Lab Exercise 6 - Group 1

LIST OF BUSINESS RULES

Machine Record Management

1. machineID convention is “MC0001”
2. The 4 types of machines are PC, Switch, Router, and Server only
3. ‘Healthy’ status refers to no issues, and not in need of a patch
4. ‘Vulnerable’ status refers to issues present, needs patch
5. ‘Scheduled for Patch’ status refers to a machine that has been assigned for patch work
6. The status field can only be edited to ‘Healthy’ or ‘Vulnerable’ by the technician
7. The status field will automatically be set to “Scheduled for Patch” when the machine’s associated patch record is assigned to a technician through Transaction 3
8. Deletion will change the machine’s status to “Inactive”, this action is allowed only when the machine has no associated maintenance record with “Not Started” or “In progress” status.

Technician Record Management

1. technicianID convention is “TH0001”
2. Each technician’s email should be unique
3. Newly added technicians will have a default status field “Available”
4. The status field can not be edited directly by the technician, it modifies dynamically depending on the number of patch work assigned to the technician
5. Technician’s password should be at least 6 characters
6. ‘Desktop Support’ technicians will only work with PC machines
7. ‘Network Admin’ technicians will only work with Switch, and Router machines
8. ‘System Admin’ technicians will only work with Server machines
9. ‘Available’ status refers to Technician with less than 3 patch work assigned
10. ‘Unavailable’ status refers to Technician with exactly 3 patch work assigned
11. An “assigned patch work” refers to a maintenance record the technician is assigned to, with a “Not Started” or “In progress” status.
12. Maintenance record with “Done” status are not counted in the no. of assigned patch work of a technician
13. Deletion of a technician record will just change the status to “Inactive”, this can only be done if the technician has no “Not Started”, or “In progress” patch work
14. “Inactive” Technicians cannot login

Software Record Management

1. softwareID convention is “SF0001”
2. The 5 types of software are Application, System, Programming, Network, and Server only
3. ‘Application’, ‘System’, ‘Programming’ software is only for PC machines
4. ‘Network’ software is only for Router, and Switch machines
5. ‘Server’ software is only for Server machines

6. Deleting a software record refers to changing the status to “Inactive”. The status of the software record should also be able to be reactivated which would change the status back to “Active” at any time.

Tester Record Management

1. testerID convention is “TS0001”
2. Deleting a tester record would simply change its status to “Inactive”
3. “Inactive” Testers cannot login

Maintenance Record Management

1. maintenanceID convention is “MT0001”
2. The 2 types of maintenance work are Deploy, and Rollback only
3. The status field will either be “Not Started”, “In progress”, “Done”
4. Adding a new maintenance record is only done through Transaction 3
5. Editing a maintenance record is only done through Transaction 4, maintenance record with “Done” status cannot be edited
6. A maintenance record can only be deleted when the status is “Done”, deleting pertains it will no longer exist in the database

Patch Record Management

1. patchID convention is “PT0001”
2. The 5 types of patches are Application, System, Programming, Network, and Server only
3. Adding a new patch record is only done through Transaction 1
4. Each patch record must have a description
5. Deleting a patch record will just change the status to “Inactive”

Feedback Record Management

1. feedbackID convention is “FB0001”
2. Adding a new feedback record is only done through Transaction 2
3. Only testers are allowed to add, edit, delete feedback records
4. Deleting a feedback record refers to full deletion of the record, it will no longer exist in the database

Transaction 1: Release new patch

1. The patch type must be equal to the software type of the software record
2. The software type must correspond to the machine type of the machine record
3. The machine’s status should be ‘Vulnerable’
4. One technician can release multiple unique patches
5. A new patch record will have a default ‘New’ status

Transaction 2: Send feedback on patch

1. A tester can only provide feedback on a patch record with a corresponding maintenance record with “Done” status
2. One patch can receive multiple feedback from multiple testers

