Name : Koki Yamanaka
Title: Assignment 2 PROJECT CASE STUDY
COURSE : COMP 3610
DATE : MARCH 5th 2023

1.0 Brief description of the problem and solution

The problem is to design and implement a database for Fortune Car rentals to keep records of their vehicles, clients, and daily reservations. The database should follow certain rules and regulations, such as clients need to register first to make reservations, all cars are registered in the vehicle table, a client cannot rent a car if they are under 18, etc.
The solution is to create the necessary tables and implement functions, procedures, and triggers to ensure the rules and regulations are followed.

2.0 Code of the procedure

2.1 Definition and creation of the tables

First, we define our structure for each table.
VEHICLE:

```
CREATE TABLE VEHICLE (V_ID INT PRIMARY KEY, V_MODEL VARCHAR(50), V_MAKE VARCHAR(50), COST_PER_DAY DECIMAL(10,2));
```

CLIENT:

```
CREATE TABLE CLIENT (C_ID INT PRIMARY KEY, F_NAME VARCHAR(50), L_NAME VARCHAR(50), DOB DATE, CITY VARCHAR(50), GENDER CHAR(1));
```

RESERVATION:

```
CREATE TABLE RESERVATION (R_ID INT PRIMARY KEY, START_DATE DATE, END_DATE DATE, C_ID INT, V_ID INT, TOTAL_COST DECIMAL(10,2), FOREIGN KEY (C_ID) REFERENCES CLIENT (C_ID), FOREIGN KEY (V_ID) REFERENCES VEHICLE (V_ID));
```

ARCHIVED_RESERVATION:

```
CREATE TABLE ARCHIVED_RESERVATION (R_ID INT PRIMARY KEY, START_DATE DATE, END_DATE DATE, C_ID INT, V_ID INT, TOTAL_COST DECIMAL(10,2), ADDED_BY VARCHAR(50), ADDED_DATE DATE, FOREIGN KEY (C_ID) REFERENCES CLIENT (C_ID), FOREIGN KEY (V_ID) REFERENCES VEHICLE (V_ID));
```

BLACK_LIST:

```
CREATE TABLE BLACK_LIST (C_ID INT PRIMARY KEY, F_NAME VARCHAR(50), L_NAME VARCHAR(50), DOB DATE, CITY VARCHAR(50), GENDER CHAR(1), REASON VARCHAR(100), ADDED_BY VARCHAR(50), ADDED_DATE DATE);
```

STATS:

```
CREATE TABLE STATS (YEAR INT, C_ID INT, AGE INT, GENDER CHAR(1), TOT_RESERVATIONS INT, TOT_DAYS_OF_RESERVATIONS INT, TOT_COST_OF_RESERVATIONS DECIMAL(10,2), PRIMARY KEY (YEAR, C_ID), FOREIGN KEY (C_ID) REFERENCES CLIENT (C_ID));
```

Next we insert sample data into our. We omit pasting the code here, since it's too long.
We placed them at the end of the page. Next, we check our sample data

```
-- check data samples
SELECT * FROM VEHICLE; SELECT * FROM RESERVATION;SELECT * FROM CLIENT;SELECT * FROM ARCHIVED_RESERVATION;SELECT * FROM BLACK_LIST; SELECT * FROM STATS;
```

OUTPUT of our inserted data: From vehicle to Stats

V_ID	V_MODEL	V_MAKE	COST_PER_DAY
1	Civic	Honda	50
2	Accord	Honda	70
3	Corolla	Toyota	60
4	Camry	Toyota	80

R_ID	START_DATE	END_DATE	C_ID	V_ID	TOTAL_COST
1	10-MAR-23	15-MAR-23	1	1	250
2	20-MAR-23	25-MAR-23	2	2	420
3	01-APR-23	05-APR-23	3	3	240
4	10-APR-23	15-APR-23	4	4	560

C_ID	F_NAME	L_NAME	DOB	CITY	GENDER
1	John	Doe	09-JAN-90	New York	M
2	Jane	Doe	05-MAY-95	Los Angeles	F
3	Bob	Smith	08-OCT-85	Chicago	M
4	Alice	Johnson	20-FEB-00	Houston	F

