

# Package ‘omopgenerics’

May 17, 2024

**Title** Methods and Classes for the OMOP Common Data Model

**Version** 0.2.1

**Description** Provides definitions of core classes and methods used by analytic pipelines that query the OMOP (Observational Medical Outcomes Partnership) common data model.

**License** Apache License (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Imports** cli, dbplyr, dplyr, glue, methods, rlang, snakecase, stringr, tidyr

**Depends** R (>= 4.1)

**Suggests** covr, knitr, rmarkdown, testthat (>= 3.0.0)

**URL** <https://darwin-eu-dev.github.io/omopgenerics/>

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Marti Catala [aut, cre] (<<https://orcid.org/0000-0003-3308-9905>>),  
Edward Burn [aut] (<<https://orcid.org/0000-0002-9286-1128>>),  
Mike Du [ctb] (<<https://orcid.org/0000-0002-9517-8834>>),  
Yuchen Guo [ctb] (<<https://orcid.org/0000-0002-0847-4855>>),  
Adam Black [ctb] (<<https://orcid.org/0000-0001-5576-8701>>),  
Marta Alcalde-Herraiz [ctb] (<<https://orcid.org/0009-0002-4405-1814>>)

**Maintainer** Marti Catala <[marti.catalasabate@ndorms.ox.ac.uk](mailto:marti.catalasabate@ndorms.ox.ac.uk)>

**Repository** CRAN

**Date/Publication** 2024-05-16 22:40:02 UTC

**R topics documented:**

achillesColumns . . . . .	3
achillesTables . . . . .	4
attrition . . . . .	5
attrition.cohort_table . . . . .	5
bind . . . . .	6
bind.cohort_table . . . . .	7
bind.summarised_result . . . . .	8
cdmFromTables . . . . .	9
cdmName . . . . .	10
cdmReference . . . . .	11
cdmSelect . . . . .	12
cdmSource . . . . .	12
cdmSourceType . . . . .	13
cdmTableFromSource . . . . .	14
cdmVersion . . . . .	15
checkCohortRequirements . . . . .	16
cohortCodelist . . . . .	17
cohortColumns . . . . .	18
cohortCount . . . . .	19
cohortTables . . . . .	20
collect.cdm_reference . . . . .	21
collect.cohort_table . . . . .	22
compute.cdm_table . . . . .	22
dropSourceTable . . . . .	23
dropTable . . . . .	23
emptyAchillesTable . . . . .	24
emptyCdmReference . . . . .	25
emptyCodelist . . . . .	26
emptyCohortTable . . . . .	26
emptyOmopTable . . . . .	27
emptySummarisedResult . . . . .	28
estimateTypeChoices . . . . .	29
exportSummarisedResult . . . . .	29
getCohortId . . . . .	30
getCohortName . . . . .	30
insertFromSource . . . . .	31
insertTable . . . . .	31
listSourceTables . . . . .	32
newAchillesTable . . . . .	32
newCdmReference . . . . .	33
newCdmSource . . . . .	34
newCdmTable . . . . .	34
newCodelist . . . . .	35
newCohortTable . . . . .	35
newConceptSetExpression . . . . .	36
newLocalSource . . . . .	37

newOmopTable . . . . . 38

newSummarisedResult . . . . . 38

omopColumns . . . . . 39

omopTables . . . . . 40

participants . . . . . 41

print.cdm\_reference . . . . . 41

print.codelist . . . . . 42

print.conceptSetExpression . . . . . 43

readSourceTable . . . . . 44

recordCohortAttrition . . . . . 44

resultColumns . . . . . 45

settings . . . . . 46

settings.cohort\_table . . . . . 46

settings.summarised\_result . . . . . 47

sourceType . . . . . 49

summary.cdm\_reference . . . . . 49

summary.cohort\_table . . . . . 50

summary.summarised\_result . . . . . 51

suppress . . . . . 52

suppress.summarised\_result . . . . . 53

tableName . . . . . 54

tableSource . . . . . 55

tmpPrefix . . . . . 56

toSnakeCase . . . . . 56

uniqueTableName . . . . . 57

validateNameLevel . . . . . 57

[[.cdm\_reference . . . . . 58

[[<-.cdm\_reference . . . . . 59

\$.cdm\_reference . . . . . 59

\$<-.cdm\_reference . . . . . 60

**Index** **62**

---

achillesColumns      *Required columns for each of the achilles result tables*

---

**Description**

Required columns for each of the achilles result tables

**Usage**

achillesColumns(table, required = TRUE, version = "5.3")

**Arguments**

table	Table for which to see the required columns. One of "achilles_analysis", "achilles_results", or "achilles_results_dist".
required	Whether to include only required fields.
version	Version of the OMOP Common Data Model.

**Value**

Character vector with the column names

**Examples**

```
library(omopgenerics)
achillesColumns("achilles_analysis")
achillesColumns("achilles_results")
achillesColumns("achilles_results_dist")
```

---

achillesTables

*Names of the tables that contain the results of achilles analyses*

---

**Description**

Names of the tables that contain the results of achilles analyses

**Usage**

```
achillesTables(version = "5.3")
```

**Arguments**

version	Version of the OMOP Common Data Model.
---------	--

**Value**

Names of the tables that are contain the results from the achilles analyses

**Examples**

```
library(omopgenerics)
achillesTables()
```

---

attrition	<i>Get attrition from an object.</i>
-----------	--------------------------------------

---

**Description**

Get attrition from an object.

**Usage**

```
attrition(x)
```

**Arguments**

x                    An object for which to get an attrition summary.

**Value**

A table with the attrition.

---

attrition.cohort_table	<i>Get cohort attrition from a cohort_table object.</i>
------------------------	---

---

**Description**

Get cohort attrition from a cohort\_table object.

**Usage**

```
## S3 method for class 'cohort_table'  
attrition(x)
```

**Arguments**

x                    A cohort\_table

**Value**

A table with the attrition.

