University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

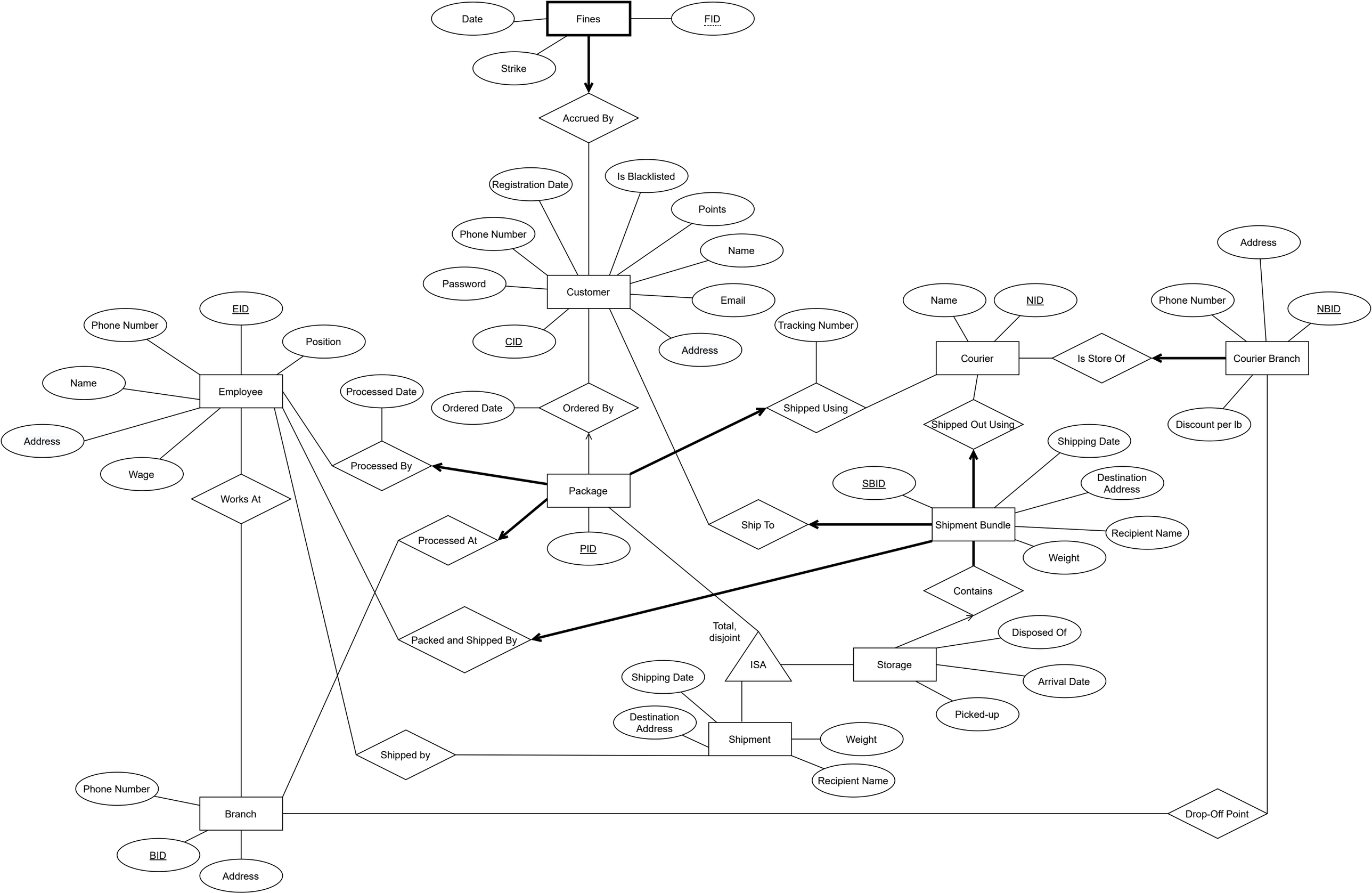
Date: March 4, 2021

Group Number: 72

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Emily Sun	32474158	a7f1b	emilyzqsun@gmail.com
Mellie Vo	48109722	w1e2b	hello@mellie.dev

