

Deliverable 2

(CS 425) DATABASE ORGANISATION

Project FLEET Management Systems (FMS)

Create database

```
-- Create database
create database fms;
use fms;
```

Create table

1. -- create vehicle table

```
CREATE TABLE Vehicle (
    ID INT PRIMARY KEY,
    VIN VARCHAR(17) UNIQUE,
    model VARCHAR(255),
    year INT,
    make VARCHAR(255),
    license_plate VARCHAR(15) UNIQUE,
    odometer_reading INT,
    fuel_type VARCHAR(50),
    acquisition_date DATE,    status
    VARCHAR(50),
    assigned_driver_ID INT,
    FOREIGN KEY (assigned_driver_ID) REFERENCES Driver(ID)
);
```

91 • `select * from Vehicle;`
 92
 93

Result Grid

ID	VIN	model	year	make	license_plate	odometer_reading	fuel_type	acquisition_date	status	assigned_driver_ID
1	JTEBU5JR2E5406763	Reatta	1991	Buick	KIY659	72890	electric	2014-04-26	reserved	5
2	WAUMFBFL4DN453...	Odyssey	2012	Honda	XAI191	53324	diesel	2010-04-20	under maintena...	3
3	WAUED54B41N732457	300E	1993	Mercedes-B...	RSF959	72916	hydrogen	2008-02-24	parked	11
4	5N1AN0NW0DN993...	Aspire	1996	Ford	JVO476	58658	coal	2008-03-12	scrapped	11
5	1G6AF5S32D0183837	7 Series	2012	BMW	YJY716	62411	ethanol	2011-10-15	scrapped	2
6	WAUXL68E84A097258	V12 Vant...	2012	Aston Martin	KSU377	66563	butanol	2013-11-11	repossessed	1
7	5TFAW5F19EX570769	Jetta	2010	Volkswagen	CAW879	87819	wood gas	2013-02-22	parked	13
8	1GT3C0BG4AF433415	Jetta	1985	Volkswagen	XHJ452	72214	hybrid	2013-07-28	repossessed	10
9	WBAVD53587A032582	C30	2009	Volvo	PBB278	83659	hybrid	2014-05-08	parked	1
10	SCBFH7ZA6FC217392	Tacoma Xtra	1995	Toyota	TZN285	74760	diesel	2014-09-06	in use	7
11	5NPEB4ACXDH078908	Xterra	2002	Nissan	F1J658	50938	solar	2009-06-23	repossessed	6
12	1GYUCBEF4AR304201	Cruze	2012	Chevrolet	DVA964	95406	coal	2015-04-25	available	14
13	KNAFT4A22A5490080	3 Series	2007	BMW	YFE542	90040	vegetabl...	2010-06-05	repossessed	8
14	WP1AE2AZBL100797	RDX	2012	Acura	AZZ490	74955	biofuel	2015-11-07	repossessed	14
15	WDDHF5GB9AA337...	1500	1997	Chevrolet	DAF632	86480	diesel	2007-03-27	active	15
	NULL NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Vehicle 2 x Apply Revert

2. -- create Driver table

CREATE TABLE Driver (

 ID INT PRIMARY KEY,
 name VARCHAR(255),
 license_number VARCHAR(20) UNIQUE,
 address VARCHAR(255), contact_info
 VARCHAR(255),
 hire_date DATE,
 driving_record TEXT
);

89 • `select* from Driver;`

	ID	name	license_number	address	contact_info	hire_date	driving_record
▶	1	Hephzibah M...	QWE456	951 Victoria Road	546-861-5131	2014-07-20	no violations
	2	Suzi Bilbery	654MNO	36965 Anniversary ...	868-759-7590	2012-04-17	failure to yield citation
	3	Ardyth Bayst...	123ABC	73 Pearson Way	171-431-7158	2014-07-12	minor speeding ticket
	4	Dodi Hannan	VWX789	61915 Ronald Rega...	820-413-0412	2021-03-11	failure to yield citation
	5	Kahlil Van Son	JKL321	69 Hoard Point	308-781-5167	2014-04-05	speeding ticket
	6	Yetta Collyear	321JKL	99 Bobwhite Crossing	697-269-4583	2015-05-15	DUI conviction
	7	Rickie Blaze	MNO987	3 Carey Avenue	655-843-9704	2019-09-15	DUI conviction
	8	Janessa Ricc...	XYZ789	34 Burning Wood P...	435-203-3361	2016-09-05	hit and run incident
	9	Gib Tuson	GHI987	74532 Carberry Ju...	422-244-1268	2018-12-17	no violations
	10	Irma Suller	WHZ789	74 Lindbergh Crossing	273-936-4350	2014-12-02	DUI conviction
	11	Sibley Tonn	DEF654	33 Anthes Avenue	739-366-5416	2018-04-20	improper lane change citation
	12	Bert Emm	IJK987	29432 Westridge Park	152-318-1397	2016-07-10	minor speeding ticket
	13	Burnard Pordall	NOP987	9013 Washington C...	387-359-5518	2012-11-09	improper lane change citation
	14	Olia Morling	LMN321	763 Vermont Parkway	822-671-5027	2016-03-14	speeding ticket
	15	Minne Devey	ABC789	080 Westridge Street	746-224-3342	2017-05-25	suspended license
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Driver 3 x Apply Revert

3. -- create Maintenance table

CREATE TABLE Maintenance (

ID INT PRIMARY KEY,
 vehicle_ID INT, date DATE,
 type VARCHAR(50),
 mileage INT, cost
 DECIMAL(10,2),
 description TEXT,
 technician VARCHAR(255),
 next_service_due DATE,
 part_used INT,
 FOREIGN KEY (vehicle_ID) REFERENCES Vehicle(ID),
 FOREIGN KEY (part_used) REFERENCES Inventory(ID)
);

91 • `select * from Maintenance;`

92

93

94

Result Grid										
Filter Rows:										
Edit:										
Export/Import:										
Wrap Cell Content:										
	ID	vehicle_ID	date	type	mileage	cost	description	technician	next_service_due	part_used
▶	1	1	2015-11-27	corrective	93547	457.55	car wash	Alex Mechanic	2018-05-10	1
	2	1	2018-04-12	preventive	74305	363.81	engine tune-up	Sara Mechanic	2018-07-10	2
	3	1	2015-12-19	corrective	82317	228.71	headlight bulb replacement	Mike Mechanic	2018-04-25	3
	4	4	2018-06-30	corrective	87127	290.30	windshield wiper replacement	Emily Mechanic	2018-12-08	4
	5	2	2018-07-25	corrective	69975	136.09	power steering fluid flush	Jake Mechanic	2019-07-17	5
	6	6	2018-06-05	preventive	82101	307.80	suspension system inspection	Lily Mechanic	2019-12-06	6
	7	7	2017-08-23	preventive	83719	181.04	spark plug replacement	Max Mechanic	2019-01-15	7
	8	4	2017-05-24	preventive	68568	499.58	brake inspection	Sophie Mechanic	2018-03-26	8
	9	6	2020-02-12	preventive	69362	249.25	wheel alignment	Ben Mechanic	2020-09-27	9
	10	7	2019-04-09	corrective	91378	100.47	exhaust system inspection	Olivia Mechanic	2019-06-10	10
	11	12	2020-02-25	corrective	74189	270.90	brake inspection	Luke Mechanic	2020-04-25	11
	12	10	2019-06-17	preventive	68149	279.97	car wash	Ava Mechanic	2019-10-03	12
	13	10	2017-03-04	corrective	84972	280.25	air filter replacement	Chris Mechanic	2018-06-19	13
	14	8	2016-06-25	preventive	63154	275.40	AC recharge	Nora Mechanic	2018-06-28	14
	15	7	2015-05-19	preventive	69906	258.64	suspension system inspection	Ethan Mechanic	2017-08-08	15
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. -- create Fuel table

```
CREATE TABLE Fuel (
  ID INT
  PRIMARY KEY,
  vehicle_ID INT,
  date DATE,
  gallons_purchased
  DECIMAL(10,2),
  odometer_reading
  INT,
  price_per_gallon
  DECIMAL(10,2),
  total_cost
  DECIMAL(10,2),
  fuel_station
  VARCHAR(255),
  FOREIGN KEY (vehicle_ID) REFERENCES Vehicle(ID)
);
```