## Examples

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort <- tibble(
  cohort_definition_id = c(1, 1, 1, 2),
  subject_id = 1,
  cohort_start_date = as.Date(c("2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01")),
  cohort_end_date = as.Date(c("2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01")),
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "my_example_cdm",
  cohortTables = list("cohort1" = cohort)
)

attrition(cdm$cohort1)
```

---

bind

*Bind two or more objects of the same class.*

---

## Description

Bind two or more objects of the same class.

## Usage

```
bind(...)
```

## Arguments

...                    Objects to bind.

## Value

New object.

---

bind.cohort_table	<i>Bind two or more cohort tables</i>
-------------------	---------------------------------------

---

## Description

Bind two or more cohort tables

## Usage

```
## S3 method for class 'cohort_table'  
bind(..., name)
```

## Arguments

...	Generated cohort set objects to bind. At least two must be provided.
name	Name of the new generated cohort set.

## Value

The cdm object with a new generated cohort set containing all of the cohorts passed.

## Examples

```
library(omopgenerics)  
library(dplyr, warn.conflicts = FALSE)  
  
cohort1 <- tibble(  
  cohort_definition_id = 1,  
  subject_id = 1:3,  
  cohort_start_date = as.Date("2010-01-01"),  
  cohort_end_date = as.Date("2010-01-05")  
)  
cohort2 <- tibble(  
  cohort_definition_id = c(2, 2, 3, 3, 3),  
  subject_id = c(1, 2, 3, 1, 2),  
  cohort_start_date = as.Date("2010-01-01"),  
  cohort_end_date = as.Date("2010-01-05")  
)  
cdm <- cdmFromTables(  
  tables = list(  
    "person" = tibble(  
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,  
      race_concept_id = 0, ethnicity_concept_id = 0  
    ),  
    "observation_period" = tibble(  
      observation_period_id = 1:3, person_id = 1:3,  
      observation_period_start_date = as.Date("2000-01-01"),  
      observation_period_end_date = as.Date("2025-12-31"),  
      period_type_concept_id = 0  
    )  
  )  
)
```

```

    )
  ),
  cdmName = "mock",
  cohortTables = list("cohort1" = cohort1, "cohort2" = cohort2)
)

cdm <- bind(cdm$cohort1, cdm$cohort2, name = "cohort3")
settings(cdm$cohort3)
cdm$cohort3

```

---

bind.summarised\_result

*Bind two or summarised\_result objects*

---

### Description

Bind two or summarised\_result objects

### Usage

```

## S3 method for class 'summarised_result'
bind(...)

```

### Arguments

... summarised\_result objects

### Value

A summarised\_result object the merged objects.

### Examples

```

library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  )
),

```



```
    cdmName = "mock",
    cohortTables = list("cohort1" = tibble(
      cohort_definition_id = 1,
      subject_id = 1:3,
      cohort_start_date = as.Date("2010-01-01"),
      cohort_end_date = as.Date("2010-01-05")
    ))
  )

result1 <- summary(cdm)
result2 <- summary(cdm$cohort1)

mergedResult <- bind(result1, result2)
mergedResult
```

---

cdmFromTables

*Create a cdm object from local tables*

---

## Description

Create a cdm object from local tables

## Usage

```
cdmFromTables(tables, cdmName, cohortTables = list(), cdmVersion = NULL)
```

## Arguments

tables	List of tables to be part of the cdm object.
cdmName	Name of the cdm object.
cohortTables	List of tables that contains cohort, cohort_set and cohort_attrition can be provided as attributes.
cdmVersion	Version of the cdm_reference

## Value

A cdm\_reference object.

## Examples

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
```

```

)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test"
)

```

---

cdmName

*Get the name of a cdm\_reference associated object*


---

## Description

Get the name of a cdm\_reference associated object

## Usage

```
cdmName(x)
```

## Arguments

x                    A cdm\_reference or cdm\_table object.

## Value

Name of the cdm\_reference.

## Examples

```

library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  )

```

```

    )
  ),
  cdmName = "mock"
)

cdmName(cdm)

cdmName(cdm$person)

```

---

cdmReference	<i>Get the cdm_reference of a cdm_table.</i>
--------------	--

---

### Description

Get the cdm\_reference of a cdm\_table.

### Usage

```
cdmReference(table)
```

### Arguments

table            A cdm\_table.

### Value

A cdm\_reference.

### Examples

```

library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

```

```
)
  cdmReference(cdm$person)
```

---

cdmSelect	<i>Restrict the cdm object to a subset of tables.</i>
-----------	---

---

### Description

Restrict the cdm object to a subset of tables.

### Usage

```
cdmSelect(cdm, ...)
```

### Arguments

cdm	A cdm_reference object.
...	Selection of tables to use, it supports tidysselect expressions.

### Value

A cdm\_reference with only the specified tables.

### Examples

```
cdm <- emptyCdmReference("my cdm")
cdm

cdm |>
  cdmSelect("person")
```

---

cdmSource	<i>Get the source of a cdm_reference.</i>
-----------	---

---

### Description

Get the source of a cdm\_reference.

### Usage

```
cdmSource(cdm)
```

**Arguments**

cdm                    A cdm\_reference object.

**Value**

A cdm\_source object.

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

cdmSource(cdm)
```

---

cdmSourceType                    *Get the source type of a cdm\_reference object.*

---

**Description**

Get the source type of a cdm\_reference object.

**Usage**

```
cdmSourceType(cdm)
```

**Arguments**

cdm                    A cdm\_reference object.

**Value**

A character vector with the type of source of the cdm\_reference object.

**Examples**

```

library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

cdmSourceType(cdm = cdm)

```

---

cdmTableFromSource	<i>This is an internal developer focused function that creates a cdm_table from a table that shares the source but it is not a cdm_table. Please use insertTable if you want to insert a table to a cdm_reference object.</i>
--------------------	---

---

**Description**

This is an internal developer focused function that creates a cdm\_table from a table that shares the source but it is not a cdm\_table. Please use insertTable if you want to insert a table to a cdm\_reference object.

**Usage**

```
cdmTableFromSource(src, value)
```

**Arguments**

src	A cdm_source object.
value	A table that shares source with the cdm_reference object.

**Value**

A cdm\_table.

---

cdmVersion	<i>Get the version of a cdm_reference.</i>
------------	--

---

**Description**

Get the version of a cdm\_reference.

**Usage**

```
cdmVersion(cdm)
```

**Arguments**

cdm            A cdm\_reference object.

**Value**

Version of the cdm\_reference.

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

cdmVersion(cdm)
```

---

`checkCohortRequirements`*Check whether a cohort table satisfies requirements*

---

**Description**

Check whether a cohort table satisfies requirements

**Usage**

```
checkCohortRequirements(  
  cohort,  
  checkEndAfterStart = TRUE,  
  checkOverlappingEntries = TRUE,  
  checkMissingValues = TRUE,  
  checkInObservation = TRUE,  
  type = "error",  
  call = parent.frame()  
)
```

**Arguments**

<code>cohort</code>	cohort_table object.
<code>checkEndAfterStart</code>	If TRUE a check that all cohort end dates come on or after cohort start date will be performed.
<code>checkOverlappingEntries</code>	If TRUE a check that no individuals have overlapping cohort entries will be performed.
<code>checkMissingValues</code>	If TRUE a check that there are no missing values in required fields will be performed.
<code>checkInObservation</code>	If TRUE a check that cohort entries are within the individuals observation periods will be performed.
<code>type</code>	Can be either "error" or "warning". If "error" any check failure will result in an error, whereas if "warning" any check failure will result in a warning.
<code>call</code>	The call for which to return the error message.

