Milestone #3

1. For each user view from milestone # 1 formulate a few requests for information (use English sentences) which require browsing the business data from your Oracle database implemented in milestone # 2. You need to formulate 35 requests.

1. Retrieve the names of customers who ordered an RV and it is available.
2. Find the total number of RV orders with a payment amount over $6,000.
3. Retrieve the details of all RV orders along with the associated customer names along with total order value.
4. List all employees hold a position as salesperson and group by their contact information.
5. Find out the total number of RV orders with payment amounts greater than the average payment amount.
6. Retrieve a list of service appointments, including appointment ID, service date, service type, and details of the recreational vehicle (RV) assigned to each appointment, such as RV ID and model.
7. Display the total number of payments made for RV orders placed by customers with last names starting with 'S'.
8. List the payment details associated with all RV orders, including the payment amount, date, and customer's first name.
9. Find the total number of RVs available in each condition (New, Used, Demo) which are subtypes of super type Recreational Vehicle.
10. Retrieve supplier information along with the RV details they are associated with.
11. list all RV parts manufactured in Thornhill Service Centre that are under $100
12. List the average RV purchase price for customers under 40.
13. Display a list of RV service appointments, sorted by appointment date using ORDER() method, including the appointment ID, service date, service type, and the recreational vehicle model associated with each appointment.

14. Retrieve the details of RV orders placed by customers who were born after 1990, including the customer's name, order ID, and order date.
15. . Retrieve the total number of service appointments scheduled for each RV model.
16. Retrieve the payment details associated with all RV orders, including the payment amount and date along with customer id.
17. Display the name, address, phone number, and details of recreational vehicles including make, model, year, price, condition, and availability from the Supplier.
18. Retrieve the details of RV orders where the payment amount is greater than $1000 and more than the average payment amount, including the order ID, order date, RV ID, make, model, year, price, and payment amount.
19. List the depreciation and service details for used recreational vehicles.
20. Retrieve customer information along with the associated employee names from the database, ensuring that only valid employee names are included.
21. List the details of RV parts with prices below $110, along with available services and associated RV details.
22. Retrieve the appointment IDs, service types, and RV models of all RV service appointments where the service date is after February 25, 2024.
23. Retrieve customer orders along with their details (order ID, order date, customer's first and last name, and order total) and the corresponding service types of RV service appointments.
24. Retrieve the names and positions of employees whose contact number starts with '555' and whose EmployeeID is greater than 2.
25. Retrieve the total available RVs along with their makes, models, and prices for RVs that are available and have a price greater than $40,000.
26. Retrieve the names and positions of employees whose contact number starts with '55' and whose EmployeeID is greater than 1 and who are involved in servicing RVs with Model1 or Model2.
27. Retrieve the first names and last names of customers who placed orders with a total amount greater than $20,000 and the RV models of the RVs they ordered.

28. Retrieve the customer IDs, payment dates, and payment amounts for all payments where the payment amount exceeds $5500.
29. Retrieve the employee details, including name, position, and contact information, along with the first name and last name of the customer who placed an order with order ID 4, for the employee with EmployeeID 4 whose contact number ends with '333'.
30. Retrieve the service types and RV models for all RV service appointments where the RV model is either Model1 or Model2 and the service date is before February 28, 2024.
31. Retrieve the names and positions of employees whose contact number contains '987' and who are involved in servicing RVs with ModelX.
32. Retrieve the total available RVs along with their makes, models, and prices for RVs that are available and have a price greater than $44,000.
33. What are the service type and recreational vehicle model for the RV service appointment with AppointmentID 4, where the RV model is 'ModelZ'.
34. Retrieve the names and positions of employees whose contact number contains '32' and who are involved in servicing RVs with Model2.
35. Retrieve the service dates and service types of all RV service appointments where the service type is either "Annual Maintenance" or "Repair" and the service date is before March 7, 2024.
36. Get all models and service dates where service type is “Oil Change”.
37. Retrieve the first names, last names, and order dates for all RV orders placed by customers whose last name starts with 'W' and the order date is after February 25, 2024.

2. For 15 requests formulate SQL statements with Oracle XML functions returning XML documents such that:
    1. All SQL queries access object data from more than one object table using references;
    2. All XML documents contain more than two levels of element hierarchy;
    3. Five of your documents are to contain XML tags with attributes;
    4. Five queries must contain GROUP BY clause. Each query must return only one XML document and demonstrate grouping of elements of the same type;

5. Demonstrate application of XMLFOREST, XMLAGG and XMLROOT functions;
6. All the documents must be well-formed: contain XML declaration and the root element.