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


Date
Fines
FID
Strike
Accrued By
Registration Date
Is Blacklisted
Points
Phone Number
Name
Password
Customer
Email
CID
Address
EID
Phone Number
Position
Name
Employee
Address
Wage
Works At
Processed Date
Processed By
Ordered Date
Ordered By
Package
Processed At
PID
Tracking Number
Shipped Using
Name
NID
Courier
Is Store Of
Address
Phone Number
NBID
Courier Branch
Discount per lb
Shipped Out Using
Shipping Date
SBID
Destination Address
Ship To
Shipment Bundle
Recipient Name
Weight
Contains
Total, disjoint
ISA
Disposed Of
Storage
Arrival Date
Picked-up
Packed and Shipped By
Shipping Date
Destination Address
Shipment
Weight
Recipient Name
Shipped by
Phone Number
Branch
BID
Address
Drop-Off Point

Changes to ER diagram from milestone 1

1. Removed Point of Sale entity to distance away from the blacklisted retail aspect.
2. Removed 'Membership' weak entity, instead adding its attributes to that of 'Customer' as the previous design had a 1-to-1 relationship between the two entities (with one being a weak entity), thus becoming redundant.
3. Added 'Courier Branch' entity which keeps track of branches of each courier in the vicinity of a branch of our project company. This addition allows keeping track of where each branch drops off packages to be shipped when they are using a particular courier.
4. Added the option for customers to order a 'Shipment Bundle' which describes a service where customers can choose for their packages that have arrived at and are stored at a branch to be packed together then shipped directly to the customer's address (or a single package to be shipped directly to them). So now the company has 3 main services:
    - storing parcels for customers to pick up in person
    - taking parcel(s) dropped off in-person and shipping them
    - acting as a middle point to receive parcels, then packaging them all together and shipping them to the customer (not in-person)
5. Added 'Fines' as a weak entity to 'Customer' to document whether a customer has violated any policies (such as keeping a package in storage and not picking up for more than 180 days, at which point the parcel is disposed of and customer is fined). When a customer repeatedly violates the policy then they are blacklisted.

Schema, Normalization, SQL DDL

*Global Constraints: Dates(including ArrivalDate, ShipDate, etc) are strings in the form of YYYY-MM-DD (ie. SQL Type DATE)

Employee(EID: int, PhoneNumber: int, Name: string, Address: string, Wage: float, Position: string)

Primary key: EID
Candidate key: {Name, PhoneNumber}, {Name, Address}
Foreign key: None

Functional Dependencies:
EID → PhoneNumber, Name, Address, Wage, Position
Name, PhoneNumber → EID, Address, Wage, Position
Name, Address → EID, PhoneNumber, Wage, Position
Position → Wage

Normalize:
This is currently in 2NF since it has no partial key dependency but Position is not a superkey nor part of a key. We normalize to BCNF and 3NF by replacing with the following schema

Employee(EID: int, PhoneNumber: int, Name: string, Address: string, **Position: string**)
Primary key: EID
Candidate key: {Name, PhoneNumber}, {Name, Address}
Foreign key: Position references PositionToWage

PositionToWage(Position: string, Wage: float)
Primary key: EID
Candidate key: {Name, PhoneNumber}, {Name, Address}
Foreign key: None

Branch(BID: int, Address: string, PhoneNumber: int)

Primary key: BID
Candidate key: Address, PhoneNumber
Foreign key: None

Functional Dependencies:
BID → Address, PhoneNumber
Address → BID, PhoneNumber
PhoneNumber → BID, Address

Normalize:
This is already in BCNF since each of BID, Address, and PhoneNumber is a superkey

---

WorksAt(**EID: int**, **BID: int**)
Primary key: {EID, BID}
Candidate key: None
Foreign key: EID references Employee, BID references Branch

Functional Dependencies: None

Normalize: This is already in BCNF and 3NF because it has no FDs and there are only 2 attributes

