Mall');

INSERT INTO explosive (PID, storage_temp)
VALUES (5, 20),
       (7, 20),
       (8, 20),
       (9, 20),
       (10, 15);

INSERT INTO inventory (WID, PID, SID, threshold, item_count, price)
VALUES (1, 1, 1, 50, 200, 2.99),
       (1, 2, 2, 50, 200, 2.99),
       (1, 3, 3, 50, 200, 2.99),
       (1, 4, 4, 50, 200, 5.99),
       (1, 5, 5, 10, 40, 2000);

INSERT INTO insurance (IID, policy, name)
VALUES (1, 'some_policy', 'Combo1'),
       (2, 'some_policy_1', 'Combo2'),
       (3, 'some_policy_2', 'Combo3'),
       (4, 'some_policy_3', 'Combo4'),
       (5, 'some_policy_4', 'Combo5');

INSERT INTO inspector (InID, name, scpeciliaty)
VALUES (1, 'John Smith', 1),
       (2, 'Alice Smith', 1),
       (3, 'John Doe', 1),
       (4, 'Bob Johnson', 1),
       (5, 'Jane Doe', 1);

INSERT INTO audit (WID, InID, audit_type, audit_date, score)
VALUES (1, 1, 0, '2023-10-18', 90),
       (1, 2, 1, '2023-10-18', 100),
       (1, 3, 2, '2023-10-18', 60),
       (2, 1, 0, '2023-9-18', 80),
       (2, 2, 1, '2023-9-18', 94);

INSERT INTO Insurances (WID, IID, dates, SecuritySysemPercent)
```

```
VALUES (1, 1, '2023-10-18', 80),
    (1, 2, '2023-10-18', 99),
    (1, 3, '2023-10-18', 100),
    (1, 4, '2023-10-18', 40),
    (1, 5, '2023-10-18', 07);

INSERT INTO covers (SecuritySysemPercent, Cover)
VALUES (80, 20),
    (99, 0),
    (100, 0),
    (40, 60),
    (07, 100);

INSERT INTO client (CID, name, street_address, postal_code, email)
VALUES (1, 'John Doe', '123 Main St', 'V9P', 'john.doe@example.com'),
    (2, 'Alice Smith', '1234 Granvile St', 'V6R', 'alice.smith@example.com'),
    (3, 'Bob Johnson', '4th Main St', 'V3A', 'bob.johnson@example.com'),
    (4, 'Eva Williams', '798 Parksville St', 'V5C', 'eva.williams@example.com'),
    (5, 'Jane Doe', '1123 Hillcrest Ave', 'V9Z', 'jane.doe@example.com');

INSERT INTO Client3 (postal_code, city)
VALUES ('V9P', 'Vancouver'),
    ('V6R', 'Nanaimo'),
    ('V3A', 'Victoria'),
    ('V5C', 'Richmond'),
    ('V9Z', 'Ottowa');

INSERT INTO Client2 (city, province)
VALUES ('Vancouver', 'BC'),
    ('Nanaimo', 'BC'),
    ('Victoria', 'BC'),
    ('Richmond', 'BC'),
    ('Ottowa', 'ON');

INSERT INTO orders (OID, CID, price)
VALUES (1, 1, 100),
    (1, 2, 130),
    (1, 3, 260),
    (1, 4, 432),
    (1, 5, 121);
INSERT INTO ItemsOrder (OID, CID, WID, PID, SID, quantity)
VALUES (1, 1, 1, 1, 1, 10),
    (1, 2, 1, 2, 1, 15),
    (1, 3, 1, 1, 1, 20),
    (1, 4, 1, 1, 1, 7),
    (1, 5, 1, 1, 1, 3);
```