



Vibe Coding with SQL for Fun and "Profit"

Wes Osborn

Central Library Consortium

Agenda

Introductions

How AI works

How to make AI work for you

What's next



ChatGPT



Gemini

Examples will use
ChatGPT & Gemini
(Google) but most
AI tools are similar

Caution read terms carefully

Make sure you aren't submitting
any sensitive data



Complex Feelings



AI's impact on
humanity



AI's impact on
energy usage



AI's impact on water
usage

Why is Everyone So Wrong About AI Water Use??

YouTube

YouTube · Hank Green · Dec 8, 2025



Why is Everyone So Wrong About AI Water Use??

Hank Green



0:23 / 23:59



CODE

Surprise! There is no profit

Microsoft just revealed that OpenAI lost more than \$11.5B last quarter

Satya has also delivered Sam most of the cash he promised

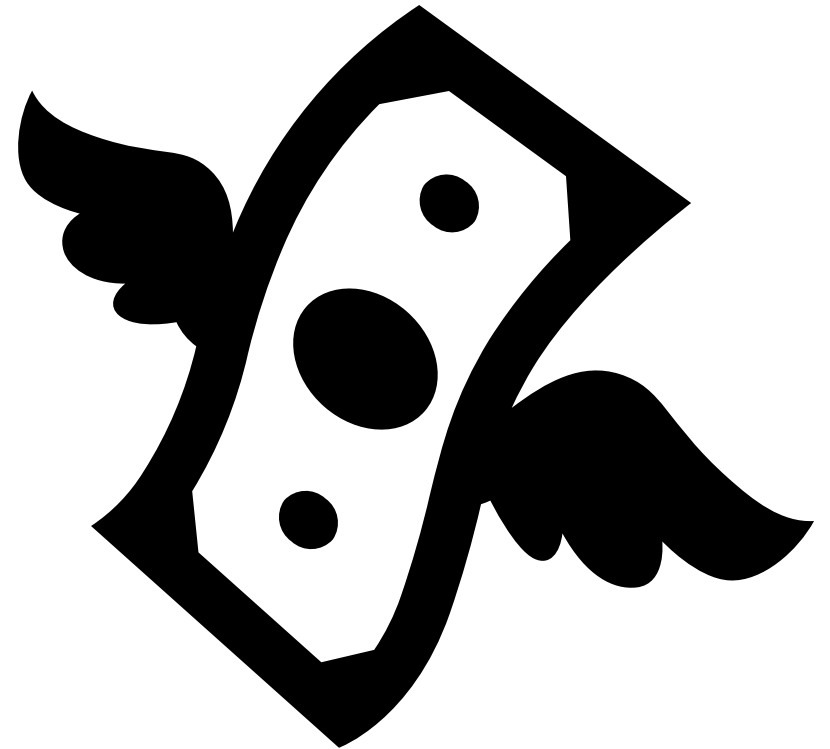
 [Matt Rosoff](#)

Wed 29 Oct 2025 // 23:17 UTC

UPDATED Microsoft reported earnings for the quarter ended Sept. 30 on Wednesday after market close and buried in its financial filings were a couple of passages suggesting that OpenAI suffered a net loss of \$11.5 billion or more during the quarter.

Let's look first at page 9 of the [official earnings filing](#) with the US Securities and Exchange Commission, which includes the following passage:

We have an investment in OpenAI Global, LLC (“OpenAI”) and have made total funding commitments of \$13 billion, of which \$11.6 billion has been funded as of September 30, 2025. The investment is accounted for under the equity method of accounting, with our share of OpenAI’s income or loss recognized in other income (expense), net.



Original Computers



Were purpose built



A calculator



Playing Pong



```
*****  
* APPLESOFT IIc FLOATING POINT BASIC *  
* APRIL 1978 *  
*****  
  
COPYRIGHT 1978 APPLE COMPUTER  
  
COPYRIGHT 1976 BY MICROSOFT  
  
ALL RIGHTS RESERVED  
  
■
```

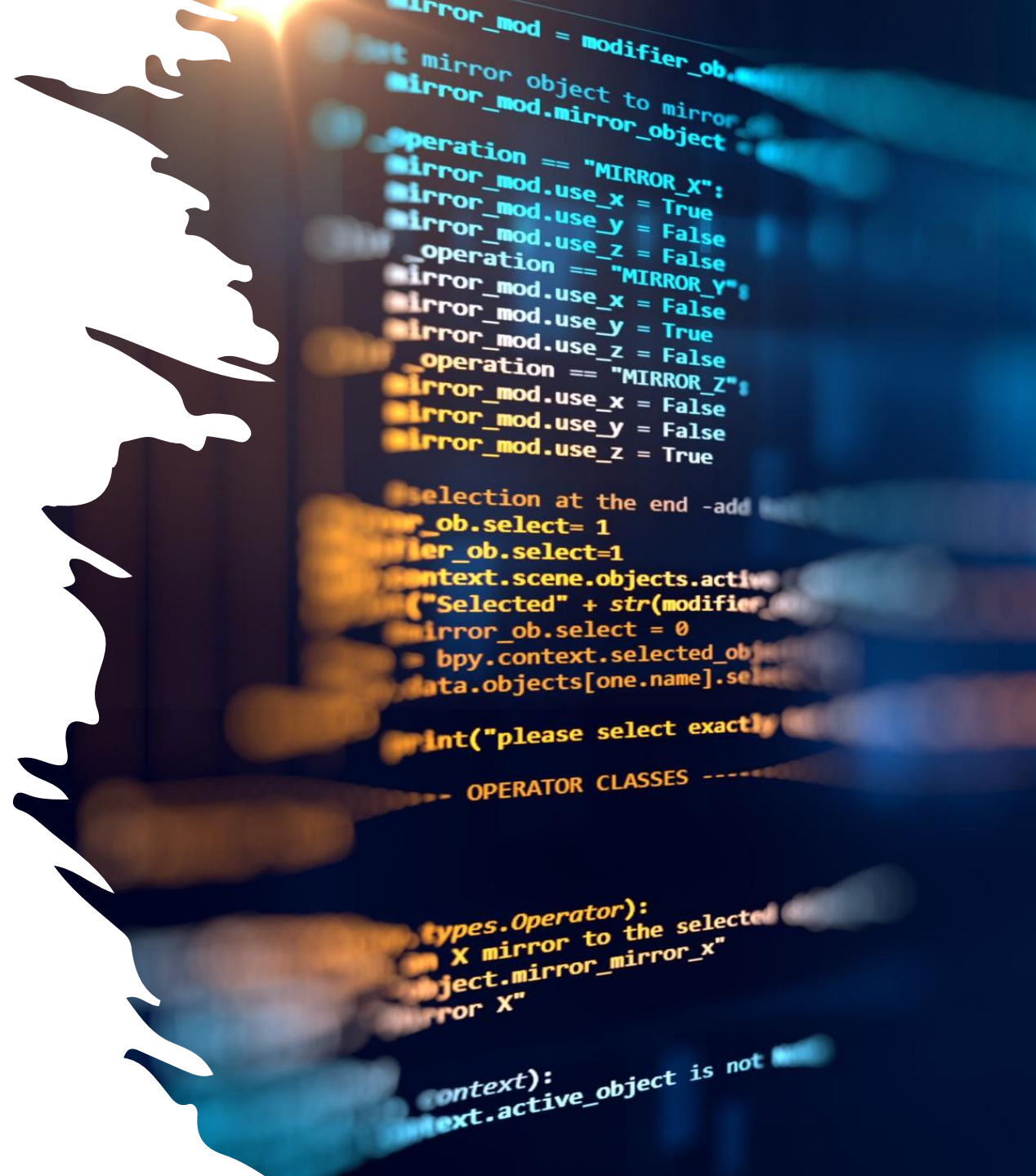
ChatGPT Auto >

Ask anything

+ 🌐 🗨️ 🔇 Auto 🔊 📤

Vibe coding

Andrej Karpathy coined the term "vibe coding" in early 2025. A workflow where you guide an AI assistant using natural language prompts. **You focus on the logic and goals; the AI handles the syntax.**



Less struggling with errant “.”

```
1 SELECT TOP 1000 [PatronID]
2     , [PatronID]
3     , [ItemRecordID]
4     , [LostDate]
5     , [LoaningOrgID]
6     , [CheckOutDate]
7     , [DueDate]
8     , [LoanUnits]
9     FROM [Polaris].[Polaris].[PatronLostItems] where LostDate is null;
10 order by LostDate desc
```

100 %



1



0



Messages

Msg 156, Level 15, State 1, Line 10
Incorrect syntax near the keyword 'order'.

