horizontal line

DATABASE DESIGN FOR DOORDASH  
CS 6360 Final Project

INSTRUCTOR: Nurcan Yuruk, Ph.D.

Team 20: yxw200031 Yao-Kun Wu | yxz190007 Ying-Bo Zhao | txl160830 Tina Harter

TABLE OF CONTENTS

[**ENTITIES & REQUIREMENTS**](#_ectyjglzn2hi) **3**

[**ER DIAGRAM**](#_hk7un7mg2sht) **6**

[**MAPPING TO RELATIONAL MODEL**](#_lkz5fbe5kzyk) **7**

[**DEPENDENCIES**](#_fdpblyn31v1y) **11**

[**NORMALIZATION**](#_p57my6tkrxlj) **12**

[**FINAL RELATIONAL SCHEMA AFTER NORMALIZATION**](#_j80onujkqpra) **14**

[**SQL STATEMENT FOR DOORDASH DATABASE CREATION**](#_llx7pxo90s44) **19**

[**PL/SQL - TRIGGER 1**](#_93dvteajgk3n) **27**

[**PL/SQL - TRIGGER 2**](#_fhc79p3s1yhd) **28**

[**PL/SQL - PROCEDURE 1**](#_wiqgfqx6otxa) **29**

[**PL/SQL - PROCEDURE 2**](#_1mfyztxq53lx) **30**

# ENTITIES & REQUIREMENTS

DOOR DASH INTRODUCTION

Doordash is a food delivery system that users can order food from thousands of restaurants online and the order will be delivered by dasher. Customers can sign up their account by their personal information and they can choose restaurants on the application. Dashers use a different system with customers, they can accept the order and get order information as well as customer address. Their location will be updated in real time once they accept the order and start the delivery. On merchants’ side, they can accept the order on the system and update the progress online. In this project, we will design and implement a database for Doordash, the food delivery system, by using ER diagram, DB normalization, relational schema, SQL query, stored procedures and triggers.

1. CUSTOMER

Customers are people who ordered the food online. There are a number of registered customers. Each customer has a unique id, name, email address and phone number. Each customer may or may not purchase a DashPass for free delivery. Each customer may have multiple payment cards and multiple delivery addresses. Each card has a unique card number,, cardholder name, expiration date, and security code. Each address has a unique id, street, city, state, zip code, apt number, drop-off option and drop-off instruction.

* **Attributes:** 
  + **Customer:** id, name, email address, phone number, DashPass, AddressId, CardNo
  + **Address:** id, street, city, state, zip code, apt number, drop-off option, drop-off instruction
  + **Card:** card number, cardholder name, expiration date, security code

1. RESTAURANTS

Restaurants are the merchants who provide the food. There are a number of partner restaurants and each restaurant has a unique id, name, address, email, phone number and rating. Restaurants are categorized based on types. There are 17 main types such as Fast Food, Mexican, Chinese etc. Each restaurant provides multiple food items. Each restaurant is rated by customers from 1 star to 5 stars on a delivery basis. Each rating is associated with multiple tags and one optional feedback.

* **Attributes:**
  + **Restaurants:** id, name, address, email, phone number, rating, category

1. DASHERS

Dashers are people who deliver the food from restaurant to customer. There are a number of registered dashers. Each dasher has a unique id, name, email address, phone number and SSN. The dashers can have vehicles for delivery. Vehicles for dashers are either cars or bicycles. Each car has a make, model and color. Each dasher can schedule durations of time at a certain location they are available for delivery, where each location and duration has an associated bonus for each delivery. Each dasher has number of deliveries completed, number of stars rated by customer, number of delivery rated by customer, number of delivery opportunities received, number of delivery accepted and number of delivery completed on time or early. Each dasher can have a bank account number where they can deposit earnings. Each dasher is rated by customers from 1 star to 5 stars on a delivery basis. Each rating is associated with multiple tags and one optional feedback.

* **Attributes:**
  + **Dasher:** id, name, email address, phone number, SSN, delivery option, bank account number, number of rating received, total stars, number of deliveries accepted, number of deliveries completed on time, number of deliveries opportunity received, number of deliveries completed, vehicleId
  + **Vehicle:** LicensePlateNum, make, model, color

1. FOOD ITEMS

Food items are listed in the restaurant menu, customers can choose several food items and place orders. Each item has a unique id, name, food category, price, description, photo, required selections, optional additions.

