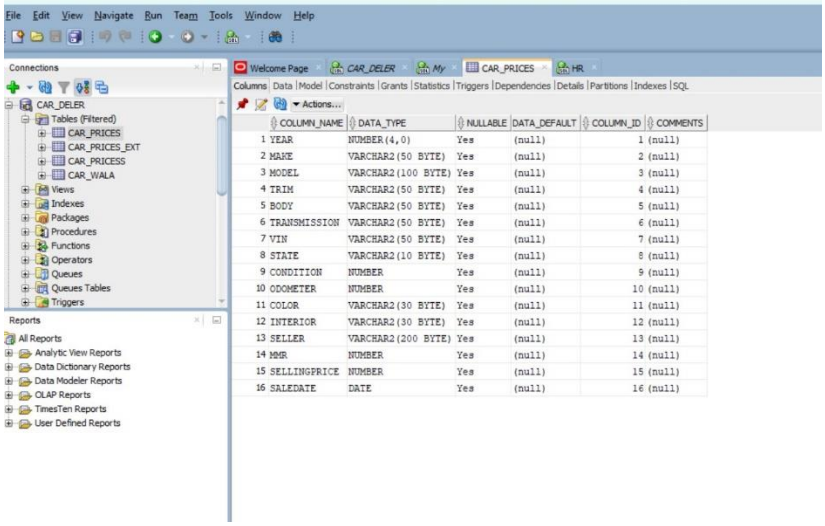
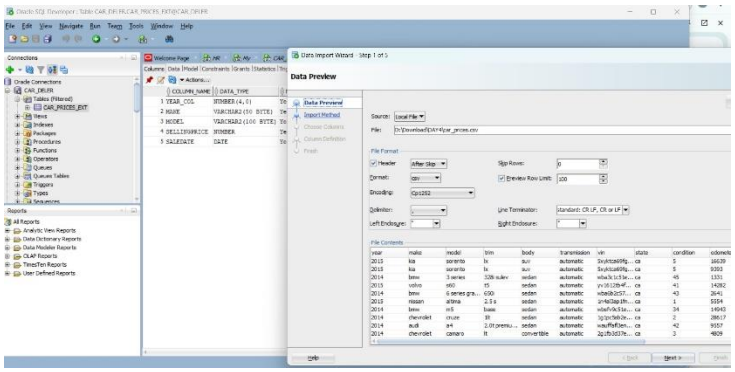
STEP1: CREATED TABLE (car_prices) :

```
CREATE TABLE CAR_PRICES (
    YEAR NUMBER(4),
    MAKE VARCHAR2(50),
    MODEL VARCHAR2(100),
    TRIM VARCHAR2(100),
    BODY VARCHAR2(50),
    TRANSMISSION VARCHAR2(50),
    VIN VARCHAR2(17),
    STATE VARCHAR2(10),
    CONDITION NUMBER(3),
    ODOMETER NUMBER(10),
    COLOR VARCHAR2(50),
    INTERIOR VARCHAR2(50),
    SELLER VARCHAR2(200),
    MMR NUMBER(10),
    SELLINGPRICE NUMBER(10),
    SALEDATE DATE
);
```



STEP2:

I have Imported my data set car_prices.csv to Oracle DB using below commands and granted access to my SQL USER 'HR':



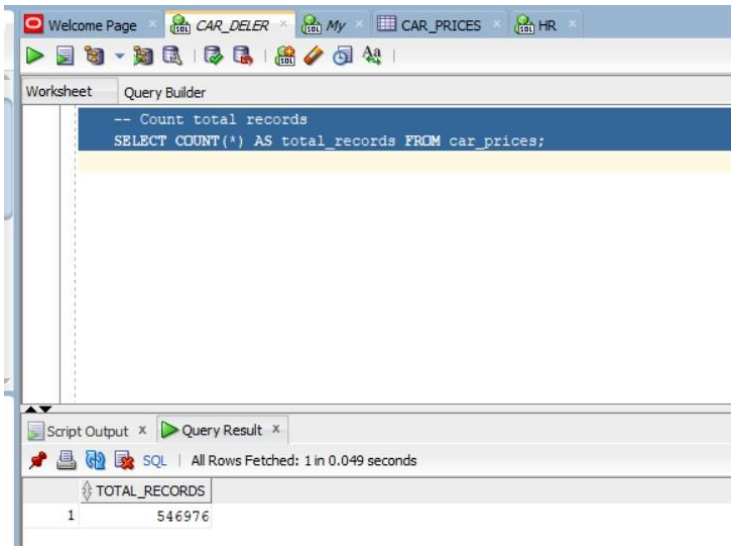
CREATE OR REPLACE DIRECTORY car_dir AS 'D:\Download\DAY4\car_prices.csv';

