Punjab University College of Information Technology (PUCIT)

Project Title:

Ride Sharing System

Course Name:

Database System

Submitted to:

DR. ASIF SOHAIL

Submitted By:

Roll No	Name
BITF24M018	Muhammad Haroon
BITF24M036	Umar Draz
BITF24M041	Talha Yaseen

Introduction

This Ride/Driver Management System is a comprehensive software solution designed to streamline ride-hailing operations and enhance administrative processes within a transportation service. At its core, the system manages customer-related information, including customer identification (CUSTOMERID) and associated details such as names, phone numbers, and email addresses.

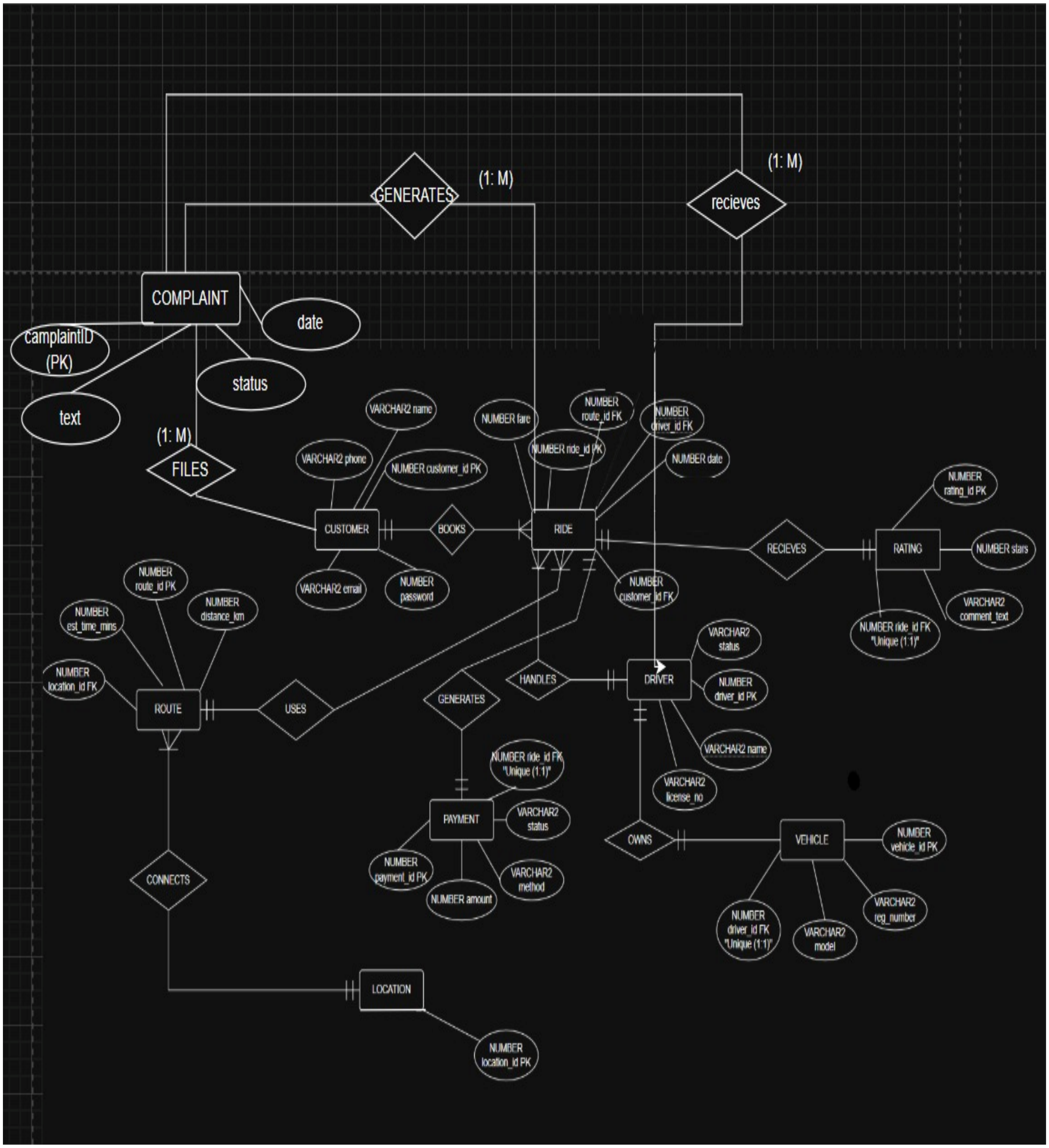
The system also manages drivers through the DRIVER table, capturing essential details like driver identification (DRIVERID), names, contact information, license numbers, current status (ACTIVE, INACTIVE, ON RIDE), and ratings. Each driver may operate one or more vehicles, with vehicle details stored in the VEHICLE table, including vehicle type, model, plate number, and operational status. Vehicles are directly linked to their respective drivers via foreign keys.

To handle ride logistics, the system incorporates LOCATION and ROUTE tables. LOCATION stores city and area details, while ROUTE maps pickup and drop locations and the corresponding distance in kilometers. The RIDE table integrates customer, driver, vehicle, and route information, recording each ride's date, fare, and current status (PENDING, COMPLETED, CANCELLED).

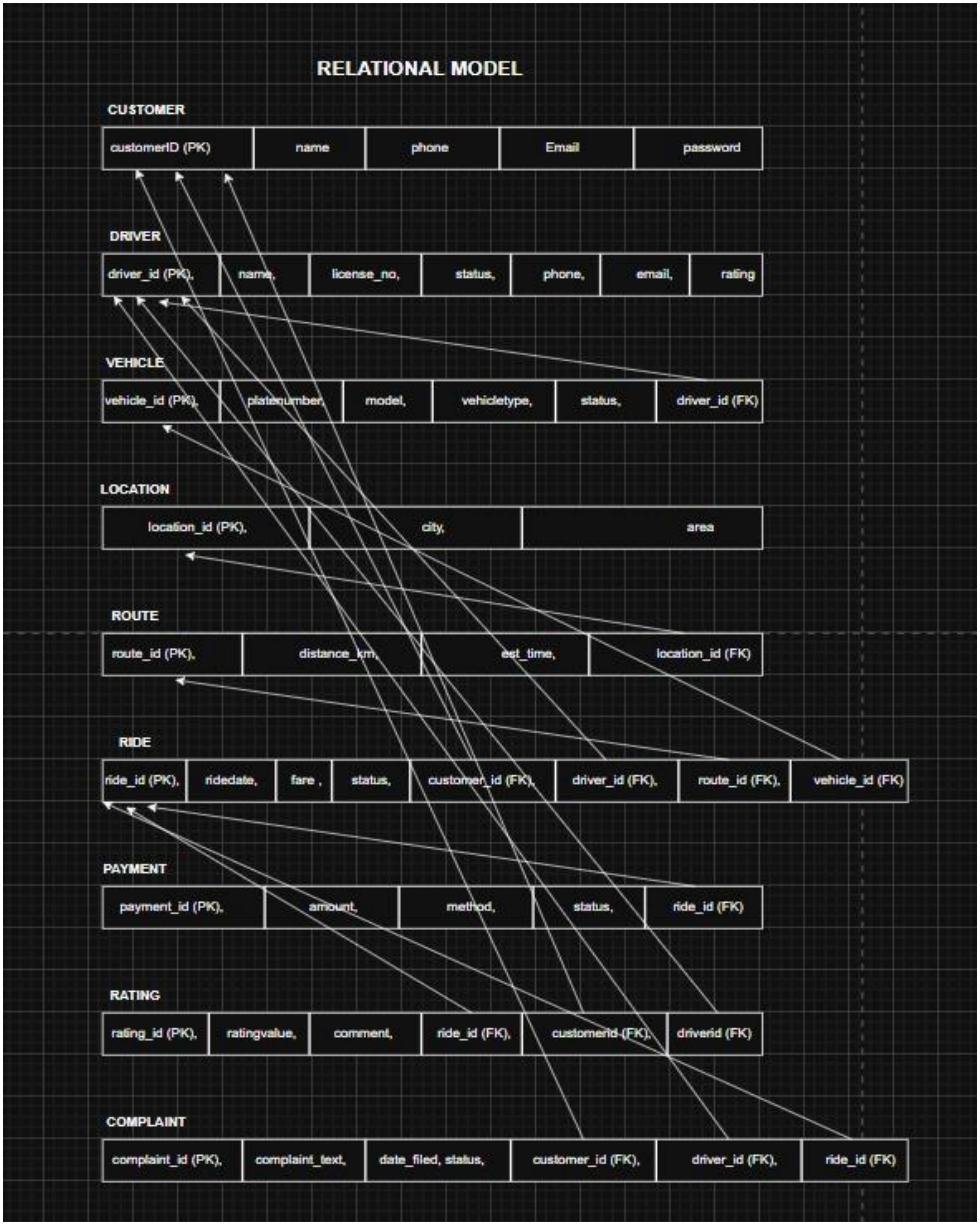
Payment management is facilitated through the PAYMENT table, which tracks ride payments, payment methods (CASH, CARD, WALLET), and payment statuses (PAID, PENDING, FAILED). Additionally, the system captures feedback and service quality through the RATING table, linking ratings to riders, customers, and drivers. Complaints or service issues are recorded in the COMPLAINT table, documenting complaint details, filing dates, and resolution statuses.

This system effectively integrates all key components needed for managing customers, drivers, vehicles, rides, payments, ratings, and complaints, providing a robust framework for efficient ride management and enhanced user satisfaction.

ERD OF RIDE SHARING SYSTEM



RELATIONAL MODEL OF RIDE SHARING SYSTEM



RELATIONAL MODEL

CUSTOMER

customerID (PK)	name	phone	Email	password

DRIVER

driver_id (PK),	name,	license_no,	status,	phone,	email,	rating

VEHICLE

vehicle_id (PK),	platenumber,	model,	vehicletype,	status,	driver_id (FK)

LOCATION

location_id (PK),	city,	area

ROUTE

route_id (PK),	distance_km,	est_time,	location_id (FK)

RIDE

ride_id (PK),	ridedate,	fare ,	status,	customer_id (FK),	driver_id (FK),	route_id (FK),	vehicle_id (FK)

PAYMENT

payment_id (PK),	amount,	method,	status,	ride_id (FK)

RATING

rating_id (PK),	ratingvalue,	comment,	ride_id (FK),	customerid (FK),	driverid (FK)

COMPLAINT

complaint_id (PK),	complaint_text,	date_filed, status,	customer_id (FK),	driver_id (FK),	ride_id (FK)

Relational Schemas

1. CUSTOMER(CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD)

Attributes:

CUSTOMERID: Unique identifier for each customer (Primary Key)

NAME: Full name of the customer

PHONE: Customer's phone number (unique)

EMAIL: Customer's email address (unique)

PASSWORD: Customer account password

Primary Key (PK): CUSTOMERID

2. DRIVER(DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING)

Attributes:

DRIVERID: Unique identifier for each driver (Primary Key)

NAME: Full name of the driver

PHONE: Driver's phone number (unique)

EMAIL: Driver's email address (unique)

LICENSENUMBER: Driver license number

STATUS: Current status of the driver (Active, Offline, On Ride)

RATING: Driver rating (0–5)

Primary Key (PK): DRIVERID

3. VEHICLE(VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)

Attributes:

VEHICLEID: Unique identifier for each vehicle (Primary Key)

DRIVERID: References the driver operating the vehicle (Foreign Key)

VEHICLETYPE: Type of vehicle (Car, Bike, Rickshaw, Van, etc.)

MODEL: Vehicle model

PLATENUMBER: Vehicle registration plate (unique)

STATUS: Current status of the vehicle (Active, Offline)

Primary Key (PK): VEHICLEID

Foreign Key (FK): DRIVERID references DRIVER(DRIVERID)

4. LOCATION(LOCATIONID, CITY, AREA)

Attributes:

LOCATIONID: Unique identifier for each location (Primary Key)

CITY: City name

AREA: Area or locality

Primary Key (PK): LOCATIONID

5. ROUTE(ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM)

Attributes:

ROUTEID: Unique identifier for each route (Primary Key)

PICKUPLOCATION: References LOCATIONID of the pickup point (Foreign Key)

DROPLOCATION: References LOCATIONID of the drop point (Foreign Key)

DISTANCEKM: Distance in kilometers (must be greater than 0)

Primary Key (PK): ROUTEID

Foreign Keys (FKs):

PICKUPLOCATION references LOCATION(LOCATIONID)

DROPLOCATION references LOCATION(LOCATIONID)

6. RIDE(RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE, STATUS)

Attributes:

RIDEID: Unique identifier for each ride (Primary Key)

CUSTOMERID: References to the customer booking the ride (Foreign Key)

DRIVERID: References the driver assigned (Foreign Key)

VEHICLEID: References the vehicle used (Foreign Key)

ROUTEID: References on the route taken (Foreign Key)

RIDEDATE: Date of the ride

FARE: Fare amount (must be greater than 0)

STATUS: Current ride status (Pending, Completed, Cancelled)

Primary Key (PK): RIDEID

Foreign Keys (FKs):

CUSTOMERID references CUSTOMER(CUSTOMERID)

DRIVERID references DRIVER(DRIVERID)

VEHICLEID references VEHICLE(VEHICLEID)

ROUTEID references ROUTE(ROUTEID)

7. PAYMENT(PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS)

Attributes:

PAYMENTID: Unique identifier for each payment (Primary Key)

RIDEID: References the ride associated with the payment (Foreign Key)

AMOUNT: Payment amount

METHOD: Payment method (Cash, Card, Wallet)

STATUS: Payment status (Paid, Pending, Failed)

Primary Key (PK): PAYMENTID

Foreign Key (FK): RIDEID references RIDE(RIDEID)

8. RATING(RATINGID, RIDEID, CUSTOMERID, DRIVERID, RATINGVALUE, COMMENT)

Attributes:

RATINGID: Unique identifier for each rating (Primary Key)

RIDEID: References the ride being rated (Foreign Key)

CUSTOMERID: References the customer giving the rating (Foreign Key)

DRIVERID: References the driver being rated (Foreign Key)

RATINGVALUE: Rating value (1–5)

COMMENT: Optional text comment

Primary Key (PK): RATINGID

Foreign Keys (FKs):

RIDEID references RIDE(RIDEID)

CUSTOMERID references CUSTOMER(CUSTOMERID)

DRIVERID references DRIVER(DRIVERID)

9. COMPLAINT(COMPLAINTID, RIDEID, CUSTOMERID, DRIVERID, COMPLAINTTEXT, DATEFILED, STATUS)

Attributes:

COMPLAINTID: Unique identifier for each complaint (Primary Key)

RIDEID: References the ride associated with the complaint (Foreign Key)

CUSTOMERID: References to the customer filing the complaint (Foreign Key)

DRIVERID: References the driver involved (Foreign Key)

COMPLAINTTEXT: Description of the complaint

DATEFILED: Date the complaint was filed (default to current date)

STATUS: Complaint status (Pending, Resolved, Rejected)

Primary Key (PK): COMPLAINTID

Foreign Keys (FKs):

RIDEID references RIDE(RIDEID)

CUSTOMERID references CUSTOMER(CUSTOMERID)

DRIVERID references DRIVER(DRIVERID)

Description of the Relations

1) Customer Table

Attribute	Data Type	Size	Constraints
Customer ID	Number	4	Primary key
Name	Varchar2	50	Not Null
Phone	Varchar2	15	Unique
Email	Varchar2	50	Unique
Password	Varchar2	20	Not Null

2) Driver Table

Attribute	Data Type	Size	Constraints
Driver ID	Number	4	Primary key
Name	Varchar2	50	Not Null
Phone	Varchar2	15	Unique
Email	Varchar2	50	Unique
License No	Varchar2	50	Not
Status	Varchar2	15	CHECK(STATUS IN('ACTIVE','INACTIVE','ON RIDE')) DEFAULT 'ACTIVE
Rating	Number	2	Check(rating between 0 and 5)

3) Vehicle Table

Attribute	Data Type	Size	Constraints
Vehicle ID	Number	4	Primary key
Driver ID	Number	4	N/A
Vehicle Type	Varchar2	20	Not Null
Model	Varchar2	20	Not Null
Plate Number	Varchar2	15	Not Null
Status	Varchar2	15	Foreign Key(driver Id) References Driver(DriverID)

4) Location Table

Attribute	Data Type	Size	Constraints
Location ID	Number	4	Primary key
City	Varchar2	50	Not Null
Area	Varchar2	15	Not Null

5) Route Table

Attribute	Data Type	Size	Constraints
Route ID	Number	4	Primary key
Pick up Location	Number	4	N/A
Drop Location	**Number**	**4**	**N/A**
Distance Km	**Number**	**5**	**Check(Distance Km>0), Forkey key(Pick up Location) references Location(LocationID), Foreign key(Drop Location) references Location(Location ID)**