**Value**

An error will be returned if any of the selected checks fail.



## Examples

```

library(omopgenerics)
person <- dplyr::tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- dplyr::tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test"
)
cdm <- insertTable(cdm, name = "cohort1", table = dplyr::tibble(
  cohort_definition_id = 1, subject_id = 1,
  cohort_start_date = as.Date(c("2020-01-01", "2020-01-10")),
  cohort_end_date = as.Date(c("2020-01-10", "2020-01-25"))
))
cdm$cohort1 <- newCohortTable(cdm$cohort1, .softValidation = TRUE)
# checkCohortRequirements(cdm$cohort1)

```

---

cohortCodelist

*Get codelist from a cohort\_table object.*

---

## Description

Get codelist from a cohort\_table object.

## Usage

```

cohortCodelist(
  cohortTable,
  cohortId,
  type = c("index event", "inclusion criteria", "exclusion criteria", "exit criteria")
)

```

## Arguments

cohortTable	A cohort_table object.
cohortId	A particular cohort definition id that is present in the cohort table.
type	The reason for the codelist. Can be "index event", "inclusion criteria", or "exit criteria".

**Value**

A table with the codelists used.

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort <- tibble(
  cohort_definition_id = c(1, 1, 1, 2),
  subject_id = 1,
  cohort_start_date = as.Date(c(
    "2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01"
  )),
  cohort_end_date = as.Date(c(
    "2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01"
  ))
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "my_example_cdm",
  cohortTables = list("cohort1" = cohort)
)
cdm$cohort1 <- newCohortTable(table = cdm$cohort1,
  cohortCodelistRef = dplyr::tibble(
    cohort_definition_id = c(1,1,1,2,2),
    codelist_name =c("disease X", "disease X", "disease X",
      "disease Y", "disease Y"),
    concept_id = c(1,2,3,4,5),
    type = "index event"
  ))
cohortCodelist(cdm$cohort1, cohortId = 1, type = "index event")
```

---

cohortColumns

*Required columns for a generated cohort set.*

---

**Description**

Required columns for a generated cohort set.

**Usage**

```
cohortColumns(table, required = TRUE, version = "5.3")
```

**Arguments**

table	Either cohort, cohort_set or cohort_attrition
required	Whether to include only required fields.
version	Version of the OMOP Common Data Model.

**Value**

Character vector with the column names  
Required columns

**Examples**

```
library(omopgenerics)  
cohortColumns("cohort")
```

---

cohortCount	<i>Get cohort counts from a cohort_table object.</i>
-------------	--

---

**Description**

Get cohort counts from a cohort\_table object.

**Usage**

```
cohortCount(cohort)
```

**Arguments**

cohort	A cohort_table object.
--------	------------------------

**Value**

A table with the counts.

**Examples**

```

library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort <- tibble(
  cohort_definition_id = c(1, 1, 1, 2),
  subject_id = 1,
  cohort_start_date = as.Date(c(
    "2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01"
  )),
  cohort_end_date = as.Date(c(
    "2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01"
  )),
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "my_example_cdm",
  cohortTables = list("cohort1" = cohort)
)

cohortCount(cdm$cohort1)

```

---

cohortTables	<i>Cohort tables that a cdm reference can contain in the OMOP Common Data Model.</i>
--------------	--

---

**Description**

Cohort tables that a cdm reference can contain in the OMOP Common Data Model.

**Usage**

```
cohortTables(version = "5.3")
```

**Arguments**

version	Version of the OMOP Common Data Model.
---------	--

**Value**

cohort tables

**Examples**

```
library(omopgenerics)
cohortTables()
```

---

collect.cdm\_reference *Retrieves the cdm reference into a local cdm.*

---

**Description**

Retrieves the cdm reference into a local cdm.

**Usage**

```
## S3 method for class 'cdm_reference'
collect(x, ...)
```

**Arguments**

x                    A cdm\_reference object.  
...                  For compatibility only, not used.

**Value**

A local cdm\_reference.

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = dplyr::tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = dplyr::tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  )
)
```

```

    ),
    cdmName = "mock"
  )

  collect(cdm)

```

---

collect.cohort\_table *To collect a cohort\_table object.*

---

### Description

To collect a cohort\_table object.

### Usage

```

## S3 method for class 'cohort_table'
collect(x, ...)

```

### Arguments

x	cohort_table object.
...	Not used (for compatibility).

### Value

A data frame with the cohort\_table

---

compute.cdm\_table *Store results in a table.*

---

### Description

Store results in a table.

### Usage

```

## S3 method for class 'cdm_table'
compute(x, name = uniqueTableName(), temporary = TRUE, overwrite = TRUE, ...)

```

### Arguments

x	Table in the cdm.
name	Name to store the table with.
temporary	Whether to store table temporarily (TRUE) or permanent (FALSE).
overwrite	Whether to overwrite previously existing table with name same.
...	For compatibility (not used).

**Value**

Reference to a table in the cdm

---

dropSourceTable	<i>Drop a table from a cdm object.</i>
-----------------	--

---

**Description**

Drop a table from a cdm object.

**Usage**

```
dropSourceTable(cdm, name)
```

**Arguments**

cdm	A cdm reference.
name	Name(s) of the table(s) to insert. Tidysselect statements are supported.

**Value**

The table in the cdm reference.

---

dropTable	<i>Drop a table from a cdm object.</i>
-----------	--

---

**Description**

Drop a table from a cdm object.

**Usage**

```
dropTable(cdm, name)
```

**Arguments**

cdm	A cdm reference.
name	Name(s) of the table(s) to drop Tidysselect statements are supported.

**Value**

The cdm reference.

