

## Queries

1. The query retrieves information about vehicles in the database with mileage less than or equal to 25,000 miles and orders the results in descending order based on the vehicle mileage.

```
SELECT * FROM Vehicles WHERE VehicleMileage <= 25000 ORDER BY VehicleMileage DESC;
```

Vehicles Table: (Sample data obtained by using multiple *insert into* statements.)

	VEHICLEID	BRANCHID	VEHICLETYPEID	VEHICLEMILEAGE	VEHICLELICENSEPLATE
1	1	1	1	1	20000 ABC123
2	2	2	2	2	15000 XYZ789
3	3	1	1	3	30000 DEF456

Performing the query and the result:

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'Car Rental' with a schema named 'VEHICLES'. Below it, the 'Reports' section lists various report types. In the center, the 'Worksheet' tab contains the SQL query:

```
Select * from Vehicles Where VehicleMileage <= 25000 Order By VehicleMileage DESC;
```

Below the worksheet, the 'Script Output' tab shows the execution results:

All Rows Fetched: 2 in 0.1 seconds

VEHICLEID	BRANCHID	VEHICLETYPEID	VEHICLEMILEAGE	VEHICLELICENSEPLATE
1	1	1	1	20000 ABC123
2	2	2	2	15000 XYZ789

2. The query retrieves payment details from the database with an expiration date on or before '23-10-03'. (TODAY)

**SELECT \* FROM PaymentDetails WHERE PaymentDetailsExpDate <= '23-10-03' ;**

PaymentDetails Table:

	PAYMENTDETAILSID	CUSTOMERID	PAYMENTDETAILSCARDNUMBER	PAYMENTDETAILSCVV	PAYMENTDETAILSEXDATE
1		1	1	1234567890123456	123 25-12-31
2		2	2	9876543210987654	456 21-11-30
3		3	3	1111222233334444	789 23-10-02

Performing the query and the result:

The screenshot shows a database management interface with several panes:

- Connections:** A tree view showing available connections, with "Car Rental" selected.
- Worksheet:** Displays the SQL query: `Select * from PaymentDetails Where PaymentDetailsExpDate <= '2023-10-03';`
- Script Output:** Shows the execution status: "All Rows Fetched: 2 in 0.018 seconds".
- Query Result:** A table showing the retrieved data:

	PAYMENTDETAILSID	CUSTOMERID	PAYMENTDETAILSCARDNUMBER	PAYMENTDETAILSCVV	PAYMENTDETAILSEXDATE
1	2	2	9876543210987654	456	21-11-30
2	3	3	1111222233334444	789	23-10-02

3. The query retrieves distinct booking start dates from the "Bookings" table and calculates the total payments sum for each date. The results are ordered in ascending order based on the booking start date.

```
SELECT DISTINCT BookingLeaseStartDate, SUM(BookingDuePayment) FROM Bookings ORDER BY BookingLeaseStartDate ASC;
```

Bookings Table:

	BOOKINGID	CUSTOMERID	VEHICLEID	EMPLOYEEID	BOOKINGISACTIVE	BOOKINGLEASESTARTDATE	BOOKINGLEASEENDDATE	BOOKINGDUEPAYMENT
1	1	1	1	1	1	1 23-09-20	23-09-25	250
2	2	2	2	2	2	1 23-09-20	23-10-15	280
3	3	3	3	3	3	0 23-08-15	23-08-20	220

Performing the query and the result:

The screenshot shows the SSMS interface with the following details:

- Connections:** A tree view showing a connection to "Car Rental" and a table named "BOOKINGS".
- Worksheet:** The query is entered in the worksheet window:
 

```
SELECT BookingLeaseStartDate, SUM(BookingDuePayment) FROM Bookings GROUP BY BOOKINGLEASESTARTDATE ORDER BY BookingLeaseStartDate ASC ;
```
- Script Output:** Shows the execution message: "All Rows Fetched: 2 in 0.009 seconds".
 

