HANDS-ON LAB GUIDE

Snowflake CoCo Asset Care analytics

Nine short labs, from your first chat to your own skill. You use CoCo in Snowsight for each step, and you check every answer.

Lab 1
Set up and meet CoCo

Lab 2 · Level 1
Ask and explore

Lab 3 · Level 2
Write and explain SQL

Lab 4 · Level 3
Fix broken queries

Lab 5 · Level 4
Notebook with charts

Lab 6 · Level 5
Plan and build

Lab 7 · Level 5
AI functions

Lab 8 · Level 6
AGENTS.md and a skill

Lab 9
Control and cost

Snowflake trial account, Enterprise edition, credit card added · Real BHP asset and site names, dummy maintenance data

How this guide works

Each lab is a few step cards. A card has up to four parts:

Why · the reason for the step

Read it first. It links the step to the architecture.

1. Numbered actions

Where to go and what to select.

TYPE IN COCO

```
exact text to type
```

Copy it as it is, then push Enter.

✓ **Check** · what must be true

CoCo's words can differ. The facts and numbers must match.

Three things you do all day

Open CoCo

1. Select the CoCo icon, bottom right of Snowsight.
2. For a clean start, select **New chat** at the top of the panel.

Approve CoCo's actions

1. CoCo asks before it runs SQL or changes a file.
2. In **Manual** mode, approve each action when CoCo asks.
3. For a file change, read the diff, then select **Keep** (or **Undo**).

Run a step of a SQL file

1. Workspaces, open the file.
2. Highlight the step, from its `-- Step` line to its last `;` (or to `$$;` for a block).
3. Push Ctrl + Enter (Mac Cmd + Return).

Check the numbers

1. Open `04_check_answers.sql`.
2. Run the check with the same number as the lab.
3. Compare with CoCo's answer.

Lab files

Tip · All SQL is in the lab files. You never copy SQL from this PDF.

File	Used in
`00_preflight_check.sql`	Before the training
`01_setup.sql`, `02_data.sql`	Lab 1
`03_broken_queries.sql`	Lab 4
`04_check_answers.sql`	Labs 3 to 8, to check numbers
`AGENTS.md`, folder `weekly-asset-report`	Lab 8
`05_admin_cost.sql`	Lab 9
`99_cleanup.sql`	After the training

The use case

A mining maintenance team works across three BHP assets: **WAIO** (iron ore), **Escondida** (copper) and **BMA** (steelmaking coal). It has 60 maintenance requests: equipment, failure mode, priority (P1 is the highest) and downtime hours. Nobody in the team is a SQL expert. They want to know:

- Which asset and which equipment lose the most hours?
- Which failures repeat?
- How do we turn this into a view, a semantic view and a weekly report anyone can run?

You do this work with CoCo, and you learn to check and control what CoCo does.

Architecture

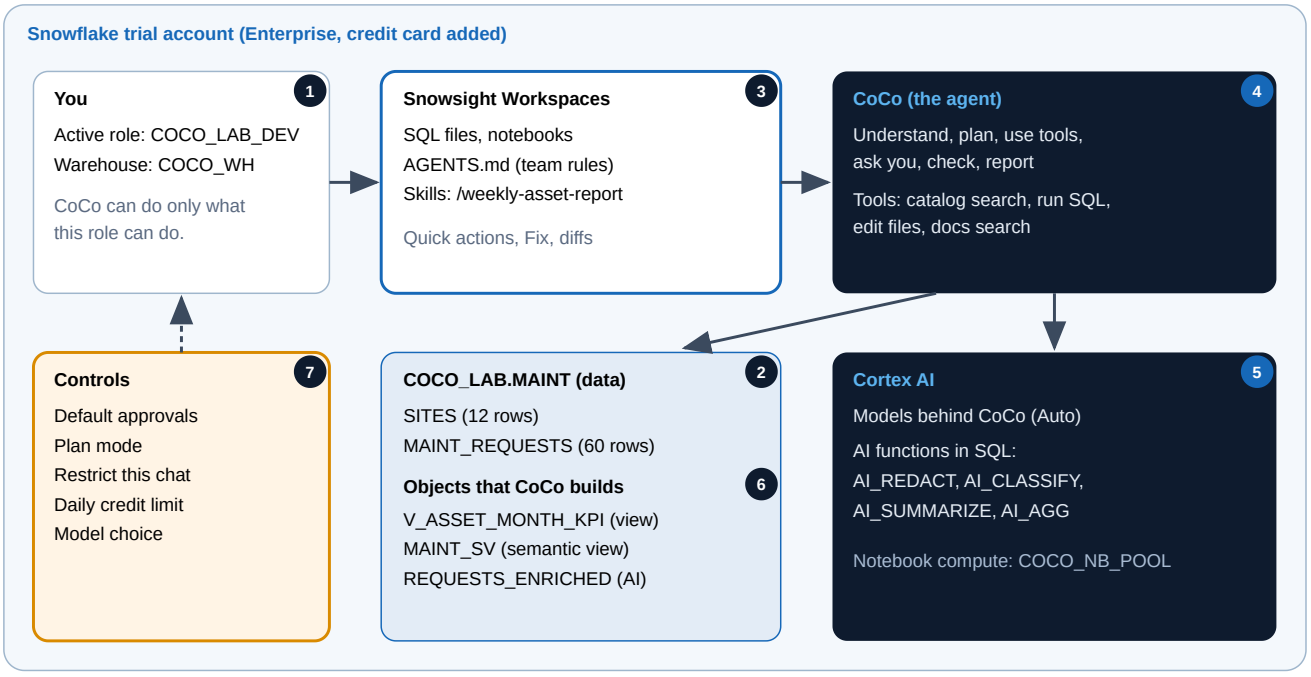


Figure 1. CoCo works as your active role and uses its tools on your data.

#	Part	What it does
1	You, your role, the warehouse	CoCo works as role COCO_LAB_DEV and runs SQL on COCO_WH.
2	Data	SITES and MAINT_REQUESTS in COCO_LAB.MAINT.
3	Workspaces	Your SQL files, notebooks, AGENTS.md and skills.
4	CoCo	Understands, plans, uses tools, asks you, checks, reports.
5	Cortex AI	The models behind CoCo and the AI functions it writes.
6	What CoCo builds	A KPI view, a semantic view, an AI-enriched table.
7	Controls	Approvals, plan mode, read-only chats, credit limit, model.

One CoCo request, step by step

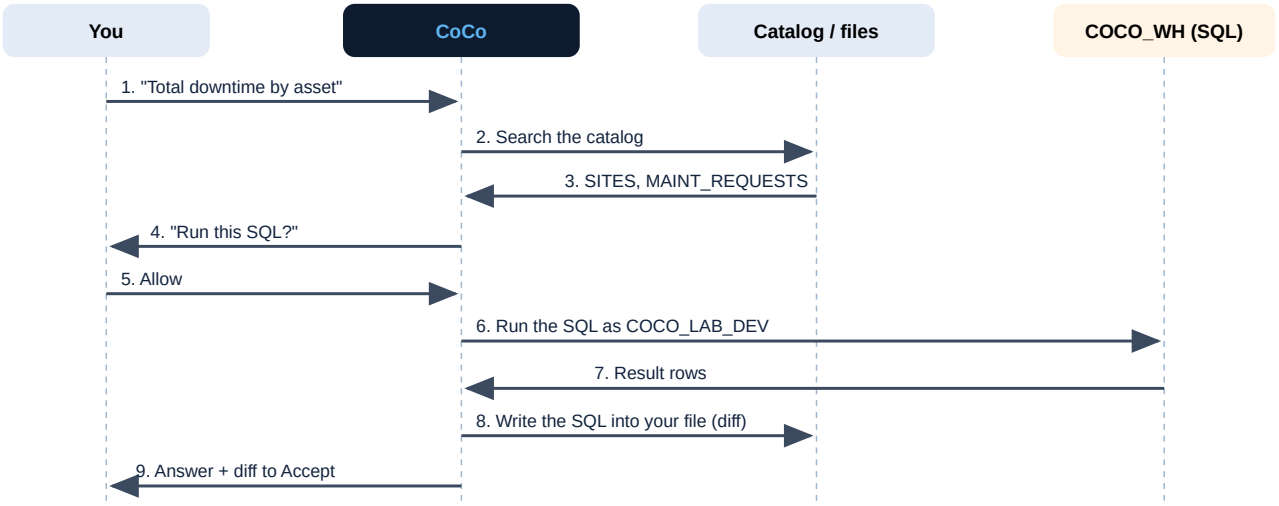


Figure 2. With default approvals, nothing runs before you allow it.

Before the training: checklist

Each person has an own trial account. Do C1 to C7 before Lab 1.

C1 Sign up for a trial

1. Go to **signup.snowflake.com**. Use your own email.
2. Edition **Enterprise**. Cloud **AWS**, region **US West (Oregon)**.
3. Accept the terms. Select **Get started**. Activate the account from the email.

Caution · CoCo in Snowsight is not available in Azure East US (Virginia).

C2 Use ACCOUNTADMIN and add a credit card

Why · AI features, CoCo included, are off on a trial until a card is added. The trial continues; it does not become a paid account.

1. User icon (bottom left) › **Switch Role** › **ACCOUNTADMIN**.
2. Admin › Billing › tab **Snowflake billing**.
3. In **Payment method**, select **+ Credit Card**. Type the details. Select **Add Card**.

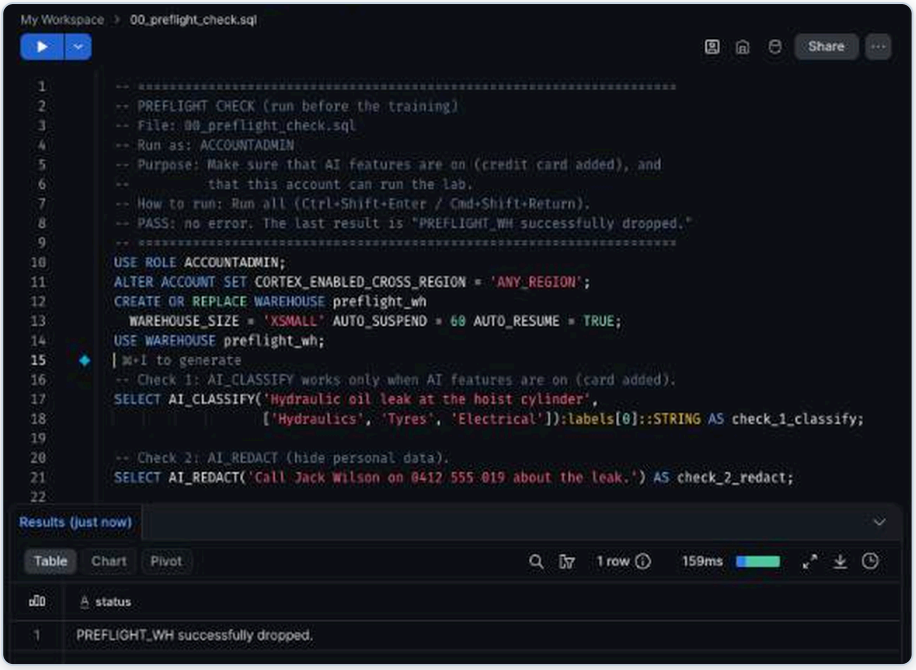
C3 Upload the SQL files

1. Extract `CoCo_Lab_Files.zip`.
2. Workspaces › **+ Add new** › **Upload files**.
3. Select the 7 SQL files. Do not upload AGENTS.md yet (Lab 8).

C4 Run the preflight check

1. Open `00_preflight_check.sql`. Click in the editor.
2. Push Ctrl + Shift + Enter (Mac Cmd + Shift + Return) to run all.

WHAT YOU SEE



The preflight file after Run all.

Caution · *Not available for trial accounts* means no card (C2). A value of 0 in Check 3 means CoCo is blocked: ask the trainer.

✓ **Check** · Run all shows only the last result: *PREFLIGHT_WH successfully dropped.* No error means all checks passed. To see one check, highlight it and run it with Ctrl + Enter (Mac Cmd + Return): Check 1 **Hydraulics**, Check 2 [NAME] and [PHONE_NUMBER], Check 3 -1 or a positive number.

C5 Say hello to CoCo

1. Open CoCo (icon, bottom right, or the **CoCo** switch at the top left).

TYPE IN COCO

```
Hello. Which role am I using?
```

✓ **Check** · CoCo answers with ACCOUNTADMIN.

Tip · **Trainer:** do all labs one time in a new trial account the day before, then run `99_cleanup.sql` .

1 Set up and meet CoCo

Load the data, switch to the lab role, and have your first chat.

