Database System Final Project – Part 1

Description Design and Assumptions

The first part of the Database final project aims to design a possible relational database schema that resolves business needs for a car rental company. The assumptions that we make when designing our model are as follows:

- The rental office may or may not have rental cars. The M:N relationship is optional.
- The car is a specific type of vehicle identified by a unique VIN(Vehicle Identification Number). This relationship is 1:1.
- The entity customer is the supertype of the individual customer and the corporate customer. The discriminator is the attribute CustType in table customer.
- A customer can make many or zero orders. The M:N relationship is optional.
- One order is associated with only one invoice.
- One invoice can be paid with multiple payments.
- The rental office has its own address. It's a 1:1 relationship. Same as the customer's address.
- An address can be the pickup or dropoff location for many orders.
- When the customer places an order, it needs to declare where the car would be returned, which is the dropoff location in the order.
- One customer must have one corresponding address.
- One office must have one corresponding address, one address could only have one office.
- One car may not be ready to be assigned to any office, therefore the relation between car and office is optional.
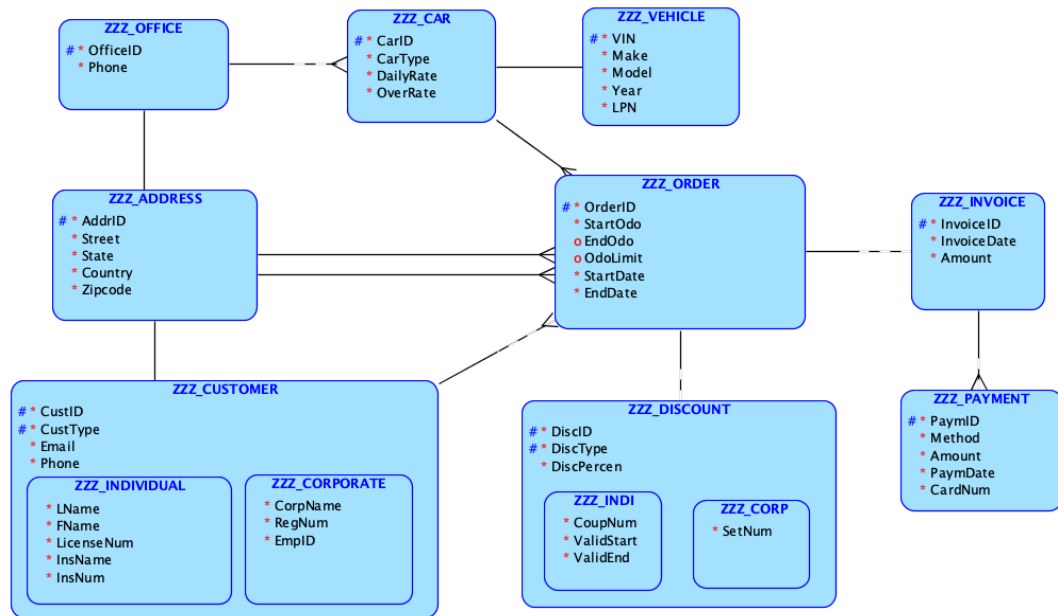
Logical Model

ZZZ_OFFICE
- # * OfficeID
- * Phone

ZZZ_CAR
- # * CarID
- * CarType
- * DailyRate
- * OverRate

ZZZ_VEHICLE
- # * VIN
- * Make
- * Model
- * Year
- * LPN

ZZZ_ADDRESS
- # * AddrID
- * Street
- * State
- * Country
- * Zipcode

ZZZ_ORDER
- # * OrderID
- * StartOdo
- o EndOdo
- o OdoLimit
- * StartDate
- * EndDate

ZZZ_INVOICE
- # * InvoiceID
- * InvoiceDate
- * Amount

ZZZ_CUSTOMER
- # * CustID
- # * CustType
- * Email
- * Phone

ZZZ_INDIVIDUAL
- * LName
- * FName
- * LicenseNum
- * InsName
- * InsNum

ZZZ_CORPORATE
- * CorpName
- * RegNum
- * EmpID

ZZZ_DISCOUNT
- # * DiscID
- # * DiscType
- * DiscPercen

ZZZ_INDI
- * CoupNum
- * ValidStart
- * ValidEnd

ZZZ_CORP
- * SetNum

ZZZ_PAYMENT
- # * PaymID
- * Method
- * Amount
- * PaymDate
- * CardNum

Figure 1: Logical model for WOW.

Relational Model

ZZZ_OFFICE
- P * OfficeID   NUMBER (6)
- * Phone   VARCHAR2 (20)
- F * AddrID   NUMBER (10)
- ZZZ_OFFICE_PK (OfficeID)
- ZZZ_OFFICE_ZZZ_ADDRESS_FK (AddrID)
- ZZZ_OFFICE__IDX (AddrID)

ZZZ_CAR
- P * CarID   NUMBER (6)
- * CarType   VARCHAR2 (20)
- * DailyRate   NUMBER (5,2)
- * OverRate   NUMBER (5,2)
- F   OfficeID   NUMBER (6)
- F   VIN   VARCHAR2 (20)
- ZZZ_CAR_PK (CarID)
- ZZZ_CAR_ZZZ_VEHICLE_FK (VIN)
- ZZZ_CAR_ZZZ_OFFICE_FK (OfficeID)
- ZZZ_CAR__IDX (VIN)

ZZZ_VEHICLE
- P * VIN   VARCHAR2 (20)
- * Make   VARCHAR2 (30)
- * Model   VARCHAR2 (30)
- * Year   NUMBER (4)
- * LPN   VARCHAR2 (15)
- ZZZ_VEHICLE_PK (VIN)

ZZZ_INVOICE
- P * InvoiceID   NUMBER (10)
- * InvoiceDate   DATE
- * Amount   NUMBER (8)
- F * OrderID   NUMBER (10)
- ZZZ_INVOICE_PK (InvoiceID)
- ZZZ_INVOICE_ZZZ_ORDER_FK (OrderID)

ZZZ_ADDRESS
- P * AddrID   NUMBER (10)
- * Street   VARCHAR2 (30)
- * State   VARCHAR2 (30)
- * Country   VARCHAR2 (30)
- * Zipcode   NUMBER (5)
- ZZZ_ADDRESS_PK (AddrID)

ZZZ_ORDER
- P * OrderID   NUMBER (10)
- * StartOdo   NUMBER (10)
-    EndOdo   NUMBER (10)
-    OdoLimit   NUMBER (10)
- * StartDate   DATE
- * EndDate   DATE
- F * CustID   NUMBER (6)
- F * CustType   CHAR (1)
- F * PickUp   NUMBER (10)
- F * DropOff   NUMBER (10)
- F * CarID   NUMBER (6)
- F   DiscID   NUMBER (10)
- F   DiscType   CHAR (1)
- ZZZ_ORDER_PK (OrderID)
- ZZZ_ORDER_ZZZ_CAR_FK (CarID)
- ZZZ_ORDER_ZZZ_CUSTOMER_FK (CustID, CustType)
- ZZZ_ORDER_ZZZ_ADDRESS_FK (DropOff)
- ZZZ_ORDER_ZZZ_DISCOUNT_FK (DiscType, DiscID)
- ZZZ_ORDER_ZZZ_ADDRESS_FKv2 (PickUp)
- ZZZ_ORDER__IDX (DiscType, DiscID)

ZZZ_PAYMENT
- P * PaymID   NUMBER (10)
- * Method   VARCHAR2 (20)
- * Amount   NUMBER (8)
- * PaymDate   DATE
- * CardNum   NUMBER (20)
- F * InvoiceID   NUMBER (10)
- ZZZ_PAYMENT_PK (PaymID)
- ZZZ_PAYMENT_ZZZ_INVOICE_FK (InvoiceID)

ZZZ_CUSTOMER
- P * CustID   NUMBER (6)
- P * CustType   CHAR (1)
- * Email   VARCHAR2 (20)
- * Phone   VARCHAR2 (20)
- F * AddrID   NUMBER (10)
- ZZZ_CUSTOMER_PK (CustID, CustType)
- ZZZ_CUSTOMER_ZZZ_ADDRESS_FK (AddrID)
- ZZZ_CUSTOMER__IDX (AddrID)

