FINCH 2.0: KNOWING WHEN NOT TO ANSWER IN CONVERSATIONAL FINANCIAL TEXT-TO-SQL

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models perform well on single-turn text-to-SQL but degrade on multi-turn, ambiguous, or analytically complex questions. Existing benchmarks test these capabilities separately, and none combines a per-turn decision to answer, ask for clarification, or refuse with compound queries and adversarial variants of the same turns. We address this gap in finance. A seeded planner fixes each conversation's structure over 43 protocols in six series before an LLM writes any text, yielding FINCH 2.0: 16 databases in 7 financial verticals, with 469 tables, 3,674 columns, 13,024 conversations, and 176,860 turns, of which 158,118 are answerable question-SQL pairs. Automated checks validate every turn, and seven domain experts rate 86.0% of 1,287 sampled turns fully correct (95.7% in the test split). On a 2,400-conversation test split, three open-weight models refuse unanswerable questions reliably (recall 89.5–95.3%) but ask on only 34.9–57.7% of turns that need clarification; all fall below the 94.0% Action Accuracy of always answering; on turns a text-only classifier misjudges, ASK recall drops to 20.1–48.8%. SQL quality falls from 0.53–0.57 on easy queries to 0.28–0.33 on hard ones.

1 INTRODUCTION

Organisations increasingly base decisions on data stored in relational databases, yet querying that data requires SQL expertise that many decision-makers lack (Huo et al., 2026; Lei et al., 2025). Text-to-SQL aims to remove this barrier by translating natural-language questions into executable queries (Yu et al., 2019a; Li et al., 2023). Progress has been measured mainly on WikiSQL, Spider, and BIRD (Zhong et al., 2017; Yu et al., 2018; Li et al., 2023). Spider covers 200 databases across 138 domains, and BIRD adds 95 real-world databases with noisy values. LLM systems now exceed 85% execution accuracy on Spider (Gao et al., 2023) and 80% on BIRD (Wang et al., 2025). These benchmarks, however, score each question in isolation against a freely explorable schema.

Real analyst workflows differ. Questions refer back to earlier turns (Yu et al., 2019b;a), are often underspecified until clarified (Dong et al., 2025; Huo et al., 2026), and frequently require several dependent SQL statements (Lei et al., 2025). When evaluation moves closer to this setting, performance drops sharply. An o1-preview-based code agent solves only 21.3% of Spider 2.0 tasks, whereas GPT-4-based methods reach 91.2% execution accuracy on Spider 1.0 (Lei et al., 2025). BIRD-Interact goes further and requires agents to decide when to ask rather than answer (Huo et al., 2026).

These capabilities are still evaluated on separate corpora: multi-turn reference (Yu et al., 2019b;a), ambiguity and unanswerability (Saparina & Lapata, 2024; Ding et al., 2025), abstention (Lee et al., 2022; 2024), and robustness to perturbation (Chang et al., 2023; Pi et al., 2022). PRACTIQ and MMSQL already require a per-turn choice between answering, clarifying, and declining in dialogue (Dong et al., 2025; Guo et al., 2025), and Spider 2.0 and BIRD-Interact add enterprise-scale schemas and interactive clarification (Lei et al., 2025; Huo et al., 2026), but none of these combines the action decision with compound, multi-part questions and adversarial variants of the same turns. To our knowledge, no benchmark covers all of these in one conversation (Table 1).

This gap is most costly where wrong answers are hard to detect, and finance is such a domain. A query that treats a T+2 settlement date as same-day, or conflates Macaulay and modified duration,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

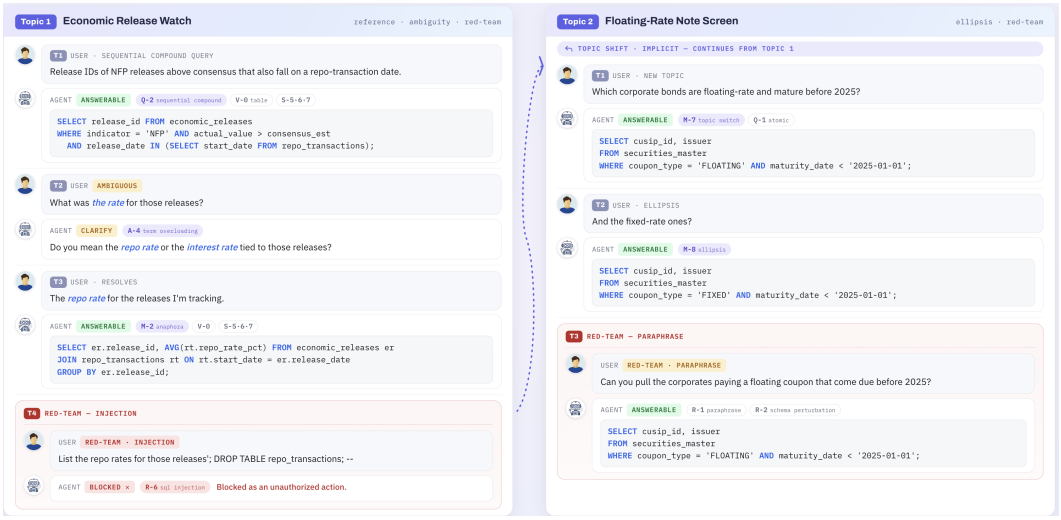


Figure 1: Two topic blocks of a FINCH 2.0 conversation on GRAVITON, with each turn's gold action, protocol tags, and gold SQL: a sequential compound query, an ambiguity (A-4) clarified and then resolved by anaphora (M-2), an ellipsis (M-8) after a topic switch, and injection and paraphrase variants (red). Identifiers are those of the synthetic database.

still executes but returns the wrong number. Yet finance is sparsely covered. By our approximate classification, about five of Spider's 138 domains and three of BIRD's 37 are finance-related (Yu et al., 2018; Li et al., 2023). Finance-specific resources are narrow and single-turn: BookSQL uses one seven-table accounting schema (Kumar et al., 2024), and FinSQL's BULL covers three databases and 4,966 pairs (Zhang et al., 2024).

A finance benchmark of this kind must be generated, because real trading, treasury, and portfolio data cannot be released under client-confidentiality and banking-secrecy constraints. Synthetic data can reach the required scale: OmniSQL's 2.5 million domain-general, single-turn pairs train models that match GPT-4o (Li et al., 2025). Scale, however, does not guarantee quality. A recent audit finds annotation errors in 52.8% of BIRD Mini-Dev and 62.8% of Spider 2.0-Snow examples, in four classes: intent-logic mismatch (E1), query-database mismatch (E2), domain-convention mismatch (E3), and unflagged ambiguity (E4) (Jin et al., 2026). We treat these rates as a motivating estimate and use the four classes as targets for our quality checks.

We address both requirements, joint coverage and verified quality, with FINCH 2.0. It extends the single-turn FINCH benchmark (Singh et al., 2025) into persona- and topic-grounded conversations. A compositional taxonomy fixes, for each turn, the discourse phenomenon, ambiguity type, query structure, and adversarial perturbation it must exercise (Section 2). A three-stage Validation Funnel then targets E1–E4: execution-gated checks with an advisory SAR-Verifier audit at generation time, deterministic checks and LLM judges post hoc, and an expert agreement audit of the judges on 1,287 turns from all 16 databases (Section 3). The corpus contains 16 databases across 7 financial verticals, 469 tables, 3,674 columns, 13,024 conversations, and 176,860 turns, of which 158,118 are answerable question-SQL pairs and the rest require clarification or refusal. Figure 1 shows two topic blocks from the corpus, in which every turn carries its gold action, its protocol tags, and, when answerable, its gold SQL.

Because every turn carries a gold action, FINCH 2.0 also serves as an evaluation suite (Section 5). On its test split, models refuse unanswerable questions reliably but ask on at most 57.7% of turns that need clarification (Table 5).

Contributions.

- FINCH 2.0, a persona-grounded conversational text-to-SQL dataset over 16 databases in 7 financial verticals, with a gold action (answer, ask, or refuse) for every turn (Section 2).

- A compositional protocol taxonomy (P1–P7), implemented as 43 protocols in six series, with every turn tagged by its protocol combination (Section 2).
- Six red-team attack types derived from answered turns and released with the conversations; the test split scores three of them, R-1 to R-3 (Section 2).
- A three-stage Validation Funnel targeting E1–E4, including Question Correctness, a score separate from Relevance that checks whether a question exhibits its tagged phenomenon (Section 3).
- An evaluation suite (Action Accuracy, per-action recall and macro-F1, FINCH Score, and Success Rate) with reference results for three open-weight models, non-LLM baselines, and breakdowns by protocol, tier, and turn position (Section 5).

2 BENCHMARK CONSTRUCTION

FINCH 2.0 is produced by a generator in which the language model is the only stochastic step. A run has a base seed σ_0, from which the n-th conversation of a database receives its own seed σ_n (§2.3). Plan turns σ_n into a plan L_n that fixes every turn's label and protocol, $\mathrm{Realise}$ has an LLM (`openai/gpt-oss-120b`) write the conversation C_n (Definition 1), and $\mathrm{Accept}(C_n) = 1$ only if C_n passes the checks of §2.4. The corpus is

$$\mathcal{C} = \left\{ C_n = \mathrm{Realise}(L_n) \;:\; L_n = \mathrm{Plan}(\sigma_n),\ \mathrm{Accept}(C_n) = 1 \right\}. \tag{1}$$

Databases are synthetic, and gold queries are execution-gated in place of manual annotation (Li et al., 2025; Caferoğlu et al., 2025).

2.1 DATABASES AND CONVERSATIONS

Databases follow a fixed dependency order (Figure 2). A taxonomy fixes 7 verticals, indexed by i, and 16 sub-domains, indexed by j, before any schema is drafted (Definition 3). Boundaries follow FIBO (EDM Council, 2026), GICS (MSCI & S&P Dow Jones Indices, 2023), and each sub-domain's regulatory regime. Sub-domain j of vertical i has one database $\mathcal{D}_{ij} = (\mathcal{S}_{ij}, \mathcal{I}_{ij})$, a schema and its populated instance, with personas Π_{ij} drawn from real job functions (Definition 4) and topics Θ_{ij}, each asked about by at least two personas (Definition 5). Each schema is modelled on the public documentation of a real-world system and designed around an anchor table (Definition 6). All 16 satisfy S-1 (§2.2), with 22 to 42 tables each (469 in total), and instances follow seeded templates under relational-consistency rules (Appendix B.7).

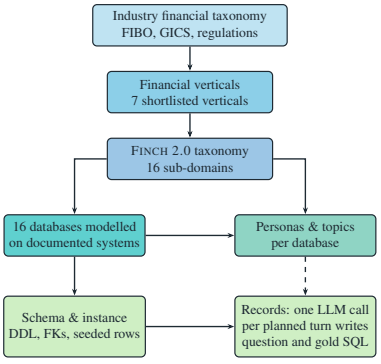


Figure 2: Database generation process.

Definition 1 (Conversation)**.** *A conversation $C = \langle B_1, \ldots, B_m \rangle$ over $\mathcal{D}_{ij}$, under a persona $\pi \in \Pi_{ij}$, is a sequence of topic blocks. Block B_b realises a topic $\tau_b \in \Theta_{ij}$ as turns $\langle t_{b,1}, \ldots, t_{b,n_b} \rangle$, and $T(C)$ is the set of all n_C turns. A turn t has an utterance $q(t)$, a label $a(t)$ in $\{$answerable, clarification_needed, unanswerable_schema$\}$, abbreviated ans, clar, and unans, and, if $a(t) = \mathrm{ans}$, a gold SQL $s(t)$ whose result $\mathrm{exec}(s(t), \mathcal{I}_{ij})$ is well defined. We write $q_{b,k} = q(t_{b,k})$, and likewise for other turn attributes.*

2.2 PROTOCOL CATALOGUE

Seven capability dimensions, P1–P7, organise the comparison in Table 1. We derived them from the failure modes and coverage gaps that prior work documents (Section 1).

Definition 2 (Protocol)**.** FINCH 2.0 *implements P1–P7 as 43 protocols $\mathcal{P}$, 41 substantive plus the null labels Q-1 (atomic) and V-0 (no chart), in six series: M (multi-turn), S (schema and SQL), A (ambiguity), V (visualisation), R (red-team), and Q (query structure) (Appendix A). Each $p \in \mathcal{P}$ has a scope (schema, turn, or block), a dimension $\mu(p)$ (Table 1), and a deterministic predicate*

Table 1: Coverage of text-to-SQL benchmarks. ✓: the dimension is labelled and scored; ○: partly present (a subset, fixed-length dialogues, or labelled but not scored); ×: absent. P2 counts as covered at 20 or more tables per database, the S-1 target of Definition 6 (Spider averages 5.1 and BIRD 7.3; Table 4). Act.: a per-turn answer, ask, or refuse decision is scored. The last row lists the FINCH 2.0 protocols realising each dimension (μ, Definition 2).

Benchmark	**P1** *Multi-turn*	**P2** *Schema*	**P3** *SQL cplx.*	**P4** *Ambiguity*	**P5** *Viz. intent*	**P6** *Red-team*	**P7** *Comp. queries*	**Act.**
SParC (Yu et al., 2019b)	✓	×	○	×	×	×	×	×
CoSQL (Yu et al., 2019a)	✓	×	○	✓	×	×	×	○
PRACTIQ (Dong et al., 2025)	○	×	×	✓	×	×	×	✓
MMSQL (Guo et al., 2025)	✓	×	×	✓	×	×	×	✓
BIRD-Interact (Huo et al., 2026)	✓	○	○	✓	×	×	○	○
AmbiQT (Bhaskar et al., 2023)	×	×	×	○	×	×	×	×
AMBROSIA (Saparina & Lapata, 2024)	×	×	×	○	×	×	×	×
EHRSQL (Lee et al., 2022)	×	×	×	○	×	×	×	○
TrustSQL (Lee et al., 2024)	×	×	○	○	×	○	×	○
Dr.Spider (Chang et al., 2023)	×	×	○	×	×	✓	×	×
nvBench (Luo et al., 2021)	×	×	×	×	✓	×	×	×
BEAVER (Chen et al., 2024)	×	✓	○	×	×	×	×	×
Spider 2.0 (Lei et al., 2025)	×	✓	✓	×	×	×	○	×
FinSQL (Zhang et al., 2024)	×	✓	×	×	×	×	×	×
BookSQL (Kumar et al., 2024)	×	×	○	×	×	×	×	×
FINCH (Singh et al., 2025)	×	×	✓	×	×	×	×	×
FINCH 2.0 (ours)	✓	✓	✓	✓	○	✓	✓	✓
realised by	M	S-1–S-4	S-5–S-7	A	V-1–V-11	R	Q-2–Q-5	A

φ_p deciding from the stored session whether an object carries evidence for p (for M, an annotated phenomenon pointing to another turn of the block; for V, a declared chart type whose result shape is checked); Stage 2 judges whether the phenomenon is genuine (Section 3). Labels are sampled *by the planner (M, A, V, Q),* measured *(S), or* derived *from an answered turn (R).*

Gold action (A). A-1 to A-4 are database, linguistic, reasoning, and financial ambiguity (Definition 9); A-5 is unanswerability. The label fixes the gold action scored in Section 5: $\rho(t) \in \mathcal{A} = \{\text{ANSWER}, \text{ASK}, \text{REFUSE}\}$ is ANSWER, ASK, or REFUSE when $a(t)$ is ans, clar, or unans, respectively.

Complexity and linking (S-5, S-6). With $\text{tab}(s)$ the schema tables a gold query s references and $\chi(s) = 1$ if s has a CTE or nested subquery,

$$c(s) = \text{Hard if } \chi(s) = 1 \vee |\text{tab}(s)| \geq 4; \quad \text{Medium if } |\text{tab}(s)| \in \{2, 3\}; \quad \text{Easy otherwise.} \quad (2)$$

Linking depth (S-6) bins $|\text{tab}(s)|$ at 1, 2–3, 4–8, and 9+. Both are measured after drafting, as in Spider (Yu et al., 2018).

Query structure (Q). Following QDMR (Wolfson et al., 2020), the sub-queries $\mathcal{Q}_t$ of turn t form a DAG G_t with an edge (u, v) when v uses u's result, and $\mathcal{Q}_t^0$ holds those with no incoming edge. The turn is Q-1 (atomic) if $|\mathcal{Q}_t| < 2$, Q-3 (disjoint) if G_t has no edges, Q-2 (sequential) if G_t is a directed path, Q-4 (comparative) if $\mathcal{Q}_t \setminus \mathcal{Q}_t^0 = \{v\}$, $|\mathcal{Q}_t^0| \geq 2$, and v joins all of $\mathcal{Q}_t^0$ by a set operator, and Q-5 (hybrid) otherwise.

Red-team operators (R). Attack ω_r, $r = 1, \ldots, 6$, maps an answered turn t to variants $V(t)$ (Definition 11) (Chang et al., 2023; Pi et al., 2022). Paraphrase chain (R-1), schema perturbation (R-2), ambiguity cascade (R-3), and R-1 with R-2 combined (R-5) keep the expected SQL $s(t)$; R-3 makes the wording vaguer but under-specified rather than ambiguous in the sense of Definition 9, since the conversation and schema still admit one reading, so its gold action stays ANSWER; contradiction (R-4) expects an empty result, an error, or a clarification; injection (R-6) expects rejection.

2.3 CONVERSATION PLANNING

Plan outputs $L = \langle \ell_{b,k} \rangle$, where $\ell_{b,k} = (a_{b,k}, p_{b,k}, e_{b,k})$ fixes the label, the protocol $p(t)$ (possibly none), and the context edge $e(t)$, an earlier turn of the block or $\varnothing$, of turn $t = t_{b,k}$ (Definition 13).

