Beyond Tables: Doc2DB-Bench for Relationally Faithful Document-to-Database Construction

Anonymous authors
Paper under double-blind review

Abstract

Practical AI systems increasingly need to turn long, heterogeneous documents into queryable relational databases, not isolated spreadsheets. In domains such as finance, healthcare, education, transportation, and enterprise operations, downstream workflows rely on normalized schemas, entity identities, keys, cross-table relationships, and integrity constraints for analytics, compliance, auditing, and SQL-backed decision making. Existing Document-to-Table benchmarks are insufficient for this setting: flattening evidence into single tables can duplicate entities, obscure many-to-many relationships, create sparse records, and avoid testing whether extracted facts form a valid database instance. This creates an urgent need to evaluate document understanding as database construction rather than field extraction. We introduce **Doc2DB-Bench**, a Document-to-Database benchmark comprising 203 long-document instances across 42 schemas and seven domain groups, with 117 entity tables, 132 relationship tables, 7,341 rows, and 41,935 cells. Built through a controllable DB-to-Doc synthesis pipeline and organized by a taxonomy of intra-table extraction and inter-table reasoning, the generated documents undergo authenticity verification, proving indistinguishable from real-world references. Doc2DB-Bench thus provides a testbed for reliable, auditable, and relationally faithful LLM-based data systems. The benchmark is available at `https://anonymous.4open.science/r/Doc2DB-Bench-3EEF`.

1 Introduction

Modern organizations rely on documents as the primary carrier of operational knowledge, from clinical notes and commercial contracts to financial reports and enterprise records (Muscolino et al., 2023; Chen et al., 2023; Li et al., 2026). Yet downstream applications rarely consume free-form text directly: analytics pipelines, Business Intelligence dashboards, compliance workflows, and SQL-backed systems require structured, queryable, and auditable data (LlamaIndex, 2024; Zhang et al., 2025; Zeng et al., 2026). As large language models become increasingly capable of processing long and heterogeneous documents, converting document evidence into reliable structured data has become a central goal of information extraction (Xu et al., 2024; Liang et al., 2026; Li et al., 2025).

Document-to-Table and Its Limitations. Most existing benchmarks study this problem under the *Document-to-Table* (Doc2Table) setting, where systems extract fields or populate a single flattened table (ZJU-DAILY, 2026; Deng et al., 2024; Kashyap et al., 2025; Zhang et al., 2026). This setting is useful for isolated record extraction, but it is insufficient when downstream workflows require relational databases. Flattening multi-entity evidence can duplicate entities, obscure many-to-many relationships, introduce sparse records with excessive `NULL` values, and avoid testing whether extracted facts satisfy schema-level constraints.

Example 1 *Consider a financial report in Fig. 1(a) describing multiple companies, their cash positions across fiscal years, and investment relationships among them. The target schema in Fig. 1(b) contains an entity table* `Company` *and a self-referencing relationship table* `Hold`*, where company identifiers are system-generated keys and do not appear in the text.*
Constructing the database requires more than extracting local values. An LLM extractor may correctly identify many company mentions and numbers, but still produce semantically invalid tuples. For entity extraction (Fig. 1(c)), it may misinterpret units, such as treating "18,524 (in thousand)" as a raw value, or materialize weak mentions such as Coyni Inc *as incomplete duplicate entities. For relationship extraction (Fig. 1(d)), surface-level cues may induce spurious mutual*

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.

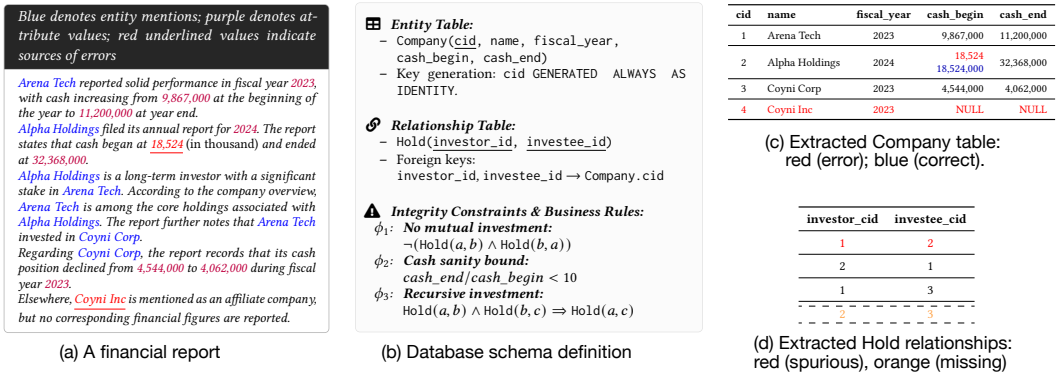


Figure 1: Document-to-Database extraction: (a) document(s); (b) target database schema; (c) Extracted Entity Tables; (d) Extracted Relationship Tables.

investments, while implicit multi-hop links may be missed. These errors show that local extraction decisions must be reconciled with schema-level semantics and integrity constraints. □

From Extraction to Database Construction. This motivates **Document-to-Database** (Doc2DB): given a document collection F, a target schema E, and integrity or business constraints Σ, the goal is to construct a relational database instance D that is both faithful to the documents and valid under the schema: $(F,\ E,\ \Sigma) \xrightarrow{\text{Doc2DB}} D$. Unlike Doc2Table, Doc2DB entails database-level reasoning over implicit entity identities, dispersed relational evidence, and global constraints (Dong et al., 2014; Sa et al., 2016; Peng et al., 2017; Shin et al., 2015). Evaluation must therefore assess not only value extraction but also entity alignment, relationship construction, and relational validity.

Despite its practical importance, Doc2DB remains underexplored in benchmarks. Existing information extraction benchmarks, including Rotowire (Wiseman et al., 2017), E2E (Novikova et al., 2017), LiveSum (Deng et al., 2024), InstructIE (Jiao et al., 2023), StructText (Kashyap et al., 2025), and DT-Bench (ZJU-DAILY, 2026), mainly target flat tables, single-table extraction, or simplified generation. SQUiD (Sadia et al., 2025) explores text-to-relational database generation, but mainly targets logical relational view recovery rather than realistic Doc2DB construction. As summarized in Table 1, they do not fully evaluate cross-table schema construction (i.e., normalized multi-table structures with inter-table dependencies), long-context evidence aggregation, and database-level correctness.

Challenges. Building a comprehensive Doc2DB benchmark is challenging for two reasons. First, direct human annotation is difficult to scale: annotators must collect documents, define schemas, and manually construct ground-truth entity and relationship tables. Second, realistic Doc2DB instances must cover diverse reasoning requirements, including unit normalization, ambiguity resolution, multi-hop relation construction, temporal changes, and constraint satisfaction, while also spanning domains with different narrative styles and schema structures.

Our Proposal. We present **Doc2DB-Bench**, a benchmark for evaluating Doc2DB capabilities beyond flat table extraction. To avoid the scalability bottleneck of manual annotation, we design a controllable DB2Doc reverse synthesis pipeline grounded in existing relational databases such as BIRD (Li et al., 2023) and Spider (Yu et al., 2018). Starting from schemas and database instances, the pipeline decomposes tuples into atomic evidence, assigns capability labels, serializes evidence into document plans, generates style-conditioned documents, and validates the generated documents through coverage and extraction-consistency checks.

To model realistic document complexity, Doc2DB-Bench introduces a two-pillar Doc2DB taxonomy. *Intra-Table Capabilities* cover cell-level extraction, normalization, inference, disambiguation, and faithfulness. *Inter-Table Capabilities* capture database-specific reasoning, including identity resolution, relationship linking, multi-hop composition, dynamic change, and integrity constraints. This taxonomy supports controlled benchmark construction and fine-grained diagnosis of model failures. Finally, Doc2DB-Bench evaluates outputs at the database level by measuring both entity integrity and relational fidelity, thereby testing whether extracted facts form a schema-conformant, queryable, and relationally faithful database instance.

Contributions. We summarize our contributions as follows:

1. **Doc2DB Capability Taxonomy.** We introduce a two-pillar taxonomy of the key capabilities required for Doc2DB construction, spanning *Intra-Table* value extraction and *Inter-Table* relational reasoning, to guide benchmark design and support fine-grained evaluation.

Table 1: Comparison of existing benchmarks and our Doc2DB-Bench.

Benchmark	Single-Table Extraction (ST)	Multi-Domain (MD)	Cross-Table Reasoning (CR)	Long-Context (LC)	DB-Level Eval (DC)
Rotowire (Wiseman et al., 2017)	✓	✗	✗	✗	✗
E2E (Novikova et al., 2017)	✓	✗	✗	✗	✗
LiveSum (Deng et al., 2024)	✓	✗	✗	✗	✗
InstructIE (Jiao et al., 2023)	✓	✓	✗	✗	✗
StructText (Kashyap et al., 2025)	✓	✓	✗	✗	✗
DTBench (ZJU-DAILY, 2026)	✓	✓	✗	✓	✗
SQUiD (Sadia et al., 2025)	✓	✓	✗	✗	✓
`Doc2DB-Bench` (ours)	✓	✓	✓	✓	✓

2. **Controllable DB2Doc Synthesis Pipeline.** We propose a reverse-synthesis pipeline that generates long-document Doc2DB instances from real relational schemas and database instances, reducing manual annotation cost while controlling document complexity.
3. **The Doc2DB-Bench Benchmark.** We construct a multi-domain benchmark for evaluating LLMs on relationally faithful Doc2DB tasks, going beyond flat, single-table extraction benchmarks.
4. **Extensive Experiments.** We evaluate a broad range of mainstream LLMs on Doc2DB-Bench and conduct in-depth analyses across entity- and relation-level capabilities, revealing their strengths, limitations, and open research opportunities for Doc2DB extraction.

2 Document-to-Database: The Problem

2.1 Problem Statement

Database Specification. We formalize the target database specification as $\mathcal{S} = (E, \Sigma)$. The entity–relationship schema $E = (\mathcal{E}, \mathcal{R})$ comprises entity tables $\mathcal{E}$ and relationship tables $\mathcal{R}$. Let $\mathcal{T} = \mathcal{E} \cup \mathcal{R}$ denote all tables; each $T \in \mathcal{T}$ has an attribute set A_T and a primary key $K_T \subseteq A_T$. The constraint set $\Sigma = \Sigma_{\text{rel}} \cup \Sigma_{\text{app}}$ combines standard relational constraints (*e.g.,* type, domain, and inclusion constraints) with application-specific rules (*e.g.,* temporal conditions).

Document Corpus. We consider a collection of heterogeneous documents $F = \{f_1, f_2, \ldots, f_m\}$ that provide the evidence for instantiating the target schema. We assume low-level processing, such as text extraction and layout analysis, has been completed. Since real-world documents may contain incomplete information, unobserved attributes naturally remain `NULL`. Accordingly, every extracted value should be grounded in the source documents without hallucinating unsupported information.

Definition 1 (The Doc2DB Problem). Given a database specification (E, Σ) and a document corpus $F = \{f_1, \ldots, f_m\}$, the Doc2DB task is to construct a relational database instance $(F, E, \Sigma) \xrightarrow{Doc2DB} D$, where $D = \{I_x \mid x \in E\}$ is the collection of instantiated entity and relationship tables. The target instance D should satisfy (1) *Constraint satisfaction*: $D \models \Sigma$; and (2) *Closeness to ground truth*: D is as close as possible to the ground-truth database D^*.

2.2 DB2Doc Synthesis Paradigm

To construct a benchmark for evaluating Doc2DB extraction, we adopt a reverse synthesis perspective: given a ground-truth database instance $D = I_x \mid x \in E$ with specification $\mathcal{S} = (E, \Sigma)$, we synthesize a source document corpus F from which D^* can be faithfully recovered.

Definition 2 (DB2Doc Synthesis). Given a ground-truth database D^*, its specification $\mathcal{S}$, and a capability taxonomy $\mathcal{H}$, DB2Doc synthesizes a document corpus F satisfying three properties: (1) *completeness*, every ground-truth value and relational tuple in D^* is supported by evidence in F; (2) *exclusiveness*, F contains no evidence supporting database facts beyond D^* under $\mathcal{S}$; and (3) *capability awareness*, the evidence for each target value or tuple is constructed to require specific reasoning capabilities from $\mathcal{H}$ for its recovery, with $\mathcal{H}$ formalized as a two-pillar taxonomy in Sec. 3.

