IBM Data Analyst Capstone Project

Utpoul Kumar MONDAL

14/09/2025

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

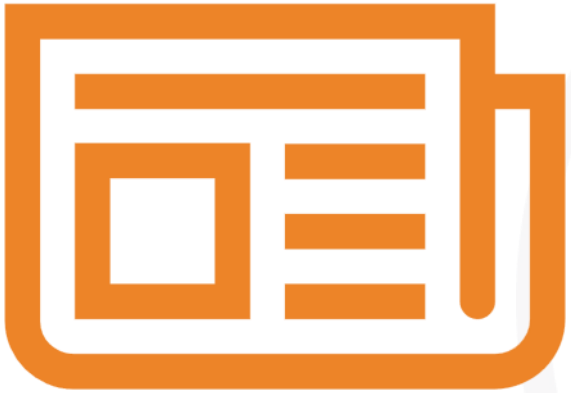
- The survey highlights who the respondents are (age, country, education) and what technologies they use today vs. what they want to adopt in the future.
- We have 3 major trends, adaptations and current streams to deal with:
  - Current Usage: Today's stack is dominated by JavaScript, SQL, HTML/CSS, with PostgreSQL, MySQL, and SQLite as top databases. AWS, Azure, Google Cloud lead the platforms, and React, Node.js dominate frameworks.
  - Future Trends: Respondents show strong desire for TypeScript, Rust, Go (languages with growth potential). PostgreSQL and MongoDB remain highly desired. Cloud continues to dominate (AWS, Azure, Google Cloud). Framework adoption is moving toward Next.js, React, and Vue.js.
  - Demographics: Majority of respondents are aged 25–34, heavily represented in countries like the US, with most holding a Bachelor's degree.
- Key insight: There's a gap between current usage and future interest—emerging technologies (e.g., Rust, Next.js) show much higher desire compared to current penetration.

INTRODUCTION

Major Key Points

- Context: The survey captures the **global developer landscape** across demographics, skills, and preferences.
- Purpose: To compare **current technology usage** with **future aspirations**, helping stakeholders identify adoption trends, training needs, and market opportunities.
- Scope: Three dashboards:
  - *Demographics* (who the respondents are)
  - *Current Usage* (what technologies are used today)
  - *Future Trends* (what technologies developers want to work with next year)

METHODOLOGY



Key Methodologies Found and Utilised

- **Data Source**: Survey dataset (survey_data_updated.csv).
- **Processing**:
  - Multi-select fields (languages, databases, platforms, frameworks) split and normalized in **BigQuery**.
  - Nulls cleaned, coding years converted to numeric, and “Top 10” pre-aggregated for clarity.
- **Visualization**:
  - Built in **Google Looker Studio**.
  - Demographics: Pie, Map, Bar + Line, Stacked Bar.
  - Usage/Trends: Bar, Column, Word Cloud/Treemap, Bubble Charts.
- **Metrics**: Primarily Record Count (number of respondents mentioning each technology), with derived percentages for distribution clarity.

RESULTS

1. Demographics

- **41%** are aged **25–34**.
- Respondents are widely distributed, with strong clusters in North America and Europe.
- **Bachelor's degree holders** form the majority, followed by Master's.
- Education distribution varies by age: younger respondents skew toward "Some college" while older respondents show more graduate degrees.

2. Current Technology Usage

- **JavaScript (≈15K mentions)** is the most widely used language, followed by SQL and HTML/CSS.
- **PostgreSQL, MySQL, and SQLite** dominate databases.
- **AWS, Azure, Google Cloud** are the top cloud platforms, far ahead of smaller providers.
- **React and Node.js** are the leading frameworks.

RESULTS

3. Future Technology Trends

- **TypeScript, Rust, and Go** show rising interest—highlighting the trend toward safer, modern languages.
- **PostgreSQL remains strong**, but **MongoDB and Redis** are gaining more traction in future plans.
- **AWS, Azure, Google Cloud** continue to dominate platforms, but interest in **Cloudflare and Firebase** is also noticeable.
- Framework preferences lean toward **Next.js, React, and Vue.js**, showing momentum for full-stack JavaScript ecosystems.