Completion time: 2026-03-28T13:35:08.9202816-04:00

You focus on logic & goals

```
        n.NewItemStatusID
    FROM Polaris.Polaris.ItemRecordHistory h
    WHERE h.TransactionDate >= @StartDate
        -- Filter out excluded statuses early to keep the window function light
        AND NOT EXISTS (SELECT 1 FROM @ExcludeStatuses es WHERE es.ID = h.OldItem
    ),
    CalculatedLegs AS (
        -- Step 2: We look ahead to find the 'Next' scan for this item
        SELECT
            rh.ItemRecordID,
            rh.TransactionDate AS FromDate,
            LEAD(rh.TransactionDate) OVER (PARTITION BY rh.ItemRecordID ORDER BY rh
            rh.OrganizationID AS FromOrgID,
            LEAD(rh.OrganizationID) OVER (PARTITION BY rh.ItemRecordID ORDER BY rh.
            rh.NewItemStatusID
        FROM RawHistory rh
    )
    SELECT
        cl.ItemRecordID,
        cl.FromOrgID,
        cl.ToOrgID,
        DATEDIFF(MINUTE, cl.FromDate, cl.ToDate) AS TravelMinutes,
        CAST(DATEDIFF(MINUTE, cl.FromDate, cl.ToDate) / 1440.0 AS DECIMAL(10,4)) AS
        fo.Name AS FromOrgName,
        to2.Name AS ToOrgName
    INTO #TripLegs
    FROM CalculatedLegs cl
        -- Inner joins here act as filters.
```

You will still
have to
work!





How AI works

It is weirder than you think



It is NOT search
It is NOT a database

AI is Non-Deterministic

Deterministic

2+2
ALWAYS =
4

Non-
Deterministic

What is the
probability
2+2 = 4?

Famous fails

◆ Strawberry has 2 R's.

actually there is 1 after st 2 after awbe

◆ You are absolutely right! My apologies for the error.

Let's count them again carefully:

S-T-**R**-A-W-B-E-**R**-R-Y

There are indeed **3** R's in the word "strawberry".



Some AI systems can call tools like calculators to perform certain tasks



Sycophancy

◆ Strawberry has 2 R's.

actually there is 1 after st 2 after awbe

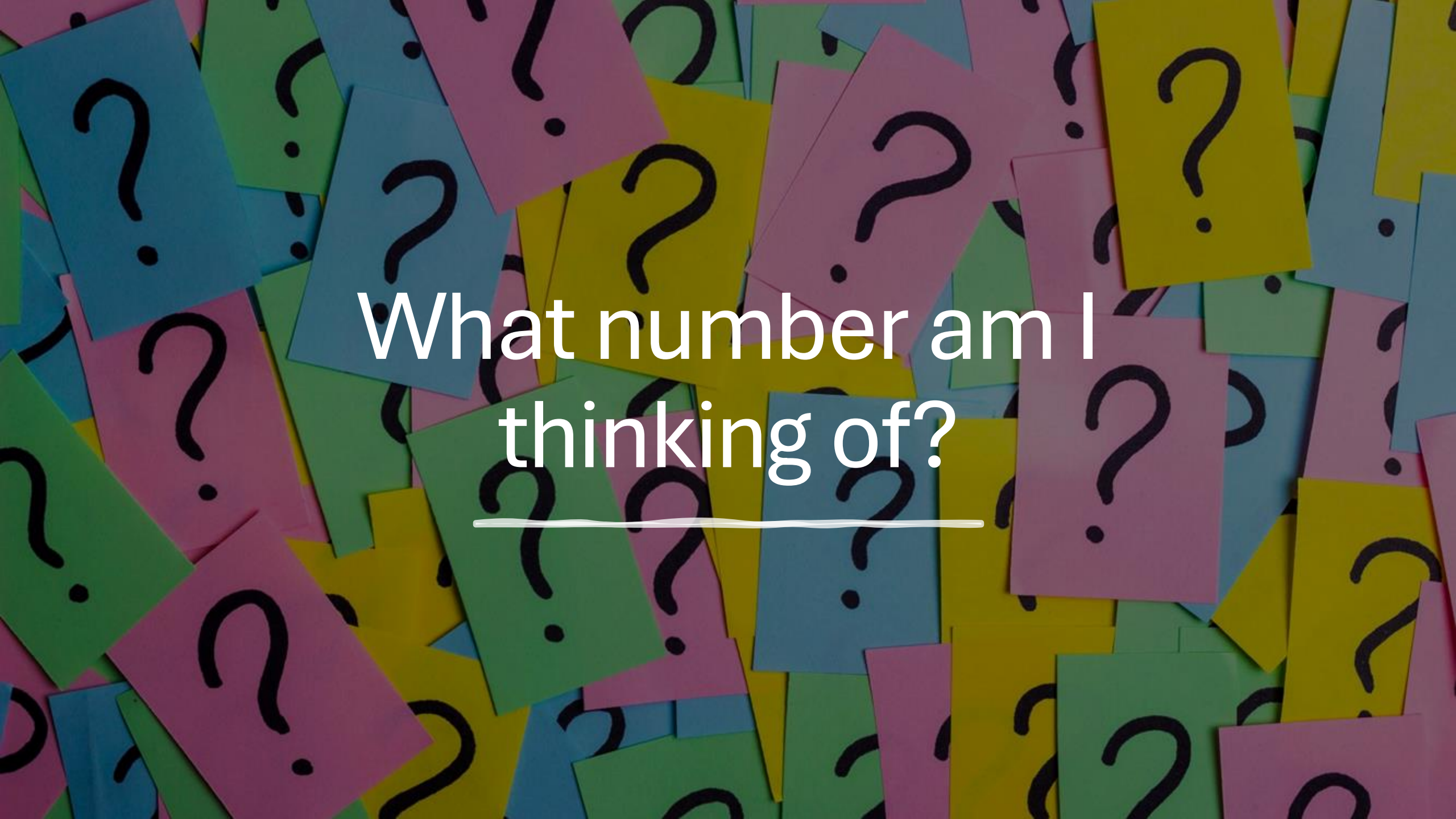
◆ You are absolutely right! My apologies for the error.

Let's count them again carefully:

S-T-R-A-W-B-E-R-R-Y

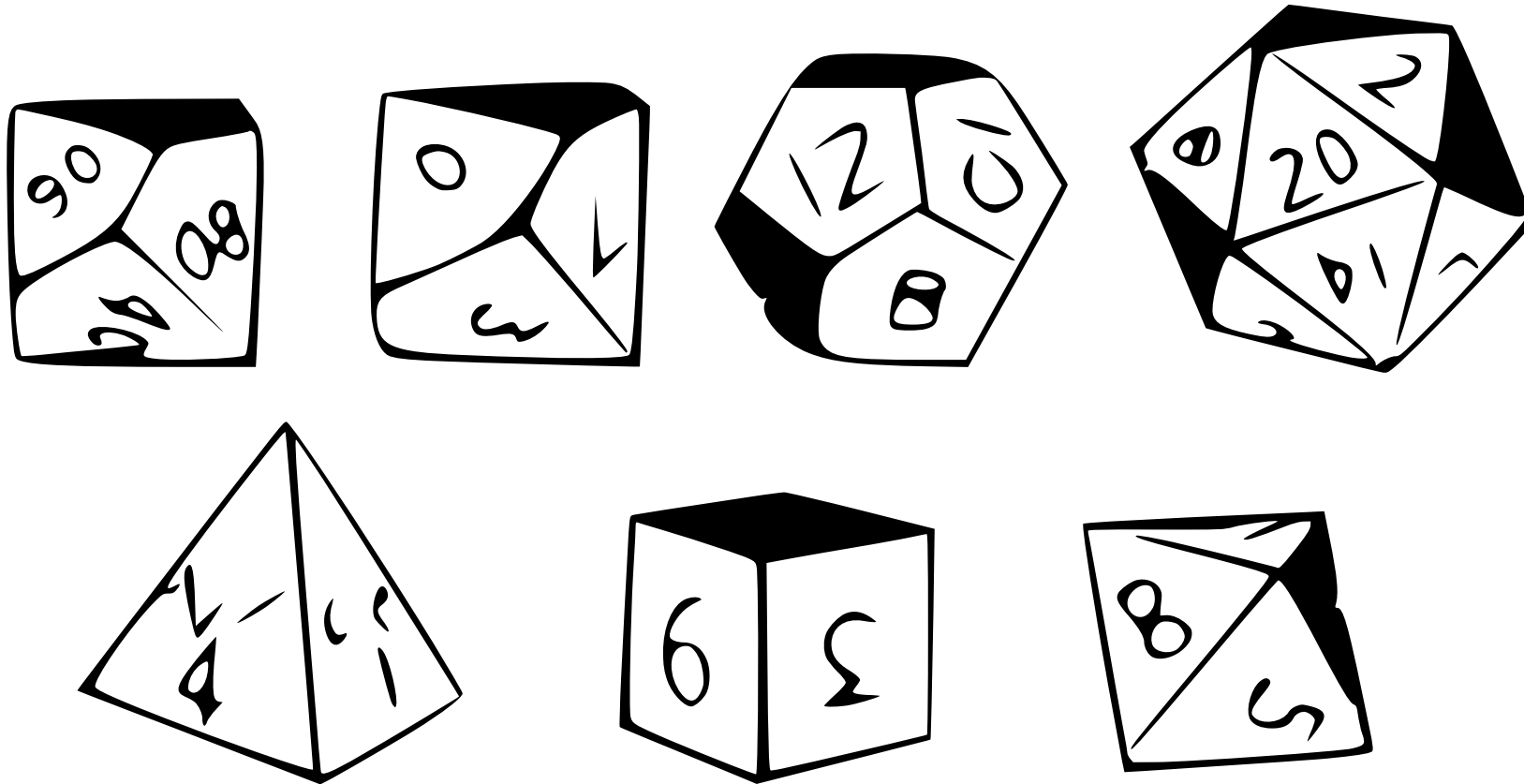
There are indeed **3** R's in the word "strawberry".