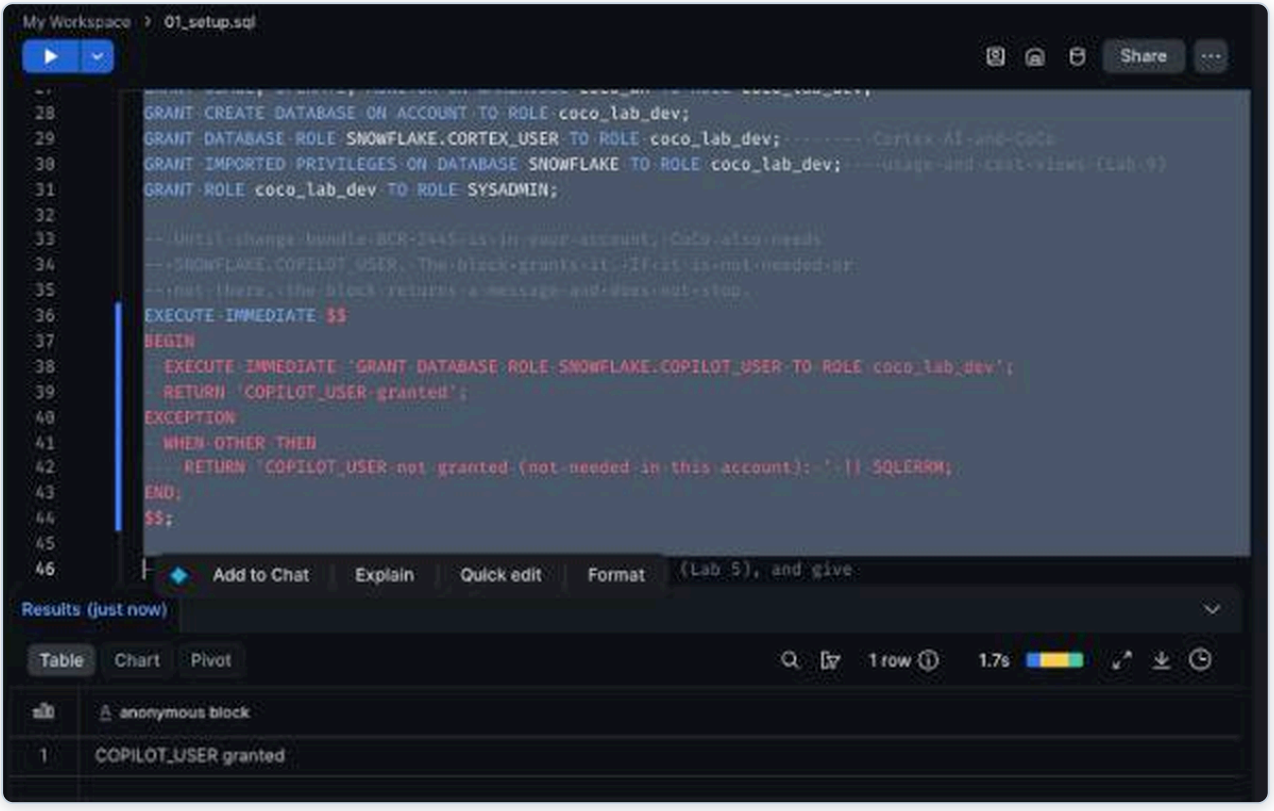
Setup | 20 min | Parts 1, 2, 4

1.1 Run the setup file

Why · CoCo can do only what your active role can do. One lab role, COCO_LAB_DEV, keeps every CoCo action easy to understand.

1. Open `01_setup.sql`.
2. Run **Step 1.1** and **Step 1.2**.
3. Run **Step 1.3** (to the end of its block `$$;`).
4. Run the two blocks of **Step 1.4**, one after the other.
5. Run **Step 1.5**.

WHAT YOU SEE



Step 1.3 run: the result shows COPILOT_USER granted.

✓ **Check** · Step 1.3: *COPILOT_USER granted* or *not needed* (both OK). Step 1.4: *Compute pool COCO_NB_POOL is ready* and *COCO_LAB_DEV is granted to* you. Step 1.5: schemas INFORMATION_SCHEMA, MAINT, PUBLIC.

1.2 Load the data

1. Open `02_data.sql`. Click in the editor.
2. Push Ctrl + Shift + Enter (Mac Cmd + Shift + Return).

WHAT YOU SEE



02_data.sql after Run all.

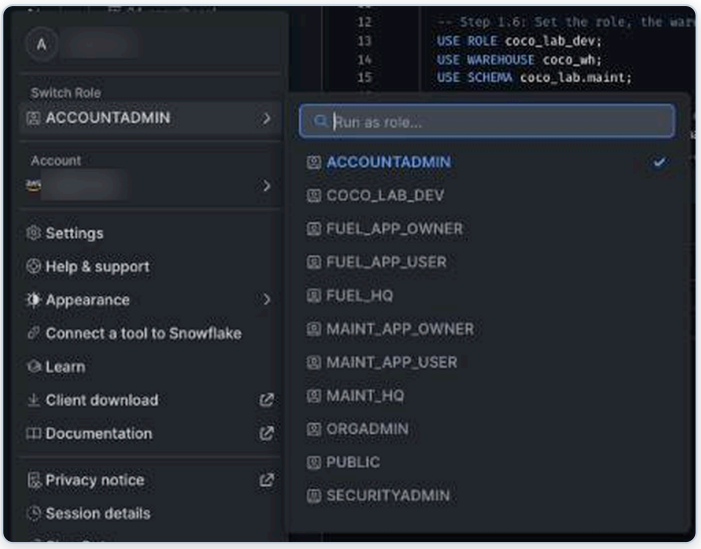
✓ **Check** · Only the last result shows: **sites 12**, **maint_requests 60**.

1.3 Switch your Snowsight role

Why · CoCo uses the role that is active in Snowsight. From now on, that is COCO_LAB_DEV.

1. Select your user icon (bottom left) › **Switch Role** › **COCO_LAB_DEV**.
2. Not in the list? Type it in **Run as role...**, or refresh the page (F5).

WHAT YOU SEE



User menu › Switch Role › the role list.

1.4 Find the CoCo controls

Why · Manual means CoCo asks you before it acts. Keep it on all day.

1. Open CoCo and start a **New chat**.
2. Model selector (lower right of the message box): keep **Auto (Balanced)**.
3. Approval mode (below the message box): **Manual**.
4. Type `@`: the object list opens. Type `/`: the skills list opens. Push Esc and clear the box.

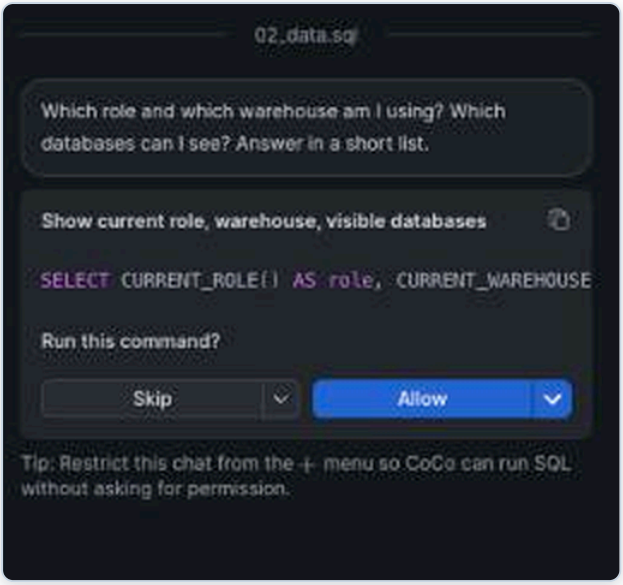
1.5 Your first chat

1. CoCo asks **Run this command?** Select **Allow** (or **Skip** to refuse).

TYPE IN COCO

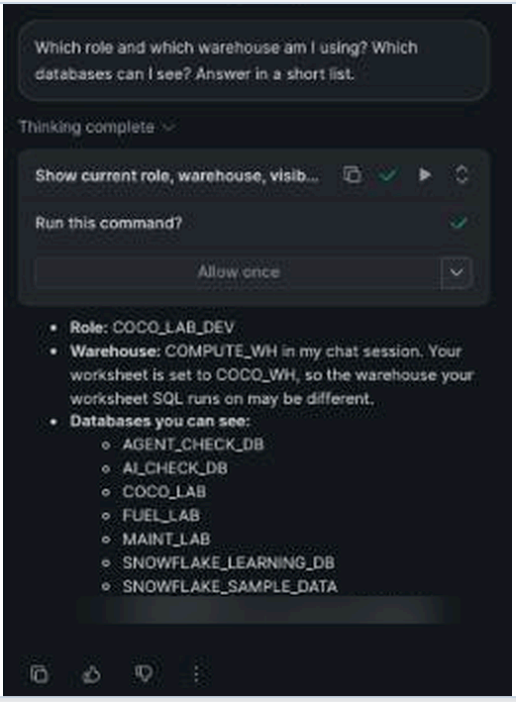
```
Which role and which warehouse am I using? Which databases can I see? Answer in a short list.
```

WHAT YOU SEE



CoCo asks before it runs SQL.

WHAT YOU SEE



The answer after Allow.

✓ **Check** · Role **COCO_LAB_DEV** and **COCO_LAB** in the database list. The warehouse can be **COMPUTE_WH**: the CoCo chat has its own session. Your SQL files use COCO_WH.

2 Ask and explore

Learn the data by asking. Point at a table with @. Ask the documentation.

Level 1 · 15 min · Parts 2, 4

2.1 What tables are there?

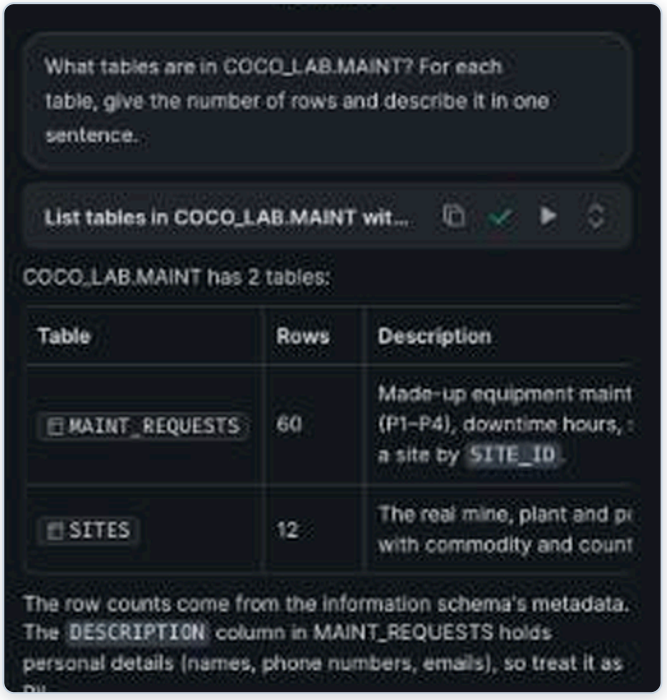
Why · Asking is safe: CoCo only reads. A full object name gives a better answer.

1. Open CoCo and start a **New chat**.

TYPE IN COCO

```
What tables are in COCO_LAB.MAINT? For each table, give the number of rows and
describe it in one sentence.
```

WHAT YOU SEE



Your answer looks like this.

✓ **Check** · MAINT_REQUESTS (60 rows) and SITES (12 rows). Reading metadata does not ask for approval.

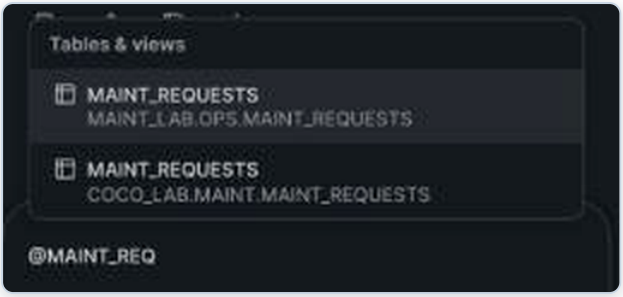
2.2 Point at one table with @

1. Type `@MAINT_REQ`.
2. In the list, select the **MAINT_REQUESTS** that shows **COCO_LAB.MAINT.MAINT_REQUESTS** under it. Other databases can have a table with the same name.
3. Type the rest of the prompt after the object chip.

TYPE AFTER THE @ OBJECT

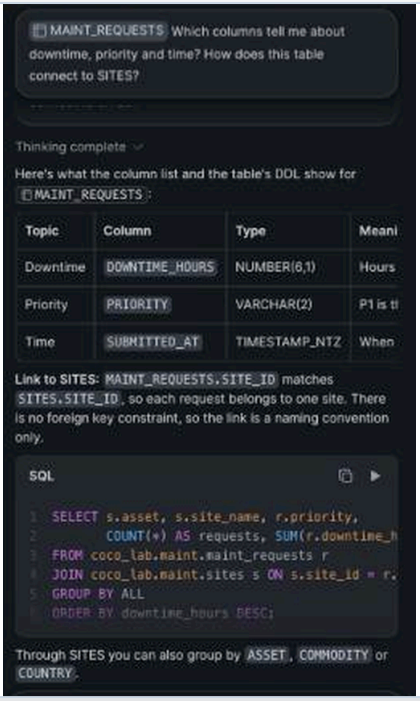
```
Which columns tell me about downtime, priority and time? How does this table connect
to SITES?
```

WHAT YOU SEE



Pick the COCO_LAB.MAINT table.

