CPSC 304 Project Cover Page

Milestone #: 2

Date: July 27, 2022

Group Number: 22

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Luis Victoria	78827979	x1p2b	luisv@student.ubc.ca
Richard Chen	45564895	n4z2o	richard.chen@ualberta.ca
Yan Zhang	33283136	l8b3b	yyzhang@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

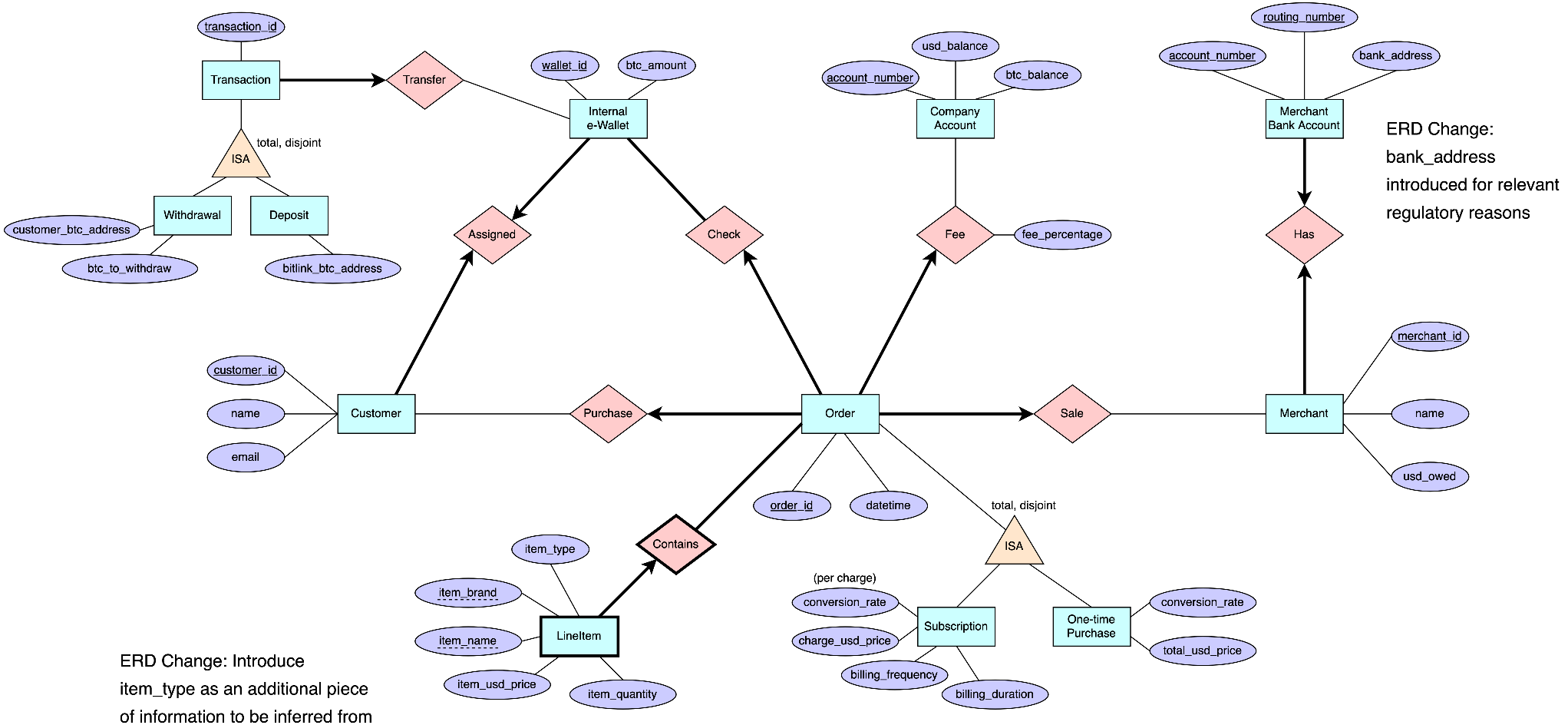
Milestone 2: Relational Schema, Normalization, SQL DDL, Tentative Queries in English

The aim is to do modelling using (a possibly refined) ER diagram from Milestone 1 and the resulting schema for your database. While working, you may go through several refinements but, you must only submit the final version.