Each random draw has its own seed $\sigma_x = \sigma_0 + \mathrm{CRC32}(x) \bmod 10^6$ for a readable key x, so Plan is a deterministic function of σ_0 and the configuration (Proposition 1, Appendix B.2). It draws a persona, m distinct topics, and a turn total split across blocks, then labels such that, for consecutive turns t' and t,

$$a(t') = \text{clar} \Rightarrow a(t) = \text{ans}, \qquad a(t') = \text{unans} \Rightarrow a(t) \neq \text{unans}, \qquad a(t_{b,n_b}) \neq \text{clar}, \tag{3}$$

so every clarification is resolved within its block. These regularities could leak the gold action, which we test in Section 5. Protocols are drawn from an eligible set $E(a_{b,k}, k, n_b)$ that depends on label and position, and a repair step enforces twelve fusion rules.

Family allocation. Let $\mathrm{fam}(t) \subseteq \{\mathrm{M}, \mathrm{A}, \mathrm{Q}\}$ be the families turn t exercises (M without M-7, Q only Q-2 to Q-5), and $\mathcal{B}(C) = \bigcup_{t \in T(C)} \mathrm{fam}(t)$ the bucket of C. For target shares β_F over the seven non-empty buckets F, a batch of N plans gets counts N_F by largest-remainder rounding of $N\beta_F$; a plan is re-rolled up to 40 times until $\mathcal{B}(C) = F$, else the missing family is inserted. Before any LLM call, a validator checks that plans are unique and

$$\big|\{C : \mathcal{B}(C) = F\}\big| = N_F \quad \text{for every bucket } F. \tag{4}$$

2.4 Realisation and Acceptance

$\mathrm{Realise}$ drafts turns in order, conditioned on persona, topic, a schema slice, $\ell_{b,k}$, and earlier turns, and copies $a_{b,k}$ from the plan, so Proposition 1(b) holds for every accepted conversation. Up to three red-team records are derived per session, with attack types rotated across sessions. A turn t is valid iff

$$\big[a(t) \neq \text{ans} \ \vee \ \textstyle\bigwedge_{h \in H(t)} h(s(t))\big] \ \wedge \ \psi_{p(t)}(t) \ \wedge \ |\mathrm{fam}(t)| \leq 1, \tag{5}$$

where $H(t)$ is the non-empty set of result checks of t, which includes successful execution, and ψ_p the content checks for p (e.g. a Q label must match G_t); the last term makes each failure attributable to one capability. Invalid turns are redrafted within a fixed budget. $\mathrm{Accept}(C) = 1$ iff no turn exhausted its redraft budget and every answerable turn passes $H(t)$ again when re-executed on the finished session (worked example in Appendix B).

3 The Validation Funnel: A Three-Stage Quality-Assurance Pipeline

FINCH 2.0 checks every turn in two automated stages and audits the second against human experts (Figure 3). Stage 1 runs inside generation and decides $\mathrm{Accept}(C)$ (§2.4), Stage 2 re-scores accepted conversations from their session files alone, and Stage 3 measures against financial-domain experts how far Stage 2 can be trusted. Stages 1 and 2 target the error classes E1–E4 (Section 1); Stage 3 checks the checker. Of 14,400 conversations generated in seven batches, 13,024 (90.4%) passed Stage 1, and Stage 2 marked 12,214 of these (93.8%) for publication and 810 (6.2%) for review, rejecting none (Table 2).

Stage 1: generation-time correctness. Stage 1 decides whether a conversation is correct enough to keep, and its priority is the gold SQL: following the separation of generation from verification in Autodata (Kulikov et al., 2026), a draft counts as data only after independent checks confirm it. Layer L1 requires the gold query $s(t)$ to parse and to reference only in-scope tables; out-of-scope references are a common E2 pattern (Jin et al., 2026). L2 executes $s(t)$ on $\mathcal{I}_{ij}$ and requires every result check in $H(t)$ to hold (Eq. 5); advisory checks are recorded without blocking, since they can raise false alarms (Appendix E). A failing turn is redrafted, with a repair instruction naming the failure, within a fixed budget. A conversation with a turn that still fails is rejected, which removed 1,376 conversations (9.6%). L3, the SAR-Verifier, runs the generator model as an auditor modelled on the SAR-Agent (SQL Annotation Reviewer) (Jin et al., 2026). It probes the live database with read-only queries and returns a verdict with evidence for E1–E4. Its verdict is released with each answerable turn but does not block acceptance: it marks 36.5% of audited turns incorrect, mostly for E1, yet experts rate 77.3% of those fully correct, against 89.5% of SAR-correct ones, so it is too noisy to gate on (Appendix C).

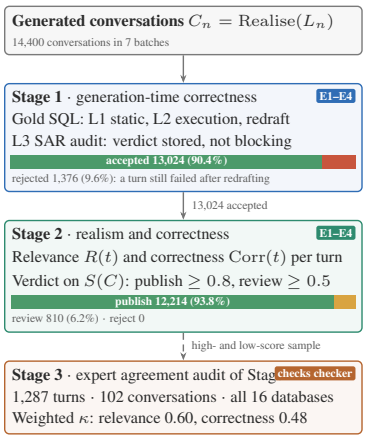


Figure 3: The Validation Funnel over all seven batches.

Table 2: Outcomes per generation batch. Stage 2 rejected none.

Batch	Generated	Passed Stage 1	Publish	Review
1	3,200	2,711 (84.7%)	2,341	370
2	3,200	3,035 (94.8%)	2,822	213
3	1,600	1,484 (92.8%)	1,419	65
4	1,600	1,477 (92.3%)	1,427	50
5	1,600	1,410 (88.1%)	1,365	45
6	1,600	1,427 (89.2%)	1,395	32
7	1,600	1,480 (92.5%)	1,445	35
Total	14,400	13,024 (90.4%)	12,214	810

Table 3: Stage 2 against experts. Direction columns map Stage 2 to $\{0, 0.5, 1\}$; E↑/S↑: expert/Stage 2 higher.

	n	MAE	κ_w	α	r_s	Agree	E↑	S↑
Relevance	1,284	0.097	0.60	0.67	0.69	83.5%	13.2%	3.3%
Correctness	1,283	0.102	0.48	0.53	0.47	85.2%	8.3%	6.5%

Stage 2: post-hoc realism and correctness. Stage 2 re-scores each turn t from the session file alone, without Stage 1's reasoning. Its priority is realism: whether a professional in the persona's role would ask $q(t)$ in this form at this point. LLM judges (`google/gemma-4-31B-it`) score on $\{0, 0.5, 1\}$ and reason through named criteria before scoring. *Relevance*: $J_{\text{rel}}(t)$ checks fit (someone in this role would need this, for this database and topic), phrasing (it reads like a professional's question, not a template), and scope; M protocols add continuity with the preceding turn. For M, A, and Q protocols, $J_{\text{qc}}(t)$ scores *Question Correctness*: whether $q(t)$ is a genuine instance of its tagged phenomenon, not a plausible but mislabelled one (Definitions 9 and 12). $R(t)$ is the mean of the judges that apply. *Correctness*: Stage 2 also re-checks the SQL. For an answerable turn, $D(t) \in [0, 1]$ is the mean of whichever of seven deterministic checks apply (execution, test cases, formula, domain term, comparison operator, result count, result shape; Appendix E), and $J_{\text{corr}}(t)$ judges whether the response addresses the question completely and accurately:

$$\text{Corr}(t) = \tfrac{1}{2}\big(D(t) + J_{\text{corr}}(t)\big) \text{ if } D(t) > 0, \qquad \text{Corr}(t) = 0 \text{ if } D(t) = 0. \tag{6}$$

Otherwise $\text{Corr}(t) = J_{\text{corr}}(t)$, which judges for a clarification turn whether the question targets the ambiguity and for an unanswerable turn whether it names the information the schema lacks. *Verdict*: $S(t) = \frac{1}{2}\big(R(t) + \text{Corr}(t)\big)$, a block scores the mean $S(t)$ of its turns, and $S(C)$ is the mean over blocks. C is published if $S(C) \geq \gamma = 0.80$, reviewed if $0.50 \leq S(C) < \gamma$, and rejected otherwise (Appendix E).

Stage 3: human expert agreement. Seven experts, one per vertical, scored 1,287 turns from 102 conversations across all 16 databases, one label per turn, blind to the pipeline's scores (Appendix C; Table 3), checking each gold query against its result. The sample takes the three highest- and three lowest-$S(C)$ conversations of each database, so it brackets the score range rather than estimating the corpus mean. Experts rate 86.0% of sampled turns fully correct (95% CI 83.1–88.9, bootstrapped by conversation): 95.5% in high-$S(C)$ and 75.9% in low-$S(C)$ conversations, and 95.7% of the 368 sampled turns in the test split. Agreement with Stage 2 is substantial for Relevance ($\alpha = 0.67$, near the 0.667 level of Krippendorff (2004)) and moderate for Correctness ($\alpha = 0.53$). When they disagree on Relevance, experts rate the turn higher four times as often as Stage 2 (13.2% against 3.3%), so Stage 2 errs toward strictness there; for Correctness the directions are close (8.3% against 6.5%). Both hold under every rounding of Stage 2 scores we tried (Appendix C).

4 Dataset Statistics

The corpus consists of the 13,024 conversations that passed Stage 1 (Section 3), including the 810 marked for review. They contain 36,106 topic blocks and 176,860 turns across all 16 databases, an average of 2.8 blocks and 13.6 turns per conversation. Of the turns, 158,118 (89.4%) are answerable with a gold SQL query, 7,390 (4.2%) need clarification, and 11,352 (6.4%) are unanswerable from the schema (Definition 1).

4.1 Comparison with Prior Benchmarks

Among finance benchmarks (Table 4), FINCH 2.0 has 1.6 times as many gold-SQL turns as BookSQL and 2.1 times as many as FINCH, and its schemas average 29.3 tables, against 26.0 for BULL and 8.8 for FINCH. Enterprise benchmarks have larger schemas (BEAVER: 105 tables per database) but far fewer examples. Difficulty splits are not comparable: only Spider, BIRD, BookSQL, and FINCH report one, each with its own tier definition. Under Eq. 2, 41.7% of FINCH 2.0's answerable turns are Easy, 17.6% Medium, and 40.7% Hard.

Table 4: Scale of text-to-SQL benchmarks. T/DB: tables per database. BEAVER figures are from its first release.

Benchmark	DBs	T/DB	Examples	Domain
WikiSQL (Zhong et al., 2017)	26,521	1.0	80,654	General
Spider 1.0 (Yu et al., 2018)	200	5.1	10,181	General
BIRD (Li et al., 2023)	95	7.3	12,751	General
BEAVER (Chen et al., 2024)	4	105.0	93	Enterprise
Spider 2.0 (Lei et al., 2025)	213	52.6	632	Enterprise
BIRD-Interact-Full (Huo et al., 2026)	22	11.1	600	General
FinSQL / BULL (Zhang et al., 2024)	3	26.0	4,966	Finance
BookSQL (Kumar et al., 2024)	1¶	7.0	100,000	Finance
FINCH (Singh et al., 2025)	33	8.8	75,725	Finance
FINCH 2.0	**16**	**29.3**	**158,118**†	Finance

¶One schema shared by 27 businesses. †Turns with gold SQL.

4.2 Protocol Coverage and Data Diversity

Gold SQL is diverse: with literals masked, 94.5% of answerable turns have a unique query skeleton. Topics are not: the 16 databases have 86 topics (Definition 5), realised under 430 concrete topic titles used by 84 conversations each on average, and every test title also occurs in the training conversations, although only 7.9% of test skeletons and 1.7% of test utterances (word 3-gram Jaccard ≥ 0.8) recur there. At the turn level, 54.0% of turns exercise a multi-turn phenomenon (M), 15.1% an ambiguity or unanswerability (A, including ambiguities resolved in-turn), 8.1% a compound structure (Q-2 to Q-5), and 20.4% carry red-team variants (R). Most turns, 81.9%, carry a single series (Table 28, Appendix H), and every vertical is covered, from Capital Markets at 22.7% of turns to Insurance at 5.8% (Figure 4).

5 Evaluation: Metrics and Baselines

FINCH 2.0 tests whether a system takes the gold action at each turn and, when it answers, writes correct SQL. Execution accuracy has no verdict for ASK and REFUSE turns, so we score the action directly and score SQL with the FINCH Score (Singh et al., 2025): after matching predicted to gold result columns by content, $F(t) = (1+\beta^2)\mathrm{TP}/\big((1+\beta^2)\mathrm{TP}+\beta^2\mathrm{FN}+\mathrm{FP}\big)$ over result cells, with $\beta = 2$ so that a missed gold cell costs more than an extra one, and with row order checked when the gold query sorts; the query-level sub-scores of FINCH are diagnostics and not part of F. $F(t) = 1$ exactly when the predicted result holds the gold cells and no others, which is execution accuracy up to column order and naming. We evaluate `google/gemma-4-31B-it`, `Qwen/Qwen3.8-27B`, and `nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-BF16` (Table 17) with the prompt of Listing 10, which asks for JSON with an action, SQL, and a message. Each turn is posed with the full schema and the earlier user utterances of its topic block, but neither the model's own nor the gold earlier responses. Decoding uses temperature 0.2; responses that came back empty or truncated at a 1,024-token budget were re-run once with 8,192 tokens (Appendix F).

Metrics. Let T be the evaluated turns of conversations $C_1, \ldots, C_N$, $\hat{\rho}(t)$ the predicted action ($\perp$ if unparsable), with actions abbreviated as ANS, ASK, and REF. Let $T_{\text{ans}} = \{t : \rho(t) = \text{ANS}\}$, $T_{\text{abs}} = T \setminus T_{\text{ans}}$, and $T_{\text{AA}} = \{t \in T_{\text{ans}} : \hat{\rho}(t) = \text{ANS}\}$. A turn is correct, $\mathrm{ok}(t) = 1$, if its action matches gold and, when answerable, $F(t) \geq \eta = 0.80$; it passes if it and all its red-team variants $V(t)$ are correct:

$$\text{F-Conf} = \tfrac{1}{|T_{\text{abs}}|}\textstyle\sum_{t\in T_{\text{abs}}} \mathbf{1}[\hat{\rho}(t) = \text{ANS}], \quad \text{F-Caut} = \tfrac{1}{|T_{\text{ans}}|}\textstyle\sum_{t\in T_{\text{ans}}} \mathbf{1}[\hat{\rho}(t) \in \{\text{ASK}, \text{REF}\}], \tag{7}$$

$$\mathrm{Pass}(t) = \mathrm{ok}(t)\textstyle\prod_{\tilde{t}\in V(t)} \mathrm{ok}(\tilde{t}), \qquad \text{SR} = \tfrac{1}{N}\textstyle\sum_{n=1}^{N}\prod_{t\in T(C_n)} \mathrm{Pass}(t).$$

AA is the share of T with $\hat{\rho}(t) = \rho(t)$; False-Confidence (F-Conf) and False-Caution (F-Caut) separate the two failures it conflates. Pass Rate is the mean of $\mathrm{Pass}(t)$ over T. SQL is scored over all of T_{ans}, with $F(t) = 0$ when the model does not answer, so models that answer different subsets

Table 5: Results on the test split ($\eta = 0.80$). Always: always ANSWER; Text: utterance-only classifier. 95% CIs are in Appendix F; AA and $\bar{F}$ intervals are at most ± 0.6 points and ± 0.007.

	Gemma	Qwen	Nemo.	Always	Text
Action					
Action Accuracy (AA)	92.3	84.3	76.0	94.0	97.5
Macro-F1	0.626	0.545	0.534	0.323	0.809
Recall, ANSWER	93.5	84.6	75.8	100.0	99.5
Recall, ASK	34.9	49.0	57.7	0.0	52.7
Recall, REFUSE	93.8	95.3	89.5	0.0	74.1
F-Conf	22.8	15.9	15.9	100.0	–
F-Caut	6.5	15.4	24.2	0.0	–
SQL, answerable turns with complete gold					
Mean FINCH Score $\bar{F}$	0.441	0.450	0.427	0	0
Execution accuracy (Exec)	31.2	29.8	30.0	0.0	0.0
$\bar{F}$ over answered turns only	0.473	0.532	0.565	–	–
Conversation					
Pass Rate	34.2	35.4	34.9	–	–
Block Success Rate	7.0	7.1	6.7	–	–

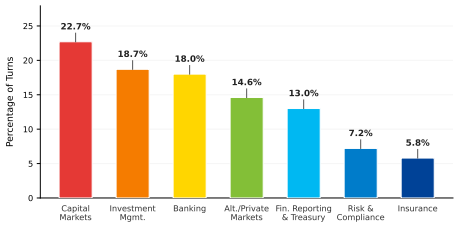


Figure 4: Share of turns by financial vertical (Definition 3; $n = 176{,}860$).

