

centennial college | Winter 2016

COMP122 Introduction to database concepts section -004 project: 12 Parsecs delivery service

Samuel buensucesco, ostap hamarnyk, sarina luu, Kevin Ma, alexander sikalveski

2016

Table of Contents

[Mission statement 2](#_Toc448999781)

[Business Rules 2](#_Toc448999782)

[Entity Relationship Diagram 4](#_Toc448999783)

[DDL 4](#_Toc448999784)

[Table creation 4](#_Toc448999785)

[Constraints 6](#_Toc448999786)

[Triggers 6](#_Toc448999787)

# Mission statement

To provide deliveries faster, more reliable, more affordable, and more secure than other delivery companies. Maintain a financially strong company-with broad employee ownership-that provides a long-term competitive return to our customers.

# Business Rules

1. Each **customer** can order many shipments; each **shipment** is ordered by only 1 customer.
2. Each shipment is delivered by 1 **employee**; every employee delivers many shipments.
3. Each shipment generates 1 **invoice**; each invoice is generated by 1 shipment.
4. Every shipment originates from 1 **source**; each source provides many shipments.
5. Every shipment arrives in 1 **destination**; each destination receives many shipments.
6. Each shipment generates 1 **return\_shipment**; each return\_shipment is generated by 1 shipment.
7. Each shipment contains many packages; each **package** is part of 1 shipment.
8. Each package contains many hubinventorypackages; every **hubinventorypackage** belongs to 1 package.
9. Every hubinventorypackage is located in 1 hub; each hub contains many hubinventorypackages.
10. Each **hub** stations many employees; every employee is stationed at 1 hub.
11. Every package is recorded when it arrived and departs from a delivery hub. this is recorded as a hubinventorypackage
12. Every customer can view their package with their unique tracking number.
13. The price of a package is determined as:
    1. Default of $5 per package
    2. For each package heavier than 10 kg, charge an additional $0.80 per kilogram over 10 kg.
14. The package component of the price of an invoice is determined as:
    1. The summation of all package prices related to the shipment instance the invoice instance is related to.
15. The shipping component of the price of an invoice is determined as:
    1. Default: $10 Standard shipping using truck mode of transport
    2. Express shipping: $5 additional fee
    3. Airplane mode of transport: $25 additional
16. The grand total of the invoice is the summation of the shipping, package and tax components related to a shipment instance.
17. Total shipment weight cannot be greater than 250kg.
    1. Shipment weight is determined by summation of all package weights related to it.
    2. Consequently, any package cannot be greater than 250 kg as well.
18. Customers receive a cumulative discount. The discount applies to the grand total cost of shipment. The types of discounts are:
    1. If total number of packages for one shipment exceeds 5, an additional 10% discount is applied to the grand total cost of shipment per package after the 5th
    2. If the destination city is the same as the source city for a shipment
       1. And if there is a hub located within that same city, apply an additional 5% discount to total cost of shipment.
    3. If an express shipment that is transported via airplane arrives later than three days of the shipment’s departure date, apply a discount of 50% to the shipment’s grand total cost.
    4. The cumulative discount cannot be greater than 50%.
19. A shipment must be of either standard (‘s’) or express (‘e’) type.
20. A shipment’s mode of transportation must be of either truck (‘t’) or airplane (‘a’).
21. A customer’s gender must be male (‘m’) or female (‘f’).
22. The city of a hub must be either be ‘Toronto’ or ‘Calgary’ or ‘Montreal’ or ‘Vancouver’
23. A customer must be at least 18 years of age.
24. The volume of a package cannot exceed 300 cm3.
25. If the volume of a package exceeds 72 cm3 the mode of transportation for the shipment must be airplane.
26. A return shipment cannot be generated for a shipment if 31 days have passed since past the shipment arrived at the destination.
27. The employee who delivers a shipment must belong to the same city the source originates from.
28. A shipment can only be delivered by an employee if the employee is not shipping any other delivery during the delivery time of the new shipment.
    1. New shipment departure date cannot be between ANY departure date and arrival date for shipments belonging to employee. New shipment department date must be after a shipment arrival date and before a shipment department date for a shipment belonging to the employee.
29. 13% tax is applied to the summation of package and shipment fees.