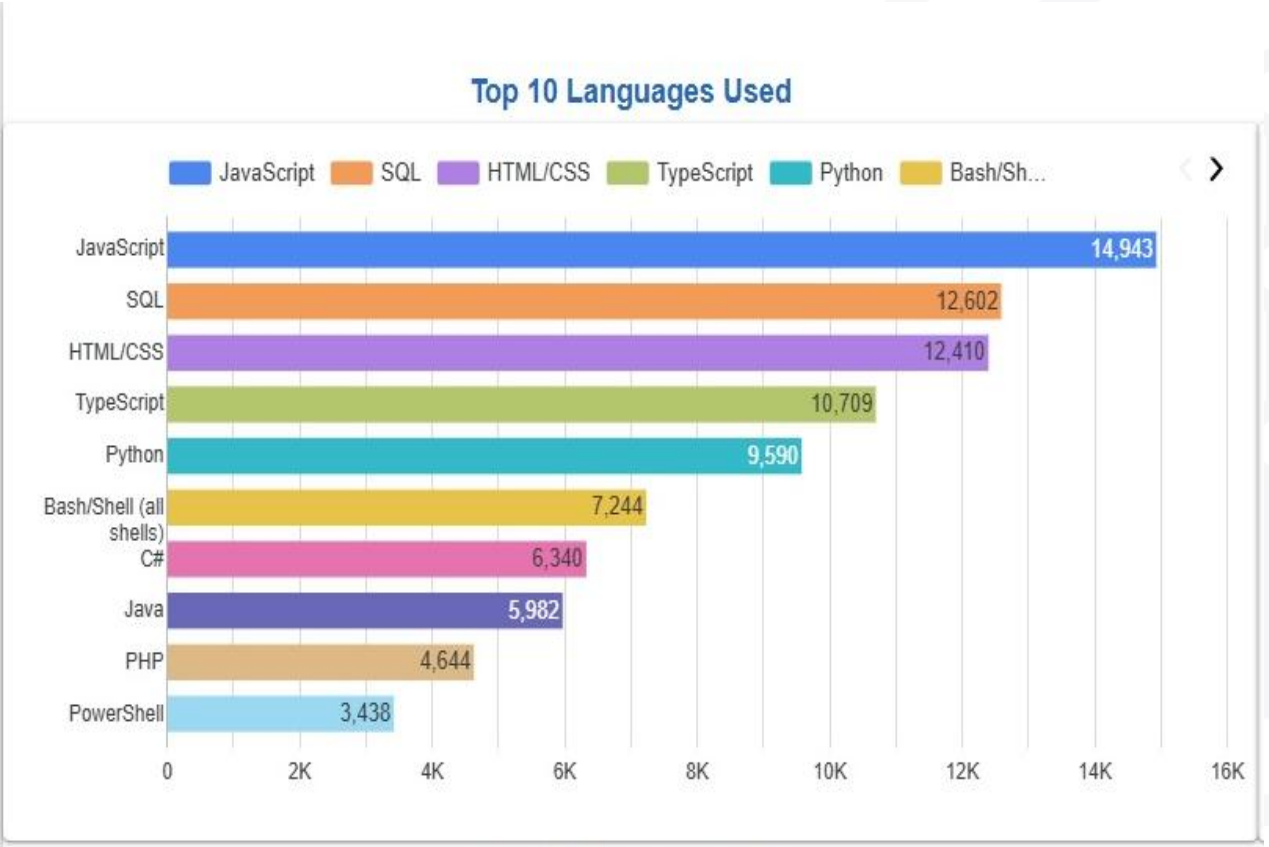
RESULTS

🎯 For Stakeholders / Clients

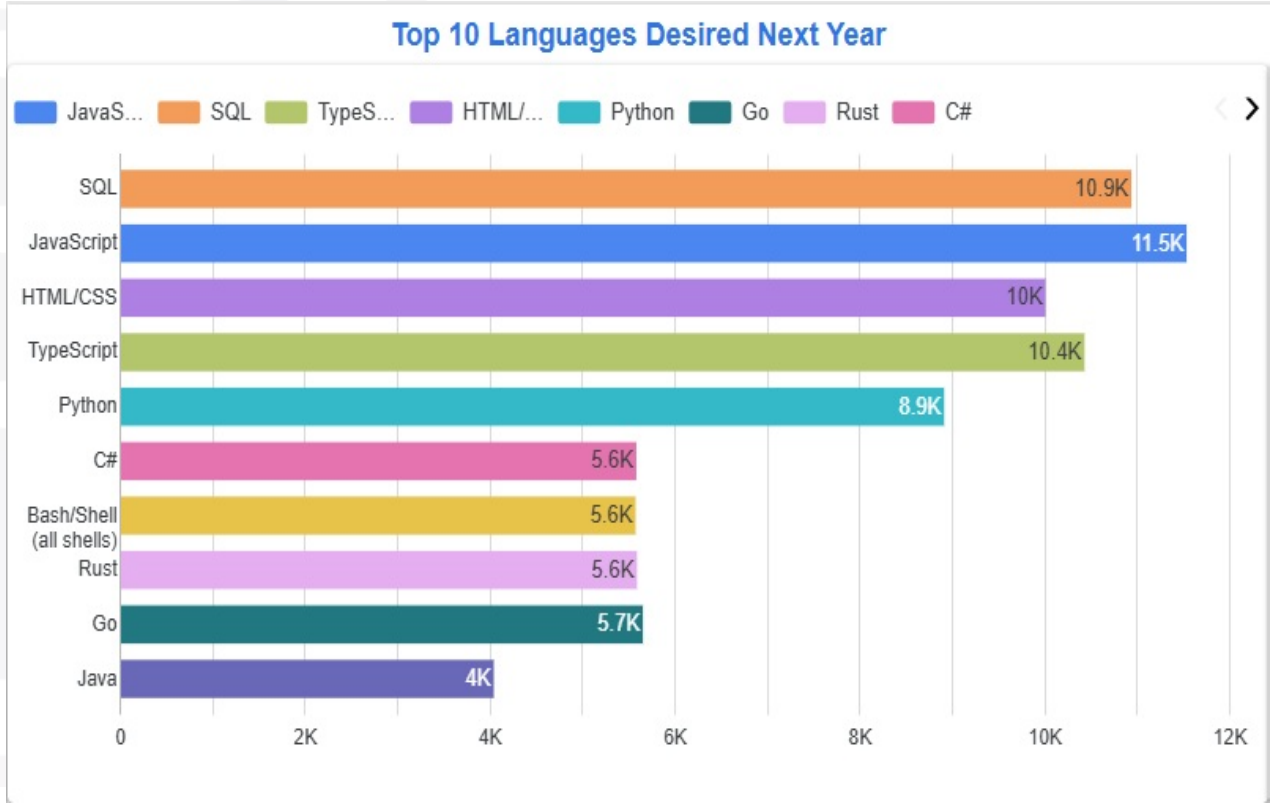
- **Training & Upskilling**: Developers want to move toward **Rust, TypeScript, Next.js** → so, invest in training content and support would be preferable.
- **Product Strategy**: Ensure support for **PostgreSQL, MongoDB, Redis** as they align with future demand.
- **Cloud Strategy**: Continue prioritizing partnerships with **AWS, Azure, Google Cloud**, but monitor growing players like **Cloudflare**.
- **Recruitment Insight**: Expect candidates with strong JavaScript/React backgrounds but prepare for growing demand in modern stacks (TypeScript, Rust, Next.js).