* **Attributes:** 
  + **FoodItem:** id, name, food category, price, description, photo, required selections, optional additions.

1. ORDER

Order will be created once the customer pays the money. The order detail will be received by both restaurant and dasher. Order progress will be updated in real time on the order page. Each order is placed with a unique delivery id at a certain time and is taken by a dasher. Each order contains specified quantities of one or more food items from a single restaurant. Each order has a receipt, delivery address, status, date placed. Each receipt shows items ordered, price paid for each item, subtotal, delivery fee, service fee, dasher tip, tax and total amount.

* **Attributes:**
  + **Oder:** id, status, place date, delivery addr., receiptId
  + **Receipt:** id, total amount, delivery fee, tax, subtotal, tips, service fee

# 

# 

# 

# 

# ER DIAGRAM

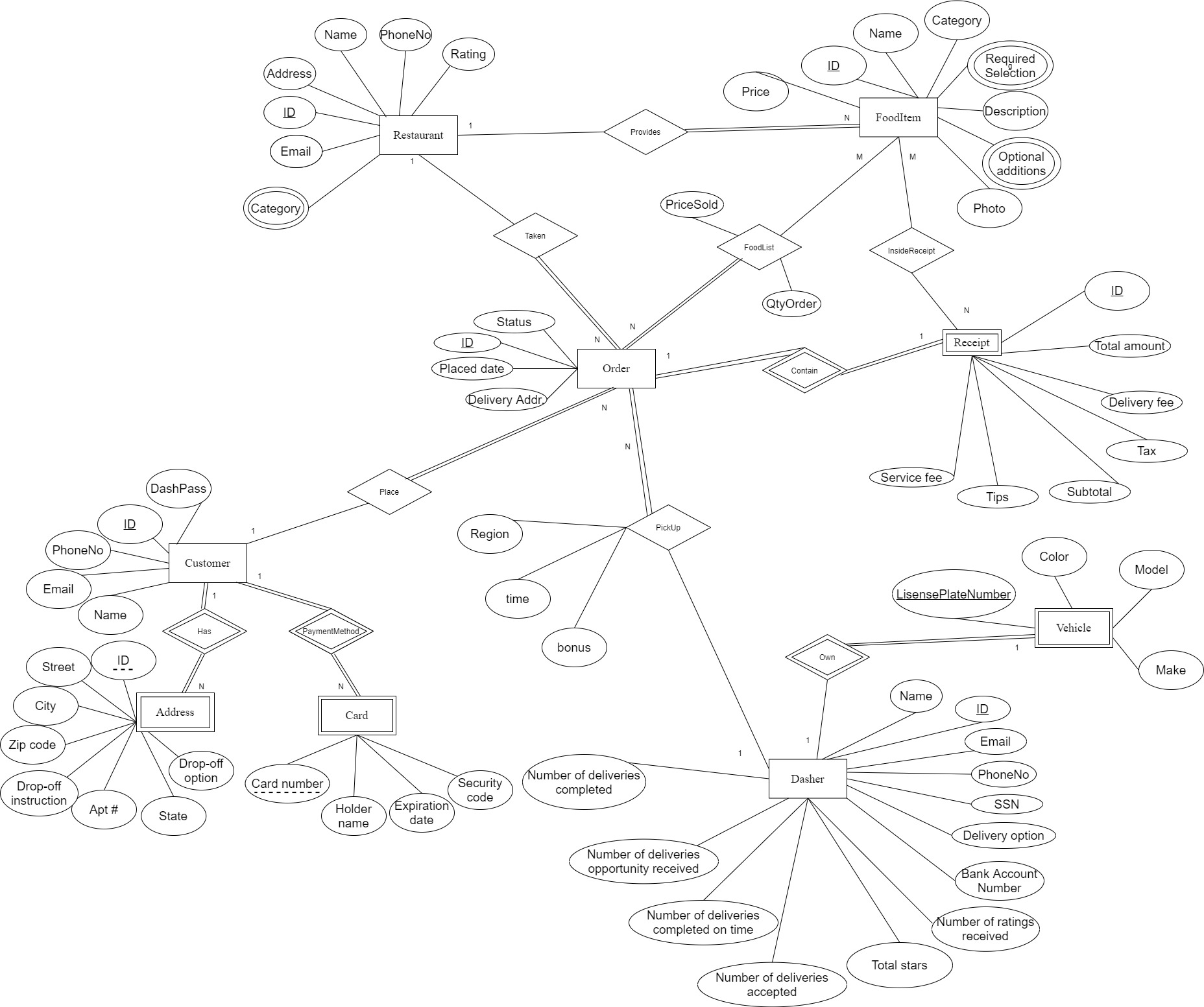


FIGURE 1. ER DIAGRAM OF DOORDASH

# 

# 

# MAPPING TO RELATIONAL MODEL

1. RESTAURANTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | RName | REmail | RAddress | RPhoneNo | RRating |

* Primary Key: RID
* Foreign Key: None

1. CUSTOMER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CID | DashPass | CPhoneNo | CEmail | CName |

* Primary Key: CID
* Foreign Key: None

1. DASHER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DID | DName | DEmail | DPhoneNo | SSN | DeliveryOp | BAcctNo |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NumsRating | TotalStar | NumsAccp | NumsComp | NumsOnTime | NumsOpRece |

* Primary Key: DID
* Foreign Key: None

1. ORDER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| OID | Status | RID | CID | DID | PlacedDate | DeliveryAddr |

|  |  |  |
| --- | --- | --- |
| PickUpRegion | PickUpTime | PickUpBonus |

* Primary Key: OID
* Foreign Key: CID references Customer(CID), DID references Dasher(DID), RID references Restaurant(RID)

