SAMPLE

Building Tomorrow's Data Engineers

The Lineage's Keeper in the Machine Era

Aiman JoeHanz

Free sample: Introduction & Chapter 1

putrabooks.github.io

Buy the full book:

Carousell carousell.sg/p/building-tomorrow-s-data-engineers-ebook-pdf-1471384945

Ko-fi

Payhip

Contents

Introduction: The Number and Where It Came From

Chapter 1 — The Lineage

Chapter 2 — The Territories

Chapter 3 — The Path

Chapter 4 — The Working Day

Chapter 5 — The Lineage

Chapter 6 — The Silence

Chapter 7 — The Conversation

Chapter 8 — The Instruments

Chapter 9 — The Year Whole

Chapter 10 — The Long Path

Introduction: The Number and Where It Came From

There is a moment in every organisation's life when somebody in a meeting says a number, and nobody in the room can say where it came from.

The number is in a dashboard. It has a decimal point and a trend line. It has been discussed in three previous meetings, and a decision was taken on the strength of it last quarter. And if you ask the room what it counts, which population it includes, what happens to an order that is refunded the following day, or whether it agrees with the figure the finance team keeps in its own system, the room will fall silent and then change the subject.

The person who can answer those questions is the subject of this book. Not the person who built the chart, and not the person who owns the database, but the person who holds the unbroken path from a source system to a number somebody makes a decision with. That path is what this trade keeps, and its name is lineage.

Lineage is not a diagram, although a diagram is how it is usually drawn. It is a claim about the world: that this figure was produced from these records by this transformation at this time, that it can be reproduced, and that if it is wrong, we will be able to establish where. An organisation with lineage can investigate itself. An organisation without it has opinions about its own numbers, and the opinions are held with confidence in inverse proportion to the evidence.

The temptation is to treat this as a technical problem, and it is partly one. There is real engineering in it: pipelines that recover from partial failure, transformations that produce the same result when run twice, tests that fail when a source changes shape, storage arranged so that history is available when somebody asks a question about last March. But the technical part is the smaller part. The larger part is a set of judgments about what to trust, what to publish, what to refuse, and how to say no to a number that somebody senior has already put in a slide.

This book describes three commitments that hold up against the pressure those judgments are made under.

Trace it or do not publish it. A number whose path cannot be reconstructed is not a fact about the business. It is a rumour with a decimal point. The work of establishing the path comes before the work of making the chart, which is the opposite of the order in which most organisations do them.

Assume the source will change without telling you. Upstream systems are not hostile and they are not careful. A column is renamed, a value becomes null where it was never null, a timestamp changes timezone, a table that used to contain one row per order begins to contain one row per order line. Every one of these is ordinary maintenance performed by somebody with their own priorities, and none of them will be announced. The estate that survives is the one that treats schema change as a certainty rather than an incident.

Make failure visible and repair repeatable. A pipeline that fails silently is worse than one that does not run, because the dashboard keeps working and the number quietly stops being true. The counterpart is that repair must be safe: rerunning a job must not double the revenue, and backfilling six months must not corrupt the tables that were fine.

The chapters that follow describe the territories this work occupies, the routes people take into it, an ordinary working day, the craft of the lineage itself, the particular exhaustion of maintaining something that is only noticed when it breaks, the conversations the trade requires, the instruments it depends on, one composite practitioner's whole year, and the long haul. They address a practitioner in year three or five who can already put together a pipeline and has started to suspect that putting together pipelines is not the job.

It is not. The job is being the person who can answer the question the room went quiet about.

Chapter 1 — The Lineage

The Thing That Makes a Number Usable

Every organisation produces numbers faster than it produces the ability to defend them, and the gap is where this trade lives.

A figure on a dashboard has three possible states. It can be a direct reading from a system of record, in which case it is only as trustworthy as that system. It can be a computation over records, in which case it carries the whole history of the computation. Or it can be a number somebody assembled by hand once and which has been carried forward in a slide ever since, in which case it is folklore and it will outlive the person who made it.