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year



PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings (Current Usage – “Top 10 Languages Used”)

- **JavaScript** is the clear leader (≈15K mentions), reflecting its dominance in web development and frontend ecosystems.
- **SQL and HTML/CSS** remain foundational, underpinning databases and web interfaces.
- **Python** is strong, reflecting its dual role in data science and backend development.
- **C# and Java** are still widely used, highlighting their entrenched role in enterprise environments.
- **Bash/Shell** indicates importance of scripting and automation skills.
- **PHP and PowerShell** are present but declining in share compared to others.

Findings (Future Aspirations – “Top 10 Languages Desired Next Year”)

- **JavaScript and SQL** remain highly desired → no signs of slowing down.
- **TypeScript** has surged into the top tier, reflecting demand for stronger typing and maintainability in large JavaScript codebases.
- **Python** maintains high desirability, ensuring it stays relevant in data and AI-driven work.
- **Rust and Go** emerge strongly:
  - Rust = admired for performance and memory safety, increasingly seen in systems programming and security-sensitive applications.
  - Go = praised for simplicity and concurrency handling, making it attractive for cloud-native and distributed systems.
- **C# and Java** still appear but trail newer, modern languages in desirability.

PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

TRENDS

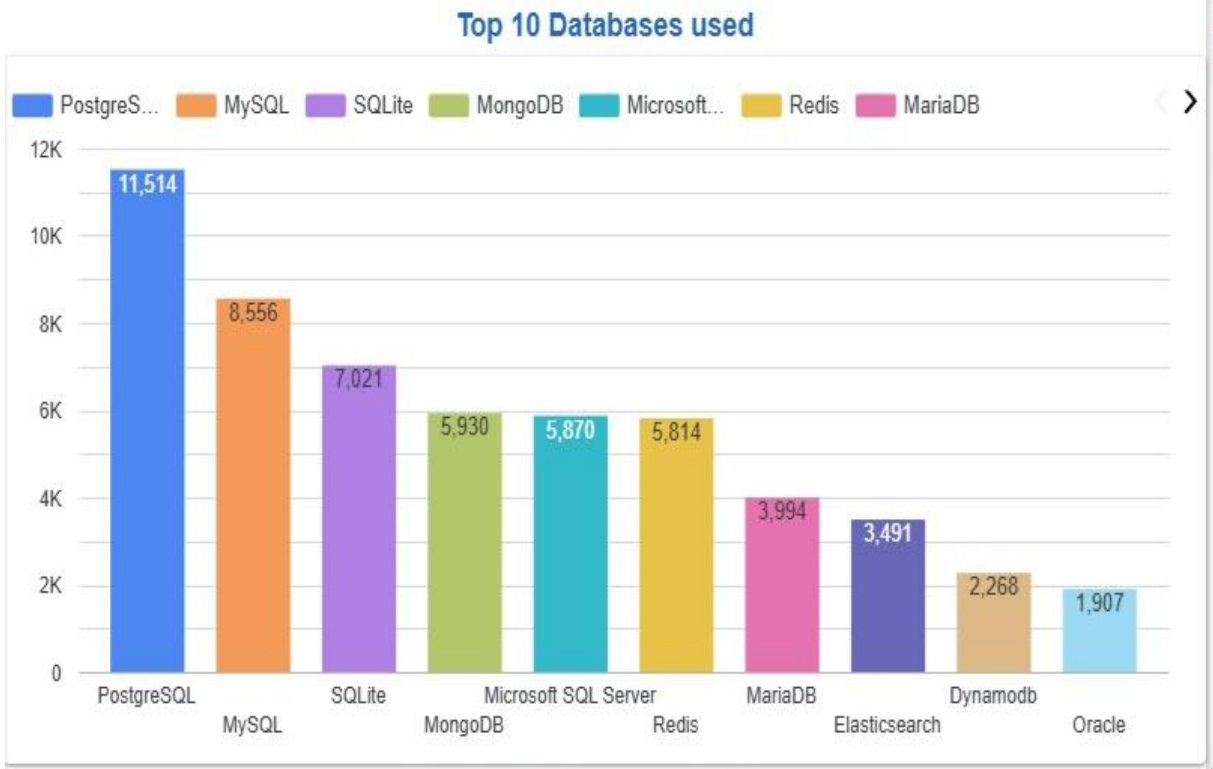
- Shift from **traditional enterprise languages (C#, Java, PHP)** toward **modern, safe, and scalable languages (TypeScript, Rust, Go)**.
- **JavaScript** remains the lingua franca of development, but its ecosystem is increasingly **TypeScript-first**.
- **Python** continues to bridge multiple domains (data, AI, backend).
- **Rust's rise** signals a growing emphasis on performance and safety in mainstream development.