R_ID	START_DATE	END_DATE	C_ID	V_ID	TOTAL_COST	ADDED_BY	ADDED_DATE
1	10-MAR-22	15-MAR-22	1	1	250	Admin	31-DEC-22
2	20-MAR-22	25-MAR-22	2	2	420	Admin	31-DEC-22
3	01-APR-22	05-APR-22	3	3	240	Admin	31-DEC-22
4	10-APR-22	15-APR-22	4	4	560	Admin	31-DEC-22

C_ID	F_NAME	L_NAME	DOB	CITY	GENDER	REASON	ADDED_BY	ADDED_DATE
1	John	Doe	01-JAN-90	New York	M	Unpaid reservation	Admin	15-MAR-23
3	Bob	Smith	10-AUG-85	Chicago	M	Damaged vehicle	Admin	05-APR-23

YEAR	C_ID	AGE	GENDER	TOT_RESERVATIONS	TOT_DAYS_OF_RESERVATIONS	TOT_COST_OF_RESERVATIONS
2023	1	33	M	2	10	120
2023	2	28	F	1	5	70
2023	3	38	M	1	5	240
2023	4	23	F	1	5	112

2.2 Definition and creation of at least two functions with results

2.2.1 1st function check if client is at least 18 years old by checking customer id

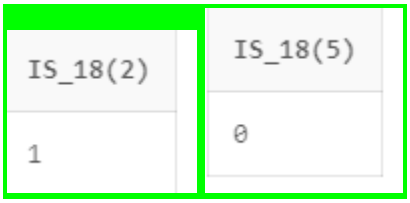
```
CREATE OR REPLACE FUNCTION is_18 (
  p_c_id IN CLIENT.C_ID%TYPE
)
RETURN NUMBER
AS
  age INT;
BEGIN
  SELECT abs(EXTRACT(YEAR FROM DOB ) - EXTRACT(YEAR FROM SYSDATE))
INTO age FROM CLIENT WHERE C_ID = p_c_id;
  IF age >= 18 THEN
      RETURN 1;
  ELSE
      RETURN 0;
  END IF ;

END;
```

– TEST CASE

```
 – add a client age under 18
INSERT INTO CLIENT (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER) VALUES (5,
'Yīng', 'Wáng', TO_DATE('02-20-2018','MM-DD-YYYY'), 'Houston', 'F');
SELECT is_18(2) FROM dual; -- over 18
SELECT is_18(5) FROM dual; -- under 18
```

– OUTPUT



IS_18(2)
1

IS_18(5)
0

2.2.2 2nd function - calculate estimated cost of a reservation based on vehicles rate and duration of using the car

```
CREATE OR REPLACE FUNCTION calculate_estimated_cost (
  p_v_id VEHICLE.V_ID%TYPE,
  p_start_date RESERVATION.START_DATE%TYPE,
  p_end_date RESERVATION.END_DATE%TYPE
) RETURN NUMBER
AS
  v_cost_per_day DECIMAL(10,2);
  v_num_days NUMBER;
  v_estimated_cost DECIMAL(10,2);
BEGIN
    -- Get the cost per day of the vehicle
    SELECT COST_PER_DAY
    INTO v_cost_per_day
    FROM VEHICLE
    WHERE V_ID = p_v_id;

    -- Calculate the number of days between start and end dates
    v_num_days := p_end_date - p_start_date;

    -- Calculate the estimated cost of the reservation
    v_estimated_cost := v_cost_per_day * v_num_days;

    -- Return the estimated cost
    RETURN v_estimated_cost;
END;
```

– TEST CASE

```
-- get car with v_id = 2 and rent it for 5 days
SELECT calculate_estimated_cost(2,'10-MAR-23','15-MAR-23') FROM dual;
```

– OUTPUT

CALCULATE_ESTIMATED_COST(2,'10-MAR-23','15-MAR-23')
350

2.3 Definition and creation of at least four procedures with results

2.3.1 PROCEDURE 1: procedure to register a new client and assign them a client ID

```
-- create sequence for customer ID
CREATE SEQUENCE c_id_sequence
    START WITH 5
    INCREMENT BY 1;

CREATE OR REPLACE PROCEDURE register_new_clientID (
          p_first_name CLIENT.F_NAME%TYPE,
    p_last_name CLIENT.L_NAME%TYPE,
    p_dob CLIENT.DOB%TYPE,
    p_city CLIENT.CITY%TYPE,
    p_gender CLIENT.GENDER%TYPE
) AS
   new_client_id   NUMBER;
BEGIN
    -- get sequence
  SELECT c_id_sequence.NEXTVAL INTO new_client_id FROM dual;

  -- insert into table
  INSERT INTO CLIENT (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER)
  VALUES (new_client_id, p_first_name, p_last_name, p_dob, p_city, p_gender);

END;
```

