

7.4.0 Schema Changes

Last Modified on 17/01/2025 11:03 am ACDT

Schema changes between CareRight V7.3.0 and CareRight V7.4.0

db/migrate/.rb

db/migrate/.rb

New Tables

- postcodes

New Columns

- correspondence_settings.postcode_last_update
- appointments.checklist_instance_id
- merge_forms.url
- program_sessions.attendees_count
- program_sessions.providers_count
- program_schedules.enrolment_count
- program_schedules.sessions_count
- program_schedules.providers_count
- sms_messages.program_schedule_id
- program_schedules.sms_count
- branding_settings.logo_width_px
- branding_settings.logo_height_px

Changed Columns

None

Deleted Tables

None

Deleted Columns

None

CareRight 7.3.1 Changes

Receipt Numbers

We have swapped from referring to the *f_counters* table and the *ct_rec_no* field to a proper database sequence (*receipt_number*).

- Removed column *ct_counters.ct_rec_no*
- Create sequence *receipt_numbers*

This does not change the definition of affected columns that referenced this number. However it changed the **behaviour**. Importantly, where a transaction rolled back; **previously this would discard any attempts to assign a new receipt number**.

From this release, this will **continually increment the receipt numbers even if the transaction is rolled back** This may manifest as "missing" receipt numbers; but is an expected behaviour.

See also: <https://learn.microsoft.com/en-us/sql/relational-databases/sequence-numbers/sequence-numbers?view=sql-server-ver16>

```

existing_data = CounterSet.where(ct_number: 1).first
start = 1
start = CounterSet.next_counter_value("CT_REC_NO") if existing_data
# historically went into:
# f_statement.stat_ref
# refunds.number
# receipt.number
check = (Statement.maximum(:stat_ref) || 0) + 1
start = check if check > start
check = (Receipt.maximum(:number) || 0) + 1
start = check if check > start
check = (Refund.maximum(:number) || 0) + 1
start = check if check > start
case ActiveRecord::Base.connection.adapter_name.downcase
when "sqlserver"
  execute <<~END_OF_SQL
    CREATE SEQUENCE receipt_number START WITH #{start} INCREMENT BY 1
  END_OF_SQL
when "postgresql"
  execute <<~END_OF_SQL
    CREATE SEQUENCE receipt_number START #{start}
  END_OF_SQL
else
  raise "Unhandled database type"
end
remove_column :f_counters, :ct_rec_no
end

```

While this change is small and should be routine, we are recommending customers have:

- Appropriate database backups in place
 - A regression test plan to validate receipt creation post deployment functions as expected.
-