ZZZ_INDIVIDUAL
- PF * CustID   NUMBER (6)
- PF * CustType   CHAR (1)
- * LName   VARCHAR2 (20)
- * FName   VARCHAR2 (20)
- * LicenseNum   NUMBER (15)
- * InsName   VARCHAR2 (20)
- * InsNum   NUMBER (20)
- ZZZ_INDIVIDUAL_PK (CustID, CustType)
- ZZZ_INDIVIDUAL_ZZZ_CUSTOMER_FK (CustID, CustType)

ZZZ_CORPORATE
- PF * CustID   NUMBER (6)
- PF * CustType   CHAR (1)
- * CorpName   VARCHAR2 (20)
- * RegNum   VARCHAR2 (20)
- * EmpID   VARCHAR2 (20)
- ZZZ_CORPORATE_PK (CustID, CustType)
- ZZZ_CORPORATE_ZZZ_CUSTOMER_FK (CustID, CustType)

ZZZ_DISCOUNT
- P * DiscID   NUMBER (10)
- P * DiscType   CHAR (1)
- * DiscPercen   NUMBER (5,2)
- ZZZ_DISCOUNT_PK (DiscType, DiscID)

ZZZ_INDI
- PF * DiscID   NUMBER (10)
- PF * DiscType   CHAR (1)
- * CoupNum   VARCHAR2 (10)
- * ValidStart   DATE
- * ValidEnd   DATE
- ZZZ_INDI_PK (DiscType, DiscID)
- ZZZ_INDI_ZZZ_DISCOUNT_FK (DiscType, DiscID)

ZZZ_CORP
- PF * DiscID   NUMBER (10)
- PF * DiscType   CHAR (1)
- * SetNum   NUMBER (20)
- ZZZ_CORP_PK (DiscType, DiscID)
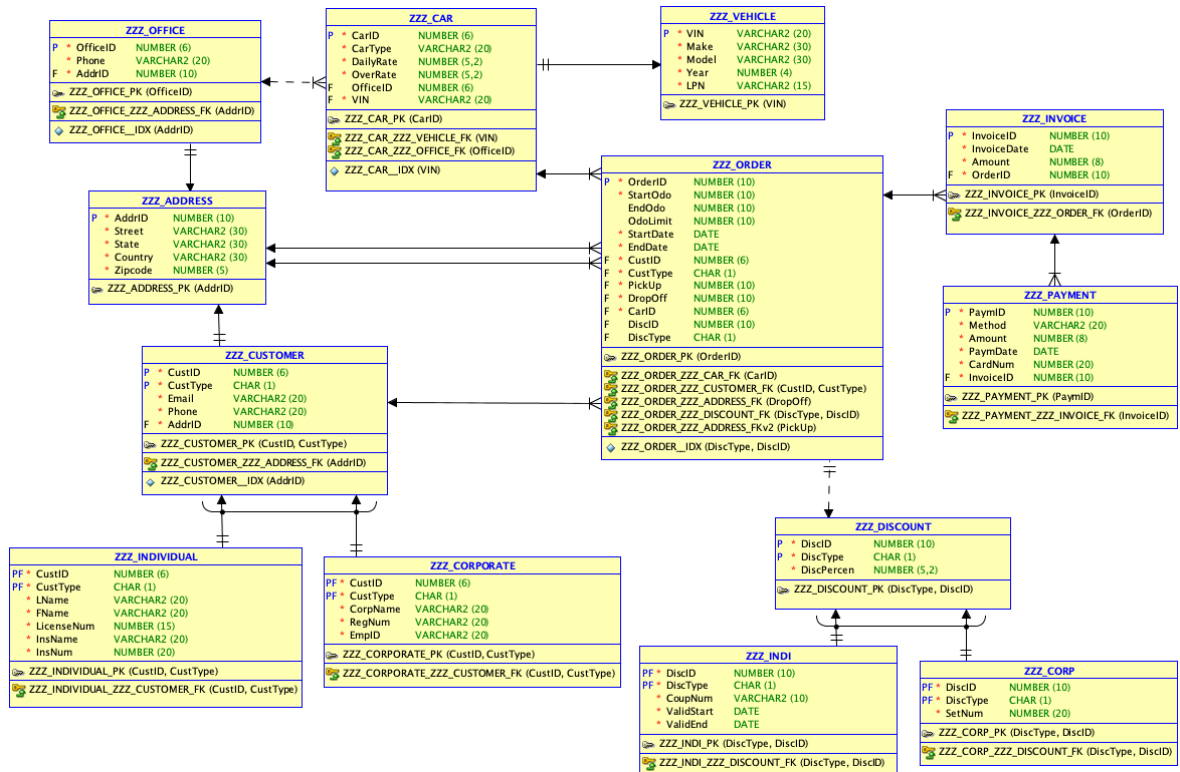- ZZZ_CORP_ZZZ_DISCOUNT_FK (DiscType, DiscID)

Figure 2: Relational model for WOW.

Tables and Record Counts

Execute the following MySQL script to display the table name and its corresponding record counts in our schema. Table 1 gives the outputs.

```
-- List of tables and counts
select table_name, table_rows
from information_schema.tables
where table_schema='WOW' and table_name like 'ZZZ%';
```

Table 1: Tables and record counts.

table_name	table_rows
zzz_address	24
zzz_car	11
zzz_corp	7
zzz_corporate	5
zzz_customer	10
zzz_discount	14
zzz_indi	7
zzz_individual	5
zzz_invoice	5
zzz_office	12
zzz_order	6
zzz_payment	1
zzz_vehicle	11

DDL Codes (MySQL)