IMPLICATIONS for Stakeholders

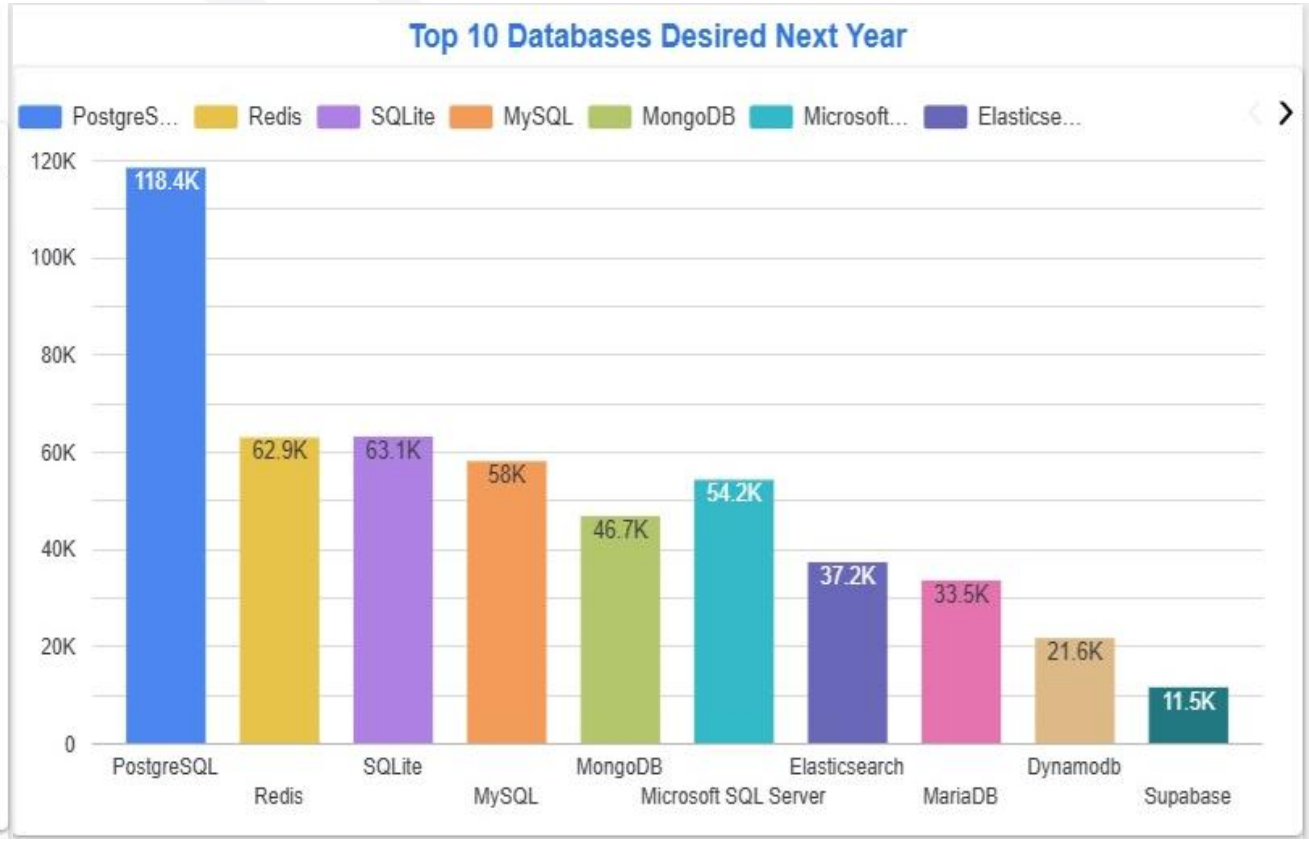
- **Training & Education**
  - Upskilling developers in **TypeScript, Rust, and Go** is critical — these languages are becoming industry priorities.
  - Continued investment in **Python** training ensures readiness for data-driven roles.
- **Technology Strategy**
  - Companies can reduce long-term technical debt by adopting **TypeScript** for large-scale JS projects.
  - Monitoring Rust adoption helps in future-proofing security-critical or performance-intensive applications.
- **Recruitment & Talent**
  - Expect candidates to arrive with **JavaScript/Python/SQL** skills, but employers should **attract talent by offering opportunities** to work with Rust and Go.
  - Future hiring will likely emphasize **modern language proficiency** alongside legacy maintenance skills.