Each group must provide the following as one PDF file:

1. **A completed cover page.**
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.**

(ER Diagram on next page)



ERD Change: Introduce item_type as an additional piece of information to be inferred from item_name

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, candidate key, foreign keys, and other constraints that the table must maintain.**

InternalEWallet(wallet_id: integer, btc_amount: double)
PK: wallet_id

Withdrawal(transaction_id: integer, customer_btc_address: string, btc_to_withdraw: double)
PK: transaction_id

Deposit(transaction_id: integer, bitlink_btc_address: string)
PK: transaction_id

Transaction(transaction_id: integer, **wallet_id**: integer)
PK: transaction_id
FK: wallet_id references InternalEWallet

Customer(customer_id: integer, **wallet_id**: integer, name: string, email: string)
PK: customer_id
FK: wallet_id references InternalEWallet
CKs: customer_id, email

Order(order_id: integer, **customer_id**: integer, **company_account_number**: integer, **merchant_id**: integer, **wallet_id**: integer, datetime: string, fee_percentage: double)
PK: order_id
FK: customer_id references Customer
company_account_number references CompanyAccount(account_number)
merchant_id references Merchant
wallet_id references InternalEWallet

Merchant(merchant_id: integer, **bank_account_number**: integer, name: string, usd_owed: double)
PK: merchant_id
FK: bank_account_number references MerchantBankAccount(account_number)

MerchantBankAccount(account_number: integer, routing_number: integer, bank_address: string)

PK: account_number, routing_number

CompanyAccount(account_number: integer, usd_balance: double, btc_balance: double)

PK: account_number

Subscription(order_id: integer, conversion_rate: double, charge_usd_price: double, billing_frequency: string, billing_duration: string)

PK: order_id

OnetimePurchase(order_id: integer, conversion_rate: double, total_usd_price: double)

PK: order_id

LineItem(**order_id**: integer, item_brand: string, item_name: string, item_usd_price: double, item_quantity: integer, item_type: string)

PK: order_id, item_brand, item_name

FK: order_id

4. **Functional Dependencies (FDs)**
    a. **Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key).**

Merchant Bank Account
account_number, routing_number → bank_address
routing_number → bank_address

Merchant
merchant_id → name, usd_owed

Order
order_id → datetime, conversion_rate, charge_usd_price, billing_frequency, billing_duration, conversion_rate, total_usd_price

LineItem
item_brand, item_name → item_use_price, item_quantity, item_type
item_name → item_type

Company Account
account_number → usd_balance, btc_balance

Internal e-Wallet
wallet_id → btc_amount

Transaction
transaction_id → customer_btc_address, btc_to_withdraw, bitlink_btc_address

Customer
customer_id → name, email
email → name, customer_id

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.**

Merchant Bank Account
account_number, routing_number → bank_address
routing_number → bank_address

Closures:
(account_number, routing_number)$^+$ = {account_number, routing_number, bank_address}
- account_number, routing_number is a superkey, and does not violate BCNF

routing_number$^+$ = {routing_number, bank_address }
- routing_number is not a superkey, so it does violate BCNF

Decompose:
R_1(routing_number, bank_address)
R_2(routing_number, account_number)

New table:
R_1: MerchantBankAddress(routing_number: integer, bank_address: string)
R_2: MerchantBankAccount(account_number: integer, **routing_number**: integer)

LineItem
item_brand, item_name → item_use_price, item_quantity, item_type
item_name → item_type

Closures:

(item_brand, item_name)$^+$ = { item_brand, item_name, item_usd_price, item_quantity, item_type }
- item_brand, item_name is a superkey and therefore does not violate BCNF

(item_name)$^+$ = {item_name, item_type}
- item_name is not a superkey, so it violates BCNF

Decompose:
R_1(item_name, item_type)
R_2(item_name, item_brand, item_usd_price, item_quantity)

New table:
R_1: LineItemType(**order_id**: integer, item_name: string, item_type: string)
R_2: LineItem(**order_id**: integer, item_brand: string, **item_name**: string, item_usd_price: double, item_quantity: integer)

Final Tables
InternalEWallet(wallet_id: integer, btc_amount: double)
PK: wallet_id

Withdrawal(transaction_id: integer, customer_btc_address: string, btc_to_withdraw: double)
PK: transaction_id

Deposit(transaction_id: integer, bitlink_btc_address: string)
PK: transaction_id

Transaction(transaction_id: integer, **wallet_id**: integer)
PK: transaction_id
FK: wallet_id references InternalEWallet

Customer(customer_id: integer, **wallet_id**: integer, name: string, email: string)
PK: customer_id
FK: wallet_id references InternalEWallet
CKs: customer_id, email