The trade's subject is the second case, because that is where almost everything an organisation reports actually comes from, and it is the case where trust has to be constructed rather than assumed. Constructing it means being able to say: this number came from these tables, filtered this way, joined on this key, deduplicated by this rule, at this time, with this known defect. Being able to say that is what makes a number usable in an argument. Being unable to say it is why so many reporting arguments end in an appeal to whoever is most senior in the room.

Two Folklores

Two stories circulate about what this work is, and both are wrong in ways that mislead people who are choosing whether to do it.

The first is the pipeline plumber. In this story the job is moving bytes: extract from here, land it there, schedule it, alert when it breaks. Success is a job that runs on time. The folklore is seductive because it is measurable, and because a great deal of the work genuinely does look like this on a Tuesday. Its error is that a pipeline which runs on time and produces a wrong number is indistinguishable, from the outside, from one that runs on time and produces a right one. The trade is judged by the second property and the folklore rewards the first.

The second is the dashboard magician. In this story the job is delivery: somebody asks a question, a chart appears, the executive is delighted. Speed of delivery is the craft, and the practitioner who ships fastest is the most valued. Its error is that a chart is a claim, and the magician has no way to defend the claim when it is challenged. When the number turns out to be wrong for six weeks, the magician has no answer, because the question was never whether the number was true.

The version that survives both is unglamorous: the job is keeping a number traceable, and delivery is the easy part once the traceability exists.

Three Commitments

The trade's discipline can be stated in three commitments, each of which is a commitment rather than a rule because applying it requires judgment.

Trace it or do not publish it. Before a figure reaches somebody who will act on it, its path must be reconstructable. The path does not have to be beautiful and it does not have to be automated, but it has to exist and it has to be written down somewhere a colleague could follow. A number nobody can trace is a claim about the business that the business cannot check.

Assume the source will change without telling you. The upstream system is maintained by people with their own deadlines, and their changes are correct from their point of view. A renamed column, a new status value, a timezone correction, a table that begins to hold a different grain: none of these are attacks and all of them break pipelines. The estate that survives is the one that treats drift as the normal condition rather than as an exception to be handled when it occurs.

Make failure visible and repair repeatable. Silence is the enemy. A job that stops producing is a problem somebody will notice; a job that keeps producing a stale or partial result is a problem nobody will notice until a decision has been made on it. And because failure is certain, repair has to be safe: rerunning must not double the revenue, and a backfill must be able to run without corrupting what was already correct.

What the Trade Is Not

The world draws this work's borders badly and in public, and the confusion taxes people in years, so plain words about the neighbours are owed.

An analyst answers questions with data and is measured on the quality of the answer. Their knowledge of the warehouse is often deeper than any engineer's, and much of what an engineer needs they already have. What nobody asks them to do is settle what a table means for everybody who will ever use it, or to be accountable for a pipeline that somebody else's decision depends on. The analyst's question is what does this show; the engineer's question is whether this will still be true when I am on leave. The distinction is not status, and an engineer who behaves as though it were will find quickly that the analyst knows which of the two hundred tables actually matter.

A software engineer builds behaviour. There is genuine overlap, and the best data platforms are built by people who write software properly. But the software engineer's world has a specification and a test that can close it. The data engineer's world has an input that will change shape next Tuesday without a release note, so the discipline is less about correctness at a point in time and more about surviving change over time.

A database administrator keeps the machine honest. Their understanding of storage, locking and recovery is deeper than almost anyone's and is exactly what the trade lacks when it is short. What nobody asks them to do is pin down what a row means, which is the question the trade is mostly about, and the transition from one to the other is harder than it looks because the second question has no correct answer to be found in a manual.

A business intelligence developer builds the presentation. In small organisations this is the same person, and the overlap is total. In large ones the distinction matters: the presentation can be rebuilt in a week, and the table underneath it cannot. The trade's leverage is in the table, and practitioners who move too early into presentation work find their influence shrinking as their visibility grows.

What the trade holds that none of these hold is accountability for the path. Every other role in the list is accountable for an artefact: an answer, a feature, a database, a dashboard. This one is accountable for a claim about how those artefacts connect, and that is why the work is more political than technical and why its failures are discovered by other people.