Transaction 3: Schedule patch work to technician

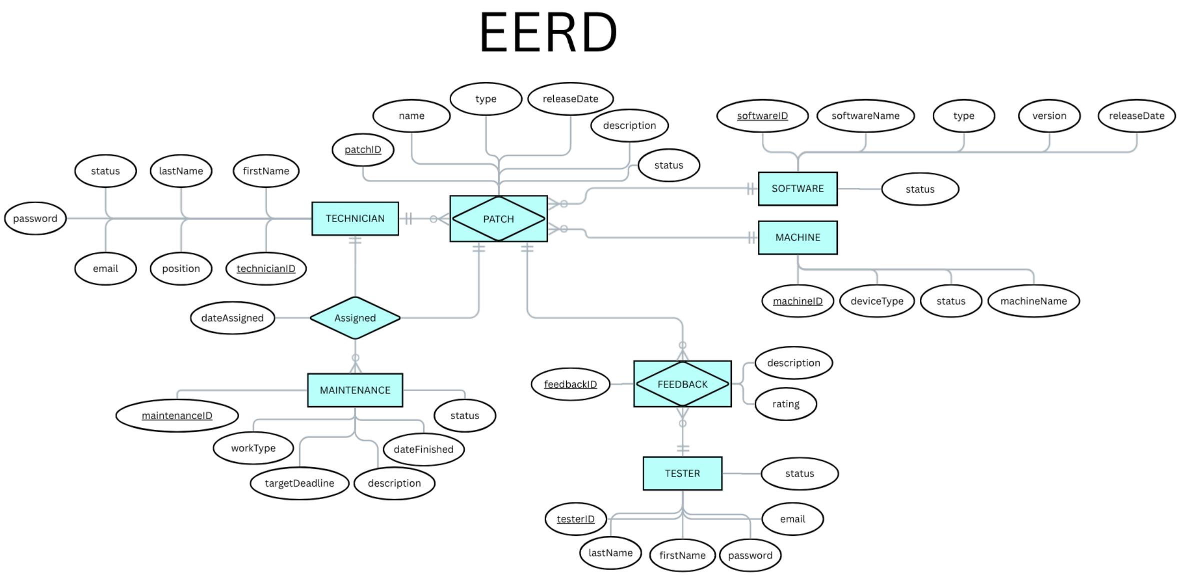
1. The Technician’s status must be ‘Available (less than 3 assigned tasks)
2. The Technician’s position must correspond to the type of machine assigned in the patch record (e.g., System Administrator -> Server).
3. For a “Deployment” work type, the assigned Patch must have either “New” or “Working” status
4. For a “Rollback” work type, the assigned Patch must have “Not Working” status
5. A patch work can only be assigned to 1 technician
6. A new maintenance record will have a default ‘Not Started’ status

Transaction 4: Update maintenance record

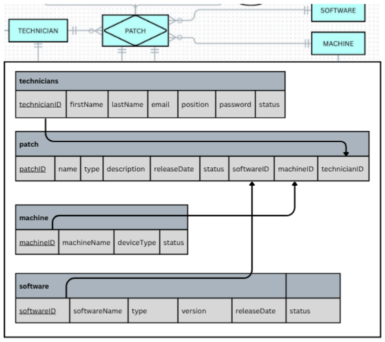
1. The technicianIDassigned is not editable, patch work can not be reassigned to another technician
2. If the patchID is edited, the Technician’s position should still correspond to the type of machine assigned in the patch record (e.g., System Administrator -> Server).
3. If the workType is edited to “Deployment”, the assigned Patch must have either “New” or “Working” status
4. If the workType is edited to “Rollback”, the assigned Patch must have “Not Working” status

Canva Link

ENHANCED ENTITY RELATIONSHIP DIAGRAM



TRANSFORMED RELATIONS



technicians

technicianID	firstName	lastName	email	position	password	status

patch

patchID	name	type	description	releaseDate	status	softwareID	machineID	technicianID

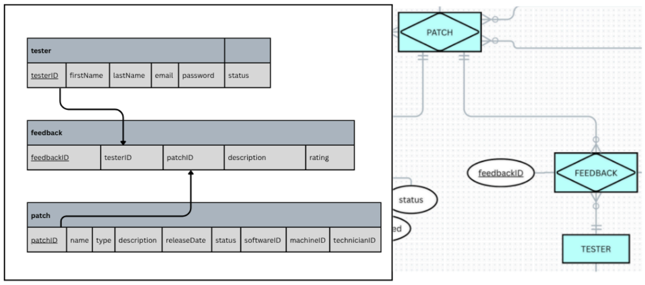
machine

machineID	machineName	deviceType	status

software

softwareID	softwareName	type	version	releaseDate	status

Transformed Relations



tester

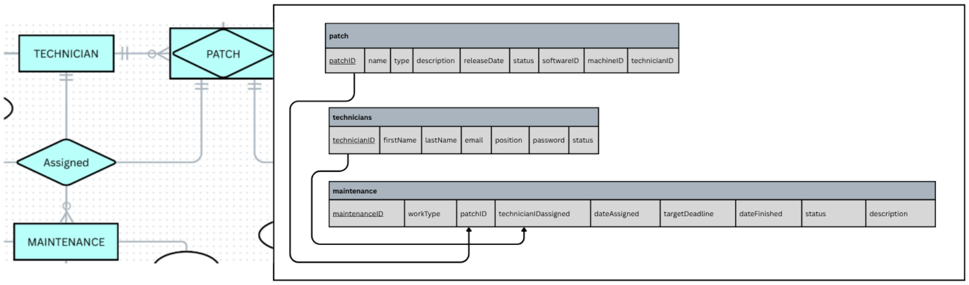
testerID	firstName	lastName	email	password	status

feedback

feedbackID	testerID	patchID	description	rating

patch

patchID	name	type	description	releaseDate	status	softwareID	machineID	technicianID



patch

patchID	name	type	description	releaseDate	status	softwareID	machineID	technicianID

