Group17_code

2024-03-13

```
all_files <- list.files("Dataset/")
all_files
```

```
##  [1] "hi_actualdeliverydate_dataset.csv"
##  [2] "hi_customerbasicinfo_dataset.csv"
##  [3] "hi_estimateddeliverydate_dataset.csv"
##  [4] "hi_membership_dataset.csv"
##  [5] "hi_orderdate_dataset.csv"
##  [6] "hi_orderproductsinfo_dataset.csv"
##  [7] "hi_product_dataset.csv"
##  [8] "hi_productcategory_dataset.csv"
##  [9] "hi_promotion_dataset.csv"
## [10] "hi_supplier_dataset.csv"
## [11] "hi_trackingnumber_dataset.csv"
## [12] "hi_transaction_dataset.csv"
```

```
prefix <- "hi_"
suffix <- "_dataset.csv"

all_files <- gsub("hi_","",all_files)
all_files <- gsub("_dataset.csv","",all_files)
all_files
```

```
##  [1] "actualdeliverydate"    "customerbasicinfo"     "estimateddeliverydate"
##  [4] "membership"            "orderdate"             "orderproductsinfo"
##  [7] "product"               "productcategory"       "promotion"
## [10] "supplier"              "trackingnumber"        "transaction"
```

Looping through - for

Check number of rows and columns

```
all_files <- list.files("Dataset/")

for (variable in all_files) {
  this_filepath <- paste0("Dataset/",variable)
  this_file_contents <- readr::read_csv(this_filepath)

  number_of_rows <- nrow(this_file_contents)
  number_of_columns <- ncol(this_file_contents)

  print(paste0("The file: ",variable,
               " has: ",
               format(number_of_rows,big.mark = ","),
               " rows and ",
               number_of_columns," columns"))
```

```
}
```

```
## [1] "The file: hi_actualdeliverydate_dataset.csv has: 1,000 rows and 4 columns"
## [1] "The file: hi_customerbasicinfo_dataset.csv has: 1,000 rows and 11 columns"
## [1] "The file: hi_estimateddeliverydate_dataset.csv has: 1,000 rows and 4 columns"
## [1] "The file: hi_membership_dataset.csv has: 1,000 rows and 3 columns"
## [1] "The file: hi_orderdate_dataset.csv has: 1,681 rows and 4 columns"
## [1] "The file: hi_orderproductsinfo_dataset.csv has: 3,401 rows and 4 columns"
## [1] "The file: hi_product_dataset.csv has: 1,000 rows and 9 columns"
## [1] "The file: hi_productcategory_dataset.csv has: 88 rows and 2 columns"
## [1] "The file: hi_promotion_dataset.csv has: 1,000 rows and 4 columns"
## [1] "The file: hi_supplier_dataset.csv has: 1,000 rows and 8 columns"
## [1] "The file: hi_trackingnumber_dataset.csv has: 1,000 rows and 3 columns"
## [1] "The file: hi_transaction_dataset.csv has: 1,000 rows and 4 columns"
```

