CIPHER RELAY NODE / DOCUMENT CHANNEL | CPH-DB-REF

HELIOS INTERNAL DATABASE

DB CONSOLE

調査用操作手引書 / FIELD GUIDE

DOCUMENT	CPH-DB-REF	REVISION	REV.05
MODE	READ-ONLY	SOURCE	CIPHER RELAY

CIPHER // TRANSMISSION 00

公開 HP で得た断片だけでは、HELIOS が隠した記録には届かない。事故日、部署名、人物名を手掛かりに、外部中継された読み取り専用 DB から事実を取り出して。構文の役割、照合するキー、調査の考え方だけを示す。

まず覚える 4 点

- **公開情報を持ち込む：**HP で見つけたキーワードを検索条件に使う。
- **CIPHER の順序に従う：**必要なテーブルを正しい形で照合すると、次の通信が届く。
- **結果の差を読む：**公式説明と内部記録、状態と症状、送信者と受信者を並べて比較する。
- **最後は別欄へ入力：**evidence_files で得た file_code は、DB ではなく証拠パッケージ入力欄へ送る。

SAFETY / READ ONLY

使用できるのは SELECT 文のみ。データ変更、複数 SQL、コメント構文、システム領域への参照は拒否される。

CIPHER CIPHER RELAY / READ-ONLY DOCUMENT SECTION 01

01 SQL・DB・操作ルール

最初に「どの表から、何を、どの条件で取り出すか」を決める。

1.1 SQL の基本形

BASIC SELECT

```
SELECT 見たい列
FROM 見たいテーブル
WHERE 条件;
```

構文	役割	使う場面
SELECT	表示する列を指定する	すべて見る場合は *
FROM	参照するテーブルを指定する	調査対象を決める
WHERE	必要な行だけに絞る	日付・部署・試験コード
JOIN / ON	共通する ID で表をつなぐ	人物名と login_id など
LIKE	文字を含む記録を探す	停止・報告書・対外発表
ORDER BY	結果を並び替える	ログやメールを時系列で読む

1.2 入力ルール

- 英数字・記号は半角で入力する。文字列はシングルクォートで囲む。
- 画面の実行ボタン、または Ctrl + Enter で実行する。
- SQL は 1 回につき 1 文。末尾のセミコロンは付けてもよい。
- INSERT / UPDATE / DELETE / DROP / ALTER / CREATE は使用できない。
- information_schema / pg_catalog などのシステム情報は参照できない。

CIPHER // METHOD

いきなり完成形を書かない。SELECT・FROM・WHERE・JOIN を 1 つずつ足す方法を確認してから調査へ進む。

1.3 SQL を 5 つの質問から組み立てる

SQL は暗記するより、「何を・どこから・どの条件で」を順番に決めると書きやすい。

NO ANSWER SQL
このページにある SQL は、記号部分を書き換えるための一般的な型である。

日本語の調査内容を SQL へ置き換える

考える質問	SQL の部分	決める内容
何を表示したいか	SELECT	列名。最初は * でもよい
どの表にあるか	FROM	テーブル名
どの行だけ必要か	WHERE	列名・比較記号・値
どの順で見たいか	ORDER BY	日時や ID などの列名
別の表も必要か	JOIN / ON	つなぐ表と共通する ID

BASIC FORM / ＜ ＞の中を置き換える

```
SELECT ＜表示したい列＞
FROM ＜調べるテーブル＞
WHERE ＜絞り込み条件＞
ORDER BY ＜並び替える列＞;
```

READ TOP TO BOTTOM
SQL は上から、表示する列 → 調べる表 → 絞り込み → 並び替えの順に読む。不要な句は省略できるが、句の順番は変えない。

1.4 1 つずつ動かしながら書く

最初から長い SQL を作らず、実行できた形へ 1 要素ずつ追加する。

最初から完成形を書かない

手順	書く内容	確認すること
1	表全体を少し確認する	列名と値の書かれ方を知る
2	必要な列だけに減らす	結果を読みやすくする
3	WHERE を 1 つ追加する	必要な行が残るか確認する
4	必要なら JOIN する	同じ種類の ID をつなぐ
5	ORDER BY を追加する	日時や ID の順で流れを読む

