**NBA Performance Analytics: 2000–2025 — Project Report**

*SQL-first portfolio project built on a daily‑updated SQLite NBA database.*

**1) Executive Summary**

This project analyzes NBA team performance, clutch scoring, back‑to‑back effects, and draft value using a SQLite database (~64k games, ~4.8k players). The work is intentionally SQL‑heavy (no Python/R required) and reproducible in DBeaver/DB Browser. Core outputs include season win%, points for/against, B2B vs. rest win% splits, clutch points per player in Q4 last 5:00 of close games, and draft pick tier vs. career longevity.

**Key confirmed result (from this run):**

* **Average seasons played by draft tier** (distinct seasons with any play-by-play event):
  + **T1 (picks 1–10): 11.63**
  + **T2 (11–30): 8.90**
  + **T3 (31–60): 5.89**
  + **T4 (Undrafted): 7.72**  
    *Note:* Undrafted players who appear at all tend to be above‑average survivors, which explains their longevity vs typical second‑rounders.

Other metrics (team trends, clutch, B2B) are computed via the included SQL; insert the latest values/figures after exporting from DBeaver.

**2) Data & Schema**

**Primary tables used**

* line\_score — one row per game containing **both** teams’ IDs, city/nickname and totals (pts\_home, pts\_away), plus game\_date\_est.
* game — game metadata; may include season\_id, game\_date\_est, is\_playoffs/season\_type.
* play\_by\_play — event‑level data (game\_id, period, pctimestring, eventmsgtype, text descriptions, player1\_id/player\_id).
* player — player directory (id, full\_name, first\_name, last\_name).
* draft\_history — draft pick per player (person\_id, overall\_pick).
* *(Optional)* team\_info\_common — official team names; some historic teams may be missing.

**Important schema notes**

* Because line\_score stores **home & away in the same row**, analyses first "unpivot" into **one row per team per game** with flags (is\_home, pts\_for, pts\_against).
* Seasons are normalized as integers: COALESCE(game.season\_id, STRFTIME('%Y', date)).
* Team names are taken from team\_info\_common when available, else **fallback** to line\_score city + nickname.

**3) Methods (SQL Overview)**

**3.1 Team Trends**

* **Win % by season.** Unpivot line\_score → compute win flag pts\_for > pts\_against → average 1/0 by team\_id, season.
* **Avg Pts For/Against.** From unpivoted rows, average pts\_for and pts\_against by team\_id, season.

**3.2 Back‑to‑Back (B2B) vs Rest**

* Build per‑team game dates (gdate = DATE(SUBSTR(game\_date\_est,1,10))).
* Use a window function to compare each game’s date to the previous one:  
  is\_b2b = (JULIANDAY(gdate) − JULIANDAY(LAG(gdate))) = 1.
* Compare win% when is\_b2b=1 vs is\_b2b=0.

**3.3 Clutch Time (Q4 last 5:00 in close games)**

* **Close game filter:** ABS(pts\_home − pts\_away) ≤ 5 from line\_score.
* **Clutch window:** play\_by\_play.period = 4 and pctimestring ≤ 5:00 → parse mm:ss/m:ss into seconds.
* **Event scoring rules:**
  + eventmsgtype = 1 **+** text contains 3PT → **3** pts
  + eventmsgtype = 1 → **2** pts
  + eventmsgtype = 3 and text not like MISS → **1** pt
  + else **0**
* Aggregate to **clutch points per game** per player; require ≥10 clutch games.
* *(Optional)* Join game to label **Playoffs vs Regular**.

**3.4 Draft Value vs Career Longevity**

* Map overall\_pick → tiers: 1–10, 11–30, 31–60, undrafted.
* For each player in play\_by\_play, count **distinct seasons** (derived via game or line\_score) → seasons\_played.
* Aggregate **average seasons** per pick tier.

**3.5 Performance & Indexing**

* Create indexes for frequent joins/filters:  
  line\_score(game\_id), play\_by\_play(game\_id), play\_by\_play(player1\_id/player\_id), game(game\_id), game(game\_date\_est).