Codes in this section are the DDL in MySQL which is converted from Oracle by using the online tool sqlines.

```
-- SQLINES DEMO *** le SQL Developer Data Modeler 21.4.1.349.1605
-- SQLINES DEMO *** -04-08 16:24:21 EDT
-- SQLINES DEMO *** le Database 21c
-- SQLINES DEMO *** le Database 21c



-- SQLINES DEMO *** no DDL - MDSYS.SDO_GEOMETRY

-- SQLINES DEMO *** no DDL - XMLTYPE

-- SQLINES LICENSE FOR EVALUATION USE ONLY
```

```sql
CREATE TABLE zzz_address (
    addrid  BIGINT NOT NULL COMMENT 'Unique ID for address. ',
    street  VARCHAR(30) NOT NULL COMMENT 'Street info for address. ',
    state   VARCHAR(30) NOT NULL COMMENT 'State info for address. ',
    country VARCHAR(30) NOT NULL COMMENT 'Country info for address. ',
    zipcode INT NOT NULL COMMENT 'Zipcode for address. '
);

ALTER TABLE zzz_address ADD CONSTRAINT zzz_address_pk PRIMARY KEY ( addrid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_car (
    carid     INT NOT NULL COMMENT 'Unique ID for each car. ',
    cartype   VARCHAR(20) NOT NULL COMMENT 'Class of the car.',
    dailyrate DECIMAL(5, 2) NOT NULL COMMENT 'Regular rental rate per day of the
rental service for the car.',
    overrate  DECIMAL(5, 2) NOT NULL COMMENT 'Extra fees per mile that exceeds the
limit.',
    officeid  INT,
    vin       VARCHAR(20) NOT NULL
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX zzz_car__idx ON
    zzz_car (
        vin
    ASC );

ALTER TABLE zzz_car ADD CONSTRAINT zzz_car_pk PRIMARY KEY ( carid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_corp (
    discid   BIGINT NOT NULL COMMENT 'Uinque ID for discount.',
    disctype CHAR(1) NOT NULL COMMENT 'Discriminator of discount type.',
    setnum   DECIMAL(20) NOT NULL COMMENT 'Number for indentifying corporation'
);

ALTER TABLE zzz_corp ADD CONSTRAINT zzz_corp_pk PRIMARY KEY ( disctype,
                                                             discid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_corporate (
    custid   INT NOT NULL COMMENT 'Unique ID for customer.',
    custtype CHAR(1) NOT NULL COMMENT 'Customer type.',
    corpname VARCHAR(20) NOT NULL COMMENT 'Corporation''s name.',
    regnum   VARCHAR(20) NOT NULL COMMENT 'Registration number of the corporation.',
    empid    VARCHAR(20) NOT NULL COMMENT 'Employee ID of the customer who rents the
```

```sql
car on a corporate account.'
);
ALTER TABLE zzz_corporate ADD CONSTRAINT zzz_corporate_pk PRIMARY KEY ( custid,
                                                                        custtype );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_customer (
    custid   INT NOT NULL COMMENT 'Unique ID for customer.',
    custtype CHAR(1) NOT NULL COMMENT 'Customer type.',
    email    VARCHAR(20) NOT NULL COMMENT 'Email address for customer. ',
    phone    VARCHAR(20) NOT NULL COMMENT 'Phone number for customers. ',
    addrid   BIGINT NOT NULL
);

ALTER TABLE zzz_customer
    ADD CONSTRAINT ch_inh_zzz_customer CHECK ( custtype IN ( 'C', 'I' ) );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX zzz_customer__idx ON
    zzz_customer (
        addrid
    ASC );

ALTER TABLE zzz_customer ADD CONSTRAINT zzz_customer_pk PRIMARY KEY ( custid,
                                                                      custtype );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_discount (
    discid     BIGINT NOT NULL COMMENT 'Uinque ID for discount.',
    disctype   CHAR(1) NOT NULL COMMENT 'Discriminator of discount type.',
    discpercen DECIMAL(5, 2) NOT NULL COMMENT 'Discount percentage %.'
);

ALTER TABLE zzz_discount
    ADD CONSTRAINT ch_inh_zzz_discount CHECK ( disctype IN ( 'C', 'I' ) );

ALTER TABLE zzz_discount ADD CONSTRAINT zzz_discount_pk PRIMARY KEY ( disctype,
                                                                      discid );


-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_indi (
    discid     BIGINT NOT NULL COMMENT 'Uinque ID for discount.',
    disctype   CHAR(1) NOT NULL COMMENT 'Discriminator of discount type.',
    coupnum    VARCHAR(10) NOT NULL COMMENT 'Coupon number.',
    validstart DATETIME NOT NULL COMMENT 'Coupon valid start date.',
    validend   DATETIME NOT NULL COMMENT 'Coupon valid end date.'
);
```

```sql
ALTER TABLE zzz_indi ADD CONSTRAINT zzz_indi_pk PRIMARY KEY ( disctype,
                                                               discid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_individual (
    custid      INT NOT NULL COMMENT 'Unique ID for customer.',
    custtype    CHAR(1) NOT NULL COMMENT 'Customer type.',
    lname       VARCHAR(20) NOT NULL COMMENT 'Last name for individual customer. ',
    fname       VARCHAR(20) NOT NULL COMMENT 'First Name for individual customer. ',
    licensenum BIGINT NOT NULL COMMENT 'Drive license number.',
    insname     VARCHAR(20) NOT NULL COMMENT 'Insurance company name.',
    insnum      DECIMAL(20) NOT NULL COMMENT 'Insurance policy number.'
);

ALTER TABLE zzz_individual ADD CONSTRAINT zzz_individual_pk PRIMARY KEY ( custid,
                                                                          custtype
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_invoice (
    invoiceid   BIGINT NOT NULL COMMENT 'Unique ID for invoice.',
    invoicedate DATETIME NOT NULL COMMENT 'Invoice data.',
    amount      INT NOT NULL COMMENT 'Invoice amount.',
    orderid     BIGINT NOT NULL
);

ALTER TABLE zzz_invoice ADD CONSTRAINT zzz_invoice_pk PRIMARY KEY ( invoiceid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_office (
    officeid INT NOT NULL COMMENT 'Unique ID for office.',
    phone    VARCHAR(20) NOT NULL COMMENT 'Office''s phone number.',
    addrid   BIGINT NOT NULL
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX zzz_office__idx ON
    zzz_office (
        addrid
    ASC );

ALTER TABLE zzz_office ADD CONSTRAINT zzz_office_pk PRIMARY KEY ( officeid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_order (
```

```sql
    orderid   BIGINT NOT NULL COMMENT 'Unique ID for each order.',
    startodo  BIGINT NOT NULL COMMENT 'Start odometer.',
    endodo    BIGINT COMMENT 'End Odometer.',
    odolimit  BIGINT COMMENT 'Daily odometer limit for the rental service.',
    startdate DATETIME NOT NULL COMMENT 'Date when the customer starts the
service.',
    enddate   DATETIME NOT NULL COMMENT 'Date when the customer ends the service.',
    custid    INT NOT NULL,
    custtype  CHAR(1) NOT NULL,
    pickup    BIGINT NOT NULL,
    dropoff   BIGINT NOT NULL,
    carid     INT NOT NULL,
    discid    BIGINT,
    disctype  CHAR(1)
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX zzz_order__idx ON
    zzz_order (
        disctype
    ASC,
        discid
    ASC );

ALTER TABLE zzz_order ADD CONSTRAINT zzz_order_pk PRIMARY KEY ( orderid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_payment (
    paymid    BIGINT NOT NULL COMMENT 'Unique ID for payment.',
    method    VARCHAR(20) NOT NULL COMMENT 'Payment method.',
    amount    INT NOT NULL COMMENT 'Payment amount.',
    paymdate  DATETIME NOT NULL COMMENT 'Payment date.',
    cardnum   DECIMAL(20) NOT NULL COMMENT 'Card number.',
    invoiceid BIGINT NOT NULL
);

ALTER TABLE zzz_payment ADD CONSTRAINT zzz_payment_pk PRIMARY KEY ( paymid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE zzz_vehicle (
    vin   VARCHAR(20) NOT NULL COMMENT 'Vehicle identification number.',
    make  VARCHAR(30) NOT NULL COMMENT 'Brand of the vehicle.',
    model VARCHAR(30) NOT NULL COMMENT 'Name of a product or a range of products.',
    year  SMALLINT NOT NULL COMMENT 'Manufacture year of the vehicle. ',
    lpn   VARCHAR(15) NOT NULL COMMENT 'The registration identifier is a numeric or
alphanumeric ID that uniquely identifies the vehicle or vehicle owner within the
issuing region''s vehicle register.'
```

```sql
);

ALTER TABLE zzz_vehicle ADD CONSTRAINT zzz_vehicle_pk PRIMARY KEY ( vin );

ALTER TABLE zzz_car
    ADD CONSTRAINT zzz_car_zzz_office_fk FOREIGN KEY ( officeid )
        REFERENCES zzz_office ( officeid );

ALTER TABLE zzz_car
    ADD CONSTRAINT zzz_car_zzz_vehicle_fk FOREIGN KEY ( vin )
        REFERENCES zzz_vehicle ( vin );

ALTER TABLE zzz_corp
    ADD CONSTRAINT zzz_corp_zzz_discount_fk FOREIGN KEY ( disctype,
                                                          discid )
        REFERENCES zzz_discount ( disctype,
                                  discid );

ALTER TABLE zzz_corporate
    ADD CONSTRAINT zzz_corporate_zzz_customer_fk FOREIGN KEY ( custid,
                                                               custtype )
        REFERENCES zzz_customer ( custid,
                                  custtype );

ALTER TABLE zzz_customer
    ADD CONSTRAINT zzz_customer_zzz_address_fk FOREIGN KEY ( addrid )
        REFERENCES zzz_address ( addrid );

ALTER TABLE zzz_indi
    ADD CONSTRAINT zzz_indi_zzz_discount_fk FOREIGN KEY ( disctype,
                                                          discid )
        REFERENCES zzz_discount ( disctype,
                                  discid );

ALTER TABLE zzz_individual
    ADD CONSTRAINT zzz_individual_zzz_customer_fk FOREIGN KEY ( custid,
                                                                custtype )
        REFERENCES zzz_customer ( custid,
                                  custtype );

ALTER TABLE zzz_invoice
    ADD CONSTRAINT zzz_invoice_zzz_order_fk FOREIGN KEY ( orderid )
        REFERENCES zzz_order ( orderid );

ALTER TABLE zzz_office
    ADD CONSTRAINT zzz_office_zzz_address_fk FOREIGN KEY ( addrid )
        REFERENCES zzz_address ( addrid );
```

```sql
ALTER TABLE zzz_order
    ADD CONSTRAINT zzz_order_zzz_address_fk FOREIGN KEY ( dropoff )
        REFERENCES zzz_address ( addrid );

ALTER TABLE zzz_order
    ADD CONSTRAINT zzz_order_zzz_address_fkv2 FOREIGN KEY ( pickup )
        REFERENCES zzz_address ( addrid );

ALTER TABLE zzz_order
    ADD CONSTRAINT zzz_order_zzz_car_fk FOREIGN KEY ( carid )
        REFERENCES zzz_car ( carid );

ALTER TABLE zzz_order
    ADD CONSTRAINT zzz_order_zzz_customer_fk FOREIGN KEY ( custid,
                                                           custtype )
        REFERENCES zzz_customer ( custid,
                                  custtype );

ALTER TABLE zzz_order
    ADD CONSTRAINT zzz_order_zzz_discount_fk FOREIGN KEY ( disctype,
                                                           discid )
        REFERENCES zzz_discount ( disctype,
                                  discid );

ALTER TABLE zzz_payment
    ADD CONSTRAINT zzz_payment_zzz_invoice_fk FOREIGN KEY ( invoiceid )
        REFERENCES zzz_invoice ( invoiceid );




-- SQLINES DEMO *** per Data Modeler Summary Report:
--
-- SQLINES DEMO ***                         13
-- SQLINES DEMO ***                          4
-- SQLINES DEMO ***                         30
-- SQLINES DEMO ***                          0
-- SQLINES DEMO ***                          0
-- SQLINES DEMO ***                          0
-- SQLINES DEMO *** DY                       0
-- SQLINES DEMO ***                          0
-- SQLINES DEMO ***                          0
-- SQLINES DEMO ***                          4
-- SQLINES DEMO ***                          0
-- SQLINES DEMO ***   TYPE                   0
-- SQLINES DEMO ***   TYPE                   0
-- SQLINES DEMO ***   TYPE BODY              0
```

```
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO *** EGMENT                       0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO *** ED VIEW                      0
-- SQLINES DEMO *** ED VIEW LOG                  0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
--
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
--
-- SQLINES DEMO ***                              0
--
-- SQLINES DEMO ***                              0
-- SQLINES DEMO *** A                            0
-- SQLINES DEMO *** T                            0
--
-- SQLINES DEMO ***                              0
-- SQLINES DEMO ***                              0
```