technicians

technicianID	firstName	lastName	email	position	password	status

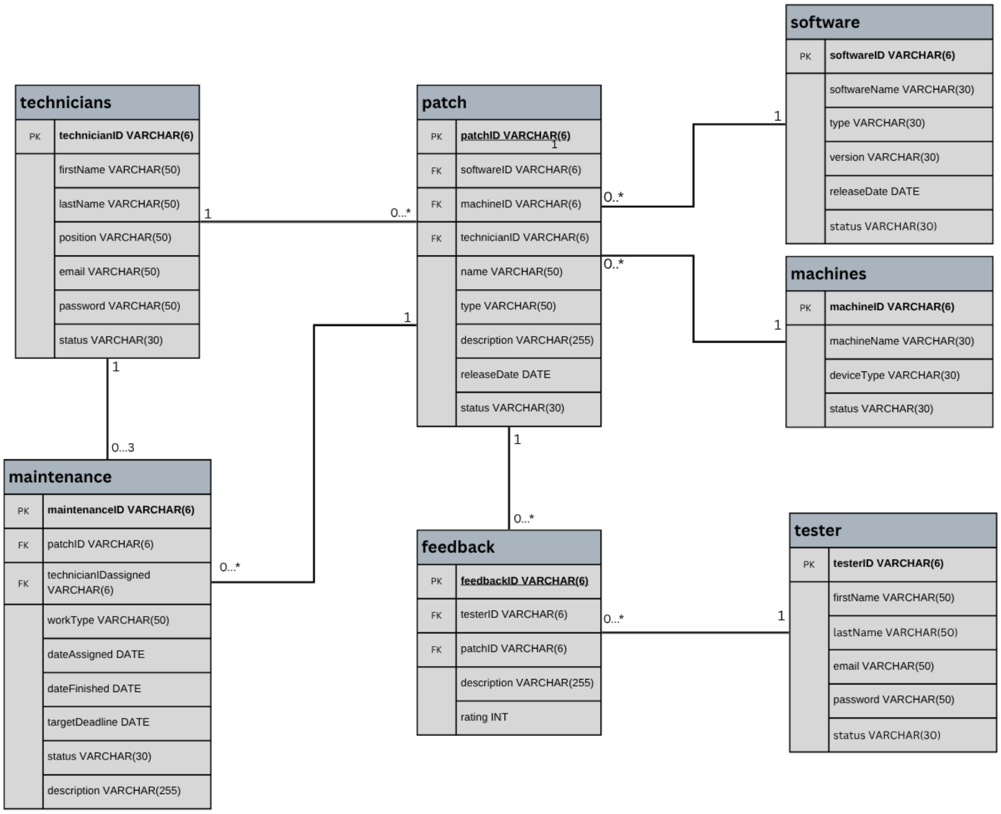
maintenance

maintenanceID	workType	patchID	technicianIDassigned	dateAssigned	targetDeadline	dateFinished	status	description

FUNCTIONAL DEPENDENCIES

3NF achieved	Functional dependencies
Technicians(technicianID, firstName, lastName, position, email, password, status)	technicianID → firstName, lastName, position, email, password, status
Tester(testerID, firstName, lastName, password, email, status)	testerID → firstName, lastName, password, email, status
Software(softwareID, softwareName, type, version, releaseDate, status)	softwareID → softwareName, type, version, releaseDate, status
Machine(machineID, machineName, deviceType, status)	machineID → machineName, deviceType, status
Maintenance(maintenanceID, workType, patchID, technicianIDassigned, dateAssigned, targetDeadline, dateFinished, status, description)	maintenanceID → workType, patchID, technicianIDassigned, dateAssigned, targetDeadline, dateFinished, status, description
Feedback(feedbackID, testerID, patchID, description, rating)	feedbackID → testerID, patchID, description, rating
Patch(patchID, name, type, description, releaseDate, status, softwareID, machineID, technicianID)	patchID → name, type, description, releaseDate, status, softwareID, machineID, technicianID

RELATIONAL DATABASE MODEL USING UML NOTATION



JUSTIFICATIONS

1. Patch is an associative entity because of Transaction 1 (Release new patch). The idea is a technician can release several patches (Many cardinality) for a certain software and machine. So, this relationship needs a new table. Hence, associative entity.
2. Feedback is an associative entity because of Transaction 2 (Send Feedback on patch). A Tester can provide several feedback to several patches. (Many cardinality)
3. We decided each transactional record will have its own identifier as primary key and not a composite primary key. This is so that users of the DB application can simply type the ID to edit, or delete a record.
4. The maintenance and patch table are connected because of Transaction 3 (Schedule patch work to technician). This is a task assignment transaction, where a Technician is assigned a patch (using patchID) to work on.
5. machineID and softwareID are not added in the maintenance table as it is redundant. Since, patchID would already have an assigned machineID and softwareID
6. The data type for the ID of each table is VARCHAR(6) for our convention (machineID convention is “MC1001”). See business rules for the convention of other tables
7. We were able to achieve 3NF as none of the fields have any transitive or partial dependencies. The functional dependencies all fully depend on the primary key

8. We added “status” fields for our core records to indicate “Inactive” status when deleted. This is to avoid deletion anomalies. However, for transaction records: maintenance and feedback. They will be fully deleted upon deletion