---

FinesAccruedBy(FID: int, **CID: int**, Date: string, Strike: int)
Primary key: {FID, CID}
Candidate key: None
Foreign key: CID references Customer
Other constraints: Strike $\in$ {1,2,3,4}

Functional Dependencies:
CID, FID $\rightarrow$ Date, Strike
Normalize:
This is already in BCNF and 3NF because {CID, FID} is a superkey and part of a minimal key

---

Customer(CID: int, Address: string, Email: string, Name: string, Points: int, RegistrationDate: string, PhoneNumber: int, Password: string, IsBlacklisted: boolean)
Primary key: CID
Candidate key: Email, {Name, Address}, {Name, PhoneNumber}
Foreign key: None

Functional Dependencies:
CID $\rightarrow$ Address, Email, Name, Points, Tier, RegistrationDate, PhoneNumber, IsBlacklisted
Email $\rightarrow$ CID, Address, Name, Points, Tier, RegistrationDate, PhoneNumber, Password, IsBlacklisted
Name, Address $\rightarrow$ CID, Email, Points, Tier, RegistrationDate, PhoneNumber, Password, IsBlacklisted
Name, PhoneNumber $\rightarrow$ CID, Address, Email, Points, Tier, RegistrationDate, Password, IsBlacklisted

Normalize:
This is already in BCNF and 3NF because each of CID, Email, {Name, Address}, {Name, PhoneNumber} is a superkey with no partial key dependencies, and are each part of a key

---

OrderedBy(**PID: int**, **CID: int**)
Primary key: CID
Candidate key: None
Foreign key: PID references PackageProcessedAtBranch, CID references customer

Functional Dependencies: None

This is already in BCNF and 3NF because it has no FDs and have only 2 attributes

---

PackageProcessedAtBranch(PID: int, **BID: int**)
Primary key: PID
Candidate key: None
Foreign key: BID references Branch

Functional Dependencies:
PID → BID

Normalize:
This is already in BCNF and 3NF because PID is a superkey and there are only 2 attributes

---

PackageProcessedByEmployee(**PID: int**, **EID: int,** ProcessedDate: string)
Primary key: PID
Candidate key: None
Foreign key: PID references PackageProcessedAtBranch, EID references Employee

Functional Dependencies:
PID → EID, ProcessedDate

Normalize:
This is already in BCNF because PID is a superkey

PackageShippedUsingCourier(**PID: int**, **NID: int**, TrackingNumber: string)
Primary key: PID
Candidate key: TrackingNumber
Foreign key: PID references PackageProcessedAtBranch, NID references Courier

Functional Dependencies:
PID → NID, TrackingNumber
TrackingNumber → PID, NID

Normalize:
This is already in BCNF because PID and TrackingNumber are both superkeys

---

Shipment(**PID: int**, ShippingDate: string, DestinationAddress: string, Weight: float, RecipientName: string)
Primary key: PID
Candidate key: None *
Foreign key: PID references PackageProcessedAtBranch
Other constraints: Weight ≤ 20kg
*Note: no candidate keys because multiple packages of the same weight may be delivered to the same recipient at the same location (for example sending products to a distribution center)

Functional Dependencies:
PID → ShippingDate, DestinationAddress, Weight, RecipientName

Normalize:
This is already in BCNF because PID is a superkey

ShipmentBundleShippedToCustomer(SBID: int, **CID: int**, Weight: float, RecipientName: string, DestinationAddress: string, ShippingDate: string)
Primary key: SBID
Candidate key: None*
Foreign key: CID references Customer
Other constraints: Weight ≤ 20kg
*Note: no candidate keys because multiple packages of the same weight may be delivered to the same recipient at the same location (for example sending products to a distribution center)

Functional Dependencies:
SBID → CID, Weight, RecipientName, DestinationAddress, ShippingDate

Normalize:
This is already in BCNF because SBID is a superkey

ShipmentBundleShippedOutUsingCourier(**SBID: int**, **NID: int**)
Primary key: {SBID, NID}
Candidate key: None
Foreign key: SBID references ShipmentBundleShippedToCustomer, NID references Courier

Functional Dependencies: None

Normalize: This is already in BCNF and 3NF because it has no FDs and there are only 2 attributes