DML Codes (MySQL)

Codes in this section are insert statements for most of our tables. Some of the DML codes will be introduced in the following section that are closely related to activate the trigger for invoice.

```sql
-- insert data to address
insert into zzz_address (addrid, street, state, country, zipcode)
values
(1, 'A St', 'NY', 'USA', '10001'),
(2, 'B St', 'NY', 'USA', '10002'),
(3, 'C St', 'PA', 'USA', '15003'),
(4, 'D St', 'PA', 'USA', '15004'),
(5, 'E St', 'CA', 'USA', '94043'),
(6, 'F St', 'CA', 'USA', '94086'),
(7, 'G St', 'CA', 'USA', '94089'),
(8, 'H St', 'MI', 'USA', '48228'),
(9, 'I St', 'MI', 'USA', '48103'),
(10, 'J St', 'NJ', 'USA', '07097'),
(11, 'K St', 'MA', 'USA', '02139'),
(12, 'L St', 'MA', 'USA', '02138'),
(13, 'M St', 'MA', 'USA', '02137'),
(14, 'N St', 'MA', 'USA', '01604'),
(15, 'O St', 'FL', 'USA', '33125'),
(16, 'P St', 'FL', 'USA', '33027'),
(17, 'Q St', 'FL', 'USA', '32244'),
(18, 'R St', 'FL', 'USA', '32808'),
(19, 'S St', 'WA', 'USA', '98101'),
(20, 'T St', 'WA', 'USA', '98102'),
(21, 'O St', 'WA', 'USA', '98004'),
(22, 'P St', 'WA', 'USA', '98005'),
(23, 'Q St', 'WA', 'USA', '98006'),
(24, 'R St', 'WA', 'USA', '98007');

-- insert data to office
insert into zzz_office (officeid, phone, addrid)
values
(12345, '(123)-456-789', 1),
(22345, '(677)-142-124', 2),
(32345, '(456)-634-565', 3),
(42345, '(535)-645-089', 4),
(52345, '(543)-579-019', 5),
(62345, '(082)-977-984', 6),
(72345, '(973)-234-893', 7),
(82345, '(279)-937-761', 8),
(92345, '(769)-733-173', 9),
(2345, '(692)-763-182', 10),
(112345, '(738)-123-918', 11),
```

```sql
(122345, '(162)-472-867', 12);

-- insert data to vehicle
insert into zzz_vehicle (vin, make, model, year, lpn)
values
('299J98JSJIW19923I', 'Volvo', 'Golf', 2001, '123989828'),
('28WHDJW92U939282', 'Honda', 'Acura', 2020, '123899233'),
('894739JUHE2932', 'General', 'Buick', 2003, '123719232'),
('283DWJEHBDU12', 'Toyota', 'Lexus', 2010, '12389SH23'),
('1231283DHUIHU', 'Ford', 'Lincoln', 2012, 'UH198929'),
('HIH289389223', 'Mazda', 'Mazda', 2015, 'HUUE97283'),
('PEI893904212', 'BMW', 'Mini', 2013, 'JIH172HU1'),
('838HIDN8929321', 'Stellantis', 'Ram', 2000, 'GHJDU288'),
('EIWDODO123871', 'BMW', 'Rolls-Royce', 2012, 'UD72JDS'),
('19HDHJ929391', 'Subaru', 'Subaru', 2014, 'DUU1738HJD'),
('BDU12378492112', 'Tesla', 'Tesla', 2020, '89273HJHA');

-- insert data to car
insert into zzz_car (carid, cartype, dailyrate, overrate, officeid, vin)
values
(1, 'small car', 1.0, 50.0, 12345,'299J98JSJIW19923I'),
(2, 'mid-size car', 78.28, 10.29, 22345, '28WHDJW92U939282'),
(3, 'luxury car', 28.17, 29.29, 32345, '894739JUHE2932'),
(4, 'SUV', 92.12, 82.23, 42345, '283DWJEHBDU12'),
(5, 'Premium SUV', 19.29, 28.39, 52345, '1231283DHUIHU'),
(6, 'Mini Van', 12.23, 34.72, 62345, 'HIH289389223'),
(7, 'Station Wagon', 49.38, 19.3, 72345, 'PEI893904212'),
(8, 'small car',28.34,20.0, 82345, '838HIDN8929321'),
(9, 'Premium SUV', 72.83, 22.92, 92345, 'EIWDODO123871'),
(10, 'Station Wagon', 48.20, 29.32,  2345,'19HDHJ929391'),
(11, 'SUV', 23.12, 28.23, 112345, 'BDU12378492112');

-- insert data to customer
insert into zzz_customer (custid, custtype, email, phone,  addrid)
values
(754710, 'I', 'erte@nyu.edu', '(162)-472-844', 13),
(754711, 'I', 'cxbd@nyu.edu', '(162)-472-845', 14),
(754712, 'I', 'yijj@nyu.edu', '(162)-472-846', 15),
(754713, 'I', 'cvfh@nyu.edu', '(162)-472-847', 16),
(754714, 'I', 'nyif@nyu.edu', '(162)-472-848', 17),
(754715, 'C', 'xcxg@nyu.edu', '(162)-472-849', 18),
(754716, 'C', 'iygj@nyu.edu', '(162)-472-850', 19),
(754717, 'C', 'xzzz@nyu.edu', '(162)-472-851', 20),
(754718, 'C', 'oyou@nyu.edu', '(162)-472-852', 21),
(754719, 'C', 'bnfg@nyu.edu', '(162)-472-853', 22);

-- insert data to individual
```

```sql
insert into zzz_individual (custid, custtype, lname, fname, licensenum, insname,
insnum)
values
(754710, 'I', 'Smith', 'Jack', '218984751', 'Good Insurance', 36571),
(754711, 'I', 'Will', 'Lily', '218984752', 'Better Insurance', 36572),
(754712, 'I', 'Wade', 'Micheal', '218984753', 'Awesome Insurance', 36573),
(754713, 'I', 'Green', 'Tom', '218984754', 'Cool Insurance', 36574),
(754714, 'I', 'Gates', 'Jessie', '218984755', 'Nice Insurance', 36575);

-- insert data to corporate
insert into zzz_corporate (custid, custtype, corpname, regnum, empid)
values
(754715, 'C', 'Amazon', 'amz123', 'amz456'),
(754716, 'C', 'Google', 'gg123', 'gg456'),
(754717, 'C', 'Meta', 'fb123', 'fb456'),
(754718, 'C', 'Apple', 'apple123', 'apple456'),
(754719, 'C', 'Linkedin', 'ln123', 'ln456');

-- insert data to discounts
insert into zzz_discount (discid, disctype, discpercen)
values
(84654, 'I', 30),
(84655, 'I', 34),
(84656, 'I', 38),
(84657, 'I', 20),
(84658, 'I', 21),
(84659, 'I', 30),
(84660, 'I', 39),
(94654, 'C', 40),
(94655, 'C', 44),
(94656, 'C', 48),
(94657, 'C', 30),
(94658, 'C', 31),
(94659, 'C', 40),
(94660, 'C', 49);

-- insert data to corp
insert into zzz_corp (discid, disctype, setnum)
values
(94654, 'C', 0146731),
(94655, 'C', 0146757),
(94656, 'C', 0158931),
(94657, 'C', 0187190),
(94658, 'C', 0167880),
(94659, 'C', 0778906),
(94660, 'C', 0167890);
```

```sql
-- insert data to indi
insert into zzz_indi (discid, disctype, validstart, validend, coupnum)
values
(84654, 'I', '2022-03-03', '2022-03-31', 9675731),
(84655, 'I', '2022-02-01', '2022-03-01', 5760897),
(84656, 'I', '2019-05-31', '2019-06-30', 0903232),
(84657, 'I', '2021-07-01', '2021-07-31', 5881648),
(84658, 'I', '2021-11-05', '2021-12-01', 5891678),
(84659, 'I', '2022-03-05', '2022-03-15', 7689352),
(84660, 'I', '2022-03-05', '2022-03-15', 7908112);


Commit;
```