GRANT READ, WRITE ON DIRECTORY car_dir TO CAR_DELER

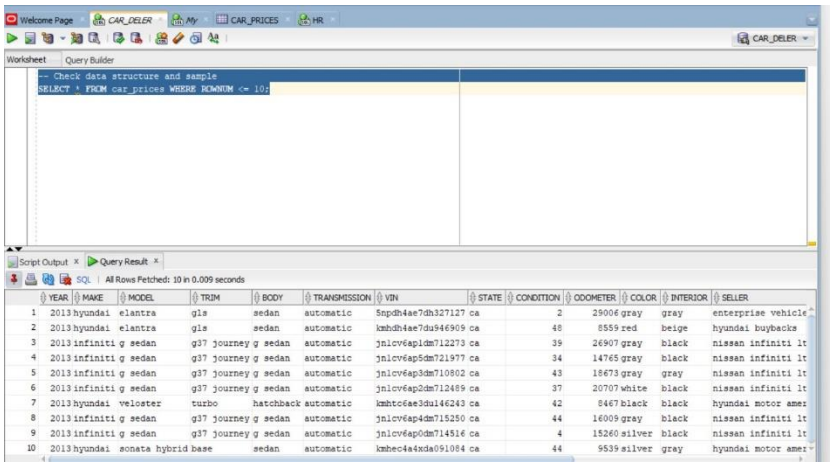
STEP3:

Basic analysis query's used:

SELECT COUNT(*) AS total_records FROM car_prices;



SELECT * FROM car_prices WHERE ROWNUM <= 10;



SELECT

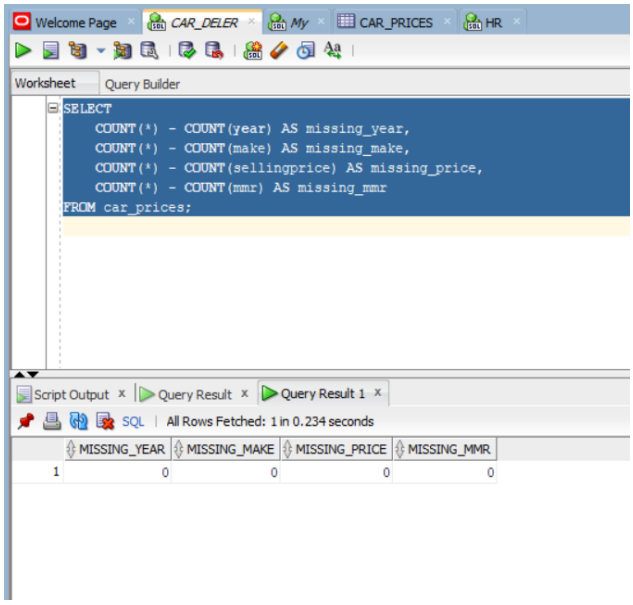
COUNT(*) - COUNT(year) AS missing_year,

COUNT(*) - COUNT(make) AS missing_make,

COUNT(*) - COUNT(sellingprice) AS missing_price,

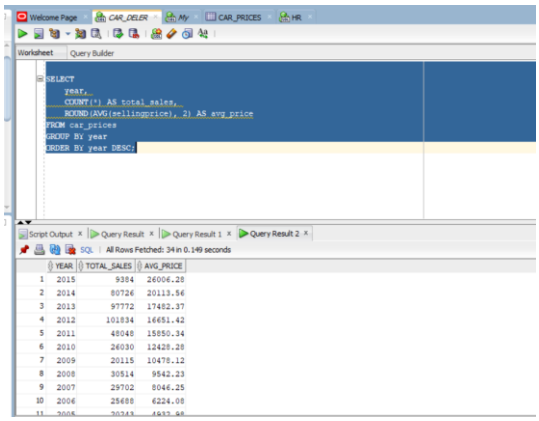
COUNT(*) - COUNT(mmr) AS missing_mmr

FROM car_prices;



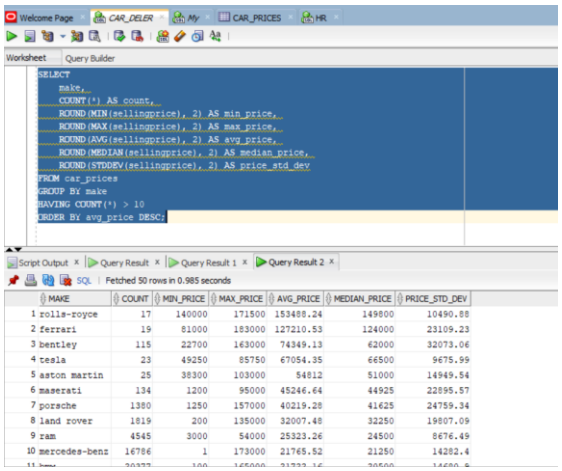
STEP4: Sales Analysis by Year

```
SELECT
    year,
    COUNT(*) AS total_sales,
    ROUND(AVG(sellingprice), 2) AS avg_price
FROM car_prices
GROUP BY year
ORDER BY year DESC;
```