STEP 1 / 列名とデータの形を確認する型

```
SELECT *
FROM ＜テーブル名＞;
```

STEP 2 / 必要な列だけにする型

```
SELECT ＜列 1＞, ＜列 2＞
FROM ＜テーブル名＞;
```

値の書き方

値の種類	書き方	注意
文字	'＜文字＞'	シングルクォートで囲む
数値	＜数値＞	通常はクォートで囲まない
日付・日時	'YYYY-MM-DD HH:MM:SS'	文字と同じくクォートで囲む
空欄の値	IS NULL	= NULL とは書かない

CIPHER // CHECK

実行前に「列名」「テーブル名」「文字のクォート」「句の順番」の 4 点を見る。エラーになったら、最後に足した部分だけを外して確認する。

1.5 WHERE・LIKE・BETWEEN で絞る

基本形が動いたあとで、必要な条件を 1 つずつ追加する。

WHERE で行を絞り込む

書き方	意味	使い方
=	同じ値	ID・名前・状態の完全一致
<>	同じではない	特定の状態を除く
AND	すべての条件を満たす	日付と対象を両方指定する
OR	どれかの条件を満たす	複数の語句や状態を探す
LIKE	文字を含む	件名・本文・説明の部分一致
BETWEEN	範囲内	日時や数値の開始から終了

TEMPLATE / 2 つの条件を両方満たす

```
WHERE ＜列 A＞ = ＜値 A＞
  AND ＜列 B＞ = ＜値 B＞
```

TEMPLATE / 文字を含む行を探す

```
WHERE ＜文字列の列＞ LIKE '%＜探す文字＞%'
```

TEMPLATE / 期間で絞って古い順に並べる

```
WHERE ＜日時列＞ BETWEEN '＜開始日時＞' AND '＜終了日時＞'
ORDER BY ＜日時列＞ ASC;
```

CONDITION CHECK

AND と OR を一緒に使う場合は、先にまとめたい条件を丸括弧 ( ) で囲む。条件を足した直後に 0 件になった場合は、その条件だけを外して実際の表記を確認する。

1.6 JOIN で別の表をつなぐ

1 つの表だけでは不足する情報を、共通する ID で横につなげる。

TABLE CONNECTION

表 A の＜共通 ID＞ ⇄ 表 B の＜共通 ID＞

左右には、同じ種類の ID を指定する。employee_id と login_id のように役割が異なる ID を直接比較しない。

JOIN TEMPLATE / a・b はテーブルの別名

```
SELECT a.＜表 A の列＞, b.＜表 B の列＞
FROM ＜表 A＞ a
JOIN ＜表 B＞ b
  ON a.＜共通 ID＞ = b.＜共通 ID＞
WHERE ＜必要な条件＞;
```

JOIN を書く順番

順番	確認内容
1	2 つの表を別々に見て、列名と共通 ID を確認する
2	FROM に調査の中心となる表を書く
3	JOIN に追加する表を書く
4	ON の左右に同じ種類の ID を書く
5	結果が出てから SELECT 列と WHERE 条件を調整する

別名 a・b の読み方

- **a.列名**：表 A にある列を指す。
- **b.列名**：表 B にある列を指す。
- **別名を使う理由**：同じ列名が複数の表にある場合でも、どの表の列か明確にできる。

CIPHER // BUILD ORDER

SQL の基本順序は SELECT → FROM → JOIN / ON → WHERE → ORDER BY。JOIN で 0 件になったら、ON の左右が同じ種類の ID かを最初に確認する。