Database Trigger Codes

There are several scenarios in how to generate the final amount in invoice. Basically, our formula for calculating the total amount of the rental service is as follows

$$ta = (ra + oa) * p$$

The percentage of the amount after using discount p

- $p = 1$ when a customer does not use discount.
- $p = 1 - dp$ when customer uses discount, more specifically
  - For a individual customer, dp is the percentage given by the coupon.
  - For a corporate customer, dp is a fixed value offered to different corporations.

The odometers limit lim for the rental service

$$lim = (ed - sd) * ol$$

For mileages less than lim, the rental office would charge

$$ra = (ed - sd) * dr$$

If the customers uses the car within the odometer limit, there is no extra fee then

$$oa = 0$$

While if the mileages exceeds lim, the rental office would charge additional fee

$$oa = (ed - sd - lim) * or$$

There are some special orders that the rental service is with unlimited mileage. For such orders, the total amount will be calculated as

$$ta = ra * p$$

otherwise,

$$ta = (ra + oa) * p$$

where **ta** is total amount aka. final amount in invoice, **ra** is regular amount that is the amount rental office would charge the customer if he/she uses the car within the odometers limits, **oa** is over amount that the customer would pay if the odometers exceed the limit, **p** is the percentage of the amount after using discount, **dp** is the discount percentage, **sd** is order's start date, **ed** is order's end date, **dr** is rent rate per day, **or** rental service for miles exceeds odometer limits per day, **ol** odometers limit per day, **lim** odometers limit for the whole service, **so** odometers when the customer starts the service, **eo** odometers when the customer ends the service.

The following are the trigger codes we added to the order table for generating the invoice. We also provide five test cases to validate if the trigger is activated properly. Find details in the comments. Figure 3 and 4 gives the results in the Table order and invoice.

```sql
-- Trigger used to generate invoice
delimiter |
CREATE TRIGGER AFTER UPDATE ON zzz_order
    FOR EACH ROW
    BEGIN
        IF NOT(NEW.endOdo <=> OLD.endOdo) THEN
            -- Calcuate amount before discount
            Set @RegularAmount := DATEDIFF(new.endDate, new.startDate) * (select
dailyrate from zzz_car where new.carid = zzz_car.carid);
            Set @OverAmount := 0;
            Set @LimitOdo := DATEDIFF(new.endDate, new.startDate) * new.OdoLimit;
            IF new.endOdo - new.startOdo > @LimitOdo THEN
                Set @OverAmount := (new.endOdo - new.startOdo - @LimitOdo)
                                    * (select overrate from zzz_car where new.carid
= zzz_car.carid);
            END IF;

            -- Calcuate discount
            Set @DiscountPerc = 1;
            IF NOT(NEW.disctype <=> NULL) THEN
                IF (NEW.disctype = 'I') THEN
                    Set @ValidStart := (select validstart from zzz_indi where
new.discid = zzz_indi.discid);
                    Set @ValidEnd := (select validend from zzz_indi where
new.discid = zzz_indi.discid);
                    IF (@ValidStart < new.enddate AND @ValidEnd > new.endDate) THEN
                        Set @DiscountPerc = 1 - 0.01 * (select discpercen from
zzz_discount where new.discid = zzz_discount.discid);
                    END IF;
                ELSEIF (NEW.disctype = 'C') THEN
                    Set @DiscountPerc = 1 - 0.01 * (select discpercen from
zzz_discount where new.discid = zzz_discount.discid);
                END IF;
            END IF;

            -- Insert the new record to invoice
            IF (NEW.OdoLimit <=> NULL) THEN
                INSERT INTO zzz_invoice(invoiceid, invoicedate, amount, orderid)
values
                    (NEW.orderid, CURDATE(), @RegularAmount, NEW.orderid);
            ELSE
                INSERT INTO zzz_invoice(invoiceid, invoicedate, amount, orderid)
values
                    (NEW.orderid, CURDATE(), (@RegularAmount + @OverAmount) *
@DiscountPerc, NEW.orderid);
            END IF;
```

```
        END IF;
    END;
|
delimiter ;
```

```
-- Add to order records
insert into zzz_order (orderid, startodo, endodo, odolimit, startdate, enddate,
custid, custtype, pickup, dropoff, carid, discid, disctype)
values
(1234568, 1000, null, 40, '2020-03-01', '2020-03-05', 754710, 'I', 2, 3, 1, 84654,
null),
(2234568, 1000, null, 40, '2020-03-01', '2020-03-05', 754710, 'I', 2, 3, 1, 84654,
null),
(3234568, 1000, null, null, '2020-03-01', '2020-03-05', 754710, 'I', 2, 2, 1,
84657, null),
(4234568, 1000, null, 40, '2020-03-01', '2020-03-05', 754710, 'I', 2, 3, 1, 84654,
'I'),
(5234568, 1000, null, 40, '2021-07-01', '2021-07-05', 754710, 'I', 2, 3, 1, 84657,
'I'),
(6234568, 1000, null, 40, '2021-07-01', '2021-07-05', 754710, 'C', 2, 3, 1, 94654,
'C');
```

```
select * from zzz_order;
```

```
-- Update enodo and trigger UpdateInvoice to generate invoice
```

```
update zzz_order set endodo=1100 where orderid=1234568; -- 4
-- Here we have endOdometer(1100) - startOdometer(1000) = 100 as total miles with 4
days traveling. Since the daily odolimit is 40, the total limit for 4 days is 160,
the total miles(100) is smaller than the total limit(160). Therefore, only regular
amount is charged:
-- 4 days * 1$ daily rate = 4$.
```

```
update zzz_order set endodo=1500 where orderid=2234568; -- 17004
-- In this situation, an individual customer has endOdometer(1500) -
startOdometer(1000) = 500 as total miles with 4 days traveling. Since the daily
odolimit is 40, the total limit for 4 days is 160, the total miles(500) is larger
than the total limit(160). Therefore, regular amount and over amount are charged:
-- 4 days * 1$ daily rate + (500 miles - 160 limits) * 50 = 17004$.
```

```
update zzz_order set endodo=1500 where orderid=3234568; -- 4
-- In this situation, an individual customer has endOdometer(1500) -
startOdometer(1000) = 500 as total miles with 4 days traveling. Since here the
odolimit part is set as null, which means that the rental service has unlimited
mileage. Therefore, only a regular amount is charged:
-- 4 days * 1$ daily rate = 4$.
```

```sql
update zzz_order set endodo=1500 where orderid=4234568; -- 17004
-- In this situation, an individual customer has endOdometer(1500) -
startOdometer(1000) =  500 as total miles with 4 days traveling. Since the daily
odolimit is 40, the total limit for 4 days is 160, the total miles(500) is larger
than the total limit(160). Therefore, regular amounts and over amounts are charged.
The customer has also used a discount coupon with ID 84657 and type 'I', after
check on the validStart date(2022-03-03) and ValidEnd date(2022-03-30),
orderEndDate(2021-03-05) is not in the range of valid dates, therefore the coupon
is invalid, the discount is not applied.
-- 4 days * 1$ daily rate + (500 miles - 160 limits) * 50 = 17004$.

update zzz_order set endodo=1500 where orderid=5234568; -- 17004 * 0.8 = 13603.2
-- In this situation, an individual customer has endOdometer(1500) -
startOdometer(1000) =  500 as total miles with 4 days traveling. Since the daily
odolimit is 40, the total limit for 4 days is 160, the total miles(500) is larger
than the total limit(160). Therefore, regular amounts and over amounts are charged.
The customer has also used a discount coupon with ID 84654 and type 'I', after
check on the validStart date(2021-07-01) and ValidEnd date(2021-07-31),
orderEndDate(2021-07-05) is in the range of valid dates, therefore the coupon is
valid, the discount is applied with discount percentage 0.8.
-- (4 days * 1$ daily rate + (500 miles - 160 limits) * 50) * 0.8 = 13603.2$.

update zzz_order set endodo=1500 where orderid=6234568; -- 17004 * 0.6 = 10202.4
-- In this situation, a corporate customer has endOdometer(1500) -
startOdometer(1000) = 500 as total miles with 4 days traveling. Since the daily
odolimit is 40, the total limit for 4 days is 160, the total miles(500) is larger
than the total limit(160). Therefore, regular amounts and over amounts are charged.
The customer has also used a discount coupon with ID 84654 and type 'I', Since
corporate customers irrespective of date of rental service, the discount is applied
with discount percentage 0.8.
-- (4 days * 1$ daily rate + (500 miles - 160 limits) * 50) * 0.6 = 10202.4$.
```