1: Retrieve the names of customers who ordered an RV and it is available.

```
SELECT XMLROOT(
        XMLELEMENT("CustomerList",
          XMLAGG(
            XMLELEMENT("Customer",
              XMLFOREST(
                ro.customer_id.first_name AS "FirstName",
                ro.customer_id.last_name AS "LastName",
                ro.customer_id.email AS "Email",
                ro.customer_id.phone_number AS "PhoneNumber"
              )
            )
          )
        ),
        VERSION '1.0'
      ) AS "CustomerOrders"
   FROM RVOrder_Table ro
   WHERE ro.rv_id.availability = 'Available';
```

2. Find the total number of RV orders with a payment amount over $6,000.

```
SELECT XMLROOT(

        XMLELEMENT("RV_Orders",

            XMLAGG(

                XMLELEMENT("RV_Order",

                    XMLATTRIBUTES(

                        o.order_date AS "Order_Date"

                    ),

                    XMLFOREST(

                        o.order_id AS "Order_ID",

                        o.payment_id.amount AS "Payment_Amount",
                        o.rv_id.model AS "RV_Model"
                    )
                )
            )
        ),
        VERSION '1.0'
    ) AS "RV_Orders_XML"
FROM RVOrder_Table o
WHERE o.payment_id.amount > 6000
GROUP BY o.order_date;
```

3. Retrieve the details of all RV orders along with the associated customer names along with total order value.

```
SELECT
    XMLROOT(
        XMLELEMENT("Customer_Details",
            XMLAGG(
                XMLELEMENT("RV_Order",
                    XMLFOREST(
                        o.order_id AS "Order_ID",
                        o.order_date AS "Order_Date",
                        o.customer_id.first_name AS "First_Name",
                        o.customer_id.last_name AS "Last_Name",
                        o.calculateOrderTotal() AS "Order_Total"
                    )
                )
            )
        ),
        VERSION '1.0'
    ).getClobVal() AS "XML_Output"
```

```
FROM
   RVOrder_Table o;
```

4. List all employees hold a position as salesperson and group by their contact information.

```
SELECT XMLROOT(
        XMLELEMENT("Employees",
           XMLAGG(
              XMLELEMENT("Employee",
                 XMLATTRIBUTES(
                    e.employee_id AS "EmployeeID"
                 ),
                XMLFOREST(
                  e.rv_id.model AS "model"
                ),
                 XMLFOREST(
                    e.employee_name AS "Name",
                    e.position AS "Position",
                    e.contact AS "Contact",
                  e.employee_id AS "EmployeeID"
                 )
              )
           )
        ),
        VERSION '1.0'
     ) AS doc
FROM Employee_Table e
WHERE e.position = 'Salesperson'
GROUP BY e.contact;
```

5. Find out the total number of RV orders with payment amounts greater than the average payment amount.

```
SELECT XMLROOT(
        XMLELEMENT("RV_Order_Total",
          XMLAGG(
            XMLELEMENT("payment_greater",
              XMLATTRIBUTES(
                o.order_id AS "Order_ID"
              ),
              XMLFOREST(
                o.order_date AS "Order_Date",
                XMLFOREST(
                  o.customer_id.first_name AS "Customer_First_Name",
                  o.customer_id.last_name AS "Customer_Last_Name"
                ) AS "Customer_Details",
                o.payment_id.amount AS "Payment_Amount"
              )
            )
          )
        ),
        VERSION '1.0'
     ) AS "doc"
FROM RVOrder_Table o
WHERE o.payment_id.amount > (SELECT AVG(o2.payment_id.amount) FROM
RVOrder_Table o2);
```

6. Retrieve a list of service appointments, including appointment ID, service date, service type, and details of the recreational vehicle (RV) assigned to each appointment, such as RV ID and model.

```
SELECT XMLROOT(
      XMLELEMENT("ServiceAppointments",
       XMLAGG(
        XMLELEMENT("ServiceAppointment",
         XMLATTRIBUTES(sa.appointment_id AS "AppointmentID"),
         XMLELEMENT("ServiceDate", sa.service_date),
         XMLELEMENT("ServiceType", sa.service_type),
         XMLELEMENT("RV_Details",
          XMLFOREST(sa.rv_id.rv_id AS "RV_ID", sa.rv_id.model AS "Model")
         )
        )
       )
      ),
      VERSION '1.0',
      STANDALONE YES
     ) AS "ServiceAppointments"
FROM RVServiceAppointment_Table sa;
```

7. Display the total number of payments made for RV orders placed by customers with last names starting with 'S'.

```
SELECT XMLROOT(
      XMLELEMENT(
        "Payment_Info",
        XMLAGG(
          XMLELEMENT(
            "Customer_Payments",
                XMLFOREST(
                    o.customer_id.last_name AS "Last_Name",
                    o.customer_id.first_name AS "First_Name",
                    COUNT(o.payment_id.payment_id) AS "Total_Payments"
                )
            )
        )
    ),
    VERSION '1.0'
) AS "Payment_XML"
FROM RVOrder_Table o
WHERE o.customer_id.last_name LIKE 'S%'
GROUP BY o.customer_id.last_name, o.customer_id.first_name;
```

8. List the payment details associated with all RV orders, including the payment amount, date, and customer's first name.

```
SELECT
    XMLROOT(
        XMLELEMENT("RVOrderList",
            XMLAGG(
                XMLELEMENT("Order",
                    XMLATTRIBUTES(o.order_id AS "oid"),
                    XMLFOREST(
                        o.payment_id.payment_id AS "pid",
                        o.payment_id.amount AS "amount",
                        o.payment_id.payment_date AS "date",
                        o.customer_id.first_name AS "firstname"
                    )
                )
            )
        ),
        VERSION '1.0'
    ) AS "Payment Details for all Orders"
FROM
    RVOrder_Table o;
```

