

6.99 Schema Changes

Last Modified on 09/08/2024 11:33 am ACST

Schema changes between CareRight V6.98.7 and 6.99

New Tables/Views

- external_medication_platform_settings

New Columns

- data_producer_oauth2_credentials.scopes
- ihc_transport_services.line_item_id

New Indexes

- None

Changed Views

- None

Changed Columns/Indexes

- data_retrievers.retriever - Now limit 50
- f_setup_rate_defs.SRD_PTYPE_1 now account_type_1
- f_setup_rate_defs.SRD_PTYPE_2 now account_type_2
- f_setup_rate_defs.SRD_PTYPE_3 now account_type_3
- f_setup_rate_defs.SRD_PTYPE_4 now account_type_4
- f_setup_rate_defs.SRD_PTYPE_5 now account_type_5

Changed Tables

- ihc_transports renamed to ihc_transport_services

Deleted Tables/Views

- None

Deleted Columns/Indexes

- admission_transports.charge_amount

Schema changes between CareRight 6.99 and 6.99.1

New Columns

- entitlements.notes
- appointment_types.show_cancellation_reason
- appointments.cancellation_reason_id
- providers.mo_routing_id
- current_assessments.submitted_for_approval_by
- current_assessments.submitted_for_approval_at
- entitlement_usages.archived_at
- people.current_information_classification_level

New Indexes

- correspondences.created_at
- documents.created_at

Other

Sql server

```
update a set current_information_classification_level = m.current_classification
from people a
join (
    select a.id,
           coalesce(b.highest_classification_level, 0) current_classification#{" "}
    from people a#{" "}
    left join (select b.id, max(classification_level) highest_classification_level#{" "}
              from information_classifications a#{" "}
              join people b on b.id = a.classificationable_id and a.classificationable_type = 'Person'
              where (a.archived is null or a.archived = 0)
              and a.classified_from < CURRENT_TIMESTAMP
              and (a.classified_until IS NULL OR a.classified_until > CURRENT_TIMESTAMP)
              group by b.id
            ) b on a.id = b.id
) m on a.id = m.id
```

Postgres

```
UPDATE people a
  SET current_information_classification_level = m.current_classification
  from people b
  join (
    select a.id,
      coalesce(b.highest_classification_level, 0) current_classification#{" "}
    from people a#{" "}
    left join (select b.id, max(classification_level) highest_classification_level#{" "}
      from information_classifications a#{" "}
      join people b on b.id = a.classificationable_id and a.classificationable_type = 'Person'
      where (a.archived is null or a.archived = 0)
      and a.classified_from < CURRENT_TIMESTAMP
      and (a.classified_until IS NULL OR a.classified_until > CURRENT_TIMESTAMP)
      group by b.id
    ) b on a.id = b.id

    ) m on m.id = b.id
WHERE a.id = b.id
```