**4) Results (insert latest numbers/figures)**

Export each result grid to CSV via DBeaver → **Export**. Replace the placeholders below with your latest run.

**4.1 Team Win % by Season**

* **Top teams by season:** *(Insert table or image)*
* **Observation(s):** *(e.g., dominance windows, rebuild years, expansion effects)*

**4.2 Average Points For/Against**

* **League‑wide pace proxy:** rising avg\_pts\_for suggests faster era.
* *(Insert table/plot: avg\_pts\_for vs avg\_pts\_against by team-season)*

**4.3 Back‑to‑Back vs Rest**

* **Win% on B2B:** *(Insert %)*
* **Win% with Rest:** *(Insert %)*
* **Delta:** *(B2B − Rest)*
* *(Optional split)* home vs away on B2B.

**4.4 Clutch Time Leaders (Q4 last 5:00, close games)**

* **Top 25 clutch scorers (PPG), min 10 clutch games:** *(Insert table)*
* *(Optional)* separate Playoffs vs Regular to highlight high‑leverage performers.

**4.5 Draft Value vs Longevity (confirmed)**

|  |  |  |
| --- | --- | --- |
| **Pick Tier** | **Players** | **Avg Seasons Played** |
| T1 (1–10) | 763 | **11.63** |
| T2 (11–30) | 1,403 | **8.90** |
| T3 (31–60) | 2,016 | **5.89** |
| T4 (Undrafted) | 4,075 | **7.72** |

**Interpretation:** early picks last longest on average; undrafted who stick tend to be exceptional (selection bias). Second‑rounders, as a group, have shorter careers than undrafted survivors.

**5) Validation & QA**

* **Schema checks:** PRAGMA table\_info(...) for line\_score, play\_by\_play, game to confirm columns.
* **Sanity tests:** spot‑check 3–5 games: winner computed from pts\_home/away matches box score; clutch window correctly includes only Q4 last 5:00.
* **Clock parsing:** handle mm:ss and m:ss lengths.
* **Join robustness:** cast join keys to TEXT when types differ (CAST(a.id AS TEXT) = CAST(b.id AS TEXT)).

**6) Limitations**

* **Historic coverage:** some early‑era/defunct teams may be absent in team\_info\_common → fallback names are used.
* **Clutch heuristic:** uses text parsing; missed FTs and turnovers are handled, but shot type tagging depends on description strings.
* **B2B definition:** exactly 1 calendar day apart; does not account for travel distance/time zones.
* **Longevity proxy:** counts seasons with **any** PBP event, not minutes‑weighted career value.

**7) Future Work**

* Add **overtime clutch** (last 2:00 of each OT) and compare to Q4 clutch.
* Compute **net rating** by season from possession estimates.
* Build **aging curves** (per‑36 stats) if a per‑player box table is available.
* Enhance draft value with **career WAR/BPM × minutes** instead of seasons.
* Model B2B effects by **home/away + travel distance** (external schedule/geo data).

**8) Reproducibility**

1. Place nba.sqlite in /data and connect via DBeaver.
2. Run sql/99\_indexes.sql for speed (optional).
3. Run:
   * sql/01\_team\_trends\_win\_pct.sql
   * sql/01\_team\_trends\_points.sql
   * sql/01b\_b2b\_vs\_rest.sql
   * sql/02\_clutch\_time.sql
   * sql/03\_draft\_value.sql
4. Export result sets to /exports and build dashboards; add screenshots to /images.

**9) Appendix — Query Inventory (short)**

* **01\_team\_trends\_win\_pct.sql:** unpivot → win flag → average by team‑season.
* **01\_team\_trends\_points.sql:** unpivot → avg points for/against.
* **01b\_b2b\_vs\_rest.sql:** per‑team game dates → LAG → is\_b2b → win% comparison.
* **02\_clutch\_time.sql:** close games filter → Q4 last 5:00 → points per event → clutch PPG per player; optional season/playoffs.
* **03\_draft\_value.sql:** pick tier from draft\_history → seasons played from PBP → avg seasons per tier.