La hiérarchie qui bouge sous vos pieds

Unités d'organisation, niveaux et groupes, et la propriété dont personne ne vous prévient — l'arbre est courant et non historique, si bien qu'une formation réaffectée en mars réécrit en silence les totaux de district de l'an dernier.

Pierre Robentz Cassion

Leçon 2 sur 8

Données de routine et DHIS2 — Comment la base est structurée

Ce que couvre cette leçon

- Un arbre, et tout y est accroché
- Les groupes ne sont pas des niveaux
- La propriété dont personne ne vous prévient
- Que faire
- La réaffectation, reconstituée
- Agréger vers le haut sans compter deux fois
- La couverture est un indicateur de niveau district
- La suite

Un arbre, et tout y est accroché — Exemple

```
Level 1  Country
Level 2  Region
Level 3  District
Level 4  Facility        <- our 38 units
```

Un arbre, et tout y est accroché — En Python

```
import pandas as pd

vax = pd.read_csv("vaccination-coverage-2024.v1.csv", parse_dates=["period"])
print(f"{vax['facility_id'].nunique()} facilities in the extract")
```

Un arbre, et tout y est accroché — En R

```
library(dplyr)
n_distinct(vax$facility_id)
```

Les groupes ne sont pas des niveaux — En Python

```
by_type = (
    vax.groupby("facility_type")["facility_id"].nunique().sort_values(ascending=False)
)
print(by_type)
```

Les groupes ne sont pas des niveaux — En R

```
vax |> summarise(facilities = n_distinct(facility_id), .by = facility_type)
```

Les groupes ne sont pas des niveaux

Type de formation	Formations
Poste de santé	20
Centre de santé	16
Hôpital de district	2

Les groupes ne sont pas des niveaux

- **Une unité a exactement un parent et un nombre quelconque de groupes.** Poste de santé est un groupe ; cela ne dit pas…
- **Les groupes peuvent se chevaucher.** Une formation peut être à la fois « poste de santé », « difficile d'accès » et «…
- **Les jeux de groupes sont la version sûre.** Un jeu de groupes est un ensemble de groupes mutuellement exclusifs — le…
- Demandez si ce sur quoi vous regroupez est un jeu de groupes — Si ce n'en est pas un, vérifiez que les groupes…

La propriété dont personne ne vous prévient

- DHIS2 stocke la hiérarchie courante, non une hiérarchie historique — Une valeur est rattachée à une formation
- Un total de district différent de la même requête lancée six mois plus tôt, sans aucune saisie entre-temps.
- Un chiffre annuel publié qu'on ne peut pas reproduire.
- Deux rapports de la même période qui divergent, tous deux exacts le jour où ils ont été produits.

Que faire

- Figez la hiérarchie avec l'extraction — Tirez la liste des unités d'organisation en même temps que les données et...

Que faire — En Python

```python
org_units = pd.read_csv("outputs/extract/2026-07/org-units.csv")
    # ea/facility id, name, parent_id, level, extracted_on

pinned = vax.merge(org_units[["facility_id", "district_id"]],
                   on="facility_id", how="left", validate="many_to_one")
assert pinned["district_id"].notna().all(), "facility with no district in the pinned tree"
```

Que faire — En R

```r
org_units <- readr::read_csv(here::here("outputs", "extract", "2026-07", "org-units.csv"))

pinned <- vax |>
  left_join(select(org_units, facility_id, district_id), by = "facility_id",
            relationship = "many-to-one")

stopifnot(!any(is.na(pinned$district_id)))
```

Que faire

- Agrégez depuis l'arbre figé, pas depuis l'arbre vivant — Un chiffre publié en mars est alors reproductible en…
- Consignez la date d'extraction sur les deux — L'avant-dernière leçon en fait une propriété du script d'extraction…

La réaffectation, reconstituée — En Python

```
reassignments = pd.DataFrame([
    {"facility_id": "FAC017", "from_district": "D01", "to_district": "D02",
     "effective": "2024-04-01", "reason": "boundary revision"},
])

def district_at(facility, period, log=reassignments):
    moves = log[(log["facility_id"] == facility)
                & (pd.to_datetime(log["effective"]) <= period)]
    return moves["to_district"].iloc[-1] if len(moves) else None
```

La réaffectation, reconstituée — En R

```
reassignments <- tibble::tribble(
  ~facility_id, ~from_district, ~to_district, ~effective,     ~reason,
  "FAC017",     "D01",          "D02",        as.Date("2024-04-01"), "boundary revision"
)
```

La réaffectation, reconstituée

Les totaux de district sont calculés sur la hiérarchie d'unités d'organisation au 28/07/2026. Une formation a été réaffectée de D01 à D02 en avril 2024 ; ses données 2024 figurent sous D02 sur toute l'année, y compris pour les périodes antérieures à la réaffectation.

La réaffectation, reconstituée

- Cette phrase est le livrable — Elle ne coûte rien et transforme un chiffre irreproductible en chiffre reproductible

Agréger vers le haut sans compter deux fois

- Une formation apparaissant deux fois dans l'arbre — Impossible dans DHIS2 — une unité a un parent — mais tout à fait...

Agréger vers le haut sans compter deux fois — En Python

```
duplicated = org_units["facility_id"].duplicated(keep=False)
assert not duplicated.any(), org_units.loc[duplicated, ["facility_id", "district_id"]]
```

Agréger vers le haut sans compter deux fois — En R

```
stopifnot(!any(duplicated(org_units$facility_id)))
```

Agréger vers le haut sans compter deux fois

- Sommer entre niveaux — Un total de district plus ses formations, c'est le district compté deux fois

Agréger vers le haut sans compter deux fois — En Python

```
print(org_units["level"].value_counts())
```

Agréger vers le haut sans compter deux fois — En R

```
org_units |> count(level)
```

Agréger vers le haut sans compter deux fois

- Toute extraction devrait être à exactement un niveau — Si ce n'est pas le cas, la première chose que fait votre script...

La couverture est un indicateur de niveau district — En Python

```
by_facility = (
    vax[vax["report_submitted"] == True]
    .query("antigen == 'penta3'")
    .groupby("facility_id")
    .agg(doses=("doses_administered", "sum"), target=("target_population", "sum"))
)
by_facility["coverage"] = by_facility["doses"] / by_facility["target"]
print(by_facility["coverage"].describe()[["min", "max"]].round(3))
```

La couverture est un indicateur de niveau district — En R

```
vax |>
  filter(report_submitted, antigen == "penta3") |>
  summarise(coverage = sum(doses_administered) / sum(target_population),
            .by = facility_id) |>
  summarise(min = min(coverage), max = max(coverage))
```

La suite

- Vous savez placer une valeur dans l'espace.

Lire la leçon complète, avec le code exécutable

https://data-analysis.cassion.dev/fr/cours/routine-data-and-dhis2/dhis2-02-org-unit-hierarchy