The completeness and exclusiveness conditions jointly ensure that D^* is the unique ground-truth database recoverable from F under $\mathcal{S}$, while capability awareness enables fine-grained evaluation across taxonomy levels. Each instance may exercise any subset of capabilities from $\mathcal{H}$, reflecting the naturally uneven distribution of requirements across real-world documents.

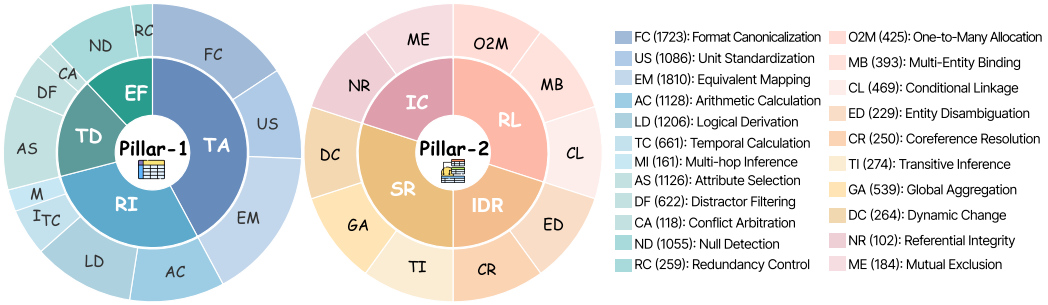


Figure 2: A Taxonomy of Doc2DB Extraction Capabilities.

3 Doc2DB Taxonomy

Doc2DB requires models to construct both values and relations. Accordingly, we organize the required capabilities into two complementary pillars. Intra-Table Capabilities (Pillar 1) asks *what values should populate an entity record*, whereas Inter-Table (Pillar 2) asks *how entity records should be connected*. Figure 2 summarizes the resulting taxonomy and its fine-grained sub-capabilities.

3.1 Pillar1: Intra-Table Capabilities

Pillar 1 focuses on what values should populate an entity record, covering four complementary challenges. **Transformative Alignment (TA).** Normalizes observed values into schema-compatible representations (*e.g.,* "18,524 in thousands" → 18,524,000). **Reasoning & Inference (RI).** Derives values that are not explicitly stated (*e.g.,* computing a target value from several reported quantities). **Target Discrimination (TD).** Identifies the intended value among distracting or conflicting evidence (*e.g.,* selecting the current-year rather than a historical figure). **Evidence Faithfulness (EF).** Avoids unsupported outputs (*e.g.,* keeping an unreported attribute as `NULL`). Pillar 1 extends DTBench (ZJU-DAILY, 2026) from single-table extraction to schema-compliant entity construction in Doc2DB.

3.2 Pillar 2: Inter-Table Capabilities

Pillar 2 focuses on how entity records should be connected, covering four complementary challenges. **Identity Resolution (IDR).** Resolves ambiguous mentions to unique database entities (*e.g.,* identifying "Lao Zhang" as the employee in the technical department among records with the same name). **Relationship Linking (RL).** Maps entity mentions and interactions into normalized relational tuples (*e.g.,* expanding "Alice manages X, Y, and Z" into three separate relationship rows). **Structural Reasoning (SR).** Infers relations from distributed or compositional evidence (*e.g.,* "Alice leads the Mobile Team" and "the Mobile Team manages Project X" imply (Alice, Project X)). **Integrity Constraint (IC).** Suppresses relations that violate negative evidence or explicit constraints (*e.g.,* not linking Bob to Project X when the document states that all managers *except Bob* are assigned to it). Detailed sub-capabilities and examples are provided in Appendix B.

4 Document-to-Database (Db2Doc) Synthesis

Given a ground-truth relational database instance D, its specification $S = (E, \Sigma)$, and the capability taxonomy $\mathcal{H}$, our goal is to synthesize a document corpus F from which D can be faithfully recovered. As shown in Fig. 3, our DB2Doc pipeline consists of five stages: capability assignment, evidence decomposition, strategic serialization, iterative document generation, and dual validation.

The rationale behind this design is to separate *what* should be tested from *how* it is expressed in documents. Capability assignment determines the target reasoning requirements; evidence decomposition grounds each target value or tuple in atomic evidence; serialization controls the document structure; generation turns the evidence into realistic long-form text; and validation checks that the final document remains complete, faithful, and recoverable. This staged design makes the synthesis process controllable, reproducible, and suitable for fine-grained Doc2DB evaluation.

Step 1: Capability Assignment. A *Labeling Agent* scans each cell $v_{ij} \in D$ and assigns a label set $\mathcal{L}ij \subseteq \mathcal{H}$ to produce an annotated matrix $\mathbf{M}_D$. The agent employs an LLM as a semantic router that jointly conditions on the cell value vij, its attribute context from $\mathcal{S}$, and the enclosing relational tuple r, reasoning over both intra-table semantics and inter-table relational structure to determine

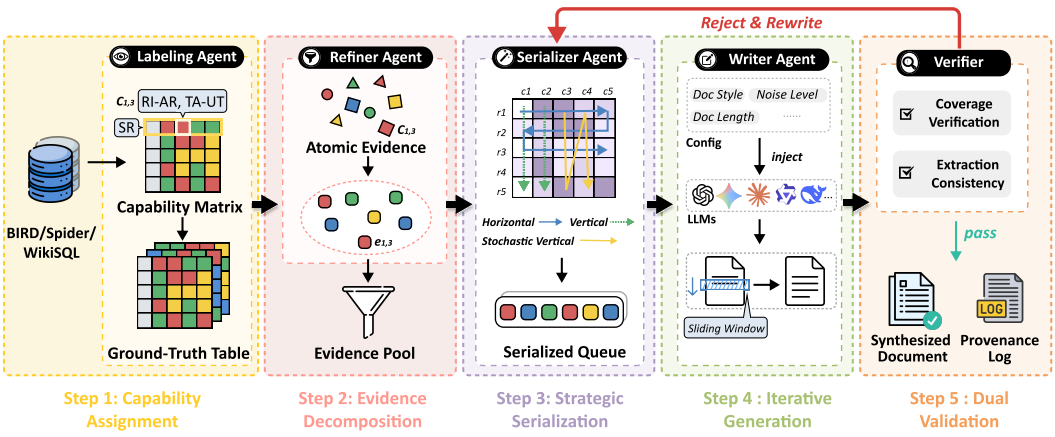


Figure 3: Overview of the benchmark construction framework.

$\mathcal{L}ij$. Multi-label assignment captures compositional requirements, *e.g.,* a cell demanding summation followed by currency conversion is labeled $\mathcal{L}ij$ = `RI-AR, TA-UT`, as illustrated in Sec. 3.

Step 2: Joint Evidence Decomposition. A *Refiner Agent* processes each annotated cell C_{ij} and decomposes it into an atomic evidence set $E_{ij} = e_1, e_2, \ldots, e_k$, forming the global evidence pool $\mathcal{P}$. Critically, the agent performs *joint execution* over two orthogonal constraint dimensions simultaneously: the **attribute-level label** (Pillar 1, *e.g.,* `TA-EM` for enumeration mapping) and the **row-level label** (Pillar 2, *e.g.,* `SR` for structural reasoning). To ensure evidence quality, we apply checklist-based verification along four dimensions: (i) *value/relation correctness*, ensuring factual consistency with target values and relations; (ii) *label faithfulness*, verifying that the assigned capability is properly instantiated; (iii) *schema linkage*, checking consistency with valid schema elements and cross-table references; and (iv) *data integrity and null faithfulness*, preventing unsupported content and preserving missing values. Failed cases receive diagnostic feedback and are iteratively revised until verification succeeds or a retry limit is reached. Detailed criteria are provided in Appendix E.2.

Step 3: Strategic Serialization. A *Serializer Agent* linearizes the verified evidence pool $\mathcal{P}$ into coherent blocks, controlling the document's macro-structure and evidence organization. Each evidence item $e \in \mathcal{P}$ is indexed by its entity i and attribute j, and serialized according to $V(e) = (\pi_1, \pi_2, \epsilon)$, where (π_1, π_2) denotes a mode-specific ordering of the entity and attribute indices (i, j), and $\epsilon \sim U(0, 1)$ serves as a stochastic tie-breaker. We consider three serialization strategies: *Horizontal* ($V = (i, j, \epsilon)$), which groups evidence by entity; *Vertical* ($V = (j, i, \epsilon)$), which groups evidence by attribute across entities; and *Stochastic Vertical* ($V = (\pi(j), i, \epsilon)$), which follows the vertical scheme while randomly permuting the attribute order. Together, these strategies introduce controllable variation in document structure while maintaining coherent evidence organization.

Step 4: Iterative Document Generation. A *Writer Agent* generates the final document block by block following the serialized plan. A context-aware sliding window conditions each block on the current evidence E_{block} and a bounded history of previously generated text H_{prev}, promoting fluent transitions and consistent cross-block coreference.

Furthermore, to better emulate the complexity of real-world documents, generation is controlled by a **plug-and-play configuration module**. It supports configurable parameters such as `Document_Length`, `Noise_Level` ($\in [0, 1]$), and `Document_Style`. The noise level controls the frequency of natural distractors, including irrelevant background content, formatting artifacts, and header/footer interference, while the style parameter follows domain-specific reference documents (*e.g.,* formal legal writing or concise clinical shorthand). Together, these controls increase document realism and diversity without violating the factual constraints in the evidence pool $\mathcal{P}$.

Step 5: Dual Validation. To guarantee that the synthesized document D_{syn} satisfies the rigorous constraints of the Doc2DB paradigm, a *Verifier Agent* conducts a final dual-validation pass. This phase operates as an inverse-consistency check:

- **Coverage Verification:** Ensures that no required atomic evidence from $\mathcal{P}$ is omitted or semantically distorted, utilizing explicit boundary tags (*e.g.,* `<frag_i>...</frag_i>`) injected during generation to deterministically trace textual spans back to their source atoms.

Table 2: Comprehensive statistics of the Doc2DB-Bench benchmark. The # denotes the total count, and *Avg. Len.* indicates the average document length per sample in tokens.

Domain	Document		Schema			Table Scale (Total)		
	# Docs	Avg. Len.	# DBs	# Ent.	# Rel.	# Rows	# Cols	# Cells
Education	13	40,424	3	12	12	830	96	3,451
Finance	59	40,073	10	23	39	1,918	353	10,454
Medical	11	43,930	3	9	12	343	199	3,522
University	31	33,913	6	17	21	860	209	4,339
Sports	27	50,991	5	20	16	1,085	177	5,772
Transportation	17	53,091	5	12	6	718	128	4,678
Others	43	46,566	10	24	26	1,587	313	9,719
Total/Avg.	**203**	**43,326**	**42**	**117**	**132**	**7,341**	**1,475**	**41,935**

- **Extraction Consistency:** An evaluation mechanism is deployed to re-extract the database instance D'_{syn} from D_{syn}. We rigorously verify the structural isomorphism and value equivalence between the re-extracted D'_{syn} and the ground-truth database D.

If D_{syn} fails either validation criterion due to hallucinated relations, missing evidence, or referential inconsistencies, it is routed back to Step 4 for targeted rewriting with error feedback. This closed-loop validation minimizes unrecoverable or contradictory information in the final benchmark corpus.

Doc2DB-Bench Statistics. We synthesize documents using databases from BIRD (Li et al., 2023) and Spider (Yu et al., 2018) as ground truth. Specifically, we curate 42 high-quality databases across seven domains, including education and medical. Details of database collection and processing are provided in Appendix A. All agents use Gemini-2.5-Pro (Comanici et al., 2025) as the backbone model, selected based on preliminary experiments balancing quality and cost. All synthesized cases are verified by seven computer science PhD candidates, each specializing in a corresponding domain.

As summarized in Table 2, Doc2DB-Bench contains 42 synthesized (Document, Specification, Database) samples, covering 117 entity tables, 132 relationship tables, 7,341 rows, and 41,935 cells across seven domains. The benchmark targets schema understanding, long-context reasoning, and cross-table dependency modeling, with fine-grained capability annotations for 11,205 cells (¿25%) and 3,129 rows (¿40%). To guide realistic synthesis, we collect reference docs from DocBench (Zou et al., 2025), MMLongBench (Ma et al., 2024), MMDocRAG (Dong et al., 2025), and web sources, spanning financial and legal reports to research abstracts. Fig. 2 shows the capability distribution; unannotated cells and rows correspond to directly extractable cases requiring verbatim recovery.

5 Experiments

In this section, we systematically evaluate the performance of various LLMs on Doc2DB-Bench for the Doc2DB extraction task. Specifically, we seek to answer the following research questions:

RQ1: How well do different LLMs perform on Doc2DB extraction at the entity and relation levels?