6) Ride Table

Attribute	Data Type	Size	Constraints
Ride ID	Number	4	Primary key
Customer ID	Number	4	N/A
Driver ID	**Number**	**4**	**N/A**
Vehicle ID	**Number**	**4**	**N/A**
Route ID	**Number**	**4**	**N/A**
Ride Date	**Date**		**Not Null**
Fare	**Number**	**4**	**Check(Fare>0)**
Status	**Varchar2**	**15**	**CHECK(STATUS IN('PENDING','COMPLETED','CANCELLED')) DEFAULT 'PENDING',FOREIGN KEY(CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID),FOREIGN KEY(DRIVERID) REFERENCES DRIVER(DRIVERID),FOREIGN KEY(VEHICLEID) REFERENCES VEHICLE(VEHICLEID),FOREIGN KEY(ROUTEID) REFERENCES ROUTE(ROUTEID)**

7) Payment Table

Attribute	Data Type	Size	Constraints
Payment ID	Number	4	Primary key
Ride ID	Number	4	N/A
Amount	Number	4	Not Null
Method	Varchar2	30	CHECK(METHOD IN('CASH','CARD','WALLET')),STATUS VARCHAR2(15) CHECK(STATUS

			IN('PAID','PENDING','FAILED')) DEFAULT 'PENDING',FOREIGN KEY(RIDEID) REFERENCES RIDE(RIDEID)

8) Rating Table

Attribute	Data Type	Size	Constraints
Rating ID	Number	4	Primary key
Ride ID	Number	4	N/A
Customer ID	Number	4	N/A
Driver ID	Number	4	N/A
Rating Value	Number	2	Check(Rating Value between 1 and 5)
Comment	Varchar2	100	Foreignn key(Ride ID) references Ride(Ride ID), Foreign Key (Customer ID) References Customer(Customer ID), Foreign Key (Driver ID) References Driver(Driver ID)

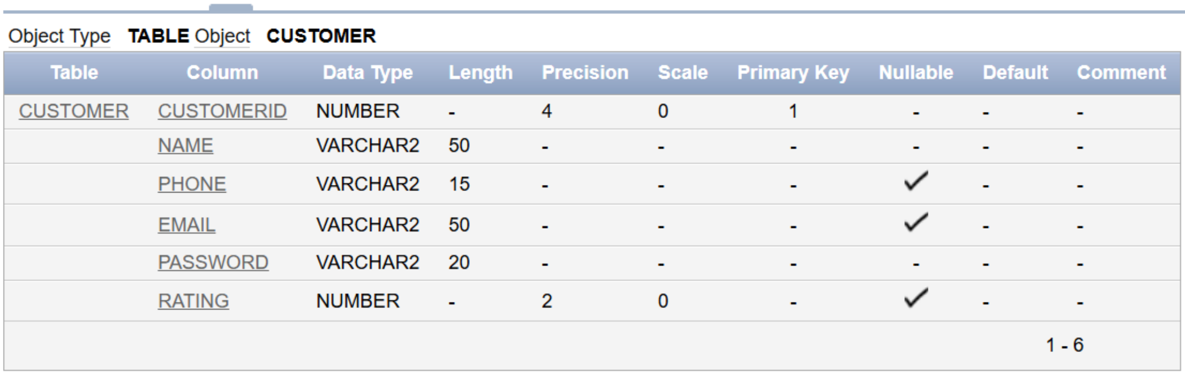
9) Complaint Table

Attribute	Data Type	Size	Constraints
Complaint ID	Number	4	Primary key
Ride ID	Number	4	N/A
Customer ID	Number	4	N/A
Driver ID	Number	4	N/A
Complaint Text	Varchar2	200	Not Null
Date Filed	Date		Default Sysdate
Status	Varchar2	15	Check(Status IN(pending’,’Resolved’,’Rejected’)) Default ‘pending’,Foreign key (Ride ID) References Ride (Ride ID), Foreign Key(Customer ID) References Customer(Customer ID), Foreign key(Driver ID) References Driver(Driver ID));

TABLES CREATION USING SQL

1) Create Customer Table

```
CREATE TABLE CUSTOMER(
CUSTOMERID NUMBER(4) PRIMARY KEY,
NAME VARCHAR2(50) NOT NULL,
PHONE VARCHAR2(15) UNIQUE,
EMAIL VARCHAR2(50) UNIQUE,
PASSWORD VARCHAR2(20) NOT NULL
);
```



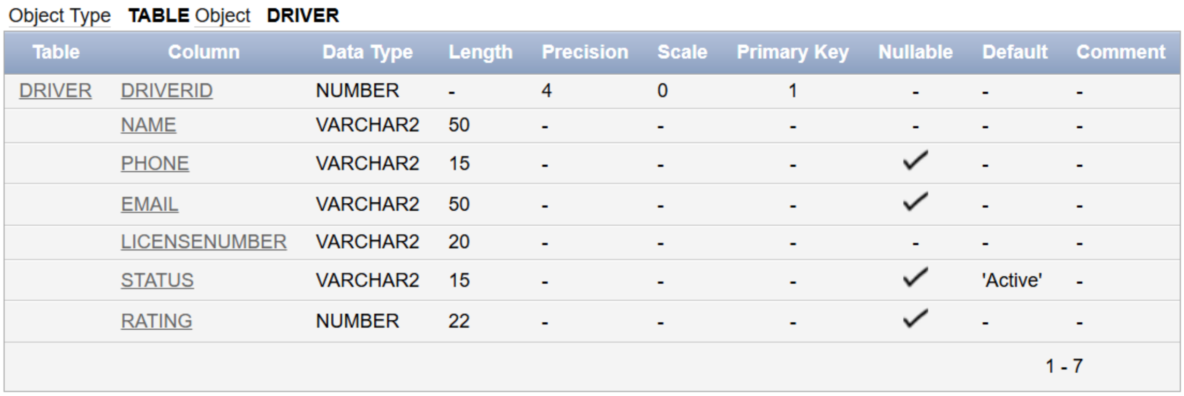
Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMERID	NUMBER	-	4	0	1	-	-	-
	NAME	VARCHAR2	50	-	-	-	-	-	-
	PHONE	VARCHAR2	15	-	-	-	✓	-	-
	EMAIL	VARCHAR2	50	-	-	-	✓	-	-
	PASSWORD	VARCHAR2	20	-	-	-	-	-	-
	RATING	NUMBER	-	2	0	-	✓	-	-

1 - 6

2) Driver Table

```
CREATE TABLE DRIVER(
DRIVERID NUMBER(4) PRIMARY KEY,
NAME VARCHAR2(50) NOT NULL,
PHONE VARCHAR2(15) UNIQUE,
EMAIL VARCHAR2(50) UNIQUE,
LICENSENUMBER VARCHAR2(20) NOT NULL,
STATUS VARCHAR2(15) CHECK(STATUS IN('ACTIVE','INACTIVE','ON RIDE')) DEFAULT
'ACTIVE',RATING NUMBER(2) CHECK(RATING BETWEEN 0 AND 5)
);
```



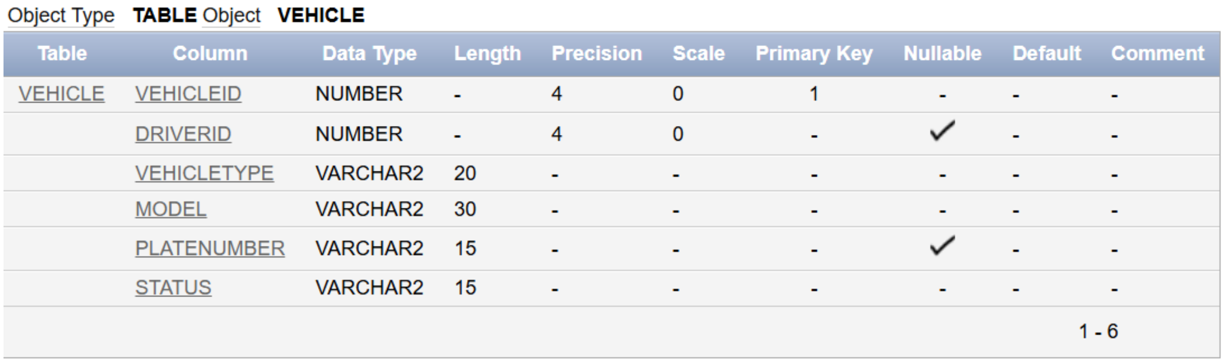
Object Type **TABLE** Object **DRIVER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER	DRIVERID	NUMBER	-	4	0	1	-	-	-
	NAME	VARCHAR2	50	-	-	-	-	-	-
	PHONE	VARCHAR2	15	-	-	-	✓	-	-
	EMAIL	VARCHAR2	50	-	-	-	✓	-	-
	LICENSENUMBER	VARCHAR2	20	-	-	-	-	-	-
	STATUS	VARCHAR2	15	-	-	-	✓	'Active'	-
	RATING	NUMBER	22	-	-	-	✓	-	-

1 - 7

3) Vehicle Table

```
CREATE TABLE VEHICLE
(VEHICLEID NUMBER(4) PRIMARY KEY,
DRIVERID NUMBER(4),
VEHICLETYPE VARCHAR2(20) NOT NULL,
MODEL VARCHAR2(30) NOT NULL,
PLATENUMBER VARCHAR2(15) UNIQUE,STATUS VARCHAR2(15) NOT NULL,
FOREIGN KEY(DRIVERID) REFERENCES DRIVER(DRIVERID)
);
```



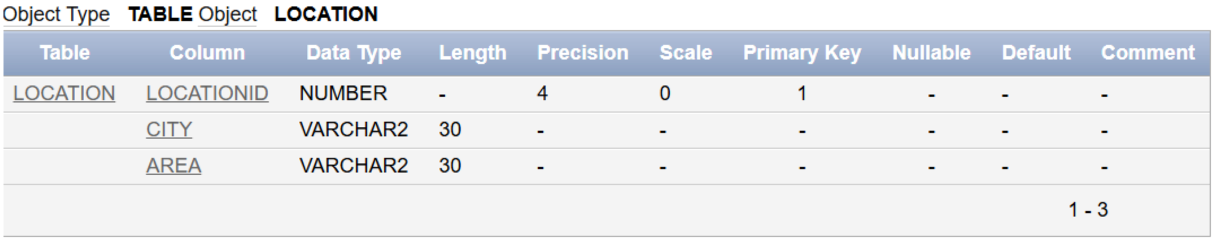
Object Type **TABLE** Object **VEHICLE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VEHICLE	VEHICLEID	NUMBER	-	4	0	1	-	-	-
	DRIVERID	NUMBER	-	4	0	-	✓	-	-
	VEHICLETYPE	VARCHAR2	20	-	-	-	-	-	-
	MODEL	VARCHAR2	30	-	-	-	-	-	-
	PLATENUMBER	VARCHAR2	15	-	-	-	✓	-	-
	STATUS	VARCHAR2	15	-	-	-	-	-	-

1 - 6

4) Location Table

```
CREATE TABLE LOCATION(
LOCATIONID NUMBER(4) PRIMARY KEY,
CITY VARCHAR2(30) NOT NULL,
AREA VARCHAR2(30) NOT NULL
);
```



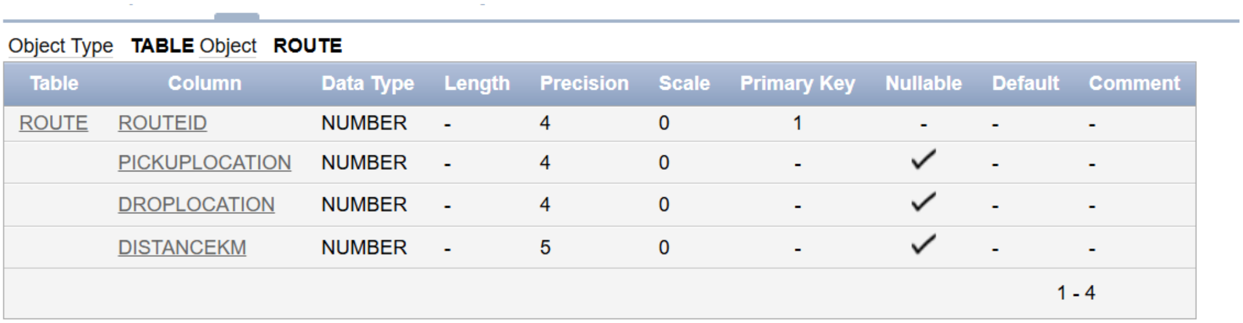
Object Type **TABLE** Object **LOCATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOCATION	LOCATIONID	NUMBER	-	4	0	1	-	-	-
	CITY	VARCHAR2	30	-	-	-	-	-	-
	AREA	VARCHAR2	30	-	-	-	-	-	-

1 - 3

5) Route Table

```
CREATE TABLE ROUTE(
ROUTEID NUMBER(4) PRIMARY KEY,
PICKUPLOCATION NUMBER(4),
DROPLOCATION NUMBER(4),
DISTANCEKM NUMBER(5) CHECK(DISTANCEKM>0),
FOREIGN KEY(PICKUPLOCATION) REFERENCES LOCATION(LOCATIONID),FOREIGN
KEY(DROPLOCATION) REFERENCES LOCATION(LOCATIONID)
);
```



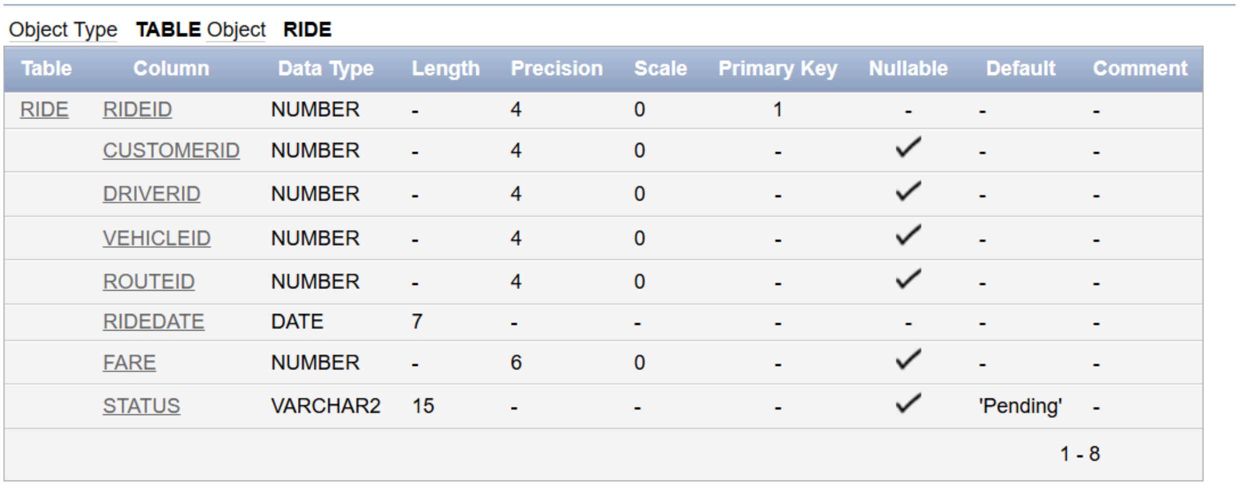
Object Type **TABLE** Object **ROUTE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROUTE	ROUTEID	NUMBER	-	4	0	1	-	-	-
	PICKUPLOCATION	NUMBER	-	4	0	-	✓	-	-
	DROPLOCATION	NUMBER	-	4	0	-	✓	-	-
	DISTANCEKM	NUMBER	-	5	0	-	✓	-	-

1 - 4

6) Ride Table

```
CREATE TABLE RIDE
(RIDEID NUMBER(4) PRIMARY KEY,
CUSTOMERID NUMBER(4),
DRIVERID NUMBER(4),
VEHICLEID NUMBER(4),
ROUTEID NUMBER(4),
RIDEDATE DATE NOT NULL,
FARE NUMBER(6) CHECK(FARE>0),
STATUS VARCHAR2(15) CHECK(STATUS IN('PENDING','COMPLETED','CANCELLED')) DEFAULT
'PENDING',
FOREIGN KEY(CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID),FOREIGN
KEY(DRIVERID) REFERENCES DRIVER(DRIVERID),FOREIGN KEY(VEHICLEID) REFERENCES
VEHICLE(VEHICLEID),FOREIGN KEY(ROUTEID) REFERENCES ROUTE(ROUTEID)
);
```