9. Find the total number of RVs available in each condition (New, Used, Demo) which are subtypes of super type Recreational Vehicle.

```
SELECT
  XMLROOT(
    XMLELEMENT(
      "RV_Conditions",
      XMLAGG(
        XMLELEMENT(
          "Condition",
          XMLATTRIBUTES(rvCondition AS "Type"),
          XMLELEMENT(
            "Total_Available",
            COUNT(*)
          ),
          XMLELEMENT(
            "RVs",
            XMLAGG(
              XMLELEMENT(
                "RV",
                XMLFOREST(
                  rv_id AS "ID",
                  make AS "Make",
                  model AS "Model",
                  year AS "Year",
                  price AS "Price",
                  availability AS "Availability"
                )
              )
            )
```

```
                )
             )
          )
       ),
       VERSION '1.0'
   )
FROM
   RV_Table
WHERE
   availability = 'Available'
GROUP BY
   rvCondition;
```

10. Retrieve supplier information along with the RV details they are associated with.

```
SELECT XMLROOT(
        XMLELEMENT("Supplier",
           XMLATTRIBUTES(MAX(s.name) AS "Name"),
           XMLFOREST(MAX(s.address) AS "Address", MAX(s.phone_number) AS
"PhoneNumber"),
           XMLAGG(
              XMLELEMENT("RV",
                 XMLATTRIBUTES(s.rv_id.make AS "Make", s.rv_id.model AS "Model"),
                 XMLFOREST(s.rv_id.year AS "Year", s.rv_id.price AS "Price",
s.rv_id.availability AS "Availability", s.rv_id.rvCondition AS "Condition")
              )
           )
        ),
        VERSION '1.0'
     ) AS "SupplierInformation"
FROM Supplier_Table s
Group by s.rv_id.rvCondition, s.rv_id.year;
```

11. list all RV parts manufactured in Thornhill Service Centre that are under $100

```
SELECT XMLROOT(
        XMLELEMENT("ThornhillRVParts",
            XMLAGG(
              XMLELEMENT("Part",
                XMLATTRIBUTES(p.part_id AS "PartID"),
                XMLFOREST(
                  p.part_name AS "PartName",
                  p.price AS "Price",
                  p.location_id.contact AS "ServiceCentreContact",
                  p.appointment_id.rv_id.rv_id AS "RV_ID"
                )
              )
            )
          ),
          VERSION '1.0'
        ) AS "Doc"
FROM RVPart_Table p
WHERE p.location_id.address = 'Thornhill Service Centre' AND p.price < 100
GROUP BY p.location_id.contact;
```

12. List the average RV purchase price for customers under 40.

```
SELECT
  XMLROOT(
    XMLELEMENT("AveragePurchasePrice",
      XMLFOREST(
        AVG(ord.rv_id.price) AS "AvgU40yoPurchasePrice"
      )
    ),
    VERSION '1.0'
  ) AS "XMLResult"
FROM
  RVOrder_Table ord
```

```
WHERE
  EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
ord.customer_id.dob) < 40;
```

13. Display a list of RV service appointments, sorted by appointment date using ORDER() method, including the appointment ID, service date, service type, and the recreational vehicle model associated with each appointment.

```
SELECT XMLROOT(
           XMLELEMENT("RVServiceAppointments",
               XMLAGG(
                   XMLELEMENT("RVServiceAppointment",
                       XMLATTRIBUTES(sa_sorted.appointment_id AS "AppointmentID"),
                       XMLFOREST(
                           sa_sorted.service_date AS "ServiceDate",
                           sa_sorted.service_type AS "ServiceType",
                           XMLELEMENT("RecreationalVehicle",
                               XMLFOREST(
                                   rv.model AS "Model"
                               )
                           ) AS "RecreationalVehicle"
                       )
                   )
               )
           ),
           VERSION '1.0'
       ) AS "XMLResult"
FROM (
    SELECT sa.*, VALUE(sa).sort(RVServiceAppointment(0, SYSDATE, 'Dummy', NULL)) as
order_value
```

```
    FROM RVServiceAppointment_Table sa
) sa_sorted, RV_Table rv
WHERE sa_sorted.rv_id = REF(rv);
```

14. Retrieve the details of RV orders placed by customers who were born after 1990, including the customer's name, order ID, and order date.

```
SELECT XMLROOT(
           XMLELEMENT("RV_Orders",
                      XMLAGG(
                         XMLELEMENT("Order",
                                    XMLFOREST(
                                       o.customer_id.first_name || ' ' || o.customer_id.last_name AS
"Customer_Name",
                                       o.order_id AS "Order_ID",
                                       o.order_date AS "Order_Date"
                                    )
                                 )
                      )
                     ),
           VERSION '1.0',
           STANDALONE YES
       ) AS "XML_Result"
FROM RVOrder_Table o
WHERE o.customer_id.dob > TO_DATE('1990-01-01', 'YYYY-MM-DD')
GROUP BY o.customer_id.customer_id;
```

15. . Retrieve the total number of service appointments scheduled for each RV model.

```
SELECT XMLROOT(
        XMLELEMENT(
           "Service_Appointments",
           XMLAGG(
              XMLELEMENT(
                 "RV_Model",
                 XMLFOREST(
                    sa.rv_id.model AS "Model",
                    COUNT(sa.appointment_id) AS "Total_Service_Appointments"
                 )
              )
           )
        ),
        VERSION '1.0'
     ) AS "Service_Appointments_XML"
FROM RVServiceAppointment_Table sa
GROUP BY sa.rv_id.model;
```

