

SQL CTE Guide PDF Page

SQL CTE Guide PDF Page: Content + <head> (Uncodemy)

FINAL URL: <https://uncodemy.com/sql-cte-guide-pdf>

Publish this page at exactly this URL (non-www, no trailing slash). Canonical, og:url and sitemap entry must match it.

Prepared: 26 October 2026 · **Primary keyword:** sql cte guide pdf · **Search volume:** not checked (keyword tool out of credits); verify in Search Console.

Read first (5 flags):

1.

Your course PDF title, page count and size were not shared. Download cards are fill-in slots ([CONFIRM]). Card 1 is your course PDF; cards 2 and 3 only if you have more files.

2.

One unique file per button. The same course PDF repeated on many pages is fine as a resource, but each page's text must stay unique and useful without the file.

3.

Free or gated? Not stated. The page says "Download", not "Free". If a form is used, add a consent checkbox and Privacy Policy link (DPDP Act 2023).

4.

No made-up stats or testimonials; no placement or salary claims without a named, dated source.

5.

Author and reviewer are placeholders; use real named trainers and one real "Last reviewed" date.

PART 1: Page plan

Item	Decision
Final URL	https://uncodemy.com/sql-cte-guide-pdf
Intent (anti-cannibalisation)	COMMON TABLE EXPRESSIONS (WITH clauses). Subqueries have their own page; window functions are on the window functions page.
Why not thin	Explainer-style sections, FAQs and cross-links make the page useful even without a download
Schema	WebSite, Organization, Breadcrumb, CollectionPage, FAQPage (only FAQs visible on the page). PDF <code>ItemList</code> provided commented, to enable after titles are final
Share image	https://uncodemy.com/image/logo-og.jpg for now; custom 1200x630 card recommended

PART 2: Page content (copy-paste ready)

SEO Title (28 chars):

SQL CTE Guide PDF | Uncodemy

Meta Description (132 chars):

Download the SQL CTE guide PDF from Uncodemy: WITH clause syntax, multiple and recursive CTEs, readability tips and tested examples.

H1: SQL CTE Guide PDF Downloads **Breadcrumb:** Home > SQL CTE Guide PDF **Final URL:** <https://uncodemy.com/sql-cte-guide-pdf>

Byline: Prepared by the Uncodemy trainer team [CONFIRM: author name and role] · Reviewed by [CONFIRM: reviewer name and role] · Last reviewed: [DATE]

Quick Answer

This page offers Uncodemy's SQL CTE guide as a PDF download. A common table expression, or CTE, is a named temporary result defined with WITH that you can use in the query that follows. CTEs make complex queries readable by breaking them into steps. The guide covers syntax, chaining, recursive CTEs, comparisons with subqueries and common mistakes.

Download SQL CTE Guide PDF

[CONFIRM: free or gated, and whether details are needed before download]

PDF 1 (main): [CONFIRM TITLE: your Data Analytics course PDF]
[CONFIRM: one line on what it covers and who it is for] · [N] pages · [X] MB · PDF · Updated [DATE]
[Download PDF] → <https://uncodemy.com/pdf/sql-cte-guide.pdf> (your course PDF, uploaded under this file name)

PDF 2 (optional): [CONFIRM TITLE]
Only if you have a second PDF that genuinely fits this topic. **[Download PDF]** → <https://uncodemy.com/pdf/sql-cte-guide-2.pdf>

PDF 3 (optional): [CONFIRM TITLE]

Same rule. **[Download PDF]** → <https://uncodemy.com/pdf/sql-cte-guide-3.pdf>

Each button must open its own file. Do not point several buttons at one file. Delete unused optional cards.

What is a CTE?

