OCR Test – DBMS (Database Management Systems)

Purpose: This PDF is designed to test OCR accuracy on mixed DBMS content including text, tables, SQL, relational algebra, and pseudo-diagrams.

Instructions: Scan or upload this PDF into your OCR pipeline and verify text extraction accuracy.

1) Plain Text (Theory)

A Database Management System (DBMS) is software that enables users to define, create, maintain, and control access to databases. Key properties include data abstraction, data independence, concurrency control, recovery, security, and integrity constraints. The CAP theorem states that in a distributed system, it is impossible to simultaneously guarantee Consistency, Availability, and Partition Tolerance.

2) Table (Relational Data)

StudentID	Name	Department	Semester	CGPA
UGS24101	Aarav	IT	4	8.9
UGS24107	Ayush	IT	4	9.2
UGS24115	Riya	AI	4	8.5
UGS24122	Kiran	CSE	4	8.7

3) SQL Queries

```
CREATE TABLE Students (
StudentID VARCHAR(10) PRIMARY KEY,
Name VARCHAR(50) NOT NULL,
Department VARCHAR(10),
Semester INT CHECK (Semester BETWEEN 1 AND 8),
CGPA DECIMAL(3,2)
);

SELECT Name, Department
FROM Students
WHERE CGPA > 8.5
ORDER BY CGPA DESC
LIMIT 2;

CREATE TABLE Enrollments (
StudentID VARCHAR(10),
CourseID VARCHAR(10),
FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

4) Relational Algebra

Find departments with budget > 50000:
σ_{budget > 50000}(Department)

Find students who have taken at least one course from every department:
π_{student_id}(Takes ÷ π_{dept_name}(Department))

Number of classrooms in IT department:
COUNT(σ_{dept_name = 'IT'}(Classroom))

5) ER Diagram (ASCII-style)

```
[STUDENT] (StudentID, Name, Dept)
|
ENROLLS
|
[COURSE] (CourseID, Title, Credits)

Participation:
- STUDENT has total participation in ENROLLS
- COURSE has partial participation in ENROLLS
```

6) Mixed Content (Edge Cases)

```
Foreign Key Syntax Example:
REFERENCES ParentTable(ParentID)

JSON Document Example:
{
"_id": "507f1f77bcf86cd799439011",
"name": "DBMS Notes",
"tags": ["SQL", "Normalization", "Indexes"],
"rating": 4.8
}

Index Query:
db.notes.find().sort({ rating: -1 }).limit(2)
```