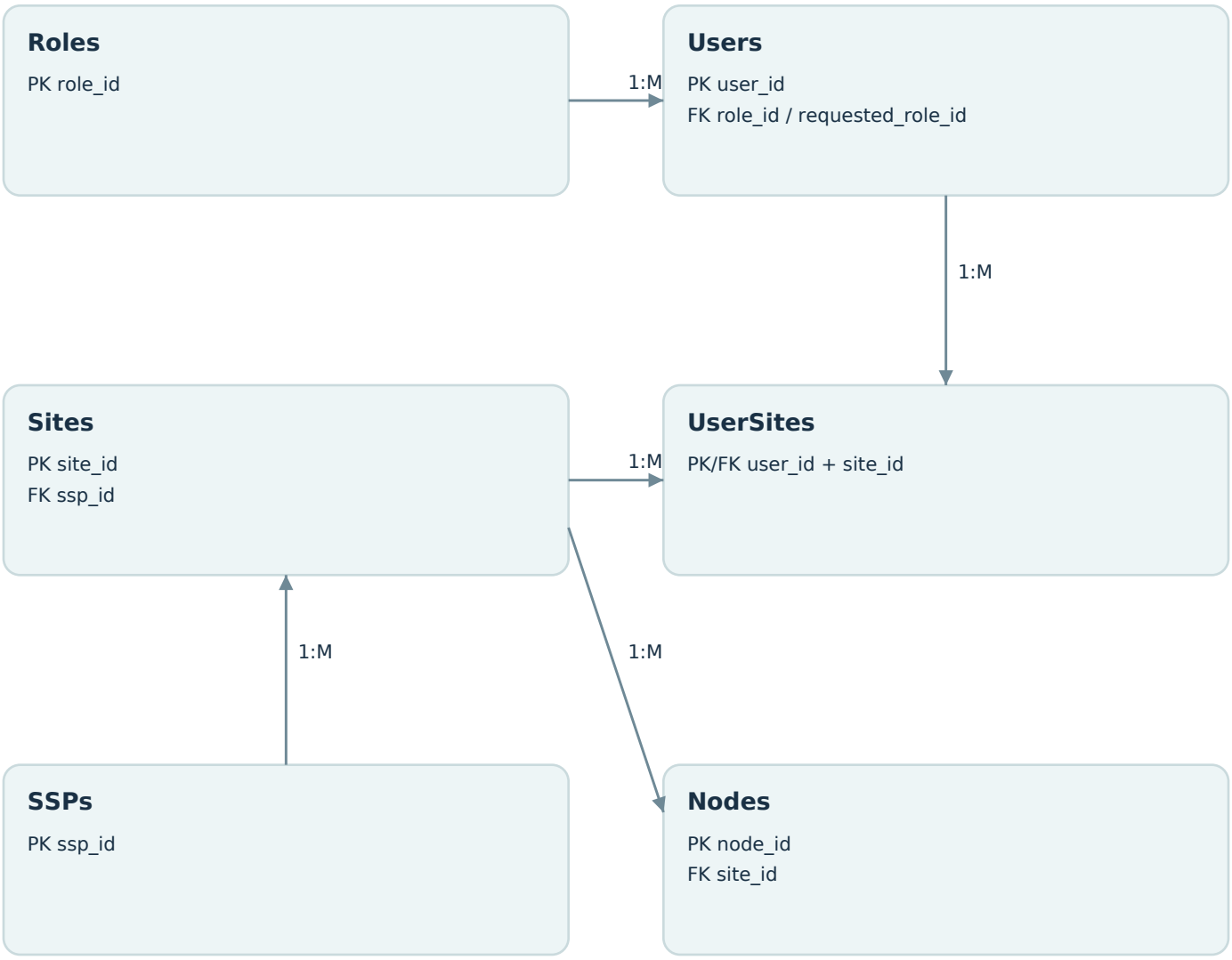
Database Tables and How They Connect

Malcom Daggett | Mercer University
Network Outage Coordination and Automated Notification System