A common table expression is a named result set that exists only for the duration of one statement. You define it with the WITH keyword before the main query, give it a name and a query, and then refer to it as if it were a table. The main benefit is clarity. Instead of nesting a subquery inside a subquery, you write each step as a named block and read the query from top to bottom like a recipe. CTEs do not store data permanently and, in most databases, they do not by themselves make a query faster; their value is in readability and in enabling recursion. The examples use these tables: customers(customer_id, name, city, signup_date), orders(order_id, customer_id, order_date, status, order_total), order_items(order_id, product_id, quantity, unit_price) and products(product_id, product_name, category). Dates are text in YYYY-MM-DD form, so the examples run in SQLite and need small changes in other databases.

Basic syntax

Revenue per customer calculated in a CTE, then filtered in the main query.

```
WITH customer_revenue AS (  
  SELECT customer_id, SUM(order_total) AS revenue  
  FROM orders  
  WHERE status <> 'cancelled'  
  GROUP BY customer_id  
)  
SELECT customer_id, revenue  
FROM customer_revenue  
WHERE revenue > 500  
ORDER BY revenue DESC;
```

Chaining several CTEs

You can define several CTEs one after another, separated by commas, and later ones can use earlier ones. This is where CTEs shine: each step does one thing, has a clear name and can be checked on its own by running the query up to that point. Name steps by what they contain, such as monthly_revenue or first_orders, rather than by number, so the query documents itself. If a step is wrong, you can select from that CTE alone to inspect it before moving on.

Two chained CTEs

Monthly revenue first, then the change from the previous month using a window function in the second step.

```
WITH monthly AS (  
  SELECT strftime('%Y-%m', order_date) AS month,  
         SUM(order_total) AS revenue  
  FROM orders  
  WHERE status <> 'cancelled'  
  GROUP BY strftime('%Y-%m', order_date)  
)  
,  
with_change AS (  
  SELECT month,  
         revenue,  
         LAG(revenue) OVER (ORDER BY month) AS prev_revenue  
  FROM monthly  
)  
SELECT month,  
       revenue,  
       prev_revenue,  
       ROUND((revenue - prev_revenue) * 100.0 / prev_revenue, 1) AS pct_change  
FROM with_change  
ORDER BY month;
```

Recursive CTEs

A recursive CTE refers to itself. It has two parts joined by UNION ALL: an anchor query that returns the starting rows and a recursive query that uses the CTE's own previous result to produce the next rows. The recursion stops when the recursive part returns no new rows. Typical uses are generating a series of dates or numbers and walking hierarchies such as an organisation chart or a category tree. Always make sure the recursion terminates, for example by moving toward an end condition, because a mistake can create an endless loop; many databases enforce a maximum recursion depth.

A recursive CTE that generates dates

Produces every date in a range so you can show days with zero orders. Syntax for generating dates differs a little between databases.

```
WITH RECURSIVE days AS (  
  SELECT DATE('2026-09-01') AS d  
  UNION ALL  
  SELECT DATE(d, '+1 day') FROM days WHERE d < '2026-09-07'  
)  
SELECT days.d AS day,  
       COALESCE(COUNT(o.order_id), 0) AS orders  
FROM days  
LEFT JOIN orders o ON o.order_date = days.d AND o.status <> 'cancelled'  
GROUP BY days.d  
ORDER BY days.d;
```

CTE, subquery, view and temporary table

Option	Lifetime	Best for

CTE	One statement	Readable multi-step queries and recursion
Subquery	One statement	Short, single-use logic
View	Stored in the database	Reusable logic shared by many queries
Temporary table	Session or until dropped	Large intermediate results reused several times

When to use a CTE

- The query has more than one logical step.
- The same intermediate result is referred to more than once.
- You need recursion for dates or hierarchies.
- A teammate will have to read and maintain the query.

Common CTE mistakes

- **Assuming CTEs are always faster:** in many databases they behave like inline subqueries, and behaviour differs by engine and version.
- **Giving vague names** like a, b and c.
- **Forgetting the comma** between several CTEs, or placing it after the last one.
- **Using a CTE where a simple join is clearer.**
- **Infinite recursion** from a missing end condition.

