

Schéma base Climato

1. But

Ce document décrit la structure de la base de données Climato. La base de données est stockée dans une base PostgreSQL avec l'extension TimeScaleDB.

2. Historique des mises à jour

- 2.0 : 21-04-2023 : re-écriture complète

3. postes : table des stations

Actuellement seuls les champs suivants sont utilisés par le serveur :

- **id** : Id de la stations, référencé comme poste_id dans les autres tables
- **meteor** : code de la station au sein du réseau meteoR.OI
- **meteoR** : sera utilisé pour les extractions pour Météo France
- **delta_timezone** : Nombre d'heures entre heure TU et heure locale (+4 pour La Réunion)
- **stop_date** : Ne pas traiter les données venant après cette date, dans le chargement des JSON, ou de mise à jour à partir des backups
- **last_obs_date** : local datetime de l'obs la plus récente, avec une « duration » non nulle. Est maintenue à jour via un trigger sur la table obs.
- **last_obs_id** : id de l'obs la plus récente
- **last_extremes_date** : date de l'extrêmes la plus récente. Est maintenue à jour via un trigger sur la table extrêmes.
- **last_extremes_id** : id de l'extrêmes la plus récente
- **load_json** : booléen permettant d'activer ou non le chargement des fichiers Json.
 - o Si True -> Les backups ne sont plus chargés, et la mise à jour est faite exclusivement à partir des fichiers JSON
 - o Si False -> La base de données est mise à jour journalièrement à partir des mises à jour des dumps venant de la station. Les fichiers JSON sont ignorés, mais sont déplacés dans le répertoire « archive »

4. mesures : table des mesures

Cette table sert au paramétrage du serveur, et éventuellement pour récupérer des propriétés des mesures au niveau de l'interface. Voici les champs de la table :

- **name** : nom de la mesure, seulement pour l'interface utilisateur
- **json_input** : nom de la colonne dans la table « obs » stockant la valeur de la mesure. Ce nom est aussi le nom de la clé utilisé dans le Json de mise à jour des mesures venant des stations
- **json_input_bis** : Si le Json ne contient pas la clé « json_input », ce synonyme peut être utilisé (Certaines mesures ont des noms différents entre la clé Json et le nom de la colonne, pour des raisons historiques)

- **archive_col** : nom de la colonne dans la table archive de WeeWX (utilisé par le programme de migration seulement)
- **archive_table** : Forçage du nom de la table pour retrouver les extrêmes dans la base de WeeWX. Par défaut, quand cette colonne est nulle, le nom du champ de WeeWX est préfixé avec « archive_day_ » pour retrouver le nom de la table de WeeWX stockant les valeurs extrêmes journaliers
- **val_deca** : décalage en heures de la valeur OMM
- **max** : Booléen contrôlant le calcul de maximum pour cette mesure
- **max_deca** : décalage de la valeur max (Mesures OMM)
- **min** : Booléen contrôlant le calcul de minimum pour cette mesure
- **min_deca** : décalage de la valeur min (Mesures OMM)
- **is_avg** : booléen qui indique si cette mesure doit être agrégée en calculant les moyennes, ou par en faisant la somme des valeurs. Ce champ est utilisé par le programme utilisateur pour utiliser la bonne méthode d'agrégation.
- **is_wind** : indique s'il faut aller chercher la direction du vent dans la table des extrêmes de WeeWX
- **omm_link** : indique l'id de la mesure de base associée avec une mesure de type OMM. Null indique que la valeur n'est pas une valeur OMM.
- **allow_zero** : non utilisé actuellement.
- **is_hourly** : indique que la mesure est mise à jour qu'une fois par heure, donc qu'une agrégation par minutes ne fait pas de sens. Non utilisé actuellement

5. exclusion : table des exclusions

non utilisé actuellement, pour une évolution future

6. obs : table des observations

Contient toutes les mesures élémentaires remontées par la station, et/ou issu de la base WeeWX de la station. Colonnes de la table :