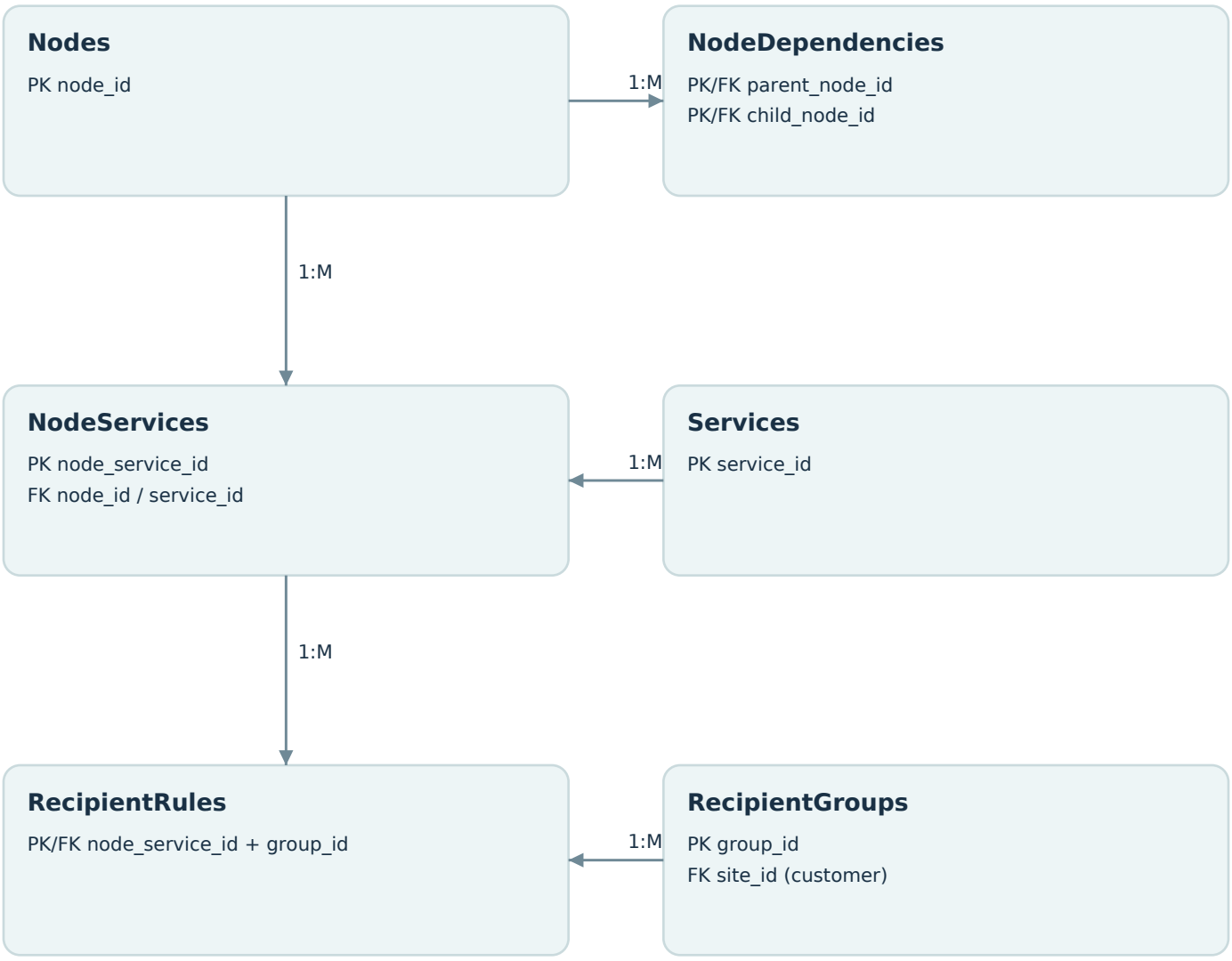
The revised design has 18 tables for accounts, site access, network dependencies, requests, and notices. PK identifies a row. FK links to a row in another table. Each diagram is one view of the same database; repeated names mean the same table.

Diagram 1 | Accounts and sites



Each arrow goes from one parent row to its related rows. A parent may have no related rows yet. Roles links to both requested and approved roles in Users. Users.approved_by is another user with trusted setup permission. UserSites.approved_by also links to Users. Those extra links are omitted from the picture for clarity.