91 • `select * from Fuel;`

92

93

94

95

96

<

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	ID	vehicle_ID	date	gallons_purchased	odometer_reading	price_per_gallon	total_cost	fuel_station
▶	1	2	2023-05-23	7.52	77851	4.64	34.89	Texo
	2	14	2023-11-14	17.08	84826	3.01	51.41	Pilot
	3	5	2023-08-03	2.07	68140	2.14	4.42	quikr
	4	4	2023-09-08	5.09	53001	3.50	17.81	chevy
	5	9	2023-07-22	25.40	57137	4.53	115.06	Pilot
	6	5	2023-09-19	15.54	88786	2.06	32.01	Wawa
	7	12	2023-08-26	29.01	54209	3.26	94.57	Gulf
	8	6	2023-08-19	18.11	72593	2.22	40.20	Shell
	9	5	2023-06-02	4.13	76767	4.61	19.04	Exxon
	10	1	2023-03-26	23.46	89075	2.55	59.82	Shell
	11	8	2023-11-05	12.90	55434	2.05	26.45	Wawa
	12	13	2023-03-08	24.23	53624	3.76	91.10	Speed
	13	6	2023-03-17	16.51	77166	4.60	75.95	Sheetz
	14	2	2023-03-02	21.05	62802	3.79	79.78	Chevy
	15	7	2023-12-25	3.42	59432	3.45	11.80	BP

5. -- create Trip table

```
CREATE TABLE Trip (
  ID INT PRIMARY KEY,
  vehicle_ID INT,
  driver_ID INT,
  start_datetime DATETIME,
  end_datetime DATETIME,
  start_odometer INT,
  end_odometer INT,
  destination VARCHAR(255),
  purpose VARCHAR(255),
  notes TEXT,
  FOREIGN KEY (vehicle_ID) REFERENCES Vehicle(ID),
  FOREIGN KEY (driver_ID) REFERENCES Driver(ID)
);
```



```

91 • select * from Trip;
92
93
94

```

Result Grid										
Filter Rows: <input type="text"/>										
Edit: Export/Import: Wrap Cell Content:										
	ID	vehicle_ID	driver_ID	start_datetime	end_datetime	start_odometer	end_odometer	destination	purpose	notes
▶	1	12	4	2023-11-27 10:00:00	2023-11-27 01:00:00	67418	67918	client meeting	Business	successful meeting
	2	2	10	2023-04-28 05:00:00	2023-04-28 08:00:00	82030	82530	delivery	work	delivered packages on time
	3	4	15	2023-11-05 04:00:00	2023-11-05 05:30:00	82716	83216	service center	maintenance	routine checkup
	4	13	3	2023-08-03 12:30:00	2023-08-03 03:50:00	76785	77285	Exam	education	successful exam
	5	8	7	2023-06-09 06:15:00	2023-06-09 08:20:00	87323	87823	shopping	personal	completed on time
	6	7	14	2023-12-13 11:07:00	2023-12-13 05:20:00	58039	58539	client meeting	Business	successful meeting
	7	7	11	2023-09-19 03:25:00	2023-09-19 09:40:00	52503	53003	delivery	work	delivered packages on time
	8	3	9	2023-03-02 02:08:00	2023-03-02 03:30:00	69247	69747	service center	maintenance	routine checkup
	9	11	6	2023-06-23 07:20:00	2023-06-23 09:45:00	79906	80406	Exam	education	successful exam
	10	5	12	2023-07-15 01:45:00	2023-07-15 05:05:00	68198	68698	shopping	personal	completed on time
	11	10	12	2023-06-08 08:50:00	2023-06-08 12:00:00	68088	68588	client meeting	Business	successful meeting
	12	1	14	2023-03-06 09:00:00	2023-03-06 01:30:00	83096	83596	delivery	work	delivered packages on time
	13	2	10	2023-07-17 06:20:00	2023-07-17 09:50:00	82778	83278	service center	maintenance	routine checkup
	14	2	9	2023-03-11 03:45:00	2023-03-11 06:45:00	80842	81342	Exam	education	successful exam
	15	11	7	2023-08-21 02:20:00	2023-08-21 05:20:00	65433	65933	shopping	personal	completed on time
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. -- create Inventory table

```

CREATE TABLE Inventory (
ID INT PRIMARY KEY,
name VARCHAR(255),
description TEXT,
stock_level INT, unit_price
DECIMAL(10,2), supplier
VARCHAR(255),
reorder_point INT
);

```

91 • `select * from Inventory`

92

93

94

95

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content

ID	name	description	stock_level	unit_price	supplier	reorder_point
1	engine	High quality engine for various models	64	351.65	AutoMax Suppliers	8
2	tire	All-season radial tire for sedans	55	105.66	Speedy Motors	30
3	brake pad	High quality brake pads for various ...	110	438.89	CarZone Distributors	9
4	headlight	High intensity of light	168	134.43	DriveTime Wholesalers	61
5	bumper	Multi-layer design	32	177.03	WheelWorks Inc.	54
6	radiator	Designing high quality cooling system	78	230.81	FastLane Auto Parts	49
7	spark plug	Generate electrical spark	210	166.48	RoadRunner Supplies	8
8	fuel pump	High quality mechanism to pump fuel	79	477.97	GearUp Motors	43
9	air filter	High performance of filter	153	307.81	AutoPro Distributors	40
10	alternator	Converts AC to DC	182	102.59	MileHigh Auto	81
11	steering ...	Made with high quality raw materials	148	300.35	CarCraft Suppliers	66
12	exhaust ...	Smoke is emitted without disturbance	120	268.77	DriveSmart Wholesa...	54
13	shock ab...	High quality material used for springs	92	73.53	AutoTech Parts	19
14	windshiel...	Made with high quality plastic	36	241.59	WheelWorld Inc.	52
15	catalytic ...	Reduces harmful pollutants emitted	97	94.15	SpeedyDrive Distrib...	36
* NULL	NULL	NULL	NULL	NULL	NULL	NULL