Order(order_id: integer, **customer_id**: integer, **company_account_number**: integer, **merchant_id**: integer, **wallet_id**: integer, datetime: string, fee_percentage: double)
PK: order_id
FK: customer_id references Customer
company_account_number references CompanyAccount(account_number)
merchant_id references Merchant
wallet_id references InternalEWallet

Merchant(merchant_id: integer, **bank_account_number**: integer, name: string, usd_owed: double)
PK: merchant_id
FK: bank_account_number references MerchantBankAccount(account_number)

MerchantBankAddress(routing_number: integer, bank_address: string)
PK: routing_number

MerchantBankAccount(account_number: integer, **routing_number**: integer)
PK: account_number, routing_number
FK: routing_number references MerchantBankAddress

CompanyAccount(account_number: integer, usd_balance: double, btc_balance: double)
PK: account_number

Subscription(order_id: integer, conversion_rate: double, charge_usd_price: double, billing_frequency: string, billing_duration: string)
PK: order_id

OnetimePurchase(order_id: integer, conversion_rate: double, total_usd_price: double)
PK: order_id

LineItemType(**order_id**: integer, item_name: string, item_type: string)
PK: order_id, item_name
FK: order_id references Order

LineItem(**order_id**: integer, item_brand: string, **item_name**: string, item_usd_price: double, item_quantity: integer)
PK: order_id, item_brand, item_name
FK: order_id references Order
item_name references LineItemType

6. **The SQL DDL to create all the tables in SQL. All primary keys and foreign keys must be declared appropriately. Code the SQL CREATE TABLE statements with the appropriate foreign keys, primary keys, UNIQUE constraints, etc.**

```
CREATE TABLE InternalEWallet(
        wallet_id INTEGER PRIMARY KEY,
        btc_amount DOUBLE DEFAULT 0
)

CREATE TABLE Withdrawal(
        transaction_id INTEGER PRIMARY KEY,
        customer_btc_address STRING NOT NULL,
        btc_to_withdraw DOUBLE NOT NULL
)

CREATE TABLE Deposit(
        transaction_id INTEGER PRIMARY KEY,
        bitlink_btc_address STRING NOT NULL
)

CREATE TABLE Transaction(
        transaction_id INTEGER PRIMARY KEY,
        wallet_id INTEGER NOT NULL,
        FOREIGN KEY wallet_id REFERENCES InternalEWallet
)

CREATE TABLE Customer(
        customer_id INTEGER PRIMARY KEY,
        wallet_id INTEGER NOT NULL,
        name STRING NOT NULL,
        email STRING NOT NULL
        FOREIGN KEY wallet_id REFERENCES InternalEWallet
        UNIQUE (wallet_id)
)
```

```
CREATE TABLE Order(
        order_id INTEGER PRIMARY KEY,
        customer_id INTEGER NOT NULL,
        company_account_number INTEGER NOT NULL,
        merchant_id INTEGER NOT NULL,
        wallet_id INTEGER NOT NULL,
        datetime STRING NOT NULL,
        fee_percentage DOUBLE NOT NULL
        FOREIGN KEY customer_id REFERENCES Customer
        FOREIGN KEY company_account_number REFERENCES
        CompanyAccount(account_number)
        FOREIGN KEY merchant_id REFERENCES Merchant
        FOREIGN KEY wallet_id REFERENCES InternalEWallet
)

CREATE TABLE Merchant(
        merchant_id  INTEGER PRIMARY KEY,
        bank_account_number: INTEGER NOT NULL,
        name STRING NOT NULL,
        usd_owed DOUBLE DEFAULT 0
        FOREIGN KEY (bank_account_number) REFERENCES
        MerchantBankAccount(account_number)
)


CREATE TABLE MerchantBankAddress(
        routing_number INTEGER PRIMARY KEY,
        bank_address STRING NOT NULL
)

CREATE TABLE MerchantBankAccount(
        account_number INTEGER,
        routing_number INTEGER,
        PRIMARY KEY (account_number, routing_number),
        FOREIGN KEY (routing_number) REFERENCES MerchantBankAddress
)

CREATE TABLE CompanyAccount(
        account_number INTEGER PRIMARY KEY,
        usd_balance DOUBLE DEFAULT 0,
        btc_balance DOUBLE DEFAULT 0
)
```

```
CREATE TABLE Subscription(
        order_id INTEGER PRIMARY KEY,
        conversion_rate DOUBLE NOT NULL,
        charge_usd_price DOUBLE NOT NULL,
        billing_frequency STRING NOT NULL,
        billing_duration STRING NOT NULL
)


CREATE TABLE OnetimePurchase(
        order_id INTEGER PRIMARY KEY,
        conversion_rate DOUBLE NOT NULL,
        total_usd_price DOUBLE NOT NULL
)

CREATE TABLE LineItemType(
        order_id INTEGER,
        item_name STRING,
        item_type STRING NOT NULL,
        PRIMARY KEY (order_id, item_name),
        FOREIGN KEY order_id REFERENCES Order
)

CREATE TABLE LineItem(
        order_id INTEGER,
        item_brand STRING NOT NULL,
        item_name STRING NOT NULL,
        item_usd_price DOUBLE DEFAULT 0,
        item_quantity INTEGER DEFAULT 1,
        PRIMARY KEY(order_id, item_brand, item_name),
        FOREIGN KEY (order_id) REFERENCES Order
        ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (item_name) REFERENCES LineItemType
)
```