---

ShipmentBundlePackedandShippedBy(**SBID: int**, **EID: int**)
Primary key: {SBID, EID}
Candidate key: None
Foreign key: SBID references ShipmentBundleShippedToCustomer, EID references Employee

Functional Dependencies: None

Normalize: This is already in BCNF and 3NF because it has no FDs and there are only 2 attributes

---

ShipmentBundleContainsStorage(**PID: int**, **SBID: int**, ArrivalDate: string, PickedUp: boolean, DisposedOf: boolean)
Primary key: {PID, SBID}
Candidate key: None
Foreign key: PID references PackageProcessedAtBranch, SBID references ShipmentBundleShippedToCustomer

Functional Dependencies:
PID, SBID → ArrivalDate, PickedUp, DisposedOf
ArrivalDate, PickedUp → DisposedOf

Normalize:
This is in 2NF because {ArrivalDate, PickedUp} is neither a superkey nor subset of a key, and it has no partial key dependency. We normalize into 2 tables to achieve BCNF.

ShipmentBundleContainsStorage(**PID: int**, **SBID: int**, **ArrivalDate: string**, **PickedUp: boolean**)
Primary key: {PID, SBID}
Candidate key: None
Foreign key: PID references PackageProcessedAtBranch, SBID references ShipmentBundleShippedToCustomer, {ArrivalDate, PickedUp} references ShipmentBundleToDisposeOf

ShipmentBundleToDisposeOf(ArrivalDate: string, PickedUp: boolean, DisposedOf: boolean)
Primary key: {ArrivalDate, PickedUp}
Candidate key: None
Foreign key: None

Courier(NID: int, Name: string)
Primary key: NID
Candidate key: Name
Foreign key: None

Functional Dependencies:
NID → Name

Normalize:
This is already in BCNF and 3NF because NID is a superkey and there are only 2 attributes

---

CourierBranchIsStoreOfCourier(NBID: int, **NID: int**, PhoneNumber: int, Address: string, DiscountPerLB: float)
Primary key: NBID

Candidate key: PhoneNumber, Address
Foreign key: NID references Courier

Functional Dependencies:
NBID → NID, PhoneNumber, Address, DiscountPerLB
PhoneNumber → NBID, NID, Address, DiscountPerLB
Address → NBID, NID, PhoneNumber, DiscountPerLB

Normalize:
This is already in BCNF because NBID, PhoneNumber, and Address are each a superkey

---

DropOffPoints(**NBID: int**, **BID: int**)

Primary key: {NBID, BID}
Candidate key: None
Foreign key: NBID references CourierBranchIsStoreOfCourier, BID references Branch

Functional Dependencies: None

Normalize: This is already in BCNF and 3NF because it has no FDs and there are only 2 attributes

---

ShippedBy(**EID: int**, **PID: int**)

Primary key: {EID, PID}
Candidate key: None
Foreign key: EID references Employee, PID references PackageProcessedAtBranch

Functional Dependencies: None

Normalize: This is already in BCNF and 3NF because it has no FDs and there are only 2 attributes

Tentative Queries

Insertion: Add a package to the list of packages processed at a particular branch

Deletion: Remove an employee from the list of workers at a particular branch

Update: Update whether a storage package (parcel delivered to branch and waiting for pick-up or shipment) has been picked up

Selection: Select Customers who ordered a package within the specified calendar year (input YYYY), and choose to show those who are now blacklisted or not (toggle IsBlacklisted)

Projection: Retrieve and return the Customer (CID), Shipment Bundle weight, and Shipment bundle shipping date attributes (from ShipmentBundleShippedToCustomer)

Join: Join the OrderedBy and PackageProcessedAtBranch tables to find the emails and names of customers who have had items processed at a particular branch

Aggregation with Group By: Find the total weight (sum) of shipment packages shipped on each day (ie. group by shipped date)

Aggregation with Having: Find the Courier Branches belonging to a particular Courier that give a minimum discount per lb of more than $1 (ie. aggregate min with group by Courier NID having discount > 1.0)

Nested Aggregation with Group By: Find the recipients who receive shipment packages with an average weight equal to the average weight of all shipped packages.