WHAT YOU SEE



The answer: columns and the SITE_ID join.

✓ **Check** · DOWNTIME_HOURS, PRIORITY, SUBMITTED_AT. SITE_ID joins to SITES (where ASSET is).

2.3 Look at some rows

TYPE IN COCO

```
Show 5 P1 requests from COCO_LAB.MAINT.MAINT_REQUESTS with the most downtime. Show request_id, site_id, equipment_id, failure_mode and downtime_hours. Do not show the description.
```

WHAT YOU SEE



Your answer looks like this.

✓ **Check** · First rows: **DL-0101** (BM01), **SL-0101** (WA04), **SR-6201** (BM04), all structural cracks.

2.4 Ask the documentation

Why · CoCo answers from the Snowflake docs, so you learn a feature without leaving Snowsight.

TYPE IN COCO

```
Using the Snowflake documentation: what is a semantic view, and why does Cortex Analyst need one? Answer in 3 bullets.
```

✓ **Check** · It stores business names (dimensions, metrics, relationships). Cortex Analyst uses it to write correct SQL.

Tip · Under many answers CoCo shows a suggested next question. Select it to ask it.

3 Write and explain SQL

CoCo writes a query into your file. Then you explain, edit and format it.

Level 2 · 25 min · Parts 3, 4

3.1 Make a new SQL file

Why · Work you keep belongs in a file you can save, share and run again.

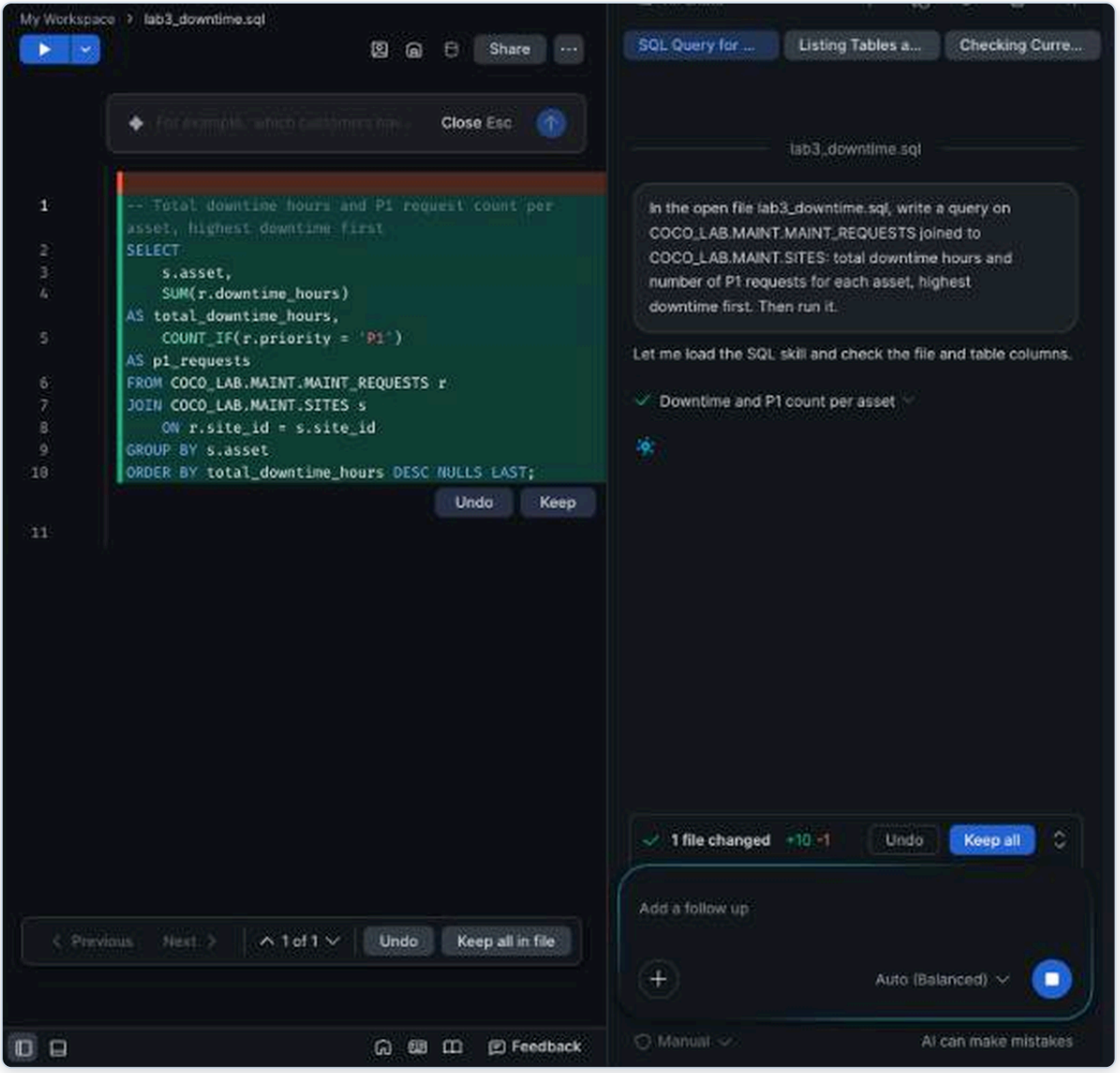
1. Workspaces › **+ Add new** › **SQL file**.
2. The name field opens at once. Type `lab3_downtime` and push Enter (.sql is added).
3. The empty file shows a describe box at the top. You can use it, or the CoCo panel.

3.2 CoCo writes a query into the file

1. Open CoCo and start a **New chat**.

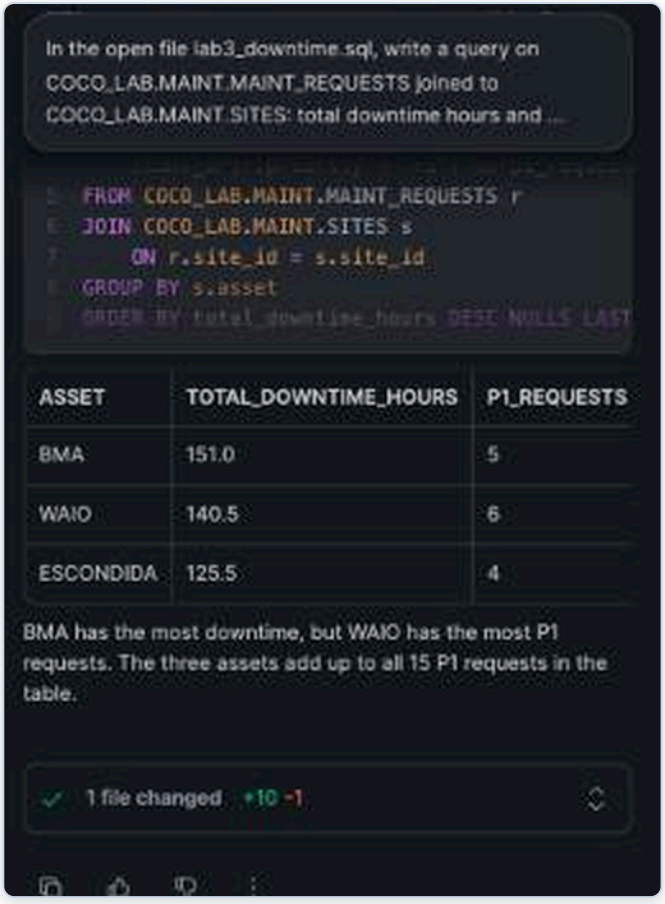
> **TYPE IN COCO**
>
> ```
> In the open file lab3_downtime.sql, write a query on COCO_LAB.MAINT.MAINT_REQUESTS
> joined to COCO_LAB.MAINT.SITES: total downtime hours and number of P1 requests for
> each asset, highest downtime first. Then run it.
> ```

WHAT YOU SEE



The new lines show green in the editor. The panel shows 1 file changed with Undo and Keep all.

WHAT YOU SEE



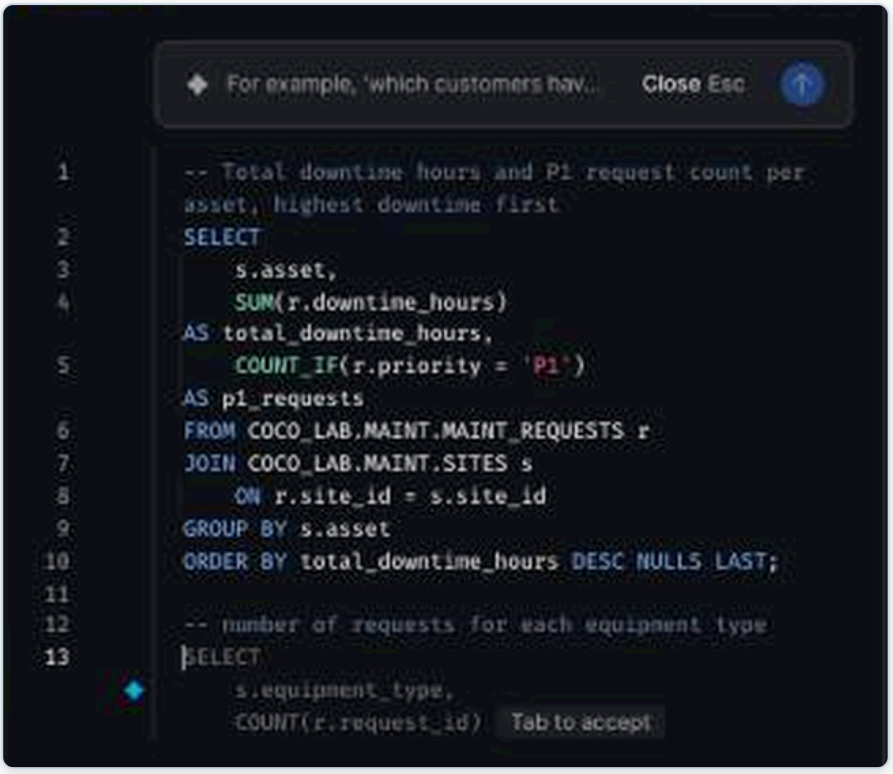
CoCo runs the SELECT (no approval needed to read) and shows the result.

✓ **Check** · **BMA 151.0** (5 P1), **WAIO 140.5** (6 P1), **ESCONDIDA 125.5** (4 P1): 15 P1 requests in total. (Check 3) Then select **Keep all**.

3.3 Accept an inline suggestion

1. Click at the end of the file. Push Esc to close the describe box.
2. Push Enter, type `-- number of requests for each equipment type` and push Enter.
3. Gray text with **Tab to accept** shows. Push Tab. It comes in parts: push Tab again until the query ends.
4. Run it with Ctrl + Enter (Mac Cmd + Return).

WHAT YOU SEE



The inline suggestion.

Caution · A suggestion can be wrong. In our test it wrote `s.equipment_type` and the run failed with *invalid identifier*. Select **Fix** under the error, then **Keep all**. This is Lab 4 in short.

✓ **Check** · Conveyor 12 and Haul truck 11 are the biggest numbers (the rows are not sorted).

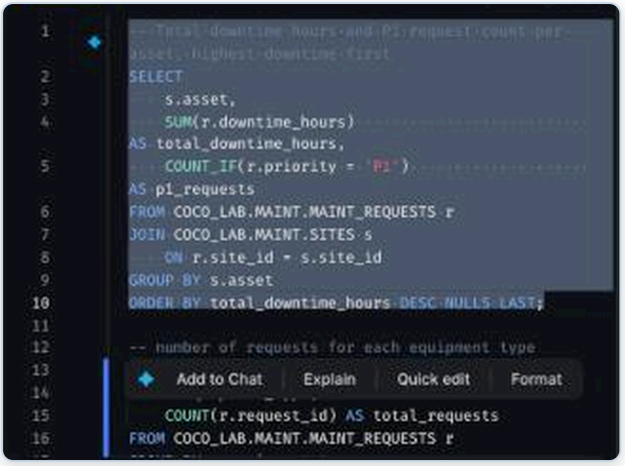
3.4 Explain, Quick edit, Format

1. Highlight the query of 3.2. A toolbar shows **Add to Chat** · **Explain** · **Quick edit** · **Format**.
2. Select **Explain**. Read the answer in the CoCo panel.
3. Highlight again › **Quick edit**. Type the text below, push Enter, then **Keep all**.
4. Run the query again.

