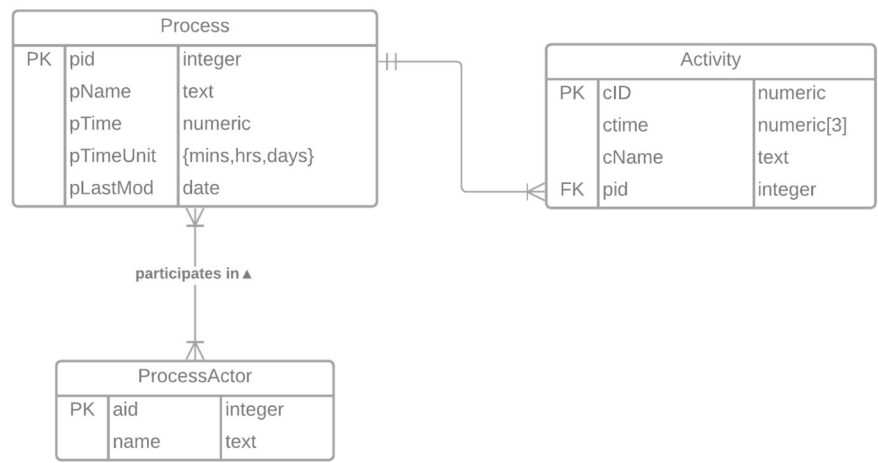
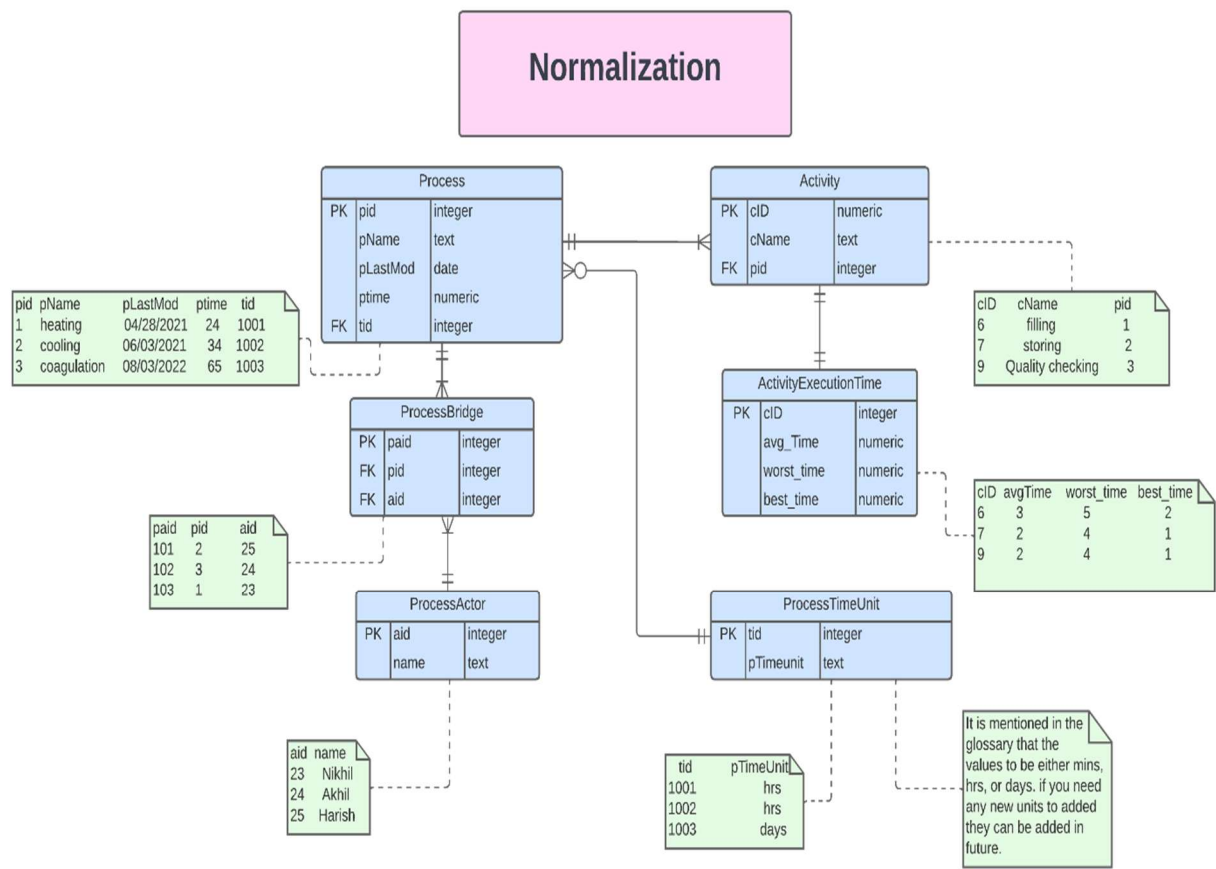