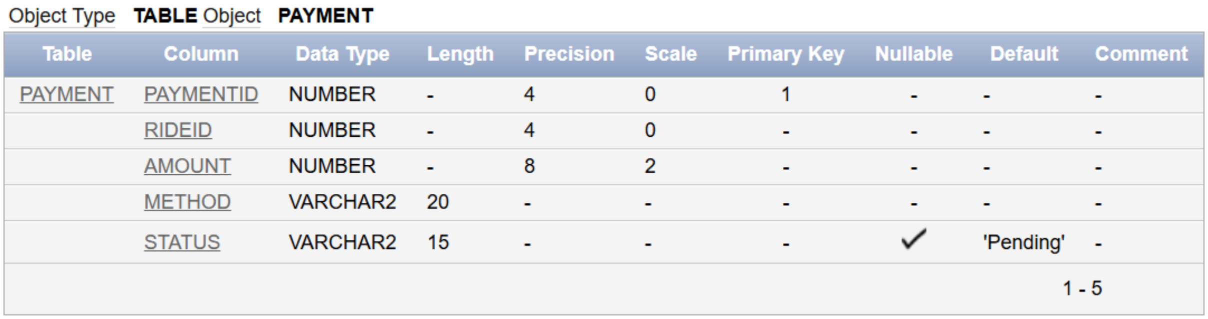
Object Type **TABLE** Object **RIDE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RIDE	RIDEID	NUMBER	-	4	0	1	-	-	-
	CUSTOMERID	NUMBER	-	4	0	-	✓	-	-
	DRIVERID	NUMBER	-	4	0	-	✓	-	-
	VEHICLEID	NUMBER	-	4	0	-	✓	-	-
	ROUTEID	NUMBER	-	4	0	-	✓	-	-
	RIDEDATE	DATE	7	-	-	-	-	-	-
	FARE	NUMBER	-	6	0	-	✓	-	-
	STATUS	VARCHAR2	15	-	-	-	✓	'Pending'	-

1 - 8

7) Payment Table

```
CREATE TABLE PAYMENT(
PAYMENTID NUMBER(4) PRIMARY KEY,
RIDEID NUMBER(4),
AMOUNT NUMBER(6) NOT NULL,
METHOD VARCHAR2(20) CHECK(METHOD IN('CASH','CARD','WALLET')),
STATUS VARCHAR2(15) CHECK(STATUS IN('PAID','PENDING','FAILED')) DEFAULT
'PENDING',FOREIGN KEY(RIDEID) REFERENCES RIDE(RIDEID)
);
```



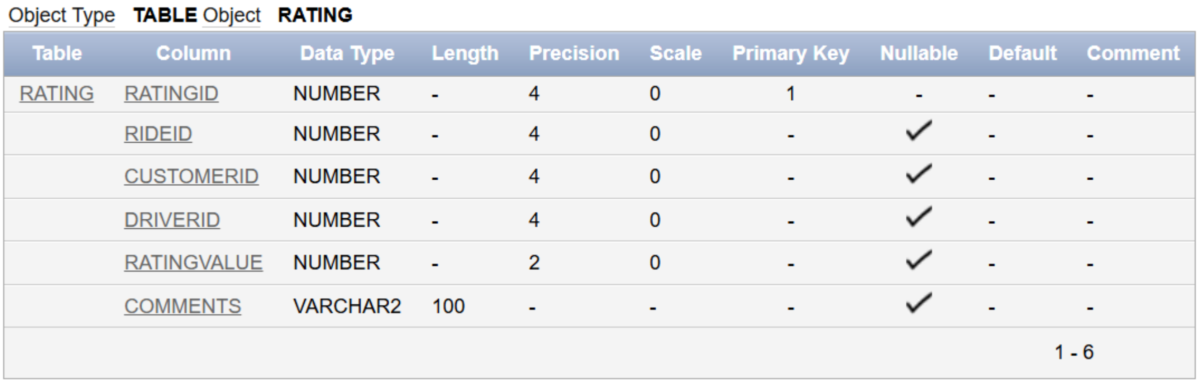
Object Type **TABLE** Object **PAYMENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	PAYMENTID	NUMBER	-	4	0	1	-	-	-
	RIDEID	NUMBER	-	4	0	-	-	-	-
	AMOUNT	NUMBER	-	8	2	-	-	-	-
	METHOD	VARCHAR2	20	-	-	-	-	-	-
	STATUS	VARCHAR2	15	-	-	-	✓	'Pending'	-

1 - 5

8) Rating Table

```
CREATE TABLE RATING(
RATINGID NUMBER(4) PRIMARY KEY,
RIDEID NUMBER(4),
CUSTOMERID NUMBER(4),
DRIVERID NUMBER(4),
RATINGVALUE NUMBER(2) CHECK(RATINGVALUE BETWEEN 1 AND 5),
COMMENT VARCHAR2(100),
FOREIGN KEY(RIDEID) REFERENCES RIDE(RIDEID),FOREIGN KEY(CUSTOMERID) REFERENCES
CUSTOMER(CUSTOMERID),FOREIGN KEY(DRIVERID) REFERENCES DRIVER(DRIVERID)
);
```



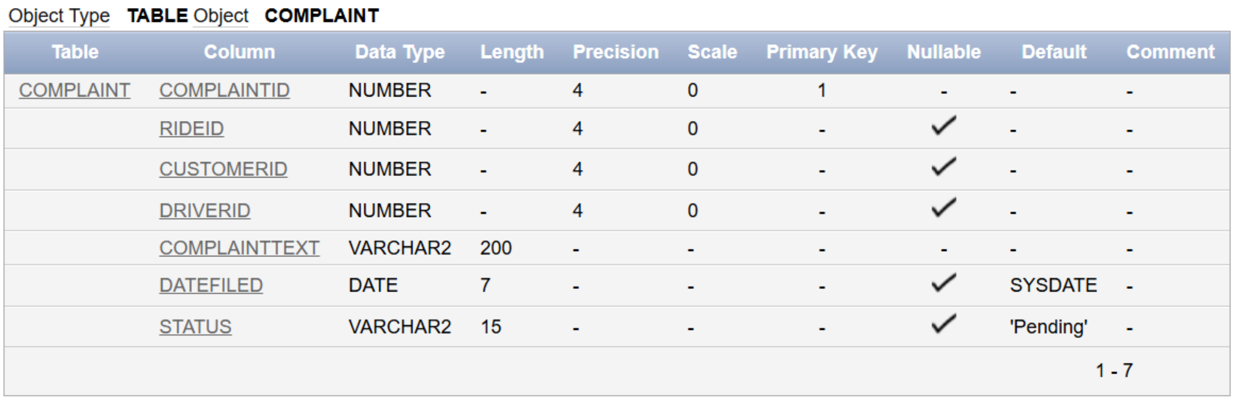
Object Type **TABLE** Object **RATING**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RATING	RATINGID	NUMBER	-	4	0	1	-	-	-
	RIDEID	NUMBER	-	4	0	-	✓	-	-
	CUSTOMERID	NUMBER	-	4	0	-	✓	-	-
	DRIVERID	NUMBER	-	4	0	-	✓	-	-
	RATINGVALUE	NUMBER	-	2	0	-	✓	-	-
	COMMENTS	VARCHAR2	100	-	-	-	✓	-	-

1 - 6

9) Complaint Table

```
CREATE TABLE COMPLAINT(
COMPLAINTID NUMBER(4) PRIMARY KEY,
RIDEID NUMBER(4),
CUSTOMERID NUMBER(4),
DRIVERID NUMBER(4),
COMPLAINTTEXT VARCHAR2(200) NOT NULL,
DATEFILED DATE DEFAULT SYSDATE,
STATUS VARCHAR2(15) CHECK(STATUS IN('PENDING','RESOLVED','REJECTED')) DEFAULT
'PENDING',
FOREIGN KEY(RIDEID) REFERENCES RIDE(RIDEID),FOREIGN KEY(CUSTOMERID) REFERENCES
CUSTOMER(CUSTOMERID),
FOREIGN KEY(DRIVERID) REFERENCES DRIVER(DRIVERID)
);
```



Object Type **TABLE** Object **COMPLAINT**

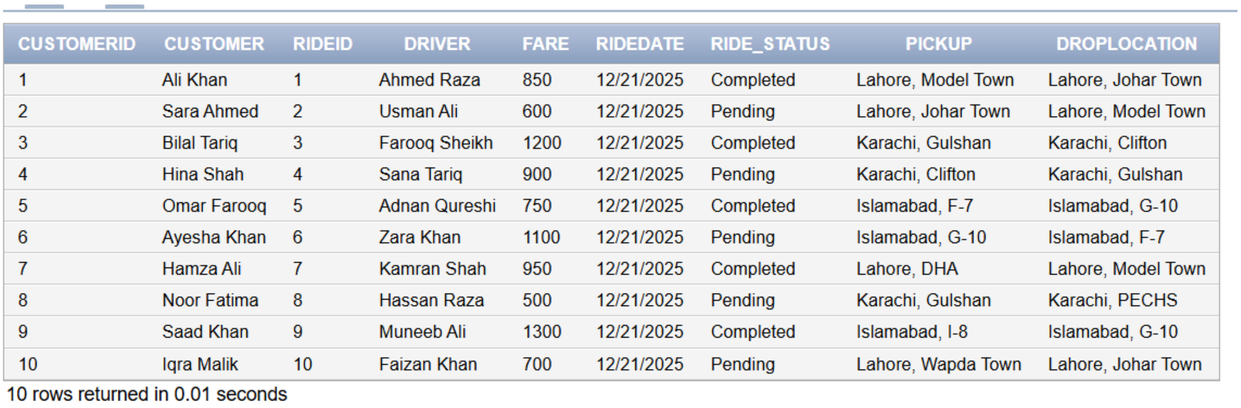
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COMPLAINT	COMPLAINTID	NUMBER	-	4	0	1	-	-	-
	RIDEID	NUMBER	-	4	0	-	✓	-	-
	CUSTOMERID	NUMBER	-	4	0	-	✓	-	-
	DRIVERID	NUMBER	-	4	0	-	✓	-	-
	COMPLAINTTEXT	VARCHAR2	200	-	-	-	-	-	-
	DATEFILED	DATE	7	-	-	-	✓	SYSDATE	-
	STATUS	VARCHAR2	15	-	-	-	✓	'Pending'	-

1 - 7

INSERTION OF DATA IN TABLES

1) Customer Table

```
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (1, 'Ali Khan', '03001234567', 'ali@gmail.com', 'ali123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (2, 'Sara Ahmed', '03007654321', 'sara@gmail.com', 'sara123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (3, 'Bilal Tariq', '03004567890', 'bilal@gmail.com', 'bilal123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (4, 'Hina Shah', '03111223344', 'hina@gmail.com', 'hina123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (5, 'Omar Farooq', '03222333444', 'omar@gmail.com', 'omar123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (6, 'Ayesha Khan', '03333444555', 'ayesha@gmail.com', 'ayesha123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (7, 'Hamza Ali', '03444555666', 'hamza@gmail.com', 'hamza123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (8, 'Noor Fatima', '03555666777', 'noor@gmail.com', 'noor123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (9, 'Saad Khan', '03666777888', 'saad@gmail.com', 'saad123');
INSERT INTO CUSTOMER (CUSTOMERID, NAME, PHONE, EMAIL, PASSWORD) VALUES (10, 'Iqra Malik', '03777888999', 'iqra@gmail.com', 'iqra123');
```

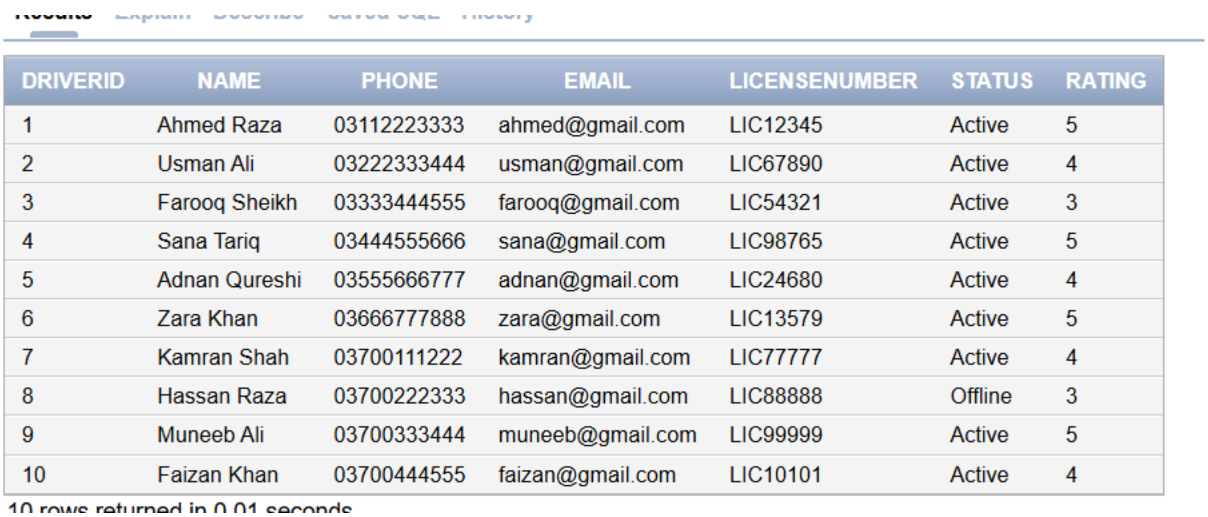


CUSTOMERID	CUSTOMER	RIDEID	DRIVER	FARE	RIDEDATE	RIDE_STATUS	PICKUP	DROPLOCATION
1	Ali Khan	1	Ahmed Raza	850	12/21/2025	Completed	Lahore, Model Town	Lahore, Johar Town
2	Sara Ahmed	2	Usman Ali	600	12/21/2025	Pending	Lahore, Johar Town	Lahore, Model Town
3	Bilal Tariq	3	Farooq Sheikh	1200	12/21/2025	Completed	Karachi, Gulshan	Karachi, Clifton
4	Hina Shah	4	Sana Tariq	900	12/21/2025	Pending	Karachi, Clifton	Karachi, Gulshan
5	Omar Farooq	5	Adnan Qureshi	750	12/21/2025	Completed	Islamabad, F-7	Islamabad, G-10
6	Ayesha Khan	6	Zara Khan	1100	12/21/2025	Pending	Islamabad, G-10	Islamabad, F-7
7	Hamza Ali	7	Kamran Shah	950	12/21/2025	Completed	Lahore, DHA	Lahore, Model Town
8	Noor Fatima	8	Hassan Raza	500	12/21/2025	Pending	Karachi, Gulshan	Karachi, PECHS
9	Saad Khan	9	Muneeb Ali	1300	12/21/2025	Completed	Islamabad, I-8	Islamabad, G-10
10	Iqra Malik	10	Faizan Khan	700	12/21/2025	Pending	Lahore, Wapda Town	Lahore, Johar Town

10 rows returned in 0.01 seconds

2) Driver Table

```
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(1, 'Ahmed Raza', '03112223333', 'ahmed@gmail.com', 'LIC12345', 'ACTIVE', 5);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(2, 'Usman Ali', '03222333444', 'usman@gmail.com', 'LIC67890', 'INACTIVE', 4);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(3, 'Farooq Sheikh', '03333444555', 'farooq@gmail.com', 'LIC54321', 'ACTIVE', 3);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(4, 'Sana Tariq', '03444555666', 'sana@gmail.com', 'LIC98765', 'ACTIVE', 5);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(5, 'Adnan Qureshi', '03555666777', 'adnan@gmail.com', 'LIC24680', 'INACTIVE', 4);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(6, 'Zara Khan', '03666777888', 'zara@gmail.com', 'LIC13579', 'ACTIVE', 5);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(7, 'Kamran Shah', '03777888999', 'kamran@gmail.com', 'LIC11223', 'ACTIVE', 4);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(8, 'Hassan Raza', '03888990000', 'hassan@gmail.com', 'LIC44556', 'ACTIVE', 4);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(9, 'Faizan Khan', '03999001111', 'faizan@gmail.com', 'LIC77889', 'ACTIVE', 5);
INSERT INTO DRIVER (DRIVERID, NAME, PHONE, EMAIL, LICENSENUMBER, STATUS, RATING) VALUES
(10, 'Iqra Malik', '04000112233', 'iqra.driver@gmail.com', 'LIC99000', 'ACTIVE', 5);
```