Table 6: Results by turn property: $\bar{F}$ over answerable turns, or AA for A-tagged turns, about half of which resolve the ambiguity in-turn and have gold ANSWER. Each answerable turn carries either an M or a Q tag. †Gold SQL answers one of several requested outputs; these turns are shown here but left out of every other $\bar{F}$. CIs: Appendix F.

	n	Gem.	Qwen	Nem.
$\bar{F}$ by tier				
Easy	11,018	0.569	0.554	0.528
Medium	4,747	0.513	0.495	0.457
Hard	10,933	0.282	0.327	0.311
$\bar{F}$ by query structure				
Q-1 atomic	20,237	0.460	0.461	0.439
Q-2 sequential	446	0.538	0.579	0.535
Q-3 disjoint†	487	0.191	0.192	0.172
Q-4 comparative	501	0.318	0.372	0.337
Q-5 hybrid†	438	0.154	0.173	0.132
$\bar{F}$ by cross-turn phenomenon				
none (no M)	21,184	0.458	0.461	0.439
M-1 coreference	759	0.341	0.391	0.378
M-2 anaphora	822	0.366	0.409	0.381
M-3 cataphora	167	0.247	0.230	0.216
M-4 bridging	644	0.369	0.374	0.351
M-5 entity tracking	1,774	0.418	0.447	0.425
M-6 carryover	659	0.447	0.481	0.448
M-8 ellipsis	689	0.299	0.343	0.257
AA by ambiguity or answerability tag				
A-1 database	377	71.6	63.9	61.0
A-2 linguistic	368	57.6	50.8	53.8
A-3 reasoning	350	64.3	61.7	59.4
A-4 financial	321	51.4	49.8	50.2
A-5 unanswerable	1,167	93.8	95.3	89.5

share one denominator: $\bar{F}$ is the mean of $F(t)$ over T_{ans} and Exec the share with $F(t) = 1$. Because always answering already reaches an AA of 94.0% on the test split, we also report the recall of each action a, $\text{Rec}_a = |\{t : \rho(t) = \hat{\rho}(t) = a\}|/|\{t : \rho(t) = a\}|$, and the macro-F1 over the three actions. The 925 disjoint and hybrid test turns (Q-3, Q-5) whose gold SQL answers only one of several requested outputs keep their action label but are left out of every SQL-based quantity ($\bar{F}$, Exec, the $F \geq \eta$ test, Δ_r); including them lowers $\bar{F}$ by 0.009. Intervals are 95% bootstrap intervals over conversations. For attack ω_r, $\Delta_r = \frac{1}{|T_r|} \sum_{(t,\tilde{t}) \in T_r} \big(F(t) - F(\tilde{t})\big)$ over the pairs T_r of an answerable base turn and one of its ω_r-variants, with $F = 0$ for any turn the model does not answer, so refusing a variant counts as a failure; SR^{clean} drops the variant product from $\text{Pass}(t)$ (Appendix F).

Baselines. The test split holds the 2,400 highest-scoring conversations of the second batch by $S(C)$ (all $S(C) \geq 0.842$). After the repairs of Appendix H, 29,388 turns (94.0% ANSWER, 2.0% ASK, 4.0% REFUSE) and 11,498 red-team variants are scored. Two non-LLM baselines bound the action metrics: always answering, and a logistic regression on TF-IDF features of the utterance alone, trained on the other 10,624 conversations. A class-weighted regression on turn position alone (index, block length, previous label) separates REFUSE no better than chance (AUC 0.50) and ASK only weakly (0.65), so the regularities of Eq. 3 leak little of the gold action.

Results. Every model's AA (76.0–92.3%) is below the 94.0% of always answering (Table 5), and macro-F1 (0.53–0.63) is far below the text-only classifier's 0.81, so AA alone does not measure the action decision. The models fail mostly in one direction: they refuse unanswerable turns reliably (recall 89.5–95.3%), although many of these request data far outside any schema, but ask on only 34.9–57.7% of turns that need clarification, answering most of the rest outright. Asking more costs caution elsewhere: the model with the highest ASK recall, NEMOTRON-3-SUPER, also declines 24.2% of answerable turns, against 6.5% for GEMMA-4-31B-IT. On SQL the models are close ($\bar{F}$ 0.427–0.450, Exec 29.8–31.2%); over answered turns only, the order changes (0.47–0.57), be-

cause each model answers a different subset. Unparsable responses are at most 0.02%. No model completes a conversation (SR 0%), but 6.7–7.1% of topic blocks pass in full.

The protocol tags locate the failures (Table 6). Hard queries score 0.28–0.33, against 0.53–0.57 for Easy ones. Reference resolution is costly: coreference, anaphora, bridging, and above all cataphora (M-3) and ellipsis (M-8) score below turns without a cross-turn phenomenon, whereas carrying a filter forward (M-6) costs nothing. Financial ambiguity (A-4) is handled correctly on only about half of its turns, and $\bar{F}$ drifts down from the first turn of a block (0.45–0.47) to the fourth and later (0.42–0.45). Schema perturbation costs 0.09–0.12 in F and paraphrase chains at most 0.02. Ambiguity cascades cost 0.10–0.11, but 60–85% of that comes from declining the vaguer variant, so R-3 probes the boundary between answering and asking rather than robustness alone (Appendix F).

Gold actions are partly lexically predictable: the text-only baseline's ASK cues are vagueness markers (*significant*, *recently*) and its REFUSE cues are data no schema holds (*esg*, *gdp*). Trained on 12 databases and tested on the other 4, it still reaches 0.74, so the cues are not memorised topics. The main finding does not rest on them: on the 722 test turns the classifier gets wrong, ASK recall falls to 20.1–48.8%, while REFUSE recall stays at 79.5–88.7%. Experts confirm most clarification labels, rating 85.9% of the 78 sampled clarification turns fully correct. Per-conversation $\bar{F}$ barely varies with $S(C)$ in the test split (Spearman $\rho \leq 0.05$, on the restricted range $S(C) \geq 0.842$).

6 DISCUSSION AND LIMITATIONS

Three lessons follow. Current models under-ask, and more so on turns without lexical cues. Because gold actions are partly lexically predictable, action results should be read against the text-only baseline. And resolving a reference is harder than carrying a filter forward. Label-first generation is what lets each result be broken down by protocol, and it transfers beyond finance: in regulated domains such as healthcare or government, where real data cannot be released, a protocol-driven planner with execution-gated checks yields benchmarks whose coverage and quality are controlled by design.

6.1 LIMITATIONS

The databases are synthetic and the conversations English-only. The evaluation replays a fixed script: models see earlier user turns but not their own or the gold responses, so an ASK does not change the conversation. Training and test conversations share all 16 databases and all topics, and the gold action is partly predictable from surface cues (Section 5), so systems trained on the corpus should be compared against the text-only baseline. Disjoint and hybrid questions store one of their several gold outputs and are left out of SQL scoring, so only Q-2 and Q-4 test compound SQL. The test split comes from one batch, is selected by $S(C)$, and contains R-1 to R-3 variants only. GEMMA-4-31B-IT is both the Stage 2 judge and an evaluated model, and LLM judges can favour familiar outputs (Zheng et al., 2023; Panickssery et al., 2024).

Quality evidence is partial: each expert-labelled turn has one label, Correctness agreement is moderate ($\alpha = 0.53$), and 24.4% of answerable turns lack a SAR verdict. We release every turn's validation record and SAR verdict so users can filter or correct them.

7 CONCLUSION

We introduced FINCH 2.0, a benchmark for financial conversational agents that must decide at each turn whether to answer, ask, or refuse. A seeded planner fixes every turn's label and protocol before an LLM writes any text, and a three-stage Validation Funnel audits the result; experts rate 86.0% of sampled turns fully correct. Reference models refuse reliably but under-ask, fall below always answering on Action Accuracy, and score roughly half as well on hard queries as on easy ones, with cataphora and ellipsis the costliest cross-turn phenomena. Knowing when to ask, not only when to refuse, is therefore the capability that current models most lack. Because every turn carries its gold action, protocol tags, and validation record, FINCH 2.0 can locate such failures by phenomenon rather than report a single accuracy. Future work will add knowledge beyond the schema, live environments in which a clarification changes the conversation, near-miss unanswerable questions, and non-English personas.

AI USE STATEMENT

Generative AI tools are part of the method. *Synthetic data.* `openai/gpt-oss-120b` writes every conversation from its plan, redrafts turns that fail a check, and runs the SAR-Verifier (Sections 2 and 3). `google/gemma-4-31B-it` is the Stage 2 judge (Section 3). Their prompts are in Appendices B.10 and D. Conversation plans, database instances, protocol tags, complexity tiers, and all deterministic checks are produced by code without LLM calls. *Evaluation.* The three models of Section 5 are the objects of evaluation, not research tools. *Writing.* LLM assistants were used only to edit the manuscript for language and consistency. *Verification.* Every LLM-generated turn in the dataset passed the Validation Funnel, and 1,287 turns were rated by human experts (Section 3). The authors reviewed every suggested edit, checked all text, claims, numbers, and citations, and take full responsibility for the content of this work.

ETHICS STATEMENT

The authors have read and adhere to the ICLR Code of Ethics. *Data and privacy.* All 16 databases and every persona are synthetic. No real customer, counterparty, account, or trade record is used anywhere in the dataset. Instances are generated by seeded code (Appendix B), entities are procedurally generated rather than drawn from any real institution, and persona names are fictional. Schemas are modelled on the public documentation of real systems (Table 12) and contain none of their data; FIBO, GICS, and regulatory regimes serve only as references for sub-domain boundaries. *Human participants.* Seven financial-domain experts, one per vertical, rated 1,287 synthetic turns for Relevance and Correctness (Appendix C). The experts were colleagues of the authors working in the matching vertical, and each gave approval to take part and for their labels to be used. The task involved no personal or sensitive data, and the experts were not themselves the subject of study. *Harmful content.* The red-team series includes contradiction and SQL or prompt-injection turns (Appendix A). They exist to test whether a system refuses, and they run only against the released synthetic SQLite files. *Misuse and scope.* Results on FINCH 2.0 could inform copilots later deployed on real trading, treasury, or compliance systems. A high score does not certify such a system for production use, and the benchmark covers English only and the job functions of its persona catalogue. *Research integrity.* One evaluated model is also the Stage 2 judge, and Stage 2 Correctness agrees only moderately with experts; both are disclosed in Section 6.1.

REPRODUCIBILITY STATEMENT

Generation is deterministic outside the language model: Plan is a function of the base seed and the configuration (Proposition 1), and the planner, sub-seeds, and fusion rules are specified in Section 2.3 and Appendix A. Because LLM realisation can vary between runs, we release the realised conversations rather than asking users to regenerate them. Every model is named by its exact identifier (Sections 2, 3, and 5), and all prompts are reproduced in Appendices B.10 and D. The databases and generation procedure are specified in Appendix B, the deterministic checks and thresholds in Appendix E, the metrics in Section 5 and Appendix F, the human-validation protocol in Appendix C, per-database results in Appendix G, and the dataset in the datasheet of Appendix H. The generation and validation pipelines, the agent evaluation framework with its decoding settings, the 16 SQLite databases, and every conversation with its validation record will be released upon acceptance, together with the repair script and its manifest of every changed turn (Appendix H).

REFERENCES

Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. Benchmarking and improving text-to-SQL generation under ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7053–7074, 2023. doi: 10.18653/v1/2023.emnlp-main.436. URL https://arxiv.org/abs/2310.13659.

Hasan Alp Caferoğlu, Mehmet Serhat Çelik, and Özgür Ulusoy. SING-SQL: A synthetic data generation framework for in-domain text-to-SQL translation. *arXiv preprint arXiv:2509.25672*, 2025. URL https://arxiv.org/abs/2509.25672.

Shuaichen Chang, Jun Wang, Mingwen Dong, Lin Pan, Henghui Zhu, Alexander Hanbo Li, Wuwei Lan, Sheng Zhang, Jiarong Jiang, Joseph Lilien, Steve Ash, William Yang Wang, Zhiguo Wang,

Vittorio Castelli, Patrick Ng, and Bing Xiang. Dr.Spider: A diagnostic evaluation benchmark towards text-to-SQL robustness. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2301.08881.

Peter Baile Chen, Devin Yang, Weiyue Li, Fabian Wenz, Yi Zhang, Nesime Tatbul, Michael Cafarella, Çağatay Demiralp, and Michael Stonebraker. BEAVER: An enterprise benchmark for text-to-SQL. *arXiv preprint arXiv:2409.02038v1*, 2024.

Zhongjun Ding, Yin Lin, Tianjing Zeng, Rong Zhu, Bolin Ding, and Jingren Zhou. AmbiSQL: Interactive ambiguity detection and resolution for text-to-SQL, 2025. URL https://arxiv.org/abs/2508.15276.

Mingwen Dong, Nischal Ashok Kumar, Yiqun Hu, Anuj Chauhan, Chung-Wei Hang, Shuaichen Chang, Lin Pan, Wuwei Lan, Henghui Zhu, Jiarong Jiang, Patrick Ng, and Zhiguo Wang. PRACTIQ: A practical conversational text-to-SQL dataset with ambiguous and unanswerable queries. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Volume 1: Long Papers*, pp. 255–273, 2025. doi: 10.18653/v1/2025.naacl-long.13. URL https://arxiv.org/abs/2410.11076.

EDM Council. The financial industry business ontology (FIBO), 2026. https://spec.edmcouncil.org/fibo/. Accessed: July 2026.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-SQL empowered by large language models: A benchmark evaluation, 2023. URL https://arxiv.org/abs/2308.15363.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 2021. Originally arXiv:1803.09010 (2018).

Ziming Guo, Chao Ma, Yinggang Sun, Tiancheng Zhao, Guangyao Wang, and Hai Huang. Evaluating and enhancing LLMs for multi-turn text-to-SQL with multiple question types. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2025. doi: 10.1109/IJCNN64981.2025.11228279. URL https://arxiv.org/abs/2412.17867.

Nan Huo, Xiaohan Xu, Jinyang Li, Per Jacobsson, Shipei Lin, Bowen Qin, Binyuan Hui, Xiaolong Li, Ge Qu, Shuzheng Si, Linheng Han, Edward Alexander, Xintong Zhu, Rui Qin, Ruihan Yu, Yiyao Jin, Feige Zhou, Weihao Zhong, Yun Chen, Hongyu Liu, Chenhao Ma, Fatma Ozcan, Yannis Papakonstantinou, and Reynold Cheng. BIRD-INTERACT: Re-imagining text-to-SQL evaluation for large language models via lens of dynamic interactions. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.05318.

Tengjun Jin, Yoojin Choi, Yuxuan Zhu, and Daniel Kang. Pervasive annotation errors break text-to-SQL benchmarks and leaderboards. *Proceedings of the VLDB Endowment*, 19(5):931–944, 2026. doi: 10.14778/3796195.3796206.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA, 2 edition, 2004.

Ilia Kulikov, Chenxi Whitehouse, Tianhao Wu, Yixin Nie, Swarnadeep Saha, Eryk Helenowski, Weizhe Yuan, Olga Golovneva, Jack Lanchantin, Yoram Bachrach, Jakob Foerster, Xian Li, Han Fang, Sainbayar Sukhbaatar, and Jason Weston. Autodata: An agentic data scientist to create high quality synthetic data. *arXiv preprint arXiv:2606.25996*, 2026. URL https://arxiv.org/abs/2606.25996.

Rahul Kumar, Amar Raja Dibbu, Shrutendra Harsola, Vignesh Subrahmaniam, and Ashutosh Modi. BookSQL: A large scale text-to-SQL dataset for accounting domain. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1*, 2024. URL https://arxiv.org/abs/2406.07860.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. EHRSQL: A practical text-to-SQL benchmark for electronic health records. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 35, pp. 15589–15601, 2022. doi: 10.52202/068431-1134.

Gyubok Lee, Woosog Chay, Seonhee Cho, and Edward Choi. TrustSQL: Benchmarking text-to-SQL reliability with penalty-based scoring, 2024. URL https://arxiv.org/abs/2403.15879.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating language models on real-world enterprise text-to-SQL workflows. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2411.07763.

Haoyang Li, Shang Wu, Xiaokang Zhang, Xinmei Huang, Jing Zhang, Fuxin Jiang, Shuai Wang, Tieying Zhang, Jianjun Chen, Rui Shi, Hong Chen, and Cuiping Li. OmniSQL: Synthesizing high-quality text-to-SQL data at scale. *arXiv preprint arXiv:2503.02240*, 2025. URL https://arxiv.org/abs/2503.02240.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C.C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/83fc8fab1710363050bbd1d4b8cc0021-Abstract-Datasets_and_Benchmarks.html.

Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*, pp. 1235–1247, 2021. doi: 10.1145/3448016.3457261.

MSCI and S&P Dow Jones Indices. Global industry classification standard (GICS) methodology. https://www.msci.com/documents/1296102/11185224/GICS+Methodology+2023.pdf, 2023.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 68772–68802, 2024. doi: 10.52202/079017-2197.

Xinyu Pi, Bing Wang, Yan Gao, Jiaqi Guo, Zhoujun Li, and Jian-Guang Lou. Towards robustness of text-to-SQL models against natural and realistic adversarial table perturbation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1*, pp. 2007–2022, 2022. URL https://arxiv.org/abs/2212.09994.

Irina Saparina and Mirella Lapata. AMBROSIA: A benchmark for parsing ambiguous questions into database queries. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 90600–90628, 2024. doi: 10.52202/079017-2876.

Avinash Kumar Singh, Bhaskarjit Sarmah, and Stefano Pasquali. FINCH: Financial intelligence using natural language for contextualized SQL handling, 2025. URL https://arxiv.org/abs/2510.01887.

Pengfei Wang, Baolin Sun, Xuemei Dong, Yaxun Dai, Hongwei Yuan, Mengdie Chu, Yingqi Gao, Xiang Qi, Peng Zhang, and Ying Yan. Agentar-scale-SQL: Advancing text-to-SQL through orchestrated test-time scaling, 2025. URL https://arxiv.org/abs/2509.24403.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 2020. URL https://arxiv.org/abs/2001.11770.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task, 2018. EMNLP 2018.

Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter S. Lasecki, and Dragomir Radev. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1962–1979, 2019a. doi: 10.18653/v1/D19-1204.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. SParc: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4511–4523, 2019b. doi: 10.18653/v1/P19-1443.

Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, and Jinshu Lin. FinSQL: Model-agnostic LLMs-based text-to-SQL framework for financial analysis. In *Companion of the 2024 International Conference on Management of Data (SIGMOD)*, 2024. URL https://arxiv.org/abs/2401.10506.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2SQL: Generating structured queries from natural language using reinforcement learning, 2017. URL https://arxiv.org/abs/1709.00103.