#Check the data structure

```
all_files <- list.files("Dataset/")

for (variable in all_files) {
  this_filepath <- paste0("Dataset/",variable)
  this_file_contents <- readr::read_csv(this_filepath)
  data_structure<-str(this_file_contents)

  print(paste0(data_structure,
               "The file: ",variable,
              " has above data structure"))
}
```

```
## spc_tbl_ [1,000 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ tracking_number     : chr [1:1000] "581-6200" "004-1482" "064-9023" "657-4120" ...
##  $ tracking_status     : chr [1:1000] "Delivered" "Delivered" "In process" "Delivered" ...
##  $ actual_delivery_date: chr [1:1000] "24/06/2023" "22/11/2023" NA "01/02/2024" ...
##  $ actual_delivery_time: 'hms' num [1:1000] 12:45:44 08:18:48 NA 07:43:05 ...
##   ..- attr(*, "units")= chr "secs"
##  - attr(*, "spec")=
##   .. cols(
##   ..   tracking_number = col_character(),
##   ..   tracking_status = col_character(),
##   ..   actual_delivery_date = col_character(),
##   ..   actual_delivery_time = col_time(format = "")
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_actualdeliverydate_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ customer_id       : chr [1:1000] "NAT-21446" "MQV-12400" "MLY-44705" "RFE-59474" ...
##  $ customer_firstname: chr [1:1000] "Mike" "Antone" "Moria" "Ichabod" ...
##  $ customer_lastname : chr [1:1000] "Berriball" "Lujan" "Llewellen" "Philson" ...
##  $ customer_title    : chr [1:1000] "Dr" "Mr" "Honorable" "Honorable" ...
##  $ customer_phone    : chr [1:1000] "+44 482 422 6609" "+44 941 356 9889" "+44 398 412 8484" "+44 114
##  $ customer_email    : chr [1:1000] "mberriball0@abc.net.au" "alujan1@qq.com" "mllewellen2@hud.gov" "
##  $ customer_building : num [1:1000] 103 436 861 271 107 72 701 21 615 122 ...
##  $ customer_street   : chr [1:1000] "Willow Street" "Spruce Street" "Willow Street" "Maple Street" .
##  $ customer_city     : chr [1:1000] "Birmingham" "Birmingham" "Bristol" "Bristol" ...
##  $ customer_postcode : chr [1:1000] "B1D 6RT" "G1A 8DD" "G4H 0NH" "M04 5UF" ...
```

```
##  $ promo_code        : chr [1:1000] "VS04A935ONO" "OQ50R17OHWT" "DU63L727XKV" "IW96D852NGT" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   customer_id = col_character(),
##   ..   customer_firstname = col_character(),
##   ..   customer_lastname = col_character(),
##   ..   customer_title = col_character(),
##   ..   customer_phone = col_character(),
##   ..   customer_email = col_character(),
##   ..   customer_building = col_double(),
##   ..   customer_street = col_character(),
##   ..   customer_city = col_character(),
##   ..   customer_postcode = col_character(),
##   ..   promo_code = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_customerbasicinfo_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ tracking_number        : chr [1:1000] "581-6200" "004-1482" "064-9023" "657-4120" ...
##  $ shipment_method        : chr [1:1000] "express" "next day" "express" "standard" ...
##  $ estimated_delivery_date: chr [1:1000] "23/06/2023" "21/11/2023" "06/03/2024" "01/02/2024" ...
##  $ estimated_delivery_time: 'hms' num [1:1000] 11:20:33 14:41:18 07:27:46 15:10:18 ...
##   ..- attr(*, "units")= chr "secs"
##  - attr(*, "spec")=
##   .. cols(
##   ..   tracking_number = col_character(),
##   ..   shipment_method = col_character(),
##   ..   estimated_delivery_date = col_character(),
##   ..   estimated_delivery_time = col_time(format = "")
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_estimateddeliverydate_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ customer_id        : chr [1:1000] "NAT-21446" "MQV-12400" "MLY-44705" "RFE-59474" ...
##  $ customer_membership: chr [1:1000] "membership" "not membership" "not membership" "membership" ...
##  $ delivery_fee       : num [1:1000] 0 4.99 5.99 0 0 2.99 5.99 0 6.99 5.99 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   customer_id = col_character(),
##   ..   customer_membership = col_character(),
##   ..   delivery_fee = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_membership_dataset.csv has above data structure"
## spc_tbl_ [1,681 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ customer_id: chr [1:1681] "YTS-92438" "BWU-36083" "CLZ-73501" "KCQ-71974" ...
##  $ order_id   : chr [1:1681] "AAD-4091" "AAK-0526" "AAK-6361" "ADJ-5614" ...
##  $ order_date : chr [1:1681] "10/11/2023" "09/02/2024" "28/03/2024" "19/11/2023" ...
##  $ order_time : 'hms' num [1:1681] 05:43:29 14:20:43 20:29:45 23:34:47 ...
##   ..- attr(*, "units")= chr "secs"
##  - attr(*, "spec")=
##   .. cols(
##   ..   customer_id = col_character(),
##   ..   order_id = col_character(),
```

```
##   ..   order_date = col_character(),
##   ..   order_time = col_time(format = "")
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_orderdate_dataset.csv has above data structure"