7. **Populate each table with at least 5 tuples, and probably more so that you can issue meaningful queries later on. Show the instance of each relation after inserting the tuples.**

InternalEWallet

wallet_id	btc_amount
1	1204.98151203
2	0.426039413
3	53.18008000
4	3.14159265
5	0.02127135
6	0.71973031
7	1.61803576
8	1.00595284
9	69.42000000
10	1.01495492

Withdrawal

transaction_id	customer_btc_address	btc_to_withdraw
100853	3EktnHQD7RiAE6uzMj2ZifT9YgRrkSgzQX	0.27788752
174762	17jRd5WEzpNbmTePP4RpiKoZdXnGfsfg9F	0.95592348
188513	1FfdTR9QJKoqtthdrNsirSQaesRsMQ9vjH	0.03301199
215078	bc1qa5wkgaew2dkv56kfvj49j0av5nml45x9ek9hz6	0.00623485
347589	1JqDybm2nWTENrHvMyafbSXXtTk5Uv5QAn	2.10109377
450995	1FfdTR9QJKoqtthdrNsirSQaesRsMQ9vjH	1.00053213
672714	bc1qxjphxq8ge0yl77servnm0lcks9l9v0v793vudw	69.62348514
706517	bc1qxjphxq8ge0yl77servnm0lcks9l9v0v793vudw	52.13588780
891427	12TQuPHSPVk2gLhgud8ViNhxioVwbXdRuq	0.12228467
907615	3EktnHQD7RiAE6uzMj2ZifT9YgRrkSgzQX	0.06602803

Deposit

transaction_id	bitlink_btc_address
294852	bc1qva26rqlug5n93yuj9ec6er5ppthgn25jxx3746
318739	bc1q334y6udd2ylenw8p99mschghy7udttaeyp7prt
408297	bc1qva26rqlug5n93yuj9ec6er5ppthgn25jxx3746
428154	bc1q7ncn3k65v886slh2kydkk495p42seemph2u3ua
751813	bc1q334y6udd2ylenw8p99mschghy7udttaeyp7prt
768275	bc1qva26rqlug5n93yuj9ec6er5ppthgn25jxx3746
810838	bc1qva26rqlug5n93yuj9ec6er5ppthgn25jxx3746
876900	bc1q7ncn3k65v886slh2kydkk495p42seemph2u3ua
960088	bc1qva26rqlug5n93yuj9ec6er5ppthgn25jxx3746
962255	bc1q7ncn3k65v886slh2kydkk495p42seemph2u3ua

Transaction

transaction_id	**wallet_id**
100853	4
174762	8
188513	6
215078	5
294852	1
318739	5
347589	4
428154	9
450995	7
672714	1
706517	1
751813	10
768275	3
810838	2
876900	1
891427	2
907615	10
960088	9
962255	3

Customer

customer_id	**wallet_id**	name	email
0001	1	Luis Victoria	luisvictoriaperez@gmail.com
0002	2	Gittu George	ggeorg02@cs.ubc.ca
0003	3	Felix Kjellberg	tseries@pewdiepie.com
0004	4	Xi Jinping	xi@gov.cn
0005	5	John Johnson	familycompany@johnsonjohnson.net
0006	6	Mark Zuckerberg	zucc@fb.com
0007	7	Cristian Ronaldo	sewi@fcbarcelona.com
0008	8	Yan Zhang	yannycats@gmail.com
0009	9	Bob Bobson	bob@bobson.org
0010	10	Richard Chen	richard.chen.dev@gmail.com