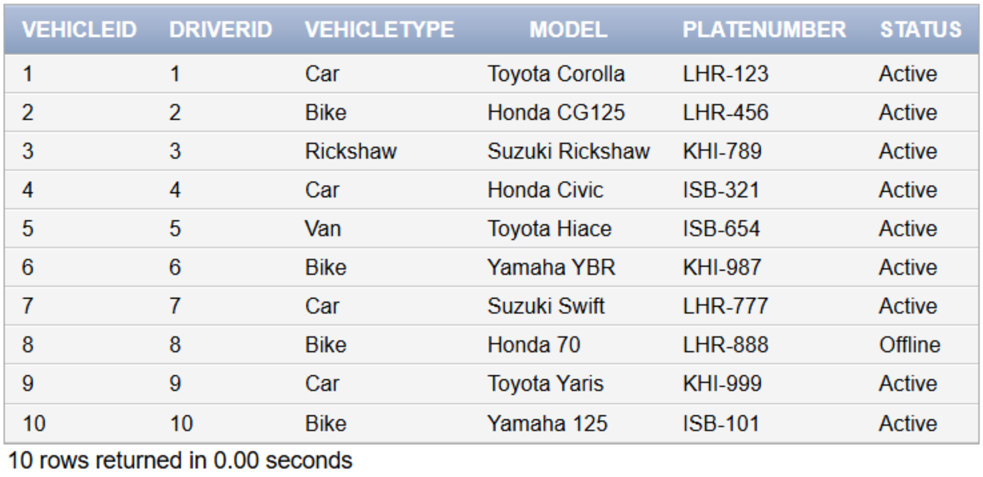


DRIVERID	NAME	PHONE	EMAIL	LICENSENUMBER	STATUS	RATING
1	Ahmed Raza	03112223333	ahmed@gmail.com	LIC12345	Active	5
2	Usman Ali	03222333444	usman@gmail.com	LIC67890	Active	4
3	Farooq Sheikh	03333444555	farooq@gmail.com	LIC54321	Active	3
4	Sana Tariq	03444555666	sana@gmail.com	LIC98765	Active	5
5	Adnan Qureshi	03555666777	adnan@gmail.com	LIC24680	Active	4
6	Zara Khan	03666777888	zara@gmail.com	LIC13579	Active	5
7	Kamran Shah	03700111222	kamran@gmail.com	LIC77777	Active	4
8	Hassan Raza	03700222333	hassan@gmail.com	LIC88888	Offline	3
9	Muneeb Ali	03700333444	muneeb@gmail.com	LIC99999	Active	5
10	Faizan Khan	03700444555	faizan@gmail.com	LIC10101	Active	4

10 rows returned in 0.01 seconds

3) Vehicle Table

```
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (1, 1, 'Car', 'Toyota Corolla', 'LHR-123', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (2, 2, 'Bike', 'Honda CG125', 'LHR-456', 'INACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (3, 3, 'Rickshaw', 'Suzuki Rickshaw', 'KHI-789', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (4, 4, 'Car', 'Honda Civic', 'ISB-321', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (5, 5, 'Van', 'Toyota Hiace', 'ISB-654', 'INACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (6, 6, 'Bike', 'Yamaha YBR', 'KHI-987', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (7, 7, 'Car', 'Suzuki Swift', 'LHR-789', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (8, 8, 'Bike', 'Honda CB125', 'KHI-654', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (9, 9, 'Car', 'Honda Accord', 'ISB-987', 'ACTIVE');
INSERT INTO VEHICLE (VEHICLEID, DRIVERID, VEHICLETYPE, MODEL, PLATENUMBER, STATUS)
VALUES (10, 10, 'Bike', 'Suzuki GS150', 'LHR-321', 'ACTIVE');
```

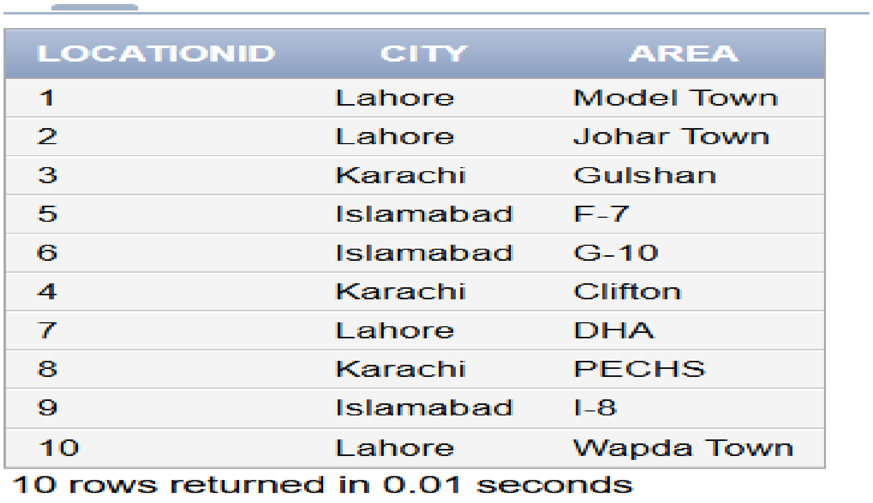


VEHICLEID	DRIVERID	VEHICLETYPE	MODEL	PLATENUMBER	STATUS
1	1	Car	Toyota Corolla	LHR-123	Active
2	2	Bike	Honda CG125	LHR-456	Active
3	3	Rickshaw	Suzuki Rickshaw	KHI-789	Active
4	4	Car	Honda Civic	ISB-321	Active
5	5	Van	Toyota Hiace	ISB-654	Active
6	6	Bike	Yamaha YBR	KHI-987	Active
7	7	Car	Suzuki Swift	LHR-777	Active
8	8	Bike	Honda 70	LHR-888	Offline
9	9	Car	Toyota Yaris	KHI-999	Active
10	10	Bike	Yamaha 125	ISB-101	Active

10 rows returned in 0.00 seconds

4) Location Table

```
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (1, 'Lahore', 'Model Town');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (2, 'Lahore', 'Johar Town');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (3, 'Karachi', 'Gulshan');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (4, 'Karachi', 'Clifton');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (5, 'Islamabad', 'F-7');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (6, 'Islamabad', 'G-10');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (7, 'Lahore', 'DHA');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (8, 'Karachi', 'PECHS');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (9, 'Islamabad', 'I-8');
INSERT INTO LOCATION (LOCATIONID, CITY, AREA) VALUES (10, 'Lahore', 'Wapda Town');
```

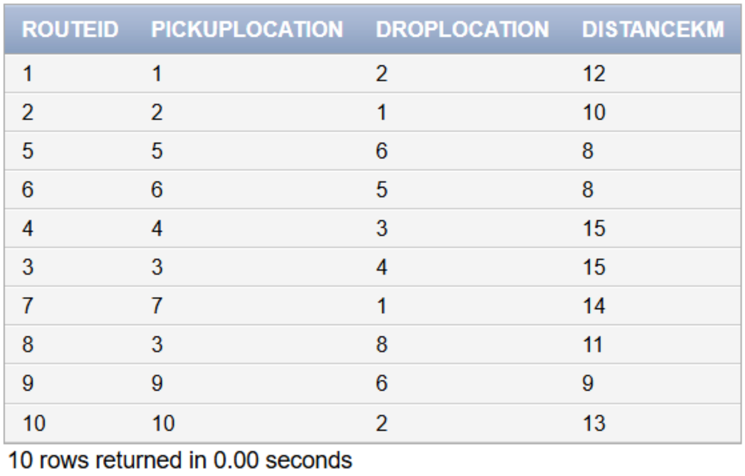


LOCATIONID	CITY	AREA
1	Lahore	Model Town
2	Lahore	Johar Town
3	Karachi	Gulshan
5	Islamabad	F-7
6	Islamabad	G-10
4	Karachi	Clifton
7	Lahore	DHA
8	Karachi	PECHS
9	Islamabad	I-8
10	Lahore	Wapda Town

10 rows returned in 0.01 seconds

5) Route Table

```
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (1, 1, 2, 12);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (2, 2, 1, 10);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (3, 3, 4, 15);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (4, 4, 3, 15);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (5, 5, 6, 8);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (6, 6, 5, 8);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (7, 7, 2, 10);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (8, 8, 3, 12);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (9, 9, 6, 15);
INSERT INTO ROUTE (ROUTEID, PICKUPLOCATION, DROPLOCATION, DISTANCEKM) VALUES (10, 10, 2, 10);
```

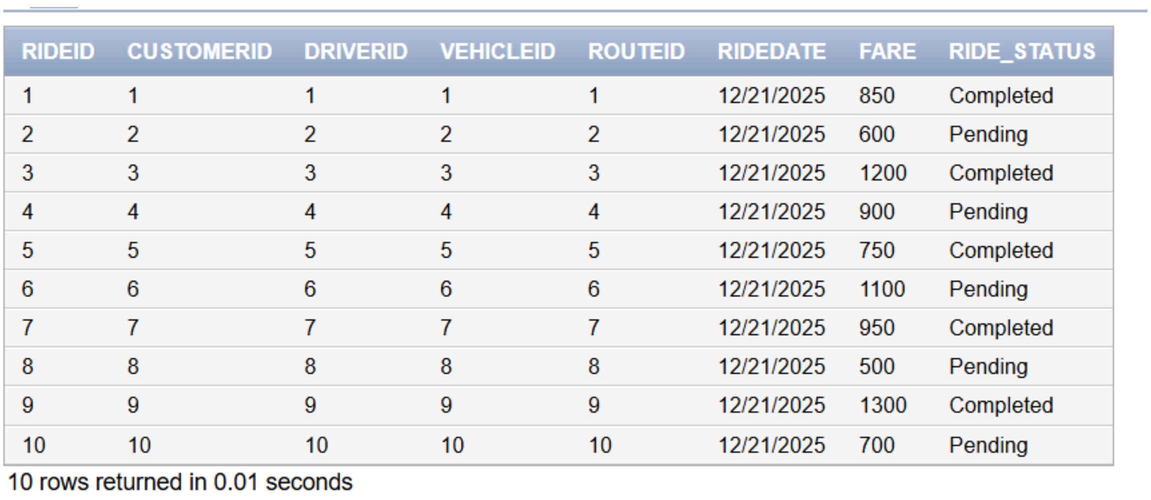


ROUTEID	PICKUPLOCATION	DROPLOCATION	DISTANCEKM
1	1	2	12
2	2	1	10
5	5	6	8
6	6	5	8
4	4	3	15
3	3	4	15
7	7	1	14
8	3	8	11
9	9	6	9
10	10	2	13

10 rows returned in 0.00 seconds

6) Ride Table

```
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (1, 1, 1, 1, 1, TO_DATE('2025-12-21','YYYY-MM-DD'), 850, 'Completed');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (2, 2, 2, 2, 2, TO_DATE('2025-12-21','YYYY-MM-DD'), 600, 'Pending');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (3, 3, 3, 3, 3, TO_DATE('2025-12-21','YYYY-MM-DD'), 1200, 'Completed');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (4, 4, 4, 4, 4, TO_DATE('2025-12-21','YYYY-MM-DD'), 900, 'Pending');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (5, 5, 5, 5, 5, TO_DATE('2025-12-21','YYYY-MM-DD'), 750, 'Completed');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (6, 6, 6, 6, 6, TO_DATE('2025-12-21','YYYY-MM-DD'), 1100, 'Pending');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (7, 7, 7, 7, 7, TO_DATE('2025-12-21','YYYY-MM-DD'), 950, 'Completed');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (8, 8, 8, 8, 8, TO_DATE('2025-12-21','YYYY-MM-DD'), 500, 'Pending');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (9, 9, 9, 9, 9, TO_DATE('2025-12-21','YYYY-MM-DD'), 1300, 'Completed');
INSERT INTO RIDE (RIDEID, CUSTOMERID, DRIVERID, VEHICLEID, ROUTEID, RIDEDATE, FARE,
STATUS) VALUES (10, 10, 10, 10, 10, TO_DATE('2025-12-21','YYYY-MM-DD'), 700, 'Pending');
```

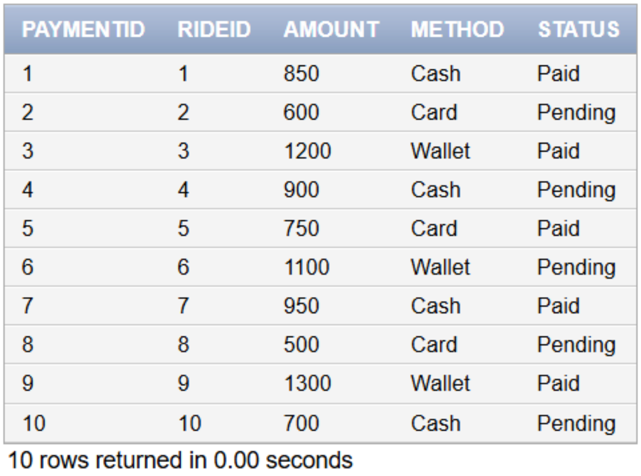


RIDEID	CUSTOMERID	DRIVERID	VEHICLEID	ROUTEID	RIDEDATE	FARE	RIDE_STATUS
1	1	1	1	1	12/21/2025	850	Completed
2	2	2	2	2	12/21/2025	600	Pending
3	3	3	3	3	12/21/2025	1200	Completed
4	4	4	4	4	12/21/2025	900	Pending
5	5	5	5	5	12/21/2025	750	Completed
6	6	6	6	6	12/21/2025	1100	Pending
7	7	7	7	7	12/21/2025	950	Completed
8	8	8	8	8	12/21/2025	500	Pending
9	9	9	9	9	12/21/2025	1300	Completed
10	10	10	10	10	12/21/2025	700	Pending

10 rows returned in 0.01 seconds

7) Payment Table

```
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (1, 1, 850, 'Cash', 'Paid');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (2, 2, 600, 'Card', 'Pending');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (3, 3, 1200, 'Wallet', 'Paid');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (4, 4, 900, 'Cash', 'Pending');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (5, 5, 750, 'Card', 'Paid');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (6, 6, 1100, 'Wallet', 'Pending');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (7, 7, 950, 'Cash', 'Paid');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (8, 8, 500, 'Card', 'Pending');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (9, 9, 1300, 'Wallet', 'Paid');
INSERT INTO PAYMENT (PAYMENTID, RIDEID, AMOUNT, METHOD, STATUS) VALUES (10, 10, 700, 'Cash', 'Pending');
```

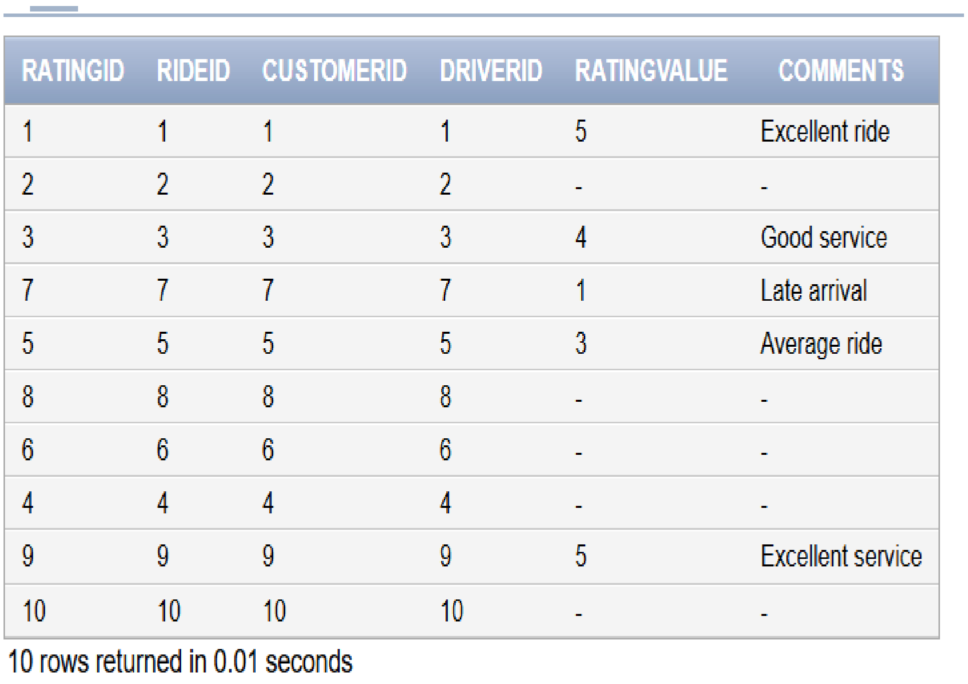


PAYMENTID	RIDEID	AMOUNT	METHOD	STATUS
1	1	850	Cash	Paid
2	2	600	Card	Pending
3	3	1200	Wallet	Paid
4	4	900	Cash	Pending
5	5	750	Card	Paid
6	6	1100	Wallet	Pending
7	7	950	Cash	Paid
8	8	500	Card	Pending
9	9	1300	Wallet	Paid
10	10	700	Cash	Pending

10 rows returned in 0.00 seconds

8) Rating Table

```
INSERT INTO RATING (RATINGID, RIDEID, CUSTOMERID, DRIVERID, RATINGVALUE, COMMENT) VALUES
(1, 1, 1, 1, 5, 'Excellent ride');
INSERT INTO RATING (RATINGID, RIDEID, CUSTOMERID, DRIVERID, RATINGVALUE, COMMENT) VALUES
(2, 3, 3, 3, 4, 'Good service');
INSERT INTO RATING (RATINGID, RIDEID, CUSTOMERID, DRIVERID, RATINGVALUE, COMMENT) VALUES
(3, 5, 5, 5, 3, 'Average ride');
INSERT INTO RATING (RATINGID, RIDEID, CUSTOMERID, DRIVERID, RATINGVALUE, COMMENT) VALUES
(4, 7, 7, 7, 1, 'Late arrival');
INSERT INTO RATING (RATINGID, RIDEID, CUSTOMERID, DRIVERID, RATINGVALUE, COMMENT) VALUES
(5, 9, 9, 9, 5, 'Excellent service');
```