## spc_tbl_ [3,401 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ customer_id: chr [1:3401] "YTS-92438" "YTS-92438" "BWU-36083" "BWU-36083" ...
##  $ order_id   : chr [1:3401] "AAD-4091" "AAD-4091" "AAK-0526" "AAK-0526" ...
##  $ product_id : chr [1:3401] "42-811-3974" "72-217-8555" "85-279-1314" "43-612-9451" ...
##  $ product_qty: num [1:3401] 9 10 5 19 16 20 14 4 13 10 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   customer_id = col_character(),
##   ..   order_id = col_character(),
##   ..   product_id = col_character(),
##   ..   product_qty = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_orderproductsinfo_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ product_id    : chr [1:1000] "34-100-2931" "60-215-8627" "10-395-8862" "84-465-9981" ...
##  $ product_name  : chr [1:1000] "Unveiled: The Life of a Visionary" "Learn & Play Alphabet Blocks" "(
##  $ product_weight: num [1:1000] 4322 614 1942 3825 471 ...
##  $ product_length: num [1:1000] 67 24 65 74 67 83 75 63 68 67 ...
##  $ product_height: num [1:1000] 35 22 10 73 31 87 100 47 33 92 ...
##  $ product_width : num [1:1000] 30 98 30 47 87 94 100 56 61 58 ...
##  $ product_price : num [1:1000] 11.6 30.8 26.3 43.2 14.8 ...
##  $ supplier_id   : chr [1:1000] "RSH-48812" "HNW-87364" "QIS-31117" "TJZ-16253" ...
##  $ category_name : chr [1:1000] "Biography" "Educational Toys" "Fresh Produce" "History" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   product_id = col_character(),
##   ..   product_name = col_character(),
##   ..   product_weight = col_double(),
##   ..   product_length = col_double(),
##   ..   product_height = col_double(),
##   ..   product_width = col_double(),
##   ..   product_price = col_double(),
##   ..   supplier_id = col_character(),
##   ..   category_name = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_product_dataset.csv has above data structure"
## spc_tbl_ [88 x 2] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ category_name     : chr [1:88] "Beauty" "Books" "Clothing" "Electronics" ...
##  $ parent_category_id: chr [1:88] NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   category_name = col_character(),
##   ..   parent_category_id = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_productcategory_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
```

```
##  $ promo_code         : chr [1:1000] "BU86M505PYD" "BG70Z584RFB" "JI24S173EUJ" "FU87P552XKO" ...
##  $ promo_start_date   : chr [1:1000] "07/07/2023" "09/11/2023" "20/03/2024" "24/11/2023" ...
##  $ promo_expire_date  : chr [1:1000] "27/12/2023" "03/03/2024" "07/07/2024" "06/04/2024" ...
##  $ percentage_discount: num [1:1000] 50 25 45 45 40 30 35 10 40 45 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   promo_code = col_character(),
##   ..   promo_start_date = col_character(),
##   ..   promo_expire_date = col_character(),
##   ..   percentage_discount = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_promotion_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ supplier_id      : chr [1:1000] "XTE-60952" "WLS-09227" "RCO-72629" "KCV-52154" ...
##  $ supplier_name    : chr [1:1000] "Hyatt and Sons" "Huels-Krajcik" "Morissette LLC" "Koepp, Bechtela
##  $ supplier_phone   : chr [1:1000] "+44 336 825 7695" "+44 515 420 8651" "+44 213 964 1394" "+44 404
##  $ supplier_email   : chr [1:1000] "mkilpatrick0@nyu.edu" "cbritt1@unesco.org" "ctrussler2@hao123.com
##  $ supplier_building: num [1:1000] 796 881 66 921 254 968 44 554 774 679 ...
##  $ supplier_street  : chr [1:1000] "Garden Road" "Meadow Road" "Garden Road" "River Road" ...
##  $ supplier_city    : chr [1:1000] "London" "Birmingham" "Birmingham" "London" ...
##  $ supplier_postcode: chr [1:1000] "KY2Y 6JZ" "B12 7TB" "DE9W 6WF" "W1 9SG" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   supplier_id = col_character(),
##   ..   supplier_name = col_character(),
##   ..   supplier_phone = col_character(),
##   ..   supplier_email = col_character(),
##   ..   supplier_building = col_double(),
##   ..   supplier_street = col_character(),
##   ..   supplier_city = col_character(),
##   ..   supplier_postcode = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_supplier_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ tracking_number      : chr [1:1000] "581-6200" "004-1482" "064-9023" "657-4120" ...
##  $ delivery_instructions: chr [1:1000] "ring bell" "delivery box" "ring bell" "leave infront of door
##  $ trans_id             : chr [1:1000] "AAA-067232" "AAC-152328" "AAD-850184" "AAR-680860" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   tracking_number = col_character(),
##   ..   delivery_instructions = col_character(),
##   ..   trans_id = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_trackingnumber_dataset.csv has above data structure"
## spc_tbl_ [1,000 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trans_id  : chr [1:1000] "EHD-784624" "SIZ-926836" "GEP-276863" "YRS-371629" ...
##  $ order_id  : chr [1:1000] "AAD-4091" "AAK-0526" "ADJ-6838" "ADV-4775" ...
##  $ trans_date: chr [1:1000] "10/11/2023" "09/02/2024" "02/08/2023" "26/10/2023" ...
##  $ trans_time: 'hms' num [1:1000] 07:56:40 15:23:52 12:30:16 14:57:31 ...
##   ..- attr(*, "units")= chr "secs"
##  - attr(*, "spec")=
```

```
##   .. cols(
##   ..   trans_id = col_character(),
##   ..   order_id = col_character(),
##   ..   trans_date = col_character(),
##   ..   trans_time = col_time(format = "")
##   .. )
##  - attr(*, "problems")=<externalptr>
## [1] "The file: hi_transaction_dataset.csv has above data structure"
```