orderid	startodo	endodo	odolimit	startdate	enddate	custid	custtype	pickup	dropoff	carid	discid	disctype
1234568	1000	1100	40	2020-03-01 00:00:00	2020-03-05 00:00:00	754710	I	2	3	1	84654	NULL
2234568	1000	1500	40	2020-03-01 00:00:00	2020-03-05 00:00:00	754710	I	2	3	1	84654	NULL
3234568	1000	NULL	NULL	2020-03-01 00:00:00	2020-03-05 00:00:00	754710	I	2	2	1	84657	NULL
4234568	1000	1500	40	2020-03-01 00:00:00	2020-03-05 00:00:00	754710	I	2	3	1	84654	I
5234568	1000	1500	40	2021-07-01 00:00:00	2021-07-05 00:00:00	754710	I	2	3	1	84657	I
6234568	1000	1500	40	2021-07-01 00:00:00	2021-07-05 00:00:00	754710	I	2	3	1	94654	C
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 3: Records in Table order.

invoiceid	invoicedate	amount	orderid
1234568	2022-04-08 00:00:00	4	1234568
2234568	2022-04-08 00:00:00	17004	2234568
4234568	2022-04-08 00:00:00	17004	4234568
5234568	2022-04-08 00:00:00	13603	5234568
6234568	2022-04-08 00:00:00	10202	6234568
NULL	NULL	NULL	NULL

Figure 4: Records in Table invoice, which are generated once we have data updated in column endodo in Table order

carid	cartype	dailyrate	overrate	officeid	vin
1	small car	1.00	50.00	12345	299J98JSJIW19923I
2	mid-size car	78.28	10.20	22345	28WHD IW02U929282

Figure 5: The record in Table car used in the validation cases

Results of queries

```sql
-- List of Tables
select table_name
from information_schema.tables
where table_name like 'ZZZ%'
order by table_name;
```

Table 2: List of Tables.

table_name
zzz_address
zzz_address
zzz_car
zzz_car
zzz_corp
zzz_corporate
zzz_corporate
zzz_customer
zzz_customer
zzz_discount
zzz_indi
zzz_individual
zzz_individual
zzz_invoice
zzz_invoice
zzz_office
zzz_office
zzz_order
zzz_order
zzz_payment
zzz_payment
zzz_vehicle
zzz_vehicle

```sql
-- List of Table Columns
select table_name, column_name, column_type
from information_schema.columns
where table_name like 'ZZZ%'
order by table_name, column_name;
```

Table 3: List of Table columns.

table_name	column_name	column_type
zzz_address	addrid	bigint(20)
zzz_address	addrid	bigint(20)
zzz_address	country	varchar(30)
zzz_address	country	varchar(30)
zzz_address	state	varchar(30)
zzz_address	state	varchar(30)
zzz_address	street	varchar(30)
zzz_address	street	varchar(30)
zzz_address	zipcode	int(11)
zzz_address	zipcode	int(11)
zzz_car	carid	int(11)
zzz_car	carid	int(11)
zzz_car	cartype	varchar(20)
zzz_car	cartype	varchar(20)
zzz_car	dailyrate	decimal(5,2)
zzz_car	dailyrate	decimal(5,2)
zzz_car	officeid	int(11)
zzz_car	officeid	int(11)
zzz_car	overrate	decimal(5,2)
zzz_car	overrate	decimal(5,2)
zzz_car	vin	varchar(20)
zzz_car	vin	varchar(20)
zzz_corp	discid	bigint(20)
zzz_corp	disctype	char(1)
zzz_corp	setnum	decimal(20,0)
zzz_corporate	corpname	varchar(20)
zzz_corporate	custid	int(11)
zzz_corporate	custid	int(11)
zzz_corporate	custtype	char(1)
zzz_corporate	empid	varchar(20)
zzz_corporate	empid	varchar(20)
zzz_corporate	name	varchar(20)
zzz_corporate	regnum	varchar(20)
zzz_corporate	regnum	varchar(20)
zzz_customer	addrid	bigint(20)
zzz_customer	addrid	bigint(20)