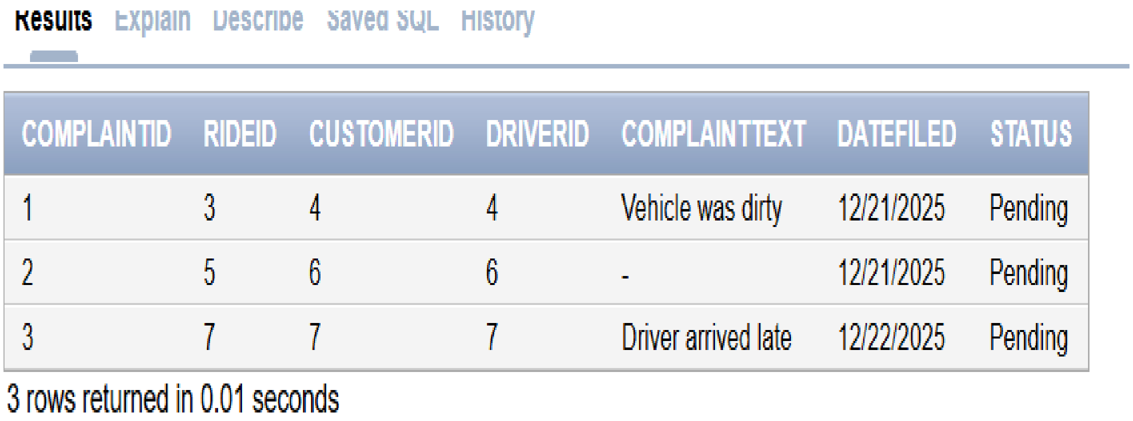


RATINGID	RIDEID	CUSTOMERID	DRIVERID	RATINGVALUE	COMMENTS
1	1	1	1	5	Excellent ride
2	2	2	2	-	-
3	3	3	3	4	Good service
7	7	7	7	1	Late arrival
5	5	5	5	3	Average ride
8	8	8	8	-	-
6	6	6	6	-	-
4	4	4	4	-	-
9	9	9	9	5	Excellent service
10	10	10	10	-	-

10 rows returned in 0.01 seconds

9) Complaint Table

```
INSERT INTO COMPLAINT (COMPLAINTID, RIDEID, CUSTOMERID, DRIVERID, COMPLAINTTEXT,
DATEFILED, STATUS) VALUES (1, 3, 4, 4, 'Vehicle was dirty', TO_DATE('2025-12-21','YYYY-MM-DD'),
'Pending');
INSERT INTO COMPLAINT (COMPLAINTID, RIDEID, CUSTOMERID, DRIVERID, COMPLAINTTEXT,
DATEFILED, STATUS) VALUES (2, 5, 6, 6, '-', TO_DATE('2025-12-21','YYYY-MM-DD'), 'Pending');
INSERT INTO COMPLAINT (COMPLAINTID, RIDEID, CUSTOMERID, DRIVERID, COMPLAINTTEXT,
DATEFILED, STATUS) VALUES (3, 7, 7, 7, 'Driver arrived late', TO_DATE('2025-12-21','YYYY-MM-DD'),
'Pending');
```

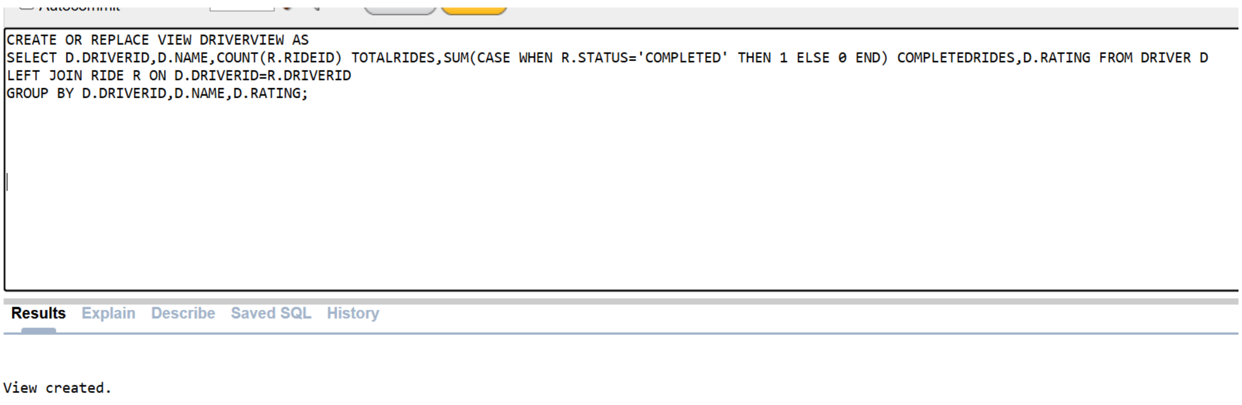


COMPLAINTID	RIDEID	CUSTOMERID	DRIVERID	COMPLAINTTEXT	DATEFILED	STATUS
1	3	4	4	Vehicle was dirty	12/21/2025	Pending
2	5	6	6	-	12/21/2025	Pending
3	7	7	7	Driver arrived late	12/22/2025	Pending

3 rows returned in 0.01 seconds

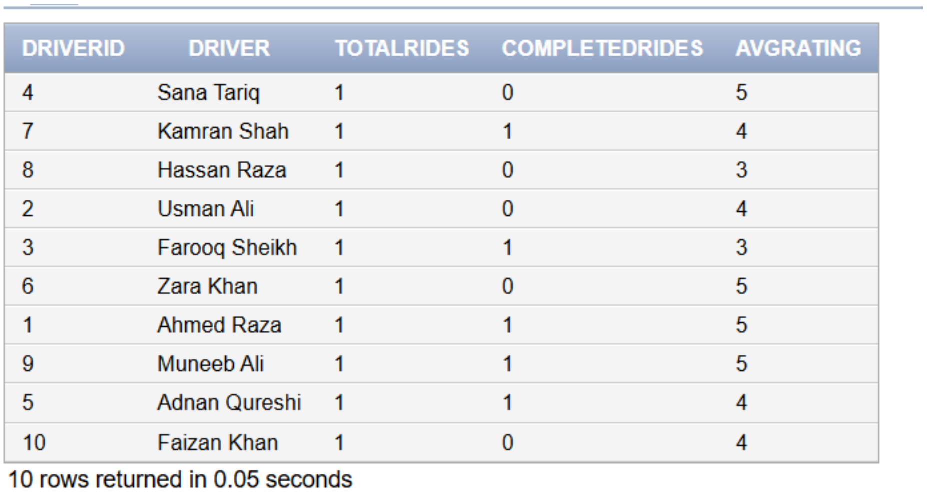
Views Via SQL

1) Driver View



```
CREATE OR REPLACE VIEW DRIVERVIEW AS
SELECT D.DRIVERID,D.NAME,COUNT(R.RIDEID) TOTALRIDES,SUM(CASE WHEN R.STATUS='COMPLETED' THEN 1 ELSE 0 END) COMPLETEDRIDES,D.RATING FROM DRIVER D
LEFT JOIN RIDE R ON D.DRIVERID=R.DRIVERID
GROUP BY D.DRIVERID,D.NAME,D.RATING;
```

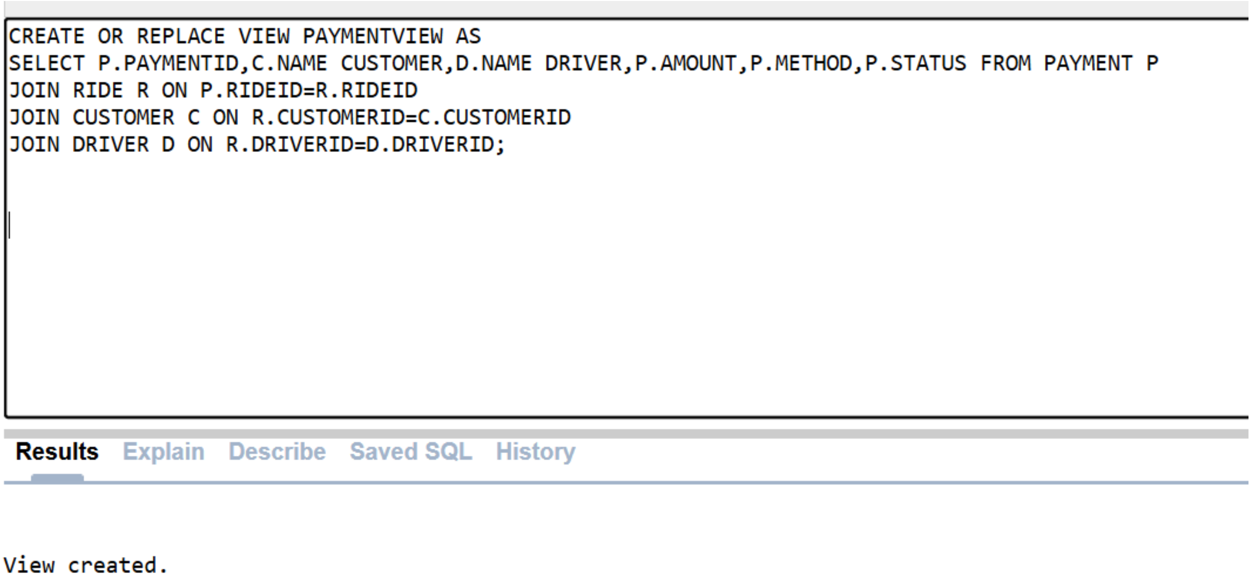
```
View created.
```



DRIVERID	DRIVER	TOTALRIDES	COMPLETEDRIDES	AVGRATING
4	Sana Tariq	1	0	5
7	Kamran Shah	1	1	4
8	Hassan Raza	1	0	3
2	Usman Ali	1	0	4
3	Farooq Sheikh	1	1	3
6	Zara Khan	1	0	5
1	Ahmed Raza	1	1	5
9	Muneeb Ali	1	1	5
5	Adnan Qureshi	1	1	4
10	Faizan Khan	1	0	4

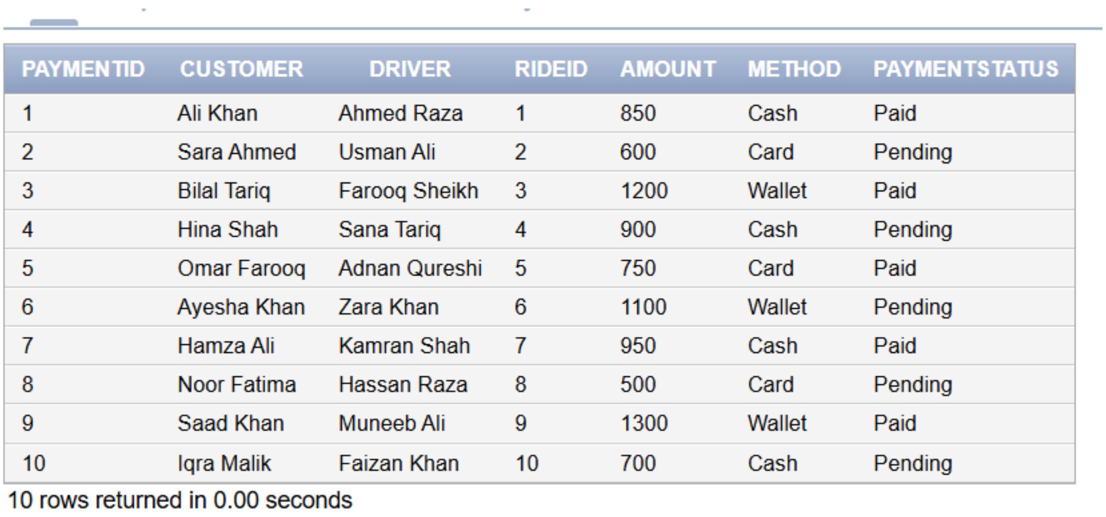
10 rows returned in 0.05 seconds

2) Payment View



```
CREATE OR REPLACE VIEW PAYMENTVIEW AS
SELECT P.PAYMENTID,C.NAME CUSTOMER,D.NAME DRIVER,P.AMOUNT,P.METHOD,P.STATUS FROM PAYMENT P
JOIN RIDE R ON P.RIDEID=R.RIDEID
JOIN CUSTOMER C ON R.CUSTOMERID=C.CUSTOMERID
JOIN DRIVER D ON R.DRIVERID=D.DRIVERID;
```

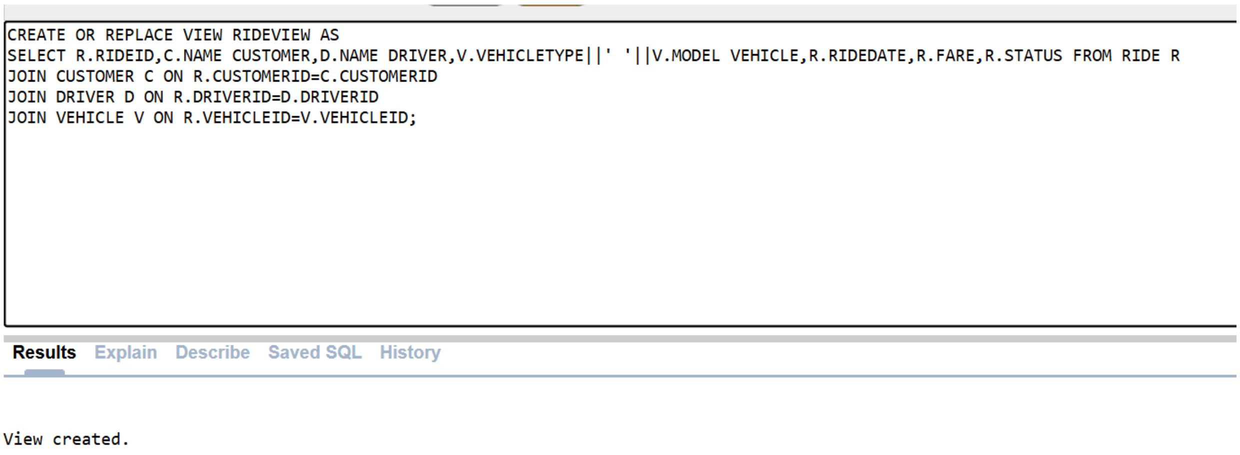
```
View created.
```



PAYMENTID	CUSTOMER	DRIVER	RIDEID	AMOUNT	METHOD	PAYMENTSTATUS
1	Ali Khan	Ahmed Raza	1	850	Cash	Paid
2	Sara Ahmed	Usman Ali	2	600	Card	Pending
3	Bilal Tariq	Farooq Sheikh	3	1200	Wallet	Paid
4	Hina Shah	Sana Tariq	4	900	Cash	Pending
5	Omar Farooq	Adnan Qureshi	5	750	Card	Paid
6	Ayesha Khan	Zara Khan	6	1100	Wallet	Pending
7	Hamza Ali	Kamran Shah	7	950	Cash	Paid
8	Noor Fatima	Hassan Raza	8	500	Card	Pending
9	Saad Khan	Muneeb Ali	9	1300	Wallet	Paid
10	Iqra Malik	Faizan Khan	10	700	Cash	Pending

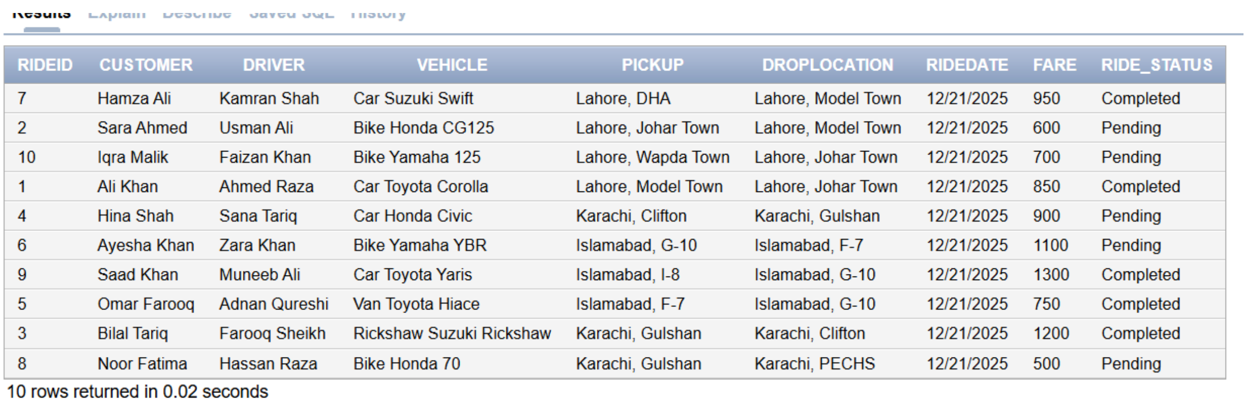
10 rows returned in 0.00 seconds

3) Ride View



```
CREATE OR REPLACE VIEW RIDEVIEW AS
SELECT R.RIDEID,C.NAME CUSTOMER,D.NAME DRIVER,V.VEHICLETYPE||' '||V.MODEL VEHICLE,R.RIDEDATE,R.FARE,R.STATUS FROM RIDE R
JOIN CUSTOMER C ON R.CUSTOMERID=C.CUSTOMERID
JOIN DRIVER D ON R.DRIVERID=D.DRIVERID
JOIN VEHICLE V ON R.VEHICLEID=V.VEHICLEID;
```

```
View created.
```