A checklist for writing a CTE query

1. Write down the grain of the final result.
2. Create one CTE per logical step and name it clearly.
3. Run each CTE separately and check its row count.
4. Join the CTEs in the final select.
5. Compare the final totals with a simple independent query.

Practice tasks

Try three tasks on your own data. First, write a CTE that returns the top five customers by revenue and join it to their cities. Second, chain two CTEs to compute each customer's first order date and then count customers by first order month. Third, write a recursive CTE that lists every date in a month and use it to report days with zero orders. Explain each query aloud in plain words; if you can, you have understood it.

Related Unacademy resources

- [SQL Subqueries Guide PDF](#)
- [SQL Analytics Functions PDF](#)
- [SQL Window Functions PDF](#)
- [SQL for Data Analytics Guide PDF](#)

Learn with a trainer [use only the line that is true]

- **Data Analytics course:** <https://unacademy.com/course/data-analytics-training-course-in-noida>. URL verified on the live site.

(One soft call to action: course link or free demo. No pop-up.)

About these PDFs

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to [CONFIRM: feedback email] and the trainer team will review it.

Frequently Asked Questions

What is a CTE in SQL?

A common table expression is a named temporary result defined with WITH that you can reference in the following query, used to make complex queries readable.

What is the difference between a CTE and a subquery?

Both define intermediate results for one statement. A CTE is named, can be referenced several times and supports recursion, while a subquery is nested inline.

Is a CTE faster than a subquery?

Not necessarily. Performance depends on the database and its optimiser. CTEs mainly improve readability.

What is a recursive CTE?

A CTE that refers to itself, with an anchor query and a recursive part joined by UNION ALL, used for generating series or traversing hierarchies.

Can I use multiple CTEs in one query?

Yes. Define them one after another separated by commas, and later ones can use earlier ones.

Are these PDFs free to download?

[CONFIRM: state clearly whether the PDFs are free, or whether a short form is needed first.]

The last FAQ contains [CONFIRM], so it is NOT in the FAQ schema. Fill it and keep it on the page, or delete it.

PART 3: <head> section (copy-paste ready)