Division: Find Customers who have ordered packages that have been shipped using all the couriers.

Link to instances of tables (also attached below)

https://docs.google.com/spreadsheets/d/1Bc4tfmoMhhp_ECuq-G0y1q3slq2OmSSgg3bkObvz-38/edit?usp=sharing

Link to SQL DDL (also attached below)

https://gist.github.com/MellieVo/ef54ffce229cfb52b259583b7d1f6efa

```
CREATE DATABASE project;

USE project;

CREATE TABLE Employees
(
      `eid` INT NOT NULL AUTO_INCREMENT,
    `phone_number` INT,
    `name` varchar(255) NOT NULL,
    `address` varchar(255) NOT NULL,
    `position` varchar(255) NOT NULL,
    PRIMARY KEY (`eid`),
      UNIQUE KEY
name_phone_number_unique(`name`,
`phone_number`),
     UNIQUE KEY
name_address_unique(`name`, `address`),
    FOREIGN KEY (`position`) REFERENCES
PositionToWage(`position`)
);

CREATE TABLE PositionToWage
(
      `position` varchar(255) NOT NULL,
    `wage` FLOAT NOT NULL DEFAULT 0,
    PRIMARY KEY (`position`)
);

CREATE TABLE Branches
(
      `bid` INT NOT NULL AUTO_INCREMENT,
    `address` varchar(255) NOT NULL,
    `phone_number` INT NOT NULL,
    PRIMARY KEY (`bid`),
    UNIQUE KEY (`address`),
    UNIQUE KEY (`phone_number`)
);

CREATE TABLE WorksAt
(
      `eid` INT NOT NULL,
    `bid` INT NOT NULL,
    PRIMARY KEY (`eid`, `bid`),
    FOREIGN KEY (`eid`) REFERENCES
Employees(`eid`),
    FOREIGN KEY (`bid`) REFERENCES
Branches(`bid`)
);

CREATE TABLE FinesAccruedBy
(
      `fid` INT NOT NULL AUTO_INCREMENT,
    `cid` INT NOT NULL,
    `date` DATE NOT NULL,
    `strike` INT NOT NULL,
    PRIMARY KEY (`fid`, `cid`),
    FOREIGN KEY (`cid`) REFERENCES
Customers(`cid`)
);

CREATE TABLE Customers
(
      `cid` INT NOT NULL AUTO_INCREMENT,
    `address` varchar(255) NOT NULL,
    `email` varchar(255) NOT NULL,
    `name` varchar(255) NOT NULL,
    `points` INT NOT NULL DEFAULT 0,
    `registration_date` DATE NOT NULL,
    `phone_number` INT NOT NULL,
    `password` BINARY(32) NOT NULL,
    `is_blacklisted` BOOL NOT NULL
DEFAULT false,
    PRIMARY KEY (`cid`),
      UNIQUE KEY (`email`),
      UNIQUE KEY
name_address_unique(`name`, `address`),
      UNIQUE KEY
name_phone_number_unique(`name`,
`phone_number`)
);

CREATE TABLE OrderedBy
(
      `pid` INT NOT NULL,
    `cid` INT NOT NULL,
    PRIMARY KEY (`cid`),
    FOREIGN KEY (`cid`) REFERENCES
Customers(`cid`),
    FOREIGN KEY (`pid`) REFERENCES
PackagesProcessedAtBranch(`pid`)
);

CREATE TABLE PackagesProcessedAtBranch
(
      `pid` INT NOT NULL AUTO_INCREMENT,
    `bid` INT NOT NULL,
    PRIMARY KEY (`pid`),
    FOREIGN KEY (`bid`) REFERENCES
Branches(`bid`)
);

CREATE TABLE PackagesProcessedByEmployee
(
      `pid` INT NOT NULL,
    `eid` INT NOT NULL,
    `processed_date` DATE NOT NULL,
    PRIMARY KEY (`pid`),
    FOREIGN KEY (`pid`) REFERENCES
PackagesProcessedAtBranch(`pid`),
    FOREIGN KEY (`eid`) REFERENCES
Employees(`eid`)
);
```

```
CREATE TABLE PackagesShippedUsingCourier
(
      `pid` INT NOT NULL,
    `nid` INT NOT NULL,
    `tracking_number` varchar(36) NOT
NULL,
    PRIMARY KEY (`pid`),
    UNIQUE KEY (`tracking_number`),
    FOREIGN KEY (`pid`) REFERENCES
PackagesProcessedAtBranch(`pid`),
    FOREIGN KEY (`nid`) REFERENCES
Couriers(`nid`)
);

CREATE TABLE Shipment
(
      `pid` INT NOT NULL,
    `shipping_date` DATE NOT NULL,
    `destination_address` varchar(255)