Inventory 1 x Apply Revert

7. -- create Location table

```
CREATE TABLE Location (
  ID INT
  PRIMARY KEY,
  vehicle_ID INT,
  datetime DATETIME,
  GPS_coordinates VARCHAR(255),
  address VARCHAR(255),
  FOREIGN KEY (vehicle_ID) REFERENCES Vehicle(ID)
);
```

```
91 • select * from Location;
```

```
92  
93  
94  
95  
96  
97  
98
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import:  					
Wrap Cell Content: 					
ID	vehicle_ID	datetime	GPS_coordinates	address	
1	6	2023-02-14 08:42:16	55.6761 N	New Orleans	
2	3	2023-06-17 23:59:16	74.0060 W	Boston	
3	3	2023-06-08 15:55:03	51.5074 N	Miami	
4	9	2023-08-26 11:29:25	2.3522 W	Atlanta	
5	10	2023-03-31 10:22:30	139.6917 W	Atlanta	
6	9	2023-06-15 19:08:54	40.4168 N	Philadelphia	
7	2	2023-12-09 07:48:09	74.0060 W	Las Vegas	
8	7	2023-03-25 07:12:03	55.7558 N	Richmond	
9	8	2023-05-07 02:58:54	52.5200 N	Michigan	
10	11	2023-06-29 17:21:26	73.5673 W	Ohio	
11	13	2023-09-13 23:57:54	23.7275 W	Charlotte	
12	15	2023-10-24 13:11:11	34.0522 N	Chicago	
13	7	2023-09-27 03:28:27	55.7558 N	Las Vegas	
14	6	2023-09-24 03:19:20	73.5673 W	California	
15	13	2023-02-01 16:03:36	40.7128 N	Las Vegas	

Index table

-- Create indexes for the Vehicle table

```
CREATE INDEX idx_vehicle_id ON Vehicle(ID);
```

```
CREATE INDEX idx_vehicle_vin ON Vehicle(VIN);
```

```
CREATE INDEX idx_vehicle_model ON Vehicle(model);
```

```
CREATE INDEX idx_vehicle_year ON Vehicle(year);
```

```
CREATE INDEX idx_vehicle_make ON Vehicle(make);
```

```
CREATE INDEX idx_vehicle_license_plate ON Vehicle(license_plate);
```

```
CREATE INDEX idx_vehicle_odometer_reading ON Vehicle(odometer_reading);
```

```
CREATE INDEX idx_vehicle_fuel_type ON Vehicle(fuel_type);
```

```
CREATE INDEX idx_vehicle_acquisition_date ON Vehicle(acquisition_date);
```

```
CREATE INDEX idx_vehicle_status ON Vehicle(status);
```

```
CREATE INDEX idx_vehicle_assigned_driver_id ON Vehicle(assigned_driver_ID);
```



```

86 • CREATE INDEX idx_vehicle_id ON Vehicle (ID);
87 • CREATE INDEX idx_vehicle_vin ON Vehicle (VIN);
88 • CREATE INDEX idx_vehicle_model ON Vehicle (model);
89 • CREATE INDEX idx_vehicle_year ON Vehicle (year);
90 • CREATE INDEX idx_vehicle_make ON Vehicle (make);
91 • CREATE INDEX idx_vehicle_license_plate ON Vehicle (license_plate);
92 • CREATE INDEX idx_vehicle_odometer_reading ON Vehicle (odometer_reading);
93 • CREATE INDEX idx_vehicle_fuel_type ON Vehicle (fuel_type);
94 • CREATE INDEX idx_vehicle_acquisition_date ON Vehicle (acquisition_date);
95 • CREATE INDEX idx_vehicle_status ON Vehicle (status);
96 • CREATE INDEX idx_vehicle_assigned_driver_id ON Vehicle (assigned_driver_ID);

```

Automatic context help is c
get help for the current c

Output			
Action Output			
#	Time	Action	Message
✓ 152	19:12:50	CREATE INDEX idx_vehicle_model ON Vehicle (model)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 153	19:12:50	CREATE INDEX idx_vehicle_year ON Vehicle (year)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 154	19:12:51	CREATE INDEX idx_vehicle_make ON Vehicle (make)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
⚠ 155	19:12:51	CREATE INDEX idx_vehicle_license_plate ON Vehicle (license_plate)	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_
✓ 156	19:12:51	CREATE INDEX idx_vehicle_odometer_reading ON Vehicle (odometer_reading)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 157	19:12:51	CREATE INDEX idx_vehicle_fuel_type ON Vehicle (fuel_type)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 158	19:12:51	CREATE INDEX idx_vehicle_acquisition_date ON Vehicle (acquisition_date)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 159	19:12:52	CREATE INDEX idx_vehicle_status ON Vehicle (status)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 160	19:12:52	CREATE INDEX idx_vehicle_assigned_driver_id ON Vehicle (assigned_driver_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 161	20:31:51	CREATE INDEX idx_vehicle_id ON Vehicle (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✗ 162	20:31:51	CREATE INDEX idx_vehicle_vin ON Vehicle (VIN)	Error Code: 1061. Duplicate key name 'idx_vehicle_vin'

Create indexes for the Driver table

```

CREATE INDEX idx_driver_id ON Driver (ID);
CREATE INDEX idx_driver_name ON Driver t(name);
CREATE INDEX idx_driver_license_number ON Driver t(license_number);
CREATE INDEX idx_driver_address ON Driver t(address);
CREATE INDEX idx_driver_contact_info ON Driver t(contact_info);
CREATE INDEX idx_driver_hire_date ON Driver t(hire_date);

```

```

94 • CREATE INDEX idx_driver_id ON Driver (ID);
95 • CREATE INDEX idx_driver_name ON Driver (name);
96 • CREATE INDEX idx_driver_license_number ON Driver (license_number);
97 • CREATE INDEX idx_driver_address ON Driver (address);
98 • CREATE INDEX idx_driver_contact_info ON Driver (contact_info);
99 • CREATE INDEX idx driver hire date ON Driver (hire date);

```

Output			
Action Output			
#	Time	Action	Message
✓ 163	20:50:27	CREATE INDEX idx_driver_id ON Driver (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 164	20:50:27	CREATE INDEX idx_driver_name ON Driver (name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 165	20:50:28	CREATE INDEX idx_driver_license_number ON Driver (license_number)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 166	20:50:28	CREATE INDEX idx_driver_address ON Driver (address)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 167	20:50:28	CREATE INDEX idx_driver_contact_info ON Driver (contact_info)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 168	20:50:28	CREATE INDEX idx_driver_hire_date ON Driver (hire_date)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Create indexes for the Maintenance table

```

CREATE INDEX idx_maintenance_id ON Maintenance(ID);
CREATE INDEX idx_maintenance_vehicle_id ON Maintenance(vehicle_ID);
CREATE INDEX idx_maintenance_date ON Maintenance(date);
CREATE INDEX idx_maintenance_type ON Maintenance(type);
CREATE INDEX idx_maintenance_mileage ON Maintenance(mileage);
CREATE INDEX idx_maintenance_cost ON Maintenance(cost);
CREATE INDEX idx_maintenance_next_service_due ON Maintenance(next_service_due);
CREATE INDEX idx_maintenance_part_used ON Maintenance(part_used);

```

CREATE INDEX idx_maintenance_technician ON Maintenance(technician);