Diagram 2 | Nodes, services, and recipient rules

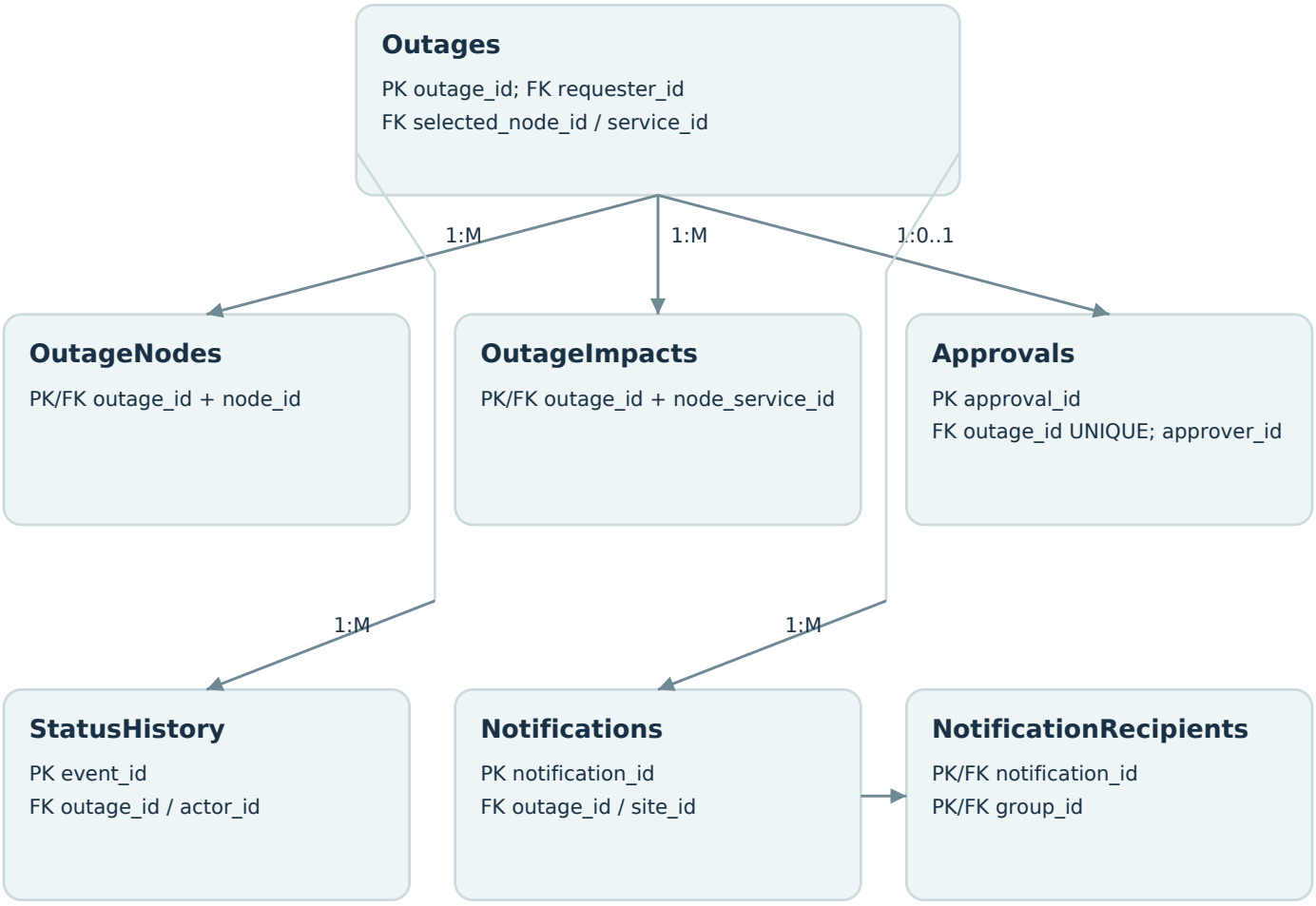


NodeDependencies has two foreign keys to Nodes: one for the upstream parent and one for the dependent child. The picture combines those links; the child is unique to enforce at most one required parent. RecipientGroups.site_id also links to Sites for customer groups.

How the branch becomes impact

For complete_node, start at the selected node and follow child links until no new nodes remain. Include each node once, then include all its NodeServices. For product, include only the selected node/service in this first version. Look up RecipientRules for those services and remove duplicate groups.

Diagram 3 | Saved outage, decision, and notices



OutageNodes.node_id and source_parent_node_id link to Nodes. OutageImpacts.node_service_id links to NodeServices. Approvals.approver_id and StatusHistory.actor_id link to Users. NotificationRecipients.group_id links to RecipientGroups. Notifications.site_id links to Sites for customer notices. These links complete the relationships not drawn in this scoped view.

Each request has zero or one final decision, many history events, and several previews or simulated notices. A saved customer notice is separated by site. The site-contact query checks both notice release and an approved UserSites assignment.

Tables 1-6

Roles

PK role_id; role_name UNIQUE

The three application roles: network_admin, approver, site_contact. Setup permission is separate.

Users

PK user_id; name; email UNIQUE; password_hash; requested_role_id FK; role_id FK optional; account_status; approved_by FK Users optional; approved_at_utc; can_manage_setup

Stores each login and approved role. New accounts are pending with setup permission off. Only the trusted manager assigns roles or activates accounts.

UserSites

PK/FK user_id + site_id; access_status; approved_by FK Users optional; approved_at_utc

Stores requested or approved site assignments. One site contact can have several sites; one site can have several contacts.