DATABASE TRENDS

Current Year



Next Year



DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings (Current Usage – “Top 10 Databases Used”)

- **PostgreSQL** is the top-used database, reflecting its reputation as the most advanced open-source RDBMS.
- **MySQL** and **SQLite** remain highly used, especially for web applications and lightweight setups.
- **MongoDB** and **Microsoft SQL Server** show strong representation across enterprises and modern web apps.
- **Redis** is widely adopted as a caching and in-memory store.
- **MariaDB**, a MySQL fork, maintains a solid niche.
- **Elasticsearch** is popular for search and log analytics use cases.
- **DynamoDB** and **Oracle** trail behind but still appear in the top 10, reflecting specialized enterprise/cloud adoption.

Findings (Future Aspirations – “Top 10 Databases Desired Next Year”)

- **PostgreSQL** strengthens its dominance → developers not only use it but also want to use it more.
- **MongoDB** moves higher in desirability, showing interest in flexible NoSQL and document-oriented approaches.
- **Redis** and **SQLite** remain strong, signifying ongoing importance of in-memory performance and lightweight DBs.
- **MySQL** still has demand but less aspiration compared to PostgreSQL.
- **Elasticsearch** maintains relevance, particularly for data search/analytics needs.
- **Supabase** enters the desired list → a new, developer-friendly PostgreSQL-based platform for modern apps.
- **DynamoDB** and **MariaDB** remain on the list but with weaker growth signals compared to emerging options.

DATABASE TRENDS - FINDINGS & IMPLICATIONS

TRENDS

- **PostgreSQL is the clear "winner"** — both most used and most desired, suggesting long-term momentum.
- Developers are increasingly attracted to **NoSQL and flexible options (MongoDB, Redis, Elasticsearch)** for modern, scalable apps.
- **Cloud-native/PostgreSQL-based platforms like Supabase** are gaining interest, highlighting a shift toward developer experience and managed services.
- Traditional enterprise databases (**Oracle, SQL Server**) are less aspirational, indicating possible long-term decline outside legacy contexts.