```
93 • CREATE INDEX idx_maintenance_id ON Maintenance (ID);
94 • CREATE INDEX idx_maintenance_vehicle_id ON Maintenance (vehicle_ID);
95 • CREATE INDEX idx_maintenance_date ON Maintenance (date);
96 • CREATE INDEX idx_maintenance_type ON Maintenance (type);
97 • CREATE INDEX idx_maintenance_mileage ON Maintenance (mileage);
98 • CREATE INDEX idx_maintenance_cost ON Maintenance (cost);
99 • CREATE INDEX idx_maintenance_next_service_due ON Maintenance (next_service_due)
100 • CREATE INDEX idx_maintenance_part_used ON Maintenance (part_used);
101 • CREATE INDEX idx_maintenance_technician ON Maintenance (technician);
102
```

Output

Action Output

#	Time	Action	Message
✓ 169	20:58:06	CREATE INDEX idx_maintenance_id ON Maintenance (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 170	20:58:06	CREATE INDEX idx_maintenance_vehicle_id ON Maintenance (vehicle_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 171	20:58:06	CREATE INDEX idx_maintenance_date ON Maintenance (date)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 172	20:58:06	CREATE INDEX idx_maintenance_type ON Maintenance (type)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 173	20:58:07	CREATE INDEX idx_maintenance_mileage ON Maintenance (mileage)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 174	20:58:07	CREATE INDEX idx_maintenance_cost ON Maintenance (cost)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 175	20:58:07	CREATE INDEX idx_maintenance_next_service_due ON Maintenance (next_service_due)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 176	20:58:07	CREATE INDEX idx_maintenance_part_used ON Maintenance (part_used)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 177	20:58:07	CREATE INDEX idx_maintenance_technician ON Maintenance (technician)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Create indexes for the Fuel table

```
CREATE INDEX idx_fuel_id ON Fuel(ID);
CREATE INDEX idx_fuel_vehicle_id ON Fuel(vehicle_ID);
CREATE INDEX idx_fuel_date ON Fuel(date);
CREATE INDEX idx_fuel_gallons_purchased ON Fuel(gallons_purchased);
CREATE INDEX idx_fuel_odometer_reading ON Fuel(odometer_reading);
CREATE INDEX idx_fuel_price_per_gallon ON Fuel(price_per_gallon);
CREATE INDEX idx_fuel_total_cost ON Fuel(total_cost);
CREATE INDEX idx_fuel_fuel_station ON Fuel(fuel_station);
```

```
93 • CREATE INDEX idx_fuel_id ON Fuel (ID);
94 • CREATE INDEX idx_fuel_vehicle_id ON Fuel (vehicle_ID);
95 • CREATE INDEX idx_fuel_date ON Fuel (date);
96 • CREATE INDEX idx_fuel_gallons_purchased ON Fuel (gallons_purchased);
97 • CREATE INDEX idx_fuel_odometer_reading ON Fuel (odometer_reading);
98 • CREATE INDEX idx_fuel_price_per_gallon ON Fuel (price_per_gallon);
99 • CREATE INDEX idx_fuel_total_cost ON Fuel (total_cost);
100 • CREATE INDEX idx_fuel_fuel_station ON Fuel (fuel_station);
```

Output

Action Output

#	Time	Action	Message
✓ 178	21:06:30	CREATE INDEX idx_fuel_id ON Fuel (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 179	21:06:30	CREATE INDEX idx_fuel_vehicle_id ON Fuel (vehicle_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 180	21:06:30	CREATE INDEX idx_fuel_date ON Fuel (date)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 181	21:06:30	CREATE INDEX idx_fuel_gallons_purchased ON Fuel (gallons_purchased)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 182	21:06:30	CREATE INDEX idx_fuel_odometer_reading ON Fuel (odometer_reading)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 183	21:06:30	CREATE INDEX idx_fuel_price_per_gallon ON Fuel (price_per_gallon)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 184	21:06:30	CREATE INDEX idx_fuel_total_cost ON Fuel (total_cost)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 185	21:06:31	CREATE INDEX idx_fuel_fuel_station ON Fuel (fuel_station)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Create indexes for the Trip table

```
CREATE INDEX idx_trip_id ON Trip(ID);
```



```

CREATE INDEX idx_trip_vehicle_id ON Trip(trip_vehicle_ID);
CREATE INDEX idx_trip_driver_id ON Trip(driver_ID);
CREATE INDEX idx_trip_start_datetime ON Trip(start_datetime);
CREATE INDEX idx_trip_end_datetime ON Trip(end_datetime);
CREATE INDEX idx_trip_start_odometer ON Trip(start_odometer);
CREATE INDEX idx_trip_end_odometer ON Trip(end_odometer);
CREATE INDEX idx_trip_destination ON Trip(destination);
CREATE INDEX idx_fuel_purpose ON Trip(purpose);

```

```

93 • CREATE INDEX idx_trip_id ON Trip (ID);
94 • CREATE INDEX idx_trip_vehicle_id ON Trip (vehicle_ID);
95 • CREATE INDEX idx_trip_driver_id ON Trip (driver_ID);
96 • CREATE INDEX idx_trip_start_datetime ON Trip (start_datetime);
97 • CREATE INDEX idx_trip_end_datetime ON Trip (end_datetime);
98 • CREATE INDEX idx_trip_start_odometer ON Trip (start_odometer);
99 • CREATE INDEX idx_trip_end_odometer ON Trip (end_odometer);
100 • CREATE INDEX idx_trip_destination ON Trip (destination);
101 • CREATE INDEX idx_trip_purpose ON Trip (purpose);

```

#	Time	Action	Message
✓ 186	21:11:48	CREATE INDEX idx_trip_id ON Trip (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 187	21:11:49	CREATE INDEX idx_trip_vehicle_id ON Trip (vehicle_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 188	21:11:49	CREATE INDEX idx_trip_driver_id ON Trip (driver_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 189	21:11:49	CREATE INDEX idx_trip_start_datetime ON Trip (start_datetime)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 190	21:11:49	CREATE INDEX idx_trip_end_datetime ON Trip (end_datetime)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 191	21:11:49	CREATE INDEX idx_trip_start_odometer ON Trip (start_odometer)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 192	21:11:49	CREATE INDEX idx_trip_end_odometer ON Trip (end_odometer)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 193	21:11:49	CREATE INDEX idx_trip_destination ON Trip (destination)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 194	21:11:49	CREATE INDEX idx_trip_purpose ON Trip (purpose)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Create indexes for the Inventory table

```

CREATE INDEX idx_inventory_id ON Inventory(ID);
CREATE INDEX idx_inventory_name ON Inventory(name);
CREATE INDEX idx_inventory_stock_level ON Inventory(stock_level);
CREATE INDEX idx_inventory_unit_price ON Inventory(unit_price);
CREATE INDEX idx_inventory_supplier ON Inventory(supplier);
CREATE INDEX idx_inventory_reorder_point ON Inventory(reorder_point);

```