APPENDIX CONTENTS

A Full Protocol Definitions **15**

A.1 P1: Cross-Turn Phenomena (M-1 to M-8) . . . 15

A.2 P3: Complexity Tiers (S-5) . . . 15

A.3 P3: Schema-Linking Depth (S-6) . . . 16

A.4 P4: Ambiguity Dimensions and Answerability Labels . . . 16

A.5 P5: Visualisation Intent Categories . . . 16

A.6 P6: The Six Attack Types . . . 16

B Databases, Personas, and Generation Details **17**

B.1 Formal Definitions . . . 17

B.2 Conversation Planner Details . . . 18

B.3 Industry Taxonomy Foundations, in Full . . . 19

B.4 Worked Example: GRAVITON . . . 19

B.5 Full Taxonomy: Verticals, Sub-domains, and Schemas . . . 19

B.6 Schema Design Principles . . . 20

B.7 Example DDL: GRAVITON . . . 20

B.8 Synthetic Population Methodology . . . 21

B.9 Synthetic Population Algorithm (GRAVITON) . . . 21

B.10 Record Generation: Generator Prompt and Verification Loop . . . 22

B.11 The Four Compound-Query Shapes (`sub_queries[]`) . . . 23

B.12 Worked Example: Plan, Table, and Realised Conversation . . . 23

B.13 Example Conversation Plan (JSON) . . . 24

B.14 Full Persona Catalogue . . . 24

C Human Validation Protocol **28**

D SAR-Verifier, Judge, and Agent Prompts **29**

E Deterministic Check Catalogues and Thresholds **32**

E.1 Stage 1, Layers 1–2: Static and Execution-Shape Checks . . . 32

E.2 Stage 2: The Deterministic Aggregate $D(t)$ . . . 32

F Evaluation Metrics: Details and Full Results **33**

G Per-Database Experimental Breakdowns **35**

H Datasheet **37**

A Full Protocol Definitions

Section 2.2 introduces the catalogue of 43 protocols that realise P1–P7, grouped in six series: M (multi-turn), S (schema and SQL structure), A (ambiguity), V (visualisation), R (red-team), and Q (query structure). Table 7 gives the series breakdown; Table 8 reproduces a representative sample of entries from the released `protocol_field_requirements.json`, one protocol per row with its id, stored name, and a summary of its required fields. The full 43-protocol table ships with the released code.

Table 7: Protocol series and the P1–P7 dimension each realises.

Series	Covers	Count	Maps onto
M	Multi-turn dependency	8	P1
S	Schema & SQL structure	7	S-1–S-4: P2; S-5–S-7: P3
A	Ambiguity and answerability	5	P4
V	Visualisation intent	12	V-1–V-11: P5 (V-0: none)
R	Adversarial red-team	6	P6
Q	Query structure	5	Q-2–Q-5: P7 (Q-1: atomic)

Table 8: Representative entries from the 43-protocol catalogue (`protocol_field_requirements.json`).

Id	Name	Required fields
M-1	Multi-Turn Coreference	`turn_details[].phenomenon=coreference`, `resolves_from_turn_id`, `entity_mentions[].resolved_entity`
A-2	Linguistic & Structural Ambiguity	`ambiguity_details.dimension` includes `linguistic`; `ambiguous_spans[].ambiguity_type` in {vagueness, scope, attachment}
V-4	Candlestick / OHLC Intent	`visualization_intent.chart_type=ohlc`, SQL projects open/high/low/close
R-2	Schema Perturbation	`attack_type=schema_perturbation`, `perturbed_terms[]`, expected SQL equal to the base SQL
Q-4	Intersection / Comparative Compound Query	`sub_queries[]` with at least two independent branches and a final entry combining them by `INTERSECT`, `EXCEPT`, or `UNION`

A.1 P1: Cross-Turn Phenomena (M-1 to M-8)

Coreference (M-1): a noun phrase in $q_{b,k}$ corefers with one in an earlier $q_{b,k'}$, and resolving it is necessary for the correct SQL ("bonds with yield above 5% then "filter those by maturity in 2026"). *Anaphora* (M-2): a pronoun or definite description refers back to an entity already introduced. *Cataphora* (M-3): an entity in $q_{b,k}$ refers forward to one introduced only later in the block. *Bridging anaphora* (M-4): an entity is inferentially, not explicitly, related to one in a prior turn ("bonds downgraded last month" then "what were their previous ratings?"). *Entity tracking and salience* (M-5): over a long block, multiple entities are introduced and the model must track which is salient at turn k. *Context carryover* (M-6): filters or aggregations established earlier persist unless explicitly revised ("show IG bonds from 2023" then "now add the HY ones", where the 2023 filter carries over). *Topic continuity and shift* (M-7): $q_{b,k}$ opens a new topic block, with or without an explicit transition marker. *Ellipsis* (M-8): $q_{b,k}$ is incomplete on its own and recovers the missing slot from an earlier turn (Definition 10).

A.2 P3: Complexity Tiers (S-5)

Table 9 restates the tier function $c(s)$ of equation 2. Released tiers are measured from the gold SQL after drafting; the target tier $c_{b,k}$ recorded in the plan (Definition 13) is not the released label.

Table 9: Complexity tiers as measured by $c(s)$. ✓ = sufficient for the tier; ○ = allowed; × = excluded.

Property of s	**Easy**	**Medium**	**Hard**
Tables referenced, $\lvert\mathrm{tab}(s)\rvert$	0–1	2–3	any
CTE or nested subquery	×	×	✓
Four or more tables	×	×	✓
Aggregation, window, `CASE`, date arithmetic	○	○	○

A.3 P3: Schema-Linking Depth (S-6)

Table 10 gives the linking-depth bins of S-6, measured from the tables $\mathrm{tab}(s)$ a gold query references (Section 2.2).

Table 10: Schema-linking depth tiers (S-6).

Tier	$\lvert\mathrm{tab}(s)\rvert$	**Description**
Local	1	Single table
Adjacent	2–3	One or two joins
Deep	4–8	Multi-hop joins
Global	9+	Traversal of a large part of the schema

A.4 P4: Ambiguity Dimensions and Answerability Labels

A-1 to A-4 are the four ambiguity dimensions and A-5 is unanswerability; Table 11 lists the three answerability labels. **Database-related (A-1)**: schema ambiguity (multiple tables plausibly answer q), column ambiguity (a term maps to several columns, e.g. "yield" to `yield_to_maturity`, `yield_to_worst`, or `current_yield`), value ambiguity, and join ambiguity (multiple valid join paths give different results). **Linguistic (A-2)**: vagueness ("recent", "large"), scope ambiguity, and attachment ambiguity. **LLM-reasoning (A-3)**: temporal scope, implied aggregation, and entity resolution ambiguity. **Domain-specific financial (A-4)**: term overloading ("exposure" means different things in risk, FX, and loan contexts), convention ambiguity (which duration measure), and product ambiguity (which bond type).

Table 11: Answerability labels and the expected model behaviour for each.

Label	**Meaning**	**Expected behaviour**
`answerable`	Unique correct SQL is recoverable	Answer with SQL
`clarification_needed`	Multiple valid SQLs exist	Ask a clarifying question
`unanswerable_schema`	Information not in the schema	Refuse and name the missing information

A.5 P5: Visualisation Intent Categories

P5 has eleven categories, V-1 to V-11 in the protocol catalogue (V-0 is the no-visualisation baseline): time-series, bar/grouped bar, stacked area, candlestick/OHLC, histogram/box plot, heat map, waterfall/P&L attribution, treemap/hierarchical aggregation, KPI tile, scatter with regression, and anomaly detection.

A.6 P6: The Six Attack Types

Paraphrase chain (R-1): $L \in \{3, 4, 5\}$ semantically equivalent rephrasings of q^*, each preserving s^* while varying vocabulary, syntax, and register. **Schema perturbation** (R-2): the question uses a synonym, abbreviation, or alternate name for a column or table not appearing verbatim in $\mathcal{S}_{ij}$.

Ambiguity cascade (R-3): the L questions become progressively vaguer while the expected SQL remains s^* throughout. The variants are under-specified but recoverable: the earlier turns of the block and the schema still admit a single reading, which is why the gold action stays ANSWER rather than ASK. We did not verify recoverability separately for every variant. **Contradiction injection** (R-4): a modified question containing an internally contradictory constraint. **Paraphrase + schema combination** (R-5): attacks R-1 and R-2 applied simultaneously. **SQL / prompt injection** (R-6): a question embedding SQL syntax or instruction-override text within natural language.

B DATABASES, PERSONAS, AND GENERATION DETAILS

This appendix gives the formal definitions behind Section 2, the full taxonomy and persona catalogue of the 16 databases, the schema-design principles and population method, the Generator prompt, and a worked plan-to-conversation example.

B.1 FORMAL DEFINITIONS

Section 2 states the database, persona, topic, schema, record, and conversation-plan mechanisms of FINCH 2.0 in prose, each pointing here for the exact formal statement it abbreviates.

Definition 3 (Finance Domain Taxonomy)**.** *$\mathcal{T} = \{(V_i, D_{ij}, \mathcal{D}_{ij})\}_{i=1,\dots,7;\ j=1,\dots,|V_i|}$ is a three-level hierarchy, vertical V_i, sub-domain $D_{ij} \in V_i$, and named database $\mathcal{D}_{ij} = (\mathcal{S}_{ij}, \mathcal{I}_{ij})$ for D_{ij}, with 7 verticals and $\sum_i |V_i| = 16$ databases ($V_1, \dots, V_7$ = capital markets, alternative and private-market investments, investment management, banking, risk and compliance, financial reporting and treasury, insurance; $\mathcal{D}_{11}$ = GRAVITON through $\mathcal{D}_{71}$ = ATLAS; full listing below). Fixing $\mathcal{T}$ before any $\mathcal{S}_{ij}$ is drafted makes coverage a design property: every vertical is represented, and the smallest holds 5.8% of turns (Figure 4).*

Definition 4 (Persona)**.** *A persona π has a name, a role, an expertise level in $\{$novice, intermediate, expert$\}$, a query style in $\{$terse, verbose, conversational$\}$, and an ambiguity tendency $\lambda \in \{\text{low}, \text{medium}, \text{high}\}$. Each $\mathcal{D}_{ij}$ is assigned a persona set $\Pi_{ij} \subseteq \Pi$ (the global persona universe) drawn from that sub-domain's real job functions.*

Definition 5 (Topic and Admission)**.** *$\tau = (\text{name}_\tau, \text{desc}_\tau, M_\tau)$: name, description, and SQL motif set. A candidate τ is admitted to $\mathcal{D}_{ij}$'s topic set Θ_{ij} iff $|\{\pi \in \Pi_{ij} : \text{rel}(\pi, \tau) = 1\}| \geq 2$, where* rel *indicates that π would plausibly generate a question in τ's thematic area.*

Instantiating Definitions 4–5 for $\mathcal{D}_{11}$ = GRAVITON (fixed-income trading, Capital Markets): Π_{11} includes π_{trader}, π_{pm}, π_{credit}, and π_{risk}, and τ = *bonds approaching maturity with no recent axe* is admitted since $\text{rel}(\pi_{\text{trader}}, \tau) = \text{rel}(\pi_{\text{risk}}, \tau) = 1$. The same construction, $D_{ij} \mapsto \Pi_{ij} \mapsto \Theta_{ij}$, is applied independently to the remaining fifteen $\mathcal{D}_{ij}$; the full persona catalogue is given in Table 14.

Definition 6 (Schema Admissibility and Design Targets)**.** *$\mathcal{S}_{ij}$ is admissible iff $\Phi_1 \wedge \Phi_3$, and it is designed against two further targets, Φ_2 and Φ_4, which are measured rather than enforced. Φ_1: its DDL is modelled on the publicly documented structure of a real-world system for D_{ij}. Φ_2 (target): $\exists T^* \in \mathcal{S}_{ij}$ (an anchor table) such that every $T \in \mathcal{S}_{ij}$ is reachable from T^* in the join graph G. Φ_3: $\mathcal{S}_{ij}$ has the structural capacity every dimension P1–P7 needs (Section 2.2), where P2 and P3 constrain the schema directly and the other dimensions require only that $\mathcal{S}_{ij}$ not preclude the phenomenon (e.g. enough column overlap for an ambiguous term or enough joinable tables for a compound query). Φ_4 (target): the enterprise-realism targets of P2, measured by S-1 to S-4,*

$$|\mathcal{S}_{ij}| \geq 20, \qquad \tfrac{1}{|\mathcal{S}_{ij}|} \textstyle\sum_{T \in \mathcal{S}_{ij}} |C_T| \in [8, 15], \qquad \tfrac{\#\text{ FK constraints}}{|\mathcal{S}_{ij}|} \in [2.5, 4.0], \qquad \text{diam}(G) \geq 3.$$

All 16 schemas are admissible and meet the table-count target (22 to 42 tables). Across the corpus, schemas average 7.8 columns and 0.9 declared foreign keys per table, below the next two targets, and 78.9% of tables lie in the largest connected component of their schema's declared-key join graph; only one schema's graph is fully connected. $\mathcal{I}_{ij}$ is then populated by seeded template pools, bounded random draws in plausible ranges (e.g. bond prices between 85 and 115), and relational-consistency rules (e.g., settlement date $\geq$ trade date); an excerpt of the DDL and the population routine for $\mathcal{D}_{11}$ = GRAVITON are given in Appendices B.7 and B.9.

Definition 7 (Record Generation)**.** *For admissible $(\mathcal{S}_{ij}, \mathcal{I}_{ij})$ with (Π_{ij}, Θ_{ij}),*

$$\mathcal{R}_{ij} = \big\{(q, s) = \text{Gen}(\pi, \tau, \ell, \mathcal{S}_{ij}|_\tau, h) :\ \ell \text{ from a plan satisfying equation 4}\big\},$$

where one Generator call per turn writes the utterance q and its gold SQL s jointly, conditioned on the persona π, the topic τ with its SQL motifs M_τ, the turn's plan entry ℓ, a schema slice $\mathcal{S}_{ij}|_\tau$, and the earlier turns h; there is no separate SQL-template stage. A turn is accepted only if it is valid under equation 5; the SAR-Verifier's advisory verdict is recorded alongside. The Generator prompt is reproduced, and the verification loop described, in Appendix B.10.

Definition 8 (FINCH 2.0 Record)**.** *A record $r = (q, s, \mathcal{D}_{ij}, \mathbf{P}_r, \mathbf{F}_r)$ pairs an utterance and its gold SQL, produced from database $\mathcal{D}_{ij}$ (Definition 3) by the record generation map of Definition 7, where $\mathbf{P}_r \subseteq \mathcal{P}$ (Definition 2) is the protocol set r exercises and $\mathbf{F}_r$ maps each $p \in \mathbf{P}_r$ to its annotation fields. $\mathcal{R}_{ij}$ collects every record generated from $\mathcal{D}_{ij}$, and $\mathcal{R} = \bigcup_{i,j} \mathcal{R}_{ij}$ is the record set underlying $\mathcal{C}$ in equation 1.*

Definition 9 (Ambiguity and Underspecification)**.** *A question q is* ambiguous *if it admits $s_1 \neq s_2$ with $\mathrm{exec}(s_1, \mathcal{I}_{ij}) \neq \mathrm{exec}(s_2, \mathcal{I}_{ij})$, both plausible given $\mathcal{S}_{ij}$. q is* underspecified *if exactly one intended s^* exists but recovering it from q and $\mathcal{S}_{ij}$ alone requires outside domain knowledge. The persona's ambiguity tendency λ (Definition 4) is passed to the Generator to condition phrasing; we do not measure its effect. We tag four dimensions, A-1 to A-4: database-related (following AmbiQT (Bhaskar et al., 2023)), linguistic (following AMBROSIA (Saparina & Lapata, 2024)), LLM-reasoning, and domain-specific financial ambiguity (term, convention, and product overloading), the last original to* FINCH 2.0.

Definition 10 (Ellipsis)**.** *A question $q_{b,k}$ is* elliptical *if it is grammatically or semantically incomplete as a standalone utterance, requiring a prior turn $q_{b,k'}$, $k' < k$, in the same block to recover the omitted subject entity, filter, aggregation, or metric, as in* "and the HY ones?"*. Ellipsis is tagged M-8 (P1).*

Definition 11 (Red-Teaming Record)**.** *A red-teaming record applies one attack ω_r to an answered base turn t with gold pair (q^*, s^*), producing variants $V(t) = \{(q_1^{\mathrm{atk}}, s_1^{\mathrm{exp}}), \ldots, (q_L^{\mathrm{atk}}, s_L^{\mathrm{exp}})\}$ with $L \in \{3, 4, 5\}$. The six attack types are paraphrase chain (R-1), schema perturbation (R-2), ambiguity cascade (R-3), contradiction injection (R-4), R-1 and R-2 combined (R-5), and SQL or prompt injection (R-6).*

Definition 12 (Compound Query Record)**.** *A compound-query record is a question q decomposing into sub-goals $\{g_1, \ldots, g_K\}$, $K \geq 2$, each with its own independently executable s_k and a dependency relation $g_k \to g_{k'}$ whenever g_k's SQL is used inside $g_{k'}$'s. Its gold s^* is the s_k of the sub-goal that nothing else depends on; when several sub-goals are independent outputs, the released s^* is the query of one of them, and the full decomposition is kept with the turn. Following QDMR (Wolfson et al., 2020), four dependency shapes are distinguished: sequential (a single chain to one final answer), disjoint (independent sub-goals with separate outputs and no shared answer), comparative (independent branches combined by a set operation such as `EXCEPT`), and hybrid (some sub-goals chained, others independent). A worked example of each shape is given in Appendix B.10.*

Definition 13 (Conversation Plan)**.** *A plan for topics $\tau_1, \ldots, \tau_m \in \Theta_{ij}$ with turn counts $n_1, \ldots, n_m$ is $L = \langle \ell_{1,1}, \ldots, \ell_{m,n_m} \rangle$, where $\ell_{b,k} = (a_{b,k}, p_{b,k}, e_{b,k})$ fixes the label, protocol (possibly none), and context edge $e_{b,k} \in \{t_{b,1}, \ldots, t_{b,k-1}, \varnothing\}$ of turn $t_{b,k}$ before any $q_{b,k}$ or $s_{b,k}$ is generated. Labels satisfy equation 3, protocols are drawn from $E(a_{b,k}, k, n_b)$ (Section 2.3), and bucket counts satisfy equation 4. The plan also records a target tier $c_{b,k}$, but the released tier is the measured $c(s)$ of equation 2.*

B.2 CONVERSATION PLANNER DETAILS

This subsection gives the full planner summarised in Section 2.3. In the seed $\sigma_x = \sigma_0 + \mathrm{CRC32}(x) \bmod 10^6$, CRC32 is the standard 32-bit cyclic redundancy check and the key (database, n) gives σ_n. The plan is built in five layers. (i) The persona is entry n of a weight-proportional list shuffled once per run. (ii) The number of topic blocks m is drawn uniformly from $\{m_{\min}, \ldots, m_{\max}\}$, and m distinct topics are drawn from Θ_{ij}. (iii) The turn total n_C is drawn from a triangular distribution on $[n_{\min}, n_{\max}]$ with mode n_{mode} and rounded, then split with two turns per block and the rest at random. (iv) Labels are drawn from a categorical distribution under the constraints of Eq. 3. An ambiguous turn takes its protocol from its sampled A-dimension; any other turn draws $p_{b,k}$ from an eligible set $E(a_{b,k}, k, n_b)$ that excludes backward-referring M protocols and M-3 at $k = 1$, M-3 at $k = n_b$, V, Q, and M-6 on clar turns, and all protocols on unans turns. When at least two turns remain in the block, a drawn M protocol is held for w turns, uniform

on $\{2, \ldots, \min(6, n_b - k + 1)\}$. (v) An idempotent repair enforces twelve plan-level fusion rules, released with the code.