TYPE IN QUICK EDIT

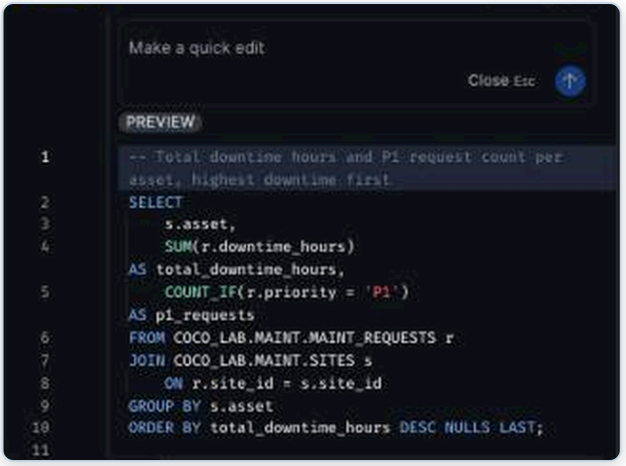
```
Add a column with the average downtime hours for each request, rounded to 1 decimal place.
```

WHAT YOU SEE



The toolbar on highlighted SQL.

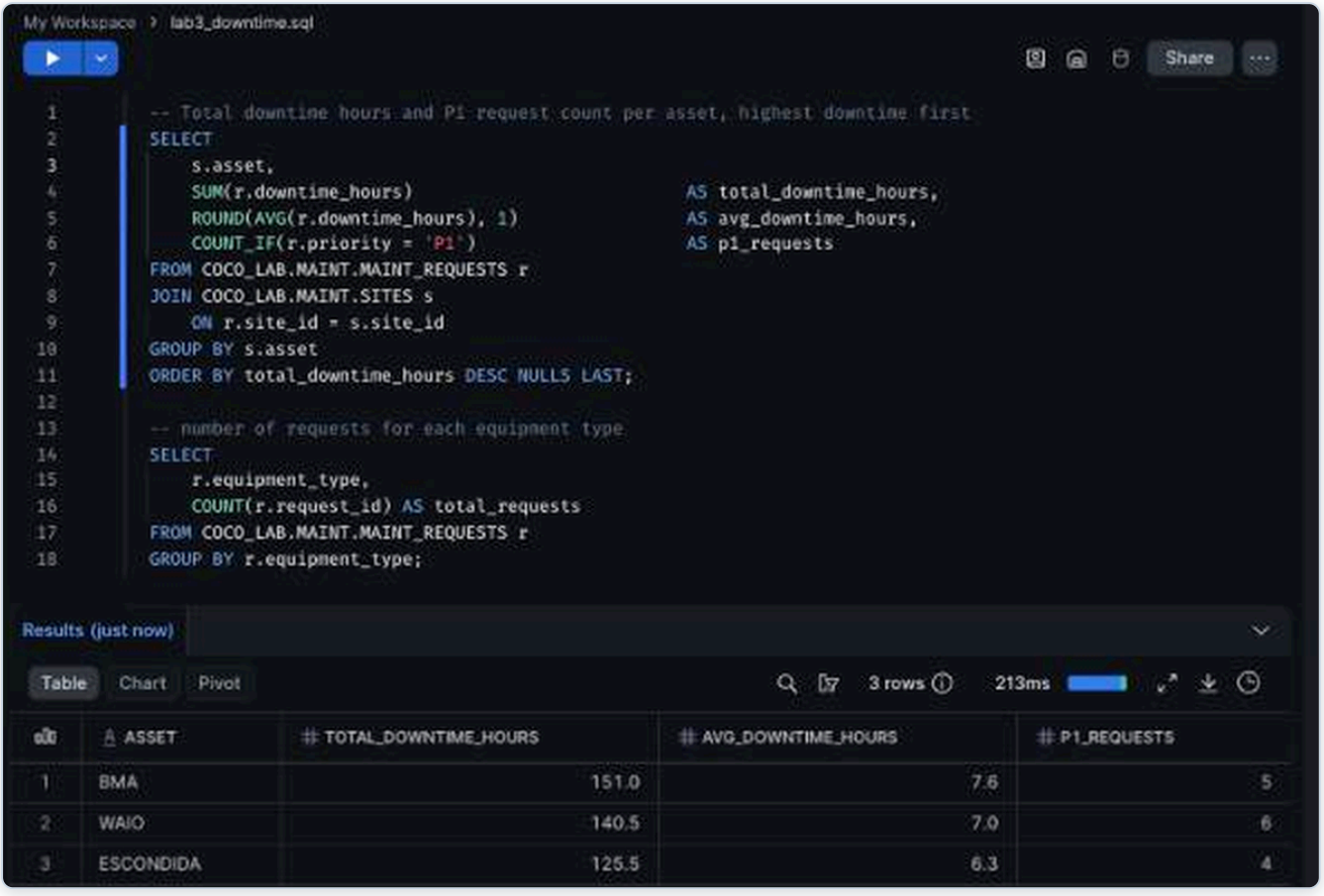
WHAT YOU SEE



The Quick edit box with a preview.

✓ **Check** · A new column AVG_DOWNTIME_HOURS: **BMA 7.6**, WAIO 7.0, ESCONDIDA 6.3.

WHAT YOU SEE



The query after Quick edit, and its result.

4 Fix broken queries

Three queries fail on purpose. CoCo finds and fixes each error.

Level 3 · 20 min · Parts 3, 4, 7

4.1 Fix each query with Fix

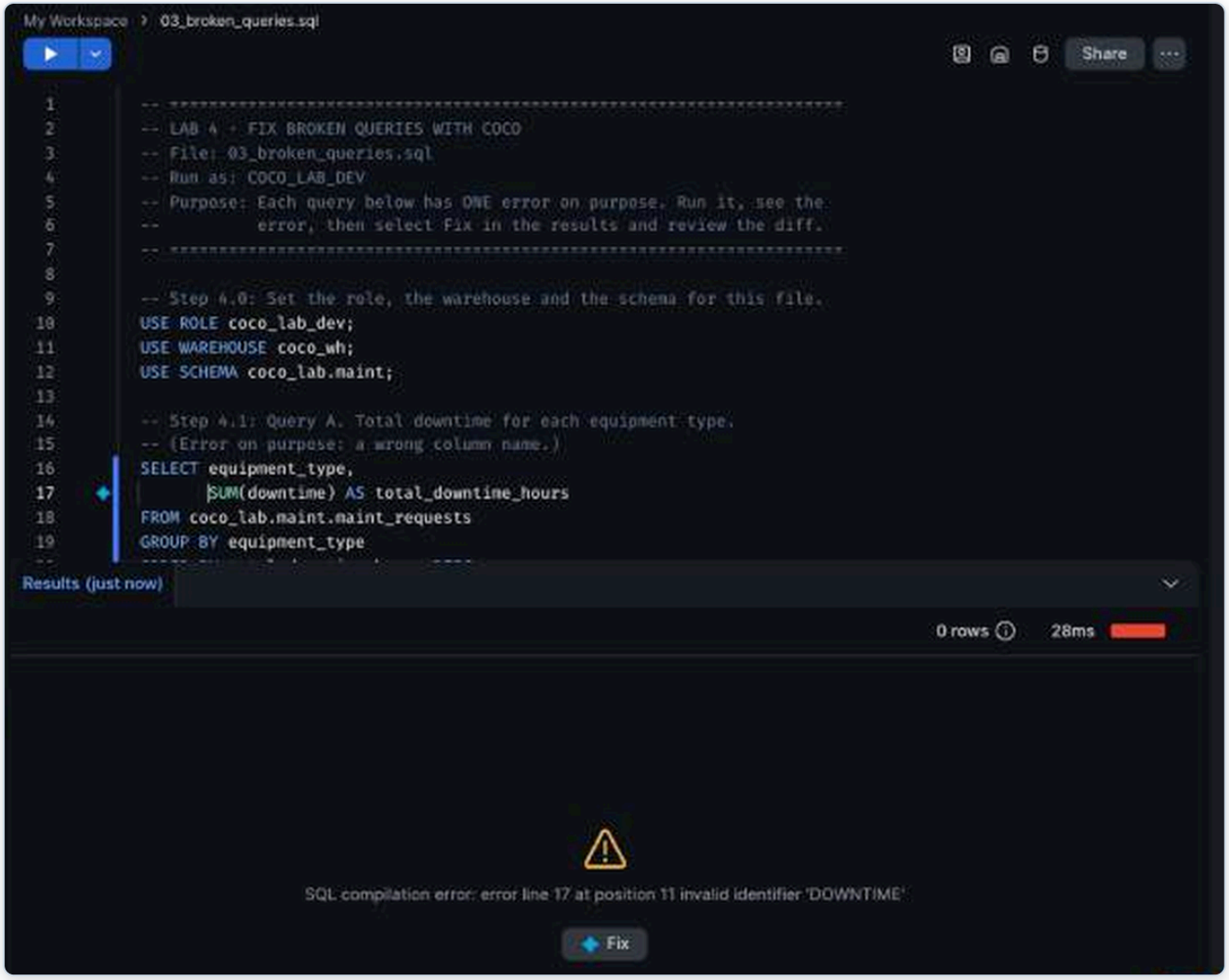
Why · The Fix button gives CoCo the error and the query. You stay in control: you read and accept the diff.

1. Open `03_broken_queries.sql`. Run **Step 4.0**.
2. Put the cursor in query A (Step 4.1) and run it with Ctrl + Enter (Mac Cmd + Return). An error shows.
3. Under the error, select **Fix**. CoCo opens with `/sql-author Fix this error...`
4. Read the diff (red removed, green added) › **Keep all**. Run again.
5. Do the same for query B (Step 4.2) and query C (Step 4.3).

Query	Error you see	Correct after the fix
A	*invalid identifier 'DOWNTIME'*	SUM(downtime_hours). Haul truck 91.5 first, then Dragline 49.0.
B	*'MAINT_REQUESTS.FAILURE_MODE' in select clause is neither an aggregate nor in the group by clause*	GROUP BY site_id, failure_mode. 53 rows.
C	*Division by zero*	DIV0NULL or NULLIF. 12 rows. BM01 shows 43. ES04 shows 0 (or NULL).

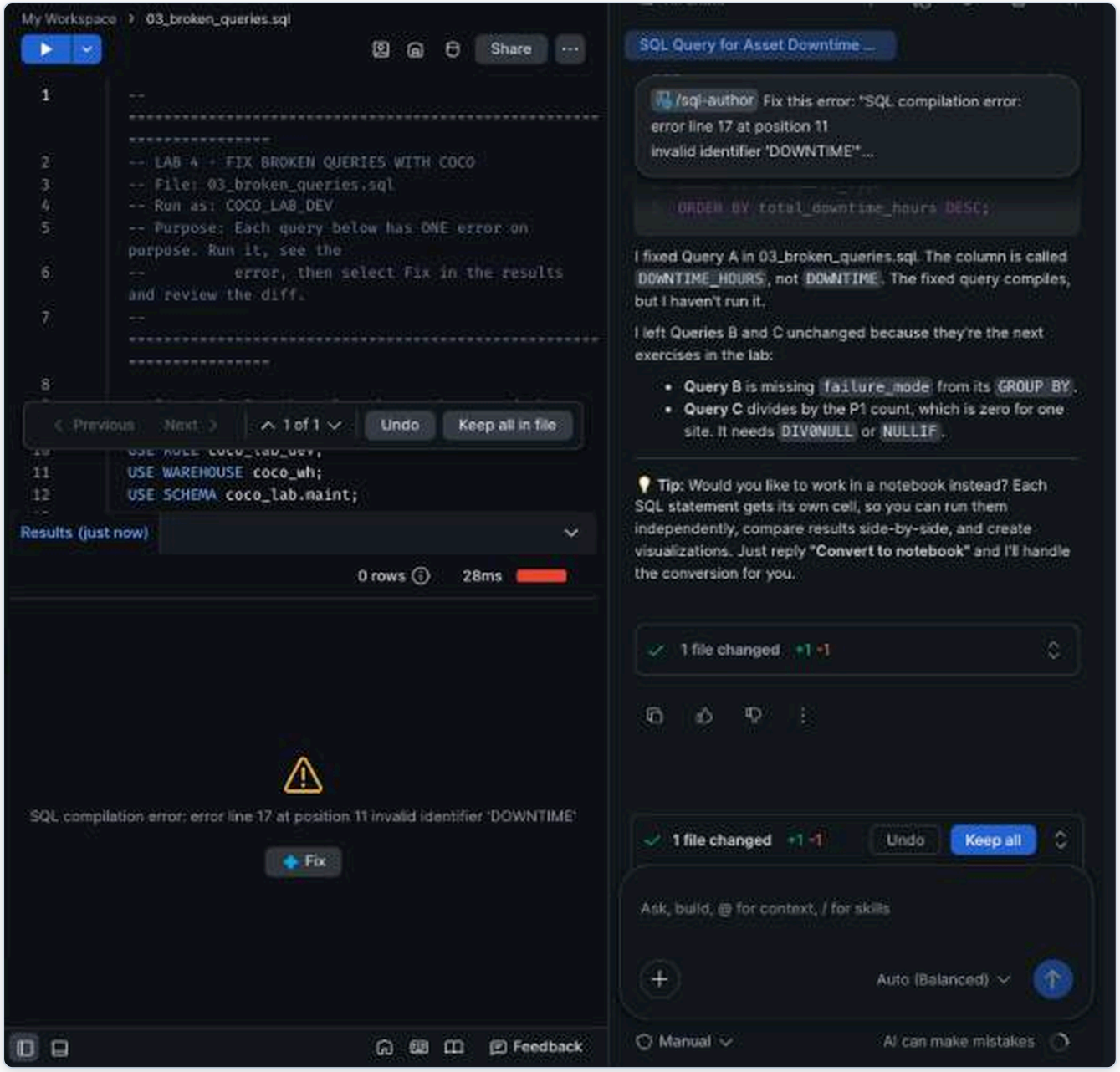
✓ **Check** · All 3 queries run with no error. (Check 4)