# Entity Relationship Diagram



NOTE: See the attached MSVisio document for a closer look at the ERD.

# DDL

## Table creation

CREATE TABLE customer(

cus\_id INT NOT NULL PRIMARY KEY,

cus\_fname VARCHAR(30) NOT NULL,

cus\_gender CHAR(1),

cus\_lname VARCHAR(30) NOT NULL,

cus\_phone INT NOT NULL,

cus\_dob DATE NOT NULL,

cus\_email VARCHAR(30) NOT NULL,

cus\_city VARCHAR(20) NOT NULL,

cus\_address VARCHAR(50) NOT NULL,

);

CREATE TABLE hub(

hub\_id INT NOT NULL PRIMARY KEY,

hub\_city VARCHAR(20) NOT NULL,

hub\_address VARCHAR(20) NOT NULL

);

CREATE TABLE employee(

emp\_id int NOT NULL PRIMARY KEY,

ship\_id int NOT NULL,

hub\_id int NOT NULL,

emp\_license CHAR(1) NOT NULL

);

CREATE TABLE shipment(

ship\_id INT NOT NULL PRIMARY KEY,

cus\_id INT NOT NULL,

src\_id INT NOT NULL,

dest\_id INT NOT NULL,

ship\_type CHAR(1) NOT NULL,

ship\_tweight DECIMAL(5,2),

ship\_dep DATE,

ship\_arr DATE,

ship\_mode CHAR(1) NOT NULL

);

CREATE TABLE invoice (

inv\_id INT NOT NULL PRIMARY KEY,

ship\_id INT NOT NULL,

inv\_packcost DECIMAL(10,2) WITH DEFAULT 0,

inv\_shipcost DECIMAL(10,2),

inv\_disc DECIMAL(3,2) WITH DEFAULT 0,

inv\_tax DECIMAL(10,2),

inv\_gtotal DECIMAL (10,2)

);

CREATE TABLE package (

pack\_trackNo int NOT NULL PRIMARY KEY,

ship\_id int NOT NULL,

pack\_weight decimal(5,2),

pack\_height decimal(5,2),

pack\_length decimal(5,2),

pack\_width decimal(5,2),

pack\_price decimal(10,2),

CONSTRAINT fk\_packFromShip FOREIGN KEY (ship\_id) REFERENCES shipment(ship\_id)

);

CREATE TABLE hubInventoryPackage (

pack\_trackNo INT NOT NULL ,

hub\_id INT NOT NULL,

hubInv\_arr DATE NOT NULL,

hubInv\_dep DATE,

CONSTRAINT fk\_trackPack FOREIGN KEY (pack\_trackNo) REFERENCES package(pack\_trackNo),

CONSTRAINT fk\_trackHub FOREIGN KEY (hub\_id) REFERENCES hub(hub\_id),

PRIMARY KEY (pack\_trackNo,hub\_id)

);

CREATE TABLE return\_shipment(

rtn\_id INT NOT NULL PRIMARY KEY,

ship\_id INT,

rtn\_date DATE,

rtn\_reason VARCHAR(128),

CONSTRAINT fk\_returnship FOREIGN KEY (ship\_id)

REFERENCES shipment(ship\_id)

);

CREATE TABLE source(

src\_id INT NOT NULL PRIMARY KEY,

src\_city VARCHAR(20),

src\_adrss VARCHAR(50)

);

CREATE TABLE destination(

dest\_id int NOT NULL PRIMARY KEY,

dest\_city VARCHAR(20),

dest\_adrss VARCHAR(50)

);

## Constraints

ALTER TABLE employee ADD CONSTRAINT FOREIGN KEY (hub\_id) REFERENCES hub(hub\_id);

ALTER TABLE employee ADD FOREIGN KEY (ship\_id) REFERENCES shipment(ship\_id);

ALTER TABLE shipment ADD FOREIGN KEY (cus\_id) REFERENCES customer(cus\_id);