## Examples

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort <- tibble(
  cohort_definition_id = c(1, 1, 1, 2),
  subject_id = 1,
  cohort_start_date = as.Date(c(
    "2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01"
  )),
  cohort_end_date = as.Date(c(
    "2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01"
  )),
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "my_example_cdm",
  cohortTables = list("cohort1" = cohort)
)

cdm

cdm <- dropTable(cdm = cdm, name = "cohort1")

cdm
```

---

emptyAchillesTable      *Create an empty achilles table*

---

## Description

Create an empty achilles table

## Usage

```
emptyAchillesTable(cdm, name)
```



**Arguments**

cdm            A cdm\_reference to create the table.  
name           Name of the table to create.

**Value**

The cdm\_reference with an achilles empty table

**Examples**

```
library(omopgenerics)
cdm <- emptyCdmReference("my_example_cdm")
emptyAchillesTable(cdm = cdm, name = "achilles_results" )
```

---

emptyCdmReference        *Create an empty cdm\_reference*

---

**Description**

Create an empty cdm\_reference

**Usage**

```
emptyCdmReference(cdmName, cdmVersion = NULL)
```

**Arguments**

cdmName        Name of the cdm\_reference  
cdmVersion     Version of the cdm\_reference

**Value**

An empty cdm\_reference

**Examples**

```
library(omopgenerics)
emptyCdmReference(cdmName = "my_example_cdm")
```

---

emptyCodelist	<i>Empty codelist object.</i>
---------------	-------------------------------

---

**Description**

Empty codelist object.

**Usage**

```
emptyCodelist()
```

**Value**

An empty codelist object.

**Examples**

```
emptyCodelist()
```

---

emptyCohortTable	<i>Create an empty cohort_table object</i>
------------------	--

---

**Description**

Create an empty cohort\_table object

**Usage**

```
emptyCohortTable(cdm, name, overwrite = TRUE)
```

**Arguments**

cdm	A cdm_reference to create the table.
name	Name of the table to create.
overwrite	Whether to overwrite an existent table.

**Value**

The cdm\_reference with an empty cohort table

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test"
)

cdm <- emptyCohortTable(cdm, "my_empty_cohort")

cdm
cdm$my_empty_cohort
settings(cdm$my_empty_cohort)
attrition(cdm$my_empty_cohort)
cohortCount(cdm$my_empty_cohort)
```

---

emptyOmotable	<i>Create an empty omop table</i>
---------------	-----------------------------------

---

**Description**

Create an empty omop table

**Usage**

```
emptyOmotable(cdm, name)
```

**Arguments**

cdm	A cdm_reference to create the table.
name	Name of the table to create.

**Value**

The cdm\_reference with an empty cohort table

## Examples

```
library(omopgenerics)

person <- dplyr::tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- dplyr::tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test"
)

cdm <- emptyOmopTable(cdm, "drug_exposure")

cdm$drug_exposure
```

---

emptySummarisedResult *Empty summarised\_result object.*

---

## Description

Empty summarised\_result object.

## Usage

```
emptySummarisedResult(settings = NULL)
```

## Arguments

**settings**      Tibble/data.frame with the settings of the empty summarised\_result. It has to contain at least result\_id column.

## Value

An empty summarised\_result object.

## Examples

```
library(omopgenerics)

emptySummarisedResult()
```

---

estimateTypeChoices    *Choices that can be present in estimate\_type column.*

---

**Description**

Choices that can be present in estimate\_type column.

**Usage**

```
estimateTypeChoices()
```

**Value**

A character vector with the options that can be present in estimate\_type column in the summarised\_result objects.

**Examples**

```
library(omopgenerics)

estimateTypeChoices()
```

---

exportSummarisedResult    *Export a summarised\_result object to a csv file.*

---

**Description**

Export a summarised\_result object to a csv file.

**Usage**

```
exportSummarisedResult(
  ...,
  minCellCount = 5,
  fileName = "results_{cdm_name}_{date}.csv",
  path = getwd()
)
```

**Arguments**

...	A set of summarised_result objects.
minCellCount	Minimum count for suppression purposes.
fileName	Name of the file that will be created. Use {cdm_name} to refer to the cdmName of the objects and {date} to add the export date.
path	Path where to create the csv file.

---

<code>getCohortId</code>	<i>Get the cohort definition id of a certain name</i>
--------------------------	---

---

**Description**

Get the cohort definition id of a certain name

**Usage**

```
getCohortId(cohort, cohortName = NULL)
```

**Arguments**

<code>cohort</code>	A <code>cohort_table</code> object.
<code>cohortName</code>	Names of the cohort of interest. If <code>NULL</code> all cohort names are shown.

**Value**

Cohort definition ids

---

<code>getCohortName</code>	<i>Get the cohort name of a certain cohort definition id</i>
----------------------------	--

---

**Description**

Get the cohort name of a certain cohort definition id

**Usage**

```
getCohortName(cohort, cohortId = NULL)
```

**Arguments**

<code>cohort</code>	A <code>cohort_table</code> object.
<code>cohortId</code>	Cohort definition id of interest. If <code>NULL</code> all cohort ids are shown.

**Value**

Cohort names

---

insertFromSource	<i>Convert a table that is not a cdm_table but have the same original source to a cdm_table. This Table is not meant to be used to insert tables in the cdm, please use insertTable instead.</i>
------------------	--

---

**Description****[Deprecated]****Usage**

```
insertFromSource(cdm, value)
```

**Arguments**

cdm	A cdm_reference object.
value	A table that shares source with the cdm_reference object.

**Value**

A table in the cdm\_reference environment

---

insertTable	<i>Insert a table to a cdm object.</i>
-------------	--

---

**Description**

Insert a table to a cdm object.

**Usage**

```
insertTable(cdm, name, table, overwrite = TRUE, temporary = FALSE)
```

**Arguments**

cdm	A cdm reference or the source of a cdm reference.
name	Name of the table to insert.
table	Table to insert to the cdm.
overwrite	Whether to overwrite an existent table.
temporary	Whether to create a temporary table.

**Value**

```
The cdm reference. library(omopgenerics) library(dplyr, warn.conflicts = FALSE)
person <- tibble( person_id = 1, gender_concept_id = 0, year_of_birth = 1990, race_concept_id
= 0, ethnicity_concept_id = 0 ) observation_period <- tibble( observation_period_id = 1, per-
son_id = 1, observation_period_start_date = as.Date("2000-01-01"), observation_period_end_date
= as.Date("2025-12-31"), period_type_concept_id = 0 ) cdm <- cdmFromTables( tables = list("person"
= person, "observation_period" = observation_period), cdmName = "my_example_cdm" )
x <- tibble(a = 1)
cdm <- insertTable(cdm = cdm, name = "new_table", table = x)
cdm$new_table
```

---

listSourceTables	<i>List tables that can be accessed though a cdm object.</i>
------------------	--

---

**Description**

List tables that can be accessed though a cdm object.