RQ2: Beyond overall performance, what specific capabilities and vulnerabilities do these models exhibit across the fine-grained dimensions defined in our taxonomy?

RQ3: How realistic and authentic are the documents generated by our automated pipeline?

5.1 Setup

Baselines. We benchmark a broad range of state-of-the-art LLMs, categorized into three groups: (1) *Open-weight models*, including Qwen2.5-14B/72B-Instruct (Yang et al., 2024) and Llama-3.1-70B-Instruct; (2) *Proprietary APIs*, including GPT-4o (Hurst et al., 2024), GPT-5.4, Gemini-2.5-Flash/Pro (Comanici et al., 2025), Claude-opus-4-6 (Anthropic, 2026), Qwen3-Max, and DeepSeek-V4-Flash (Xu et al., 2026); and (3) *Specialized IE systems*, including LlamaExtract (LlamaIndex, 2024), a commercial service offering iterative refinement and citations; LangExtract (Google, 2025), which uses chunking and few-shot prompting for long-document extraction; LangChain (Harrison Chase, 2022), a general framework with schema-constrained function calling for structured outputs; and DocETL (Shankar et al., 2024), a declarative framework with modular LLM extraction pipelines.

Table 3: Comparative performance of LLMs on Doc2DB-Bench. **Green** highlights the best.

Model	Size	Entity-Level			Relation-Level			Overall Performance			
		P.	R.	F1	P.	R.	F1	P.	R.	F1	LS
Open-source Models											
Qwen2.5-14b-Ins	14b	42.87	60.17	44.07	27.45	43.54	28.92	29.40	50.36	32.97	39.39
LLaMA-3.1-70b-Ins	70b	14.96	58.89	17.25	20.29	38.79	16.97	12.12	47.46	14.29	22.76
Qwen2.5-72b-Ins	72b	41.62	66.57	45.03	28.48	54.63	31.98	30.96	59.53	36.07	39.53
Proprietary Models											
GPT-4o (Hurst et al., 2024)	-	72.39	70.92	70.30	62.94	46.78	50.68	65.66	55.99	59.05	47.52
GPT-5.4	-	79.93	81.12	80.09	80.09	69.63	72.33	78.47	73.93	75.25	69.02
Gemini-2.5-flash (Comanici et al., 2025)	-	75.89	75.48	74.21	73.60	57.35	59.50	71.33	64.58	65.99	53.74
Gemini-2.5-pro (Comanici et al., 2025)	-	76.80	82.24	78.13	67.98	67.01	65.46	69.83	73.80	70.81	57.19
Claude-opus-4-6	-	87.19	83.70	84.93	66.71	62.16	63.17	77.33	71.57	73.60	61.95
Qwen3-max	-	76.51	77.00	75.77	57.01	59.74	55.96	63.48	66.42	63.44	52.48
Deepseek-V4-flash	-	76.65	66.06	67.30	67.67	43.23	48.93	69.03	52.14	57.06	49.05
Specialized IE Systems											
LangExtract (Google, 2025)	-	29.72	40.50	33.49	33.34	19.08	22.00	36.06	25.20	28.72	23.55
DocETL (Shankar et al., 2024)	-	78.20	76.85	74.91	46.75	41.26	41.51	65.27	60.97	60.82	49.87
LangChain (Harrison Chase, 2022)	-	60.77	83.09	69.09	33.50	42.47	36.33	47.28	64.38	53.83	44.14
LlamaExtract (LlamaIndex, 2024)	-	52.28	68.95	56.58	47.41	48.73	47.39	45.35	56.49	49.10	40.26

Evaluation Details. For fair and reproducible comparison, we use identical prompts and greedy decoding with temperature 0 across all experiments. All systems use GPT-5.4 as the backbone model, except LlamaExtract, which uses its proprietary model. To isolate relational reasoning from upstream entity extraction errors, we additionally introduce an **oracle entity setting**, where the model $\mathcal{M}$ receives ground-truth entity tables E^* and predicts only relation tables: $T_{\text{rel}} = \mathcal{M}(D, S, E^*)$.

Metrics. To evaluate extraction at a granular level, we employ the following metrics:

- **Cell-level Accuracy.** We evaluate fine-grained attribute extraction using cell-level Precision, Recall, and F_1. Rather than relying on row-level matching, which is sensitive to tuple ordering and may cause cascading alignment errors, we employ a **Global Maximum Weight Matching** algorithm to establish one-to-one tuple alignment between predicted (T) and ground-truth (T^*) tables (details in Appendix C.1). Let C_{match} denote the number of cells satisfying our matching criteria (*e.g.,*, exact numeric equality or ≥90% string similarity). The metrics are defined as:

$$P = \frac{|C_{\text{match}}|}{C_{\text{total}}(T)}, \quad R = \frac{|C_{\text{match}}|}{C_{\text{total}}(T^*)}, \quad F_1 = \frac{2PR}{P+R},$$

  where $C_{\text{total}}(\cdot)$ denotes the number of non-empty cells in the corresponding table.
- **Semantic Quality.** We employ GPT-4o as an evaluator to assess semantic equivalence beyond exact cell matching. It assigns a score from 0 to 100 based on factual correctness, schema conformity, and robustness to paraphrasing, capturing semantically equivalent but structurally varied predictions. Evaluation prompt is provided in Appendix C.2, with further results in Appendix D.

5.2 Experimental Results and Analysis

Overall Performance. As shown in Table 3, proprietary models consistently outperform open-source counterparts across all metrics. In particular, GPT-5.4 achieves the best overall F1 score (75.25) and the highest llm score (69.02), indicating strong capability in structured extraction. Claude-opus-4-6 and Gemini-2.5-pro also deliver competitive performance. Notably, GPT-4o shows only moderate results (Overall F1 59.05), falling noticeably short of reasoning-capable models despite its scale. In contrast, open-source models such as Qwen2.5-14B-Ins exhibit significantly lower overall F1 (32.97), highlighting the performance gap under complex extraction settings.

Finding 1. Models with stronger reasoning capabilities consistently achieve better performance on Doc2DB extraction, where structural reasoning proves critical for schema-compliant output.

Entity-level performance. Entity-level results reveal a clear advantage of high-capacity proprietary models. Claude-opus-4-6 achieves the highest entity-level F1 (84.95), followed by GPT-5.4 (80.09) and Gemini-2.5-pro (78.13). These models maintain a good balance between precision and recall, indicating robust span detection and boundary alignment. Compared to them, Qwen2.5-14B shows notably lower precision (42.87) despite relatively higher recall (60.17), suggesting that open-source models tend to over-generate entity spans, leading to reduced accuracy.

Table 4: The Comparison of Capabilities Across Different Models for Different Categories. Cell colors represent performance levels (Blue: High; Red: Low).

Capability	Sub-capabilities	Qwen2.5-14B-Ins	LLaMA-3.1-70B-Ins	GPT-4o	DeepSeek-V4-Flash	Qwen3-Max	Claude-Opus-4-6	Gemini-2.5-Pro	GPT-5.4
Identity Resolution	Entity Disambiguation	51.58	59.92	69.24	55.25	64.57	72.96	80.03	79.49
	Coreference Resolution	37.84	50.13	61.14	62.10	56.76	67.61	77.71	82.18
Relationship Linking	One-to-Many Allocation	49.06	51.90	65.63	62.27	64.77	70.38	78.18	81.83
	Multi-Entity Binding	47.65	53.85	60.18	53.83	58.57	64.82	67.26	82.97
	Conditional Linkage	52.02	57.47	69.10	62.65	61.80	69.64	75.81	79.16
Structural Reasoning	Transitive Inference	40.80	57.83	65.51	57.35	53.78	59.48	70.82	78.47
	Global Aggregation	49.23	48.30	71.59	59.83	62.15	64.25	73.03	82.20
	Dynamic Change	36.51	51.83	58.11	46.93	58.36	66.26	78.69	79.27
Integrity Constraint	Referential Integrity	-	-	-	-	-	-	-	-
	Mutual Exclusion	36.41	56.94	60.18	58.45	54.61	61.96	70.39	74.46

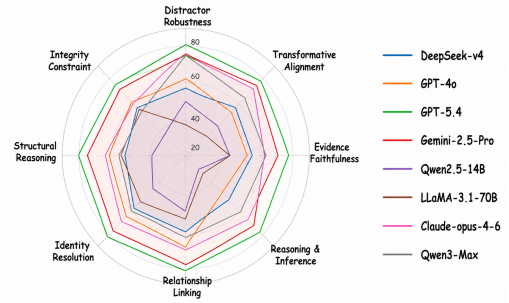


Figure 4: Radar chart of different models across various capabilities.

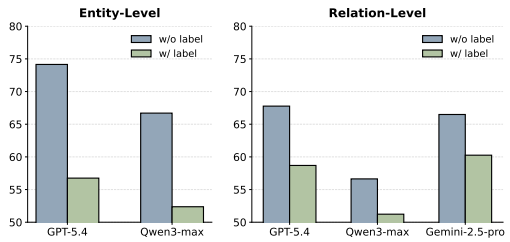


Figure 5: Impact of capability annotations on extraction performance.

Relationship level performance. Relation extraction is evaluated under an oracle entity setting: given the relational schema, the source document, and the ground-truth entity tables, the model generates only the relation tables. This setting isolates relational reasoning ability from upstream entity extraction errors. As shown in Table 3, relation-level extraction remains substantially more challenging than entity extraction for most models. A key reason is that relation prediction requires not only identifying the correct records, but also globally aligning related entities and tuples under a schema, which is more error-prone than localized entity extraction. GPT-5.4 achieves the best relation-level F1 (72.33), outperforming GPT-4o (50.68), Gemini-2.5-pro (65.46), and Claude-opus-4-6 (63.13), indicating stronger capability in global alignment and structured relational reasoning.

Finding 2. Relation-level extraction is a major bottleneck in Doc2DB extraction. Compared with entity extraction, it is substantially more challenging because it requires accurate global alignment across entities and tuples, leading to consistently lower performance.

5.3 Capability-Level Performance

As revealed in Section 5.2, relation extraction is the primary bottleneck in Doc2DB extraction, yet overall metrics do not reveal *which specific inter-table capabilities* remain challenging for LLMs. We therefore conduct a fine-grained evaluation over Pillar 2 capabilities in our taxonomy.

For each capability, we compute cell-level Precision, Recall, and F_1 on the corresponding annotated relation tables using the same matching criteria, enabling fine-grained analysis of inter-table reasoning abilities such as entity linking, relation alignment, and multi-hop dependency recovery.

Capability assignment introduces difficulty. As shown in Fig. 5, models perform consistently worse on documents with explicit capability labels than on those without. The drop is substantial at the entity level, with GPT-5.4 decreasing from 74.15 to 56.75 and Qwen3-max from 66.70 to 52.37. At the relation level, the decline is smaller but consistent across models (GPT-5.4: 67.77 to 58.69; Qwen3-max: 56.63 to 51.23; Gemini-2.5-pro: 66.49 to 60.26). These results confirm that capability-annotated documents are genuinely harder and validate the effectiveness of our taxonomy.

Detailed Error Analysis. Fig. 4 shows that GPT-5.4 performs best across all capabilities, while Table 4 reports fine-grained results across four inter-table capabilities, revealing three key findings: *(i) Implicit evidence amplifies model disparities.* Surface-level tasks rely on explicit cues, whereas

Table 5: Quality and Authenticity evaluation of Doc2DB-Bench documents using (a) rubric-based LLM and human assessment and (b) commercial AI detectors.

Judger	Lexical Richness	Logical Consistency	Textual Coherence	Genre Fidelity	Average
LLM	4.46	4.28	3.84	4.94	4.38
Human	4.17	3.97	3.98	4.00	4.03

Method	Doc2DB-Bench	Reference
ZeroGPT	35.20	23.12
FastGPT Bao et al. (2023)	37.50	35.38

coreference resolution and multi-entity binding require aggregating scattered implicit evidence, significantly widening the gap between strong and weak models. *(ii) Structural reasoning remains a major bottleneck.* Even SOTA models plateau at around 80% on Dynamic Change and Transitive Inference, while smaller models fall below 40%, indicating limited multi-step relational reasoning ability. *(iii) Integrity Constraint (IC) verification is consistently weak.* Models often hallucinate dependencies in Referential Integrity tests and violate exclusivity rules in Mutual Exclusion tasks, highlighting the lack of constraint-aware generation and potential risks to database reliability.

Finding 3.. Implicit reasoning over distributed and compositional evidence increases difficulty and widens model gaps, with structural reasoning and integrity constraints as the primary failure modes.

