

1. Lab 10 – Complex Data Types: Semi-Structured Data

Objectives

- Understand flexible, schema-less data (JSON, XML, RDF).
- Create and query JSON data in SQL.
- Generate and query XML documents.
- Represent facts as RDF triples and simulate SPARQL queries.
- Semi-structured data allows **varying attributes** per record.
- **JSON** – key-value pairs, nested objects, arrays.
- **XML** – hierarchical tags for structured documents.
- **RDF** – triples (**subject, predicate, object**) forming a graph.
- **SPARQL** – query language for RDF, analogous to SQL.

2. Part 1 – JSON in SQL Databases (6 pts)

Task 1 – Create Table

```
CREATE TABLE students (  
    id SERIAL PRIMARY KEY,  
    profile JSON  
);
```

Task 2 – Insert JSON Data

```
INSERT INTO students (profile)  
VALUES ('{  
    "name": {"first": "Ali", "last": "Karimov"},  
    "dept": "Computer Science",  
    "skills": ["SQL", "Python", "C++"],  
    "gpa": 3.8  
}');
```

3. Part 1 – Querying JSON Attributes

```
SELECT  
    profile -> 'name' ->> 'first' AS firstname,  
    profile ->> 'dept' AS department,  
    profile ->> 'gpa' AS gpa  
FROM students;
```

Deliverables

- Screenshot of table creation and query output.
- Short explanation: How JSON differs from fixed columns.

4. Part 2 – XML Generation and Querying (4 pts)

Task 1 – Generate XML from Query

```
SELECT
  XMLELEMENT(NAME "student",
    XMLFOREST(id AS "id", profile AS "profile"))
FROM students;
```

Task 2 – Query XML using XPath

```
SELECT
  EXTRACTVALUE(
    XMLTYPE('<student>
      <name>Ali Karimov</name>
      <gpa>3.8</gpa>
    </student>'),
    '/student/gpa') AS gpa_value
FROM dual;
```

5. XML Deliverables and Reflection

- Output of XML generation and XPath query.
- Short paragraph: When is XML preferred over JSON?
 - Supports ordered, mixed, and self-describing data.
 - Widely used for configuration files and document-centric systems.

6. Part 3 – RDF and Knowledge Graphs (5 pts)

Task 1 – Represent Data as Triples

```
CREATE TABLE rdf_data (
  subject  VARCHAR(50),
  predicate VARCHAR(50),
  object   VARCHAR(50)
);

INSERT INTO rdf_data VALUES
('10101', 'name', 'Srinivasan'),
('10101', 'teaches', 'sec1'),
('sec1', 'course', 'Database Systems');
```

7. Query RDF Triples and Simulate SPARQL

Task 2 – Query Triples

```
SELECT subject, object
FROM rdf_data
WHERE predicate = 'teaches';
```

Task 3 – SPARQL-Like Join Query

```
SELECT t1.subject, t3.object AS course
FROM rdf_data t1
JOIN rdf_data t2 ON t1.object = t2.subject
JOIN rdf_data t3 ON t2.subject = t3.subject
WHERE t1.predicate = 'teaches'
      AND t3.predicate = 'course';
```

Deliverables: RDF table output + query results + optional graph diagram.

8. Submission & Evaluation

Submit:

- A single PDF/Word report containing
 - SQL code and screenshots of results.
 - Explanations for each task.