Automated Data-Quality Checks

Project: Automated Data-Quality framework for the **HEXA VPN** app · **Built at:** Hazel Mobile · **Data shown:** synthetic / illustrative

> **Data-quality framework I built at Hazel Mobile for the HEXA VPN app.** Figures here are **synthetic / illustrative** for portfolio use — it shows the *method*, **not real client, user, or production data**.

What this is: a plain-language guide to an automated data-quality (DQ) framework I designed at Hazel Mobile for the **HEXA VPN** app's **connection/session telemetry**. It explains *what each check looks for* and *why it matters*, for a non-technical reader. Every check is **read-only** — it never changes the source data. **Why it matters:** a headline "success rate" often counts every session that *reported* success. Some of those aren't real wins — a session that dropped in seconds, a corrupt record, a bot, or a user who never came back. These checks find those cases and **reclassify** them, so the business sees the *true* health of the metric, not the flattering version. **The headline it produces (illustrative):** a raw **~92%** success rate can fall to a true **~68%** once the checks are applied — a ~24-point gap that this framework measures and explains.

How to read this

- **Three tiers of checks**, from "is the data clean?" up to "did the user actually have a good experience?":

1. **Row-level (#1–#11)** — is this single record clean and usable? 2. **Behavioural (B1–B10)** — is this *account* behaving like a normal human user? 3. **Journey (T1–T3)** — did the user actually get (and keep) a good session?

- **"Flag" / "Fail"** = the check caught something. Row-level = one bad record; behavioural/journey = a whole user or a fleet-wide alert.
- **"User"** = one account (identified by the account id inside the session token, so multiple devices/sessions roll up together).
- **"NOW"** = the most recent event in the data (not wall-clock), so windows like "last 7 days" are measured against the data itself.
- **"Fleet alert"** = a whole-product warning (e.g. "activation dropped below target"), not a per-record failure.

Tier 1 — Row-level checks (is this one record clean?)

Each looks at a **single record** and asks "is this a trustworthy row?"

#	Check	In plain words	It flags a record when…

#1	Timestamp validity	The clock is wrong — the event was "received" before it "happened".	received time is before the event time (or a timestamp is missing).
#2	Valid country code	The country isn't a real 2-letter code.	country is present but not a valid ISO code.
#3	Country present	The country is blank.	country is missing entirely.
#4	Non-negative duration	Session length is negative — impossible, so corrupt.	duration is below 0.
#5	Non-zero duration	Session length is exactly zero — it never really connected.	duration equals 0.
#6	Handshake completed	Too short to be a real connection (under a tenth of a second).	duration is under 100 ms.
#7	Valid server IP	A "success" has no real server address — a logging problem.	the row is a success but the server address is missing/invalid.
#8	Collection continuity	The data feed went silent — a possible outage.	there's a gap of more than 60 minutes before this record.
#9	Plausible session length	Too short for the user to have actually used it.	under 14s for ad-supported users, under 4s for others.
#10	Duration present	No recorded length, so it can't be judged.	the duration field is missing.
#11	Self-replace bug (new)	A "success" that was instantly replaced and the same user reconnected within a minute — a known app bug, not a real session.	the session ended with a "replaced" reason and the same user's next event is within 60 seconds.

> **Note:** "switching servers" is **not** a failure on its own — a single failover is normal. It only counts in **B10** (switched *and then* churned).

Tier 2 — Behavioural checks (is this account normal?)

These look at a **whole account** and ask "is this a genuine, healthy user?"

ID	Check	In plain words	It flags / alerts when…
B1	Return / repeat	How many users come back for a second session.	Fleet alert: repeat rate below target.
B2	One-session share	Share of users who connected once and never returned.	Fleet alert: one-session share above threshold.
B3	Tenure success-floor	New users whose success rate is too low for how many times they tried.	1st try < 70%, 2nd < 50%, or 3rd < 40% success.
B4	Very-churny segments	Segments where almost everyone is one-and-done.	any segment where the one-session share is 60%+.

B5	Inactivity churn (7d)	A user has gone quiet — no activity for a week.	no session in the last 7 days.
B6	Bot / burst deep-dive (new)	Traffic that doesn't look human: an automated burst, or a user stuck reconnecting.	**Likely-bot:** many sessions/minute across many servers, zero failures. **Or reconnect-churn:** repeated failures on just 1–2 servers.
B7	Rapid location hopping	One account appearing in many locations almost at once — impossible for one device.	more than 4 locations within 1 minute.
B8	Plan validity	The account's plan (free/premium) is missing or invalid.	plan field is null or out of range.
B9	Failure with no reason	A failure was logged but no reason recorded — a logging gap.	outcome is failure but the reason is blank.
B10	Churn-after-switch	A user who had to switch servers and then never came back.	switched at least once and then inactive for 7 days.

Tier 3 — Journey checks (did the user have a good experience?)

These judge the **real** experience — including the "fake successes" a plain success flag hides.

ID	Check	In plain words	It flags / alerts when…
T1	Flap	A "success" that dropped almost immediately — looked connected but didn't hold.	the session held for under 30 seconds. Fleet alert if flaps exceed 10% of successes.
T2a	Activation	Did the user's very first session actually work?	Fleet alert: activation below target.
T2b	D1 next-day return	Did the user come back the next day?	Fleet alert: next-day return below target (needs 2+ days of data).
T3	Churn-risk	Users trying a lot but mostly failing — about to quit.	5+ attempts and under 30% success.

What the framework produces (the "statements")

Running the full check suite outputs five sections. In plain terms:

Section	What it gives you
0 — Impact (before vs after)	The headline: events, users, successes, failures, and **success % before vs after** the checks. Splits the users we "lose" into *one-and-done* vs *retained*.
1 — Checks summary	One row **per check**: how many records/users it flagged, the rule in words, and a status (OK / FLAGGED / ALERT). A scoreboard.
2 — Self-check	A built-in audit: re-calculates key numbers a **second, independent way** and reports **PASS/FAIL**, so the results can be trusted.
3 — Per-check contribution	Ranks **which check caused how much** of the drop. Each flipped success is owned by one check, so the pieces add up.

4 — Record dumps	Pulls the **actual offending records** for each flagged check, so anyone can review or download the specific rows.

Trust note: every number is produced by SQL (not estimated), and the self-check re-verifies the maths, so the figures are exact and reproducible.

Bottom line (for a quick read)

- A reported **~92%** success rate is the *face-value* number. It counts flaps, instant self-replaces, bots, bad records, and never-returning users as "success".
- Applying these checks reclassifies those cases and gives a **true** success rate of **~68%** — a ~24-point gap.
- **One check dominates:** *inactivity churn (B5)* — users who connected but never came back — accounts for the large majority of the drop. The row-level integrity checks add most of the rest; the long tail catches smaller but genuine data problems.
- Everything is **read-only** and **self-validated**, so it's safe to run against a live database and the numbers can be trusted.

Method demonstration on illustrative data — no real client, product, or production figures are shown.