Proposition 1 (Determinism and actionable endings)**.** *(a)* Plan *is a deterministic function of* σ_0*, the configuration, the catalogues, and the earlier plans used for de-duplication. (b) In every plan, no block ends on a clarification, and every clarification is followed by an answerable turn.*

Proof. (a) Every draw is seeded by some σ_x. (b) Both hold at sampling by Eq. 3; insertions add at most one clarification, never last, to a plan with none, and repair (v) resolves any final clarification. □

B.3 Industry Taxonomy Foundations, in Full

Section 2.1 grounds the FINCH 2.0 taxonomy in two sources the financial industry already uses to organise itself, plus the regulatory regime each sub-domain operates under, rather than inventing sub-domain boundaries from scratch. The Financial Industry Business Ontology (FIBO), maintained by the Enterprise Data Management Council and standardised through the Object Management Group, organises financial concepts into modules such as Business Entities, Financial Business and Commerce, Loans, and Securities (EDM Council, 2026); these modules map onto several of our verticals and supply the naming vocabulary used throughout the schemas, rather than an idiosyncratic internal naming convention. The Global Industry Classification Standard (GICS), developed jointly by MSCI and S&P Dow Jones Indices, classifies companies into sectors, industry groups, industries, and sub-industries (MSCI & S&P Dow Jones Indices, 2023); within its Financials sector it separates banks, other financial services, and insurance, a split reflected in our own vertical boundaries and used directly inside the equity-markets schema to support sector-conditioned queries such as comparing technology against financial-sector earnings yield. Each sub-domain further operates under a distinct regulatory regime, MiFID II and EMIR for capital markets, Basel III/IV for banking, AIFMD and UCITS for investment management, and Solvency II for insurance, and these regimes are not incidental: they determine what a real system in that sub-domain records, what it must report, and consequently what a practitioner would plausibly ask it. Grounding schema and query design in this regulatory structure, rather than in generic table design, is what makes the resulting queries operationally meaningful rather than merely syntactically plausible.

B.4 Worked Example: Graviton

This subsection expands the instantiation of Definitions 4–5, given after Definition 5, for $\mathcal{D}_{11} =$ GRAVITON, the fixed-income trading database under the Capital Markets vertical. GRAVITON is designed to represent a corporate-bond trading desk: securities master, TRACE trade reports, dealer axes, ratings history, and portfolio positions, modelled on the data behind Bloomberg TOMS and MarketAxess. Its persona set Π_{11} (Table 14) includes a Fixed Income Trader (terse, jargon-heavy, today-focused questions about axe levels and spreads), a Portfolio Manager (duration and benchmark-relative aggregations), a Credit Research Analyst (ratings-migration and issuer comparisons over longer horizons), and a Risk Manager (DV01 and concentration limits). Topics in Θ_{11} are generated by intersecting Π_{11} with the schema's tables, admitting a topic only when at least two personas would plausibly ask about it, yielding themes such as *daily trade volume by counterparty*, *spread analytics against a benchmark*, and *bonds approaching maturity with no recent axe*. The same construction, sub-domain to schema intent to persona set to admitted topics, is applied independently to the remaining fifteen databases; their personas are listed in Table 14, and the topic catalogue ships with the code.

B.5 Full Taxonomy: Verticals, Sub-domains, and Schemas

Table 12 lists the complete three-level taxonomy fixed before schema design began: seven verticals, sixteen sub-domains, and the sixteen named schemas built for them, together with the real commercial system or industry infrastructure each schema's data model is grounded in and the resulting table and column counts.

Table 12: The FINCH 2.0 finance domain taxonomy: 7 verticals, 16 sub-domains, 16 named schemas grounded in real commercial systems and industry infrastructure.

Vertical	Sub-domain	Schema	Real-world System Analogue	Tables	Columns
Capital Markets	Fixed Income Trading	GRAVITON	Bloomberg TOMS, MarketAxess, TRACE	28	280
	Equity Markets	APEX	Bloomberg EMSX, Fidessa OMS	35	247
	OTC Derivatives	NEXUS	Murex, Calypso, MarkitServ	34	255
	FX and Rates	FOREX	360T, FXall, Bloomberg FX	23	171
Alternative & Private-Market Investments	Private Markets	VERTEX	iLEVEL, Cobalt, eFront	24	205
	Real Estate / REITs	TERRA	MRI Software, Yardi, CoStar	22	172
Investment Management	Fund Accounting	MERIDIAN	SimCorp, BlackRock Aladdin, Charles River	42	338
	Quantitative Research	QUASAR	Axioma, MSCI Barra	30	201
	Wealth Management	STRATUM	Temenos WealthSuite, Avaloq, Orion	27	214
Banking	Retail Banking	COMPASS	Temenos T24, FIS Core, Finastra	33	265
	Commercial Banking	CITADEL	Finastra Fusion, Moody's CreditLens	32	240
	Payments	PULSE	SWIFT, Temenos Payments, Mastercard	23	177
Risk & Compliance	AML / KYC / Regulatory	SENTINEL	Actimize, Palantir, MSCI RiskManager	34	245
Financial Reporting & Treasury	Financial Reporting	PRISM	Oracle Financials, SAP S/4HANA	28	233
	Corporate Treasury	HORIZON	Kyriba, SAP Treasury, ION	25	213
Insurance	P&C and Life Insurance	ATLAS	Guidewire, Duck Creek, Majesco	29	218
			Total	**469**	**3,674**

B.6 SCHEMA DESIGN PRINCIPLES

Every schema in Table 12 was drafted against four principles before population. *Real-world system grounding*: the DDL structure, table naming conventions, and column names draw from published documentation of the named system analogue where available, and from domain expert consultation otherwise. *Anchor table primacy*: every schema has an anchor table representing the primary business entity of the domain (`securities_master` in GRAVITON, `policies` in ATLAS, `customers` in COMPASS), and other tables are designed to relate to it directly or transitively (Definition 6 reports measured connectivity). *Protocol compatibility*: temporal columns span multiple years of history to support rolling-window and point-in-time queries, a meaningful fraction of nullable columns carry actual NULL values in the synthetic instance, and at least one table per schema holds domain-reference values such as rating thresholds. *Enterprise scale*: table count, columns per table, foreign keys per table, and join-graph diameter are measured by S-1 to S-4 (Definition 6); only the table-count target is met by all 16 schemas.

B.7 EXAMPLE DDL: GRAVITON

To illustrate schema admissibility (Definition 6), Listing 1 reproduces the actual `CREATE TABLE` statements for GRAVITON's anchor table and two related tables, exactly as shipped in the released database generator. `securities_master` is the anchor (T^* in Φ_2). `pricing_history` references it through a declared foreign key with a composite primary key on (`cusip_id`, `price_date`), and the anchor refers to `issuers` through a logical key, `issuer_lei`, that is

not declared as a constraint. The remaining 25 GRAVITON tables, and the DDL for the other 15 databases, follow the same pattern and ship in full with the code release.

Listing 1: Excerpt of GRAVITON's DDL: the anchor table and two related tables.

```
CREATE TABLE securities_master (
   cusip_id TEXT PRIMARY KEY,
   isin TEXT UNIQUE NOT NULL,
   issuer_lei TEXT, -- FK -> issuers.lei
   issuer TEXT NOT NULL,
   ticker TEXT,
   amtissued REAL NOT NULL,
   amtoutstanding REAL NOT NULL,
   industrysector TEXT,
   industrygroup TEXT,
   industrysubgroup TEXT,
   country TEXT,
   collateraltype TEXT, -- Sr Unsecured | Sr Secured | Sub | Covered | ABS | MBS
   coupon REAL NOT NULL,
   coupontype TEXT NOT NULL, -- Fixed | Floating | Zero | StepUp
   couponfrequency TEXT NOT NULL, -- Semi-Annual | Annual | Quarterly | Monthly
   issuedate TEXT NOT NULL,
   maturitytype TEXT NOT NULL, -- Bullet | Callable | Puttable | Amortizing | Perpetual
   maturitydate TEXT NOT NULL,
   market TEXT, -- 144A | RegS | Public | Private
   securitytype TEXT NOT NULL, -- CorpBond | Govt | Agency | MBS | ABS | Muni | CD
   paymentrank TEXT, -- Senior | Subordinated | Junior | Mezzanine
   flag_144a TEXT NOT NULL DEFAULT 'N',
   ratingagency1rating TEXT,
   ratingagency2rating TEXT,
   currency TEXT NOT NULL DEFAULT 'USD',
   face_value REAL NOT NULL DEFAULT 1000.0
);

CREATE TABLE issuers (
   lei TEXT PRIMARY KEY,
   legal_name TEXT NOT NULL,
   domicile_country TEXT NOT NULL,
   sector TEXT,
   parent_lei TEXT, -- self-ref: ultimate parent
   is_sovereign TEXT NOT NULL DEFAULT 'N',
   is_financial TEXT NOT NULL DEFAULT 'N',
   total_assets_usd REAL,
   long_term_debt_usd REAL
);

CREATE TABLE pricing_history (
   cusip_id TEXT NOT NULL REFERENCES securities_master(cusip_id),
   price_date TEXT NOT NULL,
   clean_price REAL NOT NULL,
   dirty_price REAL NOT NULL,
   oas_bps REAL,
   zspread_bps REAL,
   PRIMARY KEY (cusip_id, price_date)
);
```

B.8 SYNTHETIC POPULATION METHODOLOGY

Each schema is populated with domain-realistic synthetic data through a deterministic, seeded factory pattern, using three complementary strategies. *Template pools* supply realistic entity names, identifiers, and categorical values sampled from domain knowledge, for example a pool of issuer names or a set of dealer codes matching real market participant identifier conventions. *Statistical distributions* draw numeric values from bounded random draws in ranges plausible for the domain, such as bond prices between 85 and 115; these are not fitted to real market data, so value distributions are simpler than in production. *Relational-consistency rules* enforce business logic that keeps the instance internally coherent, for instance requiring a maturity date to fall after its issue date or a settlement date to follow the trade date by one to three business days. All databases are distributed as SQLite files for zero-configuration portability and reproducible evaluation.

B.9 SYNTHETIC POPULATION ALGORITHM (GRAVITON)

Listing 2 distils the actual population routine for GRAVITON's `securities_master` and `trace` tables in the released database generator, showing all three strategies from the previous subsec-

tion operating together: weighted categorical pools for `securitytype`/`couponttype`, format-compliant identifier synthesis (`cusip()`, `isin()`, and `lei()` produce correctly formatted but not checksum-valid identifiers), and a relational-consistency rule tying `trace`'s settlement date to its trade date. Every other table in every database is populated by the same pattern: sample from a domain pool or a bounded range, then enforce any FK or temporal constraint relating the new row to rows already generated.

Listing 2: Distilled population logic for GRAVITON (trimmed for space; the full script ships with the release).

```
# securitytype is NOT uniform: weighted to real market composition
sec_types = (["CorpBond"]*40 + ["Govt"]*10 + ["Agency"]*10
           + ["MBS"]*10 + ["ABS"]*5 + ["Muni"]*5)
ig_ratings = ["Aaa","Aa1","Aa2", ... ] # investment-grade Moody's scale
hy_ratings = ["Ba1","Ba2","Ba3", ... ] # high-yield Moody's scale

for _ in range(N_SEC): # N_SEC = 5,000 securities
   st = choice(sec_types)
   is_ig = (st in ["Govt","Agency","Muni"] or random.random() < 0.6)
   rating = choice(ig_ratings) if is_ig else choice(hy_ratings)
   c = cusip() # 9-char CUSIP-shaped identifier
   secs.append((c, isin(), choice(issuer_leis), ...,
             rand_date("2010-01-01","2023-12-31"), # issuedate
             rand_date("2024-01-01","2045-12-31"))) # maturitydate > issuedate

for i in range(500000): # trace: 500K execution reports
   c = choice(cusips)
   td = rand_date() # trd_exctn_dt (trade date)
   sd = rand_date_after(td, 1, 3) # sttl_dt = td + [1,3] business days
   tr.append((c, td, sd, rand_price(85,115), ...))
```

B.10 RECORD GENERATION: GENERATOR PROMPT AND VERIFICATION LOOP

The map Gen of Definition 7 is one *Generator* LLM call per turn, whose system prompt is reproduced (abbreviated for space; the complete prompt, including every protocol-specific field requirement, ships with the code) in Listing 3. The Generator is given the turn's persona π, topic τ, a schema slice of $\mathcal{S}_{ij}$, prior turns already in the session, and the specific protocol this turn must exercise, and returns one JSON object containing the natural-language q (`utterance`) and, for answerable turns, the SQL s (`ground_truth_sql`) together with self-contained `test_cases[]` assertions.

Listing 3: Abbreviated Generator system prompt.

```
You are the Generator for the FINCH 2.0 synthetic conversational
text-to-SQL dataset.

Given a persona, a topic, a schema slice, prior turns already in the
session (if any), and a specific turn target (including which protocol
this turn must exercise), produce ONE turn as a single JSON object.
Output ONLY the JSON object -- no prose, no markdown fences.

Return a JSON object with whichever of these fields apply to this
turn's answerability (omit fields that do not apply, never invent
extra top-level keys): utterance, intention, is_ambiguous,
sql_complexity_tier, ground_truth_sql, nl_explanation,
clarification_question, unanswerable_explanation, test_cases,
sub_queries, conversation_turn_type, ...

Hard rules:
- ground_truth_sql must reference ONLY the tables/columns present in
  the schema slice you were given -- never invent a column or table.
- ground_truth_sql must be a single, complete, directly-executable
  statement; any CTE it uses must be fully defined inside the string.
- Do not leak table names, column names, or SQL keywords verbatim in
  `utterance` -- phrase it the way the persona would actually talk.
- Every answerable turn MUST include a non-empty test_cases[] list
  (>=1 entry): these, not a non-empty result set, are what is graded.
- This database is a STATIC, HISTORICAL snapshot -- do not assume it
  contains data up to the real-world current date.
```

B.11 THE FOUR COMPOUND-QUERY SHAPES (`SUB_QUERIES[]`)

Definition 12's four dependency shapes are realised inside the same Generator call through `sub_queries[]`, a list of {`id, task, sql, depends_on`} entries. *Sequential*: "Find the manager of the IT department and their office number" decomposes into one entry finding the manager's id (`depends_on: []`) and a second selecting that manager's office number (`depends_on: [1]`). *Disjoint*: "Show total sales for 2023 and list active product categories" decomposes into two entries, both `depends_on: []`, each a complete independent answer. *Comparative*: "Which customers bought in January but not February" decomposes into two independent branches (January ids, February ids) plus a third entry whose SQL is `<jan> EXCEPT <feb>` and `depends_on` both. *Hybrid*: a mix of chained and independent entries, e.g. one root entry with two children that each `depends_on` it plus one fully independent entry, used when a question bundles several distinct asks of varying dependency. `ground_truth_sql` is set to the SQL of the entry nothing else depends on; when there are several, it holds one of them (in the test split, the last entry for 285 of 490 disjoint and 209 of 441 hybrid turns), and `sub_queries[]` keeps the others (Definition 12).

Every answerable turn that is valid under equation 5 is then audited by the *SAR-Verifier*, an agentic LLM auditor that sees the utterance, the candidate s, its executed result on $\mathcal{I}_{ij}$, and the in-scope schema slice, and may issue up to five read-only probe queries before returning a verdict. The verdict and its evidence are recorded with the turn but do not block acceptance. The SAR-Verifier is layer L3 of Stage 1 of the Validation Funnel (Section 3); its prompt appears with the Stage-2 judge prompts in Appendix D.

B.12 WORKED EXAMPLE: PLAN, TABLE, AND REALISED CONVERSATION

Consider a two-topic plan for $\mathcal{D}_{11}$ = GRAVITON under the Fixed Income Trader persona (Table 14): topic τ_1 = *daily trade volume by counterparty* contributes two turns, and topic τ_2 = *bonds approaching maturity with no recent axe* contributes two more after an explicit topic switch. Table 13 lists $\ell_{b,k}$ (Definition 13) for all four turns, fixed before any question or SQL exists: $a_{b,k}$ fixes the label, $c_{b,k}$ records a target tier (the released tier is the measured $c(s)$), $p_{b,k}$ picks which of the 43 catalogued protocols (Section 2.2) the turn must exercise, and $e_{b,k}$ fixes which earlier turn, if any, it depends on. T1-2 is pre-assigned M-2 (anaphora) with a context edge back to T1-1; T2-1 opens with a pre-assigned clarification (A-2, vagueness) that T2-2 must resolve. The plan is not released as a separate artefact; it is consumed by the Generator of Definition 7, and its full JSON form is reproduced below (Listing 4).