-- TEST CASE

```
EXEC  register_new_clientID('Xiao', 'Wang', TO_DATE('2000-01-01', 'YYYY-MM-DD'),
'Shanghai', 'F');
SELECT * FROM CLIENT;
```

– OUTPUT

C_ID	F_NAME	L_NAME	DOB	CITY	GENDER
1	John	Doe	09-JAN-90	New York	M
2	Jane	Doe	05-MAY-95	Los Angeles	F
3	Bob	Smith	08-OCT-85	Chicago	M
4	Alice	Johnson	20-FEB-00	Houston	F
6	Xiao	YAMA	01-JAN-00	Shanghai	F
5	Ying	Wáng	20-FEB-18	Houston	F

2.3.2 PROCEDURE 2 : procedure to add a new vehicle to the database and assign it a vehicle ID

```
CREATE SEQUENCE v_id_sequence
START WITH 5
INCREMENT BY 1;

CREATE OR REPLACE PROCEDURE add_new_vehicle  (
  p_v_model VEHICLE.V_MODEL%TYPE,
  p_v_make VEHICLE.V_MAKE%TYPE,
  p_cost_per_day VEHICLE.COST_PER_DAY%TYPE
) AS
   new_vehicle_id NUMBER;
BEGIN
    -- get sequence
    SELECT v_id_sequence.NEXTVAL INTO new_vehicle_id FROM dual;
           INSERT INTO VEHICLE (V_ID, V_MODEL, V_MAKE, COST_PER_DAY)
           VALUES (new_vehicle_id, p_v_model, p_v_make, p_cost_per_day);
END;
```

-- TEST CASE

```
SELECT * FROM VEHICLE
EXECUTE add_new_vehicle('Saga', 'Proton', 50.00);
```

– OUTPUT

V_ID	V_MODEL	V_MAKE	COST_PER_DAY
1	Civic	Honda	50
2	Accord	Honda	70
3	Corolla	Toyota	60
4	Camry	Toyota	80
5	Saga	Proton	50

2.3.3 PROCEDURE 3 : procedure to make a new reservation for a client, calculate the estimated cost, and insert the reservation details into the reservations table. note, this is just estimate cost, we add christmas discounts, etc stuff with other procedure later

```
-- SEQUENCE id for new reservation
CREATE SEQUENCE reservation_id_sequence
START WITH 5
INCREMENT BY 1;
CREATE OR REPLACE PROCEDURE make_reservation   (
    -- parameters
    -- rental dates
    p_start_date RESERVATION.START_DATE%TYPE,
    p_end_date RESERVATION.END_DATE%TYPE,
    -- client and vehicle ID
    p_client_id CLIENT.C_ID%TYPE,
            p_vehicle_id VEHICLE.V_ID%TYPE
) AS
    new_reservation_id NUMBER;
            duration NUMBER;
            age_client NUMBER;
            total_cost NUMBER;
            rental_cost NUMBER;
BEGIN
    -- get sequence
    SELECT v_id_sequence.NEXTVAL INTO new_reservation_id FROM dual;

            -- check if client is above 18 years old ?
            -- if yes, insert a new reservation to table

            IF is_18(p_client_id) THEN
                        -- find duration of reservation
        duration := p_end_date - p_start_date;
                        SELECT COST_PER_DAY  INTO rental_cost FROM VEHICLE WHERE V_ID =
p_vehicle_id;
        total_cost := rental_cost * duration; -- cost per day * duration
                        -- insert the reservation
                        INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST)
                        VALUES (new_reservation_id, p_start_date, p_end_date, p_client_id,
p_vehicle_id, total_cost);
            ELSE
        -- client is not eligible to make a reservation
        DBMS_OUTPUT.PUT_LINE('Client is not eligible to make a reservation');
            END IF;
END;
```

– TEST CASE

```
-- TEST CASE FOR PROCEDURE 3: rent 15 days, client 4, with id 5 car to make 6th
reservation
EXEC make_reservation('10-MAR-23','25-MAR-23',4, 5)
```