CIPHER CIPHER RELAY / READ-ONLY DOCUMENT SECTION 02

02 参照できるテーブル

9 つのテーブルを、人物側と AURORA 側の 2 系統として読む。

テーブル	主な内容	つながるキー
departments	部署名・説明	department_id
employees	社員名・部署・役職・状態	employee_id / department_id
internal_accounts	login_id・権限・アカウント状態	employee_id / login_id
access_logs	日時・操作・対象・結果	login_id / access_time
aurora_trials	試験コード・施設・目的・状態	trial_id / trial_code
incident_records	内部記録・公式説明・重大度	trial_id / trial_code
subjects	被験者状態・リスク・観察記録	trial_id / trial_code
internal_mails	停止要請・発表文修正・監査対応	sender_login_id / receiver_login_id
evidence_files	？？？	related_trial_id / trial_code

関係図

PERSON TRACE
departments → employees → internal_accounts → access_logs / internal_mails

AURORA TRACE
aurora_trials → incident_records / subjects / evidence_files

CIPHER // IMPORTANT
同じ名前が別のテーブルに直接入っているとは限らない。人物は employee_id と login_id、試験は trial_id と trial_code を介して照合する。

閲覧注意

以降の資料には、SQL を使った調査を進めるためのヒントが記載されている。
使用できる SQL 構文やテーブルの概要、情報を絞り込む方法、複数の記録を照合する考え方など、調査の手掛かりになる内容が含まれている。
閲覧することで調査を進めやすくなる一方、自分の力だけで試行錯誤しながら答えへ到達したい場合は、先に読むことで難易度や発見の楽しさが下がる可能性がある。
この資料は、必ず閲覧する必要はない。
調査に行き詰まったときだけ確認してもいいし、最初から参照しながら進めても構わない。
以降のヒントを閲覧するかどうかは、君自身で判断してほしい。

03 TRACE 00-02：部署・人物・login_id

CIPHER の最初の 3 段階。組織から人物を絞り、ログを読める状態にする。

00
TRACE

部署構成を確認する

AURORA、臨床安全、情報監査、広報、経営判断に関わる部署を把握する。

進行条件: departments だけを参照し、関連部署を結果に含める。

TRACE 00 / 調査ガイド

対象：departments

確認：部署 ID、部署名、説明

考え方：まず表の列を確認し、事件に関係しそうな部署を結果から見つける。

01
TRACE

関係部署の人物を確認する

部署名と社員名を同じ結果に表示し、事件に関係する人物を絞る。

進行条件: employees と departments を JOIN し、WHERE で関係部署を絞る。

TRACE 01 / 調査ガイド

対象：employees と departments

照合キー：department_id

考え方：人物名と部署名を同じ結果に表示し、関係部署に所属する人物だけへ絞り込む。

04 TRACE 02：人物と login_id

人物名を、アクセスログとメールで使用されるアカウント識別子へ変換する。完成形は示さないため、SECTION 01 の基本構文を組み合わせて入力する。

02
TRACE

人物名と login_id を照合する

アクセスログとメールで使われる login_id を、人物名と結び付ける。

進行条件: employees と internal_accounts を JOIN し、複数の関係人物を同時に確認する。

TRACE 02 / 調査ガイド

対象：employees と internal_accounts

照合キー：employee_id

考え方：関係人物の氏名と login_id、権限、アカウント状態を対応づける。

JOIN で混同しやすい ID

項目	役割	直接つながる先
employee_id	社員 1 人を識別する ID	employees ↔ internal_accounts
login_id	システム操作・メール送受信の ID	internal_accounts ↔ access_logs / internal_mails

