BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

WORK INTEGRATED LEARNING PROGRAMMES

COURSE HANDSON LAB ASSIGNMENT HANDOUT

M.Tech

Course Title	Database Systems and Applications
Course No(s)	SESAP ZC337
Credit Units	4 Credits, 24 Hours optimized delivery
Lab Session: Nov-Dec 25	Demonstration of the solution to given problem statement

Faculty: Balachandra A, Guest Faculty, BITS Pilani (WILP) Division
Email: balachandra.ananatharamaiah@wilp.bits-pilani.ac.in
Mob: 9113656626 / 9480475967

Contents

SAMPLE REFERECE:

SOFTWARE REQUIREMENTS SPECIFICATION (SRS) DOCUMENT

For COMPANY Database System PROJECT ASSIGNMENT - DBSA

Version: <#>
Prepared by: <Name, ID>
Date: <Date>

---

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) defines the requirements for the **COMPANY Database System**, designed to manage employee, department, project, and dependent information with **advanced normalization** (up to 6NF) and **optimized join operations**. The document specifies all functional and non-functional requirements, database schema constraints, and user interactions. It serves as a reference for database developers and system evaluators. The system supports efficient data access, analytical queries, and integrity enforcement across interconnected entities such as employees, departments, projects, and dependents.

1.2 Scope

The COMPANY Database System automates data management for a company's core operations.
Functions include:

- Maintaining employee personal and employment details.
- Tracking departmental structure and managers.
- Recording projects and their associated departments.
- Tracking employee participation in projects and dependents.
- Database is a normalized relational system supporting:
  - Employee, department, project, and dependent management.
  - Recursive relationships (supervision).
  - Multi-entity joins and derived reporting.
  - The system ensures **data integrity, referential consistency, and normalization (3NF/BCNF)**
  - Data models optimized through **5NF and 6NF** to ensure minimal redundancy and fast join performance.

1.3 Definitions, Acronyms, and Abbreviations

Term	***Meaning***
DBMS	*Database Management System*
SRS	*Software Requirements Specification*
DDL	*Data Definition Language*
DML	*Data Manipulation Language*
PK	*Primary Key*
FK	*Foreign Key*
SQL	*Structured Query Language*

1.4 References

- IEEE 830–1998, ISO/IEC/IEEE 29148:2018
- Elmasri & Navathe, *Fundamentals of Database Systems*, 7th Ed.

2. Overall Description

2.1 Product Perspective

The COMPANY Database System is a **relational database** developed using SQL and runs under any standard RDBMS (e.g., Oracle, MySQL, PostgreSQL).
It is a central data store for all company divisions and will be integrated with future applications such as payroll or project tracking tools.

2.2 Product Functions

- Manage employees and departments.
- Assign managers to departments.
- Record projects and link them to departments.
- Track which employees work on which projects.
- Record employee dependents.
- Generate summary and analytical queries.

2.3 User Characteristics

User Role	Description
Administrator	Defines database schema, manages users, backups
HR Manager	Maintains employee, department, dependent data
Project Manager	Manages project details and employee assignments
Analyst	Generates reports and queries

2.4 Constraints

- The database must be in **Third Normal Form (3NF)** or higher.
- All referential integrity constraints must be enforced through FKs.
- Access control must restrict schema modification to the Administrator.
- Must use standard SQL DDL/DML commands.

2.5 Assumptions and Dependencies

- Each employee belongs to one department.
- Each department has exactly one manager.
- Each project belongs to one department.
- The system depends on an underlying SQL RDBMS environment.

2.6 The database uses progressive normalization up to 6NF to:

- Ensure atomic attribute dependencies.
- Eliminate redundancy and update anomalies.
- Improve consistency in join-based queries (especially analytical joins).