1. FOODITEM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FID | RID | FName | FCategory | Description | Photo | Price |

* Primary Key: FID
* Foreign Key: RID references Restaurant(RID)

1. ADDRESS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AID | CID | Street | City | ZipCode | DropOffOp | DropOffInst |

|  |  |
| --- | --- |
| AptNo | State |

* Primary Key: AID
* Foreign Key: CID references Customer(CID)

1. CARD

|  |  |  |
| --- | --- | --- |
| CardNo | CID | HolderName |

* Primary Key: CID, CardNo
* Foreign Key: CID references Customer(CID)

1. RECEIPT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| OID | SID | TotalAmt | DeliveryFee | Tax | Subtotal | Tips | ServiceFee |

* Primary Key: OID, SID
* Foreign Key: OID references Order(OID)

1. VEHICLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DID | PlateNo | Color | Model | Make |

* Primary Key: DID, PlateNo
* Foreign Key: DID references Dasher(DID)

1. FOODLIST

|  |  |  |  |
| --- | --- | --- | --- |
| OID | FID | PriceSold | QtyOrder |

* Primary Key: OID, FID
* Foreign Key: OID references Order(OID), FID references FoodItem(FID)

1. RESTCATEGORY

|  |  |
| --- | --- |
| RID | RCategory |

* Primary Key: RID, RCategory
* Foreign Key: RID references Restaurant(RID)

1. FOODREQSLT

|  |  |
| --- | --- |
| FID | Option |

* Primary Key: FID,Option
* Foreign Key: FID reference FoodItem(FID)

1. FOODOPTADD

|  |  |
| --- | --- |
| FID | Addition |

* Primary Key: FID, Addition
* Foreign Key: FID reference FoodItem(FID)

1. INSIDERECE

|  |  |
| --- | --- |
| FID | SID |

* Primary Key: FID, SID
* Foreign Key: FID reference FoodItem(FID), SID references Receipt(SID)

# DEPENDENCIES

RID -> RName, RAddress,REmail, RPhoneNo, RRating

CID -> DashPass, CPhoneNo, CEmail, CName

DID -> DName, DEmail, DPhoneNo, SSN, DeliveryOp, BAcctNo, NumsRating,

TotalStar, NumsAccp, NumsComp, NumsOnTime, NumsOpRece

OID-> Status, RID, CID, DID, PlacedDate, DeliveryAddr, PickUpRegion, PickUpTime, PickUpBonus