SSPs

PK ssp_id; ssp_name

Groups fictional sites. This label helps organize the map; it does not replace dependency-based impact.

Sites

PK site_id; ssp_id FK; site_name

Stores sites such as Alpha, Bravo, Charlie, and Delta. Site names are stored once.

Nodes

PK node_id; site_id FK; node_label

Stores each selectable network node and its site. The sample uses nodes 20, 21, 22, and 30.

Tables 7-12

NodeDependencies

PK/FK parent_node_id + child_node_id; child_node_id UNIQUE

A required link from an upstream node to a dependent child. Each child has at most one parent in this version. Self-links, loops, and duplicate links are rejected.

Services

PK service_id; service_name

Stores services such as WAN Connectivity and Voice without repeating their definitions.

NodeServices

PK node_service_id; node_id FK; service_id FK; UNIQUE node_id + service_id

Lists services at each node. Node 20 has WAN and Voice; the other sample nodes have WAN.

RecipientGroups

PK group_id; group_name; audience; site_id FK optional

Internal groups have no customer site. Every customer group belongs to one site so customer notices can be separated by site.

RecipientRules

PK/FK node_service_id + group_id

Maps each node/service to the groups that need its notices. Each pair appears only once.

Outages

PK outage_id; requester_id FK Users; selected_node_id FK Nodes; selected_service_id FK Services optional; scope_type; title; description; contact; impact; start_utc; end_utc; urgent; urgent_reason; current_status; created_at_utc; submitted_at_utc optional

Stores the request. Scope is product or complete_node. A product requires a selected service available at the node; complete_node includes all services in its affected branch.

Tables 13-18

OutageNodes

PK/FK outage_id + node_id; impact_kind; source_parent_node_id FK Nodes optional

Saves direct/downstream node impact at submission, including the dependency parent that explained downstream inclusion. Old submitted impact does not follow later map changes.

OutageImpacts

PK/FK outage_id + node_service_id

Saves each affected service once. A complete node 20 example has four rows; the node 21 product example has one.

Approvals

PK approval_id; outage_id FK UNIQUE; approver_id FK Users; decision; comment; decided_at_utc

Stores one final approval or rejection per request. A waiting request has no final decision row. Only an active approver can decide.

StatusHistory

PK event_id; outage_id FK; actor_id FK Users; status; changed_at_utc; note

Keeps draft, submission, and decision events, including the person and UTC time.

Notifications

PK notification_id; outage_id FK; site_id FK optional; kind; body; state; created_at_utc; released_at_utc optional

Stores previews and simulated releases. A customer notice is for one site and contains only that site's impact. Internal updates have no site-specific customer body.

NotificationRecipients

PK/FK notification_id + group_id

Saves each reviewed group once per notice. A customer group must belong to the same site as its customer notice.

Why the information is separated

Normalization reduces repeated facts and conflicting updates. Accounts, sites, nodes, services, groups, and outage events each store a different kind of information.

First normal form | One value per field

Do not put lists of nodes, sites, or recipients in one field. Three affected nodes have three OutageNodes rows. Each approved user/site assignment and notice/group link has its own row.

Second normal form | Keep details with their owner

In UserSites, the whole user/site pair identifies an assignment. User names stay in Users and site names in Sites. Likewise, group names stay in RecipientGroups rather than being repeated in every notice/group link.

Third normal form | Avoid indirect repetition

Outages stores user and node IDs rather than copying role, site, or SSP names. Nodes links to Sites, which links to SSPs. Role definitions stay in Roles. Services are defined once and linked through NodeServices.

History saved on purpose

OutageNodes, OutageImpacts, and notice-recipient links preserve the submitted scope. Notice text preserves the wording at release. current_status repeats the latest history event for easier display; both must be updated together. Historical snapshots are deliberate exceptions to recomputing everything from current setup data.

Rules that protect the records

- Enable foreign-key checks and enforce unique emails, node/service pairs, user/site pairs, and notice/group pairs. Use allowed values for roles, account states, outage scopes, and request status.

- Store salted password hashes. Keep setup permission off on registration. The server must verify role and site access on every protected action.

- Reject dependency cycles and self-links; enforce one parent per child. Preserve submitted impact when topology changes, and do not delete referenced sample nodes or services.

- Validate UTC windows and the 240-hour rule. Save submission, impact, history, and the internal update together. Save approval, status, history, and the internal decision update together.

- Release at most one simulated customer notice per approved outage/site. A uniqueness rule on outage, customer-notice kind, and site prevents duplicates. Internal submission/decision kinds also need one saved update each.

- Customer notices require approval and the internal approval update. Their site must match their customer groups and the signed-in user's approved site access. No pending or rejected customer release is allowed.