```

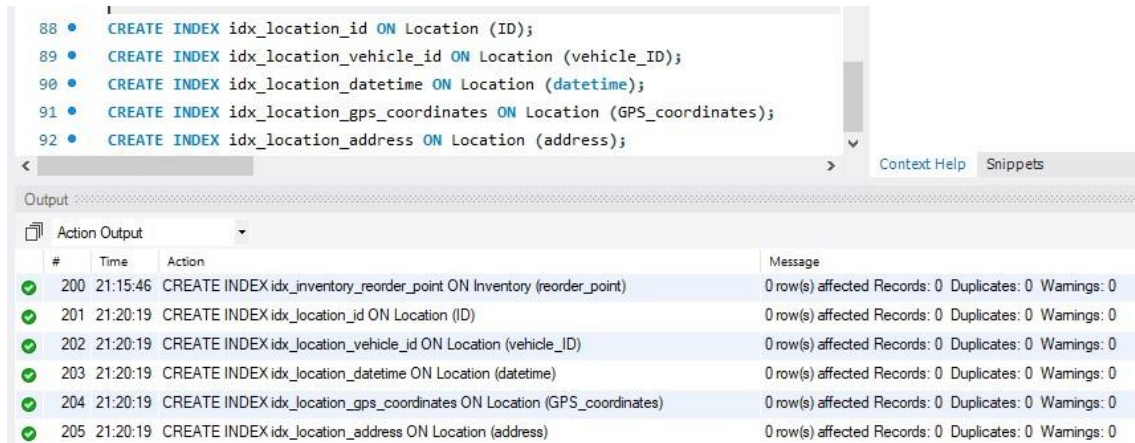
88 • CREATE INDEX idx_inventory_id ON Inventory (ID);
89 • CREATE INDEX idx_inventory_name ON Inventory (name);
90 • CREATE INDEX idx_inventory_stock_level ON Inventory (stock_level);
91 • CREATE INDEX idx_inventory_unit_price ON Inventory (unit_price);
92 • CREATE INDEX idx_inventory_supplier ON Inventory (supplier);
93 • CREATE INDEX idx_inventory_reorder_point ON Inventory (reorder_point);
94
95

```

#	Time	Action	Message
✓ 194	21:11:49	CREATE INDEX idx_trip_purpose ON Trip (purpose)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 195	21:15:46	CREATE INDEX idx_inventory_id ON Inventory (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 196	21:15:46	CREATE INDEX idx_inventory_name ON Inventory (name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 197	21:15:46	CREATE INDEX idx_inventory_stock_level ON Inventory (stock_level)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 198	21:15:46	CREATE INDEX idx_inventory_unit_price ON Inventory (unit_price)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 199	21:15:46	CREATE INDEX idx_inventory_supplier ON Inventory (supplier)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 200	21:15:46	CREATE INDEX idx_inventory_reorder_point ON Inventory (reorder_point)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

-- Create indexes for the Location table

```
CREATE INDEX idx_location_id ON Location(ID);
CREATE INDEX idx_location_vehicle_id ON Location(vehicle_ID);
CREATE INDEX idx_location_datetime ON Location(datetime);
CREATE INDEX idx_location_gps_coordinates ON Location(GPS_coordinates);
CREATE INDEX idx_location_address ON Location(address);
```

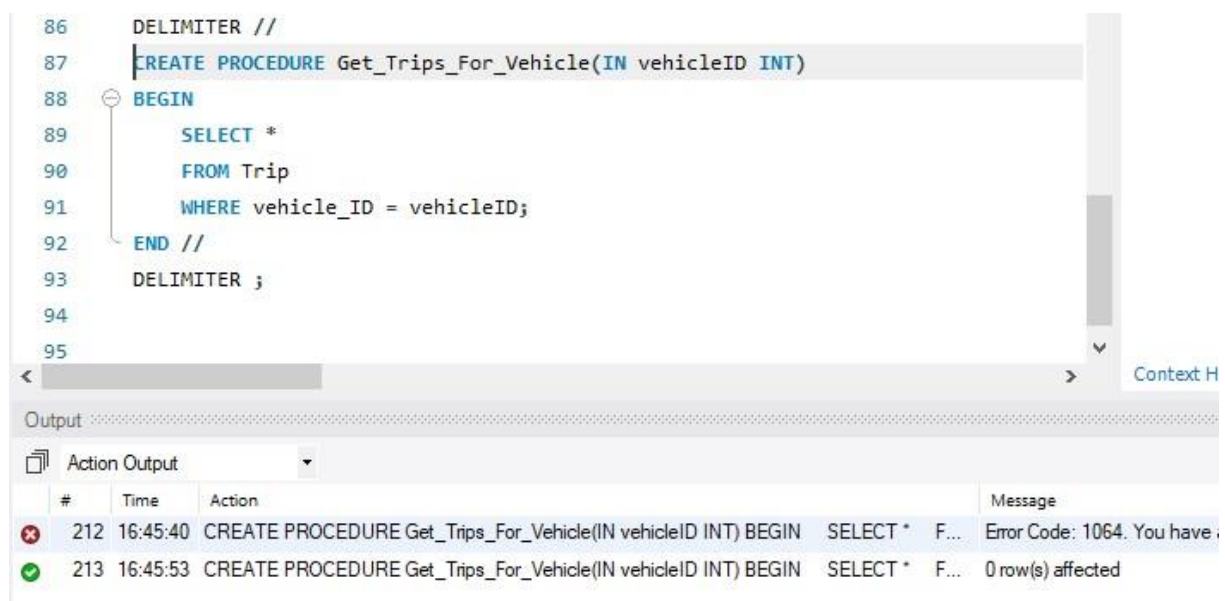


#	Time	Action	Message
200	21:15:46	CREATE INDEX idx_inventory_reorder_point ON Inventory (reorder_point)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
201	21:20:19	CREATE INDEX idx_location_id ON Location (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
202	21:20:19	CREATE INDEX idx_location_vehicle_id ON Location (vehicle_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
203	21:20:19	CREATE INDEX idx_location_datetime ON Location (datetime)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
204	21:20:19	CREATE INDEX idx_location_gps_coordinates ON Location (GPS_coordinates)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
205	21:20:19	CREATE INDEX idx_location_address ON Location (address)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Stored procedures

--Creating a stored procedure to get all trips for a specific vehicle

```
DELIMITER //
CREATE PROCEDURE Get_Trips_For_vehicle(IN vehicleID INT)
BEGIN
    SELECT*
    FROM Trip
    WHERE Vehicle_ID = vehicleID;
END//
DELIMITER;
```



#	Time	Action	Message
212	16:45:40	CREATE PROCEDURE Get_Trips_For_Vehicle(IN vehicleID INT) BEGIN SELECT * F...	Error Code: 1064. You have
213	16:45:53	CREATE PROCEDURE Get_Trips_For_Vehicle(IN vehicleID INT) BEGIN SELECT * F...	0 row(s) affected

