Tesla Automobile Database System

CS 6360.003- Database Design Project Report

Rajashree, Kamalakannan RXK180081
Sriprasath, Gujuluva Parthasarathy SXG180154
Siddhartha, Kasturi SXK180149

Fall 2019

Contents

INTRODUCTION

Tesla is one of the leading automotive and energy based company. Tesla's mission is to accelerate the world's transition to sustainable energy. It builds not only all-electric vehicles but also infinitely scalable clean energy generation. The web portal enables customer to place order (customized order) on vehicles, accessories, batteries and solar panels.

The objective of this project is to design a database for the TESLA System. A research was conducted to analyze various departments like procurement, production planning, manufacturing, logistics and marketing. We have decided to design the manufacturing sector of the company since that will be the most challenging part which deals with processing orders, maintaining inventory and warehouse while manufacturing vehicles. It was found out that the fundamental requirements of such a manufacturing system will be entities namely, Orders, vehicle, parts, Plant, supplier, customer, employee, vehicle types. Each entity should have a primary key and relations to other entities. There should also be entities such as Plant order and parts order. The flow of the system is when an order is placed by a customer, it should automatically check whether a match is present in any of the warehouse vehicles and if a match is found, then it can be delivered automatically by assigning the vehicle id to the related order. If a match is not found then the status is changed to in-progress and is routed to one of the plant. The plant, at first acquires the full set of specifications from the list of vehicle types and then this full set is checked against the inventory of that plant whether all the parts are available for processing that orders. If yes, it is assumed that the vehicle is manufactured and is assigned a vehicle id. The used parts are captured in the log table and the current sets of available unused parts are always maintained in the inventory. If a part is not present in the inventory then the plant places a part order to the supplier. One of the suppliers delivers the required parts back to the plant's inventory. When it comes to bulk production, the plant itself places a separate order which again checks the full set of specification with the given type of vehicle and follows the same procedure as a customer order (except for the initial step i.e., to check the warehouse). Employee is assigned to take care of customer orders and an employee can also act as a supervisor for a plant. Whenever the rating process takes place, the salary should be updated based on the designation and rating. The logs are also maintained with respect to each of the parts used in each of the vehicle and the order it was to delivered to with the aim to pull safety reports in case of any accident reported cases or issues.

Hence purpose of the design is as follows:

- A Customer can place an order with the listed set of specifications.
- Employee manages the plant and also handles the orders that are placed by the customer.
- Each Plant handles both the customer order and the orders that are placed by the company. Also it places orders to the supplier for parts if required.
- Each Plant has an inventory which has the parts that are currently stored and also a warehouse where the manufactured vehicles are stored.
- Supplier handles the order that is placed by the plant and supplies the parts to the inventory.
- Vehicle type has the details of the models and the information about the parts that are associated with it.
- Orders that are initially placed by the customer are checked in the warehouse with the matching specification. If not, it will be redirected to the plant for manufacturing.

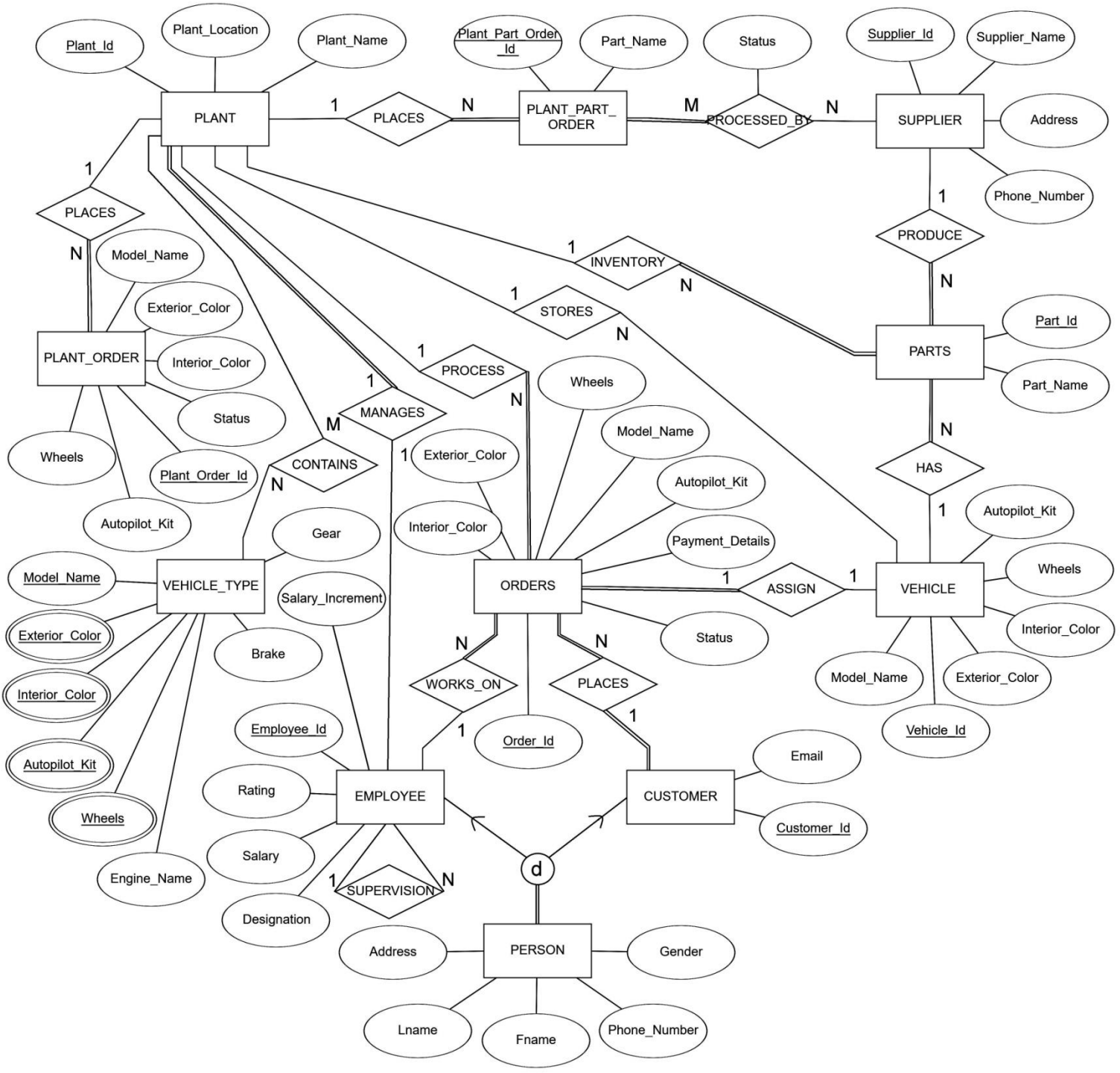
The above design is considered initially and the following sections discusses about the detailed design and implementation.

1. SPECIFICATIONS

- The first element of the system will be the CUSTOMER. Customers place order to the system with the desired specification. The customer must have CUSTOMER_ID, EMAIL, FIRST_NAME, LAST_NAME, ADDRESS, PHONE_NO, and GENDER.
- The fundamental element of the system is the ORDERS which is placed by the customer. While placing an order to the system the customer needs to provide the required set of top level specification from the given list like the MODEL_NAME, EXTERIOR_COLOR, INTERIOR_COLOR, WHEELS, and AUTOPILOT_KIT. Apart from these the order also tracks the STATUS, PAYMENT_DETAILS, EMPLOYEE_ID, PLANT_ID, CUSTOMER_ID, and VEHICLE_ID. Vehicle_id is provided after manufacturing the vehicle or if a matching vehicle is present in the inventory.
- Another important element in the system is the PLANT which has PLANT_ID, PLANT_NAME, PLANT_LOCATION handles the orders placed by the customer if a matching vehicle is not already present in the inventory, the orders placed by the plant for bulk processing. Each plant has the model types that can be manufactured in it. Each plant is also associated with an inventory and a warehouse. Each plant has a manager (MANAGER_ID) associated with it.
- VEHICLE_TYPE contains the models (MODEL_NAME) that can be manufactured and the list of specifications like EXTERIOR_COLOR, INTERIOR_COLOR, WHEELS, AUTOPILOT_KIT, ENGINE, BREAK, and GEAR.
- PLANT_ORDER also has the similar set of top level specification. This is used for bulk processing which is mostly placed by the company at the beginning.
- Each Plant has inventory which is named as the PARTS in the system. Each supplier supplies a list of parts. The PARTS which has PART_ID, PART_NAME, SUPPLIER_ID details should always have the current inventory list. Each part can be associated to a PART_ID and a supplier id. Used parts are moved to the PARTS_LOG and the entry in parts is deleted.
- Each Plant has warehouse which is named as VEHICLE in the system. Vehicle contains the manufactured vehicle with a VEHICLE_ID associated along with the specifications.
- PLANT_PART_ORDER are the orders placed by the plant when the required quantity of a particular part is not present in the inventory.
- The plant part order is processed by the SUPPLIER. In supplier, the details like SUPPLIER_ID, SUPPLIER_NAME, ADDRESS, and PHONE_NUMBER are captured.

- The system also has EMPLOYEE who manages the plant and also handles the orders. The system must capture the employee details like EMPLOYEE_ID, DESIGNATION, RATING, SUPERVISOR_ID, FIRST_NAME, LAST_NAME, ADDRESS, PHONE_NO, and GENDER.

2. ER DIAGRAM



Assumptions

- Orders can only be placed by customers and not employees.
- If all the required parts are present in the PARTS entity for an order, then it is assumed that the vehicle is manufactured and it is given a vehicle_id.
- All the suppliers supply all the parts. PART_ORDER can be placed to any of the suppliers.
- Used parts are removed from the PARTS table and it is stored in the PARTS_LOG for tracking purpose. PARTS always contain the present available parts that can be used for manufacturing.
- Each model produced by the company has a subset or the whole set of specifications listed in the VEHICLE_TYPE table.
- PLANT_ORDER is only for the orders placed by the company for bulk production.
- Each plant has an inventory (PARTS) and a warehouse (VEHICLE) associated with it.
- Delivered orders are also maintained in the system through ORDERS entity.
- Customer can either make a full payment or a half payment while placing an order.
- Exterior Color and the Interior color imply the doors, bonnet and trunk associated parts with the specified color.
- Each Part of the same type has a different part id and it is associated with the supplier id.
- Each order which is in-progress is routed to a random plant.
- PARTS_LOG and the delivered ORDERS are maintained for pulling safety reports that may be required by the government.
- Employee’s salary hike is based on the designation and rating.

One-to-one binary relationships

- Each PLANT has one EMPLOYEE as a manager and only one manager is associated with a plant.
- Each ORDER has a VEHICLE associated with it and only one vehicle is linked to an order.