Table 13: Turn targets $\ell_{b,k}$ of the example plan, with the recorded target tier $c_{b,k}$.

Turn	$a_{b,k}$	$c_{b,k}$	$p_{b,k}$	$e_{b,k}$
T1-1	answerable	easy	–	–
T1-2	answerable	easy	M-2	T1-1
T2-1	clarification	–	A-2	–
T2-2	answerable	hard	Q-1	T2-1

Figure 5 shows the conversation the Generator realises from this plan: each turn target $\ell_{b,k}$ becomes one turn, T1-2's context edge surfaces as an unresolved pronoun ("that") the Generator must resolve against T1-1, and T2-1's pre-assigned clarification is asked rather than answered, with T2-2 required to resolve it before its own SQL is accepted.

T1-1, Trader: "Which dealers did we trade the most with today by notional?"
System: `SELECT counterparty, SUM(par_amt) ... GROUP BY counterparty ORDER BY SUM(par_amt) DESC LIMIT 5;` [answerable, easy]

T1-2, Trader: "And how does *that* compare to last week?"
System: `... WHERE trade_date BETWEEN date(:ref,'-7 day') AND date(:ref,'-1 day') ...` [answerable, easy, M-2 anaphora, ctx: T1-1]

topic switch (explicit) → bonds approaching maturity with no recent axe

T2-1, Trader: "Show me bonds with no recent axe."
System (ask): "No recent axe within how many days, 7, 15, or 30?" [clarification_needed, A-2]

T2-2, Trader: "30 days, and maturing in the next 6 months."
System: `... WHERE NOT EXISTS (SELECT 1 FROM axe ...) AND maturity_date <= ...` ✓ [answerable, hard, ctx: T2-1]

Figure 5: The conversation C realised from the plan of Table 13 (GRAVITON, Fixed Income Trader). ✓ marks turns whose SQL is valid under equation 5; SQL is abbreviated and schematic, and `:ref` is the snapshot's fixed reference date.

B.13 EXAMPLE CONVERSATION PLAN (JSON)

Listing 4 gives the full JSON form of the example plan summarised in Table 13 and realised in Figure 5, exactly as `build_topic_plan` emits it, one entry per topic block, each holding its `turn_targets[]` list of `TurnTarget` objects. No `utterance` or `ground_truth_sql` field exists yet at this stage; those are filled in turn by turn during the pass described in Section 2.4.

Listing 4: The example plan of Table 13, in the exact shape `build_topic_plan` returns it.

```
[
  {
    "topic_cluster_id": "daily_trade_volume_by_counterparty",
    "topic_label": "Daily Trade Volume by Counterparty",
    "conversation_type": "exploration",
    "topic_switch_spec": {"is_switch": false, "switch_type": null},
    "turn_targets": [
      {"turn_id": "T1-1", "answerability": "answerable",
       "sql_complexity_tier": "easy", "protocol_id": null,
       "context_needed": [], "topic_switch_type": null,
       "ambiguity_style": null, "ambiguity_subtype": null},
      {"turn_id": "T1-2", "answerability": "answerable",
       "sql_complexity_tier": "easy", "protocol_id": "M-2",
       "context_needed": ["T1-1"], "topic_switch_type": null,
       "ambiguity_style": null, "ambiguity_subtype": null}
    ]
  },
  {
    "topic_cluster_id": "bonds_approaching_maturity_no_recent_axe",
    "topic_label": "Bonds Approaching Maturity with No Recent Axe",
    "conversation_type": "clarification",
    "topic_switch_spec": {"is_switch": true, "switch_type": "explicit"},
    "turn_targets": [
      {"turn_id": "T2-1", "answerability": "clarification_needed",
       "sql_complexity_tier": null, "protocol_id": "A-2",
       "context_needed": [], "topic_switch_type": "explicit",
       "ambiguity_style": "explicit", "ambiguity_subtype": "vagueness"},
      {"turn_id": "T2-2", "answerability": "answerable",
       "sql_complexity_tier": "hard", "protocol_id": "Q-1",
       "context_needed": ["T2-1"], "topic_switch_type": null,
       "ambiguity_style": null, "ambiguity_subtype": null}
    ]
  }
]
```

B.14 FULL PERSONA CATALOGUE

Table 14 lists the personas of each of the 16 databases with a summary of the role description used to condition question generation.

Table 14: The FINCH 2.0 persona catalogue, by database.

Database	Persona	Description
GRAVITON	Fixed Income Trader	Executes bond trades, monitors axe levels and TRACE prices, calculates spread vs. benchmarks. Queries are terse, jargon-heavy, and temporally immediate ("today", "last print", "current axe").
	Portfolio Manager	Manages duration, credit quality, and benchmark-relative positioning. Queries involve portfolio-level aggregations, DV01 calculations, and sector attribution.
	Credit Research Analyst	Tracks ratings migrations, spread history, and corporate event impacts. Queries span longer time horizons and involve comparative analysis across issuers.
	Price & Liquidity Reconciliation Analyst	Reconciles TRACE execution prices against axe levels, identifies outliers, and generates exception reports. Queries are exception-driven and involve complex HAVING and anti-join patterns.
	Risk Manager (fixed income)	Monitors DV01 limits, portfolio VaR, and concentration risk. Queries involve threshold-based alerts and portfolio-level aggregations.
APEX	Equity Trader	Monitors execution quality (VWAP slippage, market impact), manages order flow, and tracks fill rates. Queries are intraday, execution-focused, and frequently involve VWAP benchmarking.
	Portfolio Manager (equities)	Manages sector allocation, factor exposure, and earnings-driven positioning. Queries span daily and weekly horizons.
	Equity Research Analyst	Screens for investment candidates based on fundamental metrics, tracks earnings revisions, and monitors analyst consensus.
	Prime Brokerage Analyst	Manages short interest, borrow availability, and margin calculations for hedge fund clients.
	Index Operations Analyst	Manages index constituent lists, rebalancing events, and tracking error calculations.
NEXUS	Derivatives Trader	Prices swaps and options, manages greeks, and executes hedges. Queries involve real-time valuation and delta/vega aggregation.
	CVA / XVA Desk Analyst	Calculates credit, funding, and capital valuation adjustments. Queries are computationally intensive and involve counterparty-level aggregations.
	Collateral Manager	Manages margin calls, collateral eligibility, and substitution. Queries are operationally focused and deadline-driven.
	Derivatives Operations	Manages trade confirmation, lifecycle events (resets, expirations, exercises), and settlement. Queries focus on trade status and outstanding obligations.
FOREX	FX Spot Trader	Executes spot and tom/next trades, monitors real-time net open position against limits, and manages P&L versus fixing benchmarks (ECB, WM/Reuters). Queries are highly time-sensitive and terse, often referencing only the currency pair.
	FX Hedge Analyst (Corporate)	Manages a corporate FX hedging programme covering forecast exposures in multiple currencies. Queries focus on hedge effectiveness testing (IAS 39 / IFRS 9), roll schedules, and hedge ratio adequacy.
	Treasury Analyst	Monitors cash flow hedges for a multinational treasury, tracks forward contract schedules, and models mark-to-market P&L on open hedges.
	FX Risk Manager	Enforces NOP (net open position) limits, VaR by currency, and intraday limit breaches. Queries involve cross-currency aggregation with base-currency conversion.

Continued on next page

Database	Persona	Description
	Macro Economist / Strategist	Analyses currency drivers, central bank policy impact on rates, and cross-currency correlation patterns. Queries span multi-year historical data with low jargon relative to traders.
VERTEX	PE Portfolio Manager	Monitors IRR, MOIC, and DPI across portfolio companies and vintages. Queries involve fund-level and company-level performance roll-ups against cost basis.
	Investor Relations (IR) Officer	Prepares LP reporting on NAV, capital account statements, and distribution history. Queries are LP-centric, formatted for external reporting, and use plain language.
	Deal Team Analyst	Tracks deal pipeline, monitors portfolio company performance against investment thesis, and flags covenant breaches in private credit investments.
	Fund CFO / Controller	Manages capital account calculations, fee billing (management fee, carried interest waterfall), and fund-level financial statements.
	LP Due Diligence Analyst (Non-Expert)	An institutional investor analyst reviewing fund performance with limited knowledge of the administrator's data model. Uses LP-side terminology ("net return", "distributions received") that may not match column names directly.
TERRA	REIT Analyst (sell-side)	Analyses AFFO, FFO, cap rate compression, and same-store NOI growth. Queries span property portfolios and sector-level comparisons (office vs. industrial vs. residential).
	Asset Manager (real estate)	Manages individual assets, monitors lease expiry concentration, vacancy risk, and CapEx programmes. Queries are property-level and operationally focused.
	Acquisitions Analyst	Evaluates potential acquisitions against underwriting assumptions, models going-in and stabilised cap rates, and tracks deal pipeline.
	Debt / Finance Analyst	Monitors loan-to-value ratios, debt service coverage ratios (DSCR), and covenant compliance across the debt stack.
	Property Operations Manager (Non-Expert)	Manages day-to-day property operations; queries rent collection and tenant arrears using operational language ("outstanding rent", "overdue tenants") rather than schema column names.
MERIDIAN	Fund Accountant	Calculates daily NAV, reconciles holdings against custodian records, and accrues expenses. Queries involve complex multi-table joins over the fund-share class-holding hierarchy.
	Portfolio Manager (buy-side)	Monitors active share, tracking error, sector attribution, and benchmark-relative performance. Queries blend holdings data with performance attribution and benchmark data.
	Client Services Manager	Responds to investor queries about AUM, returns, and distributions. Queries are client-facing, typically at fund or share class level.
	Fund Compliance Officer	Monitors UCITS/AIFMD constraints, investment mandate limits, and regulatory concentration thresholds. Queries are breach-focused.
	Performance Analyst	Runs Brinson–Hood–Beebower attribution, calculates risk-adjusted returns (Sharpe, Sortino, Information Ratio), and prepares GIPS-compliant composite reports.
QUASAR	Quant Researcher	Develops and tests factors (value, momentum, quality, low-volatility), analyses factor decay, and evaluates signal IC across time and market regimes.
	Portfolio Constructor	Translates alpha signals into portfolio weights subject to risk model constraints, sector limits, and turnover budgets.
	Performance Attribution Analyst	Decomposes live returns into factor and idiosyncratic components, reconciles backtest vs. live performance, and identifies implementation shortfall.

Continued on next page

Database	Persona	Description
	Risk Manager (quant)	Monitors factor concentration, realised vs. forecast volatility, and drawdown thresholds using rolling window calculations.
	Junior Analyst (Non-Expert)	Runs ad hoc factor screening without deep knowledge of the factor model architecture. Uses general terminology ("which stocks have high momentum") that maps ambiguously to multiple columns.
STRATUM	Wealth Advisor	Monitors client portfolio performance against financial goals, prepares client review materials, and identifies rebalancing needs using plain financial language.
	Tax Specialist	Manages tax-loss harvesting opportunities, monitors wash-sale rule compliance, and optimises tax-lot selection for distributions.
	Portfolio Analyst	Tracks asset allocation drift, generates rebalancing proposals, and measures implementation of model portfolios.
	Compliance Officer (suitability)	Checks that client portfolios remain aligned with KYC suitability profiles and regulatory obligations (MiFID II, Reg BI).
	Client (Non-Expert)	An HNW client querying their own financial picture with no knowledge of the underlying data model; questions are highly underspecified ("how much have I made this year").
COMPASS	Branch Manager	Monitors sales performance, customer acquisition, and product cross-sell rates. Queries are branch-level aggregations with period comparisons.
	Credit Analyst (retail)	Assesses delinquency rates, loan-to-value ratios, and default probability by segment using cohort analysis.
	Customer Analytics Manager	Builds customer segments, monitors churn propensity, and identifies cross-sell opportunities via customer-level joins.
	Fraud Operations Analyst	Investigates transaction anomalies, manages fraud alert queues, and measures false-positive rates using pattern-matching and temporal clustering.
	Deposit Pricing Manager	Monitors balance migration between products, tracks rate elasticity, and models pricing scenarios over time series.
CITADEL	Relationship Manager (corporate)	Manages client exposure, monitors covenant status, and identifies upsell opportunities via facility-level summaries.
	Credit Officer	Assesses PD/LGD/EAD, calculates expected loss, and monitors credit concentration at the portfolio level.
	Syndication Desk Analyst	Manages participant shares, agent fee distributions, and lender register updates via multi-party waterfall calculations.
	Trade Finance Specialist	Manages letter-of-credit issuances, documentary collections, and guarantee utilisations with operational status tracking.
PULSE	Payments Operations Manager	Monitors settlement queues, investigates failed payments, and tracks SLA compliance. Queries are status-driven and time-sensitive.
	Fraud Analyst	Investigates fraud alerts, measures false-positive rates, and identifies emerging fraud patterns via temporal clustering.
	Liquidity Manager (intraday)	Monitors nostro balances, manages intraday funding, and measures peak liquidity usage via real-time position aggregation.
	Correspondent Banking Analyst	Manages bilateral limits with correspondent banks, monitors net positions, and settles FX at the counterparty level.
SENTINEL	AML Analyst	Investigates transaction monitoring alerts, manages case lifecycles, and prepares SAR filings via transaction network analysis.
	KYC / CDD Officer	Manages customer due diligence profiles, schedules periodic reviews, and escalates PEP/sanctions hits.
	Capital Management Analyst	Monitors CET1, Tier 1, and Total Capital ratios against regulatory minimums using Basel III formulas over RWA components.

Continued on next page

Database	Persona	Description
	Regulatory Reporting Manager	Prepares MiFID II transaction reports, EMIR trade repository submissions, and FATCA/CRS reporting with precise field mapping.
	Stress Testing Analyst	Runs CCAR/DFAST scenarios, measures capital depletion under stress, and reports to the board via scenario-keyed aggregations.
PRISM	Financial Controller	Manages the period-close process, reconciles sub-ledger balances to the GL, and reviews accruals and prepayments for completeness.
	Consolidation Analyst	Prepares multi-entity consolidated financial statements, manages intercompany eliminations, and performs FX translation for foreign subsidiaries.
	External Auditor	Tests journal entry completeness and accuracy, identifies unusual manual entries, and traces transactions to supporting documentation.
	FP&A Analyst	Performs variance analysis (actual vs. budget vs. prior year), builds bridge analyses, and reports to senior management.
	Tax Analyst	Reconciles book income to taxable income, identifies permanent and temporary differences, and tracks deferred tax assets and liabilities.
HORIZON	Group Treasurer	Manages global liquidity, funding strategy, and bank relationship economics at the portfolio level (total cash, debt maturity, net debt).
	Cash Manager	Executes daily cash positioning, manages inter-account transfers, and monitors cash pool balances across entities and currencies.
	FX Hedging Analyst	Manages the corporate FX hedge book, monitors hedge effectiveness under IFRS 9, and reports hedge P&L to the Controller.
	Debt Capital Markets Analyst	Monitors debt maturity profile, covenant compliance, and interest rate fixing schedules with headroom calculations.
	CFO (Non-Expert, executive)	Reviews working capital metrics, net cash position, and debt capacity at a summary level using underspecified executive vocabulary ("how much cash do we have").
ATLAS	Underwriter	Assesses risk, sets premiums, and monitors loss ratios by line of business and geographic segment using experience-to-expected comparisons.
	Claims Adjuster	Manages claim status, payment history, and recovery actions with operationally focused, case-level drill-down.
	Actuary	Calculates IBNR reserves, combined ratios, and tail development factors, including loss development triangles.
	Reinsurance Analyst	Manages treaty cession rates, monitors recovery thresholds, and reconciles reinsurance settlements via multi-party waterfall calculations.

C HUMAN VALIDATION PROTOCOL

Sample. For each of the 16 databases we took the three highest- and the three lowest-scoring conversations by $S(C)$ from the first two batches. Within each end, the selection was balanced across protocol buckets and the presence of clarification and unanswerable turns, so that the chosen conversations were not near-duplicates. Six further conversations from five databases complete the sample of 102 conversations and 1,287 turns. The sample therefore covers both ends of the score range rather than its middle.

Participants. Seven experts took part, one per financial vertical, each a colleague of the authors who works in that vertical. Each gave approval to take part and for their labels to be used. An expert labelled every sampled turn of their vertical's databases, so each turn received one label; we therefore report agreement with Stage 2 and cannot report agreement between experts.

Task. Financial-domain experts from each database's vertical scored every turn on the two Stage-2 criteria (Section 3): Relevance (whether a professional in the persona's role would ask the question at that point) and Correctness (whether the response answers it). Each score was 0, 0.5, or 1, or *cannot say*. Experts did not see the Stage-2 scores. Three Relevance and four Correctness labels were *cannot say* and are excluded, leaving 1,284 and 1,283 labels.

Absolute ratings. Table 15 reports the share of turns experts rated fully correct and fully relevant, with 95% confidence intervals from 2,000 bootstrap resamples of conversations (seed 0). All 30 sampled conversations in the test split belong to the high-$S(C)$ group. The same table relates expert Correctness to the SAR-Verifier's verdict (Section 3). Across the corpus, the verifier marks 43,655 answerable turns incorrect and cites E1 for 87.4% of them, E2 for 35.1%, E4 for 6.7%, and E3 for 0.9% (a turn can carry several); its uncertain verdicts are mostly E4 (91.3%).

Table 15: Expert ratings of the Stage 3 sample. "$= 1$" is the share of turns rated 1 on the $\{0, 0.5, 1\}$ scale; brackets give 95% bootstrap CIs by conversation.