CIPHER // TRACE 02

employee_id と login_id は同じ値ではない。人物を追うときは、まずアカウント表で対応関係を作ってからログへ進む。

05 TRACE 03：事故日のアクセスログ

公開 HP で得た事故日を、内部操作の時系列へ変換する。日時範囲と並び順を自分で組み立てる。

事故日で絞る

TRACE 03 / ACCESS LOGS

対象：access_logs

使う考え方：事故日の日時範囲で絞る / 時刻順に並べる

確認：誰が、いつ、何を対象に、どの操作を行ったか。

構文の手掛かり：WHERE・BETWEEN・ORDER BY

全件表示ではなく、公開 HP で確認した 2026 年 3 月 14 日に絞る。結果は時系列で並べ、AURORA-SOL9 がどの操作に現れるかを見る。

action	読み取れること
TRIAL_RECORD_OPENED	試験記録が参照された
SAFETY_ALERT_CREATED	安全警告が作成された
FILE_UPDATED	報告書・資料が更新された
ACCESS_LEVEL_CHANGED	閲覧権限が変更された
ACCOUNT_DISABLED	関係アカウントが停止された
LOG_DELETE_REQUESTED	ログ削除・整理が申請された

CIPHER // TRACE 03

重要なのは 1 件のログではなく順序。安全警告、記録更新、権限変更、削除申請が事故の前後でどう並ぶかを確認して。

06 TRACE 04-05：試験と事故記録

ログで見つけた AURORA-SOL9 を、試験情報と事故記録で裏付ける。照合キーを確認し、自分で SQL を構成する。

04
TRACE

AURORA-SOL9 の試験情報を確認する

試験の施設、目的、開始・終了時刻、状態を確認する。

進行条件: aurora_trials だけを参照し、trial_code で絞る。

TRACE 04 / 調査ガイド

対象：aurora_trials

絞り込みの手掛かり：ログで発見した試験コード

確認：施設、目的、開始・終了時刻、状態、補足記録。

05
TRACE

公式説明と内部記録を比較する

同じ試験について、内部で何が記録され、外部には何と説明されたかを同じ行に並べる。

進行条件: aurora_trials と incident_records を JOIN し、AURORA-SOL9 で絞る。

TRACE 05 / 調査ガイド

対象：aurora_trials と incident_records

照合キー：trial_id

確認：内部記録と公式説明を同じ結果に並べ、重大度・時刻・欠落した事実を比較する。

INTERPRETATION

official_summary は対外説明、internal_summary は内部記録。表現の差ではなく、消された事実・重大度・時刻の差を確認する。

07 TRACE 06：被験者記録

「設備だけの問題」という説明が、被験者データと一致するかを確認する。必要な列を選び、試験情報と照合する。

TRACE 06 / SUBJECT OBSERVATION

対象：subjects と aurora_trials

照合キー：trial_id

確認：試験前後の状態、リスク、同意状態、観察記録、記録の欠落。

絞り込みの手掛かり：対象となる試験コード

読み取る列

列	確認する内容
condition_before	試験前の状態。事故前との比較基準。
condition_after	反応遅延・意識混濁・神経応答の変化。
risk_level	安全上の重大度。
consent_status	同意確認が完全だったか。
observation_note	停止要請・記録欠落・原本資料への参照。
status	観察継続・記録不完全などの状態。

CIPHER // TRACE 06

試験の status だけを見て「成功」「終了」と判断しない。被験者側の状態、観察記録、欠落の有無を別に確認する。

STORY LINK / OBSERVATION

SOL9 の記録では「観測された対象」と「観測を行う側」の境界が曖昧になる。現時点では結論を出さず、後で見つかる記録の時刻とあなた自身の調査順序を比較して。

08 TRACE 07-08：停止要請と発表文修正

メールを人物名つきで読み、誰が止め、誰が判断し、何が外部から消えたかを追う。検索語は示すが、完成済み SQL は掲載しない。

login_id を人物名へ変換する JOIN

MAIL TRACE / 調査ガイド

対象：internal_mails・internal_accounts・employees

照合の流れ：メールの login_id → アカウント → 社員名

確認：送信者、受信者、件名、本文、送信時刻を同時に読む。

絞り込みの手掛かり：関連試験コードと、CIPHER が示す語句。

この基本形に語句条件を追加して、停止要請と発表文修正を別々に確認する。

TRACE	検索語の方向	確認する目的
07 停止要請	停止・中止・安全確認	瀬戸・真壁・久世の判断経路を追う
08 発表文修正	対外発表・報告書・設備トラブル	公式発表から除外された内容を追う