NOT NULL,
    `weight` FLOAT NOT NULL DEFAULT 0,
    `recipient_name` varchar(255) NOT
NULL,
    PRIMARY KEY (`pid`),
    FOREIGN KEY (`pid`) REFERENCES
PackagesProcessedAtBranch(`pid`)
);

CREATE TABLE
ShipmentBundlesShippedToCustomer
(
      `sbid` INT NOT NULL
AUTO_INCREMENT,
    `cid` INT NOT NULL,
    `weight` FLOAT NOT NULL,
    `recipient_name` varchar(255) NOT
NULL,
    `destination_address` varchar(255)
NOT NULL,
    `shipping_date ` DATE NOT NULL,
    PRIMARY KEY (`sbid`),
    FOREIGN KEY (`cid`) REFERENCES
Customers(`cid`)
);

CREATE TABLE
ShipmentBundlesShippedOutUsingCourier
(
      `sbid` INT NOT NULL,
    `nid` INT NOT NULL,
    PRIMARY KEY (`sbid`, `nid`),
    FOREIGN KEY (`sbid`) REFERENCES
ShipmentBundlesShippedToCustomer(`sbid`)
,
    FOREIGN KEY (`nid`) REFERENCES
Couriers(`nid`)
);

CREATE TABLE
ShipmentBundlesPackedandShippedBy
(
      `sbid` INT NOT NULL,
    `eid` INT NOT NULL,
    PRIMARY KEY (`sbid`, `eid`),
    FOREIGN KEY (`sbid`) REFERENCES
ShipmentBundlesShippedToCustomer(`sbid`)
,
    FOREIGN KEY (`eid`) REFERENCES
Employees(`eid`)
);

CREATE TABLE
ShipmentBundlesContainsStorage
(
      `pid` INT NOT NULL,
    `sbid` INT NOT NULL,
    `arrival_date` DATE NOT NULL,
    `picked_up` BOOL NOT NULL,
    PRIMARY KEY (`pid`, `sbid`),
    FOREIGN KEY (`pid`) REFERENCES
PackagesProcessedAtBranch(`pid`),
    FOREIGN KEY (`sbid`) REFERENCES
ShipmentBundlesShippedToCustomer(`sbid`)
,
    FOREIGN KEY (`arrival_date`,
`picked_up`) REFERENCES
ShipmentBundlesToDisposeOf(`arrival_date
`, `picked_up`)
);

CREATE TABLE ShipmentBundlesToDisposeOf
(
    `arrival_date` DATE NOT NULL,
    `picked_up` BOOL NOT NULL DEFAULT
false,
    `disposed_of` BOOL NOT NULL DEFAULT
false,
    PRIMARY KEY (`arrival_date`,
`picked_up`)
);

CREATE TABLE Couriers
(
      `nid` INT NOT NULL AUTO_INCREMENT,
    `name` VARCHAR(255) NOT NULL,
    PRIMARY KEY (`nid`)
);
```

```
CREATE TABLE
CourierBranchsIsStoreOfCourier
(
      `nbid` INT NOT NULL
AUTO_INCREMENT,
    `nid` INT NOT NULL,
    `phone_number` INT NOT NULL,
    `address` VARCHAR(255) NOT NULL,
    `discount_per_lb` FLOAT NOT NULL
DEFAULT 0,
    PRIMARY KEY (`nbid`),
    UNIQUE KEY (`phone_number`),
    UNIQUE KEY (`address`)
);

CREATE TABLE DropOffPoints
(
      `nbid` INT NOT NULL,
    `bid` INT NOT NULL,
    PRIMARY KEY (`nbid`, `bid`),
    FOREIGN KEY (`nbid`) REFERENCES
CourierBranchsIsStoreOfCourier(`nbid`),
    FOREIGN KEY (`bid`) REFERENCES
Branches(`bid`)
);

CREATE TABLE ShippedBy
(
      `eid` INT NOT NULL,
    `pid` INT NOT NULL,
    PRIMARY KEY (`eid`, `pid`),
    FOREIGN KEY (`eid`) REFERENCES
Employees(`eid`),
    FOREIGN KEY (`pid`) REFERENCES
PackagesProcessedAtBranch(`pid`)
);

/* ------------------------------------
-----------------------------------
 * Migrations Table
 *
 * NOTE: This helps with ensuring that
we only apply migrations if we need to
 *       Checking for this table isn't
needed in this script as if the database
 *       the database exists, this
script has been run and re-creating the
db
 *       will result in an error
 * ------------------------------------
----------------------------------- */
CREATE TABLE Migrations
(
      `migration_id` INT NOT NULL,
    PRIMARY KEY (`migration_id`)
);

INSERT INTO Migrations(`migration_id`)
VALUES (0)
```