zzz_customer	custid	int(11)
zzz_customer	custid	int(11)
zzz_customer	custtype	char(1)
zzz_customer	custtype	char(1)
zzz_customer	email	varchar(20)
zzz_customer	email	varchar(20)
zzz_customer	phone	varchar(20)
zzz_customer	phone	varchar(20)
zzz_discount	discid	bigint(20)
zzz_discount	discpercen	decimal(5,2)
zzz_discount	disctype	char(1)
zzz_indi	coupnum	varchar(10)
zzz_indi	discid	bigint(20)
zzz_indi	disctype	char(1)
zzz_indi	validend	datetime
zzz_indi	validstart	datetime
zzz_individual	custid	int(11)
zzz_individual	custid	int(11)
zzz_individual	custtype	char(1)
zzz_individual	fname	varchar(20)
zzz_individual	fname	varchar(20)
zzz_individual	insname	varchar(20)
zzz_individual	insname	varchar(20)
zzz_individual	insnum	decimal(20,0)
zzz_individual	insnum	decimal(20,0)
zzz_individual	licensenum	bigint(20)
zzz_individual	licensenum	bigint(20)
zzz_individual	lname	varchar(20)
zzz_individual	lname	varchar(20)
zzz_invoice	amount	int(11)
zzz_invoice	amount	int(11)
zzz_invoice	Date	datetime
zzz_invoice	invoicedate	datetime
zzz_invoice	invoiceid	bigint(20)
zzz_invoice	invoiceid	bigint(20)
zzz_invoice	orderid	bigint(20)
zzz_invoice	orderid	bigint(20)
zzz_office	addrid	bigint(20)
zzz_office	addrid	bigint(20)

zzz_office	officeid	int(11)
zzz_office	officeid	int(11)
zzz_office	phone	varchar(20)
zzz_office	phone	varchar(20)
zzz_order	carid	int(11)
zzz_order	custid	int(11)
zzz_order	custid	int(11)
zzz_order	custtype	char(1)
zzz_order	discid	bigint(20)
zzz_order	disctype	char(1)
zzz_order	dropoff	bigint(20)
zzz_order	dropoff	bigint(20)
zzz_order	enddate	datetime
zzz_order	enddate	datetime
zzz_order	endodo	bigint(20)
zzz_order	endodo	bigint(20)
zzz_order	invoiceid	bigint(20)
zzz_order	odolimit	bigint(20)
zzz_order	odolimit	bigint(20)
zzz_order	orderid	bigint(20)
zzz_order	orderid	bigint(20)
zzz_order	pickup	bigint(20)
zzz_order	pickup	bigint(20)
zzz_order	startdate	datetime
zzz_order	startdate	datetime
zzz_order	startodo	bigint(20)
zzz_order	startodo	bigint(20)
zzz_payment	amount	int(11)
zzz_payment	amount	int(11)
zzz_payment	cardnum	decimal(20,0)
zzz_payment	cardnum	decimal(20,0)
zzz_payment	Date	datetime
zzz_payment	invoiceid	bigint(20)
zzz_payment	invoiceid	bigint(20)
zzz_payment	method	varchar(20)
zzz_payment	method	varchar(20)
zzz_payment	paymdate	datetime
zzz_payment	paymid	bigint(20)
zzz_payment	paymid	bigint(20)

zzz_vehicle	lpn	varchar(15)
zzz_vehicle	lpn	varchar(15)
zzz_vehicle	make	varchar(30)
zzz_vehicle	make	varchar(30)
zzz_vehicle	model	varchar(30)
zzz_vehicle	model	varchar(30)
zzz_vehicle	vin	varchar(20)
zzz_vehicle	vin	varchar(20)
zzz_vehicle	year	smallint(6)
zzz_vehicle	year	smallint(6)

```sql
-- List of Table Column Constraints
select table_name, constraint_name, constraint_schema, referenced_table_name,
referenced_column_name
from information_schema.key_column_usage
where table_name like 'ZZZ%'
order by table_name;
```

Table 4: List of Table columns constraints.

table_name	column_name	constraint_name	constraint_schema	referenced_table_name	referenced_column_name
zzz_address	addrid	PRIMARY	Test	NULL	NULL
zzz_address	addrid	PRIMARY	WOW	NULL	NULL
zzz_car	vin	zzz_car__idx	Test	NULL	NULL
zzz_car	vin	zzz_car_zzz_vehicle_fk	Test	zzz_vehicle	vin
zzz_car	vin	zzz_car__idx	WOW	NULL	NULL
zzz_car	vin	zzz_car_zzz_vehicle_fk	WOW	zzz_vehicle	vin
zzz_car	carid	PRIMARY	Test	NULL	NULL
zzz_car	officeid	zzz_car_zzz_office_fk	Test	zzz_office	officeid
zzz_car	carid	PRIMARY	WOW	NULL	NULL
zzz_car	officeid	zzz_car_zzz_office_fk	WOW	zzz_office	officeid
zzz_corp	disctype	PRIMARY	WOW	NULL	NULL
zzz_corp	disctype	zzz_corp_zzz_discount_fk	WOW	zzz_discount	disctype
zzz_corp	discid	PRIMARY	WOW	NULL	NULL
zzz_corp	discid	zzz_corp_zzz_discount_fk	WOW	zzz_discount	discid
zzz_corporate	custid	PRIMARY	WOW	NULL	NULL
zzz_corporate	custid	zzz_corporate_zzz_customer_fk	WOW	zzz_customer	custid
zzz_corpora	custid	zzz_corporate_zzz_custome	Test	zzz_customer	custid

te		r_fk			
zzz_corporate	custtype	PRIMARY	WOW	NULL	NULL
zzz_corporate	custid	PRIMARY	Test	NULL	NULL
zzz_corporate	custtype	zzz_corporate_zzz_customer_fk	WOW	zzz_customer	custtype
zzz_customer	custid	PRIMARY	WOW	NULL	NULL
zzz_customer	addrid	zzz_customer__idx	Test	NULL	NULL
zzz_customer	addrid	zzz_customer__idx	WOW	NULL	NULL
zzz_customer	custid	PRIMARY	Test	NULL	NULL
zzz_customer	custtype	PRIMARY	WOW	NULL	NULL
zzz_customer	addrid	zzz_customer_zzz_address_fk	Test	zzz_address	addrid
zzz_customer	addrid	zzz_customer_zzz_address_fk	WOW	zzz_address	addrid
zzz_discount	disctype	PRIMARY	WOW	NULL	NULL
zzz_discount	discid	PRIMARY	WOW	NULL	NULL
zzz_indi	discid	PRIMARY	WOW	NULL	NULL
zzz_indi	discid	zzz_indi_zzz_discount_fk	WOW	zzz_discount	discid
zzz_indi	disctype	PRIMARY	WOW	NULL	NULL
zzz_indi	disctype	zzz_indi_zzz_discount_fk	WOW	zzz_discount	disctype
zzz_individual	custtype	PRIMARY	WOW	NULL	NULL
zzz_individual	custtype	zzz_individual_zzz_customer_fk	WOW	zzz_customer	custtype
zzz_individual	custid	PRIMARY	Test	NULL	NULL
zzz_individual	custid	PRIMARY	WOW	NULL	NULL
zzz_individual	custid	zzz_individual_zzz_customer_fk	WOW	zzz_customer	custid
zzz_individual	custid	zzz_individual_zzz_customer_fk	Test	zzz_customer	custid
zzz_invoice	orderid	zzz_invoice_zzz_order_fk	WOW	zzz_order	orderid
zzz_invoice	invoiceid	PRIMARY	Test	NULL	NULL
zzz_invoice	orderid	zzz_invoice_zzz_order_fk	Test	zzz_order	orderid
zzz_invoice	invoiceid	PRIMARY	WOW	NULL	NULL
zzz_invoice	orderid	zzz_invoice__idx	Test	NULL	NULL