CIPHER // MAIL TRACE

件名だけで終わらせない。本文、送信者、受信者、送信時刻を同時に見て、停止要請がどこで判断へ変わったかを確認する。

09 TRACE 09-10：原本と証拠パッケージ

整えられた発表ではなく、AURORA-SOL9 に紐づく原本の file_code を特定する。対象表と照合キーから SQL を組み立てる。

TRACE 09 / EVIDENCE FILES

対象：evidence_files と aurora_trials

照合キー：related_trial_id と trial_id

確認：資料名、説明、作成時刻、所有者、状態、証拠レベル、file_code。

目的：公開用資料ではなく、最重要の原本に付与された識別子を特定する。

何を確認するか

項目	意味
file_name	原本・安全警告・公開文・ログ削除申請などの資料名。
description	資料が何を裏付けるか。
evidence_level	証拠としての重要度。
status	restricted / internal / pending などの状態。
file_code	証拠パッケージ入力欄へ送る識別子。

FINAL TRANSMISSION

最重要の原本に付与された file_code を、「証拠パッケージ ID」欄へ入力する。SQL 入力欄へ再入力しても送信は完了しない。

10 つまずき対処・早見表

進行しない場合は、結果だけでなく「SQL の形」と「現在の TRACE」を確認する。

よくあるつまずき

状況	確認すること
0 件になる	WHERE を一度外し、実際の表記・列名・日付形式を確認する。
SQL は成功したが CIPHER が進まない	現在の TRACE が求めるテーブル、JOIN、WHERE、日付・試験コード条件を満たしているか確認する。
列が存在しない	SELECT * で列名を確認する。古い手引書の列名と混同しない。
JOIN で 0 件になる	employee_id 同士、trial_id 同士、login_id 同士を対応させているか確認する。
結果が多すぎる	WHERE で事故日・部署・AURORA-SOL9 を絞り、必要な列だけ SELECT する。
操作が拒否される	SELECT 以外、複数文、コメント、禁止語、システム領域参照を取り除く。

構文の順序

```
SQL ORDER
SELECT 列
FROM テーブル 1
JOIN テーブル 2 ON 結合条件
WHERE 絞り込み条件
ORDER BY 並び替える列;
```

調査順序の最終確認

TRACE	調査対象	次へ進む鍵
00	departments	関係部署
01	employees + departments	人物と部署
02	employees + internal_accounts	人物と login_id
03	access_logs	2026-03-14 / AURORA-SOL9
04	aurora_trials	試験の実態
05	incident_records + aurora_trials	公式説明と内部記録
06	subjects + aurora_trials	被験者異常
07	mails + accounts + employees	停止要請
08	mails + accounts + employees	対外発表・報告書修正
09	evidence_files + aurora_trials	原本の file_code
10	証拠パッケージ入力欄	外部送信

CIPHER EXTERNAL OBSERVER / END NOTE | SECTION 11

11 CIPHER 観測メモ

調査を終える前に、記録の時刻と自分自身の行動を最後に照合する。

調査経路

PUBLIC	ORGANIZATION	TIME	SOL9	EVIDENCE	OUTSIDE
HP の断片	人物と login_id	事故日の順序	試験・事故・被験者	原本 file_code	外部公開

OBSERVATION CHECK

次の 3 点を確認する。

1. 自分が調査した順序と、SOL9 関連ログの記録順が一致していないか。
2. 自分が入力する前の時刻に、同じ検索行動を示す記録が存在しないか。
3. CIPHER の通信が「人物の判断」だけで説明できるか。

CIPHER // FINAL NOTE

この資料の目的は消された記録を、検証できる順序で読み直すことだ。
公式発表より内部記録を、内部記録より時刻を、時刻より前後関係を優先して。
そして、あなたが実行する前から同じ調査順序を記したログが存在していた場合、それを偶然として処理しないで。

END OF DISTRIBUTION COPY / CPH-DB-REF REV.05 / KEEP CHANNEL OPEN