What Happens When a Number Is Wrong

The consequence of an untraceable number is not the number. It is what the organisation does next, and the pattern is consistent enough to describe.

First, the number is used. Somebody makes a decision with it: a budget, a headcount, a price, a campaign. The decision is rational given what was believed, which is why the eventual discovery is so uncomfortable for everyone involved, because nobody did anything unreasonable. This is the trade's version of an accident, and it is almost always a systems failure rather than an individual one.

Second, the discovery arrives sideways. It is rarely found by checking the number; it is found by reconciling it against something else, which means the discovery depends on somebody having an independent figure and the

standing to raise it. Organisations with no independent figure do not discover their errors; they discover them when a customer, an auditor or a regulator does.

Third, the investigation consumes far more than the number was worth. Establishing what happened requires reconstructing six weeks of transformations against a source that has since changed, and the cost of the archaeology is usually ten times the cost of the control that would have prevented it. The practitioner who proposes the control in advance will be told there is no budget; the same organisation will fund the investigation without a second meeting.

Fourth, something changes and something does not. The specific defect is fixed, because it is embarrassing and specific. The absence of a test that would have caught it is not fixed, because it is abstract and nobody owns it. A year later the same class of defect appears in a different pipeline and the cycle repeats with different people.

The practitioner's value is entirely in the fourth step. Anyone can fix the defect; the trade is the discipline of fixing the condition, and the cost of that discipline is that it looks like extra work right up until the moment it does not.

The Assumption and the Number

Every figure rests on assumptions that nobody wrote down, and the trade's real subject is those assumptions rather than the arithmetic.

A revenue number assumes that the source contains every sale, that refunds are represented consistently, that the currency conversion uses one rate rather than two, that the period boundaries are in the same timezone, and that a cancellation is either included or excluded by a rule somebody chose. Each of these is a belief about the world, most are true, some are true only for a while, and none are visible in the output.

The practitioner's work is to surface them. Not all of them, which is infinite, but the ones that are load-bearing and the ones that are probably false. The test for load-bearing is simple: if this assumption were wrong, would the number change materially. The test for probable falsity is harder and comes from experience of how source systems actually behave.

The most dangerous class is the assumption that was true and stopped being true without anybody noticing. A status value added upstream, a new region onboarded with a different date convention, a currency introduced, a product

line that reports its transactions twice. Nothing announced any of these and nothing failed, and the figure has been wrong since March.

This is why the trade's characteristic instrument is not a chart but a test, and why the test's value is in what it asserts about inputs rather than outputs. A test on a number catches a change after it has been published. A test on a column's accepted values catches it at the boundary, before anything downstream has been computed.

The practitioner who can list the five assumptions their most important figure depends on, with dates, is doing the work this book is about. One who cannot has a number that happens to be correct today.

Where the Difficulty Sits

None of the three commitments is technically hard in isolation, and the trade is difficult for a reason that is easy to misidentify.

The first difficulty is ordering. The pressure to publish arrives before the pressure to trace, every time, because the person asking for the number has a meeting on Thursday and the tracing would take until the following week. So the number ships and the tracing is deferred, and the deferral is never revisited because the number now works. Six months of this produces an organisation whose reports are load-bearing and undocumented, which is worse than having no reports, because the reports are being used.

The second is that lineage crosses boundaries nobody owns. A figure depends on a source owned by one team, a transformation owned by a second, a definition agreed by a third and a consumer in a fourth. Nobody holds the whole path, so nobody notices when it breaks, and the person who does notice is the analyst who has been asked why the figure moved.

The third is that the estate's failures are silent rather than loud. An outage produces a page; a wrong number produces nothing. The practitioner therefore works in a state of permanent unverifiable vigilance, able to know only that their checks have not fired, which is a different and harder posture than knowing that everything is up.

The fourth is that no milestone in this work announces the end. A pipeline can always be tested more, a definition always refined, a table always documented. Practitioners who cannot decide when something is good enough produce estates that are excellent in places and useless as a whole, and organisations that cannot decide the same thing produce practitioners who

burn out.

The difficulty, then, is not engineering. It is deciding what matters, in what order, with what evidence, and being willing to say so out loud to people who would prefer a number by Thursday.