#Check for NULL values

```
all_files <- list.files("data_upload/")

for (variable in all_files) {
  this_filepath <- paste0("data_upload/",variable)
  this_file_contents <- readr::read_csv(this_filepath)
  null<-sum(is.na(this_file_contents))

  print(paste0("The file: ",variable,
               " has a total of ", null,
               " NULL values"))
}
```

#Check that each primary key is unique in each table

```
all_files <- list.files("Dataset/")

for (variable in all_files) {
  this_filepath <- paste0("Dataset/",variable)
  this_file_contents <- readr::read_csv(this_filepath)
  hi <- nrow(unique(this_file_contents[,1]))== nrow(this_file_contents)

  print(paste0("The file: ",variable,
               " has unique primary key ",
               hi," columns"))
}
```

```
## [1] "The file: hi_actualdeliverydate_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_customerbasicinfo_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_estimateddeliverydate_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_membership_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_orderdate_dataset.csv has unique primary key FALSE columns"
## [1] "The file: hi_orderproductsinfo_dataset.csv has unique primary key FALSE columns"
## [1] "The file: hi_product_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_productcategory_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_promotion_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_supplier_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_trackingnumber_dataset.csv has unique primary key TRUE columns"
## [1] "The file: hi_transaction_dataset.csv has unique primary key TRUE columns"
```

For order dataset: we have a composite composed of 3 attribute, we'll check this one separately.

```
orderdate_dataset <- read.csv("Dataset/hi_orderdate_dataset.csv")
orderproductsinfo_dataset <- read.csv("Dataset/hi_orderproductsinfo_dataset.csv")
nrow(unique(orderdate_dataset[,1:3])) == nrow(orderdate_dataset)
```

```
## [1] TRUE
```

```r
nrow(unique(orderproductsinfo_dataset[,1:3])) == nrow(orderproductsinfo_dataset)
```

```
## [1] TRUE
```

```r
#sum(nrow(unique(orders[,1:2])))
#length((unique(orders$cutomer_id)))
#length((unique(orders$order_id)))
```

The file: hi_order_dataset.csv has unique primary composite key TRUE columns

Load Files in an sqlite database

```r
#setup the connection
connection <- RSQLite::dbConnect(RSQLite::SQLite(),"hi_import.db")
```

1. product_category

```
-- product_category
CREATE TABLE IF NOT EXISTS "product_category" (
  category_name VARCHAR(50) PRIMARY KEY,
  parent_category_id INT NULL
);
```

```
SELECT * FROM "product_category";
```