FID -> RID, FName, FCategory, Description, Photo, Price

CID, AID ->, Street, City, ZipCode, DropOffOp, DropOffInst, AptNo, State

CID -> CardNo, HolderName

CardNo -> HolderName

OID, SID -> TotalAmt, DeliveryFee, Tax, Subtotal, Tips, ServiceFee

OID -> SID

DID, PlateNo -> Color, Model, Make

DID -> PlateNo

OID, FID -> PriceSold, QtyOrder

RID -> RCategory

SID, FID-> FName, QtyOrder, PriceSold

# NORMALIZATION

* All Relations are in 1NF.
* Receipt relation not in 2NF







* Vehicle relation not in 2NF







* Card relation not in 3NF







# FINAL RELATIONAL SCHEMA AFTER NORMALIZATION

1. RESTAURANTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RID | RName | REmail | RAddress | RPhoneNo | RRating |

* Primary Key: RID
* Foreign Key: None

1. CUSTOMER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CID | DashPass | CPhoneNo | CEmail | CName |

* Primary Key: CID
* Foreign Key: None

1. DASHER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DID | DName | DEmail | DPhoneNo | SSN | DeliveryOp | BAcctNo |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NumsRating | TotalStar | NumsAccp | NumsComp | NumsOnTime | NumsOpRece |

* Primary Key: DID
* Foreign Key: None

1. ORDER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| OID | Status | RID | CID | DID | PlacedDate | DeliveryAddr |

|  |  |  |
| --- | --- | --- |
| PickUpRegion | PickUpTime | PickUpBonus |

* Primary Key: OID
* Foreign Key: CID references Customer(CID), DID references Dasher(DID), RID references Restaurant(RID)

1. FOODITEM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FID | RID | FName | FCategory | Description | Photo | Price |

* Primary Key: FID
* Foreign Key: RID references Restaurant(RID)

1. ADDRESS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AID | CID | Street | City | ZipCode | DropOffOp | DropOffInst |

|  |  |
| --- | --- |
| AptNo | State |

* Primary Key: AID
* Foreign Key: CID references Customer(CID)

1. CARD

|  |  |
| --- | --- |
| CardNo | CID |

* Primary Key: CID, CardNo
* Foreign Key: CID references Customer(CID)

1. CARDDETAIL

|  |  |
| --- | --- |
| CardNo | HolderName |

* Primary Key: CardNo
* Foreign Key: CardNo references Card(CardNo)

1. RECEIPT

|  |  |
| --- | --- |
| OID | SID |

* Primary Key: OID, SID
* Foreign Key: OID references Order(OID)

1. RECEIPTDETAIL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SID | TotalAmt | DeliveryFee | Tax | Subtotal | Tips | ServiceFee |

* Primary Key: SID
* Foreign Key: SID references Receipt(SID)

1. VEHICLE

|  |  |
| --- | --- |
| DID | PlateNo |

* Primary Key: DID, PlateNo
* Foreign Key: DID references Dasher(DID)

1. VEHICLEDETAIL

|  |  |  |  |
| --- | --- | --- | --- |
| PlateNo | Color | Model | Make |

* Primary Key: PlateNo
* Foreign Key: PlateNo reference Vehicle(PlateNo)

1. FOODLIST

|  |  |  |  |
| --- | --- | --- | --- |
| OID | FID | PriceSold | QtyOrder |

* Primary Key: OID, FID
* Foreign Key: OID references Order(OID), FID references FoodItem(FID)

1. RESTCATEGORY

|  |  |
| --- | --- |
| RID | RCategory |

* Primary Key: RID, RCategory
* Foreign Key: RID references Restaurant(RID)

1. FOODREQSLT

|  |  |
| --- | --- |
| FID | Option |

* Primary Key: FID,Option
* Foreign Key: FID reference FoodItem(FID)

1. FOODOPTADD

|  |  |
| --- | --- |
| FID | Addition |

* Primary Key: FID, Addition
* Foreign Key: FID reference FoodItem(FID)

1. INSIDERECE

|  |  |
| --- | --- |
| FID | SID |

* Primary Key: FID, SID
* Foreign Key: FID reference FoodItem(FID), SID references Receipt(SID)

# SQL STATEMENT FOR DOORDASH DATABASE CREATION

CREATE TABLE restaurant (

rname VARCHAR(30) NOT NULL,

raddress VARCHAR(64),

rid VARCHAR(10),

remail VARCHAR(30),

rphoneno CHAR(10),

rrating DECIMAL(2, 1),

PRIMARY KEY ( rid ),

UNIQUE ( rname )

);

CREATE TABLE customer (

cid VARCHAR(10),

dashpass char,

cphoneno CHAR(10),

cemail VARCHAR(30),

cname VARCHAR(30) NOT NULL,

PRIMARY KEY ( cid )

);