16. Retrieve the payment details associated with all RV orders, including the payment amount and date along with customer id.

```
SELECT XMLROOT(
        XMLELEMENT("Payment_Details",
           XMLAGG(
              XMLELEMENT("Order",
                 XMLATTRIBUTES(
                    ro.customer_id.customer_id AS "Customer_ID"
                 ),
                 XMLFOREST(
                  ro.customer_id.customer_id AS "Customer_ID",
                    ro.order_id AS "Order_ID",
                    ro.order_date AS "Order_Date",
                    ro.payment_id.amount AS "Amount",
                    ro.payment_id.payment_date AS "Payment_Date"
                  )
                )
              )
            ),
            VERSION '1.0', STANDALONE YES)
   FROM RVOrder_Table ro
  group by ro.order_date, ro.payment_id.payment_date;
```

3. For five other requests create XML documents using Oracle XSU:
A. Formulate SQL queries accessing data from more than one object table through the references.
B. Provide user-defined tag names.

1. Display the name, address, phone number, and details of recreational vehicles including make, model, year, price, condition, and availability from the Supplier.

```
OracleXML getXML -user "grp3/grp3there" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowTag "RV_and_Supplier_Info" \
-rowsetTag "RV_List" \
"SELECT s.name AS SupplierName,
    s.displaySupplierAddress() AS SupplierAddress,
    s.phone_number AS PhoneNumber,
    s.rv_id.make AS Make,
    s.rv_id.model AS Model,
    s.rv_id.year AS Year,
    s.rv_id.price AS Price,
    s.rv_id.rvCondition AS Condition,
    s.rv_id.displayAvailability() AS Availability
FROM Supplier_Table s"
```

2. Retrieve the details of RV orders where the payment amount is greater than $1000 and more than the average payment amount, including the order ID, order date, RV ID, make, model, year, price, and payment amount.

```
 OracleXML getXML -user "grp3/grp3there" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowTag "RVOrder" \
-rowsetTag "RVOrderList" \
"SELECT
  ro.order_id AS OrderID,
  ro.order_date AS OrderDate,
  ro.rv_id.rv_id AS RV_ID,
  ro.rv_id.make AS Make,
  ro.rv_id.model AS Model,
  ro.rv_id.year AS Year,
  ro.rv_id.price AS Price,
  ro.payment_id.amount AS AveragePaymentAmount
FROM
  RVOrder_Table ro
WHERE
  ro.payment_id.amount > 1000 AND
  ro.payment_id.amount > (SELECT AVG(p.amount) FROM Payment_Table p)";
```

3. List the depreciation and service details for used recreational vehicles.

```
OracleXML getXML -user "grp3/grp3there" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowTag "Used_RV_Deprication" \
-rowsetTag "Service_List" \
"SELECT
   sr.rv_id.make AS Make,
   sr.rv_id.model AS Model,
   sr.rv_id.year AS Year,
   sr.rv_id.price AS Price,
   sr.rv_id.calculateDeprication() AS Depreciation,
   sr.service_date AS Service_Appointment_Date,
   sr.service_type AS Service_Type
FROM
   RVServiceAppointment_Table sr"
```

4. Retrieve customer information along with the associated employee names from the database, ensuring that only valid employee names are included.

```
OracleXML getXML -user "grp3/grp3there" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowTag "CustomerInfo" \
-rowsetTag "CustomerList" \
"SELECT c.displayCustomerDetails() AS CustomerDetails,
     c.employee_id.employee_name AS EmployeeName
FROM RVCustomer_Table c
WHERE c.employee_id.employee_name IS NOT NULL"
```

5. List the details of RV parts with prices below $110, along with available services and associated RV details.

```
OracleXML getXML -user "grp3/grp3there" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowTag "PartInfo" \
-rowsetTag "Part_Service_List" \
"SELECT p.part_id AS PartID,
     p.part_name AS PartName,
     p.price AS PartPrice,
       p.location_id.listAvailableServices() AS AvailableServices,
       p.appointment_id.getRvDetails() AS RV_Details
   FROM RVPart_Table p
   WHERE p.price < 110"
```

4. Store some of the created XML files in Oracle XML DB repository using DBMS_XDB PL/SQL package. Choose XML documents which allow you to implement the specifications from bullet 5.

Creating a folder:

```
declare
  ret boolean;
  begin
  ret
  :=dbms_xdb.createfolder('/public/grp3_rv_2024');
  commit;
  end;
   /
```