BOOKINGLEASESTARTDATE	SUM(BOOKINGDUEPAYMENT)
1 23-08-15	220
2 23-09-20	530

4. This query counts the number of users in the database grouped by their respective countries.

```
SELECT UserCountry, COUNT(*) AS NumberOfUsers FROM Users GROUP BY UserCountry;
```

Users Table:

	USERID	USERFIRSTNAME	USERLASTNAME	USERMIDDLENAME	USERSTREETNAME	USERCITY	USERREGION	USERZIPCODE	USERCOUNTRY	USERDATEOFBIRTH	
1	1	John	Doe	(null)	123 Main St	Toronto	CA	90001	Canada	90-01-15	j
2	2	Alice	Smith	M	456 Elm St	New York	NY	10001	USA	85-05-20	a
3	3	Bob	Johnson	A	789 Oak St	Chicago	IL	60601	USA	92-08-10	b

Performing the query and the result:

The screenshot shows a database management interface with several panes:

- Connections:** A tree view showing available connections, with "PAYMENTDETAILSEXPORT" and "TRANSACTIONS" expanded.
- Worksheet:** The main workspace where the query is typed:

```
SELECT UserCountry, COUNT(*) AS NumberOfUsers
FROM Users
GROUP BY UserCountry;
```
- Script Output:** Shows the SQL command and the execution message: "All Rows Fetched: 2 in 0.015 seconds".
- Query Result:** A table showing the results of the query:

UserCountry	NumberOfUsers
USA	2
Canada	1
- Reports:** A section containing "All Reports" and various report types: Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports.

5. The query retrieves all customer records from the "Customers" table and sorts them in ascending order based on their driver's license numbers.

**SELECT \* FROM Customers ORDER BY customerdriverslicensenumber ASC;**

Customer Table:

	USERID	CUSTOMERDRIVERSLICENSENNUMBER
1		1 DL123456
2		2 AZ789012
3		3 DZ345678

Performing the query and the result:

The screenshot shows a database management interface with the following components:

- Connections:** A tree view showing tables: BRANCHID, BRANCHNAME, ADDRESSSTREET, ADDRESSCITY, ADDRESSREGION, ADDRESSZIPCODE, ADDRESSCOUNTRY, BRANCHPHONENUMB, CUSTOMERS (selected), EMPLOYEES, PAYMENTDETAILS.
- Worksheet:** The query `SELECT * FROM CUSTOMERS ORDER BY customerdriverslicensenumber ASC;` is entered in the Query Builder tab.
- Script Output:** Shows the execution message: `All Rows Fetched: 3 in 0.032 seconds`.
- Query Result:** Displays the same table data as the Customer Table above, sorted by CUSTOMERDRIVERSLICENSENNUMBER in ascending order.

6. The query retrieves distinct vehicle types from the "VehicleTypes" table, providing a list of unique combinations of vehicle make and model. (List all the unique vehicle models in the database)

**SELECT DISTINCT vehicletypemake, vehicletypemodel FROM VehicleTypes;**

VehicleType Table:

VEHICLETYPID	SUBTYPE	VEHICLETYPENAME	VEHICLETYPEYEAR	VEHICLETYPENAME	VEHICLETYPEDAILYRATE	SEATINGCAPACITY	MAXLOADWEIGHT	CARGOCAPACITY
1	1 C	Corolla	22-01-01	Toyota	50	5	(null)	
2	2 C	Focus	22-02-15	Ford	60	7	(null)	
3	3 T	Silverado	21-11-10	Chevrolet	80	3	2500	
4	4 C	Focus	22-04-18	Ford	60	7	(null)	

Performing the query and the result:

The screenshot shows the Toad Data Point interface. On the left, the Connections pane lists several tables under the VEHICLES category, including USERREGION, USERZIPCODE, USERCOUNTRY, USERDATEOFBIRTH, USEREMAIL, USERPHONENUMBER, USERPASSWORD, VEHICLES (with VEHICLEID, BRANCHID, VEHICLETYPEID, VEHICLEMILEAGE, VEHICLELICENSEPLATE), and VEHICLETYPES. Below this are Views, Indexes, Packages, and Procedures. On the right, the Worksheet pane contains the SQL query: `SELECT DISTINCT vehicletypemake, vehicletypemodel FROM vehicletypes;`. Below the query, the Script Output pane shows the results: `All Rows Fetched: 3 in 0.01 seconds`. The Query Result pane displays the following table:

VEHICLETYPENAME	VEHICLETYPENAME
1 Ford	Focus
2 Chevrolet	Silverado
3 Toyota	Corolla

7. The query counts the number of payments associated with each booking in the "Transactions" table. It groups the results by booking ID and orders them in ascending order based on the booking ID.

```
SELECT BookingID, COUNT(*) AS NumberOfPayments FROM Transactions
GROUP BY BookingID ORDER BY BookingID
```

Transactions Table:

	TRANSACTIONID	TRANSACTIONAMOUNT	TRANSACTIONDATE	PAYMENTDETAILSID	BOOKINGID
1	2	80	23-09-23	2	2
2	3	60	23-09-24	1	1
3	4	70	23-10-12	2	2
4	5	45	23-08-18	3	3

Performing the query and the result:

The screenshot shows a database management system interface with the following components:

- Connections:** A tree view showing database objects like PAYMENTDETAILSID, CUSTOMERID, PAYMENTDETAILSCA, PAYMENTDETAILSCV, PAYMENTDETAILSEX, TRANSACTIONS, and USERS.
- Worksheet:** The main workspace where the SQL query is entered:
 

```
SELECT BookingID, COUNT(*) AS NumberOfPayments FROM Transactions GROUP BY BookingID ORDER BY BookingID
```
- Script Output:** Shows the execution results:
 

BOOKINGID	NUMBEROFPAYMENTS
1	1
2	2
3	1

 A message at the bottom indicates: "All Rows Fetched: 3 in 0.068 seconds".

8. The query retrieves distinct employee permissions from the "Employees" table, providing a list of unique employee permission types.

**SELECT DISTINCT EmployeePermissions FROM Employees**

Employee Table:

	USERID	BRANCHID	EMPLOYEEMONTHLYSALARY	EMPLOYEESINNUMBER	EMPLOYEEPERMISSIONS
1	1	1		5000 123456789	Manager
2	2	2		4000 987654321	Employee
3	3	1		5500 456789123	Employee

Performing the query and the result:

The screenshot shows a database management environment with the following components:

- Connections:** A tree view of database objects including BRANCHPHONENUMB, CUSTOMERS, EMPLOYEES, PAYMENTDETAILS, TRANSACTIONS, and USERS.
- Worksheet:** The main area where the SQL query is entered:

```
SELECT DISTINCT EmployeePermissions FROM Employees;
```
- Script Output:** Shows the execution details: "All Rows Fetched: 2 in 0.011 seconds".
- Query Result:** Displays the results of the query:

EMPLOYEEPERMISSIONS
1 Manager
2 Employee
- Reports:** A section listing various report types: All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports.

**9.** This query retrieves branches located in the 'NY' region and orders them in ascending order based on the branch ID.

**SELECT \* FROM Branches WHERE AddressRegion = 'NY' ORDER BY BranchID**

Branchs Table:

	BRANCHID	BRANCHNAME	ADDRESSSTREET	ADDRESSCITY	ADDRESSREGION	ADDRESSZIPCODE	ADDRESSCOUNTRY	BRANCHPHONENUMBER
1	1	Downtown Branch	500 Elm St	New York	NY	10001	USA	5552223333
2	2	Uptown Branch	700 Oak St	Chicago	IL	60601	USA	5554445555
3	3	Westside Branch	800 Maple St	Toronto	ON	90001	Canada	5556667777

Performing the query and the result:

The screenshot shows a database management system interface with the following components:

- Connections:** A tree view showing a connection to "Car Rental" which contains tables like BOOKINGS, BRANCHS, CUSTOMERS, and PAYMENTDETAILS.
- Worksheet:** A SQL editor window containing the query:

```
SELECT * FROM Branches WHERE addressregion= 'NY' ORDER BY BranchID;
```
- Script Output:** A window showing the execution results:

```
All Rows Fetched: 1 in 0.255 seconds
```

	BRANCHID	BRANCHNAME	ADDRESSSTREET	ADDRESSCITY	ADDRESSREGION	ADDRESSZIPCODE	ADDRESSCOUNTRY	BRANCHPHONENUMBER
1	1	Downtown Branch	500 Elm St	New York	NY	10001	USA	5552223333