```
DELIMITER //
CREATE PROCEDURE UpdateVehicleStatus(
    IN vehicleID INT,
```

```

    IN newOdometerReading INT
)
BEGIN
    DECLARE currentStatus VARCHAR(50);
    -- Get the current status of the vehicle
    SELECT status INTO currentStatus FROM Vehicle WHERE ID = vehicleID;
    -- Update the status based on the odometer reading
    IF newOdometerReading < 50000 THEN
        UPDATE Vehicle SET status = 'Good' WHERE ID = vehicleID;
    ELSEIF newOdometerReading >= 50000 AND newOdometerReading < 100000 THEN
        UPDATE Vehicle SET status = 'Maintenance Required' WHERE ID = vehicleID;
    ELSE
        UPDATE Vehicle SET status = 'Out of Service' WHERE ID = vehicleID;
    END IF;
END //
DELIMITER ;

```

```

87 DELIMITER //
88 CREATE PROCEDURE UpdateVehicleStatus(
89     IN vehicleID INT,
90     IN newOdometerReading INT
91 )
92 BEGIN
93     DECLARE currentStatus VARCHAR(50);
94     -- Get the current status of the vehicle
95     SELECT status INTO currentStatus FROM Vehicle WHERE ID = vehicleID;
96     -- Update the status based on the odometer reading
97     IF newOdometerReading < 50000 THEN
98         UPDATE Vehicle SET status = 'Good' WHERE ID = vehicleID;
99     ELSEIF newOdometerReading >= 50000 AND newOdometerReading < 100000 THEN
100         UPDATE Vehicle SET status = 'Maintenance Required' WHERE ID = vehicleID;
101     ELSE
102         UPDATE Vehicle SET status = 'Out of Service' WHERE ID = vehicleID;
103     END IF;
104 END //
105 DELIMITER ;
106

```

Output

#	Time	Action	Message
15	21:45:09	CREATE TRIGGER Update_Total_Cost BEFORE INSERT ON Fuel FOR EACH ROW BEGI...	0 row(s) affected
16	22:24:39	CREATE PROCEDURE UpdateVehicleStatus(IN vehicleID INT, IN newOdometerReadi...	0 row(s) affected

Temporary table

```

--Creating a temporary table to store fuel purchases for vehicles with a total cost greater than 50
CREATE TEMPORARY TABLE Temp_Fuel_Purchases AS
SELECT*
FROM Fuel

```


WHERE total_cost > 50;

```
88 • CREATE TEMPORARY TABLE Temp_Fuel_Purchases AS
89 SELECT *
90 FROM Fuel
91 WHERE total_cost > 50;
```

Output				Context Help	Snippets
Action Output					
#	Time	Action	Message		
✓ 203	21:20:19	CREATE INDEX idx_location_datetime ON Location (datetime)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0		
✓ 204	21:20:19	CREATE INDEX idx_location_gps_coordinates ON Location (GPS_coordinates)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0		
✓ 205	21:20:19	CREATE INDEX idx_location_address ON Location (address)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0		
✓ 206	16:18:51	CREATE VIEW Vehicle_Driver_Info AS SELECT v.ID AS Vehicle_ID, v.license_plate, d.name...	0 row(s) affected		
✓ 207	16:22:46	CREATE TEMPORARY TABLE Temp_Fuel_Purchases AS SELECT * FROM Fuel WHERE ...	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0		

Create a temporary table for a summary of trip related information

CREATE TEMPORARY TABLE Temp_Trip_Summary AS

SELECT

t.ID AS Trip_ID, v.VIN, v.model, d.name AS Driver_Name, t.start_datetime, t.end_datetime,

t.start_odometer, t.end_odometer, t.destination, t.purpose, t.notes FROM

Trip t

JOIN Vehicle v ON t.vehicle_ID = v.ID

JOIN Driver d ON t.driver_ID = d.ID;

```
86 • CREATE TEMPORARY TABLE Temp_Trip_Summary AS
87 SELECT
88     t.ID AS Trip_ID,
89     v.VIN,
90     v.model,
91     d.name AS Driver_Name,
92     t.start_datetime,
93     t.end_datetime,
94     t.start_odometer,
95     t.end_odometer,
96     t.destination,
97     t.purpose,
98     t.notes
99 FROM
100     Trip t
101 JOIN Vehicle v ON t.vehicle_ID = v.ID
102 JOIN Driver d ON t.driver_ID = d.ID;
103
```

Output			
Action Output			
#	Time	Action	Message
✓ 12	21:14:57	CREATE VIEW MaintenanceCostView AS SELECT m.vehicle_ID AS VehicleID, SUM(m...	0 row(s) affected
✓ 13	21:20:10	CREATE TEMPORARY TABLE Temp_Trip_Summary AS SELECT t.ID AS Trip_ID, v.VI...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0

Views

--Create a view to display vehicles with their assigned drivers

CREATE VIEW Vehicle_Driver_Info AS

SELECT v.ID AS Vehicle_ID, v.license_plate, d.name AS Driver_Name

FROM Vehicle v

LEFT JOIN Driver d ON v.assigned_driver_ID = d.ID;

```
-- 88 • CREATE VIEW Vehicle_Driver_Info AS
89 SELECT v.ID AS Vehicle_ID, v.license_plate, d.name AS Driver_Name
90 FROM Vehicle v
91 LEFT JOIN Driver d ON v.assigned_driver_ID = d.ID;
```

Output

#	Time	Action	Message
✓ 202	21:20:19	CREATE INDEX idx_location_vehicle_id ON Location (vehicle_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 203	21:20:19	CREATE INDEX idx_location_datetime ON Location (datetime)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 204	21:20:19	CREATE INDEX idx_location_gps_coordinates ON Location (GPS_coordinates)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 205	21:20:19	CREATE INDEX idx_location_address ON Location (address)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 206	16:18:51	CREATE VIEW Vehicle_Driver_Info AS SELECT v.ID AS Vehicle_ID, v.license_plate, d.name AS Driver_Name	0 row(s) affected

Create a view to display fuel usage for each vehicle

```
CREATE VIEW FuelUsageView AS
SELECT
    f.vehicle_ID AS VehicleID,
    SUM(f.gallons_purchased) AS TotalGallonsPurchased,
    SUM(f.total_cost) AS TotalFuelCost
FROM Fuel f
GROUP BY f.vehicle_ID;
```

```
-- 86 • CREATE VIEW FuelUsageView AS
87 SELECT
88     f.vehicle_ID AS VehicleID,
89     SUM(f.gallons_purchased) AS TotalGallonsPurchased,
90     SUM(f.total_cost) AS TotalFuelCost
91 FROM
92     Fuel f
93 GROUP BY
94     f.vehicle_ID;
95
```

Output

#	Time	Action	Message
✓ 10	21:06:43	CREATE FUNCTION Get_Average_Trip_Distance(vehicleID INT) RETURNS DECIMAL(10,2)...	0 row(s) affected
✓ 11	21:10:47	CREATE VIEW FuelUsageView AS SELECT f.vehicle_ID AS VehicleID, SUM(f.gallons_purchased) AS TotalGallonsPurchased, SUM(f.total_cost) AS TotalFuelCost	0 row(s) affected

Create a view to display total maintenance cost for each vehicle