WHAT YOU SEE



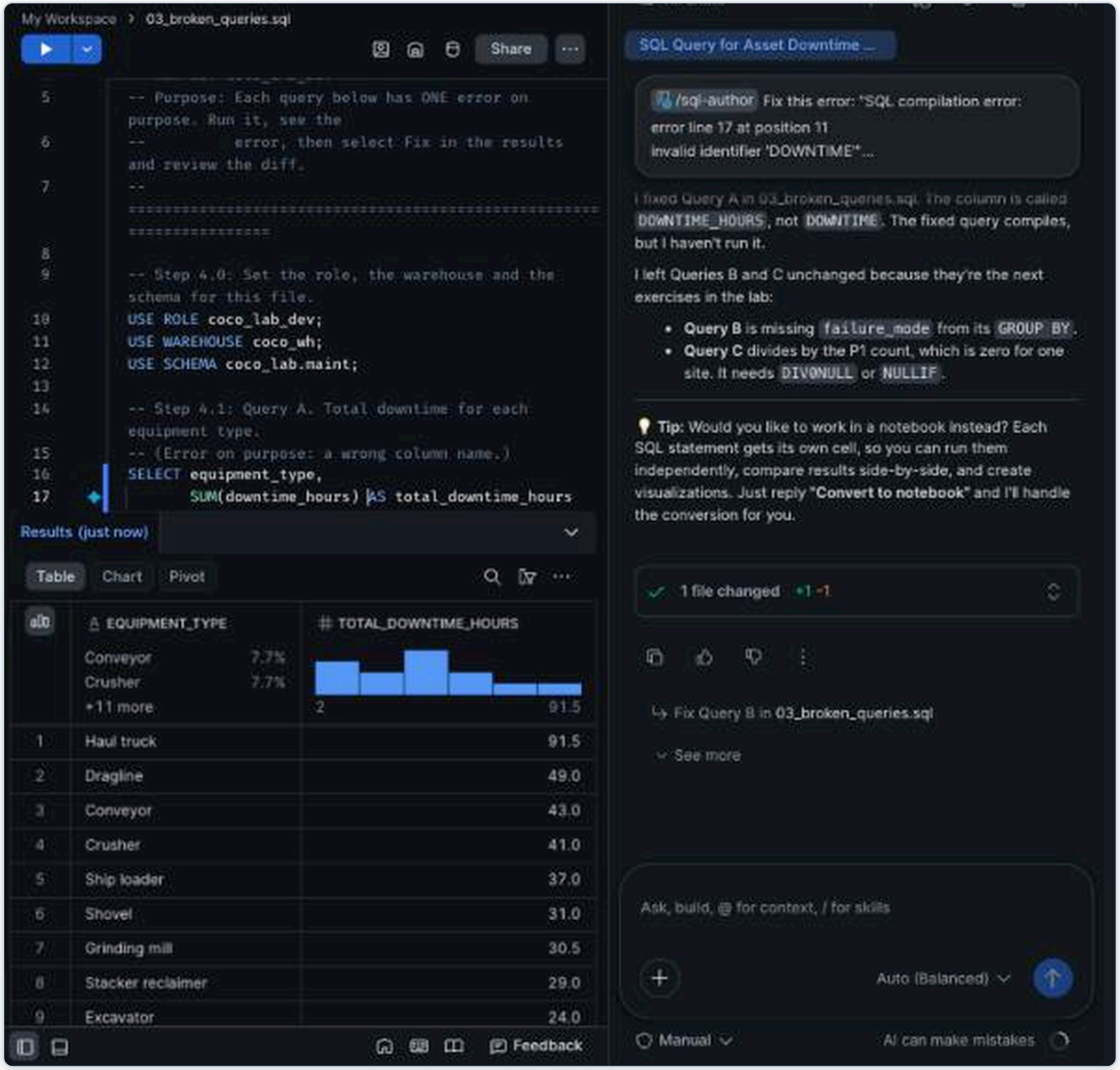
Query A fails. The Fix button is under the error.

WHAT YOU SEE



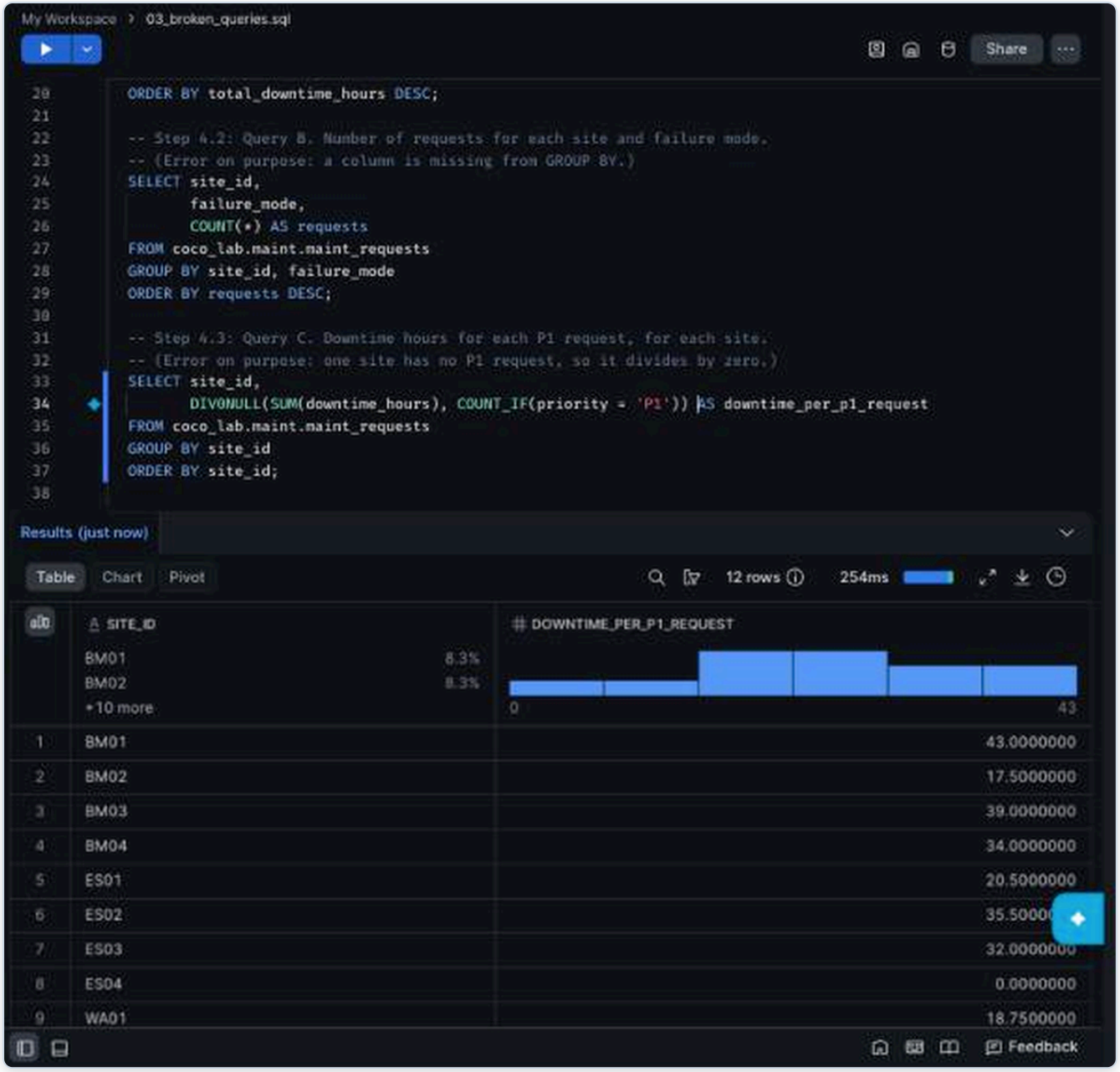
After Fix: the diff in the editor and CoCo's explanation. Select Keep all.

WHAT YOU SEE



Query A after the fix: Haul truck 91.5, Dragline 49.0.

WHAT YOU SEE



Query C after the fix: BM01 43, ES04 0.

4.2 Ask CoCo to improve the code

1. Highlight query C › **Add to Chat**. The code shows as a chip in the message box.
2. Type the text below after it.

> **TYPE IN COCO**
> `Explain why the division failed. Then add a comment line above each part of this query, for a new team member. Do not change the result.`

✓ **Check** · Site ES04 has no P1 request, so the divisor is 0. The diff adds only comment lines. Select **Keep all**.

5 Notebook with charts

CoCo builds a notebook: SQL, two charts and written findings.

Level 4 · 25 min · Parts 3, 4, 5

5.1 Ask CoCo to build the notebook

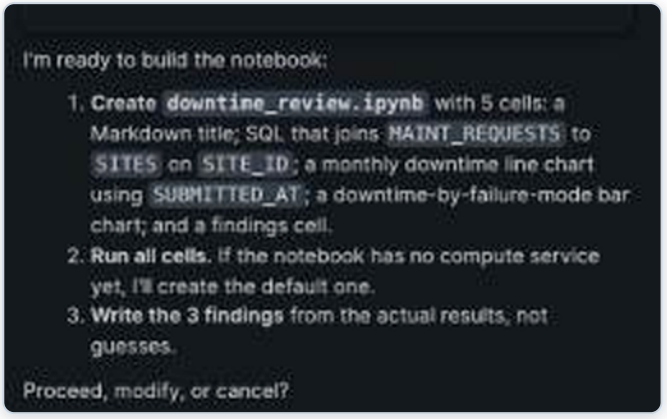
Why · A notebook keeps the question, code, charts and conclusion together, so others can read and rerun it.

1. Open CoCo and start a **New chat**.

TYPE IN COCO

```
Create a notebook called downtime_review in my workspace. Cell 1: a Markdown title.
Cell 2: SQL that loads COCO_LAB.MAINT.MAINT_REQUESTS joined to COCO_LAB.MAINT.SITES.
Cell 3: a line chart of total downtime hours per month. Cell 4: a bar chart of total
downtime hours per failure mode. Cell 5: Markdown with 3 findings. Then run all
cells.
```

WHAT YOU SEE

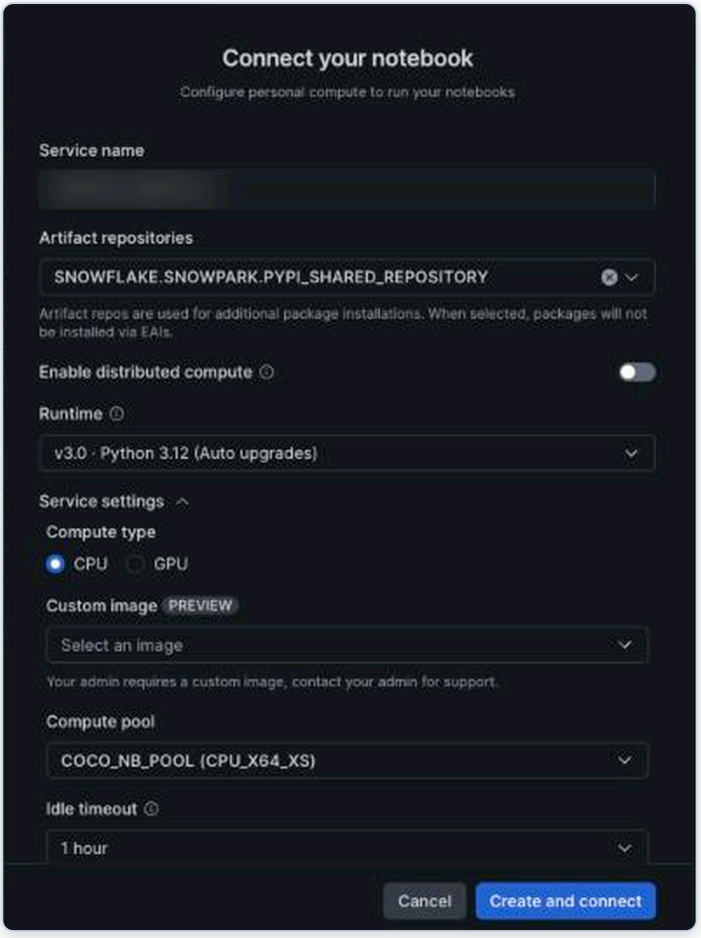


CoCo shows its plan first and asks: Proceed, modify, or cancel?