ALTER TABLE shipment ADD FOREIGN KEY (src\_id) REFERENCES source(src\_id);

ALTER TABLE shipment ADD FOREIGN KEY (dest\_id) REFERENCES destination(dest\_id);

ALTER TABLE invoice ADD CONSTRAINT fk\_inv\_shipid FOREIGN KEY (ship\_id) REFERENCES shipment(ship\_id);

ALTER TABLE customer ADD CONSTRAINT gender CHECK(cus\_gender = 'm' OR cus\_gender = 'f');

ALTER TABLE shipment ADD CONSTRAINT ship\_type CHECK(ship\_type = 'e' OR ship\_type = 's');

ALTER TABLE shipment ADD CONSTRAINT ship\_mode CHECK(ship\_mode = 't' OR ship\_mode = 'a');

ALTER TABLE hub ADD CONSTRAINT city CHECK (hub\_city = 'Toronto' OR hub\_city = 'Calgary' OR hub\_city = 'Montreal' OR hub\_city = 'Vancouver');

## Sequences

CREATE SEQUENCE ship\_seq;

CREATE SEQUENCE pack\_seq;

CREATE SEQUENCE hub\_seq;

CREATE SEQUENCE emp\_seq;

CREATE SEQUENCE cus\_seq;

CREATE SEQUENCE inv\_seq;

CREATE SEQUENCE src\_seq;

CREATE SEQUENCE rtn\_seq;

CREATE SEQUENCE dest\_seq;

## Triggers

CREATE OR REPLACE TRIGGER return\_trigger

BEFORE INSERT ON return\_shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN(days(n.rtn\_date) - (SELECT days(ship\_arr) from shipment WHERE n.ship\_id = ship\_id) > 31)

SIGNAL SQLSTATE '75004' SET MESSAGE\_TEXT = 'Due to return policy you cannot return product after 31 days';

CREATE OR REPLACE TRIGGER large\_package

AFTER INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN((n.pack\_height \* n.pack\_length \* n.pack\_width) > 72)

UPDATE shipment SET ship\_mode = 'a' WHERE n.ship\_id = ship\_id;

CREATE OR REPLACE TRIGGER pack\_oversized

BEFORE INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN((n.pack\_height \* n.pack\_length \* n.pack\_width) > 300)

SIGNAL SQLSTATE '75006' SET MESSAGE\_TEXT = 'According to company policies, the package is too large to be transported.';

-- Late Express Delivery

CREATE OR REPLACE TRIGGER late\_express\_discount

AFTER UPDATE OF ship\_arr ON shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN(n.ship\_type = 'e' AND n.ship\_mode = 'a' AND ((n.ship\_arr - n.ship\_dep) > 3))

UPDATE invoice SET inv\_disc = 0.5 WHERE n.ship\_id = ship\_id;

-- PACK\_PRICE total trigger

-- default of $5

-- additional $0.80/kg over 10 kg

CREATE OR REPLACE TRIGGER calculate\_total\_pack\_price

AFTER UPDATE OF pack\_weight ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN(n.pack\_weight > 10)

UPDATE package SET pack\_price = 5 + (0.8) \* (pack\_weight - 10) WHERE pack\_trackno = n.pack\_trackno

CREATE OR REPLACE TRIGGER calculate\_total\_pack\_price2

AFTER INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN(n.pack\_weight > 10)

UPDATE package SET pack\_price = 5 + (0.8) \* (pack\_weight - 10) WHERE pack\_trackno = n.pack\_trackno

CREATE OR REPLACE TRIGGER calculate\_total\_pack\_price3

AFTER INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN(n.pack\_weight <= 10)

UPDATE package SET pack\_price = 5 WHERE pack\_trackno = n.pack\_trackno;

CREATE OR REPLACE TRIGGER calculate\_total\_pack\_price4

AFTER UPDATE OF pack\_weight ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN(n.pack\_weight <= 10)

UPDATE package SET pack\_price = 5 WHERE pack\_trackno = n.pack\_trackno;