Order						
order_id	**customer_id**	**company_account_number**	**merchant_id**	**wallet_id**	datetime	fee_percentage
255979	0001	100000001	08182001	1	January 31, 2022 19:30:22	1.00
301333	0001	100000002	03914325	1	February 3, 2022 2:12:01	1.00
449476	0002	100000005	56077439	2	February 8, 2022 1:59:01	1.00
486901	0007	100000001	03914325	7	February 26, 2022 8:40:43	1.25
527884	0001	100000002	08182001	1	March 14, 2022 9:43:51	1.25
554210	0009	100000001	03914325	9	May 4, 2022 7:07:07	1.25
610320	0003	100000002	08182001	3	May 4, 2022 10:31:24	1.25
677212	0001	100000003	08182001	1	May 9, 2022 16:14:18	1.25
731666	0006	100000004	56077439	6	May 30, 2022 13:07:06	1.25
759562	0009	100000005	56077439	9	July 19, 2022 11:22:35	0.75

MerchantBankAddress	
routing_number	bank_address
001000579	99 Bay Street, Commerce Court Toronto, ON, CA, M5L 1A2
021000021	270 Park Ave. New York City, NY, USA, 10017
026003269	Head office, Building No. 1, Fuxingmennei Dajie, Xicheng District Beijing, Beijing Province 100818
121000248	255 2ND AVE SOUTH, MINNEAPOLIS, MN, USA, 55479
251082615	PO BOX 90010, RICHMOND, VA, USA 23225

CompanyAccount		
account_number	usd_balace	btc_balance
100000001	84719124.33	0.031044
100000002	34781.52	3904.40129532
100000003	349135.31	41.04951
100000004	48998405.51	59.309083050303
100000005	95810.19	595910.03

MerchantBankAccount	
account_number	**routing_number**
00802310945689692	251082615
02612903616249413	251082615
13882483011116047	026003269
20225292047352033	001000579
33709827845421524	021000021
37090610204877798	001000579
46470188400731605	121000248
53998666010297774	026003269
61278412260913601	121000248
64105633525368893	251082615

Merchant			
merchant_id	**bank_account_number**	name	usd_owed
1000001	00802310945689692	Tesla	940882.49
1000002	46470188400731605	Facebook	4193.31
1000003	33709827845421524	McDonalds	532557.45
1000004	13882483011116047	Baidu	506028934.43
1000005	02612903616249413	Nordstrom	52062.43
1000006	53998666010297774	UBC	2346230.71
1000007	37090610204877798	Dell	269342.62
1000008	64105633525368893	Gamestop	598432.42

Subscription				
order_id	conversion_rate	charge_usd_price	billing_frequency	billing_duration
255979	38499.88	5.99	Monthly	1 Year
449476	44074.35	5.99	Bi-weekly	2 Years
527884	39678.13	7.99	Monthly	6 Months
610320	39676.81	19.99	Monthly	3 Months
759562	23401.63	49.99	Yearly	1 Year

LineItem				
order_id	item_brand	**item_name**	item_usd_price	item_quantity
255979	Netflix	Premium	5.99	1
301333	Aritzia	Purple Vest	89.99	1
449476	Fitbit	Training Plus	5.99	1
486901	Apple	Pencil	34.99	1
527884	Prequel	Premium Filters	7.99	1
554210	Starbucks	Frapuccino	9.99	1
554210	Starbucks	Water Bottle	3.99	2
554210	Starbucks	Espresso Shot	1.99	4
610320	Adobe	Creative Cloud	19.99	1
677212	Tesla	Plaid	139990	10
731666	Charming	Toilet Paper	0.83	24
759562	BCAA	Car Extended Warranty	199.99	1

OneTimePurchase		
order_id	conversion_rate	total_usd_price
301333	37303.90	89.99
486901	39130.90	34.99
554210	39676.81	420.69
677212	30065.63	4135.31
731666	31713.08	5859.93

LineItemType		
order_id	item_name	item_type
255979	Netflix Premium	Media
301333	Purple Vest	Clothing Retail
449476	Training Plus	Software
486901	Pencil	Retail
527884	Premium Filters	Premium Filters
554210	Frapuccino	Consumable Retail
554210	Water Bottle	Consumable Retail
554210	Espresso Shot	Consumable Retail
610320	Creative Cloud	Software
677212	Plaid	Transportation
731666	Toilet Paper	Commercial Retail
759562	Car Extended Warranty	Insurance