Table 1: 0 records

category_name	parent_category_id

2. promotion

```
-- promotion
CREATE TABLE IF NOT EXISTS "promotion" (
  promo_code INT PRIMARY KEY,
  promo_start_date DATE NULL,
  promo_expire_date DATE NULL,
  percentage_discount NUMERIC NOT NULL
  );
```

```
SELECT * FROM "promotion";
```

Table 2: 0 records

promo_code	promo_start_date	promo_expire_date	percentage_discount

3. supplier

```
-- supplier
CREATE TABLE IF NOT EXISTS supplier (
  supplier_id INT PRIMARY KEY,
  supplier_name CHAR NOT NULL,
  supplier_phone INT NOT NULL,
  supplier_email VARCHAR(50) NOT NULL,
  supplier_building INT NOT NULL,
```

```
  supplier_street VARCHAR(50) NOT NULL,
  supplier_city VARCHAR(50) NOT NULL,
  supplier_postcode VARCHAR(50) NOT NULL
) ;
```

```
SELECT * FROM "supplier";
```

Table 3: 0 records

supplier_id	supplier_name	supplier_phone	supplier_email	supplier_building	supplier_street	supplier_city	supplier_postcode

4. customer

```
-- customer
CREATE TABLE IF NOT EXISTS "customer" (
  customer_id INT PRIMARY KEY,
  customer_firstname VARCHAR(50) NOT NULL,
  customer_lastname VARCHAR(50) NOT NULL,
  customer_title VARCHAR(25) NOT NULL,
  customer_phone VARCHAR(50) NOT NULL,
  customer_email VARCHAR(50) NOT NULL,
  customer_membership TEXT NOT NULL,
  delivery_fee NUMERIC NOT NULL,
  customer_building INT NOT NULL,
  customer_street VARCHAR(50) NOT NULL,
  customer_city VARCHAR(50) NOT NULL,
  customer_postcode VARCHAR(50) NOT NULL,
  promo_code INT,
  FOREIGN KEY (promo_code) REFERENCES "promotion"(promo_code)
) ;
```

```
SELECT * FROM "customer";
```

Table 4: 0 records

customer_id	customer_firstname	customer_lastname	customer_title	customer_phone	customer_email	delivery_fee	customer_building	customer_street	customer_city	customer_postcode

5. delivery

```
-- delivery
CREATE TABLE IF NOT EXISTS "delivery" (
  tracking_number INT PRIMARY KEY,
  shipment_method VARCHAR(50) NOT NULL,
  tracking_status VARCHAR(50) NOT NULL,
  estimated_delivery_date DATE NOT NULL,
  estimated_delivery_time TIME NOT NULL,
  actual_delivery_date DATE NULL,
  actual_delivery_time TIME NULL,
  delivery_instructions VARCHAR(125) NOT NULL,
  trans_id INT,
  FOREIGN KEY (trans_id) REFERENCES "transaction"(trans_id)
);
```

```
SELECT * FROM "delivery";
```

Table 5: 0 records

tracking_number	shipment_method	tracking_status	estimated_delivery_date	actual_delivery_date	delivery_instructions	transaction_id

6. product

```
-- product
CREATE TABLE IF NOT EXISTS "product" (
  product_id INT PRIMARY KEY,
  product_name VARCHAR(25) NOT NULL,
  product_weight NUMERIC NOT NULL,
  product_length NUMERIC NOT NULL,
  product_height NUMERIC NOT NULL,
  product_width NUMERIC NOT NULL,
  product_price NUMERIC NOT NULL,
  supplier_id INT,
  category_name VARCHAR(50),
  FOREIGN KEY (supplier_id) REFERENCES "supplier"(supplier_id),
  FOREIGN KEY (category_name) REFERENCES "product_category"(category_name)
) ;
```

```
SELECT * FROM "product";
```

Table 6: 0 records

product_id	product_name	product_weight	product_length	product_height	product_width	product_price	supplier_id	category_name