zzz_office	officeid	PRIMARY	Test	NULL	NULL
zzz_office	addrid	zzz_office_zzz_address_fk	Test	zzz_address	addrid
zzz_office	addrid	zzz_office__idx	WOW	NULL	NULL
zzz_office	addrid	zzz_office__idx	Test	NULL	NULL
zzz_office	officeid	PRIMARY	WOW	NULL	NULL
zzz_office	addrid	zzz_office_zzz_address_fk	WOW	zzz_address	addrid
zzz_order	carid	zzz_order_zzz_car_fk	WOW	zzz_car	carid
zzz_order	custtype	zzz_order_zzz_customer_fk	WOW	zzz_customer	custtype
zzz_order	invoiceid	zzz_order__idx	Test	NULL	NULL
zzz_order	discid	zzz_order_zzz_discount_fk	WOW	zzz_discount	discid
zzz_order	pickup	zzz_order_zzz_address_fkv2	Test	zzz_address	addrid
zzz_order	invoiceid	zzz_order_zzz_invoice_fk	Test	zzz_invoice	invoiceid
zzz_order	orderid	PRIMARY	WOW	NULL	NULL
zzz_order	discid	zzz_order__idx	WOW	NULL	NULL
zzz_order	pickup	zzz_order_zzz_address_fkv2	WOW	zzz_address	addrid
zzz_order	custid	zzz_order_zzz_customer_fk	WOW	zzz_customer	custid
zzz_order	orderid	PRIMARY	Test	NULL	NULL
zzz_order	disctype	zzz_order_zzz_discount_fk	WOW	zzz_discount	disctype
zzz_order	dropoff	zzz_order_zzz_address_fk	Test	zzz_address	addrid
zzz_order	custid	zzz_order_zzz_customer_fk	Test	zzz_customer	custid
zzz_order	disctype	zzz_order__idx	WOW	NULL	NULL
zzz_order	dropoff	zzz_order_zzz_address_fk	WOW	zzz_address	addrid
zzz_payment	invoiceid	zzz_payment_zzz_invoice_fk	WOW	zzz_invoice	invoiceid
zzz_payment	invoiceid	zzz_payment_zzz_invoice_fk	Test	zzz_invoice	invoiceid
zzz_payment	paymid	PRIMARY	WOW	NULL	NULL
zzz_payment	paymid	PRIMARY	Test	NULL	NULL
zzz_vehicle	vin	PRIMARY	WOW	NULL	NULL
zzz_vehicle	vin	PRIMARY	Test	NULL	NULL

```sql
-- List of Table Column Comments
select table_name, column_name, column_comment
from information_schema.columns
where table_name like 'ZZZ%'
order by table_name;
```

Table 5: List of Table column comments.

table_name	column_name	column_comment
zzz_address	addrid	Unique ID for address.
zzz_address	street	Street info for address.
zzz_address	state	State info for address.
zzz_address	country	Country info for address.
zzz_address	zipcode	Zipcode for address.
zzz_address	addrid	Unique ID for address.
zzz_address	street	Street info for address.
zzz_address	state	State info for address.
zzz_address	country	Country info for address.
zzz_address	zipcode	Zipcode for address.
zzz_car	carid	Unique ID for each car.
zzz_car	cartype	Class of the car.
zzz_car	dailyrate	Regular rental rate per day of the rental service for the car.
zzz_car	overrate	Extra fees per mile that exceeds the limit.
zzz_car	officeid	
zzz_car	vin	
zzz_car	carid	Unique ID for each car.
zzz_car	cartype	Class of the car.
zzz_car	dailyrate	Regular rental rate per day of the rental service for the car.
zzz_car	overrate	Extra fees per mile that exceeds the limit.
zzz_car	officeid	
zzz_car	vin	
zzz_corp	discid	Uinque ID for discount.
zzz_corp	disctype	Discriminator of discount type.
zzz_corp	setnum	Number for indentifying corporation
zzz_corporate	custid	Unique ID for customer.
zzz_corporate	name	Corporate customer's name.
zzz_corporate	regnum	Registration number of the corporation.
zzz_corporate	empid	Employee ID of the customer who rents the car on a corporate account.
zzz_corporate	custid	Unique ID for customer.
zzz_corporate	custtype	Customer type.
zzz_corporate	corpname	Corporation's name.

zzz_corporate	regnum	Registration number of the corporation.
zzz_corporate	empid	Employee ID of the customer who rents the car on a corporate account.
zzz_customer	custid	Unique ID for customer.
zzz_customer	email	Email address for customer.
zzz_customer	phone	Phone number for customers.
zzz_customer	custtype	Customer type.
zzz_customer	addrid	
zzz_customer	custid	Unique ID for customer.
zzz_customer	custtype	Customer type.
zzz_customer	email	Email address for customer.
zzz_customer	phone	Phone number for customers.
zzz_customer	addrid	
zzz_discount	discid	Uinque ID for discount.
zzz_discount	disctype	Discriminator of discount type.
zzz_discount	discpercen	Discount percentage %.
zzz_indi	discid	Uinque ID for discount.
zzz_indi	disctype	Discriminator of discount type.
zzz_indi	coupnum	Coupon number.
zzz_indi	validstart	Coupon valid start date.
zzz_indi	validend	Coupon valid end date.
zzz_individual	custid	Unique ID for customer.
zzz_individual	lname	Last name for individual customer.
zzz_individual	fname	First Name for individual customer.
zzz_individual	licensenum	Drive license number.
zzz_individual	insname	Insurance company name.
zzz_individual	insnum	Insurance policy number.
zzz_individual	custid	Unique ID for customer.

zzz_individual	custtype	Customer type.
zzz_individual	lname	Last name for individual customer.
zzz_individual	fname	First Name for individual customer.
zzz_individual	licensenum	Drive license number.
zzz_individual	insname	Insurance company name.
zzz_individual	insnum	Insurance policy number.
zzz_invoice	invoiceid	Unique ID for invoice.
zzz_invoice	Date	Invoice data.
zzz_invoice	amount	Invoice amount.
zzz_invoice	orderid	
zzz_invoice	invoiceid	Unique ID for invoice.
zzz_invoice	invoicedate	Invoice data.
zzz_invoice	amount	Invoice amount.
zzz_invoice	orderid	
zzz_office	officeid	Unique ID for office.
zzz_office	phone	Office's phone number.
zzz_office	addrid	
zzz_office	officeid	Unique ID for office.
zzz_office	phone	Office's phone number.
zzz_office	addrid	
zzz_order	orderid	Unique ID for each order.
zzz_order	startodo	Start odometer.
zzz_order	endodo	End Odometer.
zzz_order	odolimit	Daily odometer limit for the rental service.
zzz_order	startdate	Date when the customer starts the service.
zzz_order	enddate	Date when the customer ends the service.
zzz_order	invoiceid	
zzz_order	custid	
zzz_order	pickup	
zzz_order	dropoff	
zzz_order	orderid	Unique ID for each order.
zzz_order	startodo	Start odometer.
zzz_order	endodo	End Odometer.
zzz_order	odolimit	Daily odometer limit for the rental service.

zzz_order	startdate	Date when the customer starts the service.
zzz_order	enddate	Date when the customer ends the service.
zzz_order	custid	
zzz_order	custtype	
zzz_order	pickup	
zzz_order	dropoff	
zzz_order	carid	
zzz_order	discid	
zzz_order	disctype	
zzz_payment	paymid	Unique ID for payment.
zzz_payment	method	Payment method.
zzz_payment	amount	Payment amount.
zzz_payment	Date	Payment date.
zzz_payment	cardnum	Card number.
zzz_payment	invoiceid	
zzz_payment	paymid	Unique ID for payment.
zzz_payment	method	Payment method.
zzz_payment	amount	Payment amount.
zzz_payment	paymdate	Payment date.
zzz_payment	cardnum	Card number.
zzz_payment	invoiceid	
zzz_vehicle	vin	Vehicle identification number.
zzz_vehicle	make	Brand of the vehicle.
zzz_vehicle	model	Name of a product or a range of products.
zzz_vehicle	year	Manufacture year of the vehicle.
zzz_vehicle	lpn	The registration identifier is a numeric or alphanumeric ID that uniquely identifies the vehicle or vehicle owner within the issuing region's vehicle register.
zzz_vehicle	vin	Vehicle identification number.
zzz_vehicle	make	Brand of the vehicle.
zzz_vehicle	model	Name of a product or a range of products.
zzz_vehicle	year	Manufacture year of the vehicle.
zzz_vehicle	lpn	The registration identifier is a numeric or alphanumeric ID that uniquely identifies the vehicle or vehicle owner within the issuing region's vehicle register.