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement Description
FR1	The system shall store employee details (name, SSN, address, sex, salary, supervisor, department number).
FR2	The system shall store department details (name, number, manager SSN, manager start date).
FR3	The system shall store project details (name, number, location, department number).
FR4	The system shall record which employees work on which projects and the number of hours per week.
FR5	The system shall store dependent information for each employee.
FR6	The system shall enforce referential integrity between related entities.
FR7	The system shall support queries to list employees by department, project, or manager.
FR8	The system shall allow managers to update project assignments.
FR9	The system shall allow reports to be generated using joins and aggregate functions.
FR10	The system shall maintain the database in at least 5NF to ensure minimal redundancy and support efficient join decomposition.
FR11	The system shall support decomposition into 6NF where temporal or repeating data (e.g., salary history) is introduced.
FR12	The system shall execute multi-table joins for analytics, such as combining employee, department, and project data.
FR13	The system shall support view creation for join queries (e.g., EMP_PROJECT_VIEW) to simplify reporting.
FR14	The system shall enforce join integrity through constraints and triggers (e.g., preventing orphan join results).

3.2 Non-Functional Requirements

ID	Category	Requirement
NFR1	Performance	Queries on fewer than 1000 records should complete within 2 seconds.
NFR2	Security	Only authorized users may modify data.
NFR3	Reliability	The database shall prevent orphan records and null foreign keys.
NFR4	Scalability	The schema shall support growth up to 10,000 employees and 5,000 projects.
NFR5	Maintainability	Schema changes shall require minimal disruption.
NFR6	Backup & Recovery	Full backup and restore functions must be available.
NFR7	Performance	Multi-table joins involving ≤ 5 tables must execute under 3 seconds for ≤ 1000 records.
NFR8	Optimization	Use of 5NF and 6NF shall minimize redundancy while maintaining efficient join paths through indexed FKs.
NRFS	Join	<provide relevant data for join performance>

4. Database Design

4.1 Entity and Attribute Definitions

1. EMPLOYEE(Ssn, Fname, Minit, Lname, Address, Sex, Salary, Super_ssn, Dno)
2. DEPARTMENT(Dnumber, Dname, Mgr_ssn, Mgr_start_date)
3. PROJECT(Pnumber, Pname, Location, Dnum)
4. WORKS_ON(Essn, Pno, Hours)
5. DEPENDENT(Essn, Dependent_name, Sex, Birth_date, Relationship)

4.2 Relational Schema (DDL)

```
CREATE TABLE DEPARTMENT (
  Dnumber INT PRIMARY KEY,
  Dname VARCHAR(30) UNIQUE,
  Mgr_ssn CHAR(9),
  Mgr_start_date DATE,
  FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
);

CREATE TABLE EMPLOYEE (
  Ssn CHAR(9) PRIMARY KEY,
  Fname VARCHAR(15),
  Minit CHAR(1),
  Lname VARCHAR(15),
  Address VARCHAR(50),
  Sex CHAR(1),
  Salary DECIMAL(10,2),
  Super_ssn CHAR(9),
  Dno INT,
  FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
  FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
);

CREATE TABLE PROJECT (
  Pnumber INT PRIMARY KEY,
  Pname VARCHAR(30),
  Location VARCHAR(30),
  Dnum INT,
  FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber)
);

CREATE TABLE WORKS_ON (
  Essn CHAR(9),
  Pno INT,
  Hours DECIMAL(5,2),
  PRIMARY KEY (Essn, Pno),
  FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
  FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber)
);
```

```
CREATE TABLE DEPENDENT (
  Essn CHAR(9),
  Dependent_name VARCHAR(20),
  Sex CHAR(1),
  Birth_date DATE,
  Relationship VARCHAR(15),
  PRIMARY KEY (Essn, Dependent_name),
  FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn)
);
```

4.3 Normalization Process

Normal Form	Description	Achieved Purpose
1NF	Atomic attribute values; repeating groups removed.	Each column contains indivisible data.
2NF	Partial dependencies removed.	Non-key attributes depend on full PK.
3NF	Transitive dependencies removed.	No non-key attribute depends on another non-key.
BCNF	Functional dependencies refined.	Every determinant is a candidate key.
4NF	Multi-valued dependencies removed.	Independent multivalued facts isolated.
5NF	Join dependencies removed.	Table reconstructed only via lossless joins.
6NF	Temporal dependencies isolated.	Enables granular tracking (e.g., salary changes).

Result: Each table in COMPANY DB is in **5NF**, allowing optimal join-based query decomposition.

The **EMPLOYEE–WORKS_ON–PROJECT** join is lossless, maintaining full consistency.