Final URL: <https://uncodemy.com/sql-cte-guide-pdf>

```
<!-- =====
UNCODEMY <head> | FINAL URL: https://uncodemy.com/sql-cte-guide-pdf
Prepared 26 October 2026. Paste into the page <head>; keep your site-wide favicon, fonts, analytics.
BEFORE GO-LIVE: (1) set dateModified in the JSON-LD to the real publish date;
(2) replace og:image with a 1200x630 share card when available;
(3) once PDF titles are final, uncomment the ItemList block at the bottom.
===== -->

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>SQL CTE Guide PDF | Uncodemy</title>
<meta name="description" content="Download the SQL CTE guide PDF from Uncodemy: WITH clause syntax, multiple and recursive CTEs, readability tips and tested
examples.">
<link rel="canonical" href="https://uncodemy.com/sql-cte-guide-pdf">
<meta name="robots" content="index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:-1">
<meta name="author" content="Uncodemy">
<meta name="publisher" content="Uncodemy Edutech Pvt. Ltd.">
<meta name="language" content="en-IN">
<link rel="alternate" hreflang="en-in" href="https://uncodemy.com/sql-cte-guide-pdf">
<link rel="alternate" hreflang="x-default" href="https://uncodemy.com/sql-cte-guide-pdf">

<!-- Open Graph -->
<meta property="og:type" content="website">
<meta property="og:site_name" content="Uncodemy">
<meta property="og:locale" content="en-IN">
<meta property="og:url" content="https://uncodemy.com/sql-cte-guide-pdf">
<meta property="og:title" content="SQL CTE Guide PDF | Uncodemy">
<meta property="og:description" content="Download the SQL CTE guide PDF from Uncodemy: WITH clause syntax, multiple and recursive CTEs, readability tips and
tested examples.">
<meta property="og:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta property="og:image:alt" content="Uncodemy SQL CTE Guide PDF study material">

<!-- X (Twitter) -->
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:site" content="@uncodemy">
<meta name="twitter:title" content="SQL CTE Guide PDF | Uncodemy">
<meta name="twitter:description" content="Download the SQL CTE guide PDF from Uncodemy: WITH clause syntax, multiple and recursive CTEs, readability tips and
tested examples.">
<meta name="twitter:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta name="twitter:image:alt" content="Uncodemy SQL CTE Guide PDF study material">

<!-- Structured data: one @graph (WebSite, Organization, Breadcrumb, CollectionPage, FAQPage) -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "WebSite",
      "@id": "https://uncodemy.com/#website",
      "url": "https://uncodemy.com/",
      "name": "Uncodemy",
      "publisher": {
        "@id": "https://uncodemy.com/#organization"
      },
      "inLanguage": "en-IN"
    },
    {
      "@type": "EducationalOrganization",
      "@id": "https://uncodemy.com/#organization",
      "name": "Uncodemy",
      "legalName": "Uncodemy Edutech Pvt. Ltd.",
      "url": "https://uncodemy.com/",
      "logo": {
        "@type": "ImageObject",
        "url": "https://uncodemy.com/uc-layout/img/Logo.webp"
      },
      "email": "info@uncodemy.com",
      "identifier": {
        "@type": "PropertyValue",
        "name": "CIN",
        "value": "U85499UP2024PTC195902"
      },
      "contactPoint": [
        {
          "@type": "ContactPoint",
          "telephone": "+91-9818366550",
          "contactType": "admissions",
          "areaServed": "IN",
          "availableLanguage": [
            "en",
            "hi"
          ]
        }
      ],
      "sameAs": [
        "https://x.com/uncodemy",
        "https://www.facebook.com/uncodemyofficial/",
        "https://www.linkedin.com/company/uncodemy/",
        "https://www.instagram.com/uncodemyofficial/"
      ]
    },
    {
      "@type": "BreadcrumbList",
      "@id": "https://uncodemy.com/sql-cte-guide-pdf#breadcrumb",
      "itemListElement": [
        {
          "@type": "ListItem",
          "position": 1,
          "name": "Home",
          "item": "https://uncodemy.com/"
        },
        {
          "@type": "ListItem",
          "position": 2,
```

```

        "name": "SQL CTE Guide PDF",
        "item": "https://uncodemy.com/sql-cte-guide-pdf"
    }
},
{
    "@type": "CollectionPage",
    "@id": "https://uncodemy.com/sql-cte-guide-pdf#webpage",
    "url": "https://uncodemy.com/sql-cte-guide-pdf",
    "name": "SQL CTE Guide PDF Downloads",
    "description": "Download the SQL CTE guide PDF from Uncodemy: WITH clause syntax, multiple and recursive CTEs, readability tips and tested examples.",
    "inLanguage": "en-IN",
    "isPartOf": {
        "@id": "https://uncodemy.com/#website"
    },
    "publisher": {
        "@id": "https://uncodemy.com/#organization"
    },
    "breadcrumb": {
        "@id": "https://uncodemy.com/sql-cte-guide-pdf#breadcrumb"
    },
    "about": {
        "@type": "Thing",
        "name": "SQL CTE Guide"
    },
    "primaryImageOfPage": {
        "@type": "ImageObject",
        "url": "https://uncodemy.com/image/logo-og.jpg"
    },
    "dateModified": "2026-10-26"
},
{
    "@type": "FAQPage",
    "@id": "https://uncodemy.com/sql-cte-guide-pdf#faq",
    "inLanguage": "en-IN",
    "isPartOf": {
        "@id": "https://uncodemy.com/sql-cte-guide-pdf#webpage"
    },
    "mainEntity": [
        {
            "@type": "Question",
            "name": "What is a CTE in SQL?",
            "acceptedAnswer": {
                "@type": "Answer",
                "text": "A common table expression is a named temporary result defined with WITH that you can reference in the following query, used to make comp
queries readable."
            }
        },
        {
            "@type": "Question",
            "name": "What is the difference between a CTE and a subquery?",
            "acceptedAnswer": {
                "@type": "Answer",
                "text": "Both define intermediate results for one statement. A CTE is named, can be referenced several times and supports recursion, while a subq
is nested inline."
            }
        },
        {
            "@type": "Question",
            "name": "Is a CTE faster than a subquery?",
            "acceptedAnswer": {
                "@type": "Answer",
                "text": "Not necessarily. Performance depends on the database and its optimiser. CTEs mainly improve readability."
            }
        },
        {
            "@type": "Question",
            "name": "What is a recursive CTE?",
            "acceptedAnswer": {
                "@type": "Answer",
                "text": "A CTE that refers to itself, with an anchor query and a recursive part joined by UNION ALL, used for generating series or traversing
hierarchies."
            }
        },
        {
            "@type": "Question",
            "name": "Can I use multiple CTEs in one query?",
            "acceptedAnswer": {
                "@type": "Answer",
                "text": "Yes. Define them one after another separated by commas, and later ones can use earlier ones."
            }
        }
    ]
}
]
}
</script>

<!-- ADD AFTER THE PDF TITLES ARE FINAL (remove this wrapper, fill every [...] value):
<script type="application/ld+json">
{
    "@context": "https://schema.org",
    "@type": "ItemList",
    "@id": "https://uncodemy.com/sql-cte-guide-pdf#pdfs",
    "name": "SQL CTE Guide PDF downloads",
    "numberOfItems": 3,
    "itemListElement": [
        {
            "@type": "ListItem",
            "position": 1,
            "item": {
                "@type": "DigitalDocument",
                "name": "[PDF 1 TITLE]",
                "description": "[PDF 1 ONE-LINE DESCRIPTION]",
                "encodingFormat": "application/pdf",
                "url": "https://uncodemy.com/pdf/sql-cte-guide.pdf",
                "inLanguage": "en",
                "publisher": {
                    "@id": "https://uncodemy.com/#organization"
                }
            }
        }
    ]
}

