Experiment No.2
Mapping ER/EER to Relational schema model.
Date of Performance:24/01/25
Date of Submission:31/01/25

Aim: Mapping ER/EER to Relational schema model.

Objective: objective of design is to generate a formal specification of the database schema

Theory: Mapping Rules

Step 1: Regular Entity Types
Create an *entity relation* for each strong entity type. Include all single-valued attributes. Flatten composite attributes. Keys become secondary keys, except for the one chosen to be the primary key.

Step 2: Weak Entity Types
Also create an entity relation for each weak entity type, similarly including its (flattened) single-valued attributes. In addition, add the primary key of each owner entity type as a foreign key attribute here. Possibly make this foreign key CASCADE.

Step 3: Binary 1:1 Relationship Types
Let the relationship be of the form [S]——<R>——[T].

1. **Foreign key approach**: The primary key of T is added as a foreign key in S. Attributes of R are moved to S (possibly renaming them for clarity). If only one of the entities has total participation it's better to call it S, to avoid null attributes. If neither entity has total participation nulls may be unavoidable. *This is the preferred approach in typical cases.*
2. **Merged relation approach**: Both entity types are stored in the same relational table, "pre-joined". If the relationship is not total both ways, there will be null padding on tuples that represent just one entity type. Any attributes of R are also moved to this table.
3. **Cross-reference approach**: A separate relation represents R; each tuple is a foreign key from S and a foreign key from T. Any attributes of R are also added to this relation. Here foreign keys should CASCADE.

Approach	Join cost	Null-storage cost
Foreign key	1	low to moderate
Merged relation	0	very high, unless both are total
Cross-reference	2	None

Step 4: Binary 1:N Relationship Types

Let the relationship be of the form [S]——^{N}<R>^{1}——[T]. The primary key of T is added as a foreign key in S. Attributes of R are moved to S. This is the foreign key approach. The

merged relation approach is not possible for 1:N relationships. (Why?) The cross-reference approach might be used if the join cost is worth avoid null storage.

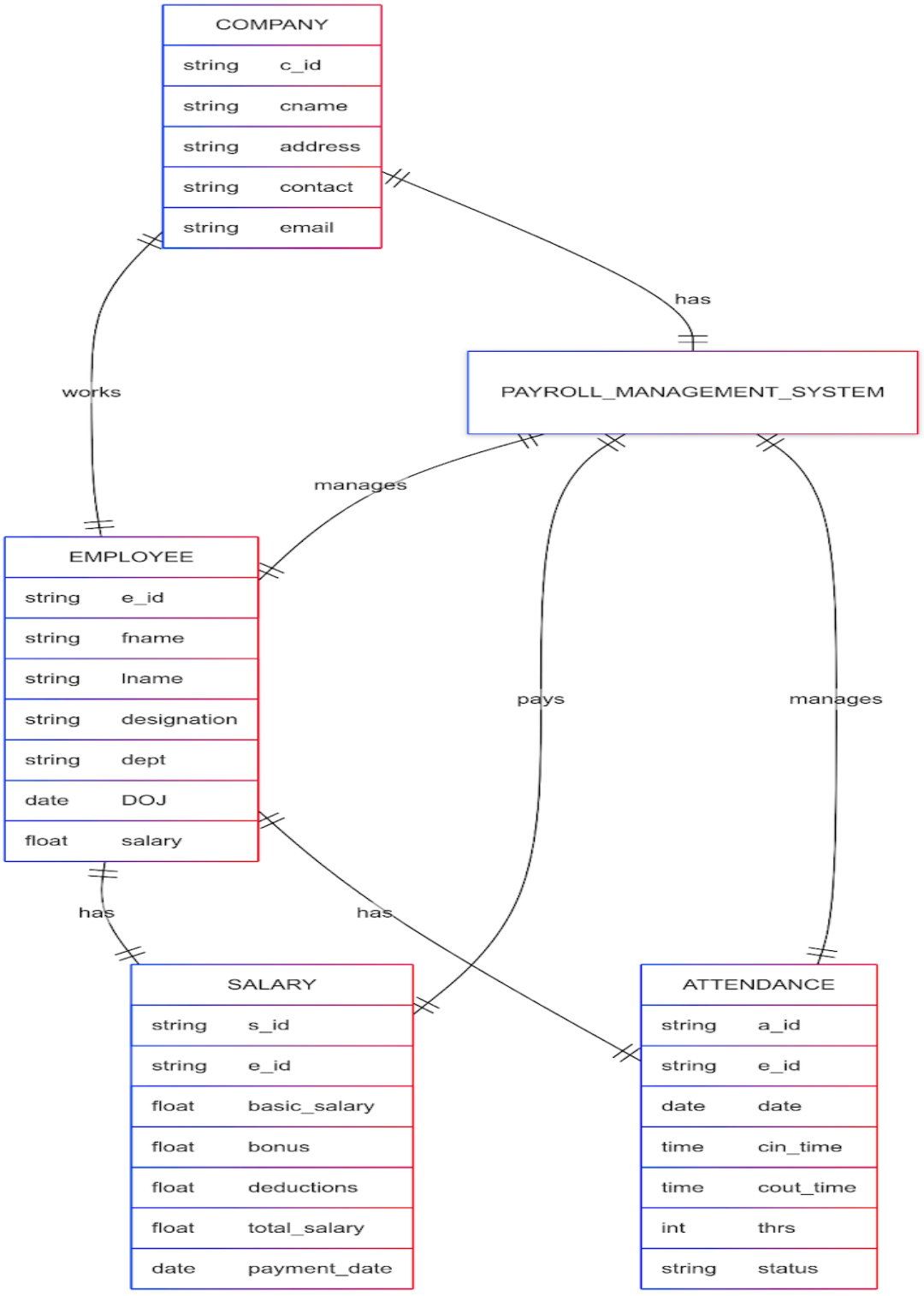
Step 5: Binary M:N Relationship Types
Here the cross-reference approach (also called a *relationship relation*) is the only possible way.

Step 6: Multivalued Attributes
Let an entity S have multivalued attribute A. Create a new relation R representing the attribute, with a foreign key into S added. The primary key of R is the combination of the foreign key and A. Once again this relation is dependent on an "owner relation" so its foreign key should CASCADE.

Step 7: Higher-Arity Relationship Types

Here again, use the cross-reference approach. For each n-ary relationship create a relation to represent it. Add a foreign key into each participating entity type. Also add any attributes of the relationship. The primary key of this relation is the combination of all foreign keys into participating entity types *that do not have a max cardinality of 1*.

Implementation:



Conclusion: The experiment successfully mapped ER/EER diagrams to relational schema models by applying structured mapping rules. This ensured a formal specification of the database schema, effectively capturing entity types, attributes, and relationships while minimizing redundancy and accommodating various participation constraints.