```
CREATE VIEW MaintenanceCostView AS
SELECT
    m.vehicle_ID AS VehicleID, SUM(m.cost) AS TotalMaintenanceCost
FROM Maintenance m
GROUP BY m.vehicle_ID;
```

```

86 • CREATE VIEW MaintenanceCostView AS
87 SELECT
88     m.vehicle_ID AS VehicleID,
89     SUM(m.cost) AS TotalMaintenanceCost
90 FROM
91     Maintenance m
92 GROUP BY
93     m.vehicle_ID;
94

```

Output				
Action Output				
#	Time	Action	Message	
✓ 11	21:10:47	CREATE VIEW FuelUsageView AS SELECT f.vehicle_ID AS VehicleID, SUM(f.gallons_...	0 row(s) affected	
✓ 12	21:14:57	CREATE VIEW MaintenanceCostView AS SELECT m.vehicle_ID AS VehicleID, SUM(m....	0 row(s) affected	

Triggers

---- Creating a trigger to update the odometer_reading of a vehicle after a trip

DELIMITER //

```

CREATE TRIGGER Update_Odometer
AFTER INSERT ON Trip
FOR EACH ROW
BEGIN
    UPDATE Vehicle
    SET odometer_reading = NEW.end_odometer
    WHERE ID = NEW.vehicle_ID;
END;
//
DELIMITER;

```

```

-- 86 DELIMITER //
87 |
88 • CREATE TRIGGER Update_Odometer
89 AFTER INSERT ON Trip
90 FOR EACH ROW
91 BEGIN
92     UPDATE Vehicle
93     SET odometer_reading = NEW.end_odometer
94     WHERE ID = NEW.vehicle_ID;
95 END;
96
97 //
98
99 DELIMITER ;
100
101

```

Output				
Action Output				
#	Time	Action	Message	
✓ 13	21:20:10	CREATE TEMPORARY TABLE Temp_Trip_Summary AS SELECT t.ID AS Trip_ID, v.VI...	15 row(s) affected Records: 15 I	
✓ 14	21:38:32	CREATE TRIGGER Update_Odometer AFTER INSERT ON Trip FOR EACH ROW BEGIN ...	0 row(s) affected	

Creating a trigger to update the total_cost when a new fuel record is inserted.

```
DELIMITER //
```

```
CREATE TRIGGER Update_Total_Cost
BEFORE INSERT ON Fuel
FOR EACH ROW
BEGIN
    SET NEW.total_cost = NEW.gallons_purchased * NEW.price_per_gallon;
END //
```

```
DELIMITER;
```



```
86 DELIMITER //
87
88 CREATE TRIGGER Update_Total_Cost
89 BEFORE INSERT ON Fuel
90 FOR EACH ROW
91 BEGIN
92     SET NEW.total_cost = NEW.gallons_purchased * NEW.price_per_gallon;
93 END //
94
95 DELIMITER ;
96
```

Output

Action Output

#	Time	Action	Message
✓ 14	21:38:32	CREATE TRIGGER Update_Odometer AFTER INSERT ON Trip FOR EACH ROW BEGIN ...	0 row(s) affected
✓ 15	21:45:09	CREATE TRIGGER Update_Total_Cost BEFORE INSERT ON Fuel FOR EACH ROW BEGI...	0 row(s) affected

Functions

--Create a function to calculate total inventory cost

```
DELIMITER //
CREATE FUNCTION Calculate_Total_Inventory_Cost()
RETURNS DECIMAL(10,2)
DETERMINISTIC
READS SQL DATA
BEGIN
    DECLARE total_cost DECIMAL(10,2);
    SELECT SUM(stock_level * unit_price) INTO total_cost
    FROM Inventory;
    IF total_cost IS NULL THEN
        SET total_cost = 0.00;
    END IF;
    RETURN total_cost;
END //
DELIMITER;
```

```

86 DELIMITER //
87
88 • CREATE FUNCTION Calculate_Total_Inventory_Cost()
89 RETURNS DECIMAL(10,2)
90 DETERMINISTIC
91 READS SQL DATA
92 BEGIN
93     DECLARE total_cost DECIMAL(10,2);
94
95     SELECT SUM(stock_level * unit_price) INTO total_cost
96     FROM Inventory;
97
98     IF total_cost IS NULL THEN
99         SET total_cost = 0.00;
100     END IF;
101
102     RETURN total_cost;
103 END //
104
105 DELIMITER ;

```

Output

#	Time	Action	Message
2	20:45:55	CREATE FUNCTION Calculate_Total_Inventory_Cost() RETURNS DECIMAL(10,2) BEGIN ...	Error Code: 1418. This fun
3	20:48:16	CREATE FUNCTION Calculate_Total_Inventory_Cost() RETURNS DECIMAL(10,2) DETER...	0 row(s) affected

Create a function to calculate total maintenance cost for a vehicle

```

DELIMITER//
CREATE FUNCTION calculate_Total_Maintenance_Cost(vehicleID INT)
RETURNS DECIMAL
DETERMINISTIC
READS SQL DATA
BEGIN
    DECLARE total_cost DECIMAL(10,2);
    SELECT SUM(cost) INTO total_cost
    FROM Maintenance
    WHERE vehicle_ID = vehicleID;
    RETURN total_cost;
END;

```