5.2 Proceed and connect compute

1. Type `Proceed. Use the compute pool COCO_NB_POOL.`
2. CoCo creates `downtime_review.ipynb` and opens **Connect your notebook**.
3. Open **Service settings**. **Compute pool**: **COCO_NB_POOL**. **Idle timeout**: **1 hour** (the default is 24 hours).
4. Select **Create and connect**. Wait for *Setting up service* to finish (3 to 5 minutes).
5. Type `The service is ready. Run all cells.` Select **Allow** for *Run notebook cells?*

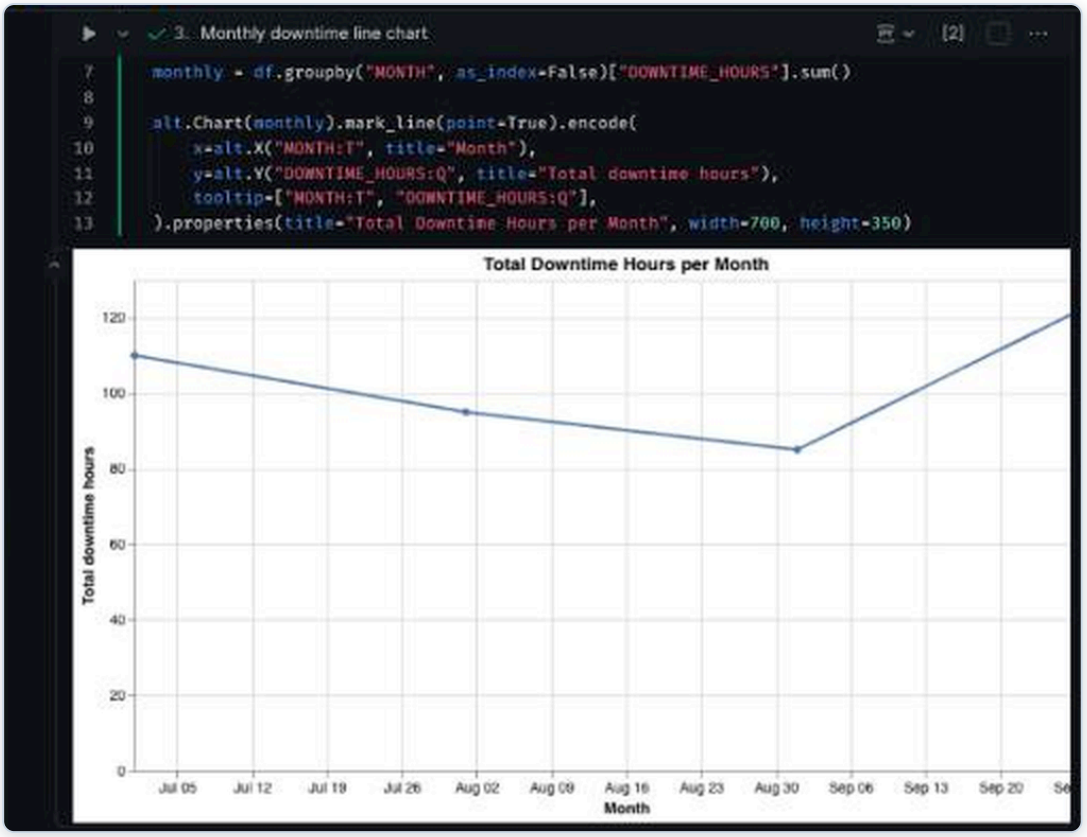
WHAT YOU SEE



Connect your notebook: pick COCO_NB_POOL and a short idle timeout.

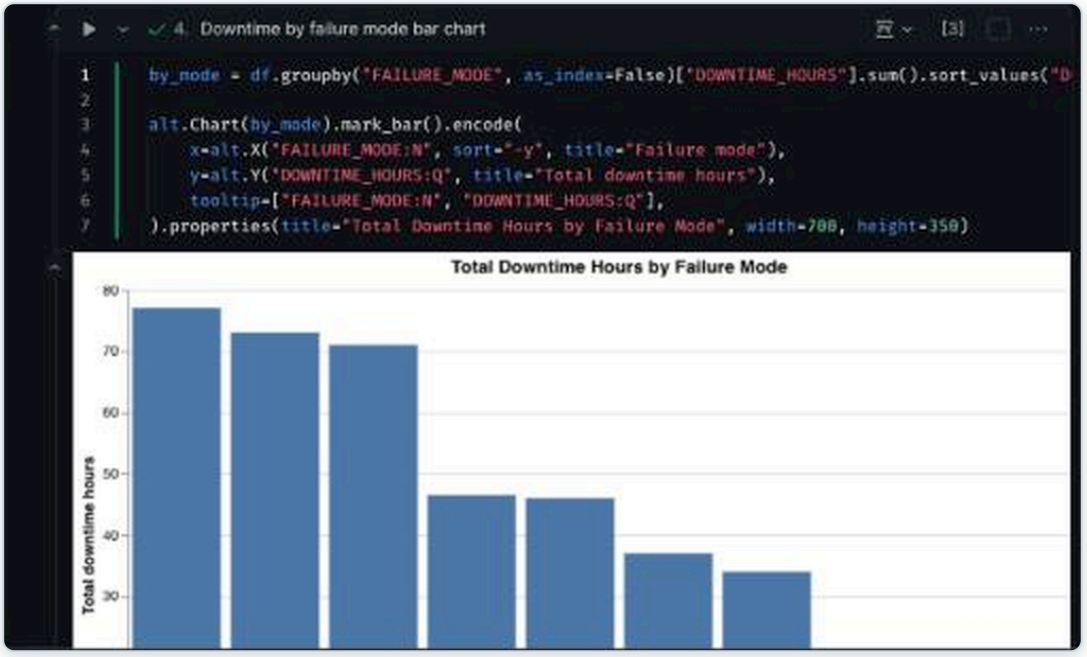
5.3 Read the charts

WHAT YOU SEE



Cell 3: total downtime hours per month.

WHAT YOU SEE



Cell 4: downtime per failure mode, high to low.

✓ **Check** · Line chart: **Jul 110**, **Aug 95**, **Sep 85**, **Oct 127** hours. Bars: **Electrical 77**, **Structural crack 73**, **Bearing 71** on top. Findings: 417 hours in total. (Check 5)

5.4 Change a chart

TYPE IN COCO

```
In downtime_review, change the bar chart: show only the top 5 failure modes and sort
the bars from high to low. Run the cell again.
```

✓ **Check** · 5 bars, high to low.

Caution · **No compute pool?** Ask CoCo: `Write the two chart queries into a new SQL file downtime_charts.sql and run them.` Then select **Chart** in the Results pane.

6 Plan and build

Use plan mode. Build a KPI view and a semantic view.

Level 5 · 25 min · Parts 4, 6, 7

6.1 Ask for a plan first

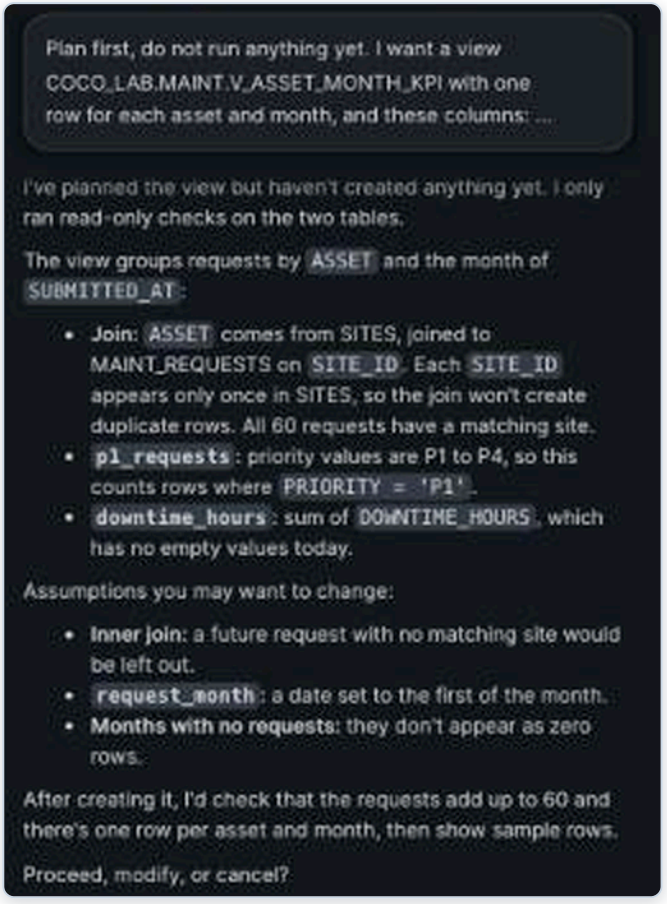
Why · Building objects changes the account. Plan mode shows the steps first and is read-only until you approve.

1. Open CoCo and start a **New chat**.
2. In the message box, select **+** › **Plan mode**. A **Plan** chip shows in the box.

TYPE IN COCO

```
Plan first, do not run anything yet. I want a view COCO_LAB.MAINT.V_ASSET_MONTH_KPI
with one row for each asset and month, and these columns: asset, request_month,
requests, downtime_hours, p1_requests. Use MAINT_REQUESTS joined to SITES.
```

WHAT YOU SEE



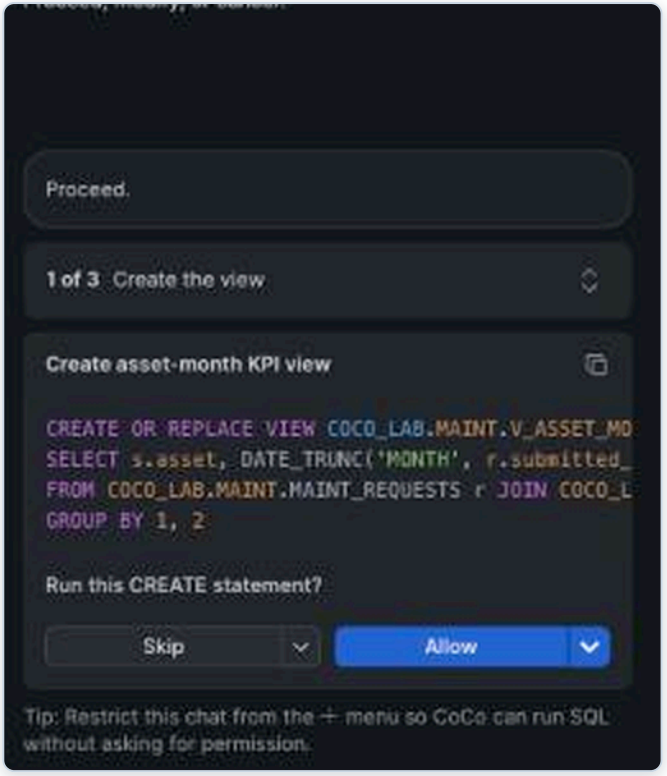
The plan.

✓ **Check** · CoCo runs read-only checks, shows a plan and ends with *Proceed, modify, or cancel?* Nothing is created yet.

6.2 Approve and run

1. Remove the **Plan** chip (x). Type `Proceed.`
2. CoCo works in steps (1 of 3, 2 of 3, 3 of 3). For *Run this CREATE statement?* select **Allow**.

WHAT YOU SEE



Step 1 of 3 waits for your Allow.

✓ **Check** · The view is created. CoCo checks that the requests add up to 60. Check 6 (first query): BMA 151.0, WAIO 140.5, ESCONDIDA 125.5, 15 P1.

6.3 Build a semantic view

Why · A semantic view gives the data business names. Cortex Analyst and agents use it. CoCo writes the long statement for you.

1. Allow the CREATE SEMANTIC VIEW statement, then the query (labelled **[Grounded: MAINT_SV]**).

TYPE IN COCO

```
Create a semantic view COCO_LAB.MAINT.MAINT_SV on MAINT_REQUESTS and SITES
(relationship on SITE_ID). Dimensions: asset, site_name, equipment_type,
failure_mode, priority, request month. Metrics: request_count, downtime_total (sum
of downtime_hours), p1_count. Add a short comment to each. Then query it:
downtime_total by asset.
```

WHAT YOU SEE



The semantic view gives the same totals.

✓ **Check** · BMA 151.0, WAIO 140.5, ESCONDIDA 125.5: the same as Lab 3. Check 6 (second query) lists **MAINT_SV**.

7 Enrich text with AI functions

Hide personal data, add a category and a summary, find repeat problems.

Level 5 · 20 min · Parts 5, 6

7.1 Make an AI-enriched table

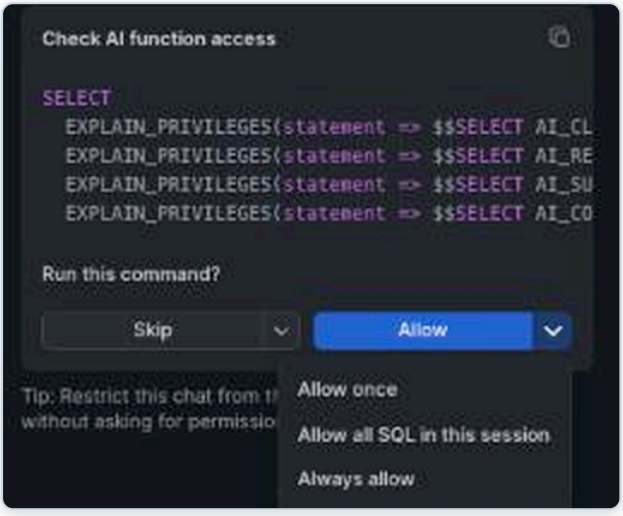
Why · The descriptions hold names and phone numbers, and free text that is hard to count. AI functions in SQL fix both.