CREATE TABLE dasher (

dname VARCHAR(30) NOT NULL,

did VARCHAR(10),

demail VARCHAR(30),

ephoneno CHAR(10),

ssn CHAR(9),

deliveryop VARCHAR(10),

bacctno VARCHAR(16),

numsrating INTEGER,

totalstar INTEGER,

numsaccp INTEGER,

numscomp INTEGER,

PRIMARY KEY ( did )

);

CREATE TABLE orderinfo (

status VARCHAR(16),

oid VARCHAR(10),

rid VARCHAR(10),

cid VARCHAR(10),

did VARCHAR(10),

placeddate timestamp,

deliveryaddr VARCHAR(64),

pickupregion VARCHAR(20),

pickuptime timestamp,

PRIMARY KEY ( oid )

);

CREATE TABLE fooditem (

fid VARCHAR(10),

rid VARCHAR(10),

fname VARCHAR(20),

fcategory VARCHAR(20),

description VARCHAR(128),

photo VARCHAR(64),

price decimal,

PRIMARY KEY ( fid )

);

CREATE TABLE address (

cid VARCHAR(10),

aid VARCHAR(10),

street VARCHAR(20),

city VARCHAR(20),

zipcode CHAR(5),

dropoffop VARCHAR(20),

dropoffinst VARCHAR(20),

aptno VARCHAR(6),

state VARCHAR(20),

PRIMARY KEY ( aid )

);

CREATE TABLE card (

cid VARCHAR(10),

cardno VARCHAR(20),

PRIMARY KEY ( cid, cardno )

);

CREATE TABLE carddetail (

cardno VARCHAR(20),

holdername VARCHAR(30),

PRIMARY KEY ( cardno )

);

CREATE TABLE receipt (

oid VARCHAR(10),

sid VARCHAR(10),

PRIMARY KEY ( oid, sid )

);

CREATE TABLE receiptdetail (

sid VARCHAR(10),

totalamt INTEGER,

deliveryfee decimal,

tax decimal,

subtotal decimal,

tips decimal,

servicefee decimal,

PRIMARY KEY ( sid )

);

CREATE TABLE vehicle (

did VARCHAR(10),

plateno VARCHAR(10),

PRIMARY KEY ( did, plateno )

);

CREATE TABLE vehicledetail (

plateno VARCHAR(10),

color VARCHAR(10),

model VARCHAR(20),

make VARCHAR(20),

PRIMARY KEY ( plateno )

);

CREATE TABLE foodlist (

oid VARCHAR(10),

fid VARCHAR(10),

pricesold decimal,

qtyorder INTEGER,

PRIMARY KEY ( oid, fid )

);

CREATE TABLE restcategory (

rid VARCHAR(10),

rcategory VARCHAR(20),

PRIMARY KEY ( rid, rcategory )

);

CREATE TABLE foodreqslt (

fid VARCHAR(10),

foodoption VARCHAR(20),

PRIMARY KEY ( fid, foodoption )

);

CREATE TABLE foodoptadd (

fid VARCHAR(10),

addition VARCHAR(20),

PRIMARY KEY ( fid, addition )

);

CREATE TABLE insiderece (

sid VARCHAR(10),

fid VARCHAR(10),

PRIMARY KEY ( sid, fid )

);

ALTER TABLE orderinfo

ADD CONSTRAINT ofkcid FOREIGN KEY ( cid )

REFERENCES customer ( cid );

ALTER TABLE orderinfo

ADD CONSTRAINT ofkdid FOREIGN KEY ( did )

REFERENCES dasher ( did );

ALTER TABLE orderinfo

ADD CONSTRAINT ofkrid FOREIGN KEY ( rid )

REFERENCES restaurant ( rid );

ALTER TABLE fooditem

ADD CONSTRAINT fifkrid FOREIGN KEY ( rid )

REFERENCES restaurant ( rid );

ALTER TABLE address

ADD CONSTRAINT addfkcid FOREIGN KEY ( cid )

REFERENCES customer ( cid );

ALTER TABLE card

ADD CONSTRAINT crdfkcid FOREIGN KEY ( cid )

REFERENCES customer ( cid );

ALTER TABLE carddetail

ADD CONSTRAINT cdfkcno FOREIGN KEY ( cardno )

REFERENCES carddetail ( cardno );

ALTER TABLE receipt

ADD CONSTRAINT recfkoid FOREIGN KEY ( oid )

REFERENCES orderinfo ( oid );