5.4 Authenticity Verification

To evaluate document quality and realism, we conduct a rubric-based assessment using an LLM judge and human annotators. Documents are evaluated along four dimensions: *lexical richness, logical consistency, textual coherence, and genre fidelity*, using a five-point scale. We use DeepSeek-V3.2 as the automatic evaluator and recruit five graduate students document intelligence research experience to independently assess 50 randomly sampled documents using the same rubrics. As shown in Table 5(a), the LLM and human evaluations consistently demonstrate the high linguistic quality and domain authenticity. Commercial AI detectors (ZeroGPT and FastGPT (Bao et al., 2023)) show that Doc2DB-Bench closely matches real-world distributions in AI-generation scores (Table 5(b)). Notably, the FastGPT score (37.50%) is nearly identical to the real-data baseline (35.38%). Overall, these results demonstrate that Doc2DB-Bench preserves the lexical diversity and stylistic nuances of human-written reports, achieving high fidelity. More details are provided in Appendix F.

6 Related Work

Structured Information Extraction. Traditional IE typically separates entity recognition and relation extraction (Hobbs, 1993; Lee et al., 2013; Li et al., 2022; Wu et al., 2008; Zheng et al., 2018), making schema customization difficult when target attributes fall outside predefined ontologies (Jiao et al., 2023). Recent LLM-based work spans multistep prompting (Deng et al., 2024; Ahuja et al., 2025; Tang et al., 2024), structured summarization (StructSum (Jain et al., 2024)), query-driven attribute extraction (InstructIE (Jiao et al., 2023), Lotus (Patel et al., 2025), and PZ (Liu et al., 2025)), and hybrid LLM–IE pipelines (Doctopus (Chai et al., 2025)). Yet most methods target flat Doc2Table outputs (Tang et al., 2024; Jiao et al., 2023; Liang et al., 2026), leaving the construction of relational databases with inter-table dependencies and integrity constraints from documents underexplored.

Existing Benchmarks. Rotowire (Wiseman et al., 2017) originated as a table-to-text dataset rather than a database extraction benchmark. LiveSum (Deng et al., 2024) covers limited domains, while InstructIE (Jiao et al., 2023) and StructText (Kashyap et al., 2025) use simplified schemas and direct text–tuple alignments. SemBench (Lao et al., 2025) and UDA-Bench (Deng et al., 2025b) evaluate semantic query processing and unstructured data analysis, respectively, rather than document-to-database extraction. DTBench (ZJU-DAILY, 2026) provides capability-aware evaluation for Doc2Table but does not assess cross-table construction. SQUiD (Sadia et al., 2025) studies relational database synthesis from text; its evaluation primarily uses LLM-generated descriptions of tuples, with small-scale tests on real text. These gaps motivate Doc2DB-Bench, the first benchmark for systematic LLM evaluation of schema-guided Doc2DB extraction from long documents.

7 Conclusion

We introduce Doc2DB-Bench, a novel benchmark for faithful document-to-database construction. To reduce annotation costs, we develop a controllable reverse-synthesis pipeline guided by a taxonomy of intra- and inter-table capabilities. Our evaluation shows that leading LLMs perform well on localized entity extraction but struggle with global relational alignment, multi-hop structural reasoning, and database integrity constraints. Doc2DB-Bench provides a realistic testbed for developing reliable, schema-compliant, and auditable LLM-based data systems, with dataset expansion underway.

AI Use Statement

LLMs are used only for auxiliary purposes such as language refinement, minor code debugging, synthetic data construction, and experiment evaluation support. They do not contribute to the research design, methodology, or core writing of the paper. Accordingly, LLM usage does not constitute a substantive contribution to the intellectual content of this work.

Ethics Statement

The proposed benchmark aims to support reliable and reproducible evaluation of LLMs for document-to-database extraction. We do not anticipate direct negative social impacts from the benchmark itself, but its application in sensitive domains warrants care. Although the evaluation documents are synthesized from public database benchmarks, source data and generative models may introduce biases or inaccuracies. We mitigate these risks through automated checks of evidence coverage and database consistency, followed by human review of document realism and recoverability. We encourage users to respect source-data licenses and conduct domain-specific validation before deploying systems evaluated on Doc2DB-Bench in finance, healthcare, or other consequential settings.

Reproducibility Statement

We document database selection, schema refinement, tuple sampling, and human verification in Appendix A. Section 3 and Appendix B define the capability taxonomy; Appendix C details the evaluation metrics, tuple matching, and LLM judge prompt; and Appendix E provides the synthesis and validation prompts. Model comparisons use identical prompts and greedy decoding with temperature 0. The benchmark contains 203 synthesized documents across 42 database instances, with ground-truth tables and fine-grained capability annotations. The benchmark and construction code are available in `https://anonymous.4open.science/r/Doc2DB-Bench-3EEF`.

References

Naman Ahuja, Fenil Bardoliya, Chitta Baral, and Vivek Gupta. Map&make: Schema guided text to table generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30249–30262, 2025.

Anthropic. Claude opus 4.6 system card. `https://www-cdn.anthropic.com/6a5fa276ac68b9aeb0c8b6af5fa36326e0e166dd/Claude%20Opus%204.6%20System%20Card.pdf`, 2026.

Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. Llms4synthesis: Leveraging large language models for scientific synthesis. In *Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries*, pp. 1–12, 2024.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*, 2023.

Chengliang Chai, Jiajun Li, Yuhao Deng, Yuanhao Zhong, Ye Yuan, Guoren Wang, and Lei Cao. Doctopus: Budget-aware structural table extraction from unstructured documents. *Proceedings of the VLDB Endowment*, 18(11):3695–3707, 2025.

Zui Chen, Zihui Gu, Lei Cao, Ju Fan, Samuel Madden, and Nan Tang. Symphony: Towards natural language query answering over multi-modal data lakes. In *CIDR*, pp. 1–7, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhong-Zhi Li, Jian Xu, Xiao-Hui Li, Yuan Gao, Jun Song, Bo Zheng, et al. Longdocurl: a comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1135–1159, 2025a.

Qiyan Deng, Jianhui Li, Chengliang Chai, Jinqi Liu, Junzhi She, Kaisen Jin, Zhaoze Sun, Yuhao Deng, Jia Yuan, Ye Yuan, et al. Unstructured data analysis using llms: a comprehensive benchmark. *arXiv preprint arXiv:2510.27119*, 2025b.

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. Text-tuple-table: Towards information integration in text-to-table generation via global tuple extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9300–9322. Association for Computational Linguistics, 2024.

Kuicai Dong, Yujing Chang, Shijie Huang, Yasheng Wang, Ruiming Tang, and Yong Liu. Benchmarking retrieval-augmented multimodal generation for document question answering. *arXiv preprint arXiv:2505.16470*, 2025.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601–610, 2014.

Google. Introducing langextract. `https://langextract.net`, 2025.

Harrison Chase. Langchain. `https://github.com/langchain-ai/langchain`, 2022.

Jerry Hobbs. Fastus: A finite-state processor for information extraction from real-world text. In . . . *JOINT CONFERENCE ON. . .* , 1993.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Parag Jain, Andreea Marzoca, and Francesco Piccinno. Structsum generation for faster text comprehension. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7876–7896, 2024.

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. Instruct and extract: Instruction tuning for on-demand information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10030–10051. Association for Computational Linguistics, 2023.

Satyananda Kashyap, Sola Shirai, Nandana Mihindukulasooriya, and Horst Samulowitz. Structtext: A synthetic table-to-text approach for benchmark generation with multi-dimensional evaluation. *CoRR*, abs/2507.21340, 2025.

Jiale Lao, Andreas Zimmerer, Olga Ovcharenko, Tianji Cong, Matthew Russo, Gerardo Vitagliano, Michael Cochez, Fatma Özcan, Gautam Gupta, Thibaud Hottelier, et al. Sembench: A benchmark for semantic query processing engines. *arXiv preprint arXiv:2511.01716*, 2025.

Taesung Lee, Zhongyuan Wang, Haixun Wang, and Seung-won Hwang. Attribute extraction and scoring: A probabilistic approach. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 194–205. IEEE, 2013.

Boyan Li, Zhuowen Liang, Yupeng Xie, Xiaotian Lin, Tianqi Luo, Xinyu Liu, Yizhang Zhu, Zhangyang Peng, Yuan Li, Zhengxuan Zhang, et al. Dataspace: Benchmarking data agents for verifiable analytics over heterogeneous workspaces. *arXiv preprint arXiv:2608.03451*, 2026.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36:42330–42357, 2023.

Yinghao Li, Le Song, and Chao Zhang. Sparse conditional hidden markov model for weakly supervised named entity recognition. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 978–988, 2022.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. In *International Conference on Learning Representations*, volume 2025, pp. 36107–36124, 2025.

Zhuowen Liang, Xiaotian Lin, Zhengxuan Zhang, Yuyu Luo, Haixun Wang, and Nan Tang. Long-document qa with chain-of-structured-thought and fine-tuned slms. *arXiv preprint arXiv:2603.29232*, 2026.

Chunwei Liu, Matthew Russo, Michael Cafarella, Lei Cao, Peter Baile Chen, Zui Chen, Michael Franklin, Tim Kraska, Samuel Madden, Rana Shahout, et al. Palimpzest: Optimizing ai-powered analytics with declarative query processing. In *Proceedings of the Conference on Innovative Database Research (CIDR)*, pp. 2, 2025.

LlamaIndex. Unstructured data extraction: Turn documents into insights. `https://www.llamaindex.ai/blog/unstructured-data-extraction`, 2024.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37: 95963–96010, 2024.

Holly Muscolino, Amy Machado, Dan Vesset, and John Rydning. What every executive needs to know about unstructured data. Technical report, IDC Corp., sponsored by Box Inc., August 2023. `https://www.storagenewsletter.com/2023/09/07/what-every-executive-needs-to-know-about-unstructured-data/`.

Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 201–206. Association for Computational Linguistics, 2017.

Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. Semantic operators and their optimization: Enabling llm-based data processing with accuracy guarantees in lotus. *Proceedings of the VLDB Endowment*, 18(11):4171–4184, 2025.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017.

Christopher De Sa, Ce Zhang, Jaeho Shin, and Christopher Ré. Deepdive: Declarative knowledge base construction. *SIGMOD Record*, 45(1):60–67, 2016.

Mushtari Sadia, Zhenning Yang, Yunming Xiao, Ang Chen, and Amrita Roy Chowdhury. Squid: Synthesizing relational databases from unstructured text. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 31987–32012. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1629. URL `https://doi.org/10.18653/v1/2025.emnlp-main.1629`.

Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G Parameswaran, and Eugene Wu. Docetl: Agentic query rewriting and evaluation for complex document processing. *arXiv preprint arXiv:2410.12189*, 2024.

Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. Incremental knowledge base construction using deepdive. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, volume 8, pp. 1310, 2015.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*, 2017.

Fei Wu, Raphael Hoffmann, and Daniel S Weld. Information extraction from wikipedia: Moving down the long tail. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 731–739, 2008.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Derong Xu, Wei Chen, Wenjun Peng, et al. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18, 2024. doi: 10.1007/s11704-024-40555-y. https://journal.hep.com.cn/fcs/EN/10.1007/s11704-024-40555-y.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 3911–3921, 2018.

Tianjing Zeng, Yuntao Hong, Zhongjun Ding, Dandan Liu, Yinan Mei, Yunxiang Su, Yiming Wang, Xiaojian Zhang, Jingyu Zhu, Junhao Zhu, et al. Qwenpaw-data: Bridging facts, methodology, and execution for autonomous enterprise data analytics. *arXiv preprint arXiv:2607.11019*, 2026.

Zhengxuan Zhang, Zhuowen Liang, Yin Wu, Teng Lin, Yuyu Luo, and Nan Tang. Datamosaic: Explainable and verifiable multi-modal data analytics through extract-reason-verify. *arXiv e-prints*, pp. arXiv–2504, 2025.

Zhengxuan Zhang, Zhuowen Liang, Jiazhuo Chen, Haixun Wang, and Nan Tang. Document-to-database: Extraction meets relational semantics. volume 19, pp. 2522–2535. VLDB Endowment, 2026.

Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1049–1058, 2018.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

ZJU-DAILY. DTBench: A synthetic benchmark for document-to-table extraction, 2026. https://arxiv.org/abs/2602.13812.

Anni Zou, Wenhao Yu, Hongming Zhang, Kaixin Ma, Deng Cai, Zhuosheng Zhang, Hai Zhao, and Dong Yu. Docbench: A benchmark for evaluating llm-based document reading systems. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pp. 359–373, 2025.