**Usage**

```
listSourceTables(cdm)
```

**Arguments**

cdm	A cdm reference or the source of a cdm reference.
-----	---

**Value**

A character vector with the names of tables.

---

newAchillesTable	<i>Create an achilles table from a cdm_table.</i>
------------------	---

---

**Description**

Create an achilles table from a cdm\_table.

**Usage**

```
newAchillesTable(table)
```

**Arguments**

table	A cdm_table.
-------	--------------



**Value**

An achilles\_table object

---

newCdmReference	cdm_reference <i>objects constructor</i>
-----------------	--

---

**Description**

cdm\_reference objects constructor

**Usage**

```
newCdmReference(tables, cdmName, cdmVersion = NULL, .softValidation = FALSE)
```

**Arguments**

tables	List of tables that are part of the OMOP Common Data Model reference.
cdmName	Name of the cdm object.
cdmVersion	Version of the cdm. Supported versions 5.3 and 5.4.
.softValidation	Whether to perform a soft validation of consistency. If set to FALSE, non overlapping observation periods are ensured.

**Value**

A cdm\_reference object.

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdmTables <- list(
  "person" = tibble(
    person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
    race_concept_id = 0, ethnicity_concept_id = 0
  ) |>
  newCdmTable(newLocalSource(), "person"),
  "observation_period" = tibble(
    observation_period_id = 1, person_id = 1,
    observation_period_start_date = as.Date("2000-01-01"),
    observation_period_end_date = as.Date("2025-12-31"),
    period_type_concept_id = 0
  ) |>
  newCdmTable(newLocalSource(), "observation_period")
)
cdm <- newCdmReference(tables = cdmTables, cdmName = "mock")
```

cdm

---

newCdmSource                      *Create a cdm source object.*

---

### **Description**

Create a cdm source object.

### **Usage**

```
newCdmSource(src, sourceType)
```

### **Arguments**

src	Source to a cdm object.
sourceType	Type of the source object.

### **Value**

A validated cdm source object.

---

newCdmTable                      *Create an cdm table.*

---

### **Description**

Create an cdm table.

### **Usage**

```
newCdmTable(table, src, name)
```

### **Arguments**

table	A table that is part of a cdm.
src	The source of the table.
name	The name of the table.

### **Value**

A cdm\_table object

---

newCodelist	<i>'codelist' object constructor</i>
-------------	--------------------------------------

---

**Description**

'codelist' object constructor

**Usage**

```
newCodelist(x)
```

**Arguments**

x                    A named list where each element contains a vector of concept IDs.

**Value**

A codelist object.

---

newCohortTable	<i>cohort_table objects constructor.</i>
----------------	--

---

**Description**

cohort\_table objects constructor.

**Usage**

```
newCohortTable(
  table,
  cohortSetRef = attr(table, "cohort_set"),
  cohortAttritionRef = attr(table, "cohort_attrition"),
  cohortCodelistRef = attr(table, "cohort_codelist"),
  .softValidation = FALSE
)
```

**Arguments**

table                    cdm\_table object with at least: cohort\_definition\_id, subject\_id, cohort\_start\_date, cohort\_end\_date.

cohortSetRef            Table with at least: cohort\_definition\_id, cohort\_name

cohortAttritionRef     Table with at least: cohort\_definition\_id, number\_subjects, number\_records, reason\_id, reason, excluded\_subjects, excluded\_records.

cohortCodelistRef

Table with at least: cohort\_definition\_id, codelist\_name, and concept\_id.

.softValidation

Whether to perform a soft validation of consistency. If set to FALSE four additional checks will be performed: 1) a check that cohort end date is not before cohort start date, 2) a check that there are no missing values in required columns, 3) a check that cohort duration is all within observation period, and 4) that there are no overlapping cohort entries

## Value

A cohort\_table object

## Examples

```
person <- dplyr::tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- dplyr::tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort1 <- dplyr::tibble(
  cohort_definition_id = 1, subject_id = 1,
  cohort_start_date = as.Date("2020-01-01"),
  cohort_end_date = as.Date("2020-01-10")
)
cdm <- cdmFromTables(
  tables = list(
    "person" = person,
    "observation_period" = observation_period,
    "cohort1" = cohort1
  ),
  cdmName = "test"
)
cdm
cdm$cohort1 <- newCohortTable(table = cdm$cohort1)
cdm
settings(cdm$cohort1)
attrition(cdm$cohort1)
cohortCount(cdm$cohort1)
```

---

newConceptSetExpression

*'conceptSetExpression' object constructor*

---

**Description**

'conceptSetExpression' object constructor

**Usage**

```
newConceptSetExpression(x)
```

**Arguments**

x                    a named list of tibbles, each of which containing concept set definitions

**Value**

A conceptSetExpression

---

<code>newLocalSource</code>	<i>A new local source for the cdm</i>
-----------------------------	---------------------------------------

---

**Description**

A new local source for the cdm

**Usage**

```
newLocalSource()
```

**Value**

A list in the format of a cdm source

**Examples**

```
library(omopgenerics)  
newLocalSource()
```

---

newOmopTable	<i>Create an omop table from a cdm table.</i>
--------------	---

---

**Description**

Create an omop table from a cdm table.

**Usage**

```
newOmopTable(table)
```

**Arguments**

table	A cdm_table.
-------	--------------

**Value**

An omop\_table object

---

newSummarisedResult	<i>'summarised_results' object constructor</i>
---------------------	--

---

**Description**

'summarised\_results' object constructor

**Usage**

```
newSummarisedResult(x, settings = attr(x, "settings"))
```

**Arguments**

x	Table.
settings	Settings for the summarised_result object.

**Value**

A summarised\_result object

**Examples**

```

library(dplyr)
library(omopgenerics)

x <- tibble(
  "result_id" = 1L,
  "cdm_name" = "cprd",
  "group_name" = "cohort_name",
  "group_level" = "acetaminophen",
  "strata_name" = "sex &&& age_group",
  "strata_level" = c("male &&& <40", "male &&& >=40"),
  "variable_name" = "number_subjects",
  "variable_level" = NA_character_,
  "estimate_name" = "count",
  "estimate_type" = "integer",
  "estimate_value" = c("5", "15"),
  "additional_name" = "overall",
  "additional_level" = "overall"
) |>
  newSummarisedResult()

x
settings(x)
summary(x)

x <- tibble(
  "result_id" = 1L,
  "cdm_name" = "cprd",
  "group_name" = "cohort_name",
  "group_level" = "acetaminophen",
  "strata_name" = "sex &&& age_group",
  "strata_level" = c("male &&& <40", "male &&& >=40"),
  "variable_name" = "number_subjects",
  "variable_level" = NA_character_,
  "estimate_name" = "count",
  "estimate_type" = "integer",
  "estimate_value" = c("5", "15"),
  "additional_name" = "overall",
  "additional_level" = "overall"
) |>
  newSummarisedResult(settings = tibble(
    result_id = 1L, result_type = "custom_summary", mock = TRUE, value = 5
  ))

x
settings(x)
summary(x)

```

---

---

omopColumns                      *Required columns that the standard tables in the OMOP Common Data Model must have.*

---

### **Description**

Required columns that the standard tables in the OMOP Common Data Model must have.

### **Usage**