Creating Resources:

```
-- RVServiceAppointment
DECLARE
  ret BOOLEAN;
  RVServiceApptXML VARCHAR2(4000) :=
    '<?xml version="1.0"?>
<RVServiceAppointments>
  <RVServiceAppointment AppointmentID="1">
    <ServiceDate>2024-02-20</ServiceDate>
  <ServiceType>Annual Maintenance</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>Model1</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="2">
  <ServiceDate>2024-02-21</ServiceDate>
  <ServiceType>Repair</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>Model2</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="3">
  <ServiceDate>2024-02-22</ServiceDate>
  <ServiceType>Inspection</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>Model1</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="4">
  <ServiceDate>2024-02-23</ServiceDate>
  <ServiceType>Tire Replacement</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>ModelZ</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="5">
  <ServiceDate>2024-02-24</ServiceDate>
  <ServiceType>Oil Change</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>Model2</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="6">
  <ServiceDate>2024-02-25</ServiceDate>
  <ServiceType>Winterization</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>ModelX</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="7">
  <ServiceDate>2024-02-26</ServiceDate>
  <ServiceType>Interior Cleaning</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>ModelZ</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="8">
```

```
  <ServiceDate>2024-02-27</ServiceDate>
  <ServiceType>Battery Check</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>ModelQ</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="9">
  <ServiceDate>2024-02-28</ServiceDate>
  <ServiceType>Tire Rotation</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>DemoModelZ</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="10">
  <ServiceDate>2024-02-29</ServiceDate>
  <ServiceType>Generator Inspection</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
      <Model>DemoModel3</Model>
    </RecreationalVehicle>
  </RecreationalVehicle>
</RVServiceAppointment>
<RVServiceAppointment AppointmentID="11">
  <ServiceDate>2024-03-01</ServiceDate>
  <ServiceType>Awning Repair</ServiceType>
  <RecreationalVehicle>
    <RecreationalVehicle>
```

```
        <Model>DemoModel9</Model>
      </RecreationalVehicle>
    </RecreationalVehicle>
  </RVServiceAppointment>
</RVServiceAppointments>';
BEGIN
  ret :=
DBMS_XDB.CREATERESOURCE('/public/grp3_rv_2024/RVServiceAppointm
ent.xml', RVServiceApptXML);
END;
/

--Supplier
DECLARE
  ret BOOLEAN;
  SupplierXML VARCHAR2(4000) :=
    '<?xml version="1.0"?>
Supplier Name="RV Service Supplier">
  <Address>789 Oak St, Anycity, USA</Address>
  <PhoneNumber>555-987-6543</PhoneNumber>
  <RV Make="Brand1" Model="Model1">
    <Year>2024</Year>
    <Price>50000</Price>
    <Availability>Available</Availability>
    <Condition>New</Condition>
  </RV>
  <RV Make="Brand3" Model="ModelZ">
    <Year>2024</Year>
    <Price>36372</Price>
    <Availability>Available </Availability>
    <Condition>New</Condition>
  </RV>
  <RV Make="Brand3" Model="Model1">
    <Year>2024</Year>
    <Price>56102</Price>
    <Availability>Unavailable </Availability>
    <Condition>New</Condition>
  </RV>
  <RV Make="Brand2" Model="Model2">
    <Year>2024</Year>
    <Price>45000</Price>
    <Availability>Available</Availability>
    <Condition>New</Condition>
  </RV>
</Supplier>';
BEGIN
  ret := DBMS_XDB.CREATERESOURCE('/public/grp3_rv_2024/Supplier.xml',
SupplierXML);
END;
/

--Payment
DECLARE
  ret BOOLEAN;
  PaymentXML VARCHAR2(4000) :=
    '<?xml version="1.0" standalone="yes"?>
<Payment_Details>
  <Order>
    <Customer_ID>2</Customer_ID>
    <Order_ID>2</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <Amount>7000</Amount>
    <Payment_Date>2024-03-23</Payment_Date>
  </Order>
  <Order>
    <Customer_ID>1</Customer_ID>
    <Order_ID>1</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <Amount>5000</Amount>
    <Payment_Date>2024-03-23</Payment_Date>
  </Order>
  <Order>
    <Customer_ID>4</Customer_ID>
    <Order_ID>4</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <Amount>6000</Amount>
    <Payment_Date>2024-03-23</Payment_Date>
  </Order>
  <Order>
    <Customer_ID>3</Customer_ID>
    <Order_ID>3</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <Amount>4500</Amount>
    <Payment_Date>2024-03-23</Payment_Date>
  </Order>
</Payment_Details>';
BEGIN
  ret := DBMS_XDB.CREATERESOURCE('/public/grp3_rv_2024/Payment.xml',
PaymentXML);
END;
/

--RV
```

```
DECLARE
  ret BOOLEAN;
  RVXML VARCHAR2(4000) :=
    '<?xml version="1.0" standalone="yes"?>
<RV_Conditions>
  <Condition Type="Demo">
    <Total_Available>2</Total_Available>
    <RVs>
      <RV>
            <ID>9</ID>
            <Make>DemoZ</Make>
            <Model>DemoModelZ</Model>
            <Year>2023</Year>
            <Price>31000</Price>
            <Availability>Available</Availability>
      </RV>
      <RV>
            <ID>12</ID>
            <Make>Demo4</Make>
            <Model>DemoModel4</Model>
            <Year>2022</Year>
            <Price>43000</Price>
            <Availability>Available</Availability>
      </RV>
    </RVs>
  </Condition>
  <Condition Type="New">
    <Total_Available>2</Total_Available>
    <RVs>
      <RV>
            <ID>1</ID>
```

```
          <Make>Brand1</Make>
          <Model>Model1</Model>
          <Year>2024</Year>
          <Price>50000</Price>
          <Availability>Available</Availability>
      </RV>
      <RV>
          <ID>2</ID>
          <Make>Brand2</Make>
          <Model>Model2</Model>
          <Year>2024</Year>
          <Price>45000</Price>
          <Availability>Available</Availability>
      </RV>
    </RVs>
</Condition>
<Condition Type="Used">
    <Total_Available>2</Total_Available>
    <RVs>
      <RV>
          <ID>5</ID>
          <Make>Brand1</Make>
          <Model>Model2</Model>
          <Year>2023</Year>
          <Price>26382</Price>
          <Availability>Available</Availability>
      </RV>
      <RV>
          <ID>6</ID>
          <Make>Brand2</Make>
          <Model>ModelX</Model>
```

```
            <Year>2022</Year>
            <Price>26832</Price>
            <Availability>Available</Availability>
      </RV>
    </RVs>
  </Condition>
</RV_Conditions>';
BEGIN
  ret := DBMS_XDB.CREATERESOURCE('/public/grp3_rv_2024/RV.xml',
RVXML);
END;
/

--Customer
DECLARE
  ret BOOLEAN;
  CustomerXML VARCHAR2(4000) :=
    '<?xml version="1.0" standalone="yes"?>
<Customer_Details>
  <RV_Order>
    <Order_ID>2</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <First_Name>Bob</First_Name>
    <Last_Name>Smith</Last_Name>
    <Order_Total>22500</Order_Total>
  </RV_Order>
  <RV_Order>
    <Order_ID>3</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <First_Name>Carol</First_Name>
    <Last_Name>Williams</Last_Name>
    <Order_Total>22500</Order_Total>
  </RV_Order>
  <RV_Order>
    <Order_ID>4</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <First_Name>David</First_Name>
    <Last_Name>Brown</Last_Name>
    <Order_Total>22500</Order_Total>
  </RV_Order>
  <RV_Order>
    <Order_ID>1</Order_ID>
    <Order_Date>2024-03-23</Order_Date>
    <First_Name>Alice</First_Name>
    <Last_Name>Johnson</Last_Name>
    <Order_Total>22500</Order_Total>
  </RV_Order>
</Customer_Details>';
BEGIN
  ret :=
DBMS_XDB.CREATERESOURCE('/public/grp3_rv_2024/Customer.xml',
CustomerXML);
END;
/

--Employee
DECLARE
  ret BOOLEAN;
  EmployeeXML VARCHAR2(4000) :=
    '<?xml version="1.0" standalone="yes"?>
<Employees>
  <Employee EmployeeID="1">
```

```
    <Name>John Doe</Name>
    <Position>Manager</Position>
    <Contact>1234567890</Contact>
    <EmployeeID>1</EmployeeID>
  </Employee>
  <Employee EmployeeID="2">
    <Name>Jane Smith</Name>
    <Position>Salesperson</Position>
    <Contact>9876543210</Contact>
    <EmployeeID>2</EmployeeID>
  </Employee>
  <Employee EmployeeID="3">
    <Name>Michael Johnson</Name>
    <Position>Technician</Position>
    <Contact>5554443333</Contact>
    <EmployeeID>3</EmployeeID>
  </Employee>
  <Employee EmployeeID="4">
    <Name>Emily Davis</Name>
    <Position>Receptionist</Position>
    <Contact>1112223333</Contact>
    <EmployeeID>4</EmployeeID>
  </Employee>
</Employees>';
BEGIN
  ret :=
DBMS_XDB.CREATERESOURCE('/public/grp3_rv_2024/Employee.xml',
EmployeeXML);
END;
/
```

