

retention migration runbook

1. pause ingestion through slack (/database-ingest suspend)
2. navigate to data directory (/var/device-database/data) and elevate permissions
3. vacuum into new database

```
vacuum into 'staging-new.db';
```

this will take a while, ideally start in screen (screen bash; then run sqlite3 shell in there) to avoid aborting operation because of network woes

to monitor progress: du staging-new.db staging.db-wal

new database will slowly climb in size

wal *should not* grow while ingestion is paused, if it does abort and investigate where unexpected writes might be coming from

4. switch sqlite3 shell to new database
5. run 20260914142229.sql
6. run 20260916152325.sql
7. insert relevant migration table changes (make tool;node out/tool/migration-table-dml.mjs)

```
-- 20260914142229.sql
insert into migration (id, name, hash, created_at) values
(20260914142229, '20260914142229.sql',
'4251de9e355e9db85f2fcd5b6b684043c40474ab9448c67b8cc4d016f8f4d1
3c', 1790763116);
-- 20260916152325.sql
insert into migration (id, name, hash, created_at) values
(20260916152325, '20260916152325.sql',
'd6286308f3e72cc52fa763481f4e3ae9dddefc203c40cea24b56aaae6539e2
14', 1790763116);
```

8. analyze

```
analyze;
```

9. swap database files

```
mkdir backup
mv staging.db* backup
mv staging-new.db staging.db
```

10. restart

more migrations being logged in the migration table than the server is aware of is supported, no need to roll out new version

```
# intentional empty space to omit from history
export NOMAD_TOKEN="<ACL-MANAGEMENT-TOKEN>"
```

```
nomad job restart device-database
```

ask Marcos Pérez Ferro for acl token