eid	phone_number	name	address	position
0	6040000000	Emp0	Addr0	employee
1	6040000001	Emp1	Addr1	cashier
2	6040000002	Emp2	Addr2	owner
3	6040000003	Emp3	Addr3	packer
4	6040000004	Emp4	Addr4	shipper

position	wage
employee	12
cashier	12
owner	50
packer	13
shipper	13

bid	address	phone_number
0	Addr5	6040000005
1	Addr6	6040000006
2	Addr7	6040000007
3	Addr8	6040000008
4	Addr9	6040000009

eid	bid
0	0
1	1
2	2
3	3
4	4

fid	cid	date	strike
0	0	2020-06-02	1
1	1	2020-06-02	1
2	2	2020-06-02	1
3	3	2020-06-02	2
4	4	2020-06-02	4

cid	address	email	name	points	registration_date	phone_number	password	is_blacklisted
0	Address10	example0@example.com	Cust0	100	2019-03-04	6040000010	5FECEB66FFC86F38D952786C6D696C79C2DBC239DD4E91B46729D73A27FB57E9	FALSE
1	Address11	example1@example.com	Cust1	100	2019-03-05	6040000010	6B86B273FF34FCE19D6B804EFF5A3F5747ADA4EAA22F1D49C01E52DDB7875B4B	FALSE
2	Address12	example2@example.com	Cust2	100	2019-03-06	6040000010	D4735E3A265E16EEE03F59718B9B5D03019C07D8B6C51F90DA3A666EEC13AB35	FALSE
3	Address13	example3@example.com	Cust3	200	2019-03-07	6040000010	4E07408562BEDB8B60CE05C1DECFE3AD16B72230967DE01F640B7E4729B49FCE	FALSE
4	Address14	example4@example.com	Cust4	400	2019-03-08	6040000010	4B227777D4DD1FC61C6F884F48641D02B4D121D3FD328CB08B5531FCACDABF8A	TRUE

pid	cid
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	3
8	4
9	4
10	4
11	4

pid	bid
0	0
1	0
2	0
3	1
4	1
5	1
6	2
7	2
8	2
9	3
10	4
11	4

pid	eid	processed_date
0	0	2020-03-04
1	0	2020-03-04
2	0	2020-03-04
3	1	2020-03-04
4	1	2020-03-04
5	1	2020-03-04
6	2	2020-03-04
7	2	2020-03-04
8	2	2020-03-04
9	3	2020-03-04
10	4	2020-03-04
11	4	2020-03-04