– OUTPUT

2.3.3 PROCEDURE 4 : Create a procedure to calculate the final cost of a reservation based on the rental rate, duration, and any applicable discounts, and update the reservation with the final cost and end date

```
CREATE OR REPLACE PROCEDURE calculate_final_cost   (
    -- (a) parameters
    p_reservation_id RESERVATION.R_ID%TYPE
) AS
    -- (b) variables
    duration NUMBER;
    v_discount NUMBER  := 0.2;
    v_total_cost NUMBER;

    v_start_date RESERVATION.START_DATE%TYPE;
    v_end_date RESERVATION.END_DATE%TYPE;

BEGIN
    -- retrive total cost
    SELECT TOTAL_COST INTO v_total_cost FROM RESERVATION WHERE R_ID =
p_reservation_id;

    -- calculate duration
    SELECT START_DATE, END_DATE INTO v_start_date, v_end_date FROM
RESERVATION WHERE R_ID = p_reservation_id;
    duration := v_end_date - v_start_date;
          DBMS_OUTPUT.PUT_LINE(duration);

    -- If the reservation is more than 10 days, apply a 10% discount.
    IF duration >= 10 THEN
        v_total_cost := v_total_cost * 0.9;
    END IF;

    -- If the reservation contains Christmas day, apply a 20% discount
    IF (EXTRACT(MONTH FROM v_start_date) <= 12 AND EXTRACT(MONTH FROM
v_end_date) >= 12 AND EXTRACT(DAY FROM v_start_date) <= 25 AND
EXTRACT(DAY FROM v_end_date) >= 25) THEN
          v_total_cost := v_total_cost * (1 - v_discount);
          END IF;

          DBMS_OUTPUT.PUT_LINE('The total cost of the reservation is: ' || v_total_cost);
END;
```

– TESTCASE

```
-- TEST CASE 0 : RESERVATION HAS NOTHING SPECIAL
EXECUTE calculate_final_cost(1);

-- TEST CASE 1 : RESERVATION WITH MORE THAN 10 DAYS -- $ 1100 original  -> $990 after
-- add reservation that is >= 10 days
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID, TOTAL_COST)
VALUES (5, TO_DATE('04-20-2023', 'MM-DD-YYYY'), TO_DATE('04-30-2023', 'MM-DD-YYYY'),
4, 4, 1100.00);
EXECUTE calculate_final_cost(5);

-- TEST CASE 2 : RESERVATION consists Dec 25th ? e.g. --$900 -> $720
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID, TOTAL_COST)
VALUES (6, TO_DATE('12-20-2025', 'MM-DD-YYYY'), TO_DATE('12-28-2025', 'MM-DD-YYYY'),
4, 1, 900.00);
EXECUTE calculate_final_cost(6);

-- TEST CASE 3 : COMBINED TEST CASE 1 AND 2, total 30% applied $900 -> $630
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID, TOTAL_COST)
VALUES (7, TO_DATE('12-15-2025', 'MM-DD-YYYY'), TO_DATE('12-28-2025', 'MM-DD-YYYY'),
4, 1, 900.00);
EXECUTE calculate_final_cost(7);
```

– OUTPUT

```
Statement processed.
5
The total cost of the reservation is: 250


1 row(s) inserted.


Statement processed.
10
The total cost of the reservation is: 990


1 row(s) inserted.


Statement processed.
8
The total cost of the reservation is: 720


1 row(s) inserted.


Statement processed.
13
The total cost of the reservation is: 648
```

2.4 : Definition and creation of at least two triggers with results

2.4.1 Trigger 1 : a trigger to automatically move completed reservations to the archived_reservations table at the end of the year

```
CREATE OR REPLACE TRIGGER move_completed_reservations
AFTER UPDATE ON RESERVATION
DECLARE
  lastday_currentyear DATE := TRUNC(ADD_MONTHS(TRUNC(SYSDATE, 'YEAR'), 12)) - 1;
BEGIN

        -- Move completed reservations to archived_reservations table
  IF :OLD.END_DATE <= lastday_currentyear  THEN
     INSERT INTO ARCHIVED_RESERVATIONS (R_ID, START_DATE, END_DATE, C_ID, V_ID, TOTAL_COST, ADDED_BY, ADDED_DATE)
     VALUES (:OLD.R_ID, :OLD.START_DATE, :OLD.END_DATE, :OLD.C_ID, :OLD.V_ID, :OLD.TOTAL_COST, USER, lastday_currentyear)

        -- Delete all reservations that is outdated
  DELETE  FROM RESERVATION WHERE END_DATE <= lastday_currentyear;

END;
```