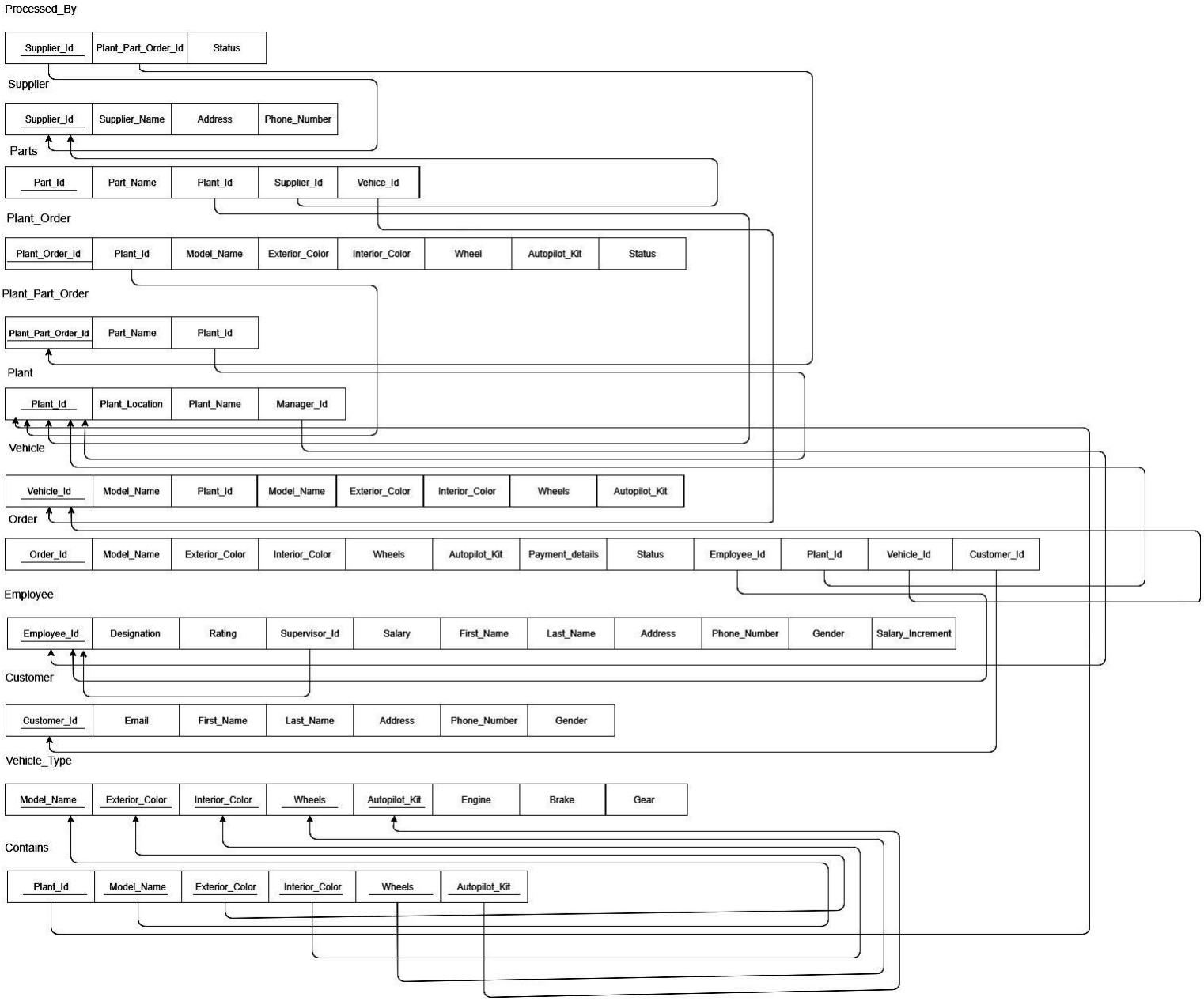
One-to-many binary relationships

- Each CUSTOMER can place multiple ORDERS. An order can belong to only one customer.
- Each EMPLOYEE can handle multiple ORDERS. An order can be handled by only on employee.
- Each ORDER belongs to only one PLANT. But a plant can handle many orders.
- Each PLANT_ORDER belongs to only one PLANT. But a plant can handle many plant_orders.
- Each PLANT_PART_ORDER belongs to only one PLANT. But a plant can handle many plant_part_orders.
- Each PLANT can have multiple PARTS (inventory). A part is associated with only one plant.
- Each PLANT can have multiple VEHICLE (warehouse). A vehicle is associated with only one plant.
- Each VEHICLE can have multiple PARTS. Each part belong to only one vehicle.
- Each SUPPLIER can supply multiple PARTS. Each part is associated with only one supplier

Many-to-many binary relationships

- A PLANT can manufacture any number of models (VEHICLE_TYPE). Also a model (VEHICLE_TYPE) can be manufactured by many plants.
- Orders (PLANT_PART_ORDER) placed by the plant can be catered by any numbers of suppliers and a supplier can process any number of orders.

3. RELATIONAL SCHEMA



4. DATABASE NORMALIZATION

Functional Dependencies for the System

PROCESSED_BY

SUPPLIER_ID → PLANT_PART_ORDER_ID, STATUS

SUPPLIER

SUPPLIER_ID → SUPPLIER_NAME , ADDRESS, PHONE_NUMBER

PARTS

PART_ID → PART_NAME, PLANT_ID, SUPPLIER_ID, VEHICLE_ID

PLANT _ORDER

PLANT_ORDER_ID → PLANT_ORDER_ID, PLANT_ID, MODEL_NAME, EXTERIOR_COLOR, INTETIOR_COLOR, WHEEL, AUTOPILOT_KIT, STATUS

PLANT_PART_ORDER

PLANT_PART_ORDER_ID → PART_NAME, PLANT_ID

PLANT

PLANT_ID → PLANT_LOCATION, PLANT_NAME STATUS, MANGER_ID

VEHICLE

VEHICLE_ID → MODEL_NAME, EXTERIOR_COLOR, INTETIOR_COLOR, WHEEL, AUTOPILOT_KIT

ORDER

ORDER_ID → MODEL_NAME, EXTERIOR_COLOR, INTETIOR_COLOR, WHEEL AUTOPILOT_KIT, PAYMENT_DETAILS, STATUS, EMPLOYEE _ID, PLANT_ID, VEHICLE_ID, CUSTOMER_ID

EMPLOYEE

EMPLOYEE_ID → DESIGNATION, RATING, SUPERVISOR_ID, SALARY, FIRST_NAME, LAST_NAME, ADDRESS, PHONE_NUMBER, GENDER, SALARY_INCREMENT