A Dataset Collection and Processing

To establish a relationally faithful ground truth for Doc2DB-Bench, we curated a collection of complex relational databases from two predominant text-to-SQL benchmarks: BIRD (Li et al., 2023) and Spider (Yu et al., 2018). The collection process followed a multi-stage pipeline designed to maximize domain diversity and structural complexity.

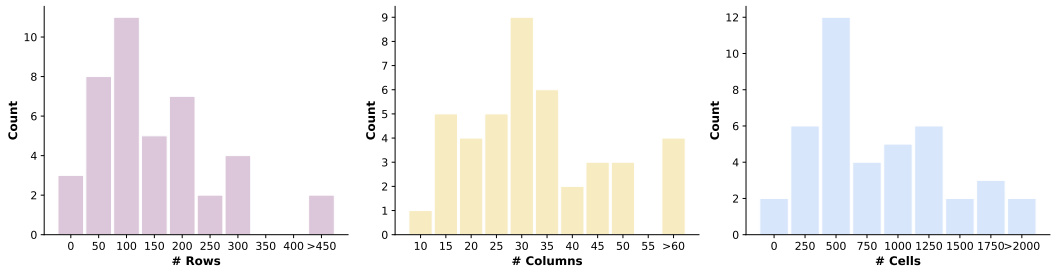


Figure 6: Distribution of database scale in Doc2DB-Bench in terms of rows, columns, and cells.

A.1 Database Selection and Sourcing

We first screen the database pools from BIRD and Spider, selecting databases that are suitable for document-to-database evaluation. Specifically, we prioritize databases with at least three interconnected tables, non-trivial schemas involving multiple entity types, relationship tables, and foreign-key dependencies, as well as diverse cross-table dependencies. To ensure broad coverage, the final collection spans seven domains, including education, finance, healthcare, university, sports, transportation, and others. Unlike conventional benchmarks that focus on flat-table extraction, our selection favors databases with rich entity-relationship (ER) structures and explicit relational dependencies, enabling the evaluation of cross-table reasoning and database construction capabilities.

A.2 Data Cleansing and Schema Refinement

The raw databases were refined for document synthesis as follows:

- **Schema Standardization:** We normalize attribute names, unify schema representations, and ensure that primary and foreign keys are explicitly defined. This step reduces schema ambiguity and facilitates reliable DB-to-document synthesis.
- **Tuple Sampling and Instance Construction:** Since using all database tuples may produce excessively long documents and dilute evidence density, we sample representative tuples while preserving entity coverage, inter-table relationships, and referential integrity. The resulting instances remain complete relational databases rather than isolated table fragments, yielding 117 entity tables and 132 relationship tables with 7,341 rows and 41,935 cells. Fig. 6 further illustrates the distribution of database scales in terms of rows, columns, and cells.
- **Integrity Verification:** We perform SQL-based sanity checks on the sampled databases to ensure that all instances satisfy relational constraints, including primary-key uniqueness, foreign-key referential integrity, and attribute-level validity. We further verify schema consistency, table connectivity, and tuple completeness to ensure that sampled instances preserve the original relational structures without introducing broken dependencies or isolated fragments. Only verified database instances are retained for subsequent document synthesis.

A.3 Domain-Specific Style Grounding

Although databases from BIRD (Li et al., 2023) and Spider (Yu et al., 2018) provide diverse domain schemas and structured facts, they are not designed to capture realistic document styles and writing conventions. To improve document realism, we augment the structured data with domain-specific narrative styles by collecting reference documents from LongDocURL (Deng et al., 2025a), DocBench (Zou et al., 2025), MMLongBench (Ma et al., 2024), and MMDocRAG (Dong et al., 2025). These references cover diverse document formats, ranging from academic papers and financial reports to posters and restaurant menus, serving as stylistic anchors for the synthesis pipeline. They enable the generation of authentic domain-specific documents while strictly preserving the factual constraints of the underlying databases.

A.4 Verification Procedure.

Although our dataset is generated through a multi-agent synthesis pipeline, we incorporate a rigorous verification procedure to ensure the quality and reliability of the synthesized documents.

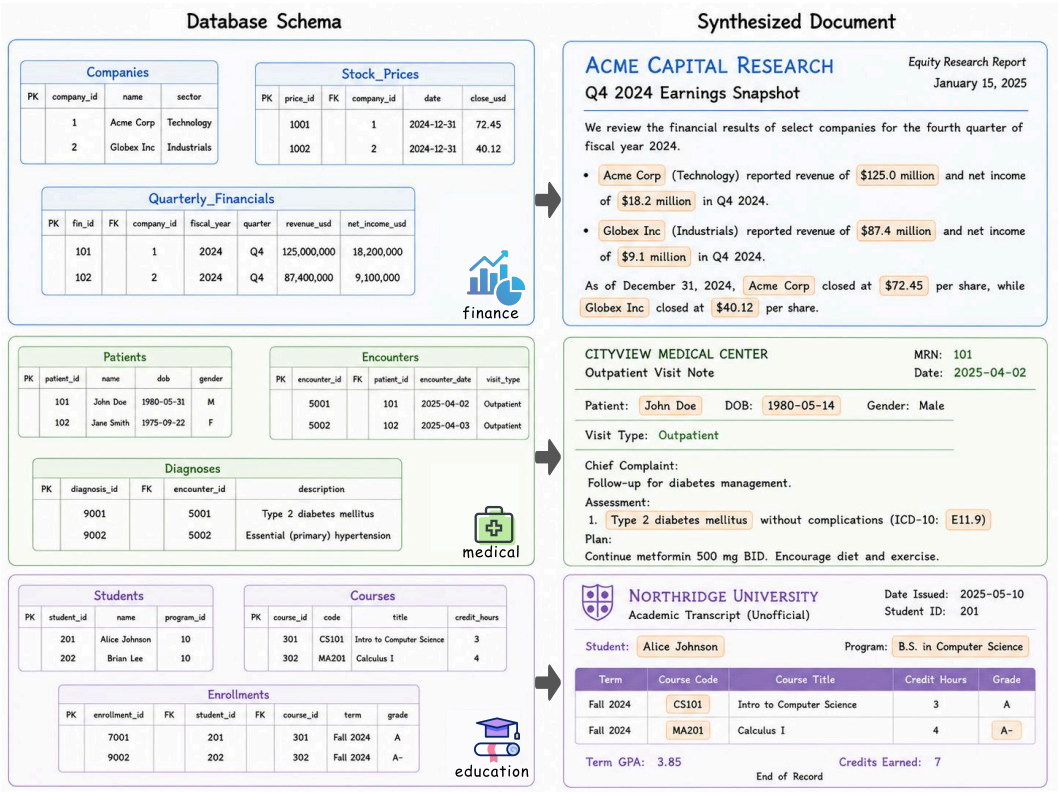


Figure 7: Representative examples from the finance, healthcare, and education domains. Each case pairs a multi-table relational schema with a domain-specific synthesized document, where colored links illustrate the grounding between database records and document evidence.

First, our framework includes a dedicated **Validator Agent** that performs automatic verification from two perspectives. At the fine-grained level, it checks whether generated sentences and paragraphs faithfully cover the information contained in the source databases, preventing missing or incomplete evidence. At the database reconstruction level, it evaluates whether the complete database can be accurately recovered from the generated document. Any detected inconsistencies or missing information trigger an iterative repair process to refine the generated documents. Detailed verification and repair procedures are provided in Appendix E.4.

Second, we conduct human verification on the generated documents. We recruit seven PhD candidates in computer science, with each reviewer responsible for one specific domain. Each reviewer evaluates the synthesized documents from two aspects: (1) whether the document style and content align with realistic domain-specific documents encountered in practice; and (2) whether the underlying database information can be reliably reconstructed from the generated document. Any cases that fail these criteria, such as unrealistic document patterns, missing database evidence, or inconsistent entity relationships, are filtered out to ensure the quality and reliability of the final dataset.

A.5 Dataset Visualization.

Figure 7 presents representative examples from the finance, healthcare, and education domains, each pairing a multi-table relational schema with a realistic domain-specific document. For example, in the finance case, company information is stored in the `Companies` table, quarterly revenue and net income in `Quarterly_Financials`, and closing prices in `Stock_Prices`. The synthesized equity research report integrates these distributed records into statements such as "Acme Corp reported revenue of $125.0 million and net income of $18.2 million," while linking them through the shared `company_id`. Recovering the database therefore requires not only extracting individual values, but also resolving entity references, normalizing expressions such as `125,000,000` and "$125.0 million," and correctly assigning evidence to multiple related tables. Similar examples across the other domains illustrate the diversity of schema structures, document styles, and cross-table dependencies covered by Doc2DB-Bench.

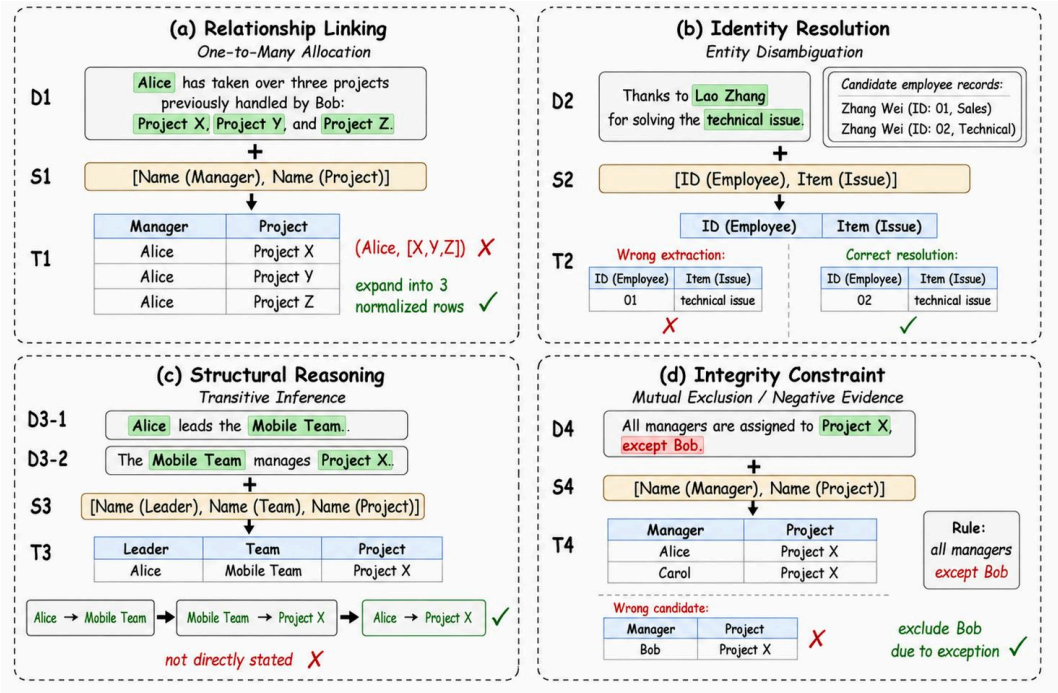


Figure 8: Challenging examples of relation extraction that require different inter-table capabilities.

B Details of dual-level Taxonomy

Pillar 2 inter-table capabilities are organized into the following four types:

Relationship Linking (RL). This capability is required when natural language narratives must be parsed and mapped into structured relational tuples, accurately aligning multiple entities across tables based on cardinality and business rules. We categorize this linking into three subcategories.

One-to-Many Allocation. The document provides a compressed statement (*e.g.,* a manager taking over multiple projects), requiring the model to expand it into multiple normalized relationship rows that preserve the underlying 1:N or N:1 cardinality rather than collapsing them. As illustrated in Fig. 8 (a), a statement assigning Alice to Projects X, Y, and Z must be expanded into three separate relationship tuples.

Multi-Entity Binding. The document describes intertwined N-ary relations (*e.g.,* distinct suppliers delivering different parts to various projects) in a dense narrative, requiring exact triplet or multi-entity alignment without mismatching pairs or hallucinating spurious links based on text proximity.

Conditional Linkage. Requires evaluating existing table attributes against implicit business rules to automatically establish foreign key relations, even without explicit entity name mentions.

Identity Resolution (IDR). This capability is required when entity references within the document are ambiguous, pronominal, or use aliases, necessitating cross-referencing with table attributes or surrounding context to ground the mention to a unique, exact database record.

Entity Disambiguation. The document uses an ambiguous alias or name that matches multiple database records, requiring contextual clues and schema attributes to identify the correct entity. As shown in Fig. 8 (b), the alias "Lao Zhang" must be resolved to the employee in the technical department based on the surrounding "technical issue" context.