5. For remaining 15 requests formulate queries in XQuery language and execute the queries against your files in XML DB repository:

1. For all queries use clause ‘for’;
2. All queries are to use selection conditions formulated under ‘where’ clause or as predicates;
3. For at least five queries use qualification conditions specified on elements;
4. For at least five queries use qualification conditions specified on tag attributes;
5. Five queries must be formulated using more than one XML file from Oracle XML

   DB repository;

6. At least five queries must return values of XML elements without XML tags.

1. Retrieve the appointment IDs, service types, and RV models of all RV service appointments where the service date is after February 25, 2024.

```
xquery
let $doc := doc("/public/grp3_rv_2024/RVServiceAppointment.xml")
for $appointment in
$doc/RVServiceAppointments/RVServiceAppointment[@AppointmentID > 5]
where xs:date($appointment/ServiceDate) > xs:date("2024-02-25")
return
   element Appointment {
      attribute AppointmentID {$appointment/@AppointmentID},
      element ServiceType {$appointment/ServiceType},
      element RVModel {$appointment/RecreationalVehicle/RecreationalVehicle/Model}
   }
/
```

Requirements 3 is met in the provided XQuery.

2. Retrieve customer orders along with their details (order ID, order date, customer's first and last name, and order total) and the corresponding service types of RV service appointments.

```
xquery
for $customer in doc("/public/grp3_rv_2024/Customer.xml")/Customer_Details/RV_Order
let $order_id := $customer/Order_ID/text()
let $order_date := $customer/Order_Date/text()
let $first_name := $customer/First_Name/text()
let $last_name := $customer/Last_Name/text()
let $order_total := $customer/Order_Total/text()
for $service in
doc("/public/grp3_rv_2024/RVServiceAppointment.xml")/RVServiceAppointments/RVService
Appointment
let $service_type := $service/ServiceType/text()
where $service/@AppointmentID = $order_id
return
   element CustomerOrder {
      attribute OrderID {$order_id},
      attribute OrderDate {$order_date},
      element Customer {
         element FirstName {$first_name},
         element LastName {$last_name}
      },
      element OrderTotal {$order_total},
      element ServiceType {$service_type}
   }
/
```

Requirements 3, 4 and 5 are met in the provided XQuery.