CUSTOMER

CUSTOMER_ID → EMAIL,FIRST_NAME, LAST_NAME, ADDRESS, PHONE_NUMBER, GENDER

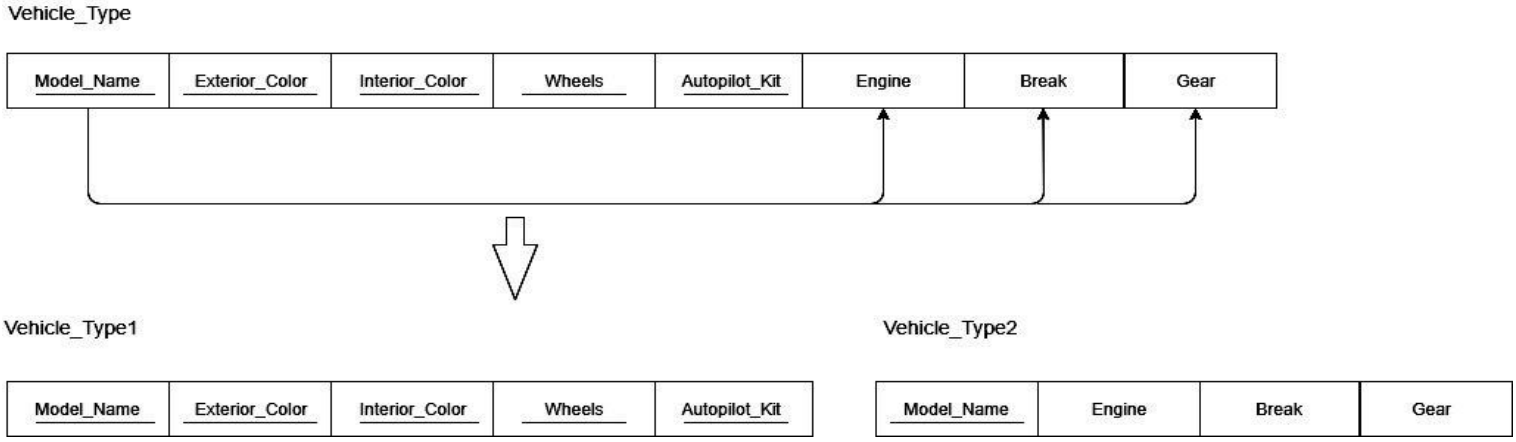
VEHICLE_TYPE

MODEL_NAME, EXTERIOR_COLOR, INTETIOR_COLOR, WHEEL, AUTOPILOT_KIT → ENGINE, BREAK, GEAR

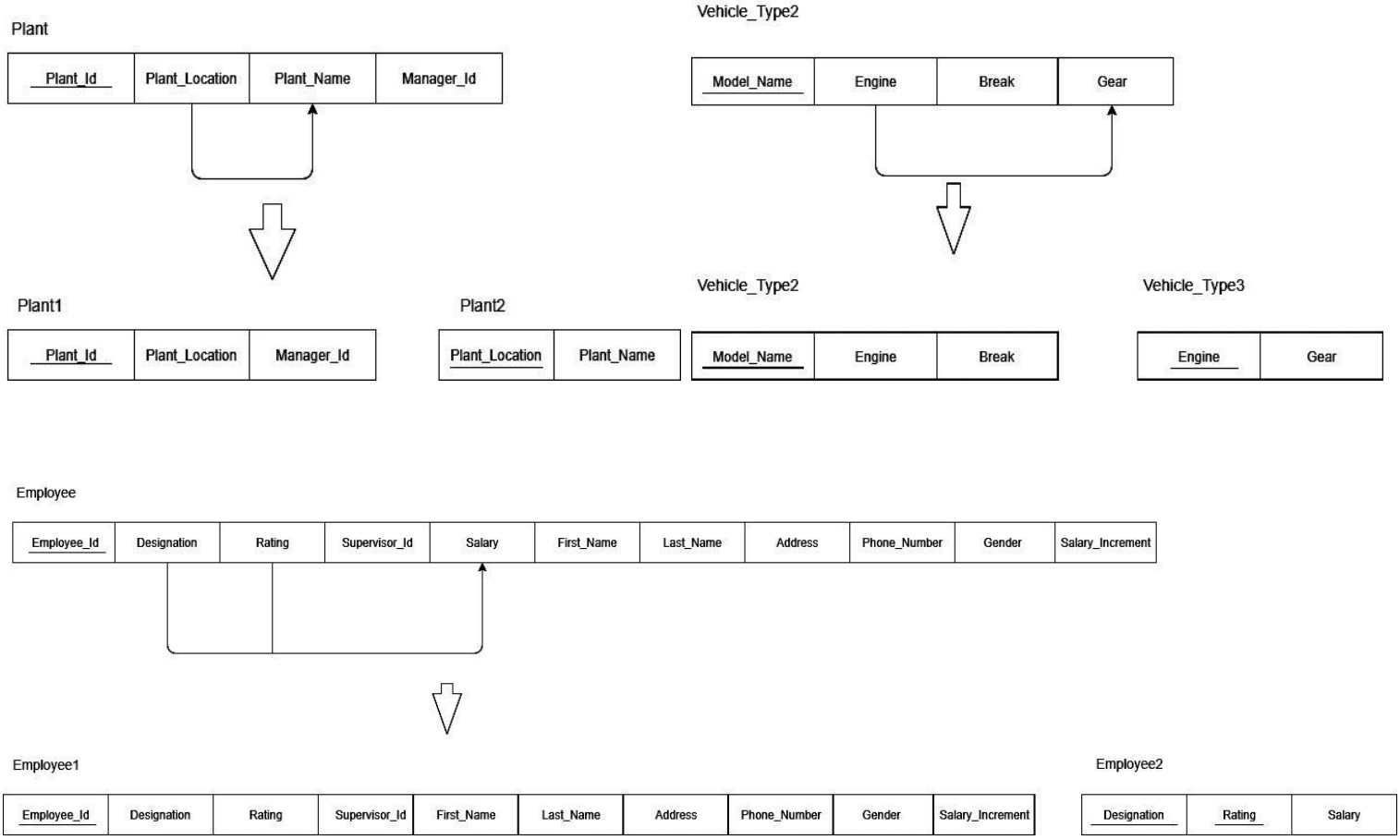
CONTAINS

PLANT_ID, MODEL_NAME EXTERIOR_COLOR, INTETIOR_COLOR, WHEEL, AUTOPILOT_KIT

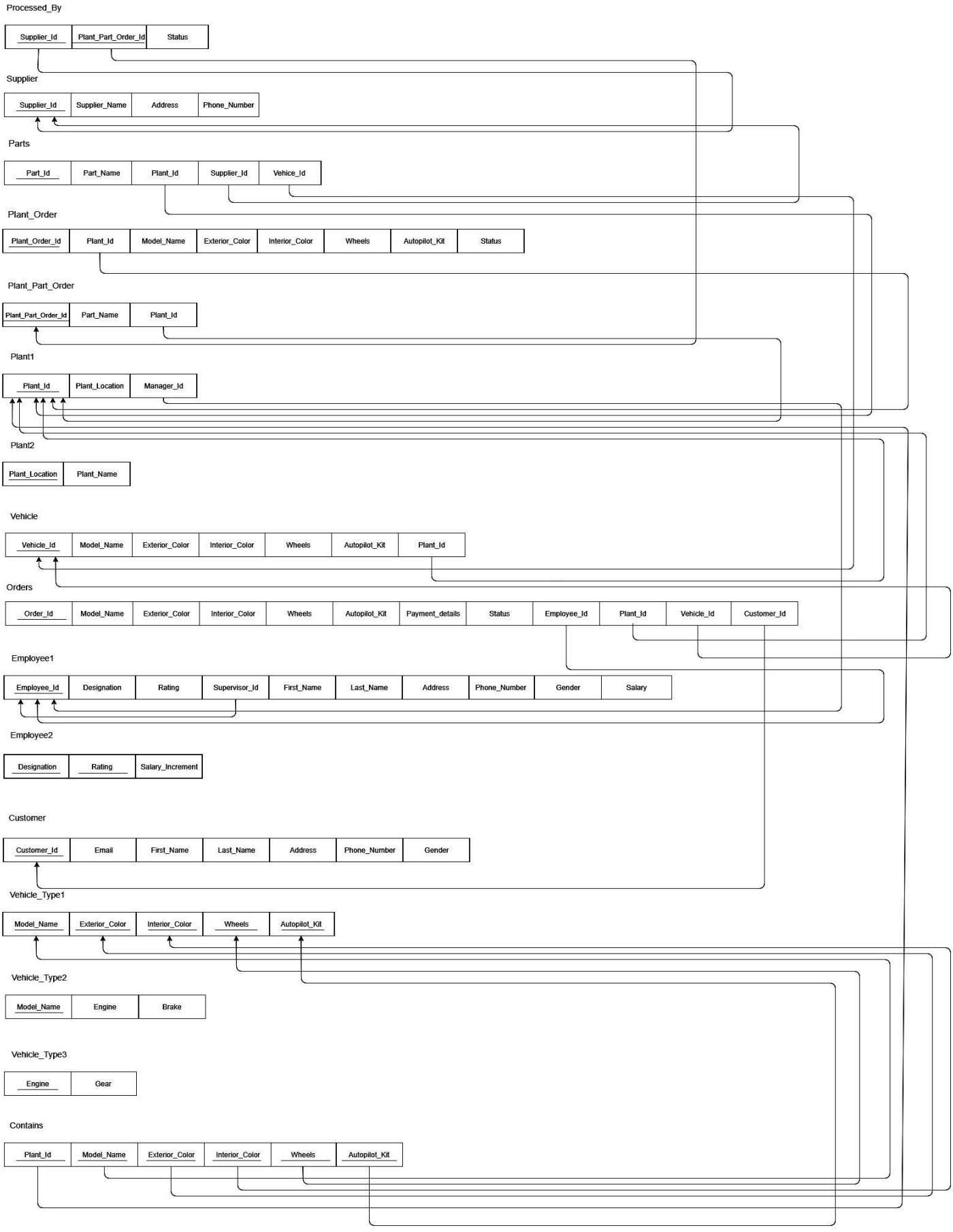
Normalization 2NF



Normalization 3NF



5. FINAL RELATIONAL SCHEMA



6. SQL CODE

DROP TABLE :-

```
DROP TABLE VEHICLE_TYPE2;
DROP TABLE VEHICLE_TYPE3;
DROP TABLE EMPLOYEE2;
DROP TABLE PLANT2;
DROP TABLE PLANT_ORDER;
DROP TABLE PARTS;
DROP TABLE PROCESSED_BY;
DROP TABLE CONTAINS;
DROP TABLE PARTS_LOG;
DROP TABLE VEHICLE_TYPE1;
DROP TABLE SUPPLIER;
DROP TABLE VEHICLE;
DROP TABLE PLANT_PART_ORDER;
DROP TABLE ORDERS;
DROP TABLE PLANT1;
DROP TABLE EMPLOYEE1;
DROP TABLE CUSTOMER;
```

CREATE TABLE :-

```
CREATE TABLE VEHICLE_TYPE2(
MODEL_NAME              VARCHAR(10),
ENGINE                  VARCHAR(20)     NOT NULL,
BRAKE                   VARCHAR(20)     NOT NULL,
CONSTRAINT PK_VEHICLE_TYPE2
PRIMARY KEY(MODEL_NAME));

CREATE TABLE VEHICLE_TYPE3(
ENGINE                  VARCHAR(20),
GEAR                    VARCHAR(20),
CONSTRAINT PK_VEHICLE_TYPE3
PRIMARY KEY(ENGINE));

CREATE TABLE EMPLOYEE1(
EMPLOYEE_ID             VARCHAR(10),
DESIGNATION             VARCHAR(20),
RATING                  INT,
SUPERVISOR_ID           VARCHAR(10),
FIRST_NAME              VARCHAR(20)     NOT NULL,
LAST_NAME               VARCHAR(20),
ADDRESS                 VARCHAR(50),
PHONE_NUMBER            INT,
GENDER                  VARCHAR(10),
SALARY                  INT,
CONSTRAINT PK_EMPLOYEE1
PRIMARY KEY(EMPLOYEE_ID),
CONSTRAINT CHECK_EMPLOYEE_GENDER
CHECK(GENDER IN('MALE','FEMALE','UNKNOWN')),
CONSTRAINT FK_EMPLOYEE1
FOREIGN KEY(SUPERVISOR_ID) REFERENCES EMPLOYEE1(EMPLOYEE_ID) ON DELETE SET NULL);
```

```
CREATE TABLE EMPLOYEE2(
DESIGNATION             VARCHAR(20),
RATING                  INT,
SALARY_INCREMENT        INT,
CONSTRAINT PK_EMPLOYEE2
PRIMARY KEY(DESIGNATION,RATING),
CONSTRAINT CHECK_EMPLOYEE_RATING
CHECK(RATING IN(0,1,2,3,4,5)));

CREATE TABLE CUSTOMER(
CUSTOMER_ID             VARCHAR(10),
EMAIL                   VARCHAR(50)     NOT NULL,
FIRST_NAME              VARCHAR(20)     NOT NULL,
LAST_NAME               VARCHAR(20),
ADDRESS                 VARCHAR(50),
PHONE_NUMBER            INT,
GENDER                  VARCHAR(10),
CONSTRAINT PK_CUSTOMER
PRIMARY KEY(CUSTOMER_ID),
CONSTRAINT CHECK_CUSTOMER_GENDER
CHECK(GENDER IN('MALE','FEMALE','UNKNOWN')));

CREATE TABLE PLANT1(
PLANT_ID                VARCHAR(10),
PLANT_LOCATION          VARCHAR(20)      NOT NULL,
MANAGER_ID              VARCHAR(10),
CONSTRAINT PK_PLANT1
PRIMARY KEY(PLANT_ID));

CREATE TABLE PLANT2(
PLANT_LOCATION          VARCHAR(20),
PLANT_NAME              VARCHAR(20)     UNIQUE,
CONSTRAINT PK_PLANT2
PRIMARY KEY(PLANT_LOCATION));

CREATE TABLE SUPPLIER(
SUPPLIER_ID             VARCHAR(10),
SUPPLIER_NAME           VARCHAR(20)     NOT NULL,
ADDRESS                 VARCHAR(50),
PHONE_NUMBER            INT,
CONSTRAINT PK_SUPPLIER
PRIMARY KEY(SUPPLIER_ID));

CREATE TABLE VEHICLE(
VEHICLE_ID              VARCHAR(10),
MODEL_NAME              VARCHAR(10)     NOT NULL,
EXTERIOR_COLOR          VARCHAR(20)     NOT NULL,
INTERIOR_COLOR          VARCHAR(20)     NOT NULL,
WHEELS                  VARCHAR(20)     NOT NULL,
AUTOPILOT_KIT           VARCHAR(20)     NOT NULL,
PLANT_ID                VARCHAR(10),
CONSTRAINT PK_VEHICLE
PRIMARY KEY(VEHICLE_ID),
CONSTRAINT FK_VEHICLE_PLANT1
FOREIGN KEY(PLANT_ID) REFERENCES PLANT1(PLANT_ID));
```

```
CREATE TABLE PLANT_ORDER(
PLANT_ORDER_ID          VARCHAR(10),
MODEL_NAME              VARCHAR(10)     NOT NULL,
EXTERIOR_COLOR          VARCHAR(20)     NOT NULL,
INTERIOR_COLOR          VARCHAR(20)     NOT NULL,