```
omopColumns(table, required = TRUE, version = "5.3")
```

### **Arguments**

table	Table to see required columns.
required	Whether to include only required fields.
version	Version of the OMOP Common Data Model.

### **Value**

Character vector with the column names

### **Examples**

```
library(omopgenerics)
omopColumns("person")
```

---

omopTables                      *Standard tables that a cdm reference can contain in the OMOP Common Data Model.*

---

### **Description**

Standard tables that a cdm reference can contain in the OMOP Common Data Model.

### **Usage**

```
omopTables(version = "5.3")
```

### **Arguments**

version	Version of the OMOP Common Data Model.
---------	--

### **Value**

Standard tables



**Examples**

```
library(omopgenerics)

omopTables()
```

---

participants	<i>It returns the participants that contributed to a particular analysis</i>
--------------	--

---

**Description**

It returns the participants that contributed to a particular analysis

**Usage**

```
participants(result, ...)
```

**Arguments**

result	A result object.
...	...

**Value**

Table with the participants

---

print.cdm_reference	<i>Print a CDM reference object</i>
---------------------	-------------------------------------

---

**Description**

Print a CDM reference object

**Usage**

```
## S3 method for class 'cdm_reference'
print(x, ...)
```

**Arguments**

x	A cdm_reference object
...	Included for compatibility with generic. Not used.

**Value**

Invisibly returns the input

**Examples**

```

library(omopgenerics)

cdm <- cdmFromTables(
  tables = list(
    "person" = dplyr::tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = dplyr::tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

print(cdm)

```

---

print.codelist

*Print a codelist*


---

**Description**

Print a codelist

**Usage**

```

## S3 method for class 'codelist'
print(x, ...)

```

**Arguments**

x	A codelist
...	Included for compatibility with generic. Not used.

**Value**

Invisibly returns the input

**Examples**

```

codes <- list("disease X" = c(1, 2, 3), "disease Y" = c(4, 5))
codes <- newCodelist(codes)
print(codes)

```

---

```
print.conceptSetExpression
      Print a concept set expression
```

---

## Description

Print a concept set expression

## Usage

```
## S3 method for class 'conceptSetExpression'
print(x, ...)
```

## Arguments

x	A concept set expression
...	Included for compatibility with generic. Not used.

## Value

Invisibly returns the input

## Examples

```
asthma_cs <- list("asthma_narrow" = dplyr::tibble(
  "concept_id" = 1,
  "excluded" = FALSE,
  "descendants" = TRUE,
  "mapped" = FALSE
),
"asthma_broad" = dplyr::tibble(
  "concept_id" = c(1,2),
  "excluded" = FALSE,
  "descendants" = TRUE,
  "mapped" = FALSE
))
asthma_cs <- newConceptSetExpression(asthma_cs)
print(asthma_cs)
```

---

readSourceTable	<i>Read a table from the cdm_source and add it to the cdm.</i>
-----------------	--

---

**Description**

Read a table from the cdm\_source and add it to the cdm.

**Usage**

```
readSourceTable(cdm, name)
```

**Arguments**

cdm	A cdm reference.
name	Name of a table to read in the cdm_source space.

**Value**

A cdm\_reference with new table.

---

recordCohortAttrition	<i>Update cohort attrition.</i>
-----------------------	---------------------------------

---

**Description**

Update cohort attrition.

**Usage**

```
recordCohortAttrition(cohort, reason, cohortId = NULL)
```

**Arguments**

cohort	A cohort_table object.
reason	A character string.
cohortId	Cohort definition id of the cohort to update attrition. If NULL all cohort_definition_id are updated.

**Value**

cohort\_table with updated attrition.

**Examples**

```

library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort <- tibble(
  cohort_definition_id = c(1, 1, 1, 2),
  subject_id = 1,
  cohort_start_date = as.Date(c("2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01")),
  cohort_end_date = as.Date(c("2020-01-01", "2021-01-01", "2022-01-01", "2022-01-01")),
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "my_example_cdm",
  cohortTables = list("cohort1" = cohort)
)

cdm$cohort1
attrition(cdm$cohort1)

cdm$cohort1 <- cdm$cohort1 |>
  group_by(cohort_definition_id, subject_id) |>
  filter(cohort_start_date == min(cohort_start_date)) |>
  ungroup() |>
  compute(name = "cohort1", temporary = FALSE) |>
  recordCohortAttrition("Restrict to first observation")

cdm$cohort1
attrition(cdm$cohort1)

```

---

resultColumns

*Required columns that the result tables must have.*


---

**Description**

Required columns that the result tables must have.

**Usage**

```
resultColumns(table = "summarised_result")
```

**Arguments**

table            Table to see required columns.

**Value**

Required columns

**Examples**

```
library(omopgenerics)
```

```
resultColumns()
```

---

settings            *Get settings from an object.*

---

**Description**

Get settings from an object.

**Usage**

```
settings(x)
```

**Arguments**

x                Object

**Value**

A table with the settings of the object.

---

settings.cohort\_table    *Get cohort settings from a cohort\_table object.*

---

**Description**

Get cohort settings from a cohort\_table object.

**Usage**

```
## S3 method for class 'cohort_table'
settings(x)
```

**Arguments**

x                    A cohort\_table object.

**Value**

A table with the details of the cohort settings.

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cohort <- tibble(
  cohort_definition_id = 1,
  subject_id = 1,
  cohort_start_date = as.Date("2010-01-01"),
  cohort_end_date = as.Date("2012-01-01")
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test",
  cohortTables = list("my_cohort" = cohort)
)

settings(cdm$my_cohort)

cdm$my_cohort <- cdm$my_cohort |>
  newCohortTable(cohortSetRef = tibble(
    cohort_definition_id = 1, cohort_name = "new_name"
  ))

settings(cdm$my_cohort)
```

---

settings.summarised\_result

*Get settings from a summarised\_result object.*

---

**Description**

Get settings from a summarised\_result object.