CREATE OR REPLACE TRIGGER taxes\_insert

AFTER INSERT ON invoice

REFERENCING NEW AS N

FOR EACH ROW

UPDATE invoice SET inv\_tax = (N.inv\_packcost + N.inv\_shipcost) \* 0.13 WHERE ship\_id = n.ship\_id;

CREATE OR REPLACE TRIGGER taxes\_update

AFTER UPDATE OF inv\_shipcost, inv\_packcost ON invoice

REFERENCING NEW AS N

FOR EACH ROW

UPDATE invoice SET inv\_tax = (N.inv\_packcost + N.inv\_shipcost) \* 0.13 WHERE ship\_id = n.ship\_id;

CREATE TRIGGER invoice\_grandtotal

AFTER UPDATE OF inv\_tax, inv\_disc ON invoice

REFERENCING NEW AS N

FOR EACH ROW

UPDATE invoice SET inv\_gtotal = (N.inv\_packcost + N.inv\_shipcost + N.inv\_tax) \* (1 - N.inv\_disc) WHERE ship\_id = n.ship\_id;

-- our company does not let customers update source/destination. Once they select it,

-- we deliver.

CREATE TRIGGER local\_delivery\_discount

AFTER INSERT ON shipment

REFERENCING NEW AS N

FOR EACH ROW

WHEN (((SELECT src\_city FROM source WHERE n.src\_id = src\_id) =

(SELECT dest\_city FROM destination WHERE n.dest\_id = dest\_id)) AND

(0 < (SELECT COUNT(\*) FROM hub WHERE hub\_city = (SELECT dest\_city FROM destination WHERE dest\_id = N.dest\_id))))

UPDATE invoice SET inv\_disc = inv\_disc + 0.05 WHERE ship\_id = N.ship\_id;

CREATE OR REPLACE TRIGGER dob\_trig

BEFORE INSERT ON customer

REFERENCING NEW AS n

FOR EACH ROW

WHEN (YEAR(n.cus\_dob) + 18 >= YEAR(current\_date))

SIGNAL SQLSTATE '75003' SET MESSAGE\_TEXT = 'Customer must be at least 18 years old';

CREATE OR REPLACE TRIGGER calculate\_ship\_total\_weight

AFTER INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

UPDATE shipment SET ship\_tweight = (SELECT SUM(pack\_weight) FROM package WHERE ship\_id = n.ship\_id)

-- Overweight Trigger - if shipment > 250 kg throw exception error

CREATE OR REPLACE TRIGGER determine\_overweight\_shipment

BEFORE INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN (n.pack\_weight + (SELECT ship\_tweight FROM shipment WHERE ship\_id = n.ship\_id) > 250)

SIGNAL SQLSTATE '75005' SET MESSAGE\_TEXT = 'Shipment cannot exceed 250 kg!';

-- give 10% discount per package over 5 packages

CREATE OR REPLACE TRIGGER discount\_10

AFTER INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

WHEN (5 < (SELECT COUNT(\*) FROM package WHERE n.ship\_id = ship\_id))

UPDATE INVOICE SET inv\_disc = inv\_disc + 0.10 WHERE ship\_id = n.ship\_id;

-- caps discount at 50%. We would go bankrupt if we allowed discount to increase infinitely

CREATE OR REPLACE TRIGGER discount\_ceiling

AFTER UPDATE OF inv\_disc ON invoice

REFERENCING NEW AS n

OLD AS o

FOR EACH ROW

WHEN(n.inv\_disc > 0.5)

UPDATE invoice set inv\_disc = o.inv\_disc where inv\_id = n.inv\_id;

CREATE TRIGGER express\_shipment

AFTER INSERT ON shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN (n.ship\_type = 'e')

INSERT INTO invoice(ship\_id, inv\_shipcost) VALUES(n.ship\_id, 15);

--This creates an automatic row in INVOICE for express shipments transported via airplane

CREATE TRIGGER extra\_airplane\_cost\_insert

AFTER INSERT ON shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN (n.ship\_mode = 'a' AND n.ship\_type = 'e')

INSERT INTO invoice(ship\_id, inv\_shipcost) VALUES(n.ship\_id, 40);