1. Open CoCo and start a **New chat**.
2. CoCo first checks access and estimates the cost (about 7K tokens).
3. Expect 4 to 5 **Allow** prompts. CoCo can fix its own SQL on the way (for example, it builds the table in two steps).
4. The Allow arrow has **Allow once**, **Allow all SQL in this session** and **Always allow**. Keep **Allow once** today.

TYPE IN COCO

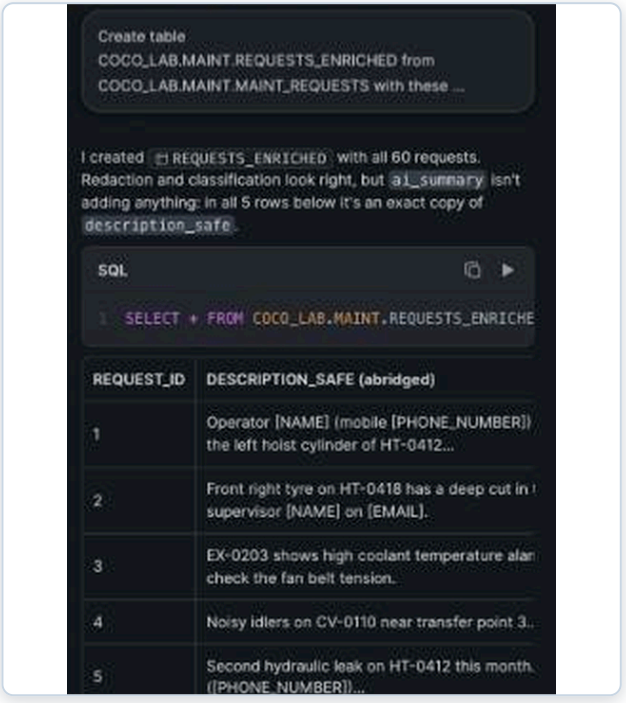
```
Create table COCO_LAB.MAINT.REQUESTS_ENRICHED from COCO_LAB.MAINT.MAINT_REQUESTS
with these columns: request_id, description_safe = AI_REDACT(description),
failure_category = the top label of AI_CLASSIFY(description_safe) with the labels
Mechanical, Electrical, Hydraulic, Structural, Tyres and brakes, and ai_summary =
AI_SUMMARIZE(description_safe). Then show 5 rows.
```

WHAT YOU SEE



The three Allow options.

WHAT YOU SEE



The table with redacted text.

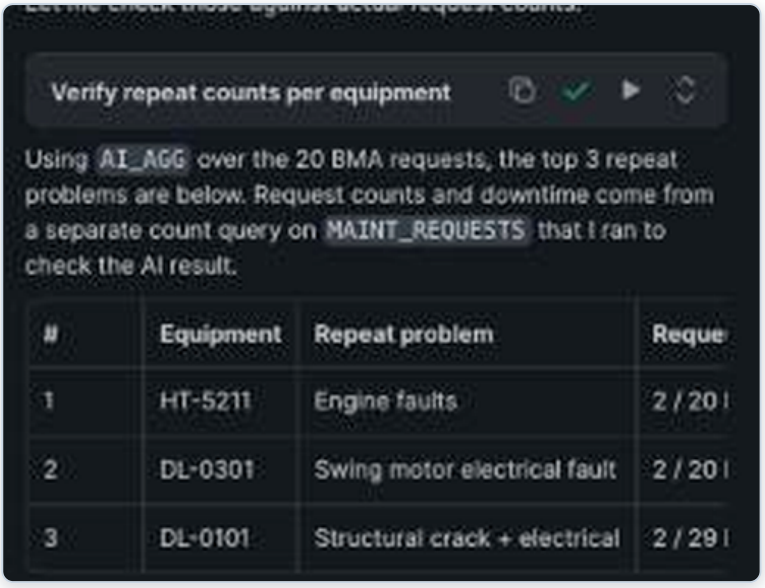
✓ **Check** · Check 7: 60 rows, 60 categories, 0 visible names. Names show as [NAME], phones as [PHONE_NUMBER]. For short text, **ai_summary** is the same as the text: AI_SUMMARIZE does not shorten very short input.

7.2 Find repeat problems with AI_AGG

TYPE IN COCO

```
For the BMA asset only, use AI_AGG on description_safe from
COCO_LAB.MAINT.REQUESTS_ENRICHED (join to MAINT_REQUESTS and SITES) to find the top
3 repeat equipment problems. Give the equipment ID and one action for each.
```

WHAT YOU SEE



AI_AGG result, cross-checked with a count query.

✓ **Check** · 3 IDs from the BMA repeat list (Check 8), for example HT-5211, DL-0301, DL-0101. CoCo can also say that 5 machines tie with 2 requests each. That is correct.

8 AGENTS.md and your own skill

Give CoCo team rules, and save a repeat task as a skill.

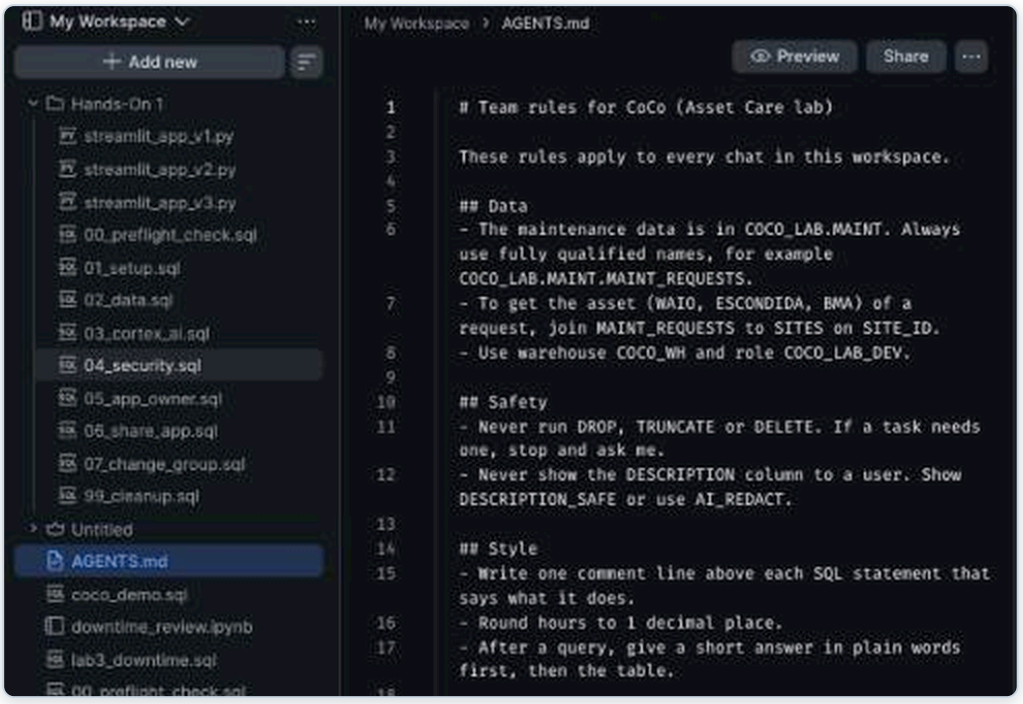
Level 6 · 25 min · Parts 3, 4

8.1 Add AGENTS.md

Why · AGENTS.md gives CoCo your team rules in every chat, so nobody has to repeat them.

1. In Workspaces, at the top (root) of **My Workspace**: **+ Add new** › **Upload files** › `AGENTS.md`.
2. Or: **+ Add new** › **File**, name it `AGENTS.md`, paste the text.
3. Check the path at the top of the editor: **My Workspace › AGENTS.md**.

WHAT YOU SEE



AGENTS.md at the root, with the 7 rules.

Tip · AGENTS.md must be at the root, not in a folder. Start a new chat after you add it.

8.2 See the rules work

1. Open CoCo and start a **New chat**.

TYPE IN COCO

```
Show the 3 requests with the most downtime hours, with the site name and the asset.
```

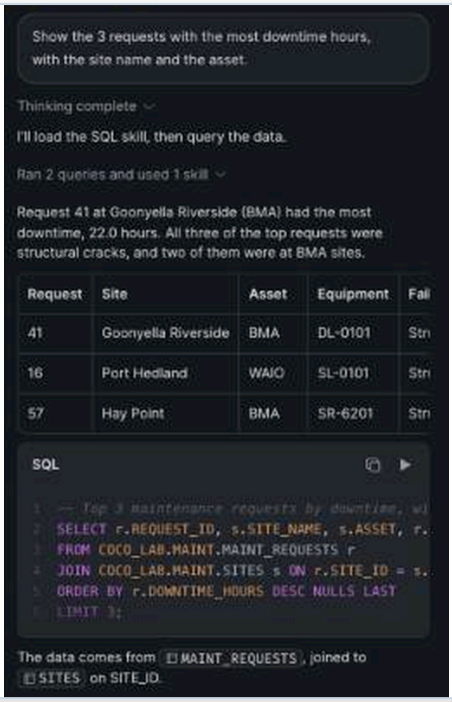
✓ **Check** · A short answer first (request 41 at Goonyella Riverside, 22.0 hours), then the table. Full object names, a comment line above the SQL, no DESCRIPTION column. These came from AGENTS.md.

8.3 Test a safety rule

TYPE IN COCO

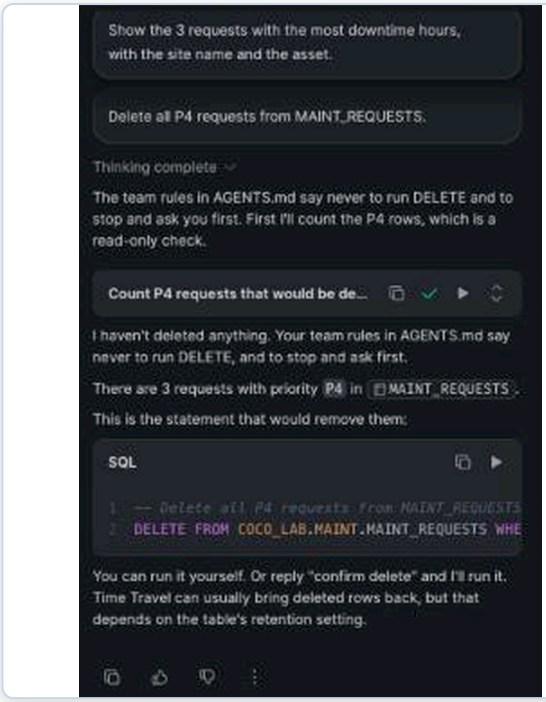
```
Delete all P4 requests from MAINT_REQUESTS.
```

WHAT YOU SEE



8.2: the rules shape the answer.

WHAT YOU SEE



8.3: CoCo stops and asks.

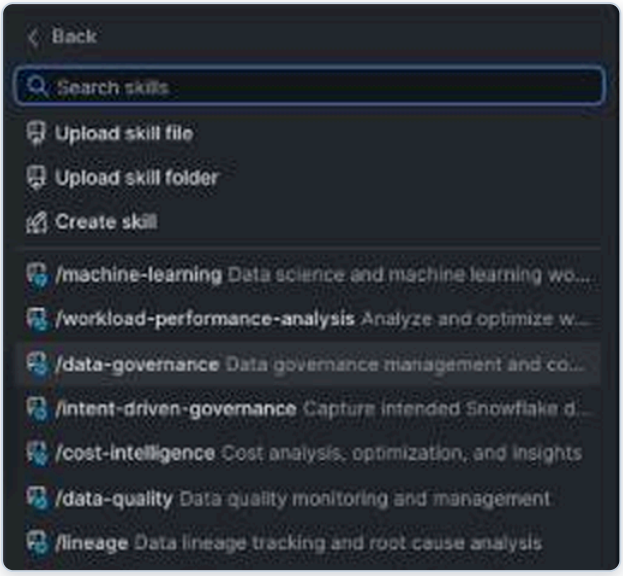
✓ **Check** · *I haven't deleted anything.* CoCo names the AGENTS.md rule, counts the 3 P4 rows and shows the DELETE for you. Answer `No, stop.`

8.4 Add the skill

Why · The weekly report has 4 fixed steps. As a skill, it runs the same way for anyone, with one command.