**Usage**

```
## S3 method for class 'summarised_result'  
settings(x)
```

**Arguments**

x                    A summarised\_result object.

**Value**

A table with the settings.

**Examples**

```
library(omopgenerics)  
library(dplyr, warn.conflicts = FALSE)  
  
person <- tibble(  
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,  
  race_concept_id = 0, ethnicity_concept_id = 0  
)  
observation_period <- tibble(  
  observation_period_id = 1, person_id = 1,  
  observation_period_start_date = as.Date("2000-01-01"),  
  observation_period_end_date = as.Date("2025-12-31"),  
  period_type_concept_id = 0  
)  
cohort <- tibble(  
  cohort_definition_id = 1,  
  subject_id = 1,  
  cohort_start_date = as.Date("2010-01-01"),  
  cohort_end_date = as.Date("2012-01-01")  
)  
cdm <- cdmFromTables(  
  tables = list("person" = person, "observation_period" = observation_period),  
  cdmName = "test",  
  cohortTables = list("my_cohort" = cohort)  
)  
  
result <- summary(cdm$my_cohort)  
  
settings(result)
```



---

sourceType	<i>Get the source type of a cdm_source.</i>
------------	---

---

**Description**

Get the source type of a cdm\_source.

**Usage**

```
sourceType(cdmSource)
```

**Arguments**

cdmSource      A cdm\_source object.

**Value**

A character vector that defines the type of cdm\_source.

---

summary.cdm_reference	<i>Summary a cdm reference</i>
-----------------------	--------------------------------

---

**Description**

Summary a cdm reference

**Usage**

```
## S3 method for class 'cdm_reference'  
summary(object, ...)
```

**Arguments**

object            A cdm reference object.  
...                For compatibility (not used).

**Value**

A summarised\_result object with a summary of the data contained in the cdm.

## Examples

```
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test"
)

summary(cdm)
```

---

summary.cohort\_table *Summary a generated cohort set*

---

## Description

Summary a generated cohort set

## Usage

```
## S3 method for class 'cohort_table'
summary(object, ...)
```

## Arguments

object            A generated cohort set object.  
...                For compatibility (not used).

## Value

A summarised\_result object with a summary of a cohort\_table.

## Examples

```
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
  race_concept_id = 0, ethnicity_concept_id = 0
```

```

)
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test",
  cohortTables = list("cohort1" = tibble(
    cohort_definition_id = 1,
    subject_id = 1,
    cohort_start_date = as.Date("2010-01-01"),
    cohort_end_date = as.Date("2010-01-05")
  ))
)

summary(cdm$cohort1)

```

---

```
summary.summarised_result
```

*Summary a summarised\_result*

---

## Description

Summary a summarised\_result

## Usage

```
## S3 method for class 'summarised_result'
summary(object, ...)
```

## Arguments

object            A summarised\_result object.  
 ...              For compatibility (not used).

## Value

A summary of the result\_types contained in a summarised\_result object.

## Examples

```
library(dplyr, warn.conflicts = FALSE)

person <- tibble(
  person_id = 1, gender_concept_id = 0, year_of_birth = 1990,
```

```
    race_concept_id = 0, ethnicity_concept_id = 0
  )
observation_period <- tibble(
  observation_period_id = 1, person_id = 1,
  observation_period_start_date = as.Date("2000-01-01"),
  observation_period_end_date = as.Date("2025-12-31"),
  period_type_concept_id = 0
)
cdm <- cdmFromTables(
  tables = list("person" = person, "observation_period" = observation_period),
  cdmName = "test"
)

result <- summary(cdm)

summary(result)
```

---

suppress

*Function to suppress counts in result objects*

---

### Description

Function to suppress counts in result objects

### Usage

```
suppress(result, minCellCount = 5)
```

### Arguments

result	Result object
minCellCount	Minimum count of records to report results.

### Value

Table with suppressed counts

---

 suppress.summarised\_result

*Function to suppress counts in result objects*


---

## Description

Function to suppress counts in result objects

## Usage

```
## S3 method for class 'summarised_result'
suppress(result, minCellCount = 5)
```

## Arguments

result            summarised\_result object.  
 minCellCount    Minimum count of records to report results.

## Value

summarised\_result with suppressed counts.

## Examples

```
library(dplyr, warn.conflicts = FALSE)
library(omopgenerics)

my_result <- tibble(
  "result_id" = "1",
  "cdm_name" = "mock",
  "result_type" = "summarised_characteristics",
  "package_name" = "omopgenerics",
  "package_version" = as.character(utils::packageVersion("omopgenerics")),
  "group_name" = "overall",
  "group_level" = "overall",
  "strata_name" = c(rep("overall", 6), rep("sex", 3)),
  "strata_level" = c(rep("overall", 6), "male", "female", "female"),
  "variable_name" = c("number records", "age_group", "age_group",
    "age_group", "age_group", "my_variable", "number records", "age_group",
    "age_group"),
  "variable_level" = c(NA, "<50", "<50", ">=50", ">=50", NA, NA,
    "<50", "<50"),
  "estimate_name" = c("count", "count", "percentage", "count", "percenatge",
    "random", "count", "count", "percentage"),
  "estimate_type" = c("integer", "integer", "percentage", "integer",
    "percentage", "numeric", "integer", "integer", "percentage"),
  "estimate_value" = c("10", "5", "50", "3", "30", "1", "3", "12", "6"),
  "additional_name" = "overall",
  "additional_level" = "overall"
```

```
)  
my_result <- newSummarisedResult(my_result)  
my_result |> glimpse()  
my_result <- suppress(my_result, minCellCount = 5)  
my_result |> glimpse()
```

---

tableName	<i>Get the table name of a cdm_table.</i>
-----------	---

---

### Description

Get the table name of a cdm\_table.

### Usage

```
tableName(table)
```

### Arguments

table            A cdm\_table.

### Value

A character with the name.

### Examples

```
library(omopgenerics)  
library(dplyr, warn.conflicts = FALSE)  
  
cdm <- cdmFromTables(  
  tables = list(  
    "person" = tibble(  
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,  
      race_concept_id = 0, ethnicity_concept_id = 0  
    ),  
    "observation_period" = tibble(  
      observation_period_id = 1:3, person_id = 1:3,  
      observation_period_start_date = as.Date("2000-01-01"),  
      observation_period_end_date = as.Date("2025-12-31"),  
      period_type_concept_id = 0  
    )  
  ),  
  cdmName = "mock"  
)  
  
tableName(cdm$person)
```

---

tableSource	<i>Get the table source of a cdm_table.</i>
-------------	---

---

### Description

Get the table source of a cdm\_table.