– TEST CASE
UPDATE RESERVATION
SET TOTAL_COST = 260
WHERE R_ID = 1;

— insert a reservation that can’t be archived (e.g. end_date is 2024)
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID, TOTAL_COST) VALUES (5, TO_DATE('04-10-2023', 'MM-DD-YYYY'), TO_DATE('04-15-2024', 'MM-DD-YYYY'), 4, 4, 560.00);

2.4.2 Trigger 2 - If client is not paying, or car is broken by the client, the client is moved to black list table with the reason

– add a new column to reservation table
ALTER TABLE reservation ADD (paid NUMBER(1,0) DEFAULT 1);
ALTER TABLE reservation ADD (CAR_STATUS NUMBER(1,0) DEFAULT 0);

```
CREATE OR REPLACE TRIGGER add_to_blacklist
AFTER UPDATE ON RESERVATION
FOR EACH ROW
DECLARE
  v_reason VARCHAR2(50);
  v_fname VARCHAR2(50);
  v_lname VARCHAR2(50);
  v_dob DATE;
  v_city VARCHAR2(50);
  v_gender CHAR(1);
BEGIN
    -- if reservation is not paid ?
    IF :OLD.PAID = 0 THEN
       DBMS_OUTPUT.PUT_LINE('There is at least some clients did not pay during
reservation');
                    v_reason := 'Reservation not paid';
          END IF;

          -- if car is damagd ?
          IF :OLD.CAR_STATUS = 1 THEN
       DBMS_OUTPUT.PUT_LINE('There is at least some clients damaged the car');
        v_reason := v_reason || 'Car damaged during reservation';
    END IF;

          -- get client information from client table
          SELECT F_NAME, L_NAME, DOB, CITY, GENDER INTO v_fname, v_lname,
v_dob, v_city, v_gender FROM CLIENT WHERE C_ID = :OLD.C_ID;

          IF v_reason IS NOT NULL THEN
          INSERT INTO BLACK_LIST (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER,
REASON, ADDED_BY, ADDED_DATE)
                    VALUES (:OLD.C_ID, v_fname, v_lname, v_dob, v_city, v_gender,
v_reason, USER, SYSDATE);
```

```
    END IF;
END;
```

– TESTCASE

```
UPDATE RESERVATION
SET PAID = 0, CAR_STATUS = 1
WHERE R_ID = 2;
SELECT * FROM RESERVATION
SELECT * FROM BLACK_LIST
```

– OUTPUT

- Sorry, I ran out of time to debug.

2.5 Problem and solutions

2.5.1. List all the active bookings for the cars.

```
-- insert a reservation that is in the past
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST) VALUES (8, TO_DATE('04-10-2022', 'MM-DD-YYYY'),
TO_DATE('04-15-2022', 'MM-DD-YYYY'), 4, 4, 560.00);
-- 1 list all active booking cars
SELECT * FROM RESERVATION WHERE END_DATE >= SYSDATE;
```

– OUTPUT (R_ID = 8 Is not listed)

R_ID	START_DATE	END_DATE	C_ID	V_ID	TOTAL_COST	PAID	CAR_STATUS
5	20-APR-23	30-APR-23	4	4	1100	0	0
6	20-DEC-25	28-DEC-25	4	1	900	0	0
7	15-DEC-25	28-DEC-25	4	1	900	0	0
1	10-MAR-23	15-MAR-23	1	1	250	0	1
2	20-MAR-23	25-MAR-23	2	2	420	0	1
3	01-APR-23	05-APR-23	3	3	240	1	0
4	10-APR-23	15-APR-23	4	4	560	0	0

2.5.2 List all the details of the clients having the cars rented today.

```
– insert a reservation that is made today
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST) VALUES (9, sysdate, sysdate+4, 4, 4, 560.00);

SELECT C.C_ID, C.F_NAME, C.L_NAME, C.DOB, C.CITY, C.GENDER, R.V_ID,
R.TOTAL_COST
FROM RESERVATION R
JOIN CLIENT C ON R.C_ID = C.C_ID
WHERE R.START_DATE = SYSDATE;
```

– OUTPUT

R_ID	START_DATE	END_DATE	C_ID	V_ID	TOTAL_COST	PAID	CAR_STATUS
9	06-MAR-23	10-MAR-23	4	4	560	1	0

2.5.3 List all blacklisted clients

SELECT * FROM BLACK_LIST;