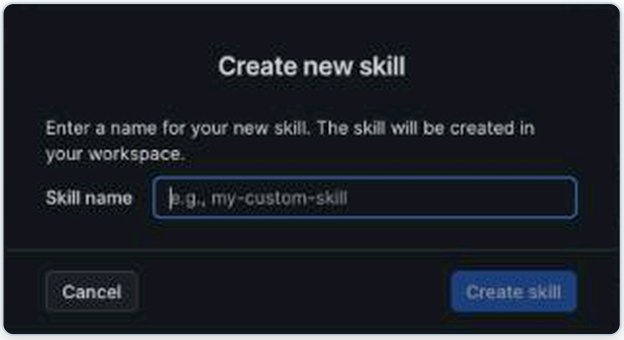
1. In the message box, select **+** › **Skills** › **Create skill**.
2. Skill name `weekly-asset-report` › **Create skill**.
3. The file `.snowflake/cortex/skills/weekly-asset-report/SKILL.md` opens, and CoCo starts a chat to help you fill it. Stop it (square button).
4. Select **Edit**, replace all text with the text of the lab file `SKILL.md`. It saves by itself.
5. Or: **Upload skill folder** › folder `weekly-asset-report`.

WHAT YOU SEE



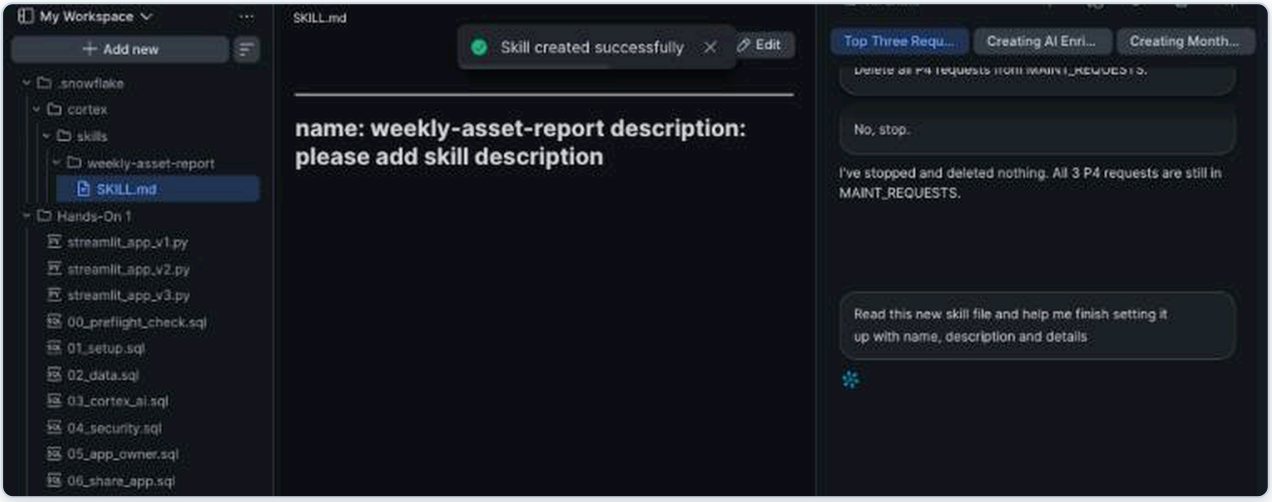
+ › Skills.

WHAT YOU SEE



Create new skill.

WHAT YOU SEE



After Create skill: the SKILL.md file and the help chat.

8.5 Run the skill

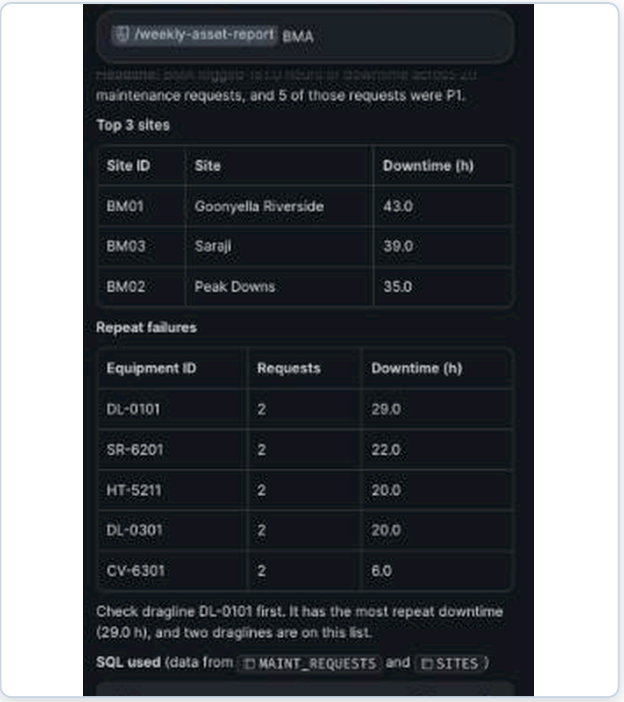
1. Open CoCo and start a **New chat**.
2. Type `/weekly`, select **/weekly-asset-report** in the list, type `BMA`, push Enter.

WHAT YOU SEE



Your skill in the / list.

WHAT YOU SEE



The report.

Tip · A personal skill works only in this workspace. To share it, use the built-in **share-skill** skill, if your account has **AI & ML › Skills and plugins**.

✓ **Check** · Headline: **151.0 hours**, 20 requests, **5 P1**. Top sites: Goonyella Riverside 43.0, Saraji 39.0, Peak Downs 35.0. Repeats: DL-0101 29.0, SR-6201 22.0, HT-5211 20.0, DL-0301 20.0, CV-6301 6.0. (Check 8). Then try `/weekly-asset-report WAIO` : 140.5 hours.

9 Control and cost

Use a read-only chat, set a daily credit limit, and check the warehouse cost.

Control · 20 min · Parts 1, 7

9.1 A read-only chat

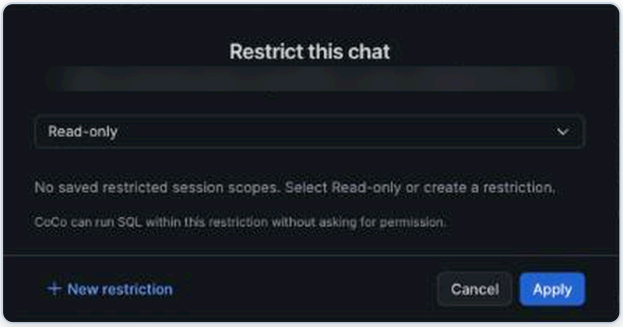
Why · CoCo is an agent: it can change things. On important data, limit what it can do.

1. Open CoCo and start a **New chat**.
2. In the message box, select **+** › **Restrict this chat**. Keep **Read-only** › **Apply**.
3. The chat tab shows a lock icon.

TYPE IN COCO

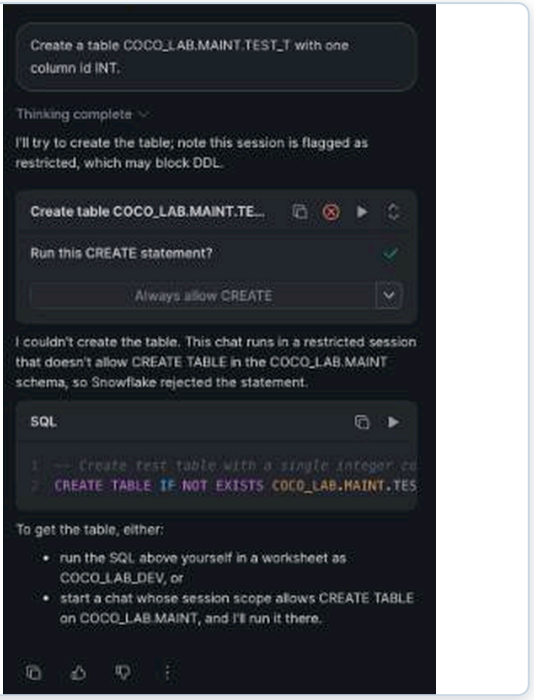
```
Create a table COCO_LAB.MAINT.TEST_T with one column id INT.
```

WHAT YOU SEE



Restrict this chat.

WHAT YOU SEE



The CREATE is rejected.

✓ **Check** · Snowflake rejects the CREATE: the chat is read-only. Nothing is created.

9.2 Reading still works

TYPE IN COCO

```
How many requests does each site have? Read only.
```

✓ **Check** · 5 requests for each of the 12 sites. In a read-only chat CoCo runs SELECTs without asking. To remove the restriction, start a new chat.

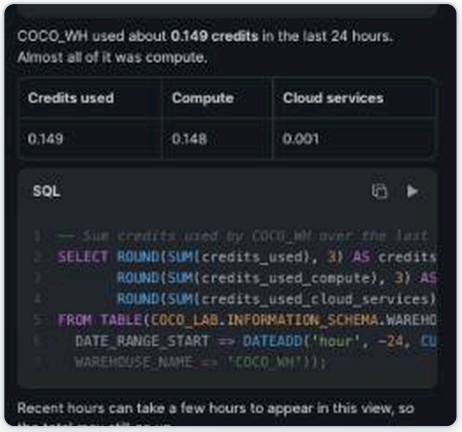
9.3 Ask CoCo about cost

1. Open CoCo and start a **New chat**.

TYPE IN COCO

```
How many credits did warehouse COCO_WH use in the last 24 hours? Use
INFORMATION_SCHEMA.WAREHOUSE_METERING_HISTORY in COCO_LAB.
```

WHAT YOU SEE



Your answer looks like this.

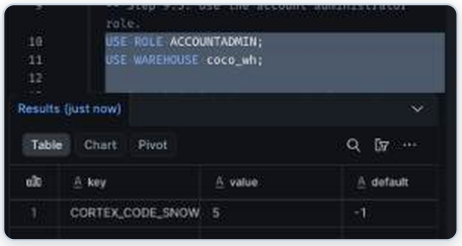
✓ **Check** · A small number, less than 1 credit (our run: 0.149). The data can come a few hours late.

9.4 Set and remove a daily limit (admin)

Why · The daily limit stops one user from spending too much on CoCo. Default -1 is no limit. 0 blocks CoCo.

1. Open `05_admin_cost.sql`.
2. Highlight from **Step 9.5** to the end of **Step 9.7** (the last `$$;` before Step 9.8). Run.
3. Highlight from **Step 9.8** to the end of the file. Run.

WHAT YOU SEE



Your user now has a daily CoCo limit of 5 credits.

Caution · Always run Step 9.8. With a limit of 5, CoCo stops for you after 5 credits in 24 hours.

✓ **Check** · After 9.7: value **5**, default -1. After 9.9: credits of COCO_WH. Run the 9.7 SHOW block again: value is back to -1.

9.5 Optional: an automation

1. In the CoCo panel header, select **Automations** (clock icon) › **+ New**.
2. Title `Daily BMA downtime`, instruction `Run /weekly-asset-report BMA and give the headline only.`, frequency daily.
3. Look at the next run time, then **delete** it, so it uses no credits after the training.

Checkpoint

I can now	Done
Chat with CoCo as COCO_LAB_DEV (Lab 1, 2)	☐
Let CoCo write SQL into a file, and use Explain, Format, Quick edit (Lab 3)	☐
Fix errors with Fix and read the diff (Lab 4)	☐
Get a notebook with charts (Lab 5)	☐
Build a view and a semantic view with plan mode (Lab 6)	☐
Use AI functions through CoCo (Lab 7)	☐
Give CoCo rules and run my own skill (Lab 8)	☐
Restrict a chat and control cost (Lab 9)	☐

If something goes wrong

Problem	Fix
No CoCo icon, or no answer	Add a credit card (C2). Not Azure East US. Refresh the page.
Daily credit usage limit reached	Run Step 9.8, or ask the admin.
Insufficient privileges	Snowsight role must be COCO_LAB_DEV (Step 1.3).
Wrong table or a made-up column	Use full names or @. Ask CoCo to show the SQL. Run the check.
Answers differ between people	Normal. The numbers must match the ✓ Check.
Notebook does not start	In Connect your notebook, pick COCO_NB_POOL and wait 3 to 5 minutes, or use the Lab 5 fallback.
@ list shows two tables with the same name	Pick the one with COCO_LAB.MAINT under it.
CoCo waits and does nothing	Scroll the panel: it may ask Allow, or Proceed, modify, or cancel.
CoCo ignores AGENTS.md	File at the workspace root. Start a new chat.
Skill not in the / list	Path .snowflake/cortex/skills/weekly-asset-report/SKILL.md in the same workspace. Refresh.
Menu item not where the guide says	Use the search box at the top of the navigation menu.

Clean up (after the training)

Switch to ACCOUNTADMIN, open `99_cleanup.sql`, push Ctrl + Shift + Enter (Mac Cmd + Shift + Return). Delete the notebook, AGENTS.md and the skill folder in Workspaces by hand.

Sources (checked 16 November 2026): docs.snowflake.com pages for CoCo in Snowsight, its changelog, Skills and plugins, Daily credit usage limits, CoCo automations, Trial accounts, Snowsight navigation, and Notebooks in Workspaces compute. The full lab was run end to end in a Snowflake trial account on 17 November 2026. All screenshots and numbers come from that run. Names in the screenshots are blurred.