7. order

```
-- order
CREATE TABLE IF NOT EXISTS "order" (
  customer_id INT,
  order_id INT,
  product_id INT,
  product_qty INT NOT NULL,
  order_date DATE NOT NULL,
  order_time TIME NOT NULL,
  PRIMARY KEY (customer_id, order_id, product_id),
  FOREIGN KEY (customer_id) REFERENCES "customer"(customer_id),
  FOREIGN KEY (product_id) REFERENCES "product"(product_id)
) ;
```

```
SELECT * FROM "order";
```

Table 7: 0 records

customer_id	order_id	product_id	product_qty	order_date	order_time

8. transaction

```sql
-- transaction
CREATE TABLE IF NOT EXISTS "transaction" (
  trans_id INT PRIMARY KEY,
  order_id INT,
  trans_date DATE NOT NULL,
  trans_time TIME NOT NULL,
  FOREIGN KEY (order_id) REFERENCES "order"(order_id)
);
```

```sql
SELECT * FROM "transaction";
```

Table 8: 0 records

trans_id	order_id	trans_date	trans_time

Now normalize to 3NF

For customer 1. customer_basic_info

```sql
-- customer_basic_info
CREATE TABLE IF NOT EXISTS "customer_basic_info" (
  customer_id INT PRIMARY KEY,
  customer_firstname VARCHAR(50) NOT NULL,
  customer_lastname VARCHAR(50) NOT NULL,
  customer_title VARCHAR(25) NOT NULL,
  customer_phone VARCHAR(50) NOT NULL,
  customer_email VARCHAR(50) NOT NULL,
  customer_building INT NOT NULL,
  customer_street VARCHAR(50) NOT NULL,
  customer_city VARCHAR(50) NOT NULL,
  customer_postcode VARCHAR(50) NOT NULL,
  promo_code INT,
  FOREIGN KEY (promo_code) REFERENCES "promotion"(promo_code)
) ;
```

```sql
SELECT * FROM customer_basic_info
```

Table 9: 0 records

customer_id	customer_firstname	customer_lastname	customer_title	customer_phone	customer_email	customer_building	customer_street	customer_city	customer_postcode	promo_code

2. customer_membership

```sql
-- customer_membership
CREATE TABLE IF NOT EXISTS "customer_memebership" (
  customer_id INT,
  customer_membership TEXT,
  delivery_fee NUMERIC NOT NULL,
  PRIMARY KEY (customer_id, customer_membership),
  FOREIGN KEY (customer_id) REFERENCES "customer_basic_info"(customer_id)
);
```

```sql
SELECT * FROM customer_memebership
```

Table 10: 0 records

customer_id	customer_membership	delivery_fee

For order

1. order_products_info

```
-- order_products_info
CREATE TABLE IF NOT EXISTS "order_products_info" (
  customer_id INT,
  order_id INT,
  product_id INT,
  product_qty INT NOT NULL,
  PRIMARY KEY (customer_id, order_id, product_id),
  FOREIGN KEY (customer_id) REFERENCES "customer_basic_info"(customer_id),
  FOREIGN KEY (product_id) REFERENCES "product"(product_id)
) ;
```

```
SELECT * FROM order_products_info
```

Table 11: 0 records

customer_id	order_id	product_id	product_qty

2. order_datetime

```
-- order_datetime
CREATE TABLE IF NOT EXISTS "order_datetime" (
  customer_id INT,
  order_id INT,
  order_date DATE NOT NULL,
  order_time TIME NOT NULL,
  PRIMARY KEY (customer_id, order_id),
  FOREIGN KEY (customer_id) REFERENCES "customer_basic_info"(customer_id)
) ;
```

```
SELECT * FROM order_datetime
```

Table 12: 0 records

customer_id	order_id	order_date	order_time

For Delivery 1. delivery_tracking

```
-- delivery_tracking
CREATE TABLE IF NOT EXISTS "delivery_tracking" (
  tracking_number INT PRIMARY KEY,
  delivery_instructions VARCHAR(125) NOT NULL,
  trans_id INT,
  FOREIGN KEY (trans_id) REFERENCES "transaction"(trans_id)
);
```

```sql
SELECT * FROM delivery_tracking
```