4.4 Advanced Joins and Analytical Views (Level 7 Optimization)

Example Derived Views:

```
CREATE VIEW EMP_PROJECT_VIEW AS
SELECT E.Ssn, E.Fname, E.Lname, D.Dname, P.Pname, W.Hours
FROM EMPLOYEE E
JOIN WORKS_ON W ON E.Ssn = W.Essn
JOIN PROJECT P ON P.Pnumber = W.Pno
JOIN DEPARTMENT D ON D.Dnumber = E.Dno;
```

Example Analytical Queries:

- *Q1:* List total hours each employee worked per project.
- *Q2:* Find employees supervised by a manager working on the same project.
- *Q3:* Aggregate average salary per department.
- *Q4:* Identify departments managing projects at multiple locations.

Performance Optimization:

- Create **indexes** on all FK columns (e.g., `Essn`, `Pno`, `Dno`).
- Use **materialized views** for complex analytical joins.
- Employ **query rewriting** for automatic join optimization.

4.5 Table Definitions (unchanged core schema, with indexes)

```
CREATE INDEX idx_employee_dno ON EMPLOYEE(Dno);
CREATE INDEX idx_works_on_pno ON WORKS_ON(Pno);
CREATE INDEX idx_works_on_essn ON WORKS_ON(Essn);
CREATE INDEX idx_project_dnum ON PROJECT(Dnum);
```

These indexes support **high-performance joins** in analytical workloads.

5. System Features

5.1 Data Integrity

- Enforced through PK/FK constraints and not null fields.
- Self-referential integrity maintained via Super_ssn.

5.2 Normalization

- Schema is in **Third Normal Form (3NF)**:
  - Each attribute is fully dependent on the key.
  - No transitive dependencies exist.
- **Up to 5NF/6NF** ensures:
  - No redundancy in join dependencies.
  - Minimal anomalies during update/delete operations.
- Ensures scalability for advanced data models (e.g., future salary history, location tracking).

5.3 Reporting Queries (Examples)

- List all employees working on a specific project.
- Display department managers and their start dates.
- Show employees earning above average salary.

5.4 Effective Joins (Level 7 Feature)

- **Optimized join paths** through indexed FKs.
- **Predefined analytical views** for HR and project reporting.
- **Lossless decomposition** ensuring data integrity across join paths.

6. External Interface Requirements

Interface Type	Description
User Interface	SQL console or DBMS GUI (e.g., MySQL Workbench, Oracle SQL Developer).
Hardware	Standard server/workstation with RDBMS software installed.
Software	RDBMS (Oracle/MySQL/PostgreSQL).
Communication	Local DB connection (JDBC/ODBC).

Supports integration with **data analytics tools** (e.g., Power BI, Tableau, or SQL-based dashboards) for join-based queries.

7. Validation and Verification

Test ID	Description	Expected Result
T1	Insert an employee with a valid department ID	Employee record accepted
T2	Insert an employee with invalid Dno	Foreign key violation
T3	Query all employees in each department	Accurate results
T4	Delete a manager assigned to a department	Deletion restricted
T5	Insert dependent for non-existent employee	Operation rejected
T6	Verify all tables are in 5NF	No non-trivial join dependencies
T7	Test 6NF decomposition for salary history	Temporal table successfully isolated
T8	Execute 4-table join (EMPLOYEE–WORKS_ON–PROJECT–DEPARTMENT)	Query executes <3s for ≤1000 records
T9	Test view EMP_PROJECT_VIEW	Returns correct integrated data
T10	Add trigger validation	FK constraint and trigger maintain referential integrity

8. Appendix

8.1 Future Enhancements

- Add **spatial attributes** (e.g., project site geometry).
- Add **image storage** (BLOB for employee or project images).
- Implement **temporal tables** to leverage full 6NF decomposition.

8.2 Document History

Version	Date	Description
1.0	Nov 2025	Initial draft for normalized COMPANY DB

Compliance:
This enhanced SRS complies with **IEEE 830** and **ISO/IEC/IEEE 29148**, including:

- Structured requirement hierarchy
- Full traceability of normalization design
- Testability of join and view operations
- Integration readiness for analytical tools