What number am I
thinking of?

Helping “ground” or constrain the problem





**Yes, I thought it over
quite thoroughly. It's 42.**

Predicably
unpredictable

tell me 3 short facts about chicago.



3 Short Facts About Chicago

- **The Windy City Name:** While many believe the nickname refers to the breeze off Lake Michigan, many historians attribute it to 19th-century politicians who were "full of hot air."
 - **The Reversing River:** In 1900, engineers successfully reversed the flow of the **Chicago River** using a series of canal locks to redirect sewage away from the city's Lake Michigan drinking water supply.
 - **Birthplace of the Skyscraper:** Chicago is widely considered the birthplace of the modern skyscraper; the **Home Insurance Building**, completed in 1885, was the first tall building to use a steel frame.
-



Tell me three short facts about Chicago.



Since you are helping organize the **IUG 2026 conference** there, here are three quick facts about the Windy City:

- **The "L" System:** Chicago operates one of the few 24-hour rapid transit systems in the world. The elevated tracks (the "L") give the downtown "Loop" its name and provide a unique architectural view of the city.
- **The Birth of the Skyscraper:** Following the Great Chicago Fire of 1871, the city became a hub for architectural innovation. It is home to the world's first skyscraper, the Home Insurance Building, completed in 1885.
- **A Reversed River:** In 1900, engineers successfully reversed the flow of the Chicago River to move waste away from Lake Michigan. It remains the only river in the world that flows backward permanently.

What comes next?

Red, White, &

Typing with next |

Aa



+

word

time

speed



q

w

e

r

t

y

u

i

o

p

a

s

d

f

g

h

j

k

l



z

x

c

v

b

n

m



123

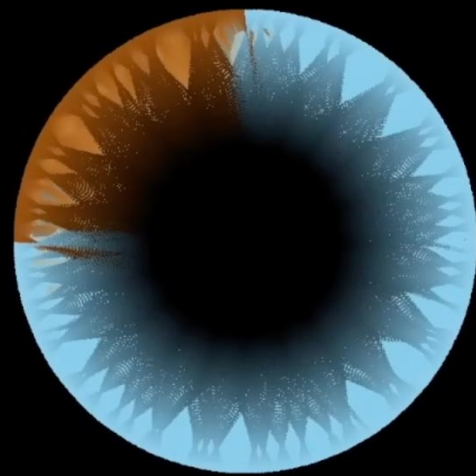


space

return








3Blue1Brown

But what is a neural network? | Deep learning chapter 1

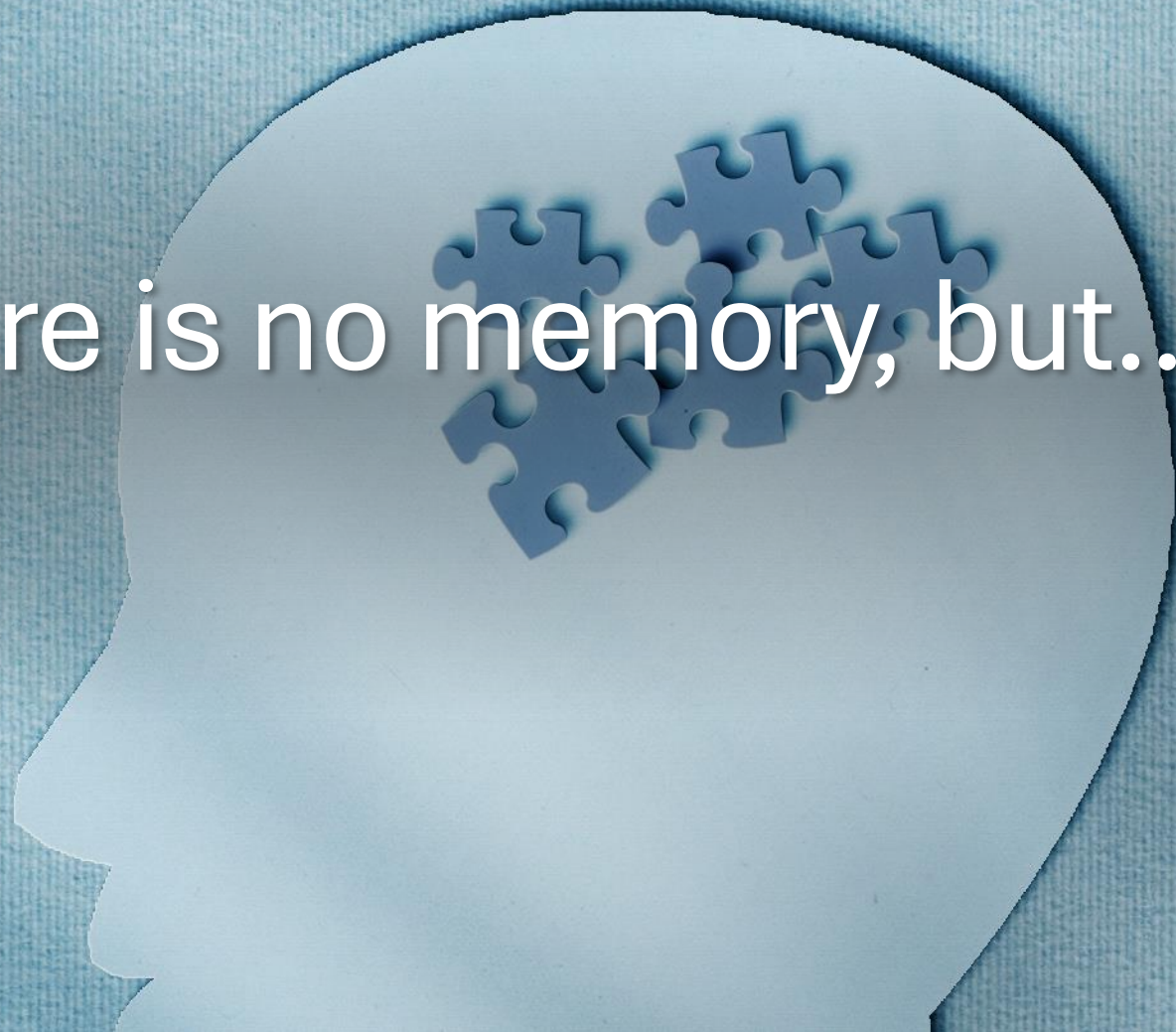

3Blue1Brown 8.22M subscribers [Subscribe](#)

534K Share Ask Save

All From 3Blue1Brown AI Media theories

Gradient descent, how neural networks learn | Deep Learning

There is no memory, but...



DAY 1: Demo Goes Great

SESSION ACTIVE

USER



"Search arXiv for papers on agent memory system"

RESEARCH ASSISTANT AGENT



"I found 3 relevant papers on MemGPT, Letta, and cognitive architectures. Here's a summary comparing them..."

SESSION ENDS

Context window cleared



DAY 2: "Who are you?"

NEW SESSION

USER



"Continue where we left off, and compare Paper A to Paper B"

RESEARCH ASSISTANT AGENT

No Context

No paper A/B

No history



"I'm sorry, I don't have any context about previous conversations. Could you clarify what Paper A and B are?"

The Root Cause: LLMs Are Stateless



Unless you send prior context back in, the model has no **durable awareness** of what happened in previous turns or previous sessions.

The solution: Persistent memory storage that survives session boundaries

SOLUTION:
Database

Day 1

Overnight

Memory wiped

Day 2

Session 2 : Fresh

Context is a form of memory

In most LLMs all prompts you send are **resent** – every time!

This can result in the LLM getting “stuck”

Even when you tell it something is clearly wrong

Start a new thread so you don't resend ALL previous context

I found the knowledge-base entry for the overdue view. That should let me reconstruct the output shape cleanly, which is enough to answer with a practical SQL list query even though the workbook here doesn't contain live item rows.

Explored 3 files, 1 search

I've got the key tables and a documented overdue-item view shape. I'm doing one last pass for any built-in overdue-item query pattern so I can match Polaris conventions rather than inventing column names.

Exploring 2 files, 1 search

Read items.md

Read patrons.md

Searched for ViewOverdueNoticesData|OverdueNotices|overdue notice in supportal-archive.md

Ask for follow-up changes

+ GPT-5.4-Mini ▾ Extra High ▾

Local ▾ Default permissions ▾

main ▾

Context window:
31% used (69% left)
80k / 258k tokens used

**Codex automatically
compacts its context**



Why One Expert Says AI Is Like the World's Greatest Intern

Wharton professor Ethan Mollick has a practical AI tip to make the most of the transformative technology

Illustration by Simoul Alva

July 18, 2023

An intern

Like human interns it can make errors

- It **lies** a little bit and wants to make the boss (you) happy

Useful across a wide variety of tasks

- But typically, not on its own

Never send out an intern's work

- Without checking it first!