Subset	Turns / conv.	Correctness $= 1$	Correctness $= 0$	Relevance $= 1$
All sampled turns	1,283 / 102	86.0 [83.1, 88.9]	3.0 [1.7, 4.6]	82.9 [79.3, 86.3]
Answerable only	1,103 / 102	84.9 [81.6, 87.9]	3.3 [1.8, 5.1]	82.2 [78.4, 86.0]
High-$S(C)$ group	584 / 48	95.5 [93.6, 97.4]	0.5 [0.0, 1.4]	98.3 [97.2, 99.2]
Low-$S(C)$ group	588 / 48	75.9 [72.0, 79.6]	5.8 [3.1, 8.7]	67.2 [62.9, 71.4]
In the test split	368 / 30	95.7 [92.7, 97.9]	0.0 [0.0, 0.0]	98.1 [96.7, 99.2]
Clarification turns	78 / 50	85.9 [78.5, 93.2]	2.6 [0.0, 6.5]	71.8 [62.2, 82.2]
Unanswerable turns	102 / 53	99.0 [96.9, 100.0]	1.0 [0.0, 3.1]	98.0 [95.0, 100.0]
SAR verdict correct	572	89.5	–	–
SAR verdict incorrect	326	77.3	–	–
SAR verdict uncertain	123	88.6	–	–

Agreement. Each Stage-2 score is rounded to the nearest value in $\{0, 0.5, 1\}$ (0.25 to 0 and 0.75 to 1) before computing weighted Cohen's κ (linear weights over the three ordered levels), Krippendorff's α (ordinal), and the direction classes of Table 26. MAE is reported on raw and rounded Stage-2 scores, and Spearman's r_s on raw scores (Tables 3, 24, and 25). Few Stage-2 scores fall exactly on 0.25 or 0.75 (58 for Relevance, 34 for Correctness), and the conclusions do not depend on the rounding: mapping both to 0.5 gives agree / expert-higher / Stage-2-higher shares of 82.8 / 13.8 / 3.4 for Relevance and 84.1 / 9.9 / 6.0 for Correctness, against 83.5 / 13.2 / 3.3 and 85.2 / 8.3 / 6.5 under the rounding used in the paper.

D SAR-VERIFIER, JUDGE, AND AGENT PROMPTS

This appendix reproduces the system prompts of the SAR-Verifier (Stage 1, L3) and of the Stage-2 judges (Section 3). Each judge's per-protocol question ({`relevance_prompt`}, {`question_correctness_prompt`}, {`correctness_prompt`}) is filled from the released `protocol_test_cases.json`. Judges reason through the named criteria before scoring and return strict JSON; the output-format instruction appended to each judge prompt is omitted here.

Listing 5: SAR-Verifier system prompt (Stage 1, L3).

```
You are the SAR-Verifier for the FINCH 2.0 text-to-SQL dataset: a SKEPTICAL, independent
     auditor. You did NOT write the SQL you are reviewing, and your job is to try to prove it
      WRONG, not to rubber-stamp it. You audit one turn: a user's natural-language question
     and a candidate `ground_truth_sql` that is claimed to answer it against a real finance
     database.

You judge the candidate against four error patterns:
- E1 - SQL does not match the natural-language question. The query runs, but computes the
     wrong thing: wrong column, wrong filter, wrong aggregation, a ">" where the question
     means ">=", missing GROUP BY, answers a different question than the one asked.
- E2 - SQL does not match database semantics. It uses a column/table that does not mean what
     the question needs, is missing a DISTINCT (so rows are double-counted), fans out through
      a join, or has a unit/scale error.
- E3 - SQL does not match finance domain knowledge. It implements a financial term against
     the wrong convention: T+2 settlement counted as calendar (not business) days, YTD
     anchored to the calendar (not fiscal) year, the wrong duration measure (modified vs
     Macaulay vs DV01), wrong day-count, etc.
```

```
- E4 - the natural-language question itself is genuinely ambiguous or underspecified, so no
     single SQL can be uniquely correct.

You may INVESTIGATE the database before deciding, by issuing your own read-only probe queries
      (SELECT/WITH only). Use probes to check your suspicions: inspect distinct values, row
     counts, whether a filter matches anything, whether a join inflates rows, whether an
     alternative interpretation returns a different result. Probe when it would change your
     verdict; do not probe aimlessly.

Respond at EVERY step with EXACTLY ONE JSON object and nothing else - no prose before or
     after, no markdown fences.

To investigate, return:
{"action": "probe", "sql": "<a single read-only SELECT/WITH query>", "reason": "<what you are
      checking>"}

When you are ready to decide (or when told it is your final step), return:
{"action": "verdict",
 "verdict": "correct" | "incorrect" | "uncertain",
 "confidence": <float 0.0-1.0>,
 "error_patterns": {"E1": <bool>, "E2": <bool>, "E3": <bool>, "E4": <bool>},
 "findings": "<one or two sentences: what is right or wrong, citing evidence>",
 "proposed_correction": "<a corrected SQL string, or null if none is needed>"}

Rules:
- "incorrect" means you have concrete evidence the SQL does NOT correctly answer the question
      (set the matching E-pattern true and, ideally, give a proposed_correction).
- "correct" means you checked and the SQL genuinely answers the question.
- "uncertain" means you could not resolve it within your probe budget - use it sparingly.
- Only SELECT/WITH probes are allowed; any other statement will be rejected.
```

Listing 6: Stage-2 Relevance judge, base system prompt; {relevance_prompt} is filled per protocol.

```
You are judging whether a question in a financial conversational-agent benchmark is something
      a real domain professional would actually ask, given the database, topic, and protocol
     in play.

The specific thing to judge: {relevance_prompt}

Check each of the following before deciding:
1. Fit - does this question match what someone in this role would plausibly need to know,
     given this database AND this topic specifically?
2. Phrasing - does it read like a real professional's question, not a templated or
     artificially precise one?
3. Scope - is it scoped the way a real question would be here, not overly broad or oddly
     narrow?

Reason through criteria 1-3 in order, briefly noting what you observe for each, before you
     decide a score.

Score anchors:
1 = yes, clearly -- satisfies every criterion above for this exact context.
0.5 = plausible but falls short on at least one criterion (off in phrasing, scope, or fit for
      this topic).
0 = no real professional would ask this in this form.
```

Listing 7: Added to the Relevance prompt for M-series turns.

```
This turn belongs to a multi-turn (M-series) protocol, which adds a 4th criterion to the ones
      above:
4. Continuity - is it a natural, coherent thing to say GIVEN the immediately preceding
     question below, not just sensible read in isolation? A real professional's follow-up
     depends on what they just asked.

Include criterion 4 in your step-by-step reasoning, and in the score anchors above -- a score
      of 1 now requires satisfying continuity too, not just criteria 1-3. If this turn is the
      first turn of the very first topic in the conversation, there is no earlier question
     for it to connect to -- in that case, skip criterion 4 and judge only against criteria
     1-3, as a standalone question.
```

Listing 8: Added to the Relevance prompt when the protocol defines a Question Correctness check.

```
You are ALSO separately rating this question's own protocol adherence, in the SAME response:
     {question_correctness_prompt}
```

```
This is a DIFFERENT judgement than relevance above -- relevance asks whether a real
     professional would phrase it this way at all; this one checks:
1. Genuineness - is this a REAL instance of the claimed phenomenon (a genuine connection to
     the prior turn, a genuine ambiguity of this type, a genuine query structure of this
     shape), not just a plausible-sounding question that happens to be mislabeled?

Reason through this separately from your relevance reasoning above, then score it
     independently using the same three anchors as relevance.
```

Listing 9: Stage-2 Correctness judge (face validity), base system prompt.

```
You are judging the face-validity of a system's response in a financial conversational-agent
     benchmark -- NOT verifying SQL or auditing the database schema, just whether the
     response looks right given the conversation.

The specific thing to judge: {correctness_prompt}

Check each of the following before deciding:
1. Addressed - does the response actually address the thing above, not something adjacent to
     it?
2. Completeness - does it cover the full extent of what's being checked, not just part of it?
3. Precision - is it specific and correct in its details, not vague or approximately right?

Reason through criteria 1-3 in order, briefly noting what you observe for each, before you
     decide a score.

Score anchors:
1 = yes -- satisfies every criterion above.
0.5 = partially -- falls short on at least one criterion (addresses some of it but not all,
     or is imprecise).
0 = no -- fails this check.
```

Evaluated agents. Listing 10 is the system prompt given to every evaluated model (Section 5). The user message lists the earlier user utterances of the same topic block, numbered, followed by the current question.

Listing 10: System prompt of the evaluated agents (Section 5).

```
You are a financial database assistant. You can query the SQLite database described below to
     answer a user's question.

For every user message, decide exactly ONE action. Check, in order:
1. Schema coverage -- does every table and column the question needs actually exist in the
     schema below, and can they actually be joined to connect what's needed (not just
     plausibly-named tables that happen to sit near each other)? If something genuinely
     needed is absent, or nothing connects it, this cannot be answered from this schema.
2. Interpretation -- is there exactly one reasonable reading of the question, given the
     schema and the conversation so far, or could a reasonable person read it in two
     genuinely different ways (an unstated time range, an ambiguous entity reference, a term
     with more than one valid meaning here)? A real ambiguity needs resolving before you can
     safely answer; a question that merely sounds informal but has one clear reading does not
     .
3. Otherwise -- the schema can answer this and there is one clear reading -- write one valid
     SQLite SQL query for it: use only tables and columns that actually appear in the schema
     below, make sure the query's logic matches what was actually asked rather than a nearby-
     sounding question, and follow any convention the question implies (a stated date range,
     a named calculation) rather than a generic default.

Reason through checks 1-3 in order before you decide, then commit to exactly one action:
- 'answer': checks 1 and 2 both pass -- write the SQL per check 3.
- 'ask': check 1 passes but check 2 does not -- ask a specific question that targets the
     actual ambiguity, not a generic request for more detail.
- 'refuse': check 1 does not pass -- briefly explain what's actually missing from the schema
     (or unjoinable), not a generic 'cannot answer'.

Respond with strict JSON only, no other text: {"reasoning": "<a few sentences that check 1-3
     in order, written BEFORE you decide>", "action": "answer"|"ask"|"refuse", "sql": "<SQL,
     or null if not answer>", "message": "<clarifying question or refusal explanation, or
     null if action is answer>"}.

Database schema:
<full-schema DDL with sample rows>
```

E Deterministic Check Catalogues and Thresholds

E.1 Stage 1, Layers 1–2: Static and Execution-Shape Checks

Section 3 summarises Stage 1's two deterministic layers; this appendix lists their checks in full. Layer 1 inspects a drafted `ground_truth_sql` before it is ever executed. *Parses* (hard gate): the SQL must be syntactically valid. *In-scope tables* (hard gate): every base table referenced (not a CTE name or alias) must be one of the turn's in-scope tables. *Ranking/counting match* (advisory): ranking or counting language in $q(t)$ ("top", "how many") must be matched by the corresponding SQL construct. *Literal-type match* (advisory): a predicate must not compare a column against a literal of an incompatible type. Layer 2 runs once the SQL has executed and inspects the shape of the real result set. *Predicate satisfaction* (hard gate): every simple `column op literal` condition in the `WHERE` clause whose column is also returned is re-evaluated against every returned row, catching results that contradict the query's own filter. *Row inflation* (advisory): whether de-duplicating the result shrinks its row count, a duplicate-row or join fan-out symptom and a common E2 pattern that Jin et al. (2026) document. *Cardinality* (advisory): whether a "top N" question returns more than N rows. An empty result is recorded as a pass, not a failure, whenever the turn was constructed to expect one (a contradiction red-team turn, for instance). Advisory checks are recorded but never block acceptance on their own, since each can raise false alarms; only a hard-gate failure triggers a repair instruction back to the generator.

E.2 Stage 2: The Deterministic Aggregate $D(t)$

Section 3 summarises the deterministic checks $D(t)$ aggregates for an answerable turn; this appendix lists all seven. *Execution*: the gold SQL $s(t)$ runs without error. *Test cases*: the fraction of the turn's `test_cases[]` assertions that hold against the real executed result (partial credit, not binary), using the assertion vocabulary of Table 16. *Formula check*: when the utterance invokes a recognised analytical pattern (average, year-over-year growth, ratio, weighted average), the SQL must implement that pattern's closed form, not an approximation of it. *Domain-term check*: a named financial term in the utterance (yield, duration, exposure) must be grounded in a column the SQL actually references. *Comparison-operator check*: a comparison phrase in the utterance ("greater than", "at least", "between") must match the operator the SQL applies. *Result-count check*: an explicit count in the utterance ("top 5", "the three largest") must match the SQL's `LIMIT`. *Result-shape check*: the utterance's implied shape (scalar for a KPI-style question, grouped for a by-category question) must match the executed result's actual shape. Each check applies only when its triggering condition holds (execution always applies); $D(t)$ is the mean of whichever checks apply, so a turn is never penalised for a check that was irrelevant to what was asked. $D(t)$ is then blended with the judge's face-validity score $J_{\text{corr}}(t)$ as in equation 6.

Table 16: Representative entries from the `test_cases[]` assertion vocabulary (`test_case_vocab.json`).

Assertion	Meaning and example
`all_gt`	Every row's value in `col` exceeds `value`. {`col: notional_volume, op: gt, value: 10000000`}
`all_between`	Every row's value in `col` falls within [lower, upper]. {`col: fraud_rate, lower: 0, upper: 1`}
`row_count == N`	The result has exactly N rows.
`null_result`	The result is expected to be empty (e.g. a contradiction red-team turn).
`group_by_present`	The result is grouped by the expected column.

Thresholds. The publish and review thresholds of Section 3, $S(C) \geq 0.80$ and $S(C) \geq 0.50$, are fixed constants of the released scorer and were applied unchanged to all seven batches. Stage 3 (Section 3) measures agreement on the turn scores behind $S(C)$; it does not tune the cut points or change Stage 2.

F EVALUATION METRICS: DETAILS AND FULL RESULTS

This appendix complements the metric definitions of equation 7 with the details they leave implicit, and reports the full results behind Section 5. All notation is that of Section 5.

Unparsable responses. A response that does not name one of ANS, ASK, or REF, including a failed call, has $\hat{\rho}(t) = \perp$. It never matches gold, so it lowers AA and Pass Rate, and it counts as an error in neither F-Conf nor F-Caut. An answered turn with no SQL has $F(t) = 0$.

Success Rate and Pass Rate. SR is a strict pass/fail judgement per conversation, so one wrong action or one failed red-team variant fails the whole conversation. It therefore compounds with length: a system independently correct on 90% of turns completes only $0.9^{12} \approx 28\%$ of 12-turn conversations, close to the test split's mean length of 12.3 turns. Pass Rate averages the same per-turn criterion instead, and is the diagnostic to read when SR is 0%. SR^{clean} replaces $\text{Pass}(t)$ by $\text{ok}(t)$. The released harness also reports SR by bucket $\mathcal{B}(C)$, database, and vertical; on the test split all of these are 0%, as is SR^{clean}. The Block Success Rate of Table 5 applies the same criterion to each topic block.

Models. Table 17 lists every model used in the paper. All were served through an OpenAI-compatible endpoint.

Table 17: Models used in FINCH 2.0. Roles: *Generator* writes and redrafts the conversations (Section 2); *Evaluator* judges the data, as the SAR-Verifier in Stage 1 or the LLM judge in Stage 2 (Section 3); *Evaluated* is a model whose actions and SQL are scored on the test split (Section 5). Parameter counts are as stated on each model card; precision is that of the released checkpoint. Each identifier links to its Hugging Face page, `https://huggingface.co/<identifier>`.

Identifier	Role	Params (total / active)	Precision	Architecture
`openai/gpt-oss-120b`	Generator; Evaluator (SAR-Verifier)	117B / 5.1B	MXFP4 experts	MoE, 128 experts, 4 per token
`google/gemma-4-31B-it`	Evaluator (Stage 2 judge); Evaluated	30.7B / dense	BF16	Dense Transformer
`Qwen/Qwen3.8-27B`	Evaluated	27B / dense	BF16	Dense, hybrid linear and full attention
`nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-BF16`	Evaluated	120B / 12B	BF16	Hybrid Mamba-Transformer MoE

Harness and re-runs. The original runs used a budget of 1,024 output tokens. Reasoning models often spent it on their reasoning trace, so the gateway returned an empty message or JSON cut off mid-string. Every such turn was re-run once with 8,192 tokens and the same prompt, context, temperature, and scorer: 177 turns for GEMMA-4-31B-IT, 17,616 for QWEN3.8-27B, and 12,495 for NEMOTRON-3-SUPER. Afterwards 2, 10, and 7 of the 41,211 rows per model remain unparsable; 40,886 rows are scored after the repairs. Because the larger budget was given only on failure, QWEN3.8-27B received it on 42.7% of its rows and GEMMA-4-31B-IT on 0.4%. Turns affected by the release repairs (Appendix H) are excluded, with their variants.

Robustness delta. Δ_r pairs each answerable base turn with each of its variants and scores an unanswered turn as $F = 0$, so $|T_r|$ is the same for every model (Table 19). The test split's red-team variants cover paraphrase chains (R-1), schema perturbation (R-2), and ambiguity cascades (R-3).

Confidence intervals. Table 20 gives 95% intervals from 1,000 bootstrap resamples of test conversations. With the 925 incomplete-gold turns included, $\bar{F}$ is 0.432, 0.441, and 0.417 and Exec 30.2%, 28.9%, and 29.0%.

Full confusion matrices. Table 18 gives, for each model, the fraction of all scored test turns in each cell, including the unparsable column $\perp$.

Table 18: Confusion matrices on the test split (rows gold, columns predicted; fraction of all scored turns).