The Habit Worth Building

The single habit that distinguishes practitioners who develop from those who plateau is unremarkable and can be started today.

The habit is to write, for every table you own, four lines: what a row means, how a row is identified, what the table deliberately excludes, and who would notice if it stopped updating. Four lines, at the top of the file, updated when the answer changes. It takes ten minutes and it converts private knowledge into something a colleague could use, which is the definition of an asset in this trade.

The second-order effect is larger than the artefact. Writing the four lines forces the practitioner to check each answer, and the check is where the discoveries are: the key that is not quite unique, the exclusion that was true two years ago, the consumer nobody can name. Practitioners report finding defects while writing the first four lines for tables they had maintained for years, because the questions had never been asked in that order.

The habit spreads without being mandated. A file with a header that answers those four questions is easier to review, easier to inherit and easier to change, and colleagues copy what reduces their own work. Nothing about it requires tooling, sponsorship or a programme.

The last property is that it survives. Programmes end, platforms are replaced, teams reorganise. Four lines at the top of a file persist through all of it, and a year of them is a lineage that no diagram can match, because a diagram asserts and a header with answers can be checked.

That is the habit worth building, and it is the only piece of advice in this book that can be acted on before the reader finishes the page.

The Metric That Cannot Be Gamed

Practitioners are frequently asked for a metric that proves the function's value, and the honest answer is that the good ones are unglamorous and the glamorous ones are corrupted within a quarter.

The metrics that get gamed are the ones attached to a person's attention. Number of tables documented produces documentation of tables that did not need it. Number of tests produces tests that pass by construction. Number of incidents resolved produces incidents split into smaller incidents. None of these are dishonest individually and the aggregate is distortion.

The metrics that resist gaming share a property: they require somebody outside the function to confirm them. A definition agreed with finance and used by finance is real. A table whose consumer count is non-zero is real. A figure in a formal report that reconciles to an independent source is real. Each is verified by a second party, which is what makes it expensive to fake.

The most useful single metric in this trade is time to answer a question about the past. How long does it take, from the request, to produce and defend a figure from six months ago. An estate that answers in an hour is understood; one that answers in a fortnight is not, and the difference is not visible in any inventory of tables.

The second is the count of figures the function can reconstruct on demand. Not tables, not tests, but published numbers whose path is written and can be walked by somebody who did not build it. Twelve is a good number for most organisations, and twelve is achievable in a year.

Neither metric appears in a dashboard and both change conversations, because they describe a capability rather than an activity.

The Day the Audit Arrived

Audits arrive on somebody else's schedule and reveal the estate's documentation more completely than any internal review.

The first request is always a list: which systems hold personal data, which reports are produced from them, who has access. An estate with a catalogue answers in an hour. One without spends a fortnight and produces something incomplete, and the incompleteness is itself a finding.

The second request is evidence for one control, chosen by the auditor, usually the one they expect to be weakest. The practitioner's response to that single sample determines the tone of everything that follows. A clean, dated artefact produces a short audit. An explanation produces a wider sample, which produces more explanations, which produces a report.

The third is the meeting at the end, where the findings are read out.

Practitioners commonly arrive intending to dispute and leave having agreed to a plan, which is the correct outcome. A finding is a request for a fix with a date; disputing it costs a fortnight and changes nothing, and agreeing to it converts an accusation into scheduled work.

The lesson worth keeping is that an audit is the only occasion on which somebody senior is guaranteed to ask the questions the practitioner has been asking for a year. The findings that emerge are usually the ones that were raised internally and not funded, and the audit supplies the funding. A practitioner who has kept a written list of what is known to be wrong can hand it over, which converts a year of ignored concerns into a remediation plan in a single meeting.

ABOUT THE AUTHOR

Data engineering, in Aiman JoeHanz's account, is the keeper of lineage: the trade that must be able to say where a number came from. This volume of the Building Tomorrow's People series follows a pipeline from the source system through the schema change that breaks it silently, the backfill that takes three days, the dashboard that was right last quarter, and the model that fails because the column it learned from stopped meaning what it used to. It is a book about plumbing in the sense engineers use the word.