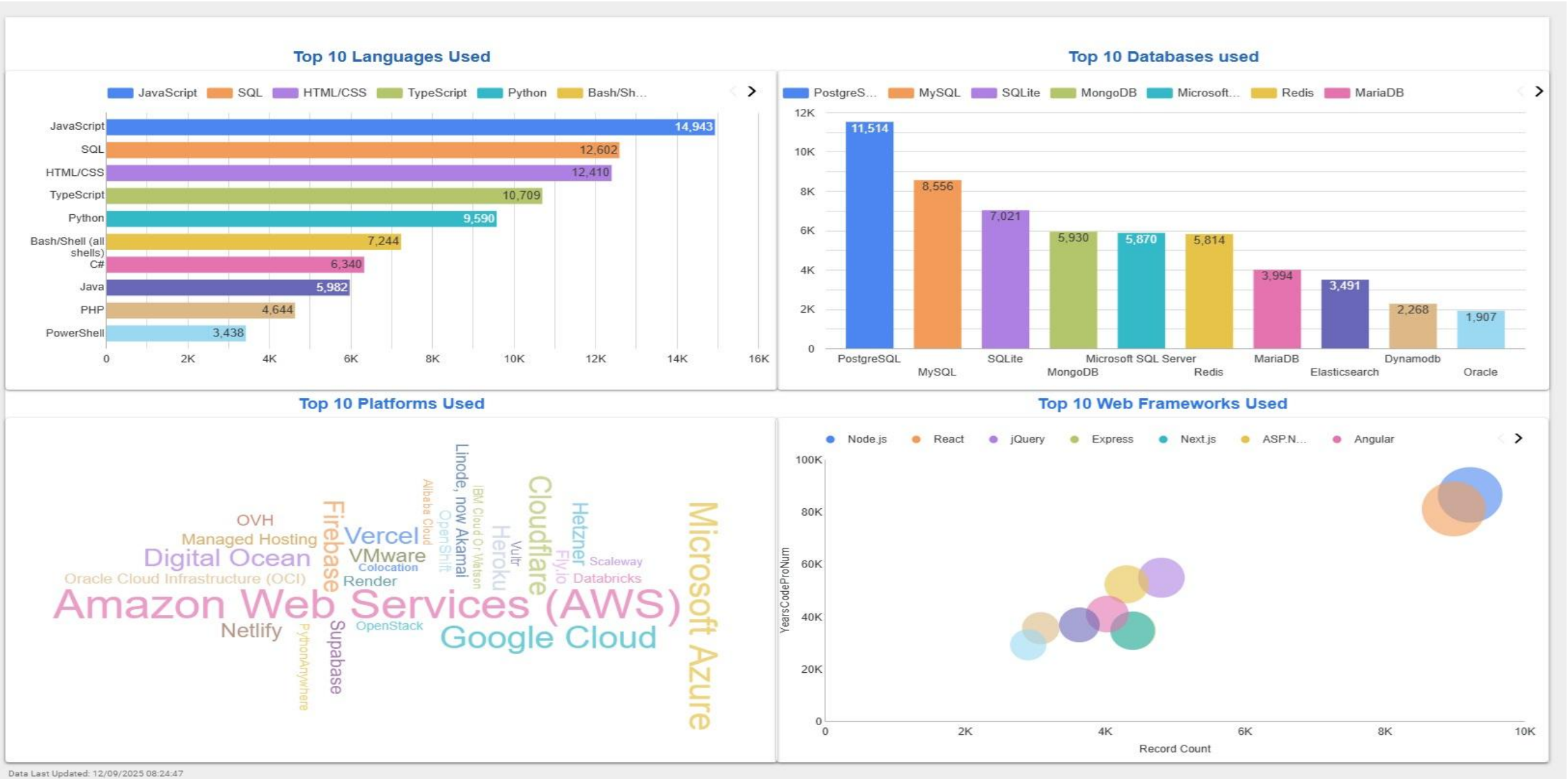
IMPLICATIONS for Stakeholders

- **Technology Strategy**
  - Invest in **PostgreSQL expertise** → it's both today's leader and tomorrow's growth area.
  - Strengthen **support for MongoDB, Redis, and Elasticsearch** as these align with modern cloud, search, and real-time workloads.
  - Consider **Supabase** as an emerging platform that may appeal to startups and modern SaaS builders.
- **Training & Upskilling**
  - Developers will need **hybrid skills**: relational DBs (PostgreSQL, MySQL) + NoSQL/real-time DBs (MongoDB, Redis, Elasticsearch).
  - Enterprise teams should prepare for gradual decline in **Oracle/SQL Server talent** and pivot training accordingly.
- **Recruitment & Talent**
  - Candidates will mostly come with **PostgreSQL, MySQL, and MongoDB experience**.
  - Offering opportunities to work on **next-gen stacks (Supabase, Redis, Elastic)** can attract ambitious developers.