3. Retrieve the names and positions of employees whose contact number starts with '555' and whose EmployeeID is greater than 2.

```
xquery
let $employeeDoc := doc("/public/grp3_rv_2024/Employee.xml")
for $employee in $employeeDoc//Employee
where starts-with($employee/Contact/text(), '555') and $employee/@EmployeeID > 2
return concat("Employee Name: ", $employee/Name/text(), ", Position: ",
$employee/Position/text())
/
```

Requirements 4 and 6 are met in the provided XQuery.

4. Retrieve the total available RVs along with their makes, models, and prices for RVs that are available and have a price greater than $40,000.

```
xquery
let $doc := doc('/public/grp3_rv_2024/RV.xml')
for $rv in $doc/RV_Conditions/Condition/RVs/RV
where $rv/Availability = 'Available' and number($rv/Price) > 40000
return
  concat("Make: ", $rv/Make/text(), ", Model: ", $rv/Model/text(), ", Price: ", $rv/Price/text(), ",
Availability: ", $rv/Availability/text())
/
```

Requirements 6 is met in the provided XQuery.

5. Retrieve the names and positions of employees whose contact number starts with '55' and whose EmployeeID is greater than 1 and who are involved in servicing RVs with Model1 or Model2.

```
    xquery
    let $employeeDoc := doc("/public/grp3_rv_2024/Employee.xml")
```

```
let $rvDoc := doc("/public/grp3_rv_2024/RV.xml")
for $employee in $employeeDoc/Employees/Employee
let $rvModels := $rvDoc/RV_Conditions/Condition[@Type = "New"]/RVs/RV/Model
let $model1 := $rvModels[text()='Model1']
let $model2 := $rvModels[text()='Model2']
where $employee/Name and $employee/Position and ($model1 or $model2) and starts-
with($employee/Contact, '55') and $employee/@EmployeeID > 1
return
  element Employee {
    attribute EmployeeID {$employee/@EmployeeID},
    element Name {$employee/Name},
    element Position {$employee/Position}
  }
/
```

Requirements 3, 4, and 5 are met in the provided XQuery.

6. Retrieve the first names and last names of customers who placed orders with a total amount greater than $20,000 and the RV models of the RVs they ordered.

```
xquery
let $customer_doc := doc('/public/grp3_rv_2024/Customer.xml')
for $customer in $customer_doc/Customer_Details/RV_Order
let $order_id := $customer/Order_ID/text()
let $order_total := xs:decimal($customer/Order_Total)
where $order_total > 20000
return
  element Customer {
    element First_Name {data($customer/First_Name/text())},
    element Last_Name {data($customer/Last_Name/text())},
    element RV_Models {
      for $order in
doc('/public/grp3_rv_2024/RVServiceAppointment.xml')/RVServiceAppointments/RVSe
rviceAppointment[@AppointmentID = $order_id]
      where $order/@AppointmentID = $customer/Order_ID
      return
        element RV_Model {
          attribute OrderID {$order_id},
          attribute ServiceDate {$order/ServiceDate/text()},
          element Model
{data($order/RecreationalVehicle/RecreationalVehicle/Model/text())}
        }
    }
  }
/
```

Requirements 3, 4, and 5 are met in the provided XQuery.

7. Retrieve the customer IDs, payment dates, and payment amounts for all payments where the payment amount exceeds $5500.

```
xquery
for $payment in doc('/public/grp3_rv_2024/Payment.xml')/Payment_Details/Order
where $payment/Amount > 5500
return
  concat("Customer ID: ", $payment/Customer_ID/text(), ", Payment Date: ",
$payment/Payment_Date/text(), ", Payment Amount: ", $payment/Amount/text())
/
```

Requirement 6 is met in the provided XQuery.

8. Retrieve the employee details, including name, position, and contact information, along with the first name and last name of the customer who placed an order with order ID 4, for the employee with EmployeeID 4 whose contact number ends with '333'.

```
xquery
let $employeeDoc := doc('/public/grp3_rv_2024/Employee.xml')
let $customerDoc := doc('/public/grp3_rv_2024/Customer.xml')
for $employee in $employeeDoc/Employees/Employee,
    $customer in $customerDoc/Customer_Details/RV_Order
where $employee/@EmployeeID = 4 and ends-with($employee/Contact/text(), '333') and
$customer/Order_ID = 4
return
    <Employee>
        <Name>{data($employee/Name)}</Name>
        <Position>{data($employee/Position)}</Position>
        <Contact>{data($employee/Contact)}</Contact>
        <Customer>
            <First_Name>{data($customer/First_Name)}</First_Name>
            <Last_Name>{data($customer/Last_Name)}</Last_Name>
        </Customer>
    </Employee>
/
```

Requirements 4 and 5 are met in the provided XQuery.