Proof Of Normalization

Given ERD



Normalized ERD



Resolving Many-many Relationship:

As we see a many-many relationship between process and process actors.

Original Functional dependencies of the Process Table:

pid-> pname, pid-ptime, pid->ptimeunit, pid -> pLastMod,p->aid

Original Functional dependencies of the ProcessActor Table:

aid->name

Modified Functional dependencies of the Process Table,ProcessActor Table, ProcessBridge table:

Process Table:

pid-> pname, pid-ptime, pid->tid, pid -> pLastMod

ProcessActor Table:

aid->name

ProcessBridge Table: paid->pid,
paid->aid

Hence, to decompose the many-many relationship I created a new table ProcessBridge table having a surrogate key "paid" as the primary key and the pid(process id) and aid(actor id) as foreign keys and I

mapped to both the tables with one to many relations to the ProcessBridge table.

Converting into 1NF:

The conditions required to be in its first Normal form is to have a single element in every table cell and to not have multivalued attributes, hence when I look at the ER diagram given, I have a multi-valued attribute "ctime" in the "Activity table" this violates the 1st Normal form and hence I need to decompose the attribute. With the information provided in the glossary, the values of the times were specified as the best, average, and worst execution time so,

Original Functional dependencies of the Activity Table:

cID -> cName, cID-> ctime[3], cID-> pid

Hence to decompose the multivalued attribute I have created a new table named Activity time keeping the primary key the "cID" itself and containing the attributes avg_time, worst_time, best_time of the data type numeric. So,

Modified functional dependencies will be: cID -> cName, cID -> avg_time, cID -> worst_time, cID -> best_time, cID -> pid

Converting to 2NF:

A database to be 2nd Normalization form it should be in 1NF and should have Single Column Primary Key that does not functionally dependent on any subset of candidate key relation, hence when seeing the ERD every class has unique primary key and is not dependent on any other candidate keys.

Converting to 3NF:

A database to be in the 3rd Normalization form should be in both 1NF and 2NF forms and should not have any transitive functional dependencies.

When I look into the ERD Process Table I could infer that the ptimeunit is dependent on the ptime as it violates the 3rd Normal form.

Original Functional dependencies of the Process Table:

pid-> pname, pid->ptime, pid->ptimeunit, ptime-> ptimeunit, pid -> pLastMod, pid->aid

I have created a new table processtimeunit where I can access the time unit. Where it contains tid as the primary key for the table and ptimeunit and I could see the ptime unit is in value set it should either be of the values mentioned in the glossary and so it should not be considered as a multivalued attribute. I would have a new foreign key in the process table which is tid as a reference to the processtimeunit table.

Now the modified functional dependencies of Process, Processtimeunit will be:

pid-> pname, pid->ptime, pid->tid, pid -> pLastMod, p->aid

tid-> timeunit