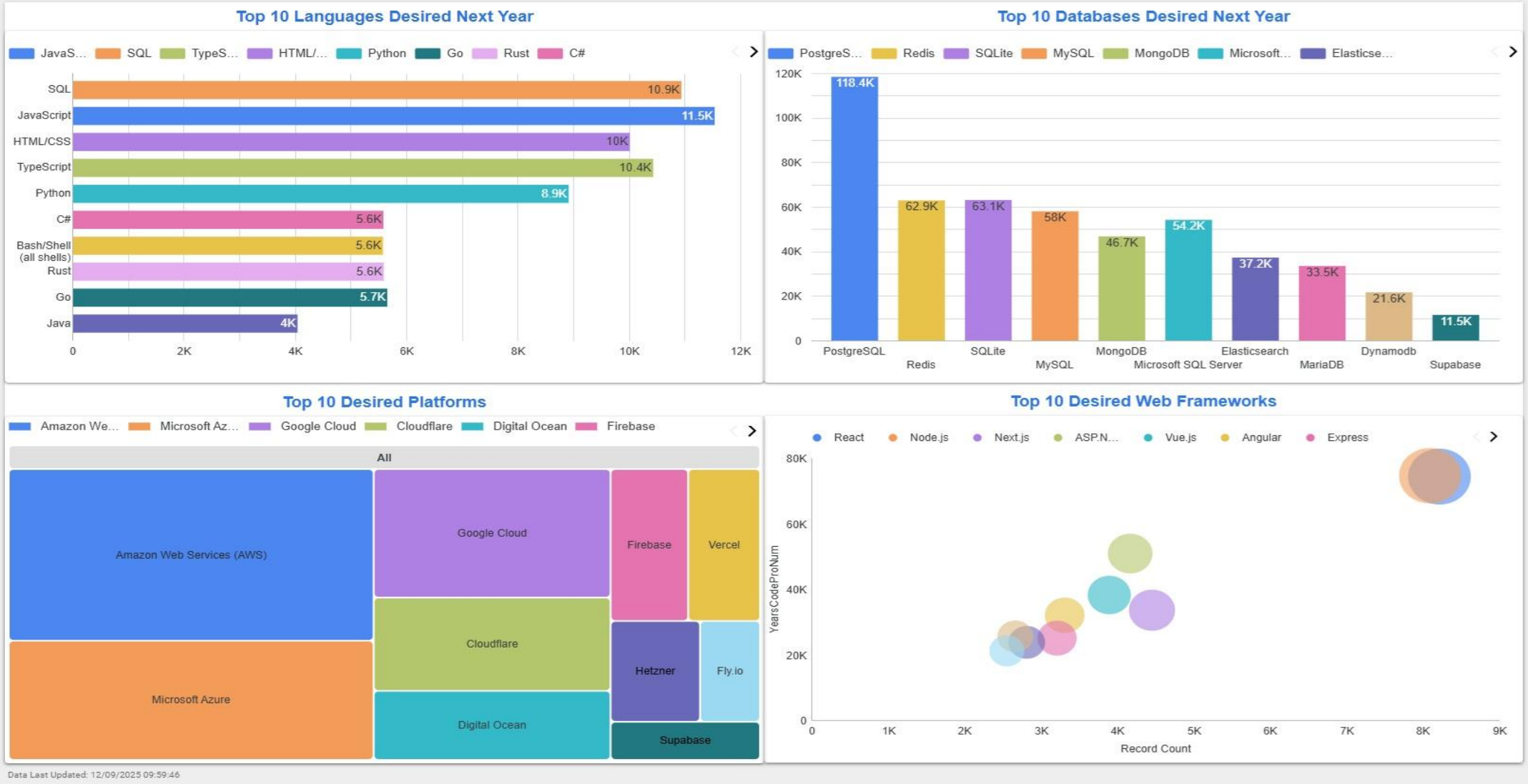
DASHBOARD

https://github.com/rk-ukm/Capstone-Project-Repo