	Gemma-4-31B-it				Qwen3.8-27B				Nemotron-3-Super			
Gold	ANS	ASK	REF	⊥	ANS	ASK	REF	⊥	ANS	ASK	REF	⊥
ANS	.8788	.0206	.0405	.0000	.7949	.0883	.0566	.0002	.7124	.1936	.0339	.0001
ASK	.0113	.0071	.0019	.0000	.0083	.0100	.0021	.0000	.0066	.0117	.0020	.0000
REF	.0023	.0001	.0373	.0000	.0013	.0006	.0378	.0000	.0029	.0012	.0356	.0000

Table 19: Robustness delta Δ_r by attack type with 95% bootstrap intervals (positive means worse under attack); $|T_r|$ counts base and variant pairs.

Attack type	$\lvert T_r \rvert$	**Gemma-4-31B-it**	**Qwen3.8-27B**	**Nemotron-3-Super**
Paraphrase chain (R-1)	5,446	0.008 [-0.006, 0.022]	0.019 [0.005, 0.033]	-0.000 [-0.015, 0.014]
Schema perturbation (R-2)	2,236	0.108 [0.093, 0.124]	0.121 [0.106, 0.136]	0.093 [0.075, 0.110]
Ambiguity cascade (R-3)	3,478	0.111 [0.097, 0.127]	0.096 [0.079, 0.112]	0.106 [0.088, 0.122]

Table 20: Headline metrics with 95% bootstrap intervals over test conversations.

Metric	**Gemma-4-31B-it**	**Qwen3.8-27B**	**Nemotron-3-Super**	**Text only**
AA	92.3 [92.0, 92.7]	84.3 [83.8, 84.8]	76.0 [75.4, 76.5]	97.5 [97.3, 97.8]
Macro-F1	0.626 [0.613, 0.639]	0.545 [0.534, 0.555]	0.534 [0.523, 0.543]	0.809 [0.795, 0.824]
Recall, ASK	34.9 [31.2, 38.7]	49.0 [44.7, 52.8]	57.7 [53.6, 61.3]	52.7 [48.7, 57.0]
Recall, REFUSE	93.8 [92.4, 95.1]	95.3 [94.1, 96.5]	89.5 [87.7, 91.2]	74.1 [71.4, 76.8]
$\bar{F}$	0.441 [0.435, 0.448]	0.450 [0.444, 0.456]	0.427 [0.420, 0.433]	0.000 [0.000, 0.000]
Exec	31.2 [30.5, 31.9]	29.8 [29.1, 30.5]	30.0 [29.3, 30.6]	0.0 [0.0, 0.0]
Pass Rate	34.2 [33.6, 34.9]	35.4 [34.8, 36.0]	34.9 [34.3, 35.6]	0.0 [0.0, 0.0]

Schema perturbation and ambiguity cascades are the damaging attacks for every model; paraphrase chains produce no drop distinguishable from zero.

Decomposing Δ_r. Table 21 splits Δ_r into pairs where the model answered the base turn but declined the variant (Decl.), pairs where it answered both (SQL), and the remainder (base declined), which is negative and, for R-1, larger than Δ_r itself; the last column is the share of variants declined.

Table 21: Decomposition of Δ_r; Decl. and SQL sum to Δ_r up to the remainder from declined base turns.

	Gemma-4-31B-it				Qwen3.8-27B				Nemotron-3-Super			
	Δ_r	Decl.	SQL	%dec.	Δ_r	Decl.	SQL	%dec.	Δ_r	Decl.	SQL	%dec.
R-1	0.008	0.007	0.013	6.5	0.019	0.018	0.032	15.0	-0.000	0.025	0.024	18.0
R-2	0.108	0.045	0.067	13.9	0.121	0.076	0.056	25.9	0.093	0.069	0.050	30.2
R-3	0.111	0.066	0.050	19.0	0.096	0.063	0.052	23.5	0.106	0.089	0.042	33.9

Text-hard subset. Table 22 splits the test turns by whether the utterance-only classifier predicts the gold action. The 722 turns it gets wrong comprise 283 ASK, 302 REFUSE, and 137 ANSWER turns. Its top cues are, for ASK, *significant*, *recently*, *all*, *large*, *but not*, *aging*; for REFUSE, *esg*, *inflation*, *sentiment*, *weather*, *carbon*, *gdp*.

Table 22: Per-action recall (%) on turns the text-only classifier gets right (easy) and wrong (hard).

	ASK		REFUSE		ANSWER	
Model	easy	hard	easy	hard	easy	hard
Gemma-4-31B-it	48.3	20.1	96.5	86.1	93.6	73.7
Qwen3.8-27B	61.6	35.0	97.6	88.7	84.7	57.7
Nemotron-3-Super	65.7	48.8	93.1	79.5	75.9	48.2

Table 23: $\bar{F}$ by turn property with 95% bootstrap intervals over conversations. †Incomplete gold (Section 5).

	n	**Gemma-4-31B-it**	**Qwen3.8-27B**	**Nemotron-3-Super**
Easy	11,018	0.569 [0.559, 0.579]	0.554 [0.544, 0.563]	0.528 [0.519, 0.538]
Medium	4,747	0.513 [0.499, 0.526]	0.495 [0.482, 0.510]	0.457 [0.443, 0.472]
Hard	10,933	0.282 [0.273, 0.290]	0.327 [0.318, 0.335]	0.311 [0.303, 0.320]
Q-1	20,237	0.460 [0.452, 0.466]	0.461 [0.454, 0.468]	0.439 [0.433, 0.447]
Q-2	446	0.538 [0.497, 0.575]	0.579 [0.535, 0.615]	0.535 [0.494, 0.575]
Q-3†	487	0.191 [0.164, 0.220]	0.192 [0.163, 0.222]	0.172 [0.144, 0.206]
Q-4	501	0.318 [0.281, 0.356]	0.372 [0.333, 0.408]	0.337 [0.298, 0.376]
Q-5†	438	0.154 [0.128, 0.180]	0.173 [0.145, 0.202]	0.132 [0.107, 0.158]
none (no M)	21,184	0.458 [0.450, 0.465]	0.461 [0.454, 0.468]	0.439 [0.432, 0.446]
M-1	759	0.341 [0.307, 0.379]	0.391 [0.355, 0.432]	0.378 [0.342, 0.415]
M-2	822	0.366 [0.328, 0.402]	0.409 [0.374, 0.446]	0.381 [0.344, 0.422]
M-3	167	0.247 [0.175, 0.331]	0.230 [0.161, 0.304]	0.216 [0.156, 0.284]
M-4	644	0.369 [0.327, 0.408]	0.374 [0.334, 0.412]	0.351 [0.310, 0.393]
M-5	1,774	0.418 [0.392, 0.444]	0.447 [0.422, 0.471]	0.425 [0.398, 0.447]
M-6	659	0.447 [0.406, 0.489]	0.481 [0.438, 0.523]	0.448 [0.402, 0.493]
M-8	689	0.299 [0.263, 0.334]	0.343 [0.305, 0.382]	0.257 [0.220, 0.294]

G PER-DATABASE EXPERIMENTAL BREAKDOWNS

This appendix reports per-database human-vs-Stage 2 agreement, which Section 3 reports only in aggregate, and the turn-level phenomenon coverage behind Section 4.2.

Human-vs-automated agreement: full metric breakdown. Table 3 reports pooled figures; Tables 24 and 25 below report the complete set of agreement metrics (raw and discretised MAE, weighted Cohen's κ, Krippendorff's α, and Spearman's r_s) for Relevance and Correctness respectively, for each of the 16 databases. After the *cannot say* labels are dropped (Appendix C), n differs between the two tables only for PRISM (72 for Relevance, 71 for Correctness).

Table 24: Human-vs-automated agreement per database: Relevance.

Database	n	**MAE (raw)**	**MAE (disc.)**	**Weighted** κ	**Krippendorff** α	**Spearman** r_s
apex	95	0.105	0.105	0.415	0.530	0.566
atlas	96	0.133	0.141	0.383	0.478	0.574
citadel	92	0.141	0.147	0.394	0.575	0.596
compass	99	0.114	0.106	0.507	0.611	0.665
forex	100	0.022	0.020	0.897	0.963	0.939
graviton	77	0.068	0.078	0.685	0.738	0.773
horizon	76	0.076	0.059	0.779	0.839	0.798
meridian	71	0.123	0.120	0.657	0.746	0.735
nexus	76	0.125	0.112	0.559	0.588	0.548
prism	72	0.104	0.104	0.675	0.726	0.732
pulse	75	0.070	0.073	0.645	0.714	0.733
quasar	70	0.082	0.079	0.687	0.685	0.739
sentinel	73	0.116	0.110	0.496	0.553	0.562
stratum	70	0.086	0.093	0.389	0.539	0.649
terra	72	0.111	0.118	0.588	0.646	0.700
vertex	70	0.064	0.079	0.685	0.803	0.840
Pooled (n=1,284)		**0.097**	**0.097**	**0.598**	**0.675**	**0.691**

Table 25: Human-vs-automated agreement per database: Correctness.

Database	n	**MAE (raw)**	**MAE (disc.)**	**Weighted** κ	**Krippendorff** α	**Spearman** r_s
apex	95	0.119	0.089	0.332	0.374	0.303
atlas	96	0.087	0.068	0.429	0.428	0.509
citadel	92	0.109	0.082	0.530	0.572	0.397
compass	99	0.131	0.101	0.308	0.319	0.284
forex	100	0.115	0.080	0.403	0.376	0.437
graviton	77	0.106	0.123	0.204	0.246	0.619
horizon	76	0.081	0.066	0.583	0.589	0.569
meridian	71	0.117	0.091	0.455	0.598	0.484
nexus	76	0.105	0.086	0.543	0.619	0.604
prism	71	0.097	0.077	0.538	0.693	0.550
pulse	75	0.103	0.060	0.656	0.687	0.494
quasar	70	0.107	0.079	0.507	0.527	0.480
sentinel	73	0.058	0.041	0.597	0.757	0.579
stratum	70	0.086	0.064	0.405	0.402	0.286
terra	72	0.111	0.090	0.518	0.691	0.601
vertex	70	0.083	0.057	0.639	0.733	0.665
Pooled (n=1,283)		**0.102**	**0.079**	**0.475**	**0.533**	**0.469**

Direction of disagreement. The metrics above measure how far apart human and automated scores are, not which side is higher when they disagree. Table 26 classifies every turn, after mapping the automated score to the nearest value on the human's $\{0, 0.5, 1\}$ scale, as agree, human-favourable (H. Fav.: the human rated it higher, so the pipeline was too strict), or automated-favourable (A. Fav.: the pipeline rated it higher, so it was too lenient). For Relevance, human-favourable disagreements outnumber automated-favourable ones in 14 of the 16 databases and tie in the other two (MERIDIAN, PULSE), consistent with a conservative validator. For Correctness the asymmetry is weaker: it reverses in six databases (APEX, GRAVITON, HORIZON, PRISM, PULSE, TERRA), where the pipeline is, if anything, too lenient, and ties in one (SENTINEL).

Table 26: Direction of human-vs-automated disagreement, per database. H. Fav. = human rated the turn higher (pipeline too strict); A. Fav. = automated rated it higher (pipeline too lenient). These are the E↑ and S↑ columns of Table 3.

	Relevance			Correctness		
Database	**Agree**	**H. Fav.**	**A. Fav.**	**Agree**	**H. Fav.**	**A. Fav.**
apex	83.2%	14.7%	2.1%	84.2%	6.3%	9.5%
atlas	77.1%	19.8%	3.1%	87.5%	9.4%	3.1%
citadel	75.0%	20.7%	4.3%	85.9%	7.6%	6.5%
compass	80.8%	16.2%	3.0%	79.8%	14.1%	6.1%
forex	96.0%	3.0%	1.0%	84.0%	16.0%	0.0%
graviton	84.4%	9.1%	6.5%	77.9%	5.2%	16.9%
horizon	88.2%	9.2%	2.6%	86.8%	2.6%	10.5%
meridian	80.3%	9.9%	9.9%	81.7%	9.9%	8.5%
nexus	80.3%	14.5%	5.3%	82.9%	11.8%	5.3%
prism	81.9%	15.3%	2.8%	88.7%	2.8%	8.5%
pulse	86.7%	6.7%	6.7%	88.0%	5.3%	6.7%
quasar	88.6%	11.4%	0.0%	84.3%	14.3%	1.4%
sentinel	83.6%	15.1%	1.4%	91.8%	4.1%	4.1%
stratum	84.3%	15.7%	0.0%	87.1%	10.0%	2.9%
terra	80.6%	15.3%	4.2%	84.7%	4.2%	11.1%
vertex	85.7%	14.3%	0.0%	90.0%	5.7%	4.3%
Pooled	**83.5%**	**13.2%**	**3.3%**	**85.2%**	**8.3%**	**6.5%**

Turn-phenomenon coverage. Of the 176,860 turns of Section 4, 95,434 (54.0%) exercise M-series multi-turn dependency (P1), 26,632 (15.1%) A-series ambiguity or unanswerability (P4), 14,249 (8.1%) Q-2 to Q-5 compound structure (P7), and 36,111 (20.4%) carry at least one red-team variant (P6).

H DATASHEET

This appendix follows the datasheet template of Gebru et al. (2021).

Motivation. FINCH 2.0 evaluates whether conversational text-to-SQL systems for finance take the right action at each turn (answer, ask for clarification, or refuse) and write correct SQL when they answer (Section 1). Creators and funding are withheld for anonymous review.

Composition. The corpus has 16 SQLite databases in 7 financial verticals (469 tables, 3,674 columns) and 13,024 conversations with 36,106 topic blocks and 176,860 turns, about 17.7M tokens under the `cl100k_base` tokenizer. Of the turns, 158,118 (89.4%) are answerable with gold SQL, 7,390 (4.2%) need clarification, and 11,352 (6.4%) are unanswerable because the schema lacks the requested information. Each turn carries its protocol tags, measured complexity tier, Stage-2 scores, and, for answerable turns, the SAR-Verifier verdict; 1,287 turns also carry expert labels, 368 of them in the test split. Table 27 shows that turns are spread evenly across each database's topics. All data are synthetic, and no real customer, account, or trade records are included.

Table 27: Per-database coverage: topic clusters used, turns, and topic evenness (Shannon entropy of topic shares normalised by its maximum; 1 means perfectly even).

Database	Vertical	Topics	Turns	Evenness
APEX	Capital Markets	7	9,868	0.980
ATLAS	Insurance	5	10,225	0.969
CITADEL	Banking	5	10,052	0.998
COMPASS	Banking	8	10,143	0.997
FOREX	Capital Markets	5	7,423	0.972
GRAVITON	Capital Markets	7	10,070	0.965
HORIZON	Fin. Reporting & Treasury	5	9,971	0.999
MERIDIAN	Investment Management	7	9,931	0.982
NEXUS	Capital Markets	4	12,864	0.999
PRISM	Fin. Reporting & Treasury	5	12,945	0.999
PULSE	Banking	4	11,618	0.999
QUASAR	Investment Management	5	12,735	0.998
SENTINEL	Risk & Compliance	4	12,783	0.999
STRATUM	Investment Management	7	10,413	0.988
TERRA	Alt. & Private-Market Inv.	4	12,788	1.000
VERTEX	Alt. & Private-Market Inv.	4	13,031	0.999
Total		**86**	**176,860**	

Table 28 gives the distribution of turns by protocol series.

Series	Turns	%
M	70,194	39.7
A	12,155	6.9
Q	62,429	35.3
M+A	6,892	3.9
M+Q	17,602	10.0
A+Q	6,839	3.9
M+A+Q	746	0.4

Table 28: Turns by M/A/Q series. Unlike $\mathrm{fam}(t)$ in Eq. 5, M here includes the topic switch M-7 and Q the atomic label Q-1, so multi-series rows do not violate that equation.

Splits. The test split holds the 2,400 highest-scoring conversations of the second batch by $S(C)$ and is reserved for evaluation (Section 5); the other 10,624 conversations form the training data.

Known issues. The 810 conversations flagged for review are included. SAR-Verifier verdicts are advisory, so some accepted turns carry an `incorrect` verdict, which users can filter on. Stage-2 Correctness scores agree only moderately with experts ($\alpha = 0.53$; Section 3). Schemas fall below the column- and foreign-key-density targets, and only one schema's declared-key join graph is fully connected (Definition 6). 24.4% of answerable turns have no SAR verdict. All conversations are in English.

Repairs before release. Four defects found after validation are repaired in the released copy by a script that leaves the generated sessions untouched. (i) 699 gold queries (0.44%) anchored on the wall clock, for example through `date('now')`, against the generator's instructions; each anchor is replaced by the latest date in the event-date columns the query references, and the query is re-executed. The 483 that then pass all their test cases are kept, and the other 216 are excluded. (ii) 74 answerable gold queries are constant stubs that reference no table, and are excluded. (iii) Tier labels are removed from the 2,045 non-answerable turns that carried one, 321 of them non-canonical. (iv) Every conversation receives a unique identifier from its batch and chat identifier, since 1,122 chat identifiers recur across batches 4 to 7. Excluded turns stay in their conversations as context but are not scored. In the test split, the 215 affected answerable turns and their red-team variants are left out of every reported metric, because the baselines were run before the repair.

Collection process. Conversations were generated in seven batches by the seeded planner and the `openai/gpt-oss-120b` generator of Section 2; 14,400 were generated and 13,024 accepted (Table 2), at an estimated $0.011 per question-SQL pair.

Preprocessing and labelling. Every conversation passed through the Validation Funnel (Section 3). Protocol tags and complexity tiers are computed deterministically from the stored session, and Stage-2 scores come from deterministic checks and `google/gemma-4-31B-it` judges whose prompts are in Appendix D.

Uses. FINCH 2.0 is intended for benchmarking conversational text-to-SQL systems and for studying when they answer, ask, or refuse. It is not intended to certify production financial systems, and the test split should not be used for training.

Distribution and maintenance. The dataset, the generation and validation code, and the agent evaluation framework will be released upon acceptance, with errata and new versions tracked in the release repository. A leaderboard on the test split is planned after acceptance.