WHEELS                  VARCHAR(20)     NOT NULL,
AUTOPILOT_KIT           VARCHAR(20)     NOT NULL,
STATUS                  VARCHAR(20),
PLANT_ID                VARCHAR(10),
CONSTRAINT PK_PLANT_ORDER
PRIMARY KEY(PLANT_ORDER_ID),
CONSTRAINT FK_PLANT_ORDER_PLANT1
FOREIGN KEY(PLANT_ID) REFERENCES PLANT1(PLANT_ID) ON DELETE CASCADE);

CREATE TABLE PLANT_PART_ORDER(
PLANT_PART_ORDER_ID     VARCHAR(10),
PART_NAME               VARCHAR(20)     NOT NULL,
PLANT_ID                VARCHAR(10),
CONSTRAINT PK_PLANT_PART_ORDER
PRIMARY KEY(PLANT_PART_ORDER_ID),
CONSTRAINT FK_PLANT_PART_ORDER_PLANT1
FOREIGN KEY(PLANT_ID) REFERENCES PLANT1(PLANT_ID) ON DELETE CASCADE);

CREATE TABLE PARTS(
PART_ID                 VARCHAR(10),
PART_NAME               VARCHAR(20)     NOT NULL,
PLANT_ID                VARCHAR(10),
SUPPLIER_ID             VARCHAR(10),
VEHICLE_ID              VARCHAR(10),
CONSTRAINT PK_PARTS
PRIMARY KEY(PART_ID),
CONSTRAINT FK_PARTS_PLANT1
FOREIGN KEY(PLANT_ID) REFERENCES PLANT1(PLANT_ID) ON DELETE CASCADE,
CONSTRAINT  FK_PARTS_SUPPLIER
FOREIGN KEY(SUPPLIER_ID) REFERENCES SUPPLIER(SUPPLIER_ID) ON DELETE CASCADE,
CONSTRAINT FK_PARTS_VEHICLE
FOREIGN KEY(VEHICLE_ID) REFERENCES VEHICLE(VEHICLE_ID) ON DELETE CASCADE);

CREATE TABLE PROCESSED_BY(
PLANT_PART_ORDER_ID     VARCHAR(10),
SUPPLIER_ID             VARCHAR(10),
STATUS                  VARCHAR(20),
CONSTRAINT PK_PROCESSED_BY
PRIMARY KEY(PLANT_PART_ORDER_ID, SUPPLIER_ID),
CONSTRAINT FK_PROCESSED_BY_PLANTPARTORDER
FOREIGN KEY(PLANT_PART_ORDER_ID) REFERENCES
PLANT_PART_ORDER(PLANT_PART_ORDER_ID) ON DELETE CASCADE,
CONSTRAINT FK_PROCESSED_BY_SUPPLIER
FOREIGN KEY(SUPPLIER_ID) REFERENCES SUPPLIER(SUPPLIER_ID) ON DELETE CASCADE);

CREATE TABLE PARTS_LOG(
PART_ID                 VARCHAR(10),
PART_NAME               VARCHAR(20),
PLANT_ID                VARCHAR(10),
SUPPLIER_ID             VARCHAR(10),
VEHICLE_ID              VARCHAR(10));
```

```
CREATE TABLE ORDERS(
ORDER_ID                VARCHAR(10),
MODEL_NAME              VARCHAR(10)     NOT NULL,
EXTERIOR_COLOR          VARCHAR(20)     NOT NULL,
INTERIOR_COLOR          VARCHAR(20)     NOT NULL,
WHEELS                  VARCHAR(20)     NOT NULL,
AUTOPILOT_KIT           VARCHAR(20)     NOT NULL,
PAYMENT_DETAILS         VARCHAR(50)     NOT NULL,
STATUS                  VARCHAR(20),
EMPLOYEE_ID             VARCHAR(10),
PLANT_ID                VARCHAR(10),
VEHICLE_ID              VARCHAR(10),
CUSTOMER_ID             VARCHAR(10),
CONSTRAINT PK_ORDER
PRIMARY KEY(ORDER_ID),
CONSTRAINT FK_ORDER_EMPLOYEE1
FOREIGN KEY(EMPLOYEE_ID) REFERENCES EMPLOYEE1(EMPLOYEE_ID) ON DELETE SET NULL,
CONSTRAINT FK_ORDER_PLANT1
FOREIGN KEY(PLANT_ID) REFERENCES PLANT1(PLANT_ID) ON DELETE SET NULL,
CONSTRAINT FK_ORDER_CUSTOMER
FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID) ON DELETE CASCADE);

CREATE TABLE VEHICLE_TYPE1(
MODEL_NAME              VARCHAR(10),
EXTERIOR_COLOR          VARCHAR(20),
INTERIOR_COLOR          VARCHAR(20),
WHEELS                  VARCHAR(20),
AUTOPILOT_KIT           VARCHAR(20),
CONSTRAINT K_VEHICLE_TYPE1
PRIMARY KEY(MODEL_NAME,EXTERIOR_COLOR,INTERIOR_COLOR,WHEELS,AUTOPILOT_KIT));

CREATE TABLE CONTAINS(
PLANT_ID                VARCHAR(10),
MODEL_NAME              VARCHAR(10) ,
EXTERIOR_COLOR          VARCHAR(20),
INTERIOR_COLOR          VARCHAR(20),
WHEELS                  VARCHAR(20),
AUTOPILOT_KIT           VARCHAR(20),
CONSTRAINT PK_CONTAINS
PRIMARY KEY(PLANT_ID, MODEL_NAME, EXTERIOR_COLOR, INTERIOR_COLOR, WHEELS,
AUTOPILOT_KIT),
CONSTRAINT FK_CONTAINS_PLANT1
FOREIGN KEY(PLANT_ID) REFERENCES PLANT1(PLANT_ID) ON DELETE CASCADE,
CONSTRAINT FK_CONTAINS_VEHICLE_TYPE1
FOREIGN KEY(MODEL_NAME,EXTERIOR_COLOR, INTERIOR_COLOR, WHEELS, AUTOPILOT_KIT)
REFERENCES VEHICLE_TYPE1(MODEL_NAME,EXTERIOR_COLOR,INTERIOR_COLOR,WHEELS,
AUTOPILOT_KIT) ON DELETE CASCADE);
```