```

```

    },
    {
      "@type": "ListItem",
      "position": 2,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 2 TITLE]",
        "description": "[PDF 2 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/sql-cte-guide-2.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    },
    {
      "@type": "ListItem",
      "position": 3,
      "item": {
        "@type": "DigitalDocument",
        "name": "[PDF 3 TITLE]",
        "description": "[PDF 3 ONE-LINE DESCRIPTION]",
        "encodingFormat": "application/pdf",
        "url": "https://uncodemy.com/pdf/sql-cte-guide-3.pdf",
        "inLanguage": "en",
        "publisher": {
          "@id": "https://uncodemy.com/#organization"
        }
      }
    }
  ]
}
</script>
-->

```

DEVELOPER CHECKLIST

- ☐ Final URL live at <https://uncodemy.com/sql-cte-guide-pdf>; canonical, og:url and sitemap identical.
- ☐ Each button: `<a href="/pdf/sql-cte-guide.pdf" download type="application/pdf">`, its own file, size shown.
- ☐ If a form precedes download: consent checkbox, Privacy Policy link, nothing pre-ticked.
- ☐ One `<h1>`; real HTML tables; code samples in `<pre><code>`.
- ☐ Remove any `<meta name="keywords">` from the template.
- ☐ Test in Google Rich Results Test and validator.schema.org after publishing.
- ☐ Add to sitemap and any PDF/resources hub.

STILL NEEDED FROM YOU

#	Item
1	Course PDF title, description, page count, size, and the file URL for each page
2	Free or gated (and form fields if gated)
3	Author, reviewer, publish and last-reviewed dates
4	Course URL for the CTA (lines marked <code>[CONFIRM]</code>)
5	Feedback email
6	Confirmation that Uncodemy owns the rights to every PDF