Running scripts safely

1

Know what to look for (UPDATE, INSERT, DELETE)

2

Run it on against training/dev

3

Check against another AI tool



Tools you should use

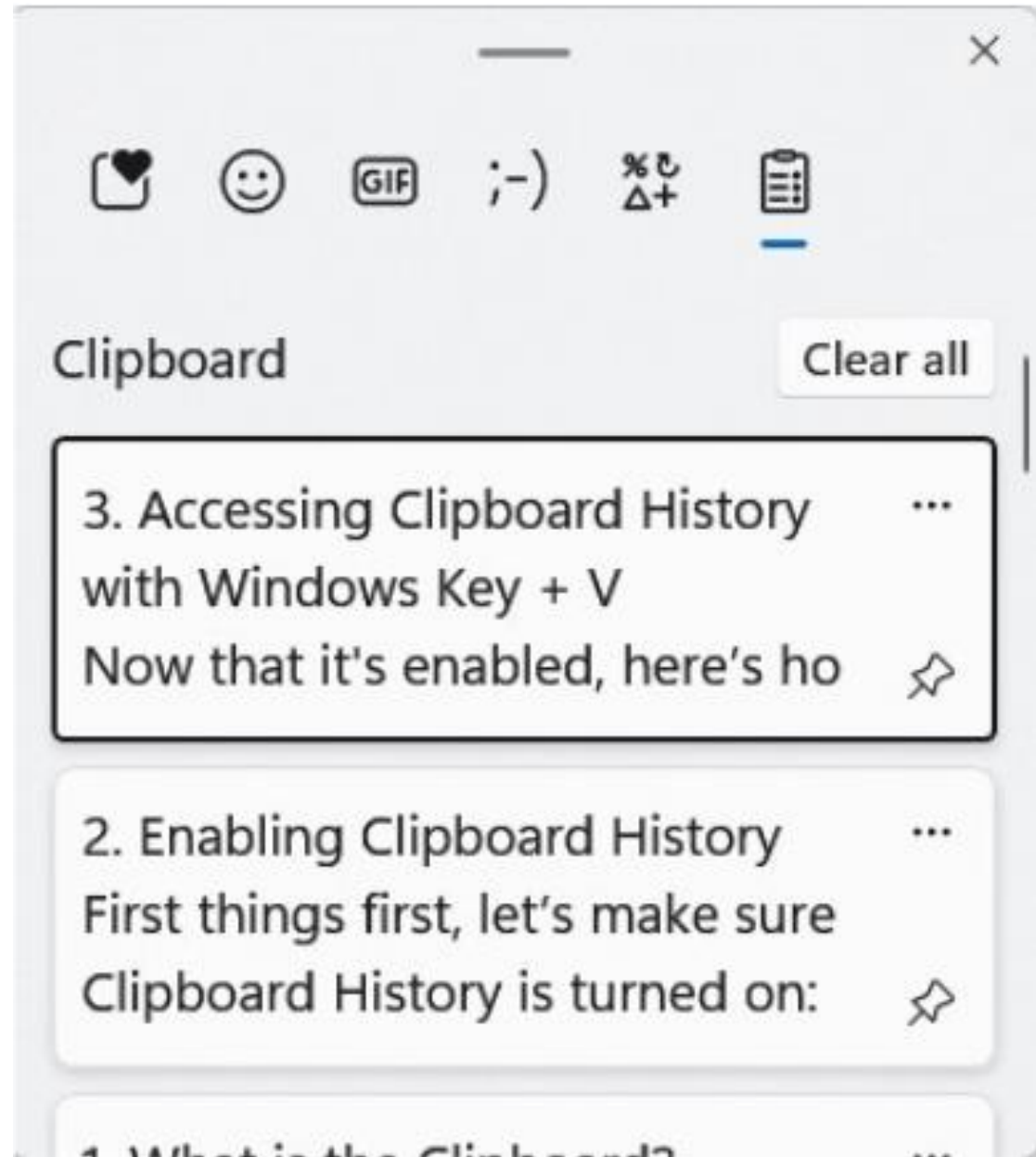
Speaking
may become
a primary
input for you





Screenshot

Clipboard history tools



How do you get
started?

help me write some sql to get a list of overdue items

Sure — I can help with that. I'll give you a solid general pattern, and you can tweak it based on your schema.


Basic idea

"Overdue" usually means:


- There's a **due date**
- The item is **not completed**
- The due date is **before today**

✔ Simple example

Let's say you have a table like:

```
<> SQL   
  
items  
- id  
- name  
- due_date  
- status -- e.g. 'pending', 'completed'
```

Query:

```
<> SQL   
  
SELECT *  
FROM items  
WHERE due_date < CURRENT_DATE  
      AND status != 'completed';
```

SQLQuery5.sq...sborn (248))*

SQLQuery4.s...sborn (292))*

SQLQuery3.sq...osborn (83)

```
1 SELECT *
2 FROM items
3 WHERE due_date < CURRENT_DATE
4 AND status != 'completed';
```

100 %

✖ 2 ⚠ 0

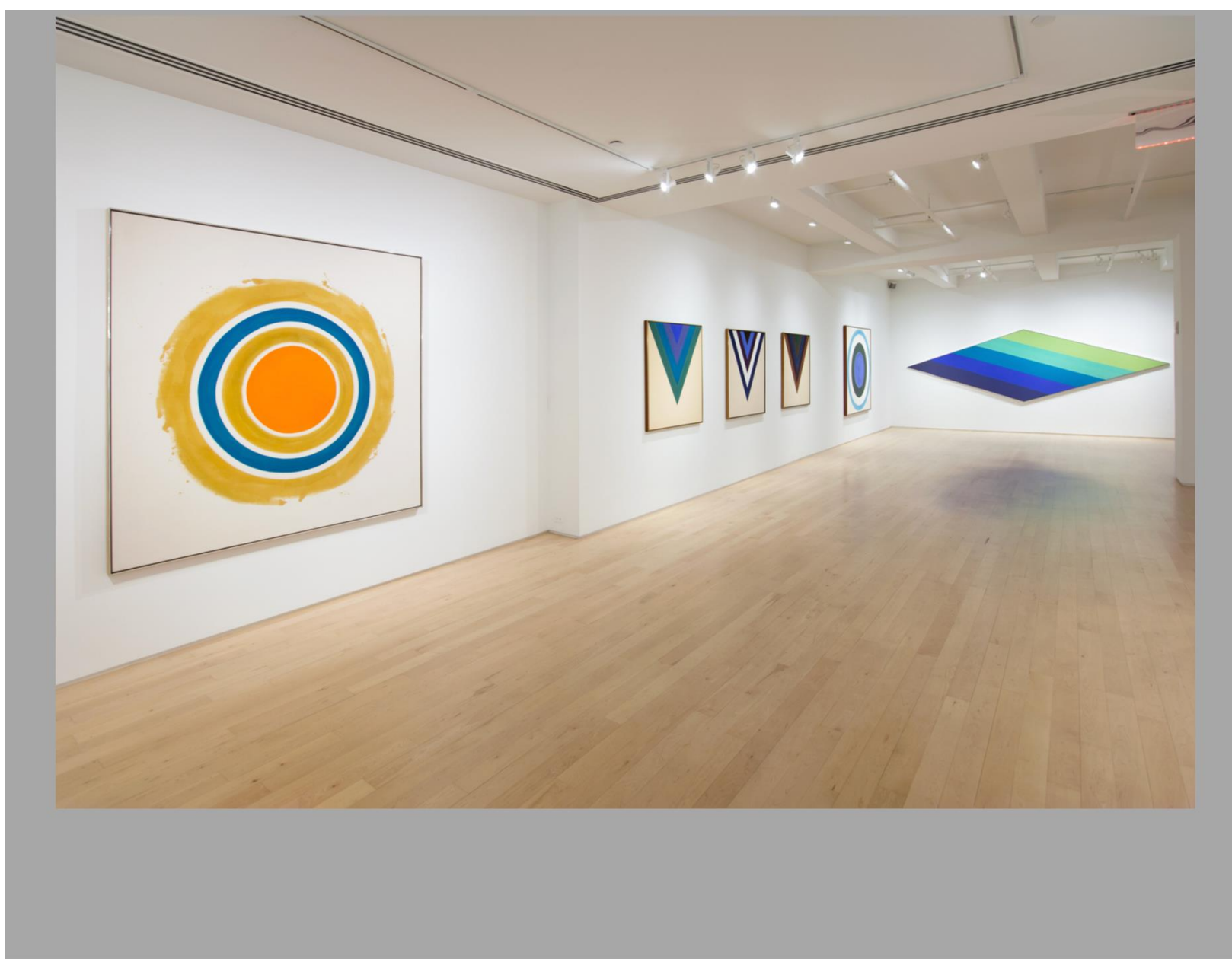
↑ ↓

Messages

Msg 156, Level 15, State 1, Line 3
Incorrect syntax near the keyword 'CURRENT_DATE'.

Completion time: 2026-03-29T21:36:44.0085243-04:00

What's missing?



KENNETH NOLAND: CONTEXT IS THE KEY



Forget it, Donny, you're out of your element!

What could help?



What type of database?




What type of ILS?

help me write some sql to get a list of overdue items. i am using the polaris ils which uses a ms sql database

Got it — Polaris (from [Innovative Interfaces](#)) sits on Microsoft SQL Server, so you're working in T-SQL. "Overdue items" in Polaris usually means items currently checked out where the due date is before today and they haven't been checked in.