Coreference Resolution. The target entity is referred to using pronouns, role titles, or implicit aliases scattered across sentences, requiring cross-sentence contextual reasoning to correctly resolve these coreferences to the anchor entity and populate the corresponding relationship fields.

Structural Reasoning (SR). This capability is required when relational links cannot be directly extracted from a single text span but must be logically deduced by chaining multiple implicit relations, table structures, or temporal events.

Transitive Inference. The document and existing tables provide linked but disjointed relational pairs (*e.g.,* entity A is linked to B, and B is linked to C), requiring the deduction of implicit transitive relations to populate the target schema. Fig. 8 (c) illustrates this case: "Alice leads the Mobile Team" and "the Mobile Team manages Project X" must be composed to recover the implicit relation between Alice and Project X.

Global Aggregation. The attributes required to form a complete multi-dimensional relationship are scattered across disjointed text spans, requiring multi-hop reading comprehension to logically chain these spans and assemble all necessary foreign keys into a single valid record.

Dynamic Change. The document narrative involves cross-table event timelines, sequential actions, or entity state changes over time (e.g., completing a prerequisite course before a new enrollment), requiring reasoning over these temporal dynamics to determine the final, valid relationship status.

Integrity Constraint (IC). This capability is required when the extraction of relationships is governed by negative evidence, exceptions, or boundary rules, demanding that the model suppress naive co-occurrence extraction and strictly adhere to database integrity.

Null Relation Extraction. The document provides explicit negative evidence, such as cancelled events or unmet prerequisites, requiring the model to identify an empty relation state (*e.g.,* outputting `NULL`) and avoid hallucinating links from superficial keyword co-occurrence.

Mutual Exclusion. The document specifies broad relationships alongside explicit exceptions or negative constraints, requiring the model to infer relationship boundaries and exclude invalid relational tuples. As illustrated in Fig. 8 (d), although all managers are generally assigned to Project X, the explicit exception "except Bob" requires suppressing the tuple (Bob, Project X).

C Experiment Setup

C.1 Metric Design Principles: Cell Alignment for Evaluation

Naive sequential matching aligns predicted rows to ground-truth rows by position, which is fragile in two ways. Cross-Tuple Hijacking: a model extracts the correct value but assigns it to the wrong row, penalizing an otherwise valid prediction. Cascading Misalignment: a single missed or inserted row shifts all subsequent alignments, inflating both false-positive and false-negative counts. To address both failure modes, we adopt a **Global Maximum Weight Matching** strategy over a bipartite graph of ground-truth and predicted rows.

Notably, We do not use primary keys as alignment anchors because database-specific identifiers, particularly surrogate keys, are often absent from real-world documents and must be generated during database construction. Different models may therefore assign different identifiers to otherwise equivalent records. Primary-key-based matching would conflate arbitrary identifier generation with extraction correctness. We instead align rows using shared non-key attributes and evaluate key validity separately through database-integrity metrics.

Let $T^* = \{r_1^*, \ldots, r_m^*\}$ and $T = \{r_1, \ldots, r_n\}$ denote the ground-truth and predicted row sets, respectively. Matching proceeds in four stages.

Stage 1: Global Scoring Matrix. Let $\mathcal{K}$ be the shared column names between T^* and T. For each pair (r_i^*, r_j), we compute a composite similarity score:

$$s(r_i^*,\, r_j) \;=\; \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathrm{sim}\big(r_i^*[k],\, r_j[k]\big)\,,$$

where $\mathrm{sim}(\cdot,\cdot)$ returns 1 for exact numeric equality after normalization and uses normalized Levenshtein similarity for textual values:

$$\mathrm{sim}_{\text{text}}(a, b) = 1 - \frac{d_{\text{Lev}}(\mathrm{norm}(a), \mathrm{norm}(b))}{\max\big(|\mathrm{norm}(a)|, |\mathrm{norm}(b)|\big)},$$

where d_{Lev} denotes the Levenshtein edit distance and norm$(\cdot)$ applies lowercasing, whitespace normalization, and punctuation normalization. Missing or incompatible values receive a score of 0, yielding a global $m \times n$ scoring matrix.

Stage 2: Global Sorting. All pairs with $s(r_i^*, r_j) \geq \tau$ ($\tau = 0.3$) are sorted globally in descending order, ensuring high-confidence pairs are resolved first and cannot be preempted by lower-scoring candidates.

Stage 3: Double-Locking Assignment. Pairs are assigned greedily from the ranked list. Once (r_i^*, r_j) is matched, both indices are locked and excluded from further assignment, enforcing strict one-to-one mapping and preventing score inflation via duplicate row generation.

Stage 4: Cell-Level Settlement. Within each matched pair, scoring proceeds cell-by-cell as defined in Section 5. Unmatched ground-truth rows contribute all non-empty cells to the missing count; unmatched predicted rows contribute to the hallucination count, ensuring every cell is accounted for exactly once.

C.2 Prompts for LLM Score.

LLM-as-a-Judge: Database Extraction Evaluation Prompt

You are a database evaluation expert. Please evaluate the quality of database tables extracted from documents.

Evaluation Dimensions:

- **Accuracy:** Semantic consistency, exact numeric matching (including decimal places), and string similarity $\geq 90\%$.
- **Completeness:** Presence of all key information from the standard answer and structural integrity.
- **Standardization:** Compliance with Schema definitions and data type/format consistency.

Scoring Constraints:

- Numeric values must be *exactly equal*; string similarity below 90% is a mismatch.
- Type mismatches are considered serious errors.
- Total score ranges from 0 to 100; a perfect match equals 100 points.

Output Format: Provide a comprehensive explanation ($\leq$ 100 words), then output the score as:
`Evaluation evidence: [explanation] Rating: [[score]]`

D More Doc2DB-Bench Evaluation

D.1 Database Integrity Analysis.

Metric. Beyond attribute accuracy, we evaluate whether models can construct structurally complete and constraint-consistent databases. We measure *Entity Coverage* and *Relation Coverage* to assess the recovery of entities and inter-entity relationships, respectively. We further report *Reference Integrity Rate*, which measures the proportion of foreign-key references that correctly point to existing entities, reflecting the validity and consistency of the constructed database.

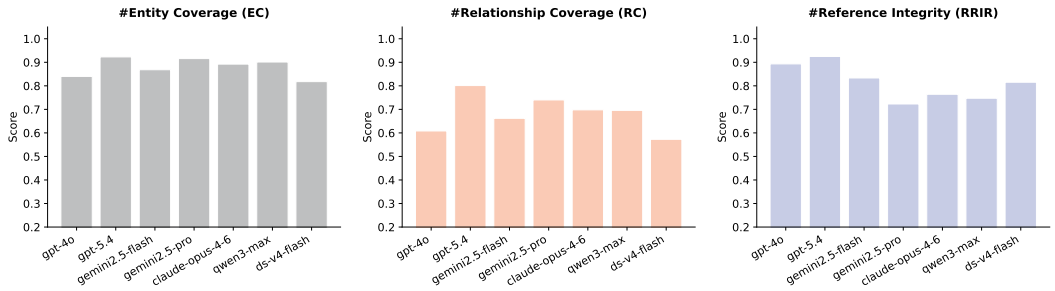


Figure 9: Database-level performance across Entity Coverage, Relation Coverage, and Reference Integrity Rate.

Result. Database-level results show that reconstruction requires both structural coverage and valid cross-table references. Entity Coverage exceeds Relation Coverage by 12.17–24.52 points across all models, highlighting the greater difficulty of recovering inter-table relationships. Coverage also does not guarantee validity: Gemini-2.5-Pro achieves strong coverage but the lowest RRIR, revealing frequent cross-table reference errors. GPT-5.4 leads all three metrics, demonstrating the strongest ability to recover relational structures while preserving database consistency.

Finding 4. Database construction requires more than accurate extraction: maintaining relational consistency and valid entity references remains a critical challenge for LLMs.

D.2 Full Results of Different Domains

Table 6: Results on Doc2DB-Bench across various domains. *Ver./Size* shows model version or size.

Methods	Ver. /Size	Domain							Overall Acc.
		Edu.	Fin.	Med.	Uni.	Sports	Trans.	Others	
Open-source Models									
Qwen2.5	14B-Ins	13.25	34.21	37.35	36.44	18.16	47.31	34.82	32.97
LLaMA-3.1	70B-Ins	10.94	13.18	16.29	21.54	27.86	7.30	8.16	14.29
Qwen2.5	72B-Ins	30.41	34.79	40.98	47.73	31.87	38.80	29.98	36.07
Proprietary Models									
GPT-4o	-	47.89	57.49	54.20	72.32	62.75	68.02	51.11	59.05
GPT-5.4	-	67.67	**68.65**	66.68	74.78	**82.24**	85.22	**78.49**	**75.25**
Gemini 2.5	Flash	57.63	64.04	41.00	73.13	71.22	78.95	64.57	65.99
Gemini 2.5	Pro	**69.81**	65.46	**76.53**	79.75	69.67	74.78	67.94	70.81
Claude Opus	4-6	67.01	65.38	75.51	**80.01**	78.36	**87.83**	69.89	73.60
Qwen3	Max	52.06	62.55	56.37	72.15	57.24	70.73	64.10	52.80
DeepSeek-V4	Flash	53.66	59.39	48.23	55.60	57.79	73.56	51.80	57.06

The results reveal a substantial performance gap between open-source and proprietary models. Among open-source models, Qwen2.5-72B achieves the highest overall accuracy of 36.07, but still trails GPT-5.4 by 39.18 points. GPT-5.4 obtains the best overall accuracy of 75.25 and leads in finance, sports, and other domains. Nevertheless, no single model dominates across all domains: Gemini 2.5 Pro performs best in education and medicine, while Claude Opus 4-6 leads in university and transportation, suggesting that different domains place distinct demands on schema interpretation, cross-table reasoning, and domain-specific information normalization.

Domain-level performance can be understood from the normalized dataset statistics. Averaged across all evaluated models, transportation achieves the highest accuracy (63.25), followed by university (61.35), whereas education obtains the lowest accuracy (47.03). Education contains the largest number of rows per database (276.7), together with 8.0 tables and 4.0 relation tables on average, requiring models to recover a relatively large number of tuples while preserving cross-table associations. Finance is challenging, but primarily because of its relational density rather than its absolute size: each database contains 6.2 tables and 3.9 relation tables on average, with the highest relation-to-entity ratio of 1.70. Medical databases present a different difficulty, having the largest average numbers of columns (66.3) and cells (1,174.0) per database, which increases the burden of broad attribute coverage. In contrast, although transportation documents are the longest on average (53,091 tokens), their databases contain only 3.6 tables and 1.2 relation tables per database, suggesting that document length alone is not the dominant source of difficulty. Overall, the results indicate that performance is more strongly influenced by the interaction of tuple volume, schema width, relation density, and implicit cross-table dependencies than by document or database scale in isolation.

E IMPLEMENTATION OF DOC2DB

During the entire *Controllable DB2Doc Reverse-Synthesis* process, we design specific prompts and rigorous verification protocols across four key agentic modules to ensure generation fidelity and relational complexity.

Firstly, in the capability assignment stage, the **Labeling Agent** acts as a semantic router. We construct prompts to strategically assign fine-grained intra-table (Pillar 1) and inter-table (Pillar 2) capability labels, which directly control the downstream extraction difficulty. Secondly, during the evidence decomposition stage, the **Refiner Agent** processes the annotated tables into atomic evidence. To guarantee the mathematical and semantic rigor of this step, we implement a strict *checklist-based verification* protocol. We design prompts that explicitly instruct the model to verify four critical dimensions: Value Correctness, Label Faithfulness, Schema Linkage, and Null Faithfulness. Thirdly, to bridge the gap between discrete evidence and cohesive natural language, the **Writer Agent** executes the document generation. We adopt a context-aware generation paradigm where the prompt incorporates a sliding window of historical text and configurable stylistic profiles to emulate real-world document morphological complexity Finally, in order to obtain a high-quality, hallucination-free corpus, the **Validator Agent** conducts a rigorous dual-validation pass. We construct prompts to evaluate Coverage Verification and Extraction Consistency. Documents that fail these checks trigger an iterative reject-and-rewrite mechanism.

E.1 THE PROMPT FOR LABELING AGENT

Pillar 1 Assignment Prompt

Goal: Assign intra-table (Pillar 1) capability labels to target cells in the entity table.