C_ID	F_NAME	L_NAME	DOB	CITY	GENDER	REASON	ADDED_BY	ADDED_DATE
1	John	Doe	01-JAN-90	New York	M	Unpaid reservation	Admin	15-MAR-23
3	Bob	Smith	10-AUG-85	Chicago	M	Damaged vehicle	Admin	05-APR-23

2.5.4 How many bookings do we have at Christmas time?

SELECT COUNT(R_ID)
FROM RESERVATION
WHERE (EXTRACT(MONTH FROM START_DATE) <= 12 AND EXTRACT(MONTH FROM END_DATE) >= 12 AND EXTRACT(DAY FROM START_DATE) <= 25 AND EXTRACT(DAY FROM END_DATE) >= 25);

SELECT * FROM RESERVATION;

COUNT(R_ID)
2

R_ID	START_DATE	END_DATE	C_ID	V_ID	TOTAL_COST	PAID	CAR_STATUS
8	10-APR-22	15-APR-22	4	4	560	1	0
5	20-APR-23	30-APR-23	4	4	1100	0	0
6	20-DEC-25	28-DEC-25	4	1	900	0	0
7	15-DEC-25	28-DEC-25	4	1	900	0	0
9	06-MAR-23	10-MAR-23	4	4	560	1	0
1	10-MAR-23	15-MAR-23	1	1	250	0	1
2	20-MAR-23	25-MAR-23	2	2	420	0	1
3	01-APR-23	05-APR-23	3	3	240	1	0
4	10-APR-23	15-APR-23	4	4	560	0	0

2.5.5 Generate the total revenue generated on a specific vehicle.

```
SELECT SUM(TOTAL_COST)
FROM RESERVATION
WHERE V_ID = 1;
```

SUM(TOTAL_COST)
2050

R_ID	START_DATE	END_DATE	C_ID	V_ID	TOTAL_COST	PAID	CAR_STATUS
8	10-APR-22	15-APR-22	4	4	560	1	0
5	20-APR-23	30-APR-23	4	4	1100	0	0
6	20-DEC-25	28-DEC-25	4	1	900	0	0
7	15-DEC-25	28-DEC-25	4	1	900	0	0
9	06-MAR-23	10-MAR-23	4	4	560	1	0
1	10-MAR-23	15-MAR-23	1	1	250	0	1
2	20-MAR-23	25-MAR-23	2	2	420	0	1
3	01-APR-23	05-APR-23	3	3	240	1	0
4	10-APR-23	15-APR-23	4	4	560	0	0

