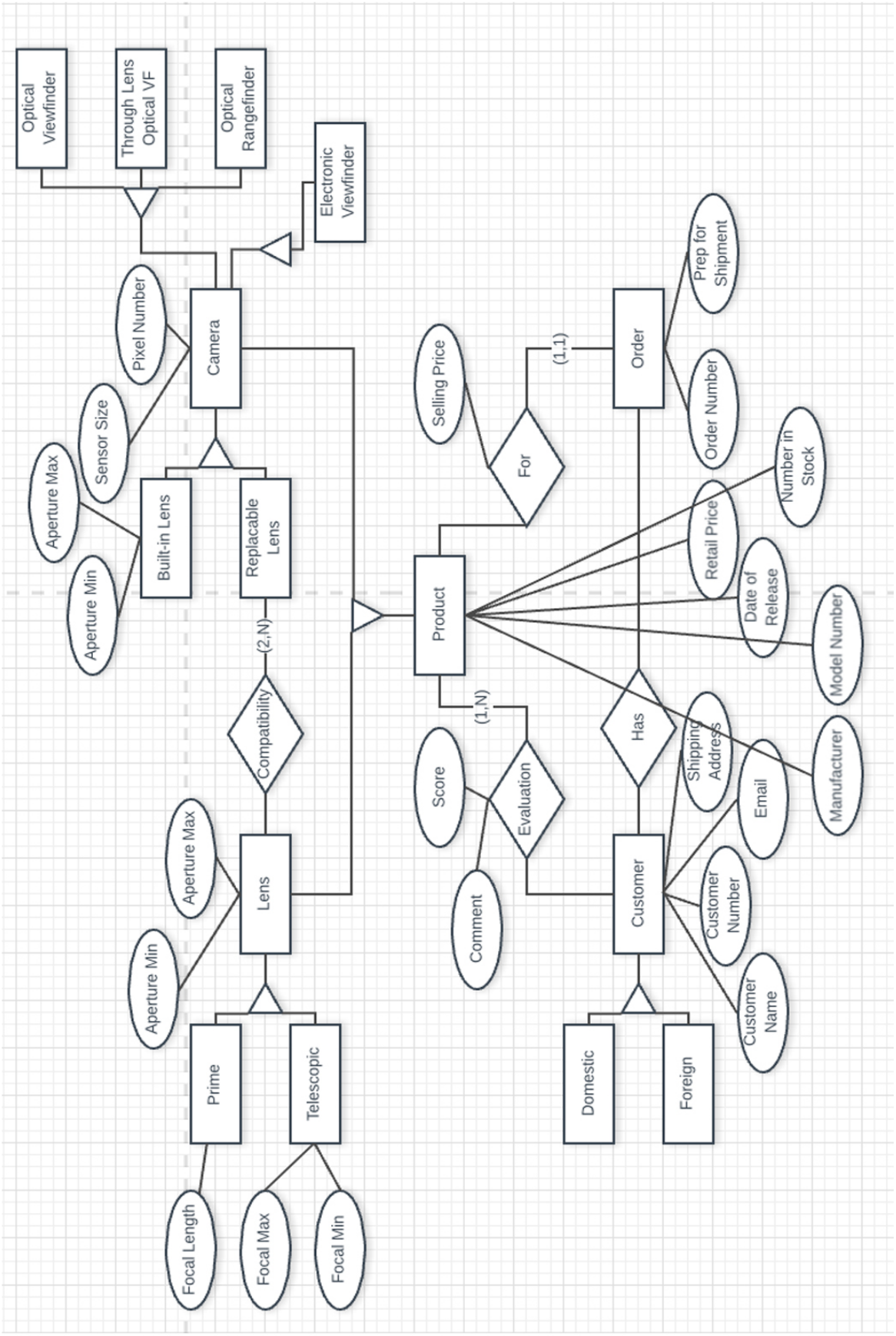
CS348 – Assignment 4
Q1.1



Q1.2

- The range of score for customer evaluation should be integers between 1 and 5 (from bad to good)
- The outstanding subset of purchased orders being prepared for shipment is not specified