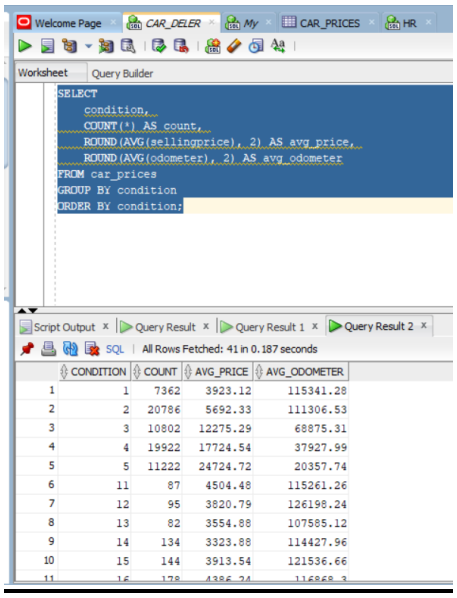
STEP5: Price Analysis By BRAND

```
SELECT
    make,
    COUNT(*) AS count,
    ROUND(MIN(sellingprice), 2) AS min_price,
    ROUND(MAX(sellingprice), 2) AS max_price,
    ROUND(AVG(sellingprice), 2) AS avg_price,
    ROUND(MEDIAN(sellingprice), 2) AS median_price,
    ROUND(STDDEV(sellingprice), 2) AS price_std_dev
FROM car_prices
GROUP BY make
HAVING COUNT(*) > 10
ORDER BY avg_price DESC;
```



STEP6: Vehicle Condition Analysis (*Average price by condition rating*)

```
SELECT
   condition,
   COUNT(*) AS count,
   ROUND(AVG(sellingprice), 2) AS avg_price,
   ROUND(AVG(odometer), 2) AS avg_odometer
FROM car_prices
GROUP BY condition
ORDER BY condition;
```



	CONDITION	COUNT	AVG_PRICE	AVG_ODOMETER
1	1	7362	3923.12	115341.28
2	2	20786	5692.33	111306.53
3	3	10802	12275.29	68875.31
4	4	19922	17724.54	37927.99
5	5	11222	24724.72	20357.74
6	11	87	4504.48	115261.26
7	12	95	3820.79	126198.24
8	13	82	3554.88	107585.12
9	14	134	3323.88	114427.96
10	15	144	3913.54	121536.66
11	16	178	4386.24	116868.3

Analyze cars with body type classification

```
WITH body_categories AS (
   SELECT 'SEDAN' AS body_type FROM DUAL
   UNION ALL SELECT 'SUV' FROM DUAL
   UNION ALL SELECT 'COUPE' FROM DUAL
   UNION ALL SELECT 'CONVERTIBLE' FROM DUAL
   UNION ALL SELECT 'WAGON' FROM DUAL
   UNION ALL SELECT 'HATCHBACK' FROM DUAL
)
SELECT
   bc.body_type,
   COUNT(cp.vin) AS vehicle_count,
   ROUND(AVG(cp.sellingprice), 2) AS avg_price,
   ROUND(AVG(cp.odometer), 2) AS avg_mileage
FROM body_categories bc
LEFT JOIN car_prices cp ON UPPER(cp.body) = bc.body_type
GROUP BY bc.body_type
ORDER BY vehicle_count DESC;
```

Welcome Page | CAR_DELER | My | CAR_PRICES | HR

Worksheet | Query Builder

```
    UNION ALL SELECT 'CONVERTIBLE' FROM DUAL
    UNION ALL SELECT 'WAGON' FROM DUAL
    UNION ALL SELECT 'HATCHBACK' FROM DUAL
)
SELECT
    bc.body_type,
    COUNT(cp.vin) AS vehicle_count,
    ROUND(AVG(cp.sellingprice), 2) AS avg_price,
    ROUND(AVG(cp.odometer), 2) AS avg_mileage
FROM body_categories bc
LEFT JOIN car_prices cp ON UPPER(cp.body) = bc.body_type
GROUP BY bc.body_type
ORDER BY vehicle_count DESC;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Re

SQL | All Rows Fetched: 6 in 0.239 seconds

	BODY_TYPE	VEHICLE_COUNT	AVG_PRICE	AVG_MILEAGE
1	SEDAN	236847	11841.12	62298.62
2	SUV	140655	16290.92	71353.05
3	HATCHBACK	25769	10131.5	51342.72
4	COUPE	17288	16192.16	66572.58
5	WAGON	15693	10271.26	73753.57
6	CONVERTIBLE	10227	18100.23	60657.51