ALTER TABLE RECEIPTDETAIL

ADD CONSTRAINT RDFKSID FOREIGN KEY(SID)

REFERENCES RECEIPT(SID);

ALTER TABLE vehicle

ADD CONSTRAINT vfkdid FOREIGN KEY ( did )

REFERENCES dasher ( did );

ALTER TABLE VEHICLEDETAIL

ADD CONSTRAINT VDFKPLATENO FOREIGN KEY(PLATENO)

REFERENCES VEHICLE(PLATENO);

ALTER TABLE foodlist

ADD CONSTRAINT flfkoid FOREIGN KEY ( oid )

REFERENCES orderinfo ( oid );

ALTER TABLE foodlist

ADD CONSTRAINT flfkfid FOREIGN KEY ( fid )

REFERENCES fooditem ( fid );

ALTER TABLE restcategory

ADD CONSTRAINT rcfkrid FOREIGN KEY ( rid )

REFERENCES restaurant ( rid );

ALTER TABLE foodreqslt

ADD CONSTRAINT frfkfid FOREIGN KEY ( fid )

REFERENCES fooditem ( fid );

ALTER TABLE foodoptadd

ADD CONSTRAINT fofkfid FOREIGN KEY ( fid )

REFERENCES fooditem ( fid );

ALTER TABLE insiderece

ADD CONSTRAINT irfksid FOREIGN KEY ( sid )

REFERENCES receipt ( sid );

ALTER TABLE insiderece

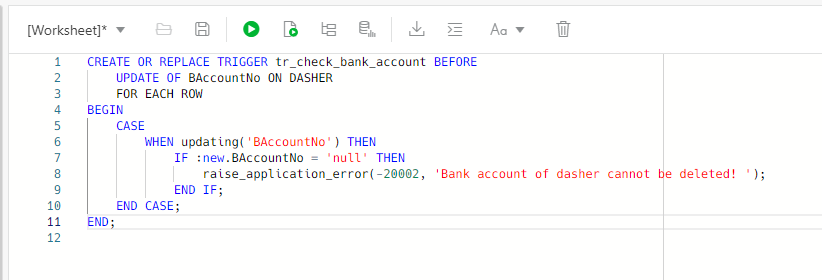
ADD CONSTRAINT irfkfid FOREIGN KEY ( fid )

REFERENCES fooditem ( fid );

horizontal line

# PL/SQL - TRIGGER 1

This trigger does not allow anyone to delete dasher’s bank account . If invalid operation is detected, the trigger will prompt the error message.



CREATE OR REPLACE TRIGGER tr\_check\_bank\_account BEFORE

UPDATE OF BAccountNo ON DASHER

FOR EACH ROW

BEGIN

CASE

WHEN updating('BAccountNo') THEN

IF :new.BAccountNo = 'null' THEN

raise\_application\_error(-20002, 'Bank account of dasher cannot be deleted! ');

END IF;

END CASE;

END;

# PL/SQL - TRIGGER 2

# This trigger updates the number of order completion on the dasher table after UPDATE order status on the order table. (Note that order status can only be updated from 0 meaning unfulfilled to fulfilled.)

# 

# CREATE TRIGGER orderCompletion AFTER

# UPDATE OF Status ON ORDER

# FOR EACH ROW

# 

# BEGIN

# UPDATE dasher SET NumsComp = NumsComp + 1

# WHERE dasher.DID = :old.DID;

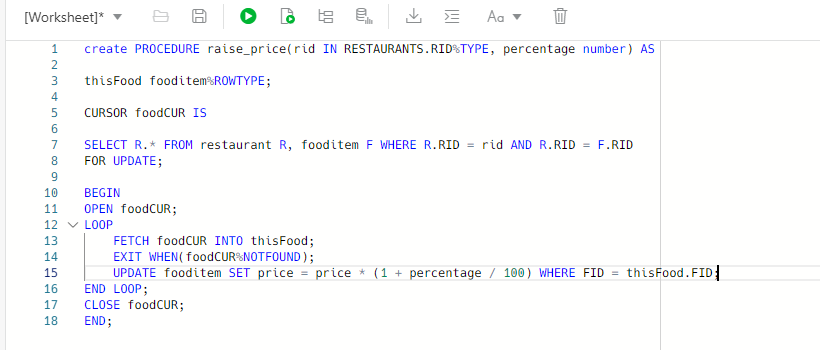
# END;

# 

# 

# PL/SQL - PROCEDURE 1

This procedure raises the prices of all items sold by a given restaurant by a given percentage.