--This adds 25 to the ship\_cost when the shipment mode is airplane

CREATE TRIGGER extra\_airplane\_cost\_update

AFTER UPDATE ON shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN (n.ship\_mode = 'a')

UPDATE invoice SET inv\_shipcost = inv\_shipcost + 25 WHERE ship\_id = n.ship\_id;

--This creates an automatic row in INVOICE when the shipment row has SHIPMENT\_MODE is equal to 'T' and SHIPMENT\_TYPE 'S'

CREATE TRIGGER standard\_shipment

AFTER INSERT ON shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN (n.ship\_mode = 't' AND n.ship\_type = 's')

INSERT INTO invoice(ship\_id, inv\_shipcost) VALUES(n.ship\_id, 10);

CREATE TRIGGER calculate\_packcost

AFTER INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

UPDATE invoice

SET inv\_packcost = inv\_packcost + n.pack\_price

WHERE ship\_id = n.ship\_id;

CREATE OR REPLACE TRIGGER airplane\_delivery\_employee

BEFORE UPDATE OF ship\_id ON employee

REFERENCING NEW AS n

FOR EACH ROW

WHEN(('a' = (SELECT ship\_mode FROM shipment WHERE ship\_id = n.ship\_id)) AND

('a' = (SELECT emp\_license FROM employee WHERE emp\_id = n.emp\_id)))

SET n.ship\_id = ship\_id;

CREATE OR REPLACE TRIGGER busy\_employee

BEFORE INSERT ON shipment

REFERENCING NEW AS n

FOR EACH ROW

WHEN((SELECT ship\_id FROM shipment WHERE n.ship\_dep >= ship\_dep OR n.ship\_dep <= ship\_arr) =

(SELECT ship\_id FROM employee))

SIGNAL SQLSTATE '74010' SET MESSAGE\_TEXT = 'This employee is delivering another package on this day already';

CREATE OR REPLACE TRIGGER increase\_ship

NO CASCADE

BEFORE INSERT ON shipment

REFERENCING NEW AS n

FOR EACH ROW

SET n.ship\_id = NEXTVAL for ship\_seq;

CREATE OR REPLACE TRIGGER increase\_pack

NO CASCADE

BEFORE INSERT ON package

REFERENCING NEW AS n

FOR EACH ROW

SET n.pack\_trackno = NEXTVAL for pack\_seq;

CREATE OR REPLACE TRIGGER increase\_hub

NO CASCADE

BEFORE INSERT ON hub

REFERENCING NEW AS n

FOR EACH ROW

SET n.hub\_id = NEXTVAL FOR hub\_seq;

CREATE OR REPLACE TRIGGER increase\_emp

NO CASCADE

BEFORE INSERT ON employee

REFERENCING NEW AS n

FOR EACH ROW

SET n.emp\_id = NEXTVAL FOR emp\_seq;

CREATE OR REPLACE TRIGGER increase\_cus

NO CASCADE

BEFORE INSERT ON customer

REFERENCING NEW AS n

FOR EACH ROW

SET n.cus\_id = NEXTVAL FOR cus\_seq;

CREATE OR REPLACE TRIGGER increase\_inv

NO CASCADE

BEFORE INSERT ON invoice

REFERENCING NEW AS n

FOR EACH ROW

SET n.inv\_id = NEXTVAL FOR inv\_seq;

CREATE OR REPLACE TRIGGER increase\_src

NO CASCADE

BEFORE INSERT ON source

REFERENCING NEW AS n

FOR EACH ROW

SET n.src\_id = NEXTVAL FOR src\_seq;

CREATE OR REPLACE TRIGGER increase\_rtn

NO CASCADE

BEFORE INSERT ON return\_shipment

REFERENCING NEW AS n

FOR EACH ROW

SET n.rtn\_id = NEXTVAL FOR rtn\_seq;

CREATE OR REPLACE TRIGGER increase\_dest

NO CASCADE

BEFORE INSERT ON destination

REFERENCING NEW AS n

FOR EACH ROW

SET n.dest\_id = NEXTVAL FOR dest\_seq;