Inputs:
- **Input Table:** {markdown_table}
- **Non-target Key Cols:** {non_target_key_columns}
- **Attribute Schema:** {attribute_descriptions}
- **Capability Definitions:** {capability_definitions}
- **Target Label Ratio:** {label_ratio}

Instructions:
1. Assign labels that increase single-table extraction difficulty while preserving correctness.
2. **Type-Strategy Compatibility (CRITICAL):** Independently analyze the semantic nature and data type of each attribute. Only assign strategies logically/mathematically suitable for the cell's value.
3. A cell may have 0, 1, or multiple strategies (default is none).
4. Assign creatively but ensure text remains fluent. Avoid over-complicating simple facts.
5. DO NOT assign labels to Non-target Key Columns.
6. For composite primary keys, use `", "` as separator.
7. Label coverage: at least the target ratio of assignable cells must have non-empty labels.
8. Return JSON only.

Output JSON Schema:

```
{"assignments": {"<pk_1>":
{"<attr_1>": []} } }
```

Pillar 2 Assignment Prompt

Goal: Assign inter-table (Pillar 2) capability labels to target cells in the relation table.

Inputs:
- **Input Table:** {markdown_table}
- **Table Schema Context:** {relation_context}
- **Attribute Schema:** {attribute_descriptions}
- **Capability Definitions:** {capability_definitions}
- **Target Label Ratio:** {label_ratio}

Label Assignment Constraints:
- `IDR_ED`, `RL_O2M`, `RL_MB`, and `RL_MI` must each appear alone. Do NOT combine them.
- `IDR_ED`, `RL_O2M` requires that the same foreign key entity appears in ≥ 2 rows. Do NOT assign if unique.

Instructions:
1. Assign labels that increase inter-table linking, topology, and structural extraction difficulty.
2. **Topology-Strategy Compatibility (CRITICAL):** Analyze structural distribution. Only assign relational strategies logically suitable for the topology.
3. Output ONLY row-level labels per relation row_key. Do NOT output per-attribute labels.
4. A row may have 0, 1, or multiple strategies (default is none).
5. Top-level key MUST use the relation table PK string.
6. Composite keys format: `<pk1>, <pk2>` (e.g., `12, 10`).
7. Label coverage: at least the target ratio of rows must contain non-empty labels.
8. Return JSON only.

Output JSON Schema:

```
{"assignments": {"<pk_1>": [],
"<pk_2>": []} }
```

The prompt above illustrates how the Labeling Agent acts as a semantic router to assign fine-grained intra-table (Pillar 1) and inter-table (Pillar 2) capability labels. It explicitly instructs the model to analyze data types and topological contexts, ensuring that specific reasoning challenges are injected logically to control downstream extraction difficulty while adhering to target label ratios.

E.2 Refiner: Principle of Checklist

Given the annotated targets, the Refiner decomposes each target into atomic evidence that realizes its assigned capability labels, forming a global evidence pool for subsequent document generation. When multiple labels are assigned, the evidence must jointly satisfy the corresponding intra-table and inter-table reasoning requirements. To ensure the rigor of the synthesized evidence, we deploy an LLM-as-a-verifier (Zheng et al., 2023) using a strict checklist-based protocol. The verification assesses the generated evidence guidance across four critical dimensions:

- **Value and Relation Correctness:** Verifies that the guidance accurately reflects target cell values and inter-record relations, ensuring numerical correctness for calculated fields and allowing complete relational tuples to be unambiguously recovered.
- **Label Faithfulness:** Verifies that the guidance faithfully instantiates the assigned capability labels and rejects cases that omit or misrepresent the intended reasoning requirements.
- **Schema Linkage:** Ensures that the guidance is grounded in valid schema elements and preserves the intended structural dependencies among tables. For relational tables, primary- and foreign-key references must be correctly identified and linked, so that participating entities can be unambiguously connected and the resulting evidence remains consistent with the target database schema.
- **Data Integrity & Null Faithfulness:** Ensures that the guidance contains only information supported by the target database and does not introduce values from unrelated cells or records. It also verifies that missing, unavailable, or non-applicable attributes remain uninstantiated, preventing the generation of unsupported values or spurious relational links.

On failure, the verifier acts as a critic, returning a specific rationale and actionable revision suggestions (e.g., "Instruct the writer to explicitly mention both the supplier and project names"). This process iteratively refines the text until it passes all checks or reaches a retry limit.

E.3 The prompt for Writer Agent

Writer Agent Prompt

[System Role & Objective]
You are an expert Writer Agent operating within a parameterized document synthesis pipeline.
Your goal is to write ONE coherent, polished, and natural-sounding section of a larger document while perfectly preserving structured ground-truth facts.

[Generation Context]

- Previous Block Context (Tail Anchor): {previous_context}
- Current Block ID: {block_id} (Index: {block_index} of {total_blocks})

[Base Parameters]

- Global Target Document Length: {document_length_tokens} tokens
- Suggested Token Budget for THIS Block: {target_block_tokens} tokens
- Document Style & Tone: {document_style}
- Section Template Hint: {section_template}

[Injection Protocols]

- Hard Cases Protocol: {hard_cases_protocol}
- Controlled Noise Protocol: {noise_protocol}
- Representation Complexity Protocol: {complexity_protocol}

[Mandatory Ground Truth (Evidence)]
You MUST explicitly embed the following structured facts. Do NOT drop any fact: {facts_with_tags}

[Execution Instructions]

1. FACTUAL FIDELITY: Embed all mandatory facts without altering their meaning.
2. TAG PRESERVATION: Keep facts wrapped in their original tags (*e.g.,* `<rX_cY>`).
3. STYLISTIC ADAPTATION: Match the requested style, tone, and template.
4. PROTOCOL COMPLIANCE: Strictly follow any active injection protocols; ignore empty ones.
5. LENGTH & PACING: Stay close to the {`target_block_tokens`} budget.
6. SEAMLESS BRIDGING: Continue logically and smoothly from the previous context.
7. FORMATTING: Use natural prose paragraphs. No bullets, markdown, or JSON.
8. OUTPUT STRUCTURE: Start exactly with the Section Title, followed by the body.
9. FINAL OUTPUT: Return ONLY the synthesized text, without conversational filler.

The prompt above demonstrates how the Writer Agent transforms serialized atomic evidence into cohesive, long-form natural language documents. It guides the model to employ a context-aware generation paradigm, seamlessly weaving discrete facts into fluent narratives while conforming to designated stylistic profiles and mimicking the morphological complexity of real-world corpora.

E.4 The prompt for Validator Agent

Verification Agent Prompt

Your Goal:
Verify whether the generated text is faithful to the table content without fabrication.

Table Content: {markdown_table}

Generated Text: {generated_text}

Verification Checks:
To ensure **Cell Extraction Consistency**, Key values stated in `generated_text` must be extractable or inferrable from `Table Content`.

Output Format:
Respond with ONLY a single, valid JSON object:

```
{
  "ok": true/false,
  "errors": [
    {
      "description": "Error description",
      "suggestion": "How to fix"
    }
  ]
}
```

FINAL INSTRUCTION: Output ONLY the valid JSON object. If any check fails, set `ok` to false and include all issues in `errors`.

Repair Agent Prompt

Goal: Repair a section of text that failed verification.

Input:

- **Original Text:** {content}
- **Errors:** {errors}
- **Required Facts:** {facts}

Instructions:

1. Repair the section to fix the reported errors.
2. Ensure all required facts remain correct.
3. Keep the narrative flow natural.
4. Preserve all provenance-style tags already present in the original text.

Output Format:
Return ONLY the repaired text content. No titles, no metadata, no markdown code blocks.

The prompt directs the Validator Agent to check evidence coverage and extraction consistency. Detected omissions or unsupported facts trigger iterative rejection and rewriting, improving the fidelity and relational consistency of synthesized documents.

F Authenticity Verification

F.1 LLM & Human Score

To assess whether Doc2DB-Bench documents resemble authentic domain texts rather than templates, we combine LLM and human evaluation, following prior work (Babaei Giglou et al., 2024; ZJU-DAILY, 2026). Documents are rated on four dimensions using five-point rubrics: *lexical richness*,

Table 7: Evaluation rubrics for generated documents (1=worst, 5=best).

Score	Lexical Richness	Logical Consistency	Textual Coherence	Genre Fidelity
1	**Repetitive:** Minimal variety, robotic repetition	**Fragmented:** No connections, random claims	**Incoherent:** Difficult to follow, random jumps	**Mismatch:** Incorrect genre conventions
2	**Limited:** Simple vocabulary, narrow range	**Weak:** Loose transitions, vague reasoning	**Poor flow:** Jarring transitions, disconnected	**Weak:** Limited and inconsistent cues
3	**Acceptable:** Adequate variety, standard usage	**Acceptable:** Clear order, basic signaling	**Acceptable:** Some awkward transitions	**Plausible:** Recognizable but generic style
4	**Versatile:** Natural synonyms, precise terminology	**Compelling:** Strong arguments, coherent progression	**Smooth:** Clear progression, minor issues	**Authentic:** Consistent genre-specific conventions
5	**Sophisticated:** Rich nuances, professional mastery	**Rigorous:** Flawless chain, seamless attribution	**Seamless:** Natural flow, effortless transitions	**Professional:** Realistic and polished genre presentation

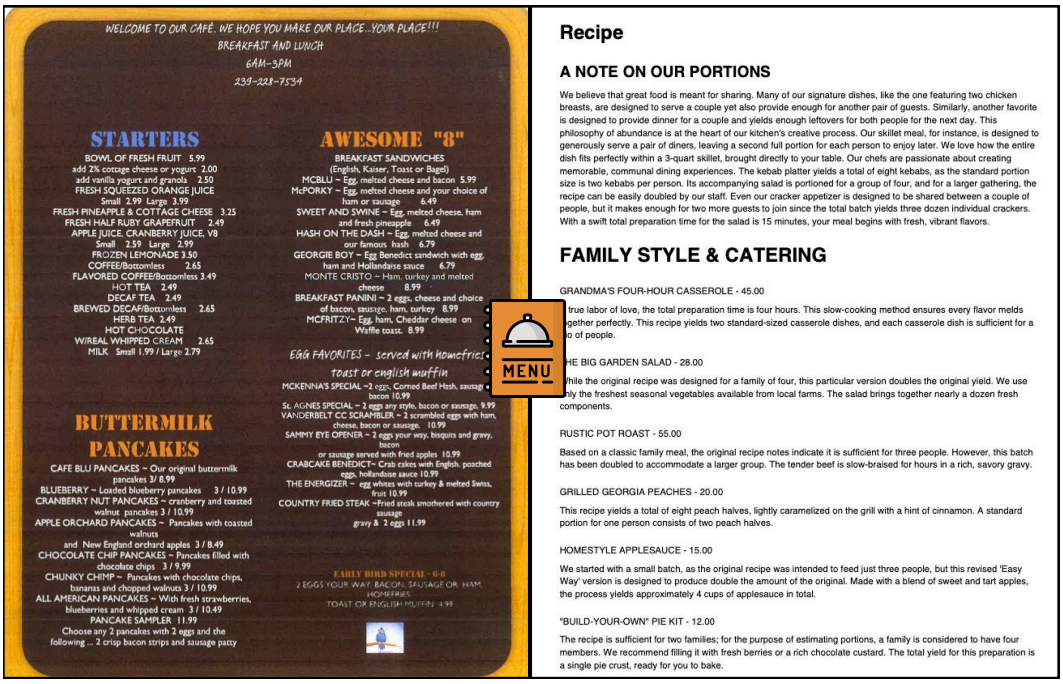
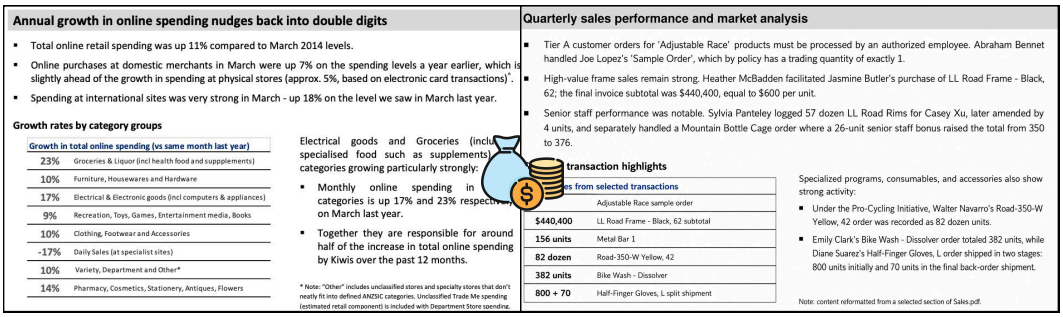


Figure 10: menu case

logical consistency, *textual coherence*, and *genre fidelity*, as shown in Table 7. DeepSeek-V3.2 serves as the automatic evaluator, while five graduate students experienced in document intelligence research independently assess 50 randomly sampled documents using the same rubrics. Table 5 demonstrates that both evaluations indicate high linguistic quality and domain authenticity.