9. Retrieve the service types and RV models for all RV service appointments where the RV model is either Model1 or Model2 and the service date is before February 28, 2024.

```
xquery
for $appt in
doc("/public/grp3_rv_2024/RVServiceAppointment.xml")//RVServiceAppointment
where (substring($appt/RecreationalVehicle/RecreationalVehicle/Model/text(), 1, 6) =
"Model1" or substring($appt/RecreationalVehicle/RecreationalVehicle/Model/text(), 1,
6) = "Model2") and xs:date($appt/ServiceDate/text()) < xs:date("2024-02-28")
return
concat("Service Type: ", $appt/ServiceType/text(), ", RV Model: ",
$appt/RecreationalVehicle/RecreationalVehicle/Model/text())
/
```

Requirement 6 is met in the provided XQuery.

10. Retrieve the names and positions of employees whose contact number contains '987' and who are involved in servicing RVs with ModelX.

```
xquery
for $employee in
doc("/public/grp3_rv_2024/Employee.xml")/Employees/Employee[contains(Contact,
'987')]
let $employeeID := $employee/@EmployeeID
for $rvAppointment in
doc("/public/grp3_rv_2024/RVServiceAppointment.xml")/RVServiceAppointments/RVS
erviceAppointment[RecreationalVehicle/RecreationalVehicle/Model = 'ModelX']
where $rvAppointment/ServiceDate >= xs:date("2024-02-25")
return
   element Employee {
      attribute EmployeeID { $employee/@EmployeeID },
      element Name { data($employee/Name/text()) },
      element Position { data($employee/Position/text()) }
   }
```

```
/
```

Requirements 3 and 5 are met in the provided XQuery.

11. Retrieve the total available RVs along with their makes, models, and prices for RVs that are available and have a price greater than $44,000.

```
xquery
let $rv_doc := doc('/public/grp3_rv_2024/RV.xml')
for $rv in $rv_doc//RVs/RV
where $rv/parent::RVs/parent::Condition[@Type='New']/RVs/RV[Price > 44000]
return (
  concat("Make: ", $rv/Make/text(), ", Model: ", $rv/Model/text(), ", Price: ",
$rv/Price/text(), "&#10;")
)
/
```

Requirements 6 is met in the provided XQuery.

12. What are the service type and recreational vehicle model for the RV service appointment with AppointmentID 4, where the RV model is 'ModelZ'.

```
xquery
for $appointment in
doc('/public/grp3_rv_2024/RVServiceAppointment.xml')/RVServiceAppointments/RVSe
rviceAppointment
let $rvModel := $appointment/RecreationalVehicle/RecreationalVehicle/Model/text()
where $appointment/@AppointmentID = 4 and $rvModel = 'ModelZ'
return <Appointment>
```

```
        <ServiceType>{$appointment/@ServiceType}</ServiceType>
        <RVModel>{$rvModel}</RVModel>
      </Appointment>
/
```

Requirements 4 is met in the provided XQuery.

13. Retrieve the names and positions of employees whose contact number contains '32' and who are involved in servicing RVs with Model2.

```
xquery
for $employee in doc('/public/grp3_rv_2024/Employee.xml')/Employees/Employee
where contains($employee/Contact/text(), '32') and
exists(doc('/public/grp3_rv_2024/RVServiceAppointment.xml')/RVServiceAppointments
/RVServiceAppointment[RecreationalVehicle/RecreationalVehicle/Model/text() =
'Model2'])
return
  <EmployeeDetails>
    <Name>{data($employee/Name)}</Name>
    <Position>{data($employee/Position)}</Position>
  </EmployeeDetails>
/
```

14. Retrieve the service dates and service types of all RV service appointments where the service type is either "Annual Maintenance" or "Repair" and the service date is before March 7, 2024.

```
xquery
for $appt in
doc("/public/grp3_rv_2024/RVServiceAppointment.xml")/RVServiceAppointments/RVS
erviceAppointment
```

```
where ($appt/ServiceType = "Annual Maintenance" or $appt/ServiceType = "Repair")
and xs:date($appt/ServiceDate) < xs:date("2024-03-07")
return concat("Service Date: ", $appt/ServiceDate/text(), ", Service Type: ",
$appt/ServiceType/text())
/
```

15. Get all models and service dates where service type is “Oil Change”.

```
xquery
for $appt in
doc("/public/grp3_rv_2024/RVServiceAppointment.xml")//RVServiceAppointment
where $appt/ServiceType = 'Oil Change'
return
   concat("Model: ", data($appt/RecreationalVehicle/RecreationalVehicle/Model),
", Service Date: ", data($appt/ServiceDate))
/
```

16. Retrieve the first names, last names, and order dates for all RV orders placed by customers whose last name starts with 'W' and the order date is after February 25, 2024.

```
xquery
for $customer in
doc("/public/grp3_rv_2024/Customer.xml")/Customer_Details/RV_Order[starts-
with(Last_Name, 'W')]
let $orderID := $customer/Order_ID
let $orderDate := $customer/Order_Date
let $serviceType :=
doc("/public/grp3_rv_2024/RVServiceAppointment.xml")/RVServiceAppointments/RVS
erviceAppointment[./@AppointmentID = $orderID]/ServiceType
where xs:date($orderDate) gt xs:date("2024-02-25")
return
   <CustomerOrder>
      <FirstName>{data($customer/First_Name)}</FirstName>
      <LastName>{data($customer/Last_Name)}</LastName>
      <OrderDate>{data($orderDate)}</OrderDate>
```

```
    </CustomerOrder>
/
```