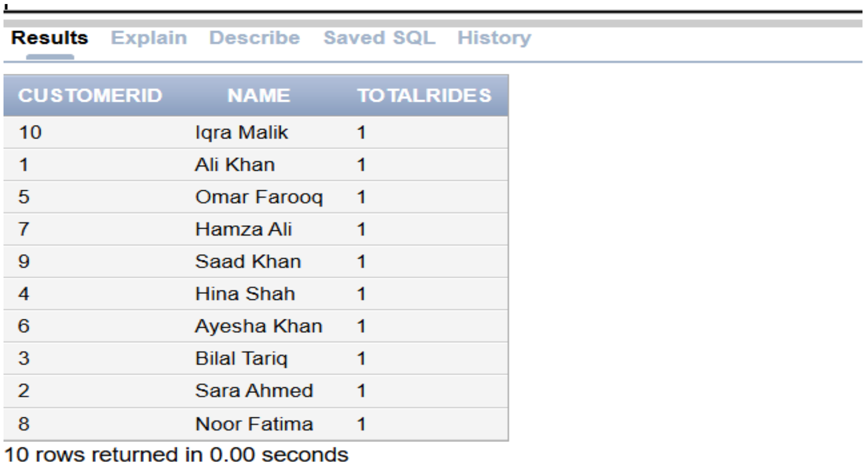
RIDEID	CUSTOMER	DRIVER	VEHICLE	PICKUP	DROPLOCATION	RIDEDATE	FARE	RIDE_STATUS
7	Hamza Ali	Kamran Shah	Car Suzuki Swift	Lahore, DHA	Lahore, Model Town	12/21/2025	950	Completed
2	Sara Ahmed	Usman Ali	Bike Honda CG125	Lahore, Johar Town	Lahore, Model Town	12/21/2025	600	Pending
10	Iqra Malik	Faizan Khan	Bike Yamaha 125	Lahore, Wapda Town	Lahore, Johar Town	12/21/2025	700	Pending
1	Ali Khan	Ahmed Raza	Car Toyota Corolla	Lahore, Model Town	Lahore, Johar Town	12/21/2025	850	Completed
4	Hina Shah	Sana Tariq	Car Honda Civic	Karachi, Clifton	Karachi, Gulshan	12/21/2025	900	Pending
6	Ayesha Khan	Zara Khan	Bike Yamaha YBR	Islamabad, G-10	Islamabad, F-7	12/21/2025	1100	Pending
9	Saad Khan	Muneeb Ali	Car Toyota Yaris	Islamabad, I-8	Islamabad, G-10	12/21/2025	1300	Completed
5	Omar Farooq	Adnan Qureshi	Van Toyota Hiace	Islamabad, F-7	Islamabad, G-10	12/21/2025	750	Completed
3	Bilal Tariq	Farooq Sheikh	Rickshaw Suzuki Rickshaw	Karachi, Gulshan	Karachi, Clifton	12/21/2025	1200	Completed
8	Noor Fatima	Hassan Raza	Bike Honda 70	Karachi, Gulshan	Karachi, PECHS	12/21/2025	500	Pending

10 rows returned in 0.02 seconds

SELECT STATEMENTS FOR COMMON REPORTS

1) Total Rides Per Customer

SELECT C.CUSTOMERID,C.NAME,COUNT(R.RIDEID) TOTALRIDES FROM CUSTOMER C JOIN RIDE R ON C.CUSTOMERID=R.CUSTOMERID GROUP BY C.CUSTOMERID,C.NAME ORDER BY TOTALRIDES DESC;

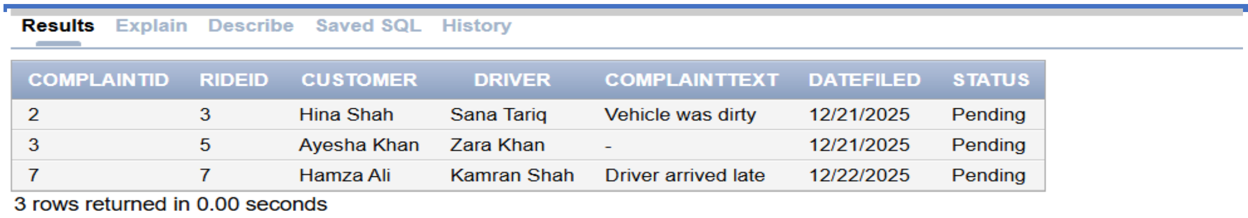


CUSTOMERID	NAME	TOTALRIDES
10	Iqra Malik	1
1	Ali Khan	1
5	Omar Farooq	1
7	Hamza Ali	1
9	Saad Khan	1
4	Hina Shah	1
6	Ayesha Khan	1
3	Bilal Tariq	1
2	Sara Ahmed	1
8	Noor Fatima	1

10 rows returned in 0.00 seconds

2) All Complaints Report

SELECT C.COMPLAINTID,R.RIDEID,CU.NAME CUSTOMER,D.NAME DRIVER,C.COMPLAINTTEXT,C.DATEFILED,C.STATUS FROM COMPLAINT C JOIN CUSTOMER CU ON C.CUSTOMERID=CU.CUSTOMERID JOIN DRIVER D ON C.DRIVERID=D.DRIVERID JOIN RIDE R ON C.RIDEID=R.RIDEID;

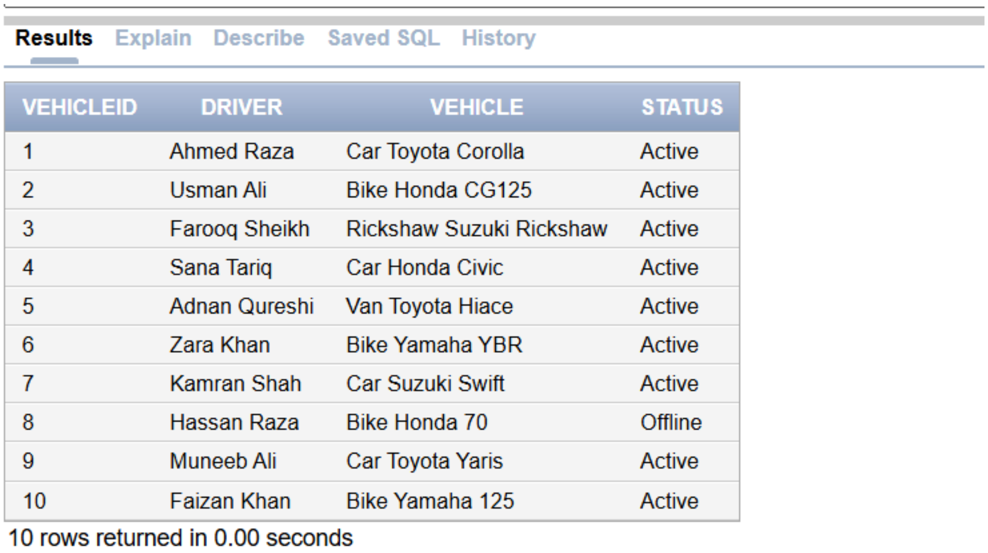


COMPLAINTID	RIDEID	CUSTOMER	DRIVER	COMPLAINTTEXT	DATEFILED	STATUS
2	3	Hina Shah	Sana Tariq	Vehicle was dirty	12/21/2025	Pending
3	5	Ayesha Khan	Zara Khan	-	12/21/2025	Pending
7	7	Hamza Ali	Kamran Shah	Driver arrived late	12/22/2025	Pending

3 rows returned in 0.00 seconds

3) Vehicle Status Report

```
SELECT V.VEHICLEID,D.NAME DRIVER,V.VEHICLETYPE||' '||V.MODEL VEHICLE,V.STATUS FROM
VEHICLE V JOIN DRIVER D ON V.DRIVERID=D.DRIVERID;
```

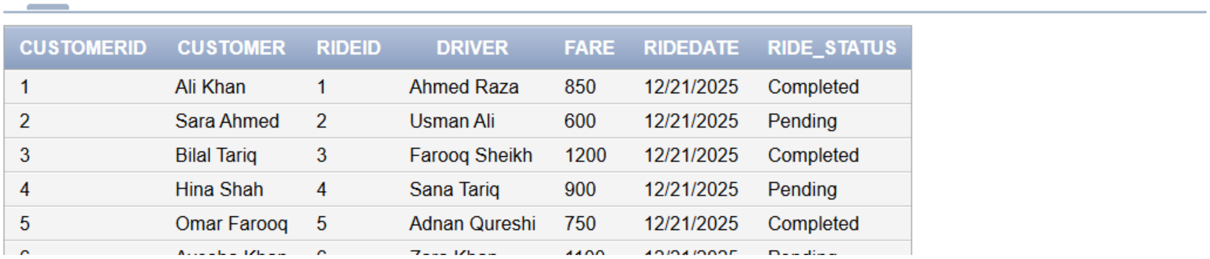


VEHICLEID	DRIVER	VEHICLE	STATUS
1	Ahmed Raza	Car Toyota Corolla	Active
2	Usman Ali	Bike Honda CG125	Active
3	Farooq Sheikh	Rickshaw Suzuki Rickshaw	Active
4	Sana Tariq	Car Honda Civic	Active
5	Adnan Qureshi	Van Toyota Hiace	Active
6	Zara Khan	Bike Yamaha YBR	Active
7	Kamran Shah	Car Suzuki Swift	Active
8	Hassan Raza	Bike Honda 70	Offline
9	Muneeb Ali	Car Toyota Yaris	Active
10	Faizan Khan	Bike Yamaha 125	Active

10 rows returned in 0.00 seconds

4) Customer Rides Report

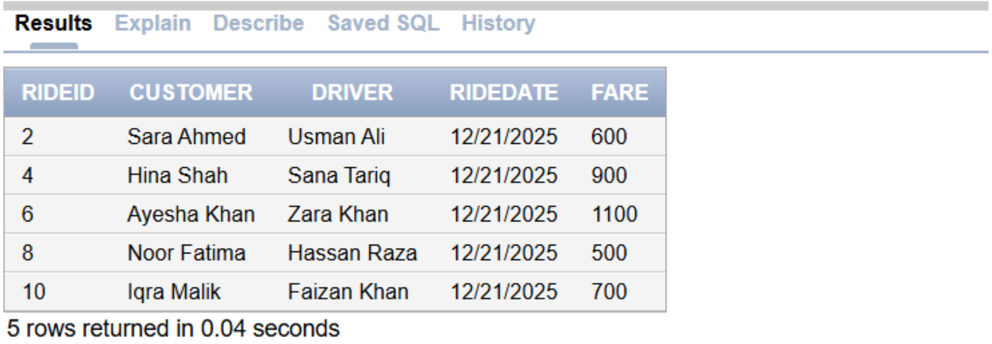
```
SELECT C.CUSTOMERID,C.NAME CUSTOMER,R.RIDEID,D.NAME
DRIVER,R.FARE,R.RIDEDATE,R.STATUS FROM CUSTOMER C JOIN RIDE R ON
C.CUSTOMERID=R.CUSTOMERID JOIN DRIVER D ON R.DRIVERID=D.DRIVERID ORDER BY
C.CUSTOMERID,R.RIDEDATE;
```



CUSTOMERID	CUSTOMER	RIDEID	DRIVER	FARE	RIDEDATE	RIDE_STATUS
1	Ali Khan	1	Ahmed Raza	850	12/21/2025	Completed
2	Sara Ahmed	2	Usman Ali	600	12/21/2025	Pending
3	Bilal Tariq	3	Farooq Sheikh	1200	12/21/2025	Completed
4	Hina Shah	4	Sana Tariq	900	12/21/2025	Pending
5	Omar Farooq	5	Adnan Qureshi	750	12/21/2025	Completed
6	Ayesha Khan	6	Zara Khan	1100	12/21/2025	Pending

5) Pending Rides Report

```
SELECT R.RIDEID,C.NAME CUSTOMER,D.NAME DRIVER,R.RIDEDATE,R.FARE FROM RIDE R JOIN
CUSTOMER C ON R.CUSTOMERID=C.CUSTOMERID JOIN DRIVER D ON
R.DRIVERID=D.DRIVERID WHERE R.STATUS='PENDING';
```

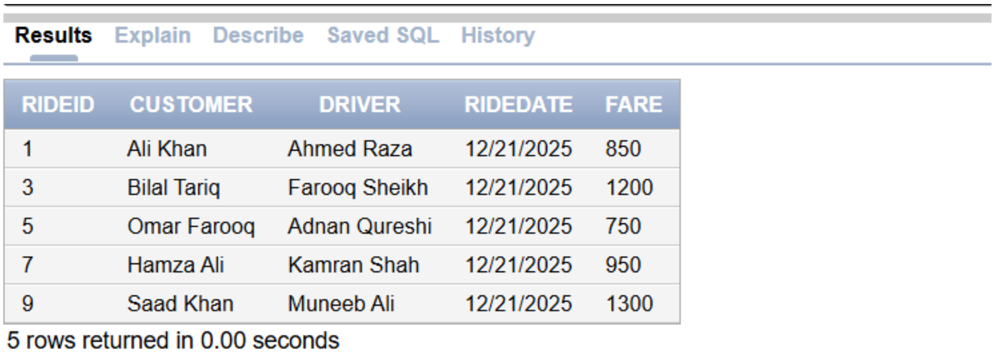


RIDEID	CUSTOMER	DRIVER	RIDEDATE	FARE
2	Sara Ahmed	Usman Ali	12/21/2025	600
4	Hina Shah	Sana Tariq	12/21/2025	900
6	Ayesha Khan	Zara Khan	12/21/2025	1100
8	Noor Fatima	Hassan Raza	12/21/2025	500
10	Iqra Malik	Faizan Khan	12/21/2025	700

5 rows returned in 0.04 seconds

6) Completed Rides Report

```
SELECT R.RIDEID,C.NAME CUSTOMER,D.NAME DRIVER,R.RIDEDATE,R.FARE FROM RIDE R JOIN
CUSTOMER C ON R.CUSTOMERID=C.CUSTOMERID JOIN DRIVER D ON R.DRIVERID=D.DRIVERID
WHERE R.STATUS='COMPLETED';
```

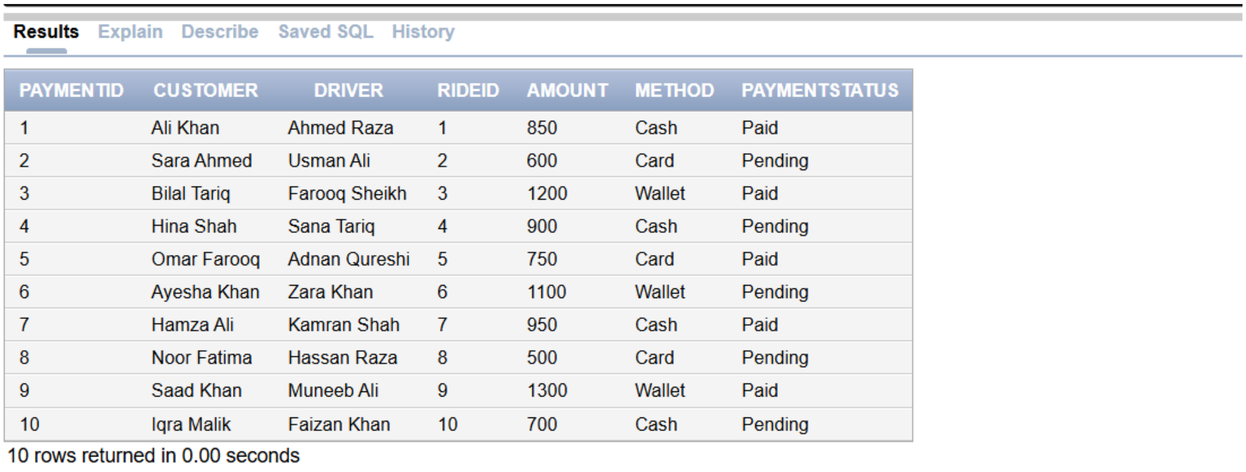


RIDEID	CUSTOMER	DRIVER	RIDEDATE	FARE
1	Ali Khan	Ahmed Raza	12/21/2025	850
3	Bilal Tariq	Farooq Sheikh	12/21/2025	1200
5	Omar Farooq	Adnan Qureshi	12/21/2025	750
7	Hamza Ali	Kamran Shah	12/21/2025	950
9	Saad Khan	Muneeb Ali	12/21/2025	1300