Q2.1

```
create table product (
    productNumber integer not null,
    modelNumber varchar(15) not null,
    manufacturer varchar(30) not null,
    releaseDate date not null,
    retailPrice float not null,
    numberInStock integer not null,
    primary key (modelNumber));

create table camera (
    productNumber integer not null,
    sensorSize float not null,
    pixelNumber float not null,
    primary key (productNumber),
    foreign key (productNumber) references product (productNumber));

create table lens (
    productNumber integer not null,
    apertureMin float not null,
    apertureMax float not null,
    primary key (productNumber),
    foreign key (productNumber) references product (productNumber));

create table builtInLensCamera (
    productNumber integer not null,
    apertureMin float not null,
    apertureMax float not null,
    primary key (productNumber),
    foreign key (productNumber) references camera (productNumber));

create table replaceableLensCamera (
    productNumber integer not null,
    primary key (productNumber),
    foreign key (productNumber) references camera (productNumber));

create table EVF (
    productNumber integer not null,
    primary key (productNumber),
    foreign key (productNumber) references camera (productNumber));

create table OVF (
    productNumber integer not null,
    primary key (productNumber),
    foreign key (productNumber) references camera (productNumber));
```

```
create table TLOVF (
    productNumber integer not null,
    primary key (productNumber),
    foreign key (productNumber) references camera (productNumber));

create table ORF (
    productNumber integer not null,
    primary key (productNumber),
    foreign key (productNumber) references camera (productNumber));

create table primeLens (
    productNumber integer not null,
    focalLength float not null,
    primary key (productNumber),
    foreign key (productNumber) references lens (productNumber));

create table telescopicLens (
    productNumber integer not null,
    focalMin float not null,
    focalMax float not null,
    primary key (productNumber),
    foreign key (productNumber) references lens (productNumber));

create table customer (
    customerNumber integer not null,
    customerName integer not null,
    email varchar(40) not null,
    shippingAddress varchar(60) not null,
    primary key (customerNumber));

create table domesticCustomer (
    customerNumber integer not null,
    primary key (customerNumber)
    foreign key (customerNumber) references customer (customerNumber));

create table foreignCustomer (
    customerNumber integer not null,
    primary key (customerNumber)
    foreign key (customerNumber) references customer (customerNumber));

create table lensForRLCam (
    cameraNumber integer not null,
    lensNumber integer not null,
    primary key (cameraNumber, lensNumber),
    foreign key (cameraNumber) references camera (productNumber),
    //make sure there are at least 2 lenses compatible with replaceableLensCamera
    numberOfLens CHECK (
```

```
            NOT EXISTS (
                (SELECT productNumber FROM replaceableLensCamera)
                EXCEPT
                (SELECT replaceableLensCamera.productNumber
                FROM replaceableLensCamera a, lensForRLCam b, lensForRLCam c
                WHERE a.productNumber = b.cameraNumber
                    AND b.cameraNumber = c.cameraNumber
                    AND b.lensNumber <> c.lensNumber))),
    foreign key (lensNumber) references lens (productNumber));

create table order (
    orderNumber integer not null,
    productNumber integer not null,
    customerNumber integer not null,
    sellingPrice float not null,
    prepForShipment bool not null,
    primary key (orderNumber),
    foreign key (productNumber) references product (productNumber),
    foreign key (customerNumber) references customer (customerNumber));

create table evaluation (
    productNumber integer not null,
    customerNumber integer not null,
    score integer not null,
    comment varchar(100) not null,
    primary key (productNumber, customerNumber),
    foreign key (productNumber) references product (productNumber),
    foreign key (customerNumber) references customer (customerNumber));
```

Q2.2

1. $\sigma_{((lrl.lensNumber=l.productNumber) \wedge (lrl.cameraNumber=c.productNumber) \wedge (l.productNumber=pl.productNumber) \wedge (c.productNumber=rlc.productNumber))}$ $(\rho_{lrl}(lensForRLCam) \times \rho_{l}(lens) \times \rho_{c}(camera) \times \rho_{pl}(primeLens) \times \rho_{rlc}(replacableLensCamera))$

2. $\sigma_{((pro1.modelNumber<>pro2.modelNumber) \wedge (pro1.retailPrice>pro2.retailPrice) \wedge (pro1.productNumber=len1.productNumber) \wedge (pro2.productNumber=len2.productNumber) \wedge (len1.apertureMin<len2.apertureMin) \wedge (len1.productNumber=pri1.productNumber) \wedge (len2.productNumber=pri2.productNumber) \wedge (pri1.focalLength=pri2.focalLength))}$
$(\rho_{pro1}(product) \times \rho_{pro2}(product) \times \rho_{len1}(lens) \times \rho_{len2}(lens) \times \rho_{pri1}(primeLens) \times \rho_{pri2}(primeLens))$

Q3.

1. {ABC->D, D->A}
   a. Candidate keys are ABC and BCD
   b. The best normal form is BCNF

2. {AB->C, BC->D, A->C}
   a. Candidate keys are AD and ABC
   b. The best normal form is BCNF

3. {AB->C, AB->D, C->A, D->B}
   a. Candidate keys are AB and CD
   b. The best normal form is BCNF