```

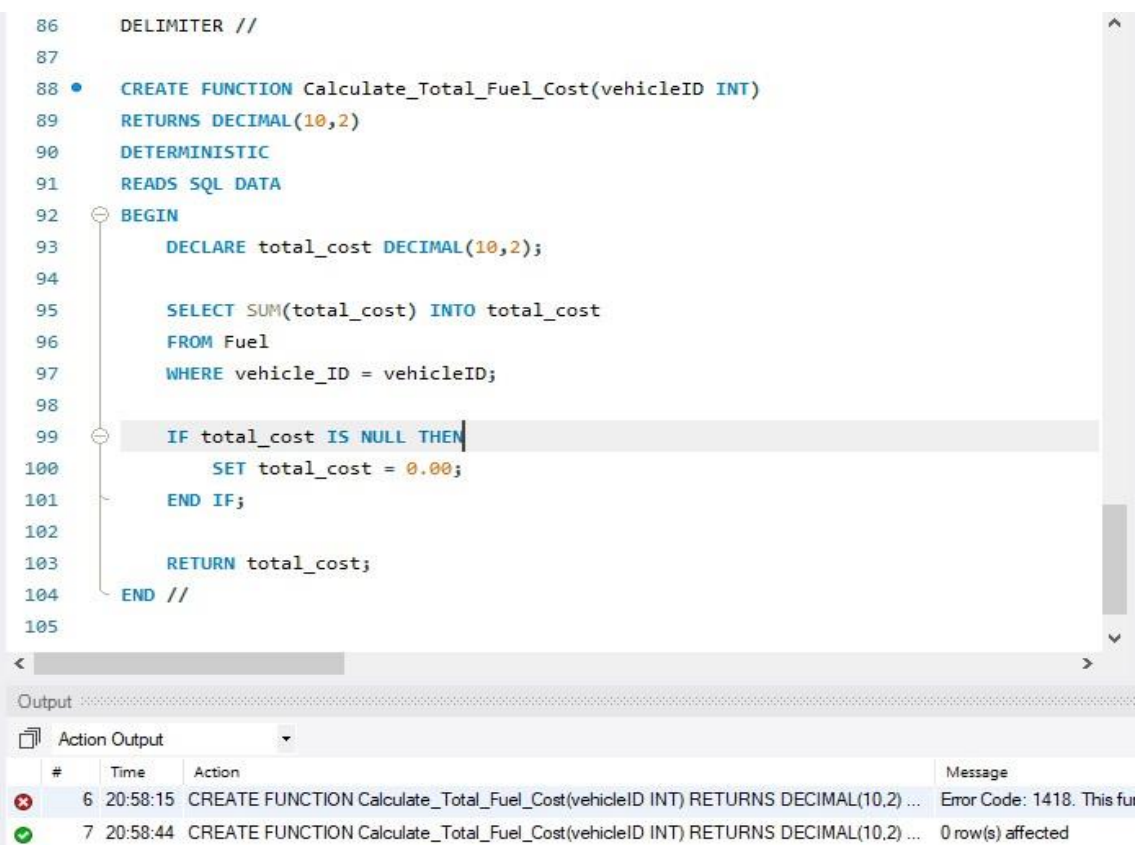
86 DELIMITER //
87
88 • CREATE FUNCTION Calculate_Total_Maintenance_Cost(vehicleID INT)
89 RETURNS DECIMAL(10,2)
90 DETERMINISTIC
91 READS SQL DATA
92 BEGIN
93     DECLARE total_cost DECIMAL(10,2);
94     SELECT SUM(cost) INTO total_cost
95     FROM Maintenance
96     WHERE vehicle_ID = vehicleID;
97     RETURN total_cost;
98 END;
99

```

Output

#	Time	Action	Message
3	20:48:16	CREATE FUNCTION Calculate_Total_Inventory_Cost() RETURNS DECIMAL(10,2) DETER...	0 row(s) affected
4	20:51:08	CREATE FUNCTION Calculate_Total_Maintenance_Cost(vehicleID INT) RETURNS DECIM...	0 row(s) affected


```
--Create a function to calculate a total fuel cost of a vehicle
DELIMITER //
CREATE FUNCTION Calculate_Total_Fuel_Cost(vehicleID INT)
RETURNS DECIMAL(10,2)
DETERMINISTIC
READS SQL DATA
BEGIN
    DECLARE total_cost DECIMAL(10,2);
    SELECT SUM(total_cost) INTO total_cost
    FROM Fuel
    WHERE vehicle_ID = vehicleID;
    IF total_cost IS NULL THEN
        SET total_cost = 0.00;
    END IF;
    RETURN total_cost;
END //
DELIMITER ;
```



```
86 DELIMITER //
87
88 • CREATE FUNCTION Calculate_Total_Fuel_Cost(vehicleID INT)
89 RETURNS DECIMAL(10,2)
90 DETERMINISTIC
91 READS SQL DATA
92 BEGIN
93     DECLARE total_cost DECIMAL(10,2);
94
95     SELECT SUM(total_cost) INTO total_cost
96     FROM Fuel
97     WHERE vehicle_ID = vehicleID;
98
99     IF total_cost IS NULL THEN
100         SET total_cost = 0.00;
101     END IF;
102
103     RETURN total_cost;
104 END //
105
```

Output

#	Time	Action	Message
6	20:58:15	CREATE FUNCTION Calculate_Total_Fuel_Cost(vehicleID INT) RETURNS DECIMAL(10,2) ...	Error Code: 1418. This fu
7	20:58:44	CREATE FUNCTION Calculate_Total_Fuel_Cost(vehicleID INT) RETURNS DECIMAL(10,2) ...	0 row(s) affected

```
Create a function to get next service due date of vehicles
DELIMITER //
CREATE FUNCTION Get_Next_Service_Due_Date(vehicleID INT)
RETURNS DATE
BEGIN
    DECLARE next_due_date DATE;
    SELECT MIN(next_service_due) INTO next_due_date
```

```

FROM Maintenance
WHERE vehicle_ID = vehicleID;
IF next_due_date IS NULL THEN
    SET next_due_date = '9999-12-31'; -- Default date if no next service due date is found
END IF;
RETURN next_due_date;
END //
DELIMITER;

```

The screenshot shows a SQL IDE with a script editor and an output window. The script editor contains the following code:

```

86 DELIMITER //
87
88 CREATE FUNCTION Get_Next_Service_Due_Date(vehicleID INT)
89 RETURNS DATE
90 DETERMINISTIC
91 READS SQL DATA
92 BEGIN
93     DECLARE next_due_date DATE;
94
95     SELECT MIN(next_service_due) INTO next_due_date
96     FROM Maintenance
97     WHERE vehicle_ID = vehicleID;
98
99     IF next_due_date IS NULL THEN
100         SET next_due_date = '9999-12-31'; -- Default date if no next service due date
101     END IF;
102
103     RETURN next_due_date;
104 END //
105

```

The output window shows the following results:

#	Time	Action	Message
7	20:58:44	CREATE FUNCTION Calculate_Total_Fuel_Cost(vehicleID INT) RETURNS DECIMAL(10,2) ...	0 row(s) affected
8	21:00:51	CREATE FUNCTION Get_Next_Service_Due_Date(vehicleID INT) RETURNS DATE DETERMINISTIC ...	0 row(s) affected

Create a function to get average trip distance of vehicle

```

DELIMITER //
CREATE FUNCTION Get_Average_Trip_Distance(vehicleID INT)
RETURNS DECIMAL(10,2) BEGIN
    DECLARE total_distance DECIMAL(10,2);
    DECLARE trip_count INT;
    SELECT SUM(end_odometer-start_odometer) INTO total_distance, COUNT(*) INTO
trip_count FROM Trip
    WHERE vehicle_ID = vehicleID;
    IF trip_count = 0 THEN
        RETURN 0.00; -- Avoid division by zero
    END IF;
    RETURN total_distance / trip_count;
END //
DELIMITER ;

```

```

86 DELIMITER //
87
88 • CREATE FUNCTION Get_Average_Trip_Distance(vehicleID INT)
89 RETURNS DECIMAL(10,2) DETERMINISTIC READS SQL DATA
90 BEGIN
91     DECLARE total_distance DECIMAL(10,2);
92     DECLARE trip_count INT;
93
94     SELECT SUM(end_odometer - start_odometer), COUNT(*) INTO total_distance, trip_count
95     FROM Trip
96     WHERE vehicle_ID = vehicleID;
97
98     IF trip_count = 0 THEN
99         RETURN 0.00; -- Avoid division by zero
100     END IF;
101
102     RETURN total_distance / trip_count;
103 END //
104
105 DELIMITER ;

```

Output

Action Output

#	Time	Action	Message
9	21:05:29	CREATE FUNCTION Get_Average_Trip_Distance(vehicleID INT) RETURNS DECIMAL(10,2...	Error Code: 1064. You have an error
10	21:06:43	CREATE FUNCTION Get_Average_Trip_Distance(vehicleID INT) RETURNS DECIMAL(10,2...	0 row(s) affected