5 rows returned in 0.00 seconds

7) Payment details Report

```
SELECT P.PAYMENTID,C.NAME CUSTOMER,D.NAME
DRIVER,R.RIDEID,P.AMOUNT,P.METHOD,P.STATUS PAYMENTSTATUS FROM PAYMENT P JOIN
RIDE R ON P.RIDEID=R.RIDEID JOIN CUSTOMER C ON R.CUSTOMERID=C.CUSTOMERID JOIN
DRIVER D ON R.DRIVERID=D.DRIVERID;
```



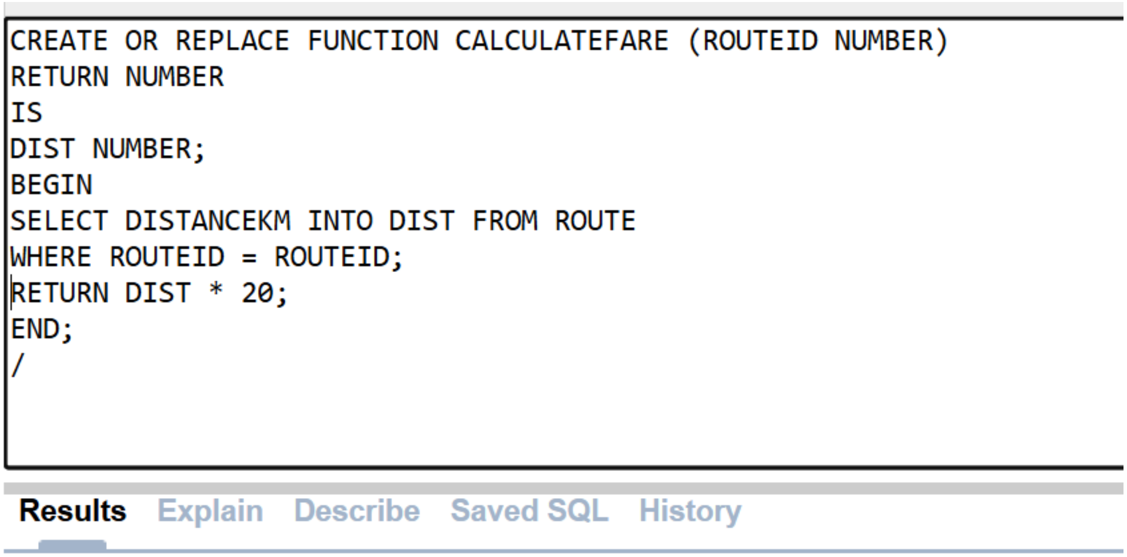
PAYMENTID	CUSTOMER	DRIVER	RIDEID	AMOUNT	METHOD	PAYMENTSTATUS
1	Ali Khan	Ahmed Raza	1	850	Cash	Paid
2	Sara Ahmed	Usman Ali	2	600	Card	Pending
3	Bilal Tariq	Farooq Sheikh	3	1200	Wallet	Paid
4	Hina Shah	Sana Tariq	4	900	Cash	Pending
5	Omar Farooq	Adnan Qureshi	5	750	Card	Paid
6	Ayesha Khan	Zara Khan	6	1100	Wallet	Pending
7	Hamza Ali	Kamran Shah	7	950	Cash	Paid
8	Noor Fatima	Hassan Raza	8	500	Card	Pending
9	Saad Khan	Muneeb Ali	9	1300	Wallet	Paid
10	Iqra Malik	Faizan Khan	10	700	Cash	Pending

10 rows returned in 0.00 seconds

IMPLEMENTATION OF FUNCTIONS

1) Calculate Fare Function

*CREATE OR REPLACE FUNCTION CALCULATEFARE(ROUTEID NUMBER) RETURN NUMBER IS DISTANCE NUMBER;FARE NUMBER;BEGIN SELECT DISTANCEKM INTO DISTANCE FROM ROUTE WHERE ROUTEID=ROUTEID;FARE:=DISTANCE*20;RETURN FARE;END;*

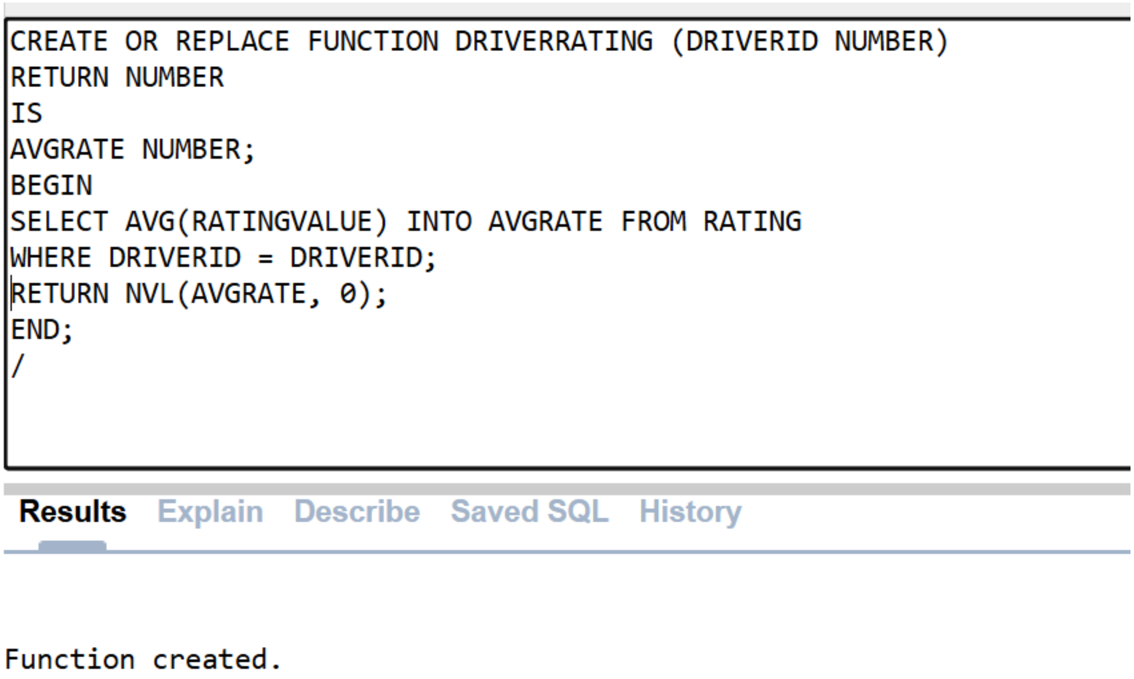


```
CREATE OR REPLACE FUNCTION CALCULATEFARE (ROUTEID NUMBER)
RETURN NUMBER
IS
DIST NUMBER;
BEGIN
SELECT DISTANCEKM INTO DIST FROM ROUTE
WHERE ROUTEID = ROUTEID;
RETURN DIST * 20;
END;
/
```

```
Function created.
```

2) Driver Rating Function

CREATE OR REPLACE FUNCTION DRIVERRATING(DRIVERID NUMBER) RETURN NUMBER IS AVG_RATING NUMBER;BEGIN SELECT AVG(RATINGVALUE) INTO AVG_RATING FROM RATING WHERE DRIVERID=DRIVERID;RETURN NVL(AVG_RATING,0);END;



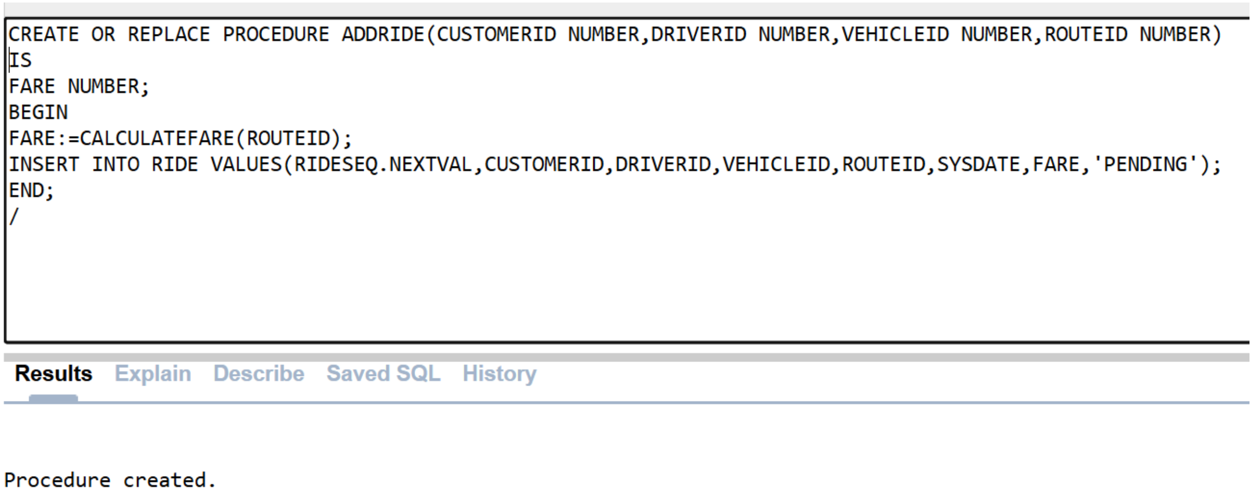
```
CREATE OR REPLACE FUNCTION DRIVERRATING (DRIVERID NUMBER)
RETURN NUMBER
IS
AVGRATE NUMBER;
BEGIN
SELECT AVG(RATINGVALUE) INTO AVGRATE FROM RATING
WHERE DRIVERID = DRIVERID;
RETURN NVL(AVGRATE, 0);
END;
/
```

```
Function created.
```

Procedures Via SQL

1) Add Ride Procedure

```
CREATE OR REPLACE PROCEDURE ADDRIDE(CUSTOMERID NUMBER,DRIVERID
NUMBER,VEHICLEID NUMBER,ROUTEID NUMBER) IS FARE NUMBER;BEGIN
FARE:=CALCULATEFARE(ROUTEID);INSERT INTO
RIDE(RIDEID,CUSTOMERID,DRIVERID,VEHICLEID,ROUTEID,RIDEDATE,FARE,STATUS)
VALUES(RIDES_SEQ.NEXTVAL,CUSTOMERID,DRIVERID,VEHICLEID,ROUTEID,SYSDATE,FARE,'PENDING');END;
```

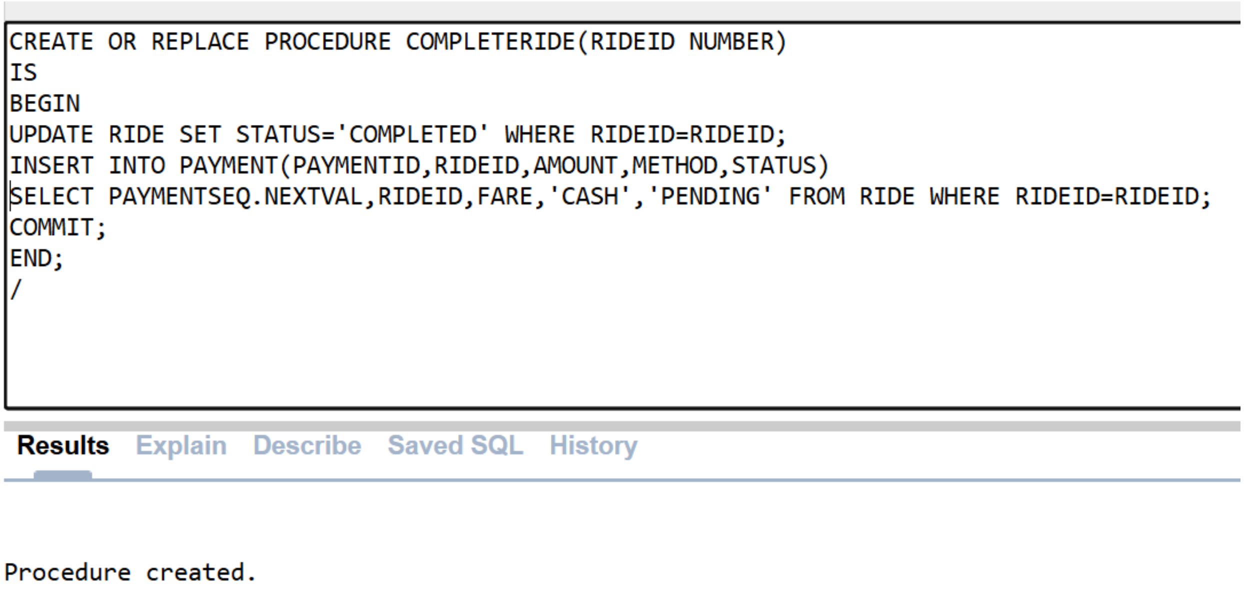


```
CREATE OR REPLACE PROCEDURE ADDRIDE(CUSTOMERID NUMBER,DRIVERID NUMBER,VEHICLEID NUMBER,ROUTEID NUMBER)
IS
FARE NUMBER;
BEGIN
FARE:=CALCULATEFARE(ROUTEID);
INSERT INTO RIDE VALUES(RIDESEQ.NEXTVAL,CUSTOMERID,DRIVERID,VEHICLEID,ROUTEID,SYSDATE,FARE,'PENDING');
END;
/
```

```
Procedure created.
```

2) Complete Ride Procedure

CREATE OR REPLACE PROCEDURE COMPLETERIDE(RIDEID NUMBER) IS BEGIN UPDATE RIDE SET STATUS='COMPLETED' WHERE RIDEID=RIDEID;INSERT INTO PAYMENT(PAYMENTID,RIDEID,AMOUNT,METHOD,STATUS) SELECT PAYMENTSEQ.NEXTVAL,RIDEID,FARE,'CASH','PENDING' FROM RIDE WHERE RIDEID=RIDEID;COMMIT;END;

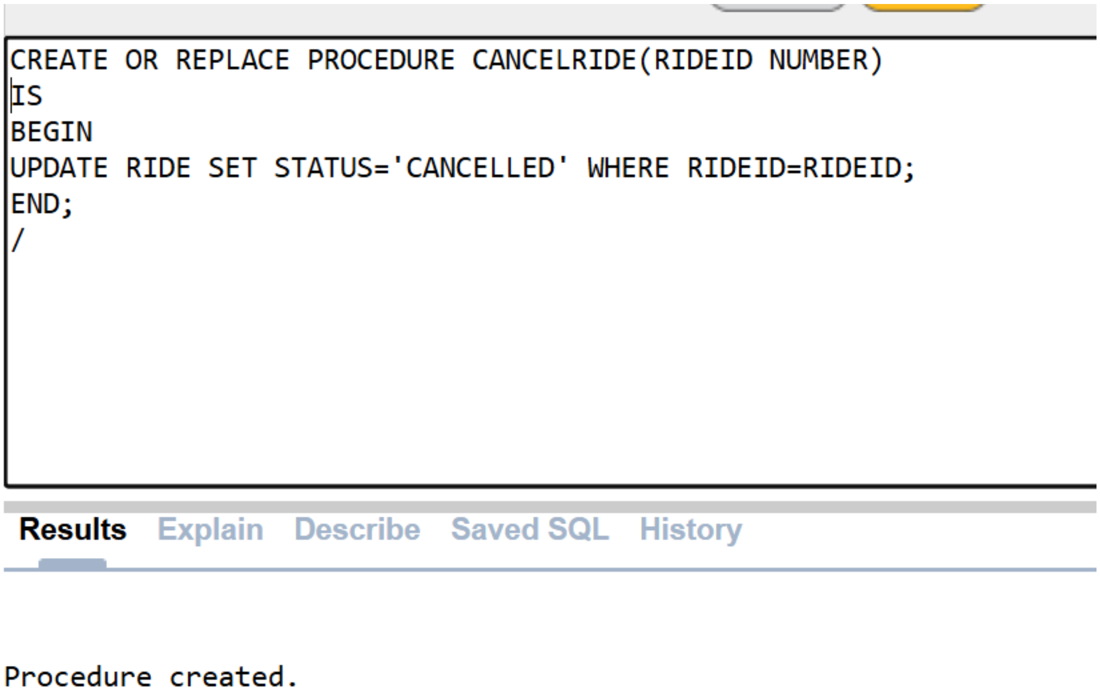


```
CREATE OR REPLACE PROCEDURE COMPLETERIDE(RIDEID NUMBER)
IS
BEGIN
UPDATE RIDE SET STATUS='COMPLETED' WHERE RIDEID=RIDEID;
INSERT INTO PAYMENT(PAYMENTID,RIDEID,AMOUNT,METHOD,STATUS)
SELECT PAYMENTSEQ.NEXTVAL,RIDEID,FARE,'CASH','PENDING' FROM RIDE WHERE RIDEID=RIDEID;
COMMIT;
END;
/
```

```
Procedure created.
```