CREATE PROCEDURE raise\_price(rid IN RESTAURANTS.RID%TYPE, percentage number) AS

thisFood fooditem%ROWTYPE;

CURSOR foodCUR IS

SELECT R.\* FROM restaurant R, fooditem F WHERE R.RID = rid AND R.RID = F.RID

FOR UPDATE;

BEGIN

OPEN foodCUR;

LOOP

FETCH foodCUR INTO thisFood;

EXIT WHEN(foodCUR%NOTFOUND);

UPDATE fooditem SET price = price \* (1 + percentage / 100) WHERE FID = thisFood.FID;

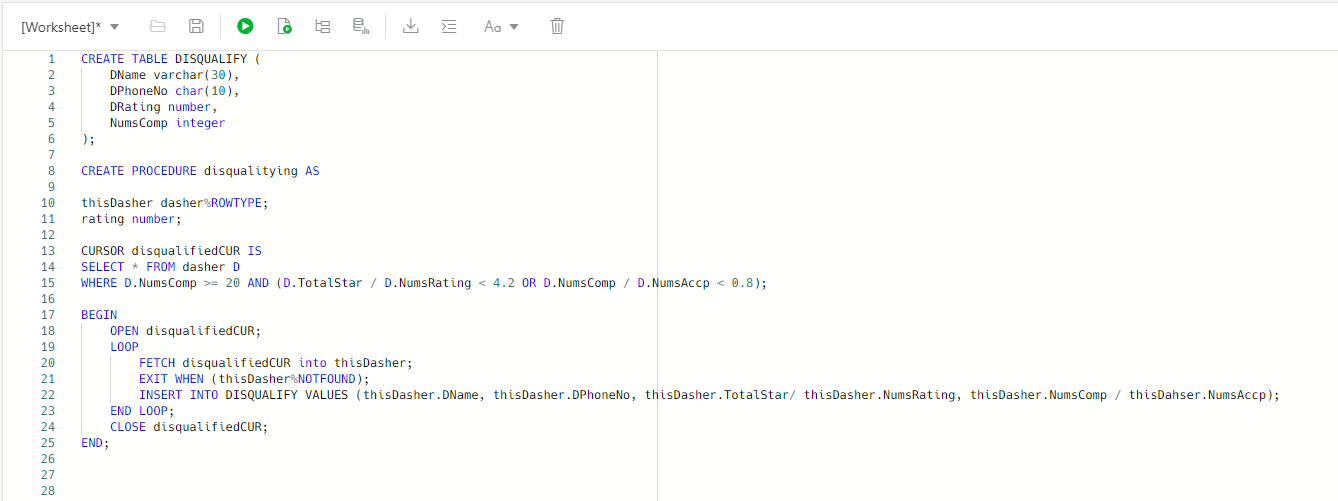
END LOOP;

CLOSE foodCUR;

END;

# PL/SQL - PROCEDURE 2

This procedure finds all dashers that have completed 20 or more orders but have customer rating below 4.2 or completion rate below 80% and insert dasher name, phone number, customer rating and completion rate into DISQUALIFY table.No input parameter is needed. (Create an additional table named DISQUALIFY with attributes: DName, DPhoneNo, DRating, NumsComp.)



CREATE TABLE DISQUALIFY (

DName varchar(30),

DPhoneNo char(10),

DRating number,

NumsComp integer

);

CREATE PROCEDURE disqualitying AS

thisDasher dasher%ROWTYPE;

rating number;

CURSOR disqualifiedCUR IS

SELECT \* FROM dasher D

WHERE D.NumsComp >= 20 AND (D.TotalStar / D.NumsRating < 4.2 OR D.NumsComp / D.NumsAccp < 0.8);

BEGIN

OPEN disqualifiedCUR;

LOOP

FETCH disqualifiedCUR into thisDasher;

EXIT WHEN (thisDasher%NOTFOUND);

INSERT INTO DISQUALIFY VALUES (thisDasher.DName, thisDasher.DPhoneNo, thisDasher.TotalStar/ thisDasher.NumsRating, thisDasher.NumsComp / thisDahser.NumsAccp);

END LOOP;

CLOSE disqualifiedCUR;

END;