- **date_local** : date et heure locale de la mesure
- **date_utc** : date et heure UTC de la mesure
- **poste_id** : id du poste concerné par la mesure
- **duration** : durée de la période d'intégration de la station (LOOP). Est en général de 5mn. Une durée égale à zéro est utilisé pour les valeurs d'observation décalée (paramètre val_deca). Il peut y avoir deux mesures pour un poste donné, et une datetime donnée, une avec une duration, et une sans.
- Ensuite une colonne par mesure gérée par le système

```
barometer, barometer_omm, dewpoint, etp, extra_humid1,
extra_humid2, extra_temp1, extra_temp2, extra_temp3, hail,
hail_rate, heatindex, heating_temp, in_humidity, in_temp,
leaf_temp1, leaf_temp2, leaf_wet1, leaf_wet2, out_humidity,
out_humidity_omm, out_temp, out_temp_omm, pressure, radiation,
rain, rain_omm, rain_rate, rx, soil_moist1,
soil_moist2, soil_moist3, soil_moist4, soil_temp1, soil_temp2,
soil_temp3, soil_temp4, uv, voltage, wind,
wind_gust, wind10, wind10_omm, windchill
```

et enfin les colonnes qui seront utilisées dans une version future quand nous utiliseront des tests de qualité :

- qa_modifications : va stocker le nombre de mise à jour de la mesure
- qa_incidents : va stocker le nombre d'incidents imputés à cette mesure
- qa_check_done : booléen qui indique si le check de qualité a été fait sur cette mesure

7. histo_obs : Mesures liées

Une mesure a une datetime donnée peut générer plusieurs enregistrements dans la table obs :

- o La mesure principale à la date de la mesure avec les données, avec la « duration » de la mesure, donc non nulle
- o Une pour plusieurs mesures secondaires à la datetime de la mesure principale augmentée du « val_dk » de la mesure OMM. Dans ce cas la duration est nulle.

Cette table contient 3 colonnes :

- **id** : Clé primaire, auto-incrémentée
- **src_obs_id** : id de la mesure principale (avec « duration » non nulle)
- **target_obs_id** : id de la mesure secondaires (avec « duration » nulle)

8. extremes : table des records

La table des records garde par mesure, les valeurs extrêmes journalières relevées. Elles pourront être agrégées à différents niveaux pour indiquer les records sur une période donnée. Les colonnes sont :

- date_local : date du jour du record (A partir des mesures en datetime local)
- poste_id : id du poste concerné par le record
- mesure_id : id de la mesure concernée par le record
- min : valeur minimale du record
- min_time : date et heure du record min
- max : valeur maximale du record
- max_time : date et heure du record max
- max_dir : direction du vent pour les max de type vent

9. histo_extreme : Lien records/mesure

Cette table a 4 colonnes :

- id : clé primaire, auto-incrémente
- target_x_id : extremes.id concerné
- src_min_obs_id : id de l'obs ayant modifié le min de l'extrême. -1 quand la mise à jour vient des extremes weewx pendant la migration. Celui avec le plus grand id, est le dernier qui a modifié le min.
- src_max_obs_id : id de l'obs ayant modifié le max de l'extrême. -1 quand la mise à jour vient des extremes weewx pendant la migration. Celui avec le plus grand id, est le dernier qui a modifié le max.

10. incidents : table des incidents

Le programme serveur indique dans cette table tous les incidents qu'il rencontre. Les incidents sont classés par

- dat : date et heure de l'incident
- source : information sur la fonction ayant généré cet incident
- level : exception, error, info
- reason : message de l'incident
- details : Json contenant des informations complémentaires a cet incident. En cas d'exception pendant l'exécution, le stack est stocké dans le json.
- active : booléen pour indiquer si l'incident a été vu et traité (donc active -> false)

11. annotations : table des annotations

Cette table est utilisée par Grafana pour stocker des annotations sur les mesures relevées, comme des événements météorologiques spéciaux

Colonnes :

- time : date et heure du début de l'annotation
- timeend : date et heure de la fin de l'annotation
- text : description de l'annotation
- tags : tag de l'annotation

12. obs_day : vue des observations journalières

Cette vue contient une agrégation journalière des mesures de la table obs. Elle est automatiquement tenue à jour par TimeScaleDB. Elle contient les mêmes colonnes que la table obs.

Les programmes utilisateurs doivent utiliser cette vue pour retrouver les mesures agrégées au niveau journalier ou supérieur. L'accès à la table obs étant réservé aux demandes nécessitant l'accès aux mesures élémentaires, comme à la suite d'un événement météorologique particulier.

13. get_last_obs(poste_id) : Dernieres mesures d'un poste

Retourne une chaine JSON avec les dernières valeurs mesurées. Les valeurs OMM avec un decalage (val_dk, min_deca, max_deca) ne sont pas retournées par cette procédure