**Normalization Report**

**Student Name: Ashish Pandya  
Course Name: Enterprise Data Architecture  
Date: 12/March/2025**

**Introduction**

This report explains the normalization process applied to the given database spreadsheet to achieve **Third Normal Form (3NF)**. The purpose of normalization is to **eliminate redundancy**, **ensure data integrity**, and **improve database efficiency** as the system scales (Elmasri & Navathe, 2020).

**Normalization Process**

**0NF (Unnormalized Form)**

The given spreadsheet consists of employee job and payroll data along with committee involvement. Issues in the **0NF** structure include:

* **Repeating groups** in the committee involvement section (employees can be part of multiple committees).
* **Multi-valued attributes** such as supervisors managing multiple departments.
* **Derived attributes** like "Person Hours Worked," which should not be stored directly in the database.
* **Data redundancy**, as job details (e.g., job code, position, pay rate) are repeated for every employee record (Connolly & Begg, 2015).

**1NF (First Normal Form)**

To bring the data into **1NF**:

* Ensured that each attribute contains atomic values (no multi-valued attributes).
* Separated repeating groups by creating a **Committees** table.
* Assigned a unique identifier (primary key) to each record in tables.

**Tables at 1NF:**

* **Employees (EmpId as PK):** Stores employee-specific details.
* **Jobs (JobCode as PK):** Stores job-related attributes.
* **Committees (CommitteeId as PK):** Stores unique committee names.
* **Employee\_Committees (EmpId, CommitteeId as composite PK):** Resolves many-to-many relationships.

**2NF (Second Normal Form)**

To achieve **2NF**, we removed **partial dependencies**:

* Employee details were **separated from job-related attributes** to prevent job-related redundancy.
* **Payroll details** were moved into a new table since they are dependent on both employee and pay period.
* **Supervisor information** was moved into a separate table, as a single supervisor manages multiple employees.

**Tables at 2NF:**

* **Employees (EmpId as PK):** Stores personal details.
* **Jobs (JobCode as PK):** Stores job positions and pay rates.
* **Payroll (PayrollId as PK, EmpId as FK):** Tracks hours worked and overtime.
* **Supervisors (SupervisorId as PK):** Stores supervisor details.
* **Supervision (EmpId, SupervisorId as composite PK):** Resolves many-to-one employee-supervisor relationship.

**3NF (Third Normal Form)**

To achieve **3NF**, we eliminated **transitive dependencies**:

* "Person Hours Worked" is a **calculated field**, so it was removed from storage and will be dynamically computed via queries.
* **All non-key attributes now fully depend on the primary key**.

**Final 3NF Tables:**

1. **Employees (EmpId as PK):** Stores employee details.
2. **Jobs (JobCode as PK):** Stores job-related data.
3. **Payroll (PayrollId as PK, EmpId as FK):** Tracks pay periods and work hours.
4. **Supervisors (SupervisorId as PK):** Stores supervisor information.
5. **Supervision (EmpId, SupervisorId as composite PK):** Links employees to supervisors.
6. **Committees (CommitteeId as PK):** Stores committee names.
7. **Employee\_Committees (EmpId, CommitteeId as composite PK):** Links employees to committees.

**Addressing Assignment Concerns**

**Concern 1: Scaling the Payroll Data Efficiently**

* In **0NF**, payroll data was embedded within the employee records, causing redundancy as the number of pay periods grew.
* **Solution:** By creating a separate **Payroll** table, the database can handle an unlimited number of payroll records efficiently.

**Concern 2: Supervisors Managing Multiple Departments**

* Originally, supervisors were stored as a **single field** inside the employee table, making it hard to track multiple relationships.
* **Solution:** Created a separate **Supervisors** table and a **Supervision** relationship table to track which employees report to which supervisor.

**Concern 3: Employees in Multiple Committees**

* The original spreadsheet stored committee involvement in **repeating columns**.
* **Solution:** Created an **Employee\_Committees** table to handle the many-to-many relationship properly.