3) Cancel Ride Procedure

CREATE OR REPLACE PROCEDURE CANCELRIDE(RIDEID NUMBER) IS BEGIN UPDATE RIDE SET STATUS='CANCELLED' WHERE RIDEID=RIDEID;END;

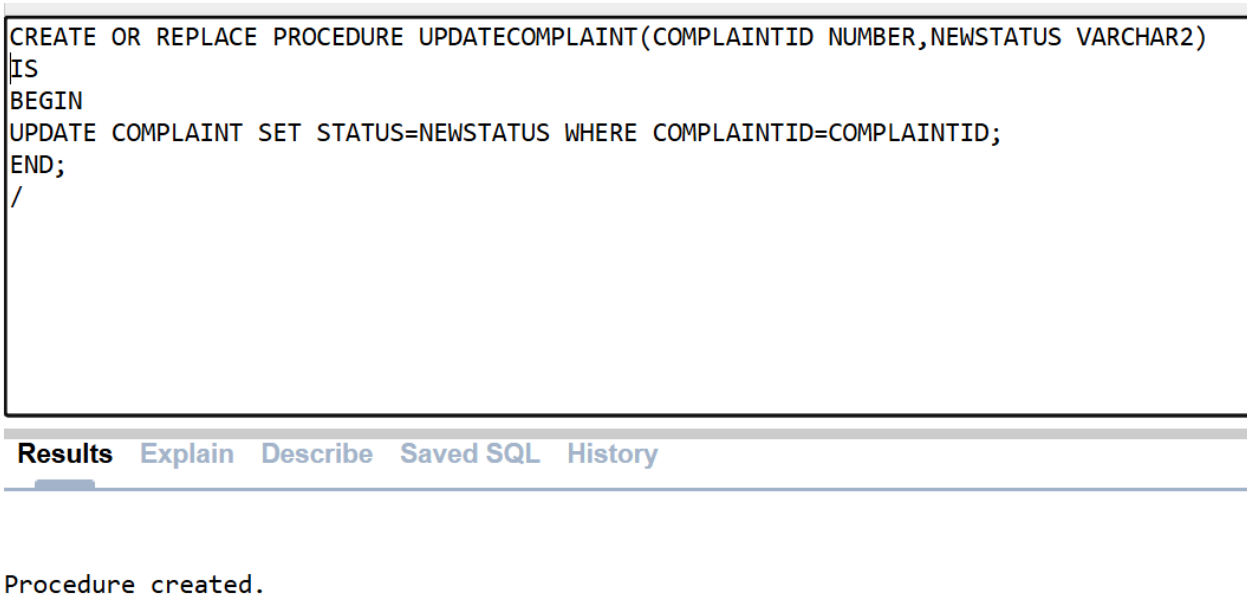


```
CREATE OR REPLACE PROCEDURE CANCELRIDE(RIDEID NUMBER)
IS
BEGIN
UPDATE RIDE SET STATUS='CANCELLED' WHERE RIDEID=RIDEID;
END;
/
```

```
Procedure created.
```

4) Update Complaint Procedure

CREATE OR REPLACE PROCEDURE UPDATECOMPLAINT(COMPLAINTID NUMBER,NEWSTATUS VARCHAR2) IS BEGIN UPDATE COMPLAINT SET STATUS=NEWSTATUS WHERE COMPLAINTID=COMPLAINTID;END;



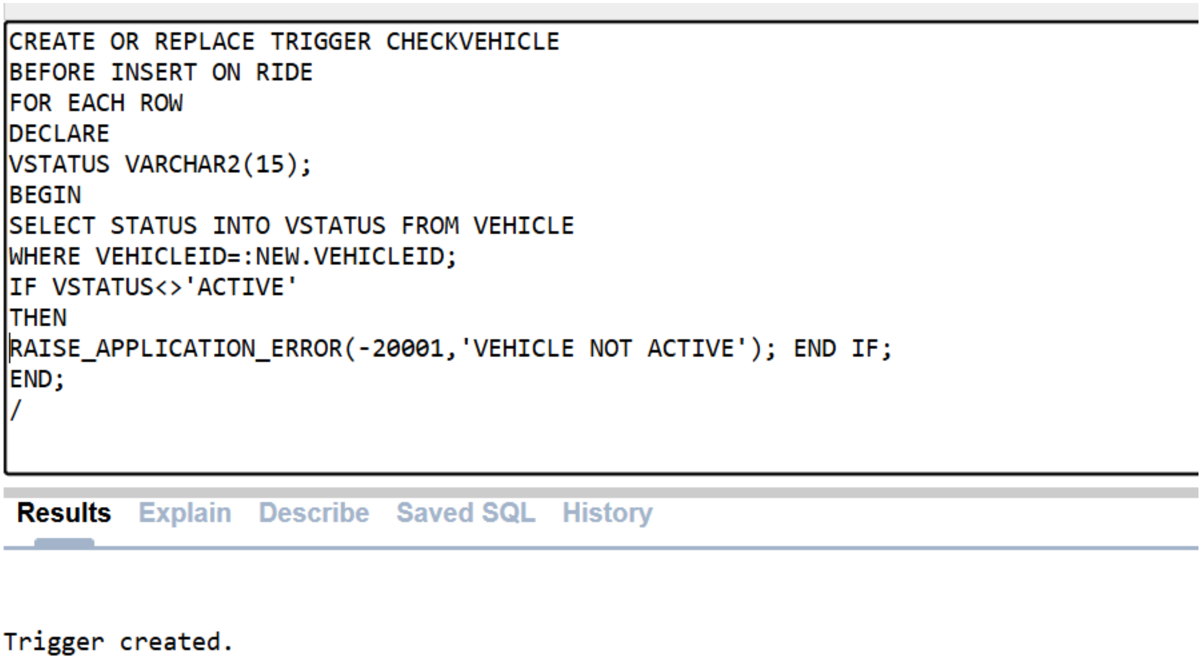
```
CREATE OR REPLACE PROCEDURE UPDATECOMPLAINT(COMPLAINTID NUMBER,NEWSTATUS VARCHAR2)
IS
BEGIN
UPDATE COMPLAINT SET STATUS=NEWSTATUS WHERE COMPLAINTID=COMPLAINTID;
END;
/
```

```
Procedure created.
```

Triggers Via PL SQL

1) Check Vehicle Trigger

CREATE OR REPLACE TRIGGER CHECKVEHICLE BEFORE INSERT ON RIDE FOR EACH ROW DECLARE VSTATUS VARCHAR2(15);BEGIN SELECT STATUS INTO VSTATUS FROM VEHICLE WHERE VEHICLEID=:NEW.VEHICLEID;IF VSTATUS<>'ACTIVE' THEN RAISE_APPLICATION_ERROR(-20001,'VEHICLE NOT ACTIVE');END IF;END;

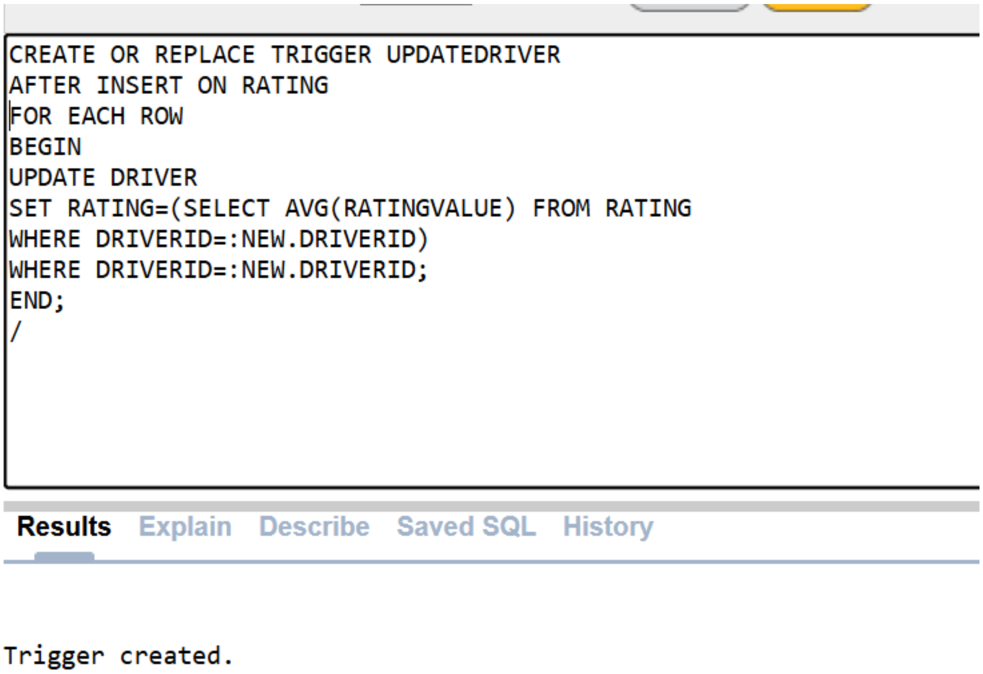


```
CREATE OR REPLACE TRIGGER CHECKVEHICLE
BEFORE INSERT ON RIDE
FOR EACH ROW
DECLARE
VSTATUS VARCHAR2(15);
BEGIN
SELECT STATUS INTO VSTATUS FROM VEHICLE
WHERE VEHICLEID=:NEW.VEHICLEID;
IF VSTATUS<>'ACTIVE'
THEN
RAISE_APPLICATION_ERROR(-20001,'VEHICLE NOT ACTIVE'); END IF;
END;
/
```

```
Trigger created.
```

2) Update Driver Trigger

CREATE OR REPLACE TRIGGER UPDATEDRIVER AFTER INSERT ON RATING FOR EACH ROW BEGIN UPDATE DRIVER SET RATING=(SELECT AVG(RATINGVALUE) FROM RATING WHERE DRIVERID=:NEW.DRIVERID) WHERE DRIVERID=:NEW.DRIVERID;END;

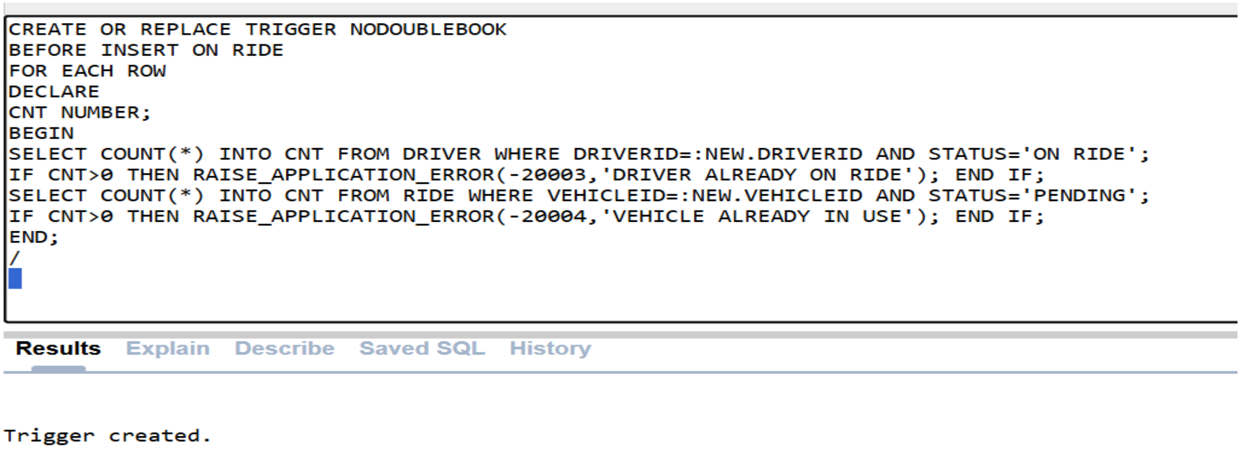


```
CREATE OR REPLACE TRIGGER UPDATEDRIVER
AFTER INSERT ON RATING
FOR EACH ROW
BEGIN
UPDATE DRIVER
SET RATING=(SELECT AVG(RATINGVALUE) FROM RATING
WHERE DRIVERID=:NEW.DRIVERID)
WHERE DRIVERID=:NEW.DRIVERID;
END;
/
```

```
Trigger created.
```

3) No Double Book Trigger

CREATE OR REPLACE TRIGGER NODOUBLEBOOK BEFORE INSERT ON RIDE FOR EACH ROW DECLARE CNT NUMBER;BEGIN SELECT COUNT(*) INTO CNT FROM DRIVER WHERE DRIVERID=:NEW.DRIVERID AND STATUS='ON RIDE';IF CNT>0 THEN RAISE_APPLICATION_ERROR(-20003,'DRIVER ALREADY ON RIDE');END IF;SELECT COUNT(*) INTO CNT FROM RIDE WHERE VEHICLEID=:NEW.VEHICLEID AND STATUS='PENDING';IF CNT>0 THEN RAISE_APPLICATION_ERROR(-20004,'VEHICLE ALREADY IN USE');END IF;END;

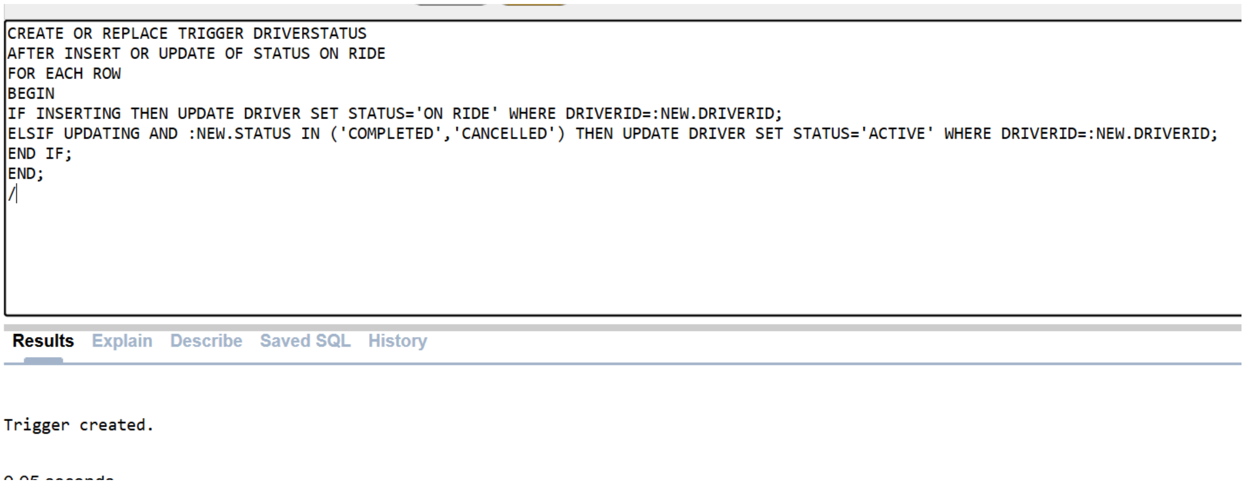


```
CREATE OR REPLACE TRIGGER NODOUBLEBOOK
BEFORE INSERT ON RIDE
FOR EACH ROW
DECLARE
CNT NUMBER;
BEGIN
SELECT COUNT(*) INTO CNT FROM DRIVER WHERE DRIVERID=:NEW.DRIVERID AND STATUS='ON RIDE';
IF CNT>0 THEN RAISE_APPLICATION_ERROR(-20003,'DRIVER ALREADY ON RIDE'); END IF;
SELECT COUNT(*) INTO CNT FROM RIDE WHERE VEHICLEID=:NEW.VEHICLEID AND STATUS='PENDING';
IF CNT>0 THEN RAISE_APPLICATION_ERROR(-20004,'VEHICLE ALREADY IN USE'); END IF;
END;
/
```

```
Trigger created.
```

4) Driver Status Trigger

CREATE OR REPLACE TRIGGER DRIVERSTATUS AFTER INSERT OR UPDATE OF STATUS ON RIDE FOR EACH ROW BEGIN IF INSERTING THEN UPDATE DRIVER SET STATUS='ON RIDE' WHERE DRIVERID=:NEW.DRIVERID;ELSIF UPDATING AND :NEW.STATUS IN('COMPLETED','CANCELLED') THEN UPDATE DRIVER SET STATUS='ACTIVE' WHERE DRIVERID=:NEW.DRIVERID;END IF;END;



```
CREATE OR REPLACE TRIGGER DRIVERSTATUS
AFTER INSERT OR UPDATE OF STATUS ON RIDE
FOR EACH ROW
BEGIN
IF INSERTING THEN UPDATE DRIVER SET STATUS='ON RIDE' WHERE DRIVERID=:NEW.DRIVERID;
ELSIF UPDATING AND :NEW.STATUS IN ('COMPLETED','CANCELLED') THEN UPDATE DRIVER SET STATUS='ACTIVE' WHERE DRIVERID=:NEW.DRIVERID;
END IF;
END;
/
```

```
Trigger created.

0.05 seconds
```