3.0 Inserted sample data

```
-- fill in sample data
INSERT INTO VEHICLE (V_ID, V_MODEL, V_MAKE, COST_PER_DAY) VALUES (1,
'Civic', 'Honda', 50.00);
INSERT INTO VEHICLE (V_ID, V_MODEL, V_MAKE, COST_PER_DAY) VALUES (2,
'Accord', 'Honda', 70.00);
INSERT INTO VEHICLE (V_ID, V_MODEL, V_MAKE, COST_PER_DAY) VALUES (3,
'Corolla', 'Toyota', 60.00);
INSERT INTO VEHICLE (V_ID, V_MODEL, V_MAKE, COST_PER_DAY) VALUES (4,
'Camry', 'Toyota', 80.00);

INSERT INTO CLIENT (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER) VALUES (1,
'John', 'Doe', TO_DATE('01-09-1990','MM-DD-YYYY'), 'New York', 'M');
INSERT INTO CLIENT (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER) VALUES (2,
'Jane', 'Doe', TO_DATE('05-05-1995','MM-DD-YYYY'), 'Los Angeles', 'F');
INSERT INTO CLIENT (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER) VALUES (3,
'Bob', 'Smith', TO_DATE('10-08-1985','MM-DD-YYYY'), 'Chicago', 'M');
INSERT INTO CLIENT (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER) VALUES (4,
'Alice', 'Johnson', TO_DATE('02-20-2000','MM-DD-YYYY'), 'Houston', 'F');

INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST) VALUES (1, TO_DATE('03-10-2023', 'MM-DD-YYYY'),
TO_DATE('03-15-2023', 'MM-DD-YYYY'), 1, 1, 250.00);
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST) VALUES (2, TO_DATE('03-20-2023', 'MM-DD-YYYY'),
TO_DATE('03-25-2023', 'MM-DD-YYYY'), 2, 2, 420.00);
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST) VALUES (3, TO_DATE('04-01-2023', 'MM-DD-YYYY'),
TO_DATE('04-05-2023', 'MM-DD-YYYY'), 3, 3, 240.00);
INSERT INTO RESERVATION (R_ID, START_DATE, END_DATE, C_ID, V_ID,
TOTAL_COST) VALUES (4, TO_DATE('04-10-2023', 'MM-DD-YYYY'),
TO_DATE('04-15-2023', 'MM-DD-YYYY'), 4, 4, 560.00);

INSERT INTO ARCHIVED_RESERVATION (R_ID, START_DATE, END_DATE, C_ID,
V_ID, TOTAL_COST, ADDED_BY, ADDED_DATE) VALUES (1, TO_DATE('03-10-2022',
'MM-DD-YYYY'), TO_DATE('03-15-2022', 'MM-DD-YYYY'), 1, 1, 250.00, 'Admin',
TO_DATE('12-31-2022', 'MM-DD-YYYY'));
INSERT INTO ARCHIVED_RESERVATION (R_ID, START_DATE, END_DATE, C_ID,
V_ID, TOTAL_COST, ADDED_BY, ADDED_DATE) VALUES (2, TO_DATE('03-20-2022',
```

```
'MM-DD-YYYY'), TO_DATE('03-25-2022', 'MM-DD-YYYY'), 2, 2, 420.00, 'Admin',
TO_DATE('12-31-2022', 'MM-DD-YYYY'));
INSERT INTO ARCHIVED_RESERVATION (R_ID, START_DATE, END_DATE, C_ID,
V_ID, TOTAL_COST, ADDED_BY, ADDED_DATE) VALUES (3, TO_DATE('04-01-2022',
'MM-DD-YYYY'), TO_DATE('04-05-2022', 'MM-DD-YYYY'), 3, 3, 240.00, 'Admin',
TO_DATE('12-31-2022', 'MM-DD-YYYY'));
INSERT INTO ARCHIVED_RESERVATION (R_ID, START_DATE, END_DATE, C_ID,
V_ID, TOTAL_COST, ADDED_BY, ADDED_DATE) VALUES (4, TO_DATE('04-10-2022',
'MM-DD-YYYY'), TO_DATE('04-15-2022', 'MM-DD-YYYY'), 4, 4, 560.00, 'Admin',
TO_DATE('12-31-2022', 'MM-DD-YYYY'));

-- Sample data for the BLACK_LIST table
INSERT INTO BLACK_LIST (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER,
REASON, ADDED_BY, ADDED_DATE)
VALUES (1, 'John', 'Doe', TO_DATE('01-01-1990', 'MM-DD-YYYY'), 'New York', 'M',
'Unpaid reservation', 'Admin', TO_DATE('03-15-2023', 'MM-DD-YYYY'));
INSERT INTO BLACK_LIST (C_ID, F_NAME, L_NAME, DOB, CITY, GENDER,
REASON, ADDED_BY, ADDED_DATE)
VALUES (3, 'Bob', 'Smith', TO_DATE('08-10-1985', 'MM-DD-YYYY'), 'Chicago', 'M',
'Damaged vehicle', 'Admin', TO_DATE('04-05-2023', 'MM-DD-YYYY'));

-- Sample data for the STATS table
INSERT INTO STATS (YEAR, C_ID, AGE, GENDER, TOT_RESERVATIONS,
TOT_DAYS_OF_RESERVATIONS, TOT_COST_OF_RESERVATIONS)
VALUES (2023, 1, 33, 'M', 2, 10, 120.00);
INSERT INTO STATS (YEAR, C_ID, AGE, GENDER, TOT_RESERVATIONS,
TOT_DAYS_OF_RESERVATIONS, TOT_COST_OF_RESERVATIONS)
VALUES (2023, 2, 28, 'F', 1, 5, 70.00);
INSERT INTO STATS (YEAR, C_ID, AGE, GENDER, TOT_RESERVATIONS,
TOT_DAYS_OF_RESERVATIONS, TOT_COST_OF_RESERVATIONS)
VALUES (2023, 3, 38, 'M', 1, 5, 240.00);
INSERT INTO STATS (YEAR, C_ID, AGE, GENDER, TOT_RESERVATIONS,
TOT_DAYS_OF_RESERVATIONS, TOT_COST_OF_RESERVATIONS)
VALUES (2023, 4, 23, 'F', 1, 5, 112.00);
```