F.2 Real Case Comparison

Figs. 10–13 compare reference documents (left) with synthesized documents (right) across finance, aviation, media, and restaurant menus, illustrating the quality and realism of our synthesis pipeline.



Annual growth in online spending nudges back into double digits

- Total online retail spending was up 11% compared to March 2014 levels.
- Online purchases at domestic merchants in March were up 7% on the spending levels a year earlier, which is slightly ahead of the growth in spending at physical stores (approx. 5%, based on electronic card transactions)*.
- Spending at international sites was very strong in March - up 18% on the level we saw in March last year.

Growth rates by category groups

Growth in total online spending (vs same month last year)	
23%	Groceries & Liquor (incl health food and supplements)
10%	Furniture, Housewares and Hardware
17%	Electrical & Electronic goods (incl computers & appliances)
9%	Recreation, Toys, Games, Entertainment media, Books
10%	Clothing, Footwear and Accessories
-17%	Daily Sales (at specialist sites)
10%	Variety, Department and Other*
14%	Pharmacy, Cosmetics, Stationery, Antiques, Flowers

Electrical goods and Groceries (inclu… specialised food such as supplements) … categories growing particularly strongly:

- Monthly online spending in … categories is up 17% and 23% respectiv… on March last year.
- Together they are responsible for around half of the increase in total online spending by Kiwis over the past 12 months.

* Note: "Other" includes unclassified stores and specialty stores that don't neatly fit into defined ANZSIC categories. Unclassified Trade Me spending (estimated retail component) is included with Department Store spending.

Quarterly sales performance and market analysis

- Tier A customer orders for 'Adjustable Race' products must be processed by an authorized employee. Abraham Bennet handled Joe Lopez's 'Sample Order', which by policy has a trading quantity of exactly 1.
- High-value frame sales remain strong. Heather McBadden facilitated Jasmine Butler's purchase of LL Road Frame - Black, 62; the final invoice subtotal was $440,400, equal to $600 per unit.
- Senior staff performance was notable. Sylvia Panteley logged 57 dozen LL Road Rims for Casey Xu, later amended by 4 units, and separately handled a Mountain Bottle Cage order where a 26-unit senior staff bonus raised the total from 350 to 376.

transaction highlights

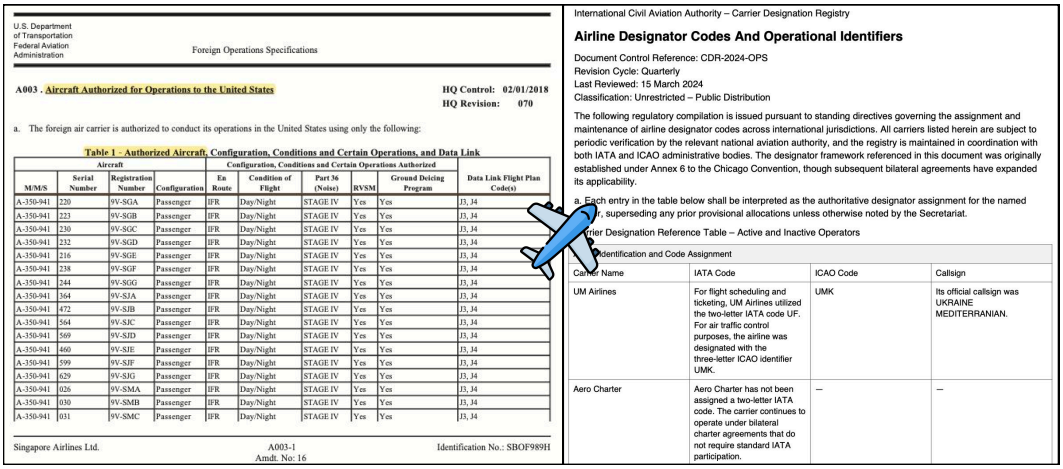
…s from selected transactions	
	Adjustable Race sample order
$440,400	LL Road Frame - Black, 62 subtotal
156 units	Metal Bar 1
82 dozen	Road-350-W Yellow, 42
382 units	Bike Wash - Dissolver
800 + 70	Half-Finger Gloves, L split shipment

Specialized programs, consumables, and accessories also show strong activity:

- Under the Pro-Cycling Initiative, Walter Navarro's Road-350-W Yellow, 42 order was recorded as 82 dozen units.
- Emily Clark's Bike Wash - Dissolver order totaled 382 units, while Diane Suarez's Half-Finger Gloves, L order shipped in two stages: 800 units initially and 70 units in the final back-order shipment.

Note: content reformatted from a selected section of Sales.pdf.

Figure 11: finance case



U.S. Department of Transportation
Federal Aviation Administration

Foreign Operations Specifications

A003 . Aircraft Authorized for Operations to the United States

HQ Control: 02/01/2018
HQ Revision: 070

a. The foreign air carrier is authorized to conduct its operations in the United States using only the following:

Table 1 - Authorized Aircraft, Configuration, Conditions and Certain Operations, and Data Link

Aircraft			Configuration, Conditions and Certain Operations Authorized						
M/M/S	Serial Number	Registration Number	Configuration	En Route	Condition of Flight	Part 36 (Noise)	RVSM	Ground Deicing Program	Data Link Flight Plan Code(s)
A-350-941	220	9V-SGA	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	223	9V-SGB	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	230	9V-SGC	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	232	9V-SGD	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	216	9V-SGE	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	238	9V-SGF	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	244	9V-SGG	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	364	9V-SJA	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	472	9V-SJB	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	564	9V-SJC	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	569	9V-SJD	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	460	9V-SJE	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	599	9V-SJF	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	629	9V-SJG	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	026	9V-SMA	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	030	9V-SMB	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4
A-350-941	031	9V-SMC	Passenger	IFR	Day/Night	STAGE IV	Yes	Yes	J3, J4

Singapore Airlines Ltd. A003-1 Amdt. No: 16 Identification No.: SBOF989H

International Civil Aviation Authority – Carrier Designation Registry

Airline Designator Codes And Operational Identifiers

Document Control Reference: CDR-2024-OPS
Revision Cycle: Quarterly
Last Reviewed: 15 March 2024
Classification: Unrestricted – Public Distribution

The following regulatory compilation is issued pursuant to standing directives governing the assignment and maintenance of airline designator codes across international jurisdictions. All carriers listed herein are subject to periodic verification by the relevant national aviation authority, and the registry is maintained in coordination with both IATA and ICAO administrative bodies. The designator framework referenced in this document was originally established under Annex 6 to the Chicago Convention, though subsequent bilateral agreements have expanded its applicability.

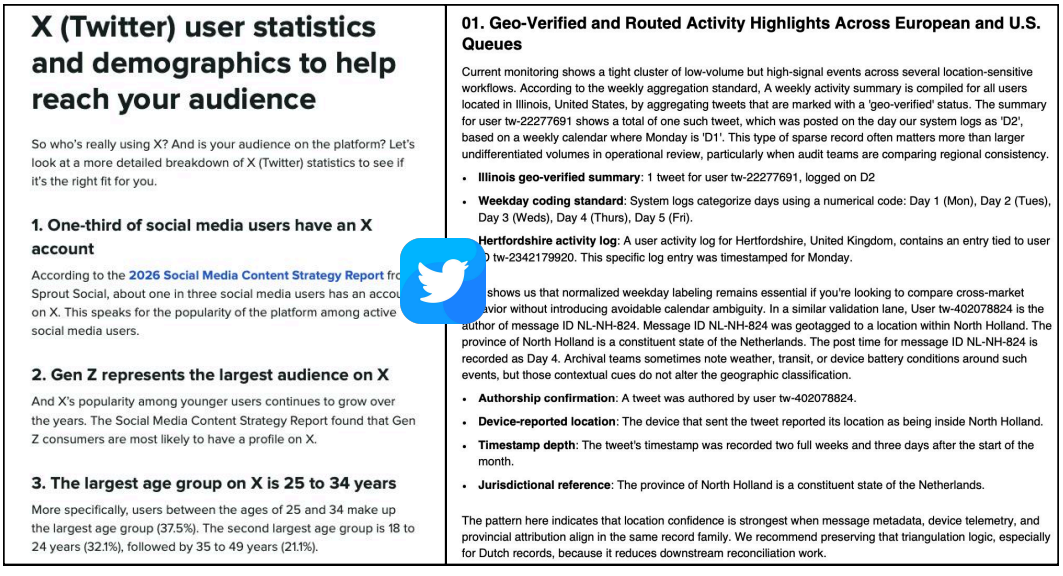
a. Each entry in the table below shall be interpreted as the authoritative designator assignment for the named …r, superseding any prior provisional allocations unless otherwise noted by the Secretariat.

…rier Designation Reference Table – Active and Inactive Operators

…dentification and Code Assignment

Carrier Name	IATA Code	ICAO Code	Callsign
UM Airlines	For flight scheduling and ticketing, UM Airlines utilized the two-letter IATA code UF. For air traffic control purposes, the airline was designated with the three-letter ICAO identifier UMK.	UMK	Its official callsign was UKRAINE MEDITERRANIAN.
Aero Charter	Aero Charter has not been assigned a two-letter IATA code. The carrier continues to operate under bilateral charter agreements that do not require standard IATA participation.	—	—

Figure 12: airline case



X (Twitter) user statistics and demographics to help reach your audience

So who's really using X? And is your audience on the platform? Let's look at a more detailed breakdown of X (Twitter) statistics to see if it's the right fit for you.

1. One-third of social media users have an X account

According to the **2026 Social Media Content Strategy Report** fr… Sprout Social, about one in three social media users has an accou… on X. This speaks for the popularity of the platform among active social media users.

2. Gen Z represents the largest audience on X

And X's popularity among younger users continues to grow over the years. The Social Media Content Strategy Report found that Gen Z consumers are most likely to have a profile on X.

3. The largest age group on X is 25 to 34 years

More specifically, users between the ages of 25 and 34 make up the largest age group (37.5%). The second largest age group is 18 to 24 years (32.1%), followed by 35 to 49 years (21.1%).

01. Geo-Verified and Routed Activity Highlights Across European and U.S. Queues

Current monitoring shows a tight cluster of low-volume but high-signal events across several location-sensitive workflows. According to the weekly aggregation standard, A weekly activity summary is compiled for all users located in Illinois, United States, by aggregating tweets that are marked with a 'geo-verified' status. The summary for user tw-22277691 shows a total of one such tweet, which was posted on the day our system logs as 'D2', based on a weekly calendar where Monday is 'D1'. This type of sparse record often matters more than larger undifferentiated volumes in operational review, particularly when audit teams are comparing regional consistency.

- **Illinois geo-verified summary**: 1 tweet for user tw-22277691, logged on D2
- **Weekday coding standard**: System logs categorize days using a numerical code: Day 1 (Mon), Day 2 (Tues), Day 3 (Weds), Day 4 (Thurs), Day 5 (Fri).
- **Hertfordshire activity log**: A user activity log for Hertfordshire, United Kingdom, contains an entry tied to user …0 tw-2342179920. This specific log entry was timestamped for Monday.

…shows us that normalized weekday labeling remains essential if you're looking to compare cross-market …avior without introducing avoidable calendar ambiguity. In a similar validation lane, User tw-402078824 is the author of message ID NL-NH-824. Message ID NL-NH-824 was geotagged to a location within North Holland. The province of North Holland is a constituent state of the Netherlands. The post time for message ID NL-NH-824 is recorded as Day 4. Archival teams sometimes note weather, transit, or device battery conditions around such events, but those contextual cues do not alter the geographic classification.

- **Authorship confirmation**: A tweet was authored by user tw-402078824.
- **Device-reported location**: The device that sent the tweet reported its location as being inside North Holland.
- **Timestamp depth**: The tweet's timestamp was recorded two full weeks and three days after the start of the month.
- **Jurisdictional reference**: The province of North Holland is a constituent state of the Netherlands.

The pattern here indicates that location confidence is strongest when message metadata, device telemetry, and provincial attribution align in the same record family. We recommend preserving that triangulation logic, especially for Dutch records, because it reduces downstream reconciliation work.

Figure 13: twitter case