Table 13: 0 records

tracking_number	delivery_instructions	trans_id

2. estimated_delivery_date

```sql
-- estimated_delivery_date
CREATE TABLE IF NOT EXISTS "estimated_delivery_date" (
  tracking_number INT,
  shipment_method VARCHAR(50),
  estimated_delivery_date DATE NOT NULL,
  estimated_delivery_time TIME NOT NULL,
  PRIMARY KEY (tracking_number, shipment_method),
  FOREIGN KEY (tracking_number) REFERENCES "delivery_tracking"(tracking_number)
);
```

```sql
SELECT * FROM estimated_delivery_date
```

Table 14: 0 records

tracking_number	shipment_method	estimated_delivery_date	estimated_delivery_time

3. actual_delivery_date

```sql
-- actual_delivery_date
CREATE TABLE IF NOT EXISTS "actual_delivery_date" (
  tracking_number INT,
  tracking_status VARCHAR(50),
  actual_delivery_date DATE NULL,
  actual_delivery_time TIME NULL,
  PRIMARY KEY (tracking_number, tracking_status),
  FOREIGN KEY (tracking_number) REFERENCES "delivery_tracking"(tracking_number)
);
```

```sql
SELECT * FROM actual_delivery_date
```

Table 15: 0 records

tracking_number	tracking_status	actual_delivery_date	actual_delivery_time

```r
# Read datasets
#order
orderdate_dataset <- read.csv("Dataset/hi_orderdate_dataset.csv",row.names = NULL)
orderproductsinfo_dataset <- read.csv("Dataset/hi_orderproductsinfo_dataset.csv",row.names = NULL)

#delivery
actualdeliverydate_dataset <- read.csv("Dataset/hi_actualdeliverydate_dataset.csv",row.names = NULL)
trackingnumber_dataset <- read.csv("Dataset/hi_trackingnumber_dataset.csv",row.names = NULL)
estimateddeliverydate_dataset <- read.csv("Dataset/hi_estimateddeliverydate_dataset.csv",row.names = NU
```

```
#customer
hi_customerbasicinfo_dataset <- read.csv("Dataset/hi_customerbasicinfo_dataset.csv",row.names = NULL)
membership_dataset <- read.csv("Dataset/hi_membership_dataset.csv",row.names = NULL)

product_dataset <- read.csv("Dataset/hi_product_dataset.csv",row.names = NULL)
productcategory_dataset <- read.csv("Dataset/hi_productcategory_dataset.csv",row.names = NULL)
promotion_dataset <- read.csv("Dataset/hi_promotion_dataset.csv",row.names = NULL)
supplier_dataset <- read.csv("Dataset/hi_supplier_dataset.csv",row.names = NULL)
transaction_dataset <- read.csv("Dataset/hi_transaction_dataset.csv",row.names = NULL)
```

```
dbWriteTable(connection, "product", product_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "product_category", productcategory_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "promotion", promotion_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "supplier", supplier_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "transaction", transaction_dataset, append = TRUE, row.names = FALSE)

#order
dbWriteTable(connection, "order_datetime", orderdate_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "order_products_info", orderproductsinfo_dataset, append = TRUE, row.names = FA

#delivery
dbWriteTable(connection, "actual_delivery_date", actualdeliverydate_dataset, append = TRUE, row.names =
dbWriteTable(connection, "delivery_tracking", trackingnumber_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "estimated_delivery_date", estimateddeliverydate_dataset, append = TRUE, row.na

#customer
dbWriteTable(connection, "customer_memebership", membership_dataset, append = TRUE, row.names = FALSE)
dbWriteTable(connection, "customer_basic_info", hi_customerbasicinfo_dataset, append = TRUE, row.names =
```

```
RSQLite::dbListTables(connection)
```

```
##  [1] "actual_delivery_date"    "customer"
##  [3] "customer_basic_info"     "customer_memebership"
##  [5] "delivery"                "delivery_tracking"
##  [7] "estimated_delivery_date" "order"
##  [9] "order_datetime"          "order_products_info"
## [11] "product"                 "product_category"
## [13] "promotion"               "supplier"
## [15] "transaction"
```

```
RSQLite::dbDisconnect(connection)
```