pid	nid	tracking_number
0	0	451fbf3d-0840-4e45-8f46-26c75ac3a0e5
1	0	f69b2cbf-6e10-4ae0-a8ee-89f5dd6d0873
2	0	d2408f4f-0fc8-4e62-85b1-2f79e97c0253
3	1	22a53fa6-1d55-4a62-a8d4-141622441d83
4	1	1823880d-81f8-404f-a131-76cf8814d93a
5	1	87e0dd7f-585a-4a9d-9f4c-1735eb505be6
6	2	1c7f9b01-2352-45f6-8d17-c52860ed50a9
7	2	f90400d5-19e1-4f2f-b2a8-076509914f8a
8	2	75f48262-8b7a-4e69-97a9-3f7eb2b7c0e8
9	3	e2eb42ea-b957-48f5-8e22-bcf8923c5e6f
10	4	0419c69b-c93a-40c7-b7a1-f5f17f78ee8e
11	4	190f9295-b5e7-4521-b246-afbd9746f7bc

pid	shipment_date	destination_address	weight	recipient_name
0	2020-03-04	Addr13	2.4	Recip0
1	2020-03-04	Addr14	5.1	Recip1
2	2020-03-04	Addr15	4.3	Recip2
3	2020-03-04	Addr16	5.2	Recip3
4	2020-03-04	Addr17	1.4	Recip4
5	2020-03-04	Addr18	4.6	Recip5
6	2020-03-04	Addr19	2.5	Recip6
7	2020-03-04	Addr20	4.5	Recip7
8	2020-03-04	Addr21	1.4	Recip8
9	2020-03-04	Addr22	5.3	Recip9
10	2020-03-04	Addr23	1.5	Recip10
11	2020-03-04	Addr24	1.5	Recip11

sbid	cid	weight	recipient_name	destination_addr	shipping_date
0	0	2.4	Recip0	Addr13	2020-03-04
1	0	5.1	Recip1	Addr14	2020-03-04
2	1	4.3	Recip2	Addr15	2020-03-04
3	1	5.2	Recip3	Addr16	2020-03-04
4	2	1.4	Recip4	Addr17	2020-03-04
5	2	4.6	Recip5	Addr18	2020-03-04
6	3	2.5	Recip6	Addr19	2020-03-04
7	3	4.5	Recip7	Addr20	2020-03-04
8	4	1.4	Recip8	Addr21	2020-03-04
9	4	5.3	Recip9	Addr22	2020-03-04
10	4	1.5	Recip10	Addr23	2020-03-04
11	4	1.5	Recip11	Addr24	2020-03-04

sbid	eid
0	4
1	4
2	4
3	4
4	4
5	4
6	4
7	4
8	4
9	4
10	4
11	4

pid	sbid	pid	arrival_date	picked_up
		0	2020-03-04	TRUE
		1	2020-03-04	FALSE
		2	2020-03-04	TRUE
		3	2020-03-04	FALSE
		4	2020-03-04	TRUE
		5	2020-03-04	FALSE
		6	2020-03-04	TRUE
		7	2020-03-04	FALSE
		8	2020-03-04	FALSE
		9	2020-03-04	FALSE
		10	2020-03-04	FALSE
		11	2020-03-04	FALSE

arrival_date	picked_up	disposed_of
2020-03-01	TRUE	FALSE
2020-03-01	FALSE	TRUE
2020-03-02	TRUE	FALSE
2020-03-02	FALSE	TRUE
2020-03-03	TRUE	FALSE
2020-03-03	FALSE	TRUE
2020-03-04	TRUE	FALSE
2020-03-04	FALSE	TRUE
2020-03-05	TRUE	FALSE
2020-03-05	FALSE	TRUE

nid	name
0	Courier0
1	Courier1
2	Courier2
3	Courier3
4	Courier4

nbid	nid	phone_number	address	discount_per_lb
0	0	7781111111	Address60	0
1	1	7781111112	Address61	0
2	2	7781111114	Address62	0
3	3	7781111117	Address63	0
4	4	7781111121	Address64	0

nbid	bid
0	0
1	1
2	2
3	3
4	4

eid	pid
4	0
4	1
4	2
4	3
4	4
4	5
4	6
4	7
4	8
4	9
4	10
4	11