### Usage

```
tableSource(table)
```

### Arguments

table            A cdm\_table.

### Value

A cdm\_source object.

### Examples

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

tableSource(cdm$person)
```

---

tmpPrefix	<i>Create a temporary prefix for tables, that contains a unique prefix that starts with tmp.</i>
-----------	--

---

**Description**

Create a temporary prefix for tables, that contains a unique prefix that starts with tmp.

**Usage**

```
tmpPrefix()
```

**Value**

A temporary prefix.

**Examples**

```
library(omopgenerics)
tmpPrefix()
```

---

toSnakeCase	<i>Convert a character vector to snake case</i>
-------------	---

---

**Description**

Convert a character vector to snake case

**Usage**

```
toSnakeCase(x)
```

**Arguments**

x	Character vector to convert
---	-----------------------------

**Value**

A snake\_case vector

**Examples**

```
toSnakeCase("myVariable")
toSnakeCase(c("cohort1", "Cohort22b"))
```



---

uniqueTableName	<i>Create a unique table name</i>
-----------------	-----------------------------------

---

**Description**

Create a unique table name

**Usage**

```
uniqueTableName(prefix = "")
```

**Arguments**

prefix	Prefix for the table names.
--------	-----------------------------

**Value**

A string that can be used as a dbplyr temp table name

**Examples**

```
library(omopgenerics)
uniqueTableName()
```

---

validateNameLevel	<i>Validate if two columns are valid Name-Level pair.</i>
-------------------	---

---

**Description**

Validate if two columns are valid Name-Level pair.

**Usage**

```
validateNameLevel(  
  x,  
  nameColumn,  
  levelColumn,  
  sep = " and | &&& ",  
  warn = FALSE  
)
```

**Arguments**

x	A tibble.
nameColumn	Column name of the name.
levelColumn	Column name of the level.
sep	Separation pattern.
warn	Whether to throw a warning (TRUE) or an error (FALSE).

---

[[.cdm\_reference      *Subset a cdm reference object.*

---

**Description**

Subset a cdm reference object.

**Usage**

```
## S3 method for class 'cdm_reference'
x[[name]]
```

**Arguments**

x	A cdm reference
name	The name or index of the table to extract from the cdm object.

**Value**

A single cdm table reference

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
)
```

```

    cdmName = "mock"
  )
  cdm[["person"]]

```

---

[[<- .cdm\_reference      *Assign a table to a cdm reference.*

---

### Description

Assign a table to a cdm reference.

### Usage

```

## S3 replacement method for class 'cdm_reference'
cdm[[name]] <- value

```

### Arguments

cdm	A cdm reference.
name	Name where to assign the new table.
value	Table with the same source than the cdm object.

### Value

The cdm reference.

---

\$.cdm\_reference      *Subset a cdm reference object.*

---

### Description

Subset a cdm reference object.

### Usage

```

## S3 method for class 'cdm_reference'
x$name

```

### Arguments

x	A cdm reference.
name	The name of the table to extract from the cdm object.

**Value**

A single cdm table reference

**Examples**

```
library(omopgenerics)
library(dplyr, warn.conflicts = FALSE)

cdm <- cdmFromTables(
  tables = list(
    "person" = tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

cdm$person
```

---

\$<-.cdm\_reference      *Assign an table to a cdm reference.*

---

**Description**

Assign an table to a cdm reference.

**Usage**

```
## S3 replacement method for class 'cdm_reference'
cdm$name <- value
```

**Arguments**

cdm	A cdm reference.
name	Name where to assign the new table.
value	Table with the same source than the cdm object.

**Value**

The cdm reference.

## Examples

```
library(omopgenerics)

cdm <- cdmFromTables(
  tables = list(
    "person" = dplyr::tibble(
      person_id = c(1, 2, 3), gender_concept_id = 0, year_of_birth = 1990,
      race_concept_id = 0, ethnicity_concept_id = 0
    ),
    "observation_period" = dplyr::tibble(
      observation_period_id = 1:3, person_id = 1:3,
      observation_period_start_date = as.Date("2000-01-01"),
      observation_period_end_date = as.Date("2025-12-31"),
      period_type_concept_id = 0
    )
  ),
  cdmName = "mock"
)

cdm$person
```

# Index

[[.cdm\_reference, 58  
[[<- .cdm\_reference, 59  
\$.cdm\_reference, 59  
\$<- .cdm\_reference, 60

achillesColumns, 3  
achillesTables, 4  
attrition, 5  
attrition.cohort\_table, 5

bind, 6  
bind.cohort\_table, 7  
bind.summarised\_result, 8

cdmFromTables, 9  
cdmName, 10  
cdmReference, 11  
cdmSelect, 12  
cdmSource, 12  
cdmSourceType, 13  
cdmTableFromSource, 14  
cdmVersion, 15  
checkCohortRequirements, 16  
cohortCodelist, 17  
cohortColumns, 18  
cohortCount, 19  
cohortTables, 20  
collect.cdm\_reference, 21  
collect.cohort\_table, 22  
compute.cdm\_table, 22

dropSourceTable, 23  
dropTable, 23

emptyAchillesTable, 24  
emptyCdmReference, 25  
emptyCodelist, 26  
emptyCohortTable, 26  
emptyOmopTable, 27  
emptySummarisedResult, 28  
estimateTypeChoices, 29

exportSummarisedResult, 29

getCohortId, 30  
getCohortName, 30

insertFromSource, 31  
insertTable, 31

listSourceTables, 32

newAchillesTable, 32  
newCdmReference, 33  
newCdmSource, 34  
newCdmTable, 34  
newCodelist, 35  
newCohortTable, 35  
newConceptSetExpression, 36  
newLocalSource, 37  
newOmopTable, 38  
newSummarisedResult, 38

omopColumns, 39  
omopTables, 40

participants, 41  
print.cdm\_reference, 41  
print.codelist, 42  
print.conceptSetExpression, 43

readSourceTable, 44  
recordCohortAttrition, 44  
resultColumns, 45

settings, 46  
settings.cohort\_table, 46  
settings.summarised\_result, 47  
sourceType, 49  
summary.cdm\_reference, 49  
summary.cohort\_table, 50  
summary.summarised\_result, 51  
suppress, 52

`suppress.summarised_result`, [53](#)

`tableName`, [54](#)

`tableSource`, [55](#)

`tmpPrefix`, [56](#)

`toSnakeCase`, [56](#)

`uniqueTableName`, [57](#)

`validateNameLevel`, [57](#)