**Concern 4: "Person Hours Worked" as a Derived Attribute**

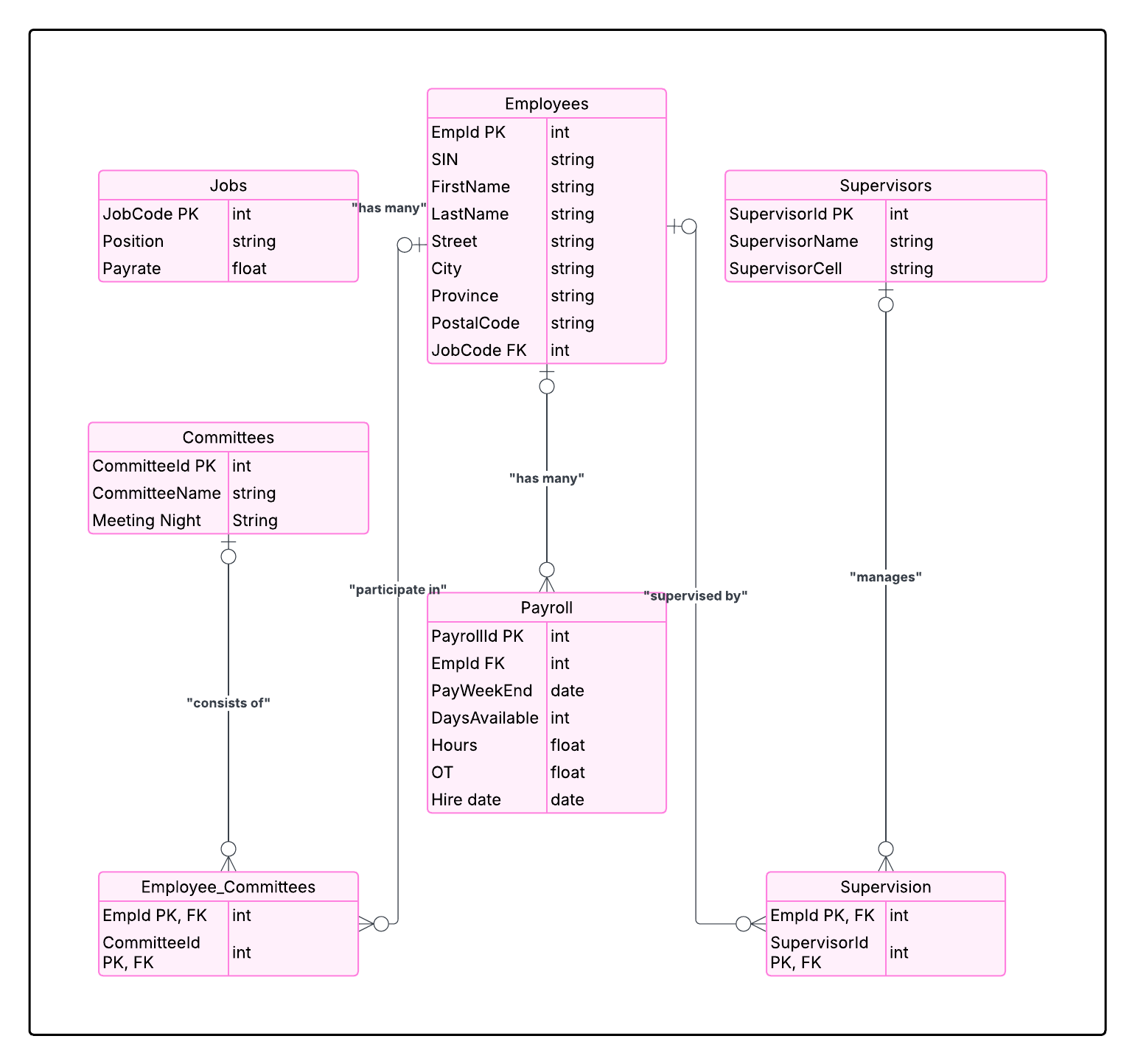
* "Person Hours Worked" was stored in the spreadsheet, but it is a **calculated value** (Hours + Overtime).
* **Solution:** Removed this field and instead compute it using an SQL **SUM() query** whenever needed.

**ERD Diagram: -**

**A computer screen shot of a computer

AI-generated content may be incorrect.**

**Crowfoot Notations: -**

****

**SQL Queries Section**

1. **Who worked more than 25 hours in any pay period and the employee’s last name include the character 's'? Please print the employee's [employee ID], [Full name] (Last name + First Name) as ‘Full Name’, [Supervisor Name], [Hours].**

A screenshot of a computer

AI-generated content may be incorrect.

1. **I want to enter an employee's ID and find their job code, manager’s name, and cell number. (No duplicate records)**

A screenshot of a computer

AI-generated content may be incorrect.

1. **Given a committee name 'OH&S', I’d like a list of people who work on it and their job description.**

**A computer screen shot of a computer

AI-generated content may be incorrect.**

1. **Given a supervisor's last name, e.g., ‘Muktadir’, provide a list of employees being supervised and their job codes.**

**A computer screen with text

AI-generated content may be incorrect.**

1. **Create a view to show the employee ID, employee’s full name, committee ID, meeting night, committee name, supervisor id and supervisor’s name, where the condition is the meeting night is ‘Tues’. No duplicate records.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Write a query to show each employee’s full name, and the number of years they are working in the company.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **List the number of employees under each supervisor. Using the ML Services and “sp\_execute\_external\_script” procedure, create a bar graph where the supervisor’s names will be on the X-axis and a number of employees managed on the Y-axis.**

A screenshot of a computer

AI-generated content may be incorrect.

**Graph Generated by Python Scripts :-**

A blue bar graph with black text

AI-generated content may be incorrect.

A graph with blue squares

AI-generated content may be incorrect.

1. **Write a query calculating the Person Hours Worked (as mentioned above) and total earnings in CAD using the Payrate CAD. Using the ML Services and “sp\_execute\_external\_script” procedure, convert each employee's total earnings to USD and send the employee name and USD earnings back to MS SQL Server OutputDataSet.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Using the ML Services and “sp\_execute\_external\_script” procedure, convert each employee's total earnings to USD and send the employee name and USD earnings back to MS SQL Server OutputDataSet.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Conclusion**

By normalizing the data to **3NF**, we have removed redundancy, improved scalability, and ensured data integrity. The final schema efficiently handles payroll records, multiple committee memberships, and supervisor relationships while eliminating derived data storage.

**References**

Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). Pearson.  
Elmasri, R., & Navathe, S. (2020). *Fundamentals of Database Systems* (7th ed.). Pearson.