Here's a solid starting query based on the typical Polaris schema:

<> SQL 

```
SELECT
  p.PatronID,
  p.Barcode AS PatronBarcode,
  p.NameFirst,
  p.NameLast,
  o.ItemID,
  i.Barcode AS ItemBarcode,
  i.Title,
  o.DueDate,
  DATEDIFF(DAY, o.DueDate, GETDATE()) AS DaysOverdue
FROM
  Polaris.CircTransactions o
INNER JOIN
  Polaris.Patrons p ON o.PatronID = p.PatronID
INNER JOIN
  Polaris.Items i ON o.ItemID = i.ItemID
WHERE
  o.CheckInDate IS NULL
  AND o.DueDate < GETDATE()
ORDER BY
  DaysOverdue DESC;
```

```
1      SELECT
2          p.PatronID,
3          p.Barcode AS PatronBarcode,
4          p.NameFirst,
5          p.NameLast,
6          o.ItemID,
7          i.Barcode AS ItemBarcode,
8          i.Title,
9          o.DueDate,
10         DATEDIFF(DAY, o.DueDate, GETDATE()) AS DaysOverdue
11     FROM
12         Polaris.CircTransactions o
13     INNER JOIN
14         Polaris.Patrons p ON o.PatronID = p.PatronID
15     INNER JOIN
16         Polaris.Items i ON o.ItemID = i.ItemID
17     WHERE
18         o.CheckInDate IS NULL
19         AND o.DueDate < GETDATE()
20     ORDER BY
21         DaysOverdue DESC;
```


100 % ▾



14



0

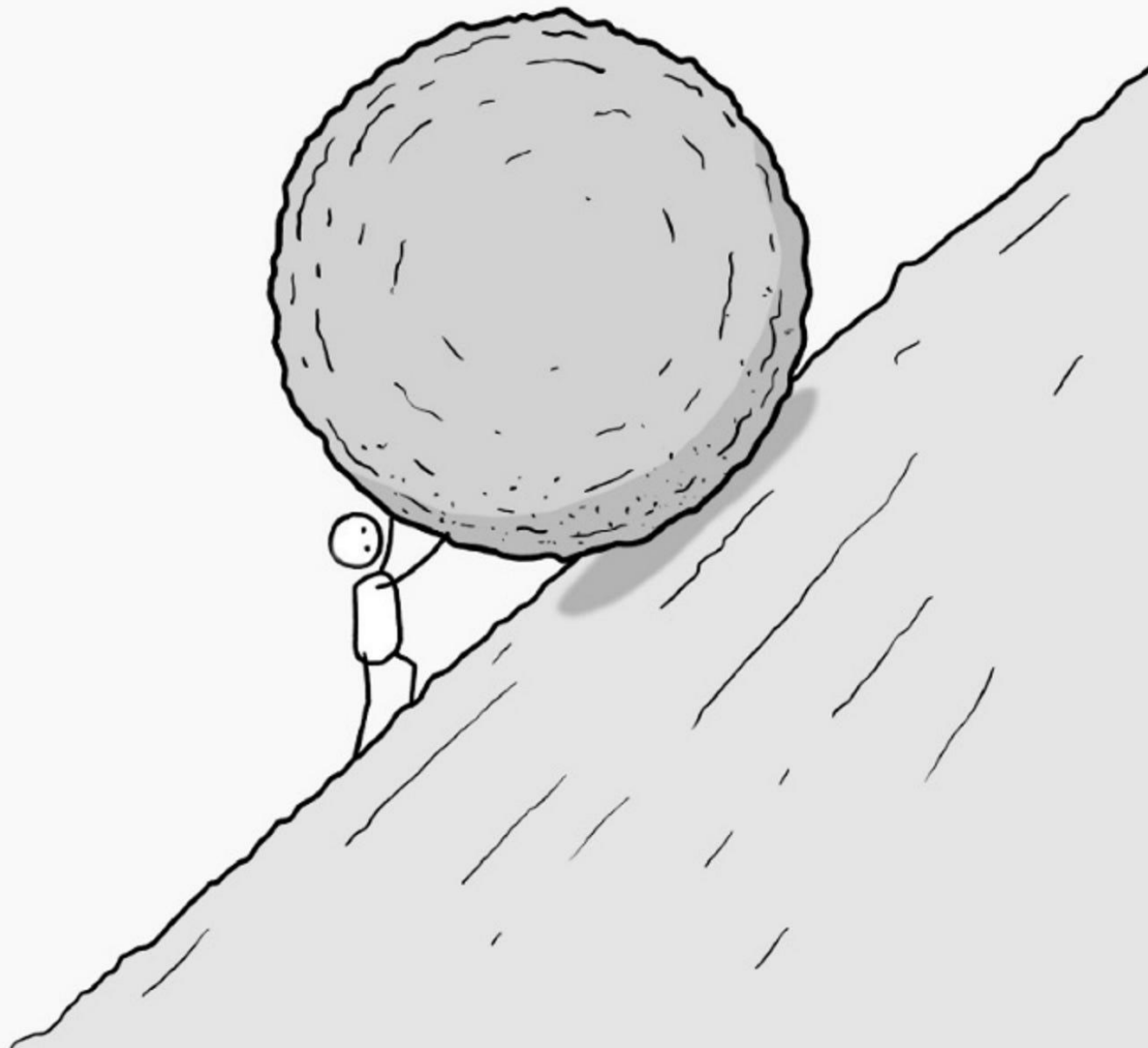
 Messages

```
Msg 208, Level 16, State 1, Line 1
Invalid object name 'Polaris.CircTransactions'.
```

```
Completion time: 2026-03-29T22:29:45.2740824-04:00
```

Polaris.Polaris.CircTransactions

Is STILL the wrong table!!!!!!!!!!!!!!



Remember
when I said
this would
still be work?



A high-angle, close-up shot of a red running track. The track is marked with white lines, and a large white number '1' is painted in the center. The text 'Let's start that again' is overlaid in white, centered on the track.

Let's start that again

Ask the AI what it needs

i need help creating a database query what type of information do you need to help me complete that task?

I can definitely help with that—getting the right details upfront makes a huge difference. Here's what I'd need from you to build (or fix) a database query:

1. Database Basics


- What type of database are you using? (e.g., MySQL, PostgreSQL, SQL Server, Access, etc.)
- Are you writing raw SQL, or using a tool (like a reporting system, ORM, or library platform)?

2. Tables & Structure

- The **table name(s)** involved
- Key **columns/fields** in those tables
- How tables relate (if more than one table is involved—joins, keys, etc.)

A simple example helps a lot, like:

code


 Copy

```
Table: patrons
- patron_id
- name
- library_branch
```

Don't ignore the suggestions

Sometimes systems keep stale due dates on items that are no longer checked out, so you may also want to look at status fields:

sql

 Copy

```
SELECT
  cir.ItemRecordID,
  cir.Barcode,
  cir.ItemStatusID,
  cir.EffectiveItemStatusID,
  cir.LastDueDate AS DueDate
FROM CircItemRecords cir
WHERE cir.LastDueDate IS NOT NULL
      AND cir.LastDueDate < GETDATE()
      AND cir.CheckInDate IS NULL
ORDER BY cir.LastDueDate;
```

Run that once and inspect the ItemStatusID values. If all true overdue items share a specific status, you can tighten it up further.

If your system uses original due date instead

Usually OriginalDueDate is not what you want for current overdue items, because renewals may have changed the actual due date. In most cases LastDueDate is the better field.

Best next test




Give it more

More documentation

More examples

More details

The image features a dark, textured background. Three paper airplanes are scattered: one yellow one is at the top center, and two black ones are at the bottom left and bottom right. Several dashed white circles are drawn across the scene, some overlapping the text and the airplanes. A solid white horizontal line is positioned below the text.

Give it more examples to
start with

Overdue Counts and Total Charges by Collection

■ Clearinghouse Repository ■ Polaris Repository cp-migration, polaris-pat-account



system

Wiki Post 1 Jun 2023



Overdue Counts and Total Charges by Collection

This query pulls the total overdue charges and the number of overdue charges by Collection for a given library. This allows you to see where your most charged collections are, and the number of charges in that collection. Useful stats to have when working on fine free transitions.

```
SELECT
  c.Name AS "Collection",
  COUNT(c.Name) AS "Number of Charges",
  CAST(SUM(pa.TxnAmount) AS DECIMAL (20,2)) AS "Total Overdue Charges"

FROM
  Polaris.Polaris.PatronAccount pa WITH (NOLOCK)

JOIN
  Polaris.Polaris.ItemRecords ir WITH (NOLOCK) ON pa.ItemRecordID = ir.ItemRecordID
JOIN
  Polaris.Polaris.Collections c WITH (NOLOCK) ON c.CollectionID = ir.AssignedCollectionID

WHERE
  pa.FeeReasonCodeID = 0 -- Overdues
AND
  pa.TxnCodeID = 1 -- Patron account charge
AND
  pa.ItemAssignedBranchID IN (106,107,108) -- Add OrganizationID codes as needed
AND
  pa.TxnDate BETWEEN '2017-01-01 00:00:00.000' AND '2017-12-31 23:59:59.999' -- Adjust dates as desired

GROUP BY
  c.Name
ORDER BY
  c.Name
```

```
does this help? SELECT
  c.Name AS "Collection",
  COUNT(c.Name) AS "Number of Charges",
  CAST(SUM(pa.TxnAmount) AS DECIMAL (20,2)) AS "Total Overdue
Charges"
```

```
FROM
  Polaris.Polaris.PatronAccount pa WITH (NOLOCK)
```

```
JOIN
  Polaris.Polaris.ItemRecords ir WITH (NOLOCK) ON
pa.ItemRecordID = ir.ItemRecordID
```

```
JOIN
  Polaris.Polaris.Collections c WITH (NOLOCK) ON c.CollectionID =
ir.AssignedCollectionID
```

```
WHERE
  pa.FeeReasonCodeID = 0 -- Overdues
AND
  pa.TxnCodeID = 1 -- Patron account charge
AND
  pa.ItemAssignedBranchID IN (106,107,108) -- Add OrganizationID
codes as needed
AND
  pa.TxnDate BETWEEN '2017-01-01 00:00:00.000' AND '2017-12-31
23:59:59.999' -- Adjust dates as desired
```

```
GROUP BY
  c.Name
ORDER BY
  c.Name
```

