Engineer

Business intelligence

turning a business's data into reports that guide decisions

What it is

Business intelligence, usually shortened to BI, is the work of turning an organisation's own records into reports, dashboards and analyses that the people who decide can act on. Used as an adjective it is often hyphenated, as in business-intelligence software.

The phrase is older than the software. In October 1958 the IBM Journal of Research and Development printed a paper by H. P. Luhn called "A Business Intelligence System". Luhn described a machine system that would take the documents an organisation receives or writes itself, abstract each one automatically, describe it by a pattern of its words, and send it to the "action points" whose profile of interests matched. An action point could be one person, a group, a department or a whole division. He gave business a deliberately wide meaning, any organised activity from science and commerce to law, government and defence, and took intelligence to mean grasping how the facts in front of you relate, well enough to steer action towards a goal. Having listed the techniques, he named the whole: "We call this a Business Intelligence System."

Luhn's material was text. BI today mostly works on numbers in tables, and Kimball Group's outline of a data warehouse and BI system shows the usual shape. A back room extracts data from the source systems, cleans it, makes it consistent and loads it. A front room presents it, organised by business process and kept at the most detailed level, with aggregated summaries added for speed. On top sit the BI applications: ad hoc queries, standard reports, dashboards and scorecards, and heavier analytic and data-mining applications.

What joins the two meanings is Luhn's question: who needs to know this, and how does it reach them in a form they will actually read?

Use the concept in a decision

Put “What is business intelligence?” to work in a real decision. We can help relate the concept to the system you are building or buying. Dashboards and reports showing the handful of figures a decision actually turns on, refreshed without being asked.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Brand communication informed by audience response — brand communication informed by audience response

Brand communication informed by audience response Get in touch →

Why it matters

Every organisation already runs on numbers. The question is whether those numbers agree. When sales counts an order on the day it is signed and finance counts it on the day it is invoiced, the two departments bring two revenue figures to the same meeting, and the meeting is spent on the difference instead of the decision. A BI system earns its cost by fixing one definition per measure, computing it in one place and showing the same figure to everyone who asks.

Luhn’s diagnosis from 1958 still fits most shared report folders. He saw organisations send information widely, just in case, and noted that the recipients drowned in paper, which made it likely that nothing got through at all. A dashboard with forty charts does the same thing on a screen. His remedy was selection. Each action point first received a short list identifying the chosen documents by number, name and author. It could then dial those numbers to get the abstracts, and accept the ones it wanted in full. Every acceptance updated its profile, so the selection improved with use. A good report today works the same way: it starts from a person and a decision, and shows the few figures that decision depends on.

There is a practical side too. The database behind an application is built for its many small reads and writes. A question that reads a year of orders competes with those writes when it runs in the same place, which is one reason BI data usually lives in a separate store loaded on a schedule. Keeping that store at the most detailed level, as the Kimball outline does, decides which questions can still be answered next year without going back to the sources.

What it looks like in practice

A typical BI question: revenue for the third quarter of 2026, by region and product, with a subtotal for each region and a grand total. Given a table `sales` with one row per sale, holding `sold_on`, `region`, `product` and `amount`, standard SQL answers it in one query:

```
SELECT region,
       product,
       sum(amount)              AS revenue,
       GROUPING(region, product) AS level
FROM   sales
WHERE  sold_on >= DATE '2026-07-01'
AND    sold_on <  DATE '2026-10-01'
GROUP  BY ROLLUP (region, product)
ORDER  BY GROUPING(region), region, GROUPING(product), product;
```

With a handful of sample rows it returns:

```
 region | product | revenue | level
--------+---------+---------+-------
 North  | Boots   |    1200 |     0
 North  | Jackets |     800 |     0
 North  |         |    2000 |     1
 South  | Boots   |     500 |     0
 South  | Jackets |    1500 |     0
 South  |         |    2000 |     1
        |         |    4000 |     3
```

`ROLLUP (region, product)` is shorthand for `GROUPING SETS ((region, product), (region), ())`: the full list, every shorter prefix of it and finally the empty set, which adds up all rows. The PostgreSQL documentation describes it as the usual tool for hierarchical data, such as salary by department, by division and for the whole company. `CUBE (region, product)` would group by every subset instead, and so add a total for each product across all regions.

In a row's grouping set, a column that is not grouped comes back as NULL, which is why the subtotal rows have blank cells. `GROUPING(region, product)` says which columns those are. It returns a bit mask with one bit per argument, the rightmost argument in the lowest bit, and a bit is 1 when that column was not grouped for

the row. So 0 marks a detail row, 1 a region subtotal (product not grouped) and 3 the grand total (neither grouped). A report reads `level` to decide which rows to print in bold.

The company's data analytics and BI dashboards service sets out its aim as dashboards and reports that show the handful of figures a decision actually turns on, refreshed without anyone having to ask. A query like this one is the layer under such a report: the totals are computed once, in the database, and the dashboard only draws them.

Where it goes wrong

Two figures under one name. Revenue, active customer and churn each have several reasonable definitions, and every report that computes its own picks one. Write each measure down once, with what it counts, from which source and as of which date, and compute it in the database, for example as a view, rather than in each report's formulas. Kimball calls the plain-language part business metadata: what data is available, where it came from and what it means.

Everything sent to everyone. A dashboard with dozens of charts and a mailing list of the whole department reproduces the flood Luhn wanted to stop. Start from a named person and the decision they make, show the figures that decision needs, and retire the reports that nobody has opened in a quarter.

Reports that slow the application down. A heavy query on the primary database of a live system competes with the customers using it. Run reporting on a read replica or in a separate warehouse loaded on a schedule. Aggregates that are expensive and asked for often can go into a materialized view that is refreshed after each load.

Totals built from totals. The average order value of each region, averaged across the regions, is not the company's average order value unless every region has the same number of orders. A count of distinct customers per month cannot be added up to a quarter, because a customer who bought in July and in August counts twice. Keep the detailed rows and aggregate each level from them. Grouping sets do exactly that: every level of the query above is computed from the sales rows, not from the level below it.

A blank that means two things. Add one sale whose region is unknown, and the query above returns a detail row with an empty region, a subtotal for it with both cells empty, and the grand total with both cells empty too. With `CUBE` it gets worse: two rows read empty region and Boots, one for boots sold where the region is unknown and one for boots across all regions. The PostgreSQL documentation warns that a NULL in the data cannot be told apart from the NULL of a grouped-away column. Test `GROUPING()`, never `IS NULL`, to find subtotal rows, and give missing values a visible label such as “unknown” when the data is loaded.

Yesterday’s figures read as today’s. When a nightly load fails, the dashboard keeps showing the last good numbers, and nothing on screen says they are old. Print the time of the last successful load on every report, and have monitoring raise an alert to a named person when a scheduled load does not finish.

Further reading

- A Business Intelligence System — H. P. Luhn’s paper in the IBM Journal of Research and Development, October 1958, where the term comes from.
- Kimball Technical DW/BI System Architecture — Kimball Group’s outline of the back room, the presentation area, the BI applications and the metadata.
- Table Expressions — PostgreSQL’s reference for `GROUPING SETS`, `ROLLUP` and `CUBE`, and how NULL stands in for an ungrouped column.
- Aggregate Functions — the `GROUPING` function and its bit mask, with a worked example.

Glossary

Choose for your actual constraints

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is business intelligence?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/business-intelligence/