INSERT :-

```
INSERT INTO SUPPLIER VALUES('S1','JOHN','DALLAS',4698765531);
INSERT INTO SUPPLIER VALUES('S2','SHERLOCK','RICHARDSON',4698765876);
INSERT INTO SUPPLIER VALUES('S3','HOLMES','PLANO',4698765432);
INSERT INTO SUPPLIER VALUES('S4','GOKUL','IRVING',4691455543);
INSERT INTO SUPPLIER VALUES('S5','PRAVEEN','FRISCO',4698769543);


INSERT INTO EMPLOYEE1(EMPLOYEE_ID,DESIGNATION,RATING,FIRST_NAME,LAST_NAME,ADDRESS,
PHONE_NUMBER,GENDER,SALARY)
VALUES('E1','SUPERVISOR',0,'PRAVEEN','TANGARAJAN','DALLAS',4698765432,'MALE',60000);
INSERT INTO EMPLOYEE1
VALUES('E2','MANAGER',0,'E1','SHANMUGA','PRIYAN','RICHARDSON',4698765886,'MALE',70000);
INSERT INTO EMPLOYEE1(EMPLOYEE_ID,DESIGNATION,RATING,FIRST_NAME,LAST_NAME,ADDRESS,
PHONE_NUMBER,GENDER,SALARY)
VALUES('E3','SUPERVISOR',0,'THIAGARAJAN','RAVICHANDRAN','PLANO',4698765878,'MALE',80000);
INSERT INTO EMPLOYEE1
VALUES('E4','MANAGER',0,'E2','NAVEEN','JAKUVA','FRISCO',4698764536,'MALE',80000);
INSERT INTO EMPLOYEE1
VALUES('E5','TECH SUPPORT',0,'E1','SARAVANA','PRABAKAR','DALLAS',4698844536,'MALE',90000);
INSERT INTO EMPLOYEE1
VALUES('E6','SUPPORT SPECIALIST',0,'E2','VIGNESH','RAVI','PLANO',4699064536,'MALE',80000);
INSERT INTO EMPLOYEE1
VALUES('E7','SUPPORT ENGINEER',0,'E1','ROHIT','ADITHYA','DALLAS',4698064536,'MALE',80000);

INSERT INTO EMPLOYEE2 VALUES('MANAGER',1,3000);
INSERT INTO EMPLOYEE2 VALUES('MANAGER',2,4000);
INSERT INTO EMPLOYEE2 VALUES('MANAGER',3,5000);
INSERT INTO EMPLOYEE2 VALUES('MANAGER',4,6000);
INSERT INTO EMPLOYEE2 VALUES('MANAGER',5,7000);

INSERT INTO EMPLOYEE2 VALUES('SUPERVISOR',1,1000);
INSERT INTO EMPLOYEE2 VALUES('SUPERVISOR',2,2000);
INSERT INTO EMPLOYEE2 VALUES('SUPERVISOR',3,3000);
INSERT INTO EMPLOYEE2 VALUES('SUPERVISOR',4,4000);
INSERT INTO EMPLOYEE2 VALUES('SUPERVISOR',5,5000);

INSERT INTO EMPLOYEE2 VALUES('SUPPORT ENGINEER',1,2000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT ENGINEER',2,3000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT ENGINEER',3,4000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT ENGINEER',4,5000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT ENGINEER',5,6000);

INSERT INTO EMPLOYEE2 VALUES('TECH SUPPORT',1,4000);
INSERT INTO EMPLOYEE2 VALUES('TECH SUPPORT',2,5000);
INSERT INTO EMPLOYEE2 VALUES('TECH SUPPORT',3,6000);
INSERT INTO EMPLOYEE2 VALUES('TECH SUPPORT',4,7000);
INSERT INTO EMPLOYEE2 VALUES('TECH SUPPORT',5,8000);

INSERT INTO EMPLOYEE2 VALUES('SUPPORT SPECIALIST',1,2000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT SPECIALIST',2,4000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT SPECIALIST',3,6000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT SPECIALIST',4,8000);
INSERT INTO EMPLOYEE2 VALUES('SUPPORT SPECIALIST',5,10000);
```

```
INSERT INTO CUSTOMER
VALUES('C1','SHANMUGA@GMAIL.COM','SHANMUGA','SUNDARAM','PLANO',4698765432,'MALE');
INSERT INTO CUSTOMER
VALUES('C2','PRIYA@GMAIL.COM','PRIYA','SANKAR','RICHARDSON',4698707432,'FEMALE');
INSERT INTO CUSTOMER
VALUES('C3','SANKAR@GMAIL.COM','SANKAR','RAVI','DALLAS',4698105432,'MALE');
INSERT INTO CUSTOMER
VALUES('C4','SANKARI@GMAIL.COM','SANKARI','SUNDARAM','FRISCO',4698701432,'FEMALE');
INSERT INTO CUSTOMER
VALUES('C5','SHRUTI@GMAIL.COM','SHRUTI','SHANMUGA','PLANO',4698791432,'FEMALE');
INSERT INTO CUSTOMER
VALUES('C6','PARKAVI@GMAIL.COM','PARKAVI','PRIYAN','PLANO',4698791156,'FEMALE');
INSERT INTO CUSTOMER
VALUES('C7','NITHYA@GMAIL.COM','NITHYA','KRISHNAN','RICHARDSON',4698791891,'FEMALE');
INSERT INTO CUSTOMER
VALUES('C8','RAM@GMAIL.COM','RAM','SHANMUGA','FRISCO',4698791516,'MALE');

INSERT INTO VEHICLE_TYPE1 VALUES('MODEL X','BLUE','VIOLET','STEEL WHEELS','AUTOPILOT_YES');
INSERT INTO VEHICLE_TYPE1
VALUES('MODEL 3','BLACK','CYAN','ALLOYS WHEELS','AUTOPILOT_NO');
INSERT INTO VEHICLE_TYPE1 VALUES('MODEL S','WHITE','GREY','MULTI PIECE','AUTOPILOT_YES');
INSERT INTO VEHICLE_TYPE1 VALUES('MODEL X','BLUE','GREY','FORGED WHEELS','AUTOPILOT_NO');
INSERT INTO VEHICLE_TYPE1
VALUES('MODEL S','BLACK','VIOLET','STEEL WHEELS','AUTOPILOT_NO');

INSERT INTO VEHICLE_TYPE2 VALUES('MODEL X','INLINE ENGINE','ELECTROMAGNETIC');
INSERT INTO VEHICLE_TYPE2 VALUES('MODEL 3','ROTARY ENGINE','SERVO BRAKE');
INSERT INTO VEHICLE_TYPE2 VALUES('MODEL S','VR AND W ENGINE','MECHANICAL BRAKE');

INSERT INTO VEHICLE_TYPE3 VALUES('INLINE ENGINE','HELICAL GEAR');
INSERT INTO VEHICLE_TYPE3 VALUES('ROTARY ENGINE','MITER GEAR');
INSERT INTO VEHICLE_TYPE3 VALUES('VR AND W ENGINE','BEVEL GEAR');

INSERT INTO PLANT1 VALUES('P1','FRISCO','E2');
INSERT INTO PLANT1 VALUES('P2','PLANO','E1');
INSERT INTO PLANT1 VALUES('P3','RICHARDSON','E2');
INSERT INTO PLANT1 VALUES('P4','DALLAS','E1');
INSERT INTO PLANT1 VALUES('P5','FORT WORTH','E2');

INSERT INTO PLANT2 VALUES('DALLAS','DALLAS PLANT');
INSERT INTO PLANT2 VALUES('PLANO','PLANO PLANT');
INSERT INTO PLANT2 VALUES('FRISCO','FRISCO PLANT');
INSERT INTO PLANT2 VALUES('RICHARDSON','RICHARDSON PLANT');
INSERT INTO PLANT2 VALUES('FORT WORTH','FORT WORTH PLANT');

INSERT INTO VEHICLE
VALUES('V1','MODEL X','WHITE','VIOLET','ALLOY WHEELS','AUTOPILOT_YES','P1');
INSERT INTO VEHICLE
VALUES('V2','MODEL 3','BLACK','CYAN','FORGED WHEELS','AUTOPILOT_NO','P2');
INSERT INTO VEHICLE VALUES('V3','MODEL S','BLUE','GREY','MULTI PIECE','AUTOPILOT_YES','P1');
INSERT INTO VEHICLE VALUES('V4','MODEL 3','WHITE','VIOLET','STEEL WHEELS','AUTOPILOT_NO','P4');
INSERT INTO VEHICLE VALUES('V5','MODEL S','BLUE','CYAN','ALLOY WHEELS','AUTOPILOT_YES','P5');
INSERT INTO VEHICLE VALUES('V8','MODEL S','BLUE','GREY','MULTI PIECE','AUTOPILOT_YES','P3');
INSERT INTO VEHICLE VALUES('V9','MODEL 3','WHITE','VIOLET','STEEL WHEELS','AUTOPILOT_NO','P4');
```

```
INSERT INTO PLANT_ORDER
VALUES('PO1','MODEL X','WHITE','VIOLET','ALLOY WHEELS','AUTOPILOT_YES','IN-PROGRESS','P1');
INSERT INTO PLANT_ORDER
VALUES('PO2','MODEL 3','BLACK','CYAN','FORGED WHEELS','AUTOPILOT_NO','IN-PROGRESS','P2');
INSERT INTO PLANT_ORDER
VALUES('PO3','MODEL S','BLUE','GREY','MULTI PIECE','AUTOPILOT_YES','DELIVERED','P3');
INSERT INTO PLANT_ORDER
VALUES('PO4','MODEL 3','WHITE','VIOLET','STEEL WHEELS','AUTOPILOT_NO','DELIVERED','P4');
INSERT INTO PLANT_ORDER
VALUES('PO5','MODEL X','WHITE','CYAN','FORGED WHEELS','AUTOPILOT_YES','','P5');
INSERT INTO PLANT_ORDER
VALUES('PO6','MODEL S','BLUE','GREY','STEEL WHEELS','AUTOPILOT_NO','IN-PROGRESS','P2');
INSERT INTO PLANT_ORDER
VALUES('PO7','MODEL X','BLUE','GREY','OEM-STYLE','AUTOPILOT_NO','IN-PROGRESS','P4');

INSERT INTO ORDERS
VALUES('O1','MODEL 3','WHITE','VIOLET','STEEL WHEELS','AUTOPILOT_NO','PAID','IN-PROGRESS',
'E6','P4','','C2');
INSERT INTO ORDERS
VALUES('O2','MODEL S','BLUE','CYAN','ALLOY WHEELS','AUTOPILOT_YES','HALF PAID','IN-PROGRESS',
'E6','P5','','C5');
INSERT INTO ORDERS
VALUES('O3','MODEL X','BLACK','GREY','STEEL WHEELS','AUTOPILOT_YES','PAID','DELIVERED',
'E7','P1','V7','C3');
INSERT INTO ORDERS
VALUES('O4','MODEL 3','WHITE','CYAN','FORGED WHEELS','AUTOPILOT_NO','PAID','DELIVERED',
'E6','P2','V6','C4');
INSERT INTO ORDERS
VALUES('O5','MODEL 3','WHITE','GREY','MULTI PIECE','AUTOPILOT_NO','HALF PAID','IN PROGRESS',
'E6','P4','','C8');
INSERT INTO ORDERS
VALUES('O6','MODEL X','BLUE','CYAN','MULTI PIECE','AUTOPILOT_NO','PAID','IN PROGRESS',
'E7','P3','','C6');
INSERT INTO ORDERS
VALUES('O7','MODEL S','BLACK','GREY','FORGED WHEELS','AUTOPILOT_NO','HALF PAID','','E6','P2','','C7');

INSERT INTO PLANT_PART_ORDER VALUES('PPO1','GREY','P4');
INSERT INTO PLANT_PART_ORDER VALUES('PPO2','BLACK','P2');
INSERT INTO PLANT_PART_ORDER VALUES('PPO3','FORGED WHEELS','P4');
INSERT INTO PLANT_PART_ORDER VALUES('PPO4','STEEL WHEELS','P4');
INSERT INTO PLANT_PART_ORDER VALUES('PPO5','INLINE ENGINE','P2');
INSERT INTO PLANT_PART_ORDER VALUES('PPO6','SERVO BRAKE','P1');
INSERT INTO PLANT_PART_ORDER VALUES('PPO7','ROTARY ENGINE','P2');

INSERT INTO PARTS VALUES('PART1','WHITE','P4','S1','');
INSERT INTO PARTS VALUES('PART2','VIOLET','P4','S2','');
INSERT INTO PARTS VALUES('PART3','STEEL WHEELS','P4','S5','');
INSERT INTO PARTS VALUES('PART4','AUTOPILOT_NO','P4','S3','');
INSERT INTO PARTS VALUES('PART5','BLUE','P5','S4','');
INSERT INTO PARTS VALUES('PART6','CYAN','P5','S1','');
INSERT INTO PARTS VALUES('PART7','ALLOY WHEELS','P5','S2','');
INSERT INTO PARTS VALUES('PART8','AUTOPILOT_YES','P5','S5','');
INSERT INTO PARTS VALUES('PART9','VR AND W ENGINE','P1','S4','');
INSERT INTO PARTS VALUES('PART10','MECHANICAL BRAKE','P2','S1','');
INSERT INTO PARTS VALUES('PART11','BEVEL GEAR','P3','S3','');
```

```
INSERT INTO PARTS_LOG VALUES('PART1','BLACK','P1','S2','V7');
INSERT INTO PARTS_LOG VALUES('PART2','GREY','P1','S4','V7');
INSERT INTO PARTS_LOG VALUES('PART3','STEEL WHEELS','P1','S5','V7');
INSERT INTO PARTS_LOG VALUES('PART4','AUTOPILOT_YES','P1','S1','V7');
INSERT INTO PARTS_LOG VALUES('PART5','WHITE','P2','S1','V6');
INSERT INTO PARTS_LOG VALUES('PART6','CYAN','P2','S3','V6');
INSERT INTO PARTS_LOG VALUES('PART7','FORGED WHEELS','P2','S5','V6');
INSERT INTO PARTS_LOG VALUES('PART8','AUTOPILOT_NO','P2','S1','V6');
INSERT INTO PARTS_LOG VALUES('PART9','BLUE','P3','S4','V8');
INSERT INTO PARTS_LOG VALUES('PART10','GREY','P3','S3','V8');
INSERT INTO PARTS_LOG VALUES('PART11','MULTI PIECE','P3','S1','V8');
INSERT INTO PARTS_LOG VALUES('PART12','AUTOPILOT_YES','P3','S2','V8');
INSERT INTO PARTS_LOG VALUES('PART13','WHITE','P4','S3','V9');
INSERT INTO PARTS_LOG VALUES('PART14','VIOLET','P4','S5','V9');
INSERT INTO PARTS_LOG VALUES('PART15','STEEL WHEELS','P4','S1','V9');
INSERT INTO PARTS_LOG VALUES('PART16','AUTOPILOT_NO','P4','S2','V9');

INSERT INTO CONTAINS VALUES('P1','MODEL X','BLUE','VIOLET','STEEL WHEELS','AUTOPILOT_YES');
INSERT INTO CONTAINS VALUES('P2','MODEL 3','BLACK','CYAN','ALLOYS WHEELS','AUTOPILOT_NO');
INSERT INTO CONTAINS VALUES('P5','MODEL S','WHITE','GREY','MULTI PIECE','AUTOPILOT_YES');
INSERT INTO CONTAINS VALUES('P2','MODEL X','BLUE','GREY','FORGED WHEELS','AUTOPILOT_NO');
INSERT INTO CONTAINS VALUES('P3','MODEL S','BLACK','VIOLET','STEEL WHEELS','AUTOPILOT_NO');

INSERT INTO PROCESSED_BY VALUES('PPO1','S2','DELIVERED');
INSERT INTO PROCESSED_BY VALUES('PPO2','S4','IN-PROGRESS');
INSERT INTO PROCESSED_BY VALUES('PPO3','S3','IN-PROGRESS');
INSERT INTO PROCESSED_BY VALUES('PPO4','S2','');
INSERT INTO PROCESSED_BY VALUES('PPO5','S1','');
```

SELECT TABLE :-

```
SELECT * FROM VEHICLE_TYPE1;
SELECT * FROM VEHICLE_TYPE2;
SELECT * FROM VEHICLE_TYPE3;
SELECT * FROM EMPLOYEE1;
SELECT * FROM EMPLOYEE2;
SELECT * FROM PLANT1;
SELECT * FROM PLANT2;
SELECT * FROM PLANT_ORDER;
SELECT * FROM PARTS;
SELECT * FROM PROCESSED_BY;
SELECT * FROM CONTAINS;
SELECT * FROM PARTS_LOG;
SELECT * FROM SUPPLIER;
SELECT * FROM VEHICLE;
SELECT * FROM PLANT_PART_ORDER;
SELECT * FROM ORDERS;
SELECT * FROM CUSTOMER;
```

7. PL/SQL

TRIGGERS

1. Trigger on inserting Order

   Checks the given vehicle specifications in VEHICLE table, if there is a match found then assign VID to the order tuple that is to be inserted and delete the VID tuple in the VEHICLE table (updating the warehouse) and change the status to delivered else change the status to In-progress.

CHECK_EXISTING_WAREHOUSE Trigger Code:

```
CREATE OR REPLACE TRIGGER CHECK_EXISTING_WAREHOUSE
BEFORE INSERT ON ORDERS
FOR EACH ROW
DECLARE
C INT;
BEGIN
SELECT COUNT(*) INTO C FROM VEHICLE V
WHERE V.MODEL_NAME=:NEW.MODEL_NAME AND V.EXTERIOR_COLOR=:NEW.EXTERIOR_COLOR
AND V.INTERIOR_COLOR=:NEW.INTERIOR_COLOR AND V.WHEELS=:NEW.WHEELS
AND V.AUTOPILOT_KIT=:NEW.AUTOPILOT_KIT;
IF C<=0 THEN
IF :NEW.STATUS IS NULL THEN
:NEW.STATUS :='IN-PROGRESS';
END IF;
ELSE
SELECT V.VEHICLE_ID INTO :NEW.VEHICLE_ID FROM VEHICLE V
WHERE V.MODEL_NAME=:NEW.MODEL_NAME AND V.EXTERIOR_COLOR=:NEW.EXTERIOR_COLOR
AND V.INTERIOR_COLOR=:NEW.INTERIOR_COLOR AND V.WHEELS=:NEW.WHEELS
AND V.AUTOPILOT_KIT=:NEW.AUTOPILOT_KIT AND ROWNUM = 1;
IF :NEW.STATUS IS NULL THEN
:NEW.STATUS :='DELIVERED';
END IF;
DELETE FROM VEHICLE WHERE VEHICLE_ID=:NEW.VEHICLE_ID;
END IF;
END;
```

```
INSERT INTO PLANT1 VALUES('P1','FREMONT',1);
INSERT INTO VEHICLE VALUES('V1','MODELX','BLUE','BLACK','AERO','YES','P1');
INSERT INTO VEHICLE VALUES('V2','MODELS','BLUE','BLACK','AERO','YES','P1');
INSERT INTO CUSTOMER
VALUES(1001,'ABC@GMAIL.COM','ABC','XYZ','FREMONT',1212,'FEMALE');
INSERT INTO ORDERS
VALUES('O1','MODELX','BLUE','BLACK','AERO','YES','COMPLETED',NULL,NULL,'P1',NULL,1001);
SELECT * FROM ORDERS
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.018 seconds

	ORDER_ID	MODEL_NAME	EXTERIOR_COLOR	INTERIOR_COLOR	WHEELS	AUTOPILOT_KIT	PAYMENT_DETAILS	STATUS	EMPLOYEE_ID	PLANT_ID	VEHICLE_ID	CUSTOMER_
1	O1	MODELX	BLUE	BLACK	AERO	YES	COMPLETED	DELIVERED	(null)	P1	V1	1001

2. Trigger on updating Employee Rating

   Increments the salary of the Employee if there is an update on the rating attribute.

RATING_INCREMENT Trigger Code:

```
CREATE OR REPLACE TRIGGER RATING_INCREMENT
BEFORE UPDATE ON EMPLOYEE1
FOR EACH ROW
DECLARE
SALARY_INCREMENT_VALUE EMPLOYEE2.SALARY_INCREMENT%TYPE;
BEGIN
SELECT SALARY_INCREMENT INTO SALARY_INCREMENT_VALUE
FROM EMPLOYEE2
WHERE EMPLOYEE2.RATING=:NEW.RATING AND EMPLOYEE2.DESIGNATION=:NEW.DESIGNATION;
:NEW.SALARY := :NEW.SALARY+SALARY_INCREMENT_VALUE;
END;

INSERT INTO EMPLOYEE2 VALUES('SOFTWARE DEVELOPER',2,2000);
INSERT INTO EMPLOYEE1 VALUES('2','SOFTWARE DEVELOPER',NULL,NULL,'SRUJAN','KRISHNA','CHICAGO',123456789,'MALE',10000);
UPDATE EMPLOYEE1 SET RATING=2 WHERE EMPLOYEE_ID=2;
SELECT * FROM EMPLOYEE1;
```

ript Output × Query Result ×

SQL | All Rows Fetched: 1 in 0.038 seconds

	EMPLOYEE_ID	DESIGNATION	RATING	SUPERVISOR_ID	FIRST_NAME	LAST_NAME	ADDRESS	PHONE_NUMBER	GENDER	SALARY
1	2	SOFTWARE DEVELOPER	2	(null)	SRUJAN	KRISHNA	CHICAGO	123456789	MALE	12000

3. Trigger on deleting Parts

   Whenever a part is deleted we update the part log table for future reference. This can ideally be performed by the application layer but we have also attempted to capture it using NOSQL which we have discussed in the later part of the report.

UPDATE_PART_LOG Trigger Code:

```
CREATE OR REPLACE TRIGGER UPDATE_PART_LOG
BEFORE DELETE ON PARTS
FOR EACH ROW
DECLARE
BEGIN
INSERT INTO PARTS_LOG VALUES(:OLD.PART_ID,:OLD.PART_NAME,:OLD.PLANT_ID,:OLD.SUPPLIER_ID,:OLD.VEHICLE_ID);
END;
INSERT INTO PARTS VALUES('PART1','BLUE','1','5',NULL);
INSERT INTO PARTS VALUES('PART2','GREEN','1','5',NULL);
DELETE FROM PARTS WHERE PART_ID='PART1';
SELECT * FROM PARTS;
SELECT * FROM PARTS_LOG;
```

	PART_ID	PART_NAME	PLANT_ID	SUPPLIER_ID	VEHICLE_ID
1	PART2	GREEN	1	5	(null)

	PART_ID	PART_NAME	PLANT_ID	SUPPLIER_ID	VEHICLE_ID
1	PART1	BLUE	1	5	(null)

4. Trigger on inserting Plant Order

   Check the given vehicle specifications by the PLANT_ORDER in VEHICLE_TYPE table, if matched assign the PLANT_ID to the order to be inserted else we display “No model is available with given specifications”

ASSIGNING_PLANT_ID Trigger Code:

```
CREATE OR REPLACE TRIGGER ASSIGNING_PLANT_ID
BEFORE INSERT ON PLANT_ORDER
FOR EACH ROW
DECLARE
COUNT_MODEL INT;
BEGIN

SELECT COUNT(*) INTO COUNT_MODEL FROM CONTAINS C
WHERE C.MODEL_NAME=:NEW.MODEL_NAME AND C.AUTOPILOT_KIT=:NEW.AUTOPILOT_KIT AND C.EXTERIOR_COLOR=:NEW.EXTERIOR_COLOR
AND C.INTERIOR_COLOR=:NEW.INTERIOR_COLOR AND C.WHEELS=:NEW.WHEELS;

IF  COUNT_MODEL < 1
THEN RAISE_APPLICATION_ERROR(-20000, 'NO MODEL IS AVAILABLE WITH GIVEN SPECIFICATIONS');
END IF;

SELECT  PLANT_ID INTO :NEW.PLANT_ID FROM CONTAINS C
WHERE C.MODEL_NAME=:NEW.MODEL_NAME AND C.AUTOPILOT_KIT=:NEW.AUTOPILOT_KIT AND C.EXTERIOR_COLOR=:NEW.EXTERIOR_COLOR
AND C.INTERIOR_COLOR=:NEW.INTERIOR_COLOR AND C.WHEELS=:NEW.WHEELS AND ROWNUM = 1;
END;

INSERT INTO PLANT1 VALUES ('1','DALLAS','');
INSERT INTO PLANT2 VALUES ('DALLAS','DALLAS PLANT');
INSERT INTO VEHICLE_TYPE1 VALUES('TESLA_SX','BLUE','BLACK','ALLOY','YES');
INSERT INTO CONTAINS VALUES('1','TESLA_SX','BLUE','BLACK','ALLOY','YES');
INSERT INTO PLANT_ORDER VALUES('100','TESLA_SX','BLUE','BLACK','ALLOY','YES','PENDING',NULL);
SELECT * FROM PLANT_ORDER;
INSERT INTO PLANT_ORDER VALUES('101','TESLA_MX','YELLOW','BLACK','STEEL','YES','PENDING',NULL);
```

Query Result

SQL | All Rows Fetched: 1 in 0.017 seconds

	PLANT_ORDER_ID	MODEL_NAME	EXTERIOR_COLOR	INTERIOR_COLOR	WHEELS	AUTOPILOT_KIT	STATUS	PLANT_ID
1	100	TESLA_SX	BLUE	BLACK	ALLOY	YES	PENDING	1

Query Result | Script Output

Task completed in 0.099 seconds

```
Error starting at line : 27 in command -
INSERT INTO PLANT_ORDER VALUES('101','TESLA_MX','YELLOW','BLACK','STEEL','YES','PENDING',NULL)
Error report -
ORA-20000: NO MODEL IS AVAILABLE WITH GIVEN SPECIFICATIONS
ORA-06512: at "SXK180149.ASSIGNING_PLANT_ID", line 10
ORA-04088: error during execution of trigger 'SXK180149.ASSIGNING_PLANT_ID'
```

PROCEDURES

1. Accident Report: Given order Id and part name we can fetch supplier Id who supplied that part this might be useful during insurance claims or accidents.

Stored Procedure ACCIDENT_REPORT

```
CREATE OR REPLACE
PROCEDURE ACCIDENT_REPORT(ORDER_ID IN ORDERS.ORDER_ID%TYPE, PART_NAME IN PARTS_LOG.PART_NAME%TYPE) IS
V_ID ORDERS.VEHICLE_ID%TYPE;
S_ID PARTS.SUPPLIER_ID%TYPE;
SUPPLIER_NAME VARCHAR(20);
BEGIN
SELECT ORDERS.VEHICLE_ID INTO V_ID FROM ORDERS WHERE ORDERS.ORDER_ID=ORDER_ID;
SELECT PARTS_LOG.SUPPLIER_ID INTO S_ID FROM PARTS_LOG WHERE PARTS_LOG.VEHICLE_ID=V_ID
AND PARTS_LOG.PART_NAME=PART_NAME;
SELECT SUPPLIER.SUPPLIER_NAME INTO SUPPLIER_NAME FROM SUPPLIER WHERE SUPPLIER.SUPPLIER_ID=S_ID;
DBMS_OUTPUT.PUT_LINE(SUPPLIER_NAME);
END;

INSERT INTO PLANT1 VALUES('1','DALLAS',NULL);
INSERT INTO VEHICLE VALUES('1','TESLA_MX','BLUE','BLACK','ARROW_WHEELS','YES',1);
INSERT INTO CUSTOMER VALUES('3','VIVEK@GMAIL.COM','VIVEK','SIMHA','PLANO','123456789','MALE');
INSERT INTO SUPPLIER VALUES('5','RAMESH','DALLAS',123457894);
INSERT INTO ORDERS VALUES('1','TESLA_MX','BLUE','BLACK','ARROW_WHEEL','YES','PAID','DELIVERED',NULL,1,1,3);
INSERT INTO PARTS_LOG VALUES('1','ARROW_WHEEL',1,5,1);
SELECT * FROM PARTS_LOG;
SET SERVEROUTPUT ON
DECLARE
ORDER_ID NUMBER;
PART_NAME PARTS.PART_NAME%TYPE;
SUPPLIER_NAME SUPPLIER.SUPPLIER_NAME%TYPE;
BEGIN
ORDER_ID:=1;
PART_NAME:='ARROW_WHEEL';
  ACCIDENT_REPORT(ORDER_ID,PART_NAME);
END;
```

2. Inventory details: Given Plant Id, it will fetch the entire inventory(current) of that plant.

Stored Procedure GET_INVENTORY_DETAILS

```
CREATE OR REPLACE PROCEDURE GET_INVENTORY_DETAILS(PLANTID IN PLANT1.PLANT_ID%TYPE) AS
THISPART PARTS%ROWTYPE;
PNAME VARCHAR(20);
TOTAL INT;
CURSOR PARTCOUNT IS
SELECT PART_NAME, COUNT(*) C FROM PARTS P WHERE PLANT_ID=PLANTID GROUP BY PART_NAME;
BEGIN
DBMS_OUTPUT.PUT_LINE('PART NAME'|| ' | ' || 'QUANTITY');
FOR I IN PARTCOUNT
LOOP
DBMS_OUTPUT.PUT_LINE(I.PART_NAME || '         ' ||I.C );
END LOOP;
END;

INSERT INTO PLANT1 VALUES('P1','FREMONT',1);
INSERT INTO PLANT1 VALUES('P2','SAN FRANCISCO',2);
INSERT INTO SUPPLIER VALUES('S1','ABC','FREMONT',123456);
INSERT INTO PARTS VALUES('PART1','BLUE','P1','S1',NULL);
INSERT INTO PARTS VALUES('PART2','BLUE','P1','S1',NULL);
INSERT INTO PARTS VALUES('PART3','AERO','P1','S1',NULL);
INSERT INTO PARTS VALUES('PART4','BLUE','P2','S1',NULL);
INSERT INTO PARTS VALUES('PART5','BLACK','P1','S1',NULL);

SET SERVEROUTPUT ON;
BEGIN
GET_INVENTORY_DETAILS('P1');
END;
```

3. Plant orders: This store procedure allows fulfilling the plant part orders by a supplier.

Stored Procedure SUPPLYING_ORDER

```
CREATE OR REPLACE PROCEDURE SUPPLYING_ORDER(PPID IN  PLANT_PART_ORDER.PLANT_PART_ORDER_ID%TYPE ,
S_ID IN  SUPPLIER.SUPPLIER_ID%TYPE) IS
TEMP_PART_NAME PARTS.PART_NAME%TYPE;
TEMP_PLANT_ID PARTS.PLANT_ID%TYPE;
BEGIN
SELECT P.PART_NAME , P.PLANT_ID INTO TEMP_PART_NAME ,TEMP_PLANT_ID FROM PLANT_PART_ORDER P
WHERE P.PLANT_PART_ORDER_ID=PPID;
INSERT INTO PARTS VALUES('1'  , TEMP_PART_NAME , TEMP_PLANT_ID,S_ID,'');
DELETE FROM PLANT_PART_ORDER PPO WHERE PPO.PLANT_PART_ORDER_ID=PPID;
END;

INSERT INTO PLANT1 VALUES('P1','DALLAS',NULL);
INSERT INTO SUPPLIER VALUES('10','RAMESH','PLANO',123456789);
INSERT INTO PLANT_PART_ORDER VALUES('1','ALLOY WHEELS','P1');
INSERT INTO PLANT_PART_ORDER VALUES('2','STEEL WHEELS','P1');
EXECUTE SUPPLYING_ORDER('1','10');
SELECT * FROM PLANT_PART_ORDER;
SELECT * FROM PARTS;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 1 in 0.015 seconds

	PLANT_PART_ORDER_ID	PART_NAME	PLANT_ID
1	2	STEEL WHEELS	P1

Script Output x | Query Result x

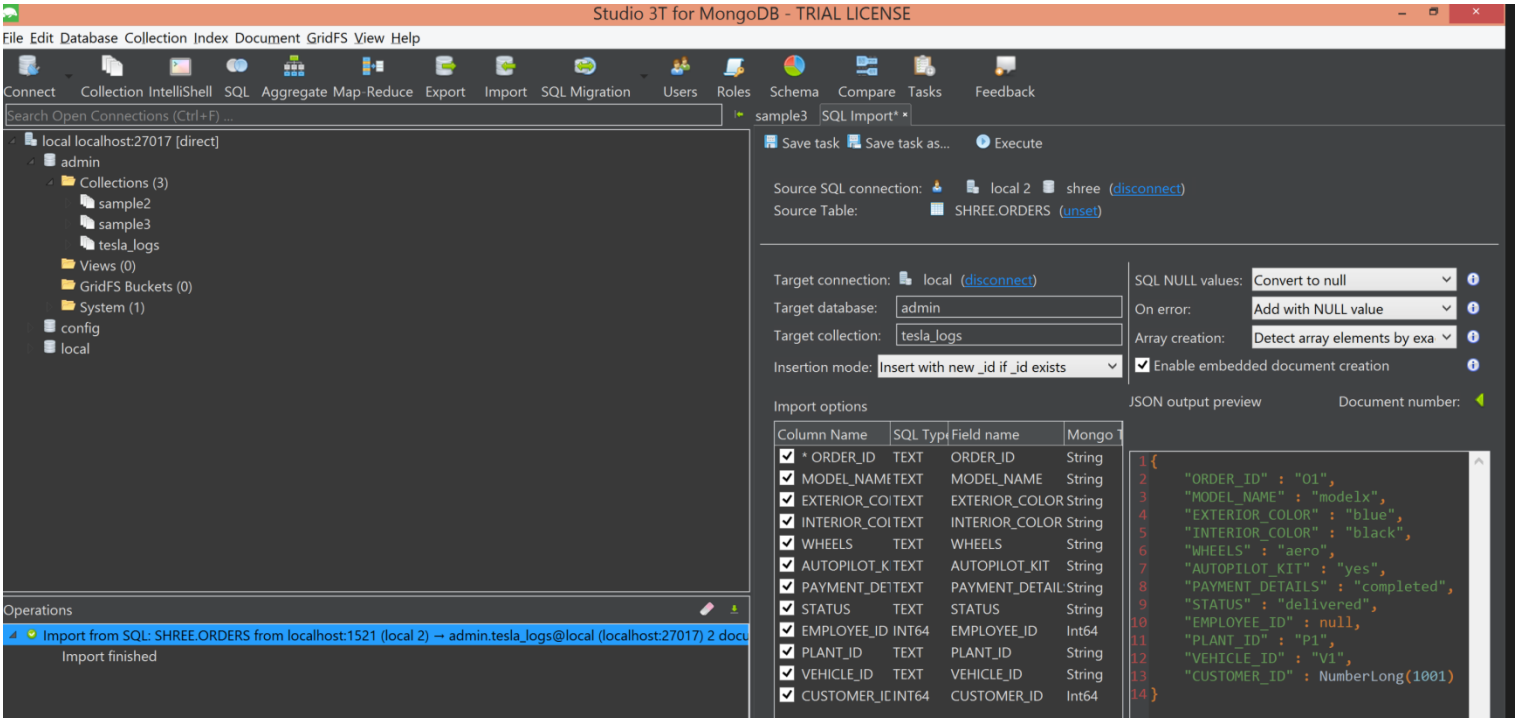
SQL | All Rows Fetched: 1 in 0.024 seconds

	PART_ID	PART_NAME	PLANT_ID	SUPPLIER_ID	VEHICLE_ID
1	1	ALLOY WHEELS	P1	10	(null)

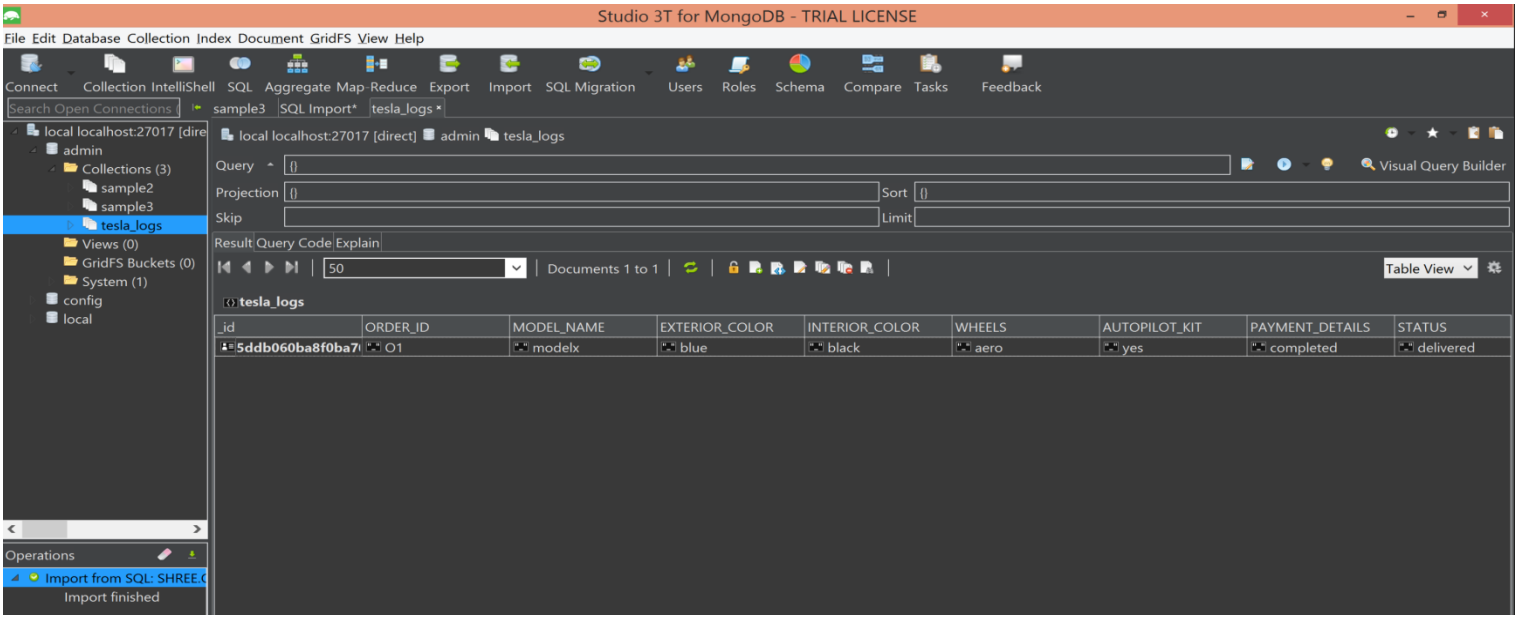
8. ASSOCIATING WITH NOSQL:

NOSQL (Not Only SQL) databases are mostly preferred in places where the data doesn't have a particular schema. We have chosen MongoDB since it is a document type NOSQL database and will be well suited for storing logs. So for the Tesla system we have attempted to capture the used parts and completed order information as a single log collection using MongoDB Studio 3T.

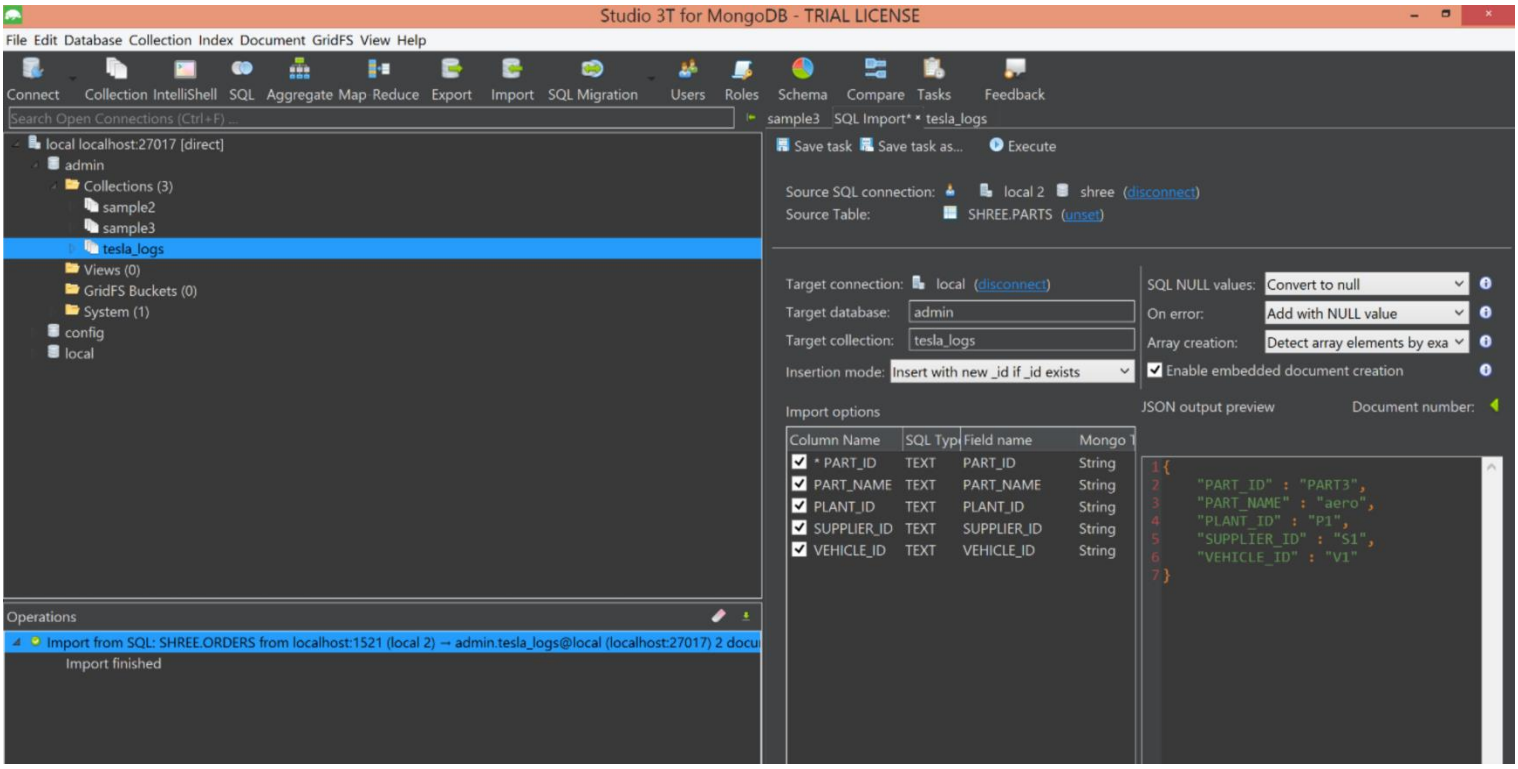
1. ORDERS table from sql developer is connected to the TESLA_LOGS collection in MongoDB.



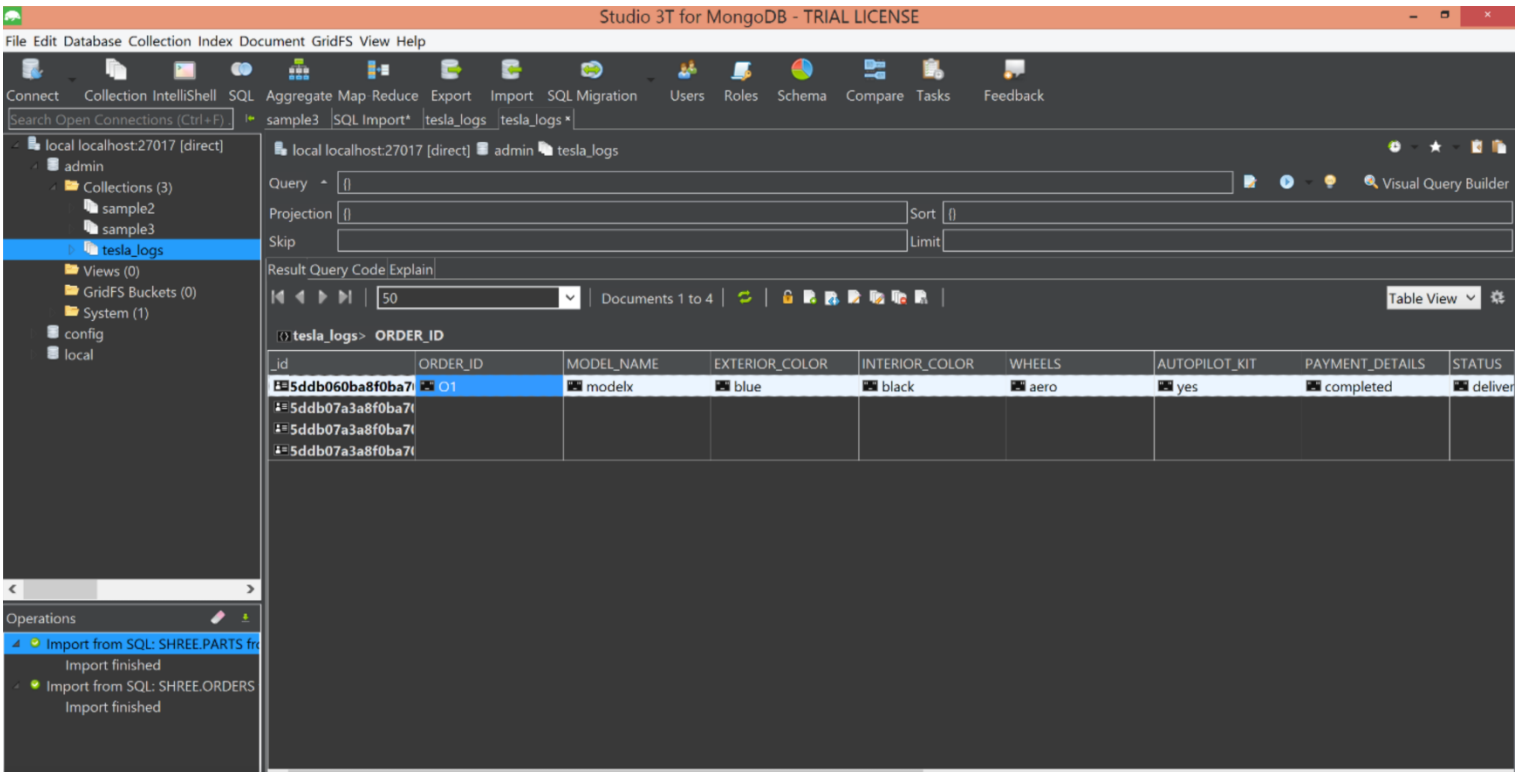
2. Results of TESLA_LOGS collection which has the delivered orders:

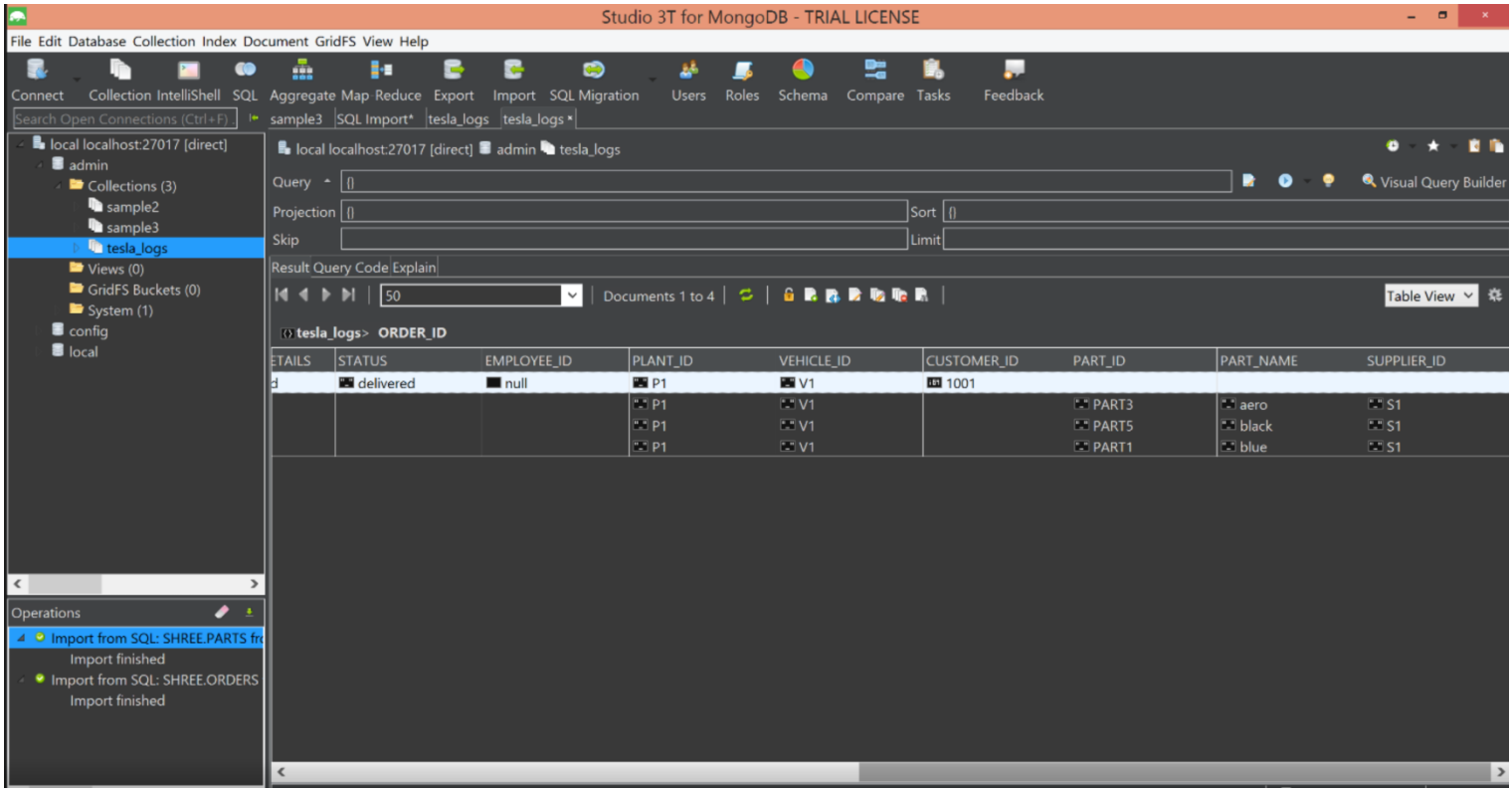


3. Connecting PARTS table from SQL developer to the same TESLA_LOGS collection in MongoDB:



4. Results of the final TESLA_LOGS collection in MongoDB where it contains the logs of all the used parts and delivered vehicle details.




Studio 3T for MongoDB - TRIAL LICENSE
File Edit Database Collection Index Document GridFS View Help
Connect Collection IntelliShell SQL Aggregate Map Reduce Export Import SQL Migration Users Roles Schema Compare Tasks Feedback
Search Open Connections (Ctrl+F)
sample3 SQL Import* tesla_logs tesla_logs *
local localhost:27017 [direct]
admin
Collections (3)
sample2
sample3
tesla_logs
Views (0)
GridFS Buckets (0)
System (1)
config
local
local localhost:27017 [direct] admin tesla_logs
Query {}
Projection {}
Sort {}
Skip
Limit
Visual Query Builder
Result Query Code Explain
50
Documents 1 to 4
Table View
tesla_logs> ORDER_ID
ETAILS STATUS EMPLOYEE_ID PLANT_ID VEHICLE_ID CUSTOMER_ID PART_ID PART_NAME SUPPLIER_ID
d delivered null P1 V1 1001
P1 V1 PART3 aero S1
P1 V1 PART5 black S1
P1 V1 PART1 blue S1
Operations
Import from SQL: SHREE.PARTS fr
Import finished
Import from SQL: SHREE.ORDERS
Import finished