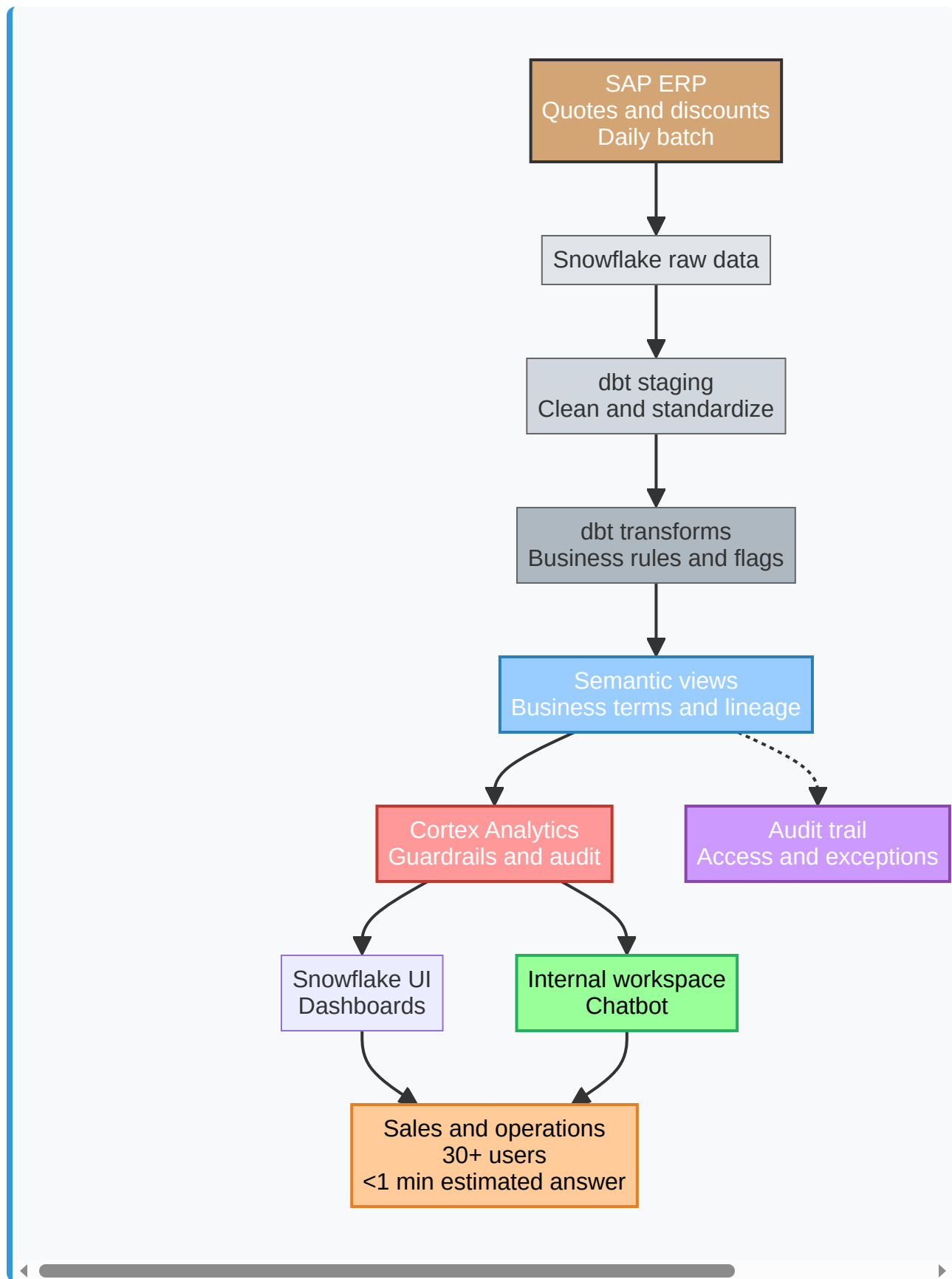


Talk-to-Data: Semantic Analytics Architecture

SAP → Snowflake → dbt → Cortex → Chatbot | <1 minute quote answers



SAP Source: Daily batch extract

Raw: Unmodified data

Staging: Data preparation

Transform: Business logic

 **Semantic:** Data contract **Cortex:** AI guardrails **Chatbot:** User interface **Audit:** Compliance trail

Data Flow Stages

1. Source

SAP ERP systems extract quotation, discount, and customer data daily at 05:00 UTC. Data includes quotation identifiers, line items, pricing, and discount conditions.

2. Raw

Snowflake External Stage ingests CSV/Parquet. Raw schema preserves source structure without modification. Data quality checks validate row counts, null patterns.

3. Staging

`stg_*` models rename columns, cast data types, and handle nulls. Example: source quotation ID → `quote_id`, source amount → `net_value`. No business logic yet.

4. Transform

`trf_quotation` joins customer policies, product portfolios, and agreement discounts. Calculates discount totals, applies governance flags (e.g., `discount_above_ceiling`). Two models: `trf_quotation` (BI) and `trf_chatbot_quotation` (LLM-optimized).

5. Semantic

Snowflake DDL semantic views expose curated dimensions and facts. Includes synonyms for natural language (e.g., "desconto cliente" → `discount_customer_pct`). Single source of truth for BI dashboards and LLM queries.

6. Cortex

[Cortex Analyst](#) applies guardrails: hides raw discount percentages, protects customer names, checks data freshness, enforces role-based access, and logs all access. LLM queries go through here.

7. Consume

Snowflake UI: Dashboards for ops/sales teams (full data visibility). **Internal Workspace:** Chatbot interface (guardrail-filtered data). Both consume the same semantic layer; governance rules ensure consistency.

Why This Architecture?

Governance-first: Guardrails (discount policies, protection of personal and customer information, and audit logs) are embedded in the data layer, not bolted on to the LLM prompt. The bot can't bypass policy; it's enforced by Cortex + dbt.

Semantic contract: The semantic layer (sv_quotation, sv_chatbot_quotation) defines the agreement between data engineers, BI teams, and LLM consumers. Column definitions, synonyms, and access rules are version-controlled.

Team efficiency: dbt lineage shows how every dimension flows from SAP → semantic layer. Data engineers and business analysts can trace a discount calculation back to SAP in seconds. Onboarding takes days, not weeks.

Prototyping speed: Snowflake + Cortex means we didn't build custom infrastructure. POC shipped in weeks. Cortex guardrails were ready on day 1; no custom policy engine to build.

Expected Outcomes

Latency: Quote research from 5-30 minutes (email + manual lookup) → <1 minute (chatbot query)

Scalability: 30+ salespeople can query the chatbot concurrently. Snowflake handles scale; dbt models are stateless (scale linearly).

Governance: All discount exceptions flagged and audited. Policy violations are visible and traceable.

Team capability: Once-daily batch + semantic layer pattern is reusable. Next use cases (contracts, pricing, serviceability) ship faster by leveraging the same playbook.

Learn More

See `README.md` for full narrative, decisions, and lessons learned. Check `dbt/` for anonymized SQL models. Review `cortex/cortex_governance_config.yaml` for guardrails and access control.