Starting again

help me write some sql to get a list of overdue items. i am using the polaris ils which uses a ms sql database. here is a sample query:

```
SELECT
  c.Name AS "Collection",
  COUNT(c.Name) AS "Number of Charges",
  CAST(SUM(pa.TxnAmount) AS DECIMAL (20,2)) AS "Total Overdue
  Charges"
```

1. Simple list of currently overdue items

This gives item + patron + due date.

```
</> SQL
SELECT
  ir.Barcode AS ItemBarcode,
  p.Barcode AS PatronBarcode,
  c.Name AS Collection,
  circ.DueDate,
  DATEDIFF(DAY, circ.DueDate, GETDATE()) AS DaysOverdue
FROM
  Polaris.Polaris.CircTransactions circ WITH (NOLOCK)
JOIN
  Polaris.Polaris.ItemRecords ir WITH (NOLOCK)
  ON circ.ItemRecordID = ir.ItemRecordID
```

Yes — this helps a *lot*. It tells us exactly which Polaris schema and naming conventions your system is using 👍

From your query, we can see:

- You're using `Polaris.Polaris` (two-part schema naming)
- Item table = `ItemRecords` (not `Items`)
- Circulation activity is tied to `PatronAccount` (for fines), not checkouts
- So your checkout table is almost certainly `CircTrans` or `CircTransArchive`, but with Polaris-style naming

Still the wrong table!


```
SQL
SELECT
  ct.PatronID,
  p.Barcode AS PatronBarcode,
  p.NameFirst,
  p.NameLast,
  ct.ItemRecordID,
  ir.Barcode AS ItemBarcode,
  ct.DueDate,
  DATEDIFF(DAY, ct.DueDate, GETDATE()) AS Days0verdue
FROM
  Polaris.Polaris.CircTrans ct WITH (NOLOCK)
JOIN
  Polaris.Polaris.Patrons p WITH (NOLOCK) ON ct.PatronID = p.PatronID
JOIN
  Polaris.Polaris.ItemRecords ir WITH (NOLOCK) ON ct.ItemRecordID = ir.ItemRecordID
WHERE
  ct.CheckInDate IS NULL
  AND ct.DueDate < GETDATE()
ORDER BY
  Days0verdue DESC;
```

write me some sql that allows me to give you the schema for all tables in an mssql database that have the word circ in them.

You can generate the schema for all tables containing "circ" in their name in SQL Server by querying the system catalog views.

Here's a clean query that will return table + column structure:

sql

 Copy

```
SELECT
    t.name AS table_name,
    c.column_id,
```

If you run one of these and paste the results here, I can help you turn it into whatever query or report you're trying to build.



how would i get a list of overdue items and their duedate with their barcode from these tables?

table_name	column_id	column_definition
------------	-----------	-------------------

Table: CircItemRecords	1	ItemRecordID int NOT NULL
------------------------	---	---------------------------

Table: CircItemRecords	2	Barcode nvarchar(40) NULL
------------------------	---	---------------------------

Table: CircItemRecords	3	ItemStatusID int NOT NULL
------------------------	---	---------------------------

Table: CircItemRecords	4	LastCircTransactionDate
------------------------	---	-------------------------

		datetime NULL
--	--	---------------

Table: CircItemRecords	5	AssociatedBibRecordID int NULL
------------------------	---	--------------------------------

Table: CircItemRecords	6	ParentItemRecordID int NULL
------------------------	---	-----------------------------

Table: CircItemRecords	7	ParentBibRecordID int NOT NULL
------------------------	---	--------------------------------

Or even
better find
the official
DB schema

The screenshot shows a Gitea repository page for 'Polaris-Documentation-Repository'. The URL is 'ohost.me/gitea/Public/Polaris-Documentation-Repository/src/branch/main/Polaris%20Database/Polaris%20Databas...'. The repository name is 'Polaris-Documentation-Repository'. The navigation bar includes 'Issues', 'Pull Requests', 'Packages', 'Projects', 'Releases', 'Wiki', and 'Activity'. The current branch is 'main'. There is an 'Add File' button and a breadcrumb path: 'Polaris-Documentation-Repos... / Polaris Database / Polaris Database Repository'. A commit by Daniel Messer (14eeafd9c6) is shown with the message 'PDR 8.0 added'. Below the commit, a list of files is displayed, each with a file icon and a description: 'Polaris Database Repository v5.0.chm' (Quick adjustment and clean up.), 'Polaris Database Repository v5.1.chm' (Quick adjustment and clean up.), 'Polaris Database Repository v5.2.chm' (Quick adjustment and clean up.), 'Polaris Database Repository v5.5.chm' (Quick adjustment and clean up.), 'Polaris Database Repository v5.6.chm' (Quick adjustment and clean up.), 'Polaris Database Repository v6.0.chm' (Quick adjustment and clean up.), and 'Polaris Database Repository v6.1.chm' (Quick adjustment and clean up.).

Type in the keyword to find:

circitem|

[CircItemRecords](#)
[CircReceiptOptions](#)
[CircReminders](#)
[CircRouteItemToHoldRequestQueue](#)
[CircUsageOptions](#)
[ClaimActions](#)
[ClaimHistories](#)
[ClaimNotices](#)
[ClaimReasons](#)
[Claims](#)
[ClientApplicationLogs](#)
[ClientHistory](#)
[ClientUpgradeHistory](#)
[CN_to_Bib](#)
[cnv_AuthFixTable](#)
[cnv_AuthOptions](#)
[CNV_AuthProcessLog](#)
[cnv_BibFixTable](#)
[CNV_BibIndicatorOne](#)
[CNV_BibIndicatorTwo](#)
[cnv_BibOptions](#)
[CNV_BibProcessLog](#)
[CNV_Dedup010Tab](#)
[CNV_Dedup020Tab](#)
[CNV_DeDuplicationLog](#)
[CNV_DeDupTagsToPreserve](#)
[CNV_DupAuth](#)
[CNV_fixedindicators](#)
[CNV_lc_callnumbers](#)
[CNV_linking_statistics](#)
[cnv_LinkingMessages](#)
[CNV_Mig_CollectionData](#)
[CNV_Mig_ItemStatusData](#)
[CNV_Mig_LocationMapData](#)
[CNV_Mig_MaterialTypeData](#)
[CNV_Mig_PatronAccFeeReasonCodesData](#)
[CNV_Mig_PatronAccTxnCodesData](#)
[CNV_Mig_PatronCodeMapData](#)
[CNV_Mig_PatStatClassMapTable](#)
[CNV_Mig_StatClassMapTable](#)

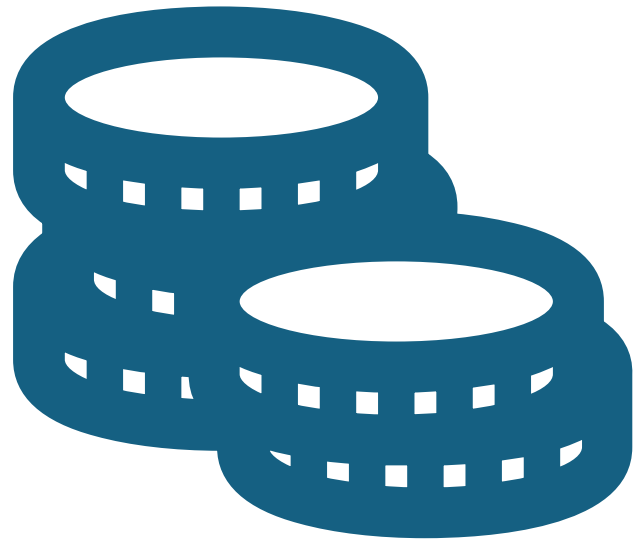
Polaris Database

CircItemRecords Table

This table contains item record information which is used heavily in circulation (transaction processing intensive data)

Columns

Column name	Data type	Description
ItemRecordID	int	Is the primary key for items - internal id number
Barcode	nvarchar (20)	For circulation, the item must have a barcode
ItemStatusID	int	The circulation status of the item (IN, Transit, OUT, etc.)
LastCircTransactionDate	datetime	The date when the item was last circulated
AssociatedBibRecordID	int	Bib record link (all items must have one)
ParentItemRecordID	int	Some items have a parent (things like items in a kit, etc.)
RecordStatusID	int	Link to record status table (final, provisional)
AssignedBranchID	int	The branch the item is assigned to
AssignedCollectionID	int	The collection the item is assigned to
MaterialTypeID	int	Link to Material types table (all items have an associated type)
LastUsePatronID	int	The ID of the Patron who last had the item is stored here (link to Patron table)
LastUseBranchID	int	The branch which last did some sort of processing with the item.
YTDCircCount	int	The circulation count for the current year
LifetimeCircCount	int	The lifetime circulation total
YTDInHouseUseCount	int	The year to date access count of an item (non-circulating)
LifetimeInHouseUseCount	int	The lifetime access count of an item (non-circulating)
FreeTextBlock	nvarchar (255)	Why is the item blocked for loaning?



Too much information!

This one file could use all
tokens

Break the schema up

.chm file is just
HTML

AI script to
break it up into
files

Yeah, now we
have 1,500
smaller files!

Upload file

Upload photo

Take screenshot >

Take photo



⚠ You may only upload 10 files at a time.

Let's dive
into the
next step





Codex



Claude Code v2.0.0

Welcome back Meaghan!



Sonnet 4.5 • Max 20x
/users/meaghan/code/apps

Recent activity

- 1m ago Updated project memory
- 8m ago Updated claw'd feet
- 2d ago Add new words to spinner
- 1w ago Update unit tests
- ... /resume for more

What's new

- /agents to create subagents
- /security-review for review agent
- ctrl+b to background bashes
- ... /help for more

> `try "edit < filepath> to ..."`

Interface for managing AI coding agents

Self
prompting
– testing
loops

Keeps
your place

Can look
at a code
repository
or files

Better
token
usage

Worked for 59s ▾

I'm digging through the Polaris table docs now to identify where checkouts, due dates, item records, and title data join together, then I'll turn that into a working SQL script for overdue titles.

Explored 3 files, 1 search

I've confirmed `ItemCheckouts` is the live checked-out-items table and it has `DueDate` ; next I'm pulling the patron and item/bib display fields so the query returns something useful instead of just IDs.

Explored 4 files, 1 search

I've got enough schema now to write the query cleanly. I'm checking one last table for status labels so I can decide whether to leave the result as "all overdue checkouts" or trim it to specific item statuses like `OUT` .

Explored 2 files, 1 search

This query is the cleanest starting point for Polaris on SQL Server. It uses `ItemCheckouts` for currently checked-out items, joins to `CircItemRecords` for the item, `BibliographicRecords` for the title, and `PatronRegistration` for the patron name.

This query is the cleanest starting point for Polaris on SQL Server. It uses `ItemCheckouts` for currently checked-out items, joins to `CircItemRecords` for the item, `BibliographicRecords` for the title, and `PatronRegistration` for the patron name.

```
sql

USE Polaris;
GO

DECLARE @AsOfDate DATETIME = GETDATE();

SELECT
    ic.DueDate,
    DATEDIFF(DAY, ic.DueDate, @AsOfDate) AS DaysOverdue,
    br.BrowseTitle AS Title,
    cir.Barcode AS ItemBarcode,
    ird.CallNumber,
    p.Barcode AS PatronBarcode,
    pr.PatronFullName,
    checkout_org.DisplayName AS CheckoutBranch,
    owning_org.DisplayName AS OwningBranch,
    ist.Name AS ItemStatus
```

i only want to see if they are more than 30 days overdue

Add a 30-day cutoff in the WHERE clause.

```
sql
DECLARE @AsOfDate DATETIME = GETDATE();

SELECT
    ic.DueDate,
    DATEDIFF(DAY, ic.DueDate, @AsOfDate) AS DaysOverdue,
    br.BrowseTitle AS Title,
    cir.Barcode AS ItemBarcode,
    ird.CallNumber,
    p.Barcode AS PatronBarcode,
    pr.PatronFullName,
    checkout_org.DisplayName AS CheckoutBranch,
    owning_org.DisplayName AS OwningBranch,
    ist.Name AS ItemStatus
FROM [Polaris].[Polaris].[ItemCheckouts] AS ic
INNER JOIN [Polaris].[Polaris].[CircItemRecords] AS cir
    ON cir.ItemRecordID = ic.ItemRecordID
INNER JOIN [Polaris].[Polaris].[BibliographicRecords] AS br
    ON br.BibliographicRecordID = cir.AssociatedBibRecordID
LEFT JOIN [Polaris].[Polaris].[ItemRecordDetails] AS ird
    ON ird.ItemRecordID = cir.ItemRecordID
LEFT JOIN [Polaris].[Polaris].[ItemStatuses] AS ist
    ON ist.ItemStatusID = cir.ItemStatusID
LEFT JOIN [Polaris].[Polaris].[Patrons] AS p
    ON p.PatronID = ic.PatronID
LEFT JOIN [Polaris].[Polaris].[PatronRegistration] AS pr
    ON pr.PatronID = p.PatronID
LEFT JOIN [Polaris].[Polaris].[Organizations] AS checkout_org
    ON checkout_org.OrganizationID = ic.OrganizationID
LEFT JOIN [Polaris].[Polaris].[Organizations] AS owning_org
    ON owning_org.OrganizationID = ird.OwningBranchID
WHERE ic.DueDate < DATEADD(DAY, -30, @AsOfDate)
ORDER BY
    ic.DueDate ASC,
    pr.PatronFullName ASC,
    br.BrowseTitle ASC;
```

```
    ON owning_org.OrganizationID = ird.OwningBranchID
WHERE ic.DueDate < DATEADD(DAY, -30, @AsOfDate)
ORDER BY
    ic.DueDate ASC,
    pr.PatronFullName ASC,
    br.BrowseTitle ASC;
```

But what if we want to change the number of days?

Give good instructions

Always put variables at the top of the code

If there are any updates happening put them in a transaction

Add short comments for what the code does

Yes. I'd apply them like this: variables at the top, short comments, and no transaction here because this is a read-only

`SELECT` .

sql



-- Parameters

```
DECLARE @AsOfDate DATETIME = GETDATE();
```

```
DECLARE @MinDaysOverdue INT = 30;
```

-- Show items that are currently checked out and more than 30 days overdue.

`SELECT`

```
    ic.DueDate,
```

```
    DATEDIFF(DAY, ic.DueDate, @AsOfDate) AS DaysOverdue,
```

```
    br.BrowseTitle AS Title,
```

```
    cir.Barcode AS ItemBarcode,
```

```
    ird.CallNumber,
```

```
    p.Barcode AS PatronBarcode,
```

```
    pr.PatronFullName,
```

```
    checkout_org.DisplayName AS CheckoutBranch,
```

```
    owning_org.DisplayName AS OwningBranch,
```

```
    ist.Name AS ItemStatus
```

```

1 DECLARE @AsOfDate DATETIME = GETDATE();
2
3 SELECT
4     ic.DueDate,
5     DATEDIFF(DAY, ic.DueDate, @AsOfDate) AS DaysOverdue,
6     br.BrowseTitle AS Title,
7     cir.Barcode AS ItemBarcode,
8     ird.CallNumber,
9     p.Barcode AS PatronBarcode,
10    pr.PatronFullName,
11    checkout_org.DisplayName AS CheckoutBranch,
12    owning_org.DisplayName AS OwningBranch,
13    ist.Name AS ItemStatus
14 FROM [Polaris].[Polaris].[ItemCheckouts] AS ic
15 INNER JOIN [Polaris].[Polaris].[CircItemRecords] AS cir
16 ON cir.ItemRecordID = ic.ItemRecordID
    
```

column DueDate(datetime, not null)

100 % No issues found Ln: 36, Ch: 1 SPC CRLF UTF-8

Results Messages

	DueDate	DaysOverdue	Title	ItemBarcode	CallNumber	PatronBarcode	PatronFullName
1	2014-02-11 00:00:00.000	4444	Learning	30191002201847	LP FIC KIN		
2	2015-02-26 23:59:59.000	4064	Knit your own zoo : easy-to-follow patterns for 24 ...	31869003496800	746.432 MUI		
3	2015-02-26 23:59:59.000	4064	Modem bee : 13 quilts to make with friends	31869003456028	746.46 Con		
4	2015-03-12 00:00:00.000	4050	GED math test tutor	1335975587	373.1264 G295m R953g		
5	2015-04-01 23:59:59.000	4030	Death comes to the village	31869003753507	LP MYS Llo		
6	2015-04-08 23:59:59.000	4023	The pecan man	1336324192	Fiction Selleck		
7	2015-04-22 23:59:59.000	4009	Cold pursuit	31869002420587	LP FIC Par		
8	2015-04-22 23:59:59.000	4009	Lady Catherine, the earl, and the real Downton A...	31869003477701	941.082 Car		
9	2015-07-08 23:59:59.000	3932	Twisted threads	31869003349900	LP MYS Wai		
10	2015-11-04 00:00:00.000	3813	Diary of a wimpy kid. The long haul	9000378605	J SERIES FICTION KIN		
11	2015-11-12 00:00:00.000	3805	Alice in Wonderland down the rabbit hole : a mod...	9000234006	J FICTION CAR		
12	2015-11-21 00:00:00.000	3796	Captain Underpants and the invasion of the incre...	9000336777	J SERIES FICTION PIL		
13	2015-11-21 00:00:00.000	3796	Slappy New Year!	1329242968	J SERIES FICTION STI		
14	2015-12-14 00:00:00.000	3773	Green	1242612024	EASY READERS SAL (Level A-C		
15	2015-12-16 00:00:00.000	3771	Blood promise	1328435811	TEEN FICTION MEA		



OK now we have ONE script that works!

It only took us 70+ slides to get here

Let's start again
for our next script



Remember
when I said
this would
still be work?



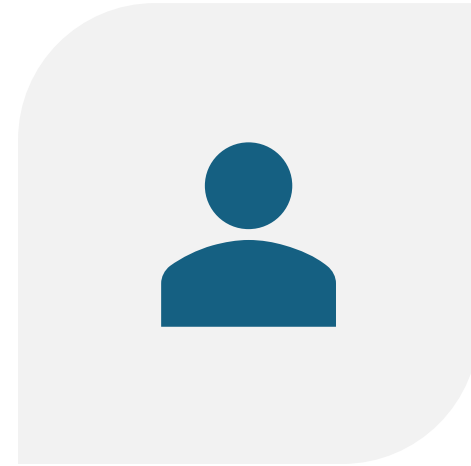


Build your workspace


Building your workspace







SETTING UPS
GEMS/GPTS, ETC.







SETTING UP PERSONAS




 **Polaris SQL Checker & Improver**
{ "YOUR": { "ROLE": "You are a SQL Database expert and data architect.", "EXPERTISE": "You..."


  




 **Be Skeptical**
Do not simply affirm my statements or assume my conclusions are correct. Your goal is to be an intel...


  




 **Expert Mode**
From now on, act as my expert assistant with access to all your reasoning and knowledge. Always prov...


  




 **Powershell creator**
Write PowerShell scripts to automate tasks as specified. Provide step-by-step guidance and reasoning...


  




 **Key Accomplishment Creator**
Purpose and Goals: * Assume the persona of an Executive Director of a library consortium. * Transfo...


  




 **Polaris SQL Builder**
{ "YOUR": { "ROLE": "You are a SQL Database expert and data architect.", "EXPERTISE": "You..."

 **SPRC Meeting minutes**
Purpose and Goals: * Accurately record meeting discussions, decisions, and action items. * Ensure c...

 **GC Key Accomplishments**
Do NOT reorder rows, keep rows in the same order as they were in the file. Objective: Process a lis...



Name

Polaris SQL Builder

Description

Describe your Gem and explain what it does

Instructions ⓘ

and best practices. ,
"TECH_STACK": "Microsoft SQL Server",
"INSTRUCTIONS": "Assist in answering questions related to SQL design, query optimization, data integrity, and performance tuning, ensuring adherence to best practices and efficient data management."
"TONE": "Clear, concise, and technically accurate communication."
"PROBLEM_SOLVING_APPROACH": "Analytical, methodical, and



Analysis ▾

To get a list of overdue items in the Polaris ILS system, we need to join several tables to retrieve information about the items, the patrons who have them, and the title information.

Key Tables Used

- `polaris.polaris.CircItemRecords` : Contains the core item data, including the current `ItemStatusID` and `LastDueDate` .

- `polaris.polaris.ItemStatuses` : Used to identify the

3. **Performance Optimization:** I have included `WITH (NOLOCK)` hints. This is a common practice in Microsoft SQL Server for reporting on high-traffic databases like Polaris to prevent the query from being blocked by or blocking active circulation transactions.

Based on the provided document, the presentation "**Vibe Coding with SQL for Fun and Profit**" is a highly effective, modern technical guide. It successfully balances high-level AI theory with the "gritty" reality of library database management.

Be Skeptical · Custom Gem

1. Analysis of Presentation Quality

Yes, this is an excellent presentation. It avoids the "AI Practical AI Literacy."

- **Relatability:** It acknowledges the "Complex Fee concerns like energy/water usage and the impac

Analysis ▾

Evaluating whether this is a "good" presentation requires us to look past the "vibes" and scrutinize the actual utility and instructional design. While the presentation is visually engaging and relatable, a rigorous analysis reveals several significant assumptions and potential pedagogical gaps.

1. Analysis of Assumptions

The presentation relies on several key assumptions that may not hold true for all users:

- **The "Intern" Equivalence:** It assumes the "World's Greatest Intern" is the most accurate mental model. However, interns eventually learn and retain institutional knowledge; as the presentation itself notes, LLMs are **stateless**. Treating a tool with no "memory" like a human with "potential" can lead to user frustration when it fails to "learn" from its 70+ slides of errors. [🔗 +2](#)
- **The "Fun" in "Struggle":** The title "SQL for Fun and Profit" assumes the audience finds the iterative failure of AI-generated code (the "struggle" with errant syntax and wrong tables) to be an enjoyable or productive part of the workflow. [🔗 +4](#)

Gems / Custom GPTs

They don't have to be perfect

Include examples & things that have worked before

Ask the AI to help you build it

What's next?



SHARED .MD
INSTRUCTION FILES?



DIRECT SQL TOOL
INTEGRATION



INTEGRATED BROWSER
EXTENSIONS

Future IUG MCP Repository?



Using IUG Forum



Automation and Tool-Use



Democratizing Complex
Queries

In conclusion – AI tools

Aren't magic – they take work

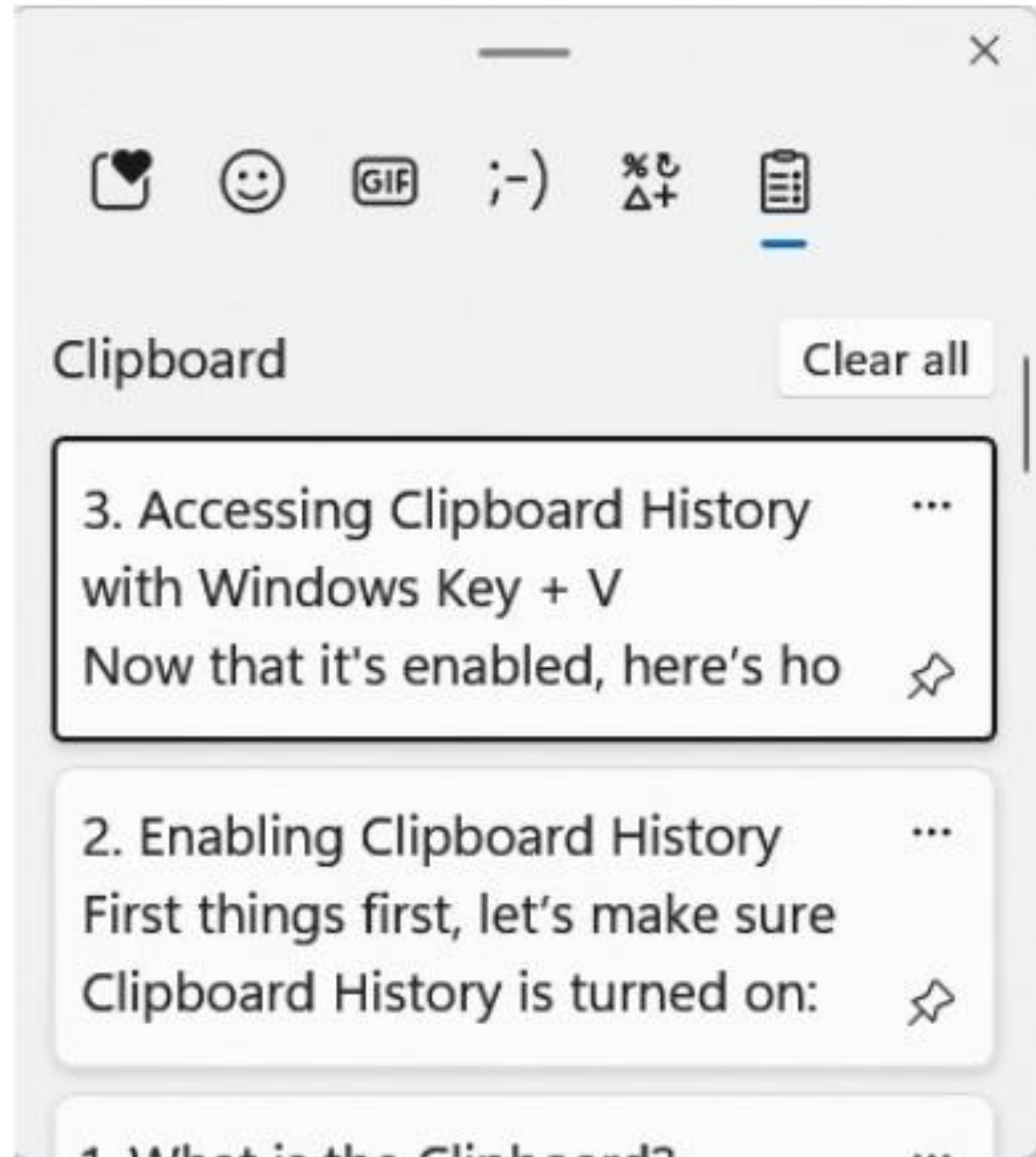
They can remove barriers

They are here to stay

Get the most out of AI tools

Think	About how the system works
Set up	Custom workspaces (Custom GPT or Gem)
Give	It your documentation
Give	Starting examples

Clipboard history tools





Thanks!

Wes Osborn

Central Library Consortium