Create Demo Test Train

```
INSERT INTO Trains VALUES (4, 1234, 1234, DEFAULT);
INSERT INTO TrainLocations VALUES (4, 'Black River Yard', DEFAULT);

INSERT INTO RollingStockCars VALUES ('XA', 'Reefer', NULL);
INSERT INTO RollingStockAtYards VALUES ('XA', 'Black River Yard', DEFAULT);
INSERT INTO Shipments VALUES (DEFAULT, 'Dairy', 'Half Circle Farms', 1, 'MMI Transfer Site 3', 1, DEFAULT);
INSERT INTO Waybills VALUES ('XA', LAST_INSERT_ID(), 'Black River Yard');
INSERT INTO ConsistedCars VALUES (4, 'XA', DEFAULT);
DELETE FROM RollingStockAtYards WHERE CarID = 'XA';

INSERT INTO RollingStockCars VALUES ('XB', 'Tank Car', NULL);
INSERT INTO RollingStockAtYards VALUES ('XB', 'Black River Yard', DEFAULT);
INSERT INTO Shipments VALUES (DEFAULT, 'Gasses', 'LGP Professionals', 4, 'Palin Interchange', 1, DEFAULT);
INSERT INTO Waybills VALUES ('XB', LAST_INSERT_ID(), 'Black River Yard');
INSERT INTO ConsistedCars VALUES (4, 'XB', DEFAULT);
DELETE FROM RollingStockAtYards WHERE CarID = 'XB';

INSERT INTO Crews VALUES ('Demo Player', NULL);
```

Creates a #4 train, two cars in a yard, and a new player. Shipping orders are created for the two cars. Waybills associate the shipping order to a car. The cars are removed from a yard and attached (consisted) to a train.

Add Crew To Train

```
SET @player = 'Demo Player';
SET @playerTrain = 4;
INSERT INTO TrainCrews VALUES (@playerTrain, @player, DEFAULT);
```

The user selects an existing train to crew. The system assigns that train to that crew and records the time.

Display List of Modules

```
SELECT *
 FROM Modules
   WHERE ModuleName
 IN (SELECT ModuleName
  FROM ModulesAvailable
       WHERE IsAvailable = TRUE);
```

Output:

ModuleName	ModuleOwner	ModuleType	ModuleShape	Description
180 Farms	Al Lowe	oNeTrak	180 Corner	NULL
Bauxen Crate	Al Lowe	Transition	Straight	Access to all lines available
Black River Yard	Mike Donnelly	oNeTrak	3-Straight	Contains the Black River Yard
Chesterfield	Al Lowe	Ntrak	2-Straight	No crossovers
Crossover	Al Lowe	Ntrak	Straight	Access to all main lines available
Grain Elevator	Al Lowe	oNeTrak	Straight	NULL
Palin Bridge	Al Lowe	oNeTrak	Straight	NULL
Pure Oil	Al Lowe	Transition	Straight	Access to all main lines available.
Scott Corner	Al Lowe	Ntrak	Corner	NULL
Trainyard Mall	Al Lowe	Ntrak	Corner	NULL
NULL	NULL	NULL	NULL	NULL

The user requests to see modules in the layout. The system displays modules that have been added and are marked as active.

Display User Train

```
SET @playerTrain = 4;
SELECT t.TrainNumber, t.LeadPower, t.DCCAddress, c.WithCrew, l.onModule, l.TimeUpdated
 FROM Trains t, TrainCrews c, TrainLocations l
 WHERE t.TrainNumber = c.OnTrain
  AND t.TrainNumber = l.TrainNumber
      AND t.TrainNumber = @playerTrain;
```

Output:

TrainNumber	LeadPower	DCCAddress	WithCrew	onModule	TimeUpdated
4	1234	1234	Demo Player	Black River Yard	2015-12-05 15:45:56

The user selects a train to view. The system displays the train number, locomotive number, digital command controller (DCC) address, which crew is associated with the train, the module the train is currently residing on, and when the train's last movement was.

Display Train Consist

```
SET @playerTrain = 4;
SELECT r.CarID, r.CarType, s.ProductType,
 (CASE
  WHEN s.ShipmentID IN (SELECT ShipmentID
   FROM ShipmentsPickedUp
   WHERE ShipmentID = s.ShipmentID)
    AND s.ShipmentID IN (SELECT ShipmentID
     FROM ShipmentsDelivered
     WHERE ShipmentID = s.ShipmentID)
  THEN (SELECT @nextDestination := (SELECT ReturnToYard
     FROM Waybills
     WHERE UsingShipmentID = s.ShipmentID))
  WHEN s.ShipmentID NOT IN (SELECT ShipmentID
   FROM ShipmentsPickedUp
   WHERE ShipmentID = s.ShipmentID)
  THEN (SELECT @nextDestination := (SELECT FromIndustry
     FROM Shipments
     WHERE ShipmentID = s.ShipmentID))
  WHEN s.ShipmentID IN (SELECT ShipmentID
   FROM ShipmentsPickedUp
   WHERE ShipmentID = s.ShipmentID)
  THEN (SELECT @nextDestination := (SELECT ToIndustry
     FROM Shipments
     WHERE ShipmentID = s.ShipmentID))
 END) AS NextDestination,
 IF(@nextDestination IN (SELECT IndustryName
   FROM Industries
   WHERE IndustryName = @nextDestination),
  (SELECT OnModule
   FROM Industries
   WHERE IndustryName = @nextDestination),
  (SELECT OnModule
   FROM Yards
   WHERE YardName = @nextDestination)) AS Module
 FROM RollingStockCars r, Waybills w, Shipments s
 WHERE r.CarID = w.OnCar
  AND w.UsingShipmentID = s.ShipmentID
  AND OnCar IN (SELECT UsingCar
   FROM ConsistedCars
   WHERE OnTrain = @playerTrain);
```

Output:

CarID	CarType	ProductType	NextDestination	Module
XA	Reefer	Dairy	Half Circle Farms	180 Farms
XB	Tank Car	Gasses	LGP Professionals	Pure Oil

The user selects a train to view waybill information. The system displays the train's consist (attached cars). For each car, it displays: car ID, car type, loaded product if exists, and car next destination.

Check-In at Module (Move Train)

```
SET @playerModule = '180 Farms';
SET @playerTrain = 4;
UPDATE TrainLocations SET OnModule = @playerModule WHERE TrainNumber = @playerTrain;
```

The user reports that his or her train has arrived or is doing work at a module. The system records the train's current location and records the time.

Display Module

```
SET @playerModule = '180 Farms';
SELECT *
 FROM Industries
 WHERE IndustryName IN (SELECT IndustryName
      FROM IndustriesAvailable
      WHERE IsAvailable = TRUE)
  AND OnModule = @playerModule;
```

Output:

IndustryName	OnModule	OnMainLine
Half Circle Farms	180 Farms	Red
NULL	NULL	NULL

The user requests to see industries on a module. The system displays active industries on that module and which main line they are on.

Load Rolling Stock (formerly Deliver Rolling Stock)

```
SET @playerIndustry = 'Half Circle Farms';
SET @playerCar = 'XA';
CALL LoadRollingStock(@playerIndustry, @playerCar);
```

The user reports that his or her train is servicing an industry and indicates that a rolling stock car has been dropped off for loading. The system ensures the product type carried matches a product type produced by the industry and verifies that the shipping order has not previously been recorded as being picked up. The car is removed from the train and added to the industry siding. The shipping order is updated to record that the load has been picked up.

Display Rolling Stock At Industry

```
SET @playerIndustry = 'Half Circle Farms';
SELECT c.CarID, c.CarType, i.AtIndustry, i.TimeArrived
 FROM RollingStockCars c
 JOIN RollingStockAtIndustries i ON c.CarID = i.CarID
   WHERE i.AtIndustry = @playerIndustry;
```

Output:

CarID	CarType	AtIndustry	TimeArrived
XA	Reefer	Half Circle Farms	2015-12-05 15:46:25

The user requests to see rolling stock servicing an industry. The system displays, for each car: car ID, car type, and arrival time.

Receive Rolling Stock

```
SET @playerCar = 'XA';
SET @playerTrain = 4;
DELETE FROM RollingStockAtIndustries WHERE CarID = @playerCar;
INSERT INTO ConsistedCars VALUES (@playerTrain, @playerCar, DEFAULT);
```

The user selects a car servicing an industry to add to a train. The system removes the car from the industry siding, consists the car to a train, and records the time.

Check-In at Module (Move Train) (Duplicate Reference)

```
SET @playerModule = 'Black River Yard';
SET @playerTrain = 4;
UPDATE TrainLocations SET OnModule = @playerModule WHERE TrainNumber = @playerTrain;
```

The user reports that his or her train has arrived or is doing work at a module. The system records the train's current location and records the time.

Display Industry

```
SET @playerIndustry = 'MMI Transfer Site 3';
SELECT s.*, p.UsingProductType
 FROM IndustrySidings s
 JOIN IndustryProducts p ON s.ForIndustry = p.ForIndustry
 WHERE p.UsingProductType NOT IN (SELECT ForProductType
  FROM SidingAssignments
        WHERE ForIndustry = @playerIndustry)
   AND s.SidingNumber NOT IN (SELECT SidingNumber
    FROM SidingAssignments
              WHERE ForIndustry = @playerIndustry)
     AND s.ForIndustry = @playerIndustry
UNION SELECT s.*, p.UsingProductType
 FROM IndustrySidings s
 JOIN IndustryProducts p ON s.ForIndustry = p.ForIndustry
 WHERE p.UsingProductType IN (SELECT ForProductType
  FROM SidingAssignments
  WHERE ForIndustry = @playerIndustry)
   AND s.SidingNumber IN (SELECT SidingNumber
    FROM SidingAssignments
    WHERE ForIndustry = @playerIndustry)
     AND s.ForIndustry = @playerIndustry;
```

Output:

ForIndustry	SidingNumber	SidingLength	UsingProductType
MMI Transfer Site 3	1	100	Dairy
MMI Transfer Site 3	1	100	Meats
MMI Transfer Site 3	1	100	Produce
MMI Transfer Site 3	2	100	Dairy
MMI Transfer Site 3	2	100	Meats
MMI Transfer Site 3	2	100	Produce
MMI Transfer Site 3	3	150	General Merchandise
MMI Transfer Site 3	3	150	Manufactured Foods

The user requests to see detailed information about an industry. The system displays industry siding information and identifies which product types can be delivered to specific sidings, if siding assignments are defined.

Unload Rolling Stock (formerly Deliver Rolling Stock)

```
SET @playerIndustry = 'MMI Transfer Site 3';
SET @playerCar = 'XA';
CALL UnloadRollingStock(@playerIndustry, @playerCar);
```

The user reports that his or her train is servicing an industry and indicates that a rolling stock car has been dropped off for unloading. The system ensures the product type carried matches a product type consumed by the industry and verifies that the shipping order has not previously been recorded as being delivered. The car is removed from the train and added to the industry siding. The shipping order is updated to record that the load has been delivered.

Receive Rolling Stock

```
SET @playerCar = 'XA';
SET @playerTrain = 4;
DELETE FROM RollingStockAtIndustries WHERE CarID = @playerCar;
INSERT INTO ConsistedCars VALUES (@playerTrain, @playerCar, DEFAULT);
```

The user selects a car servicing an industry to add to a train. The system removes the car from the industry siding, consists the car to a train, and records the time.

Return Rolling Stock to Yard	
SET @playerTrain = 4; SET @playerCar = 'XA'; SET @playerYard = 'Black River Yard'; DELETE FROM ConsistedCars WHERE OnTrain = @playerTrain AND UsingCar = @playerCar; INSERT INTO RollingStockAtYards VALUES (@playerCar, @playerYard, DEFAULT); DELETE FROM Waybills WHERE OnCar = @playerCar;	The user selects an empty car from a train to return to a yard to be reclassified into other trains. The system removes the car from the player's train. The car is assigned to the yard and the waybill is destroyed.

Remove Crew From Train	
SET @playerTrain = 4; DELETE FROM TrainCrews WHERE OnTrain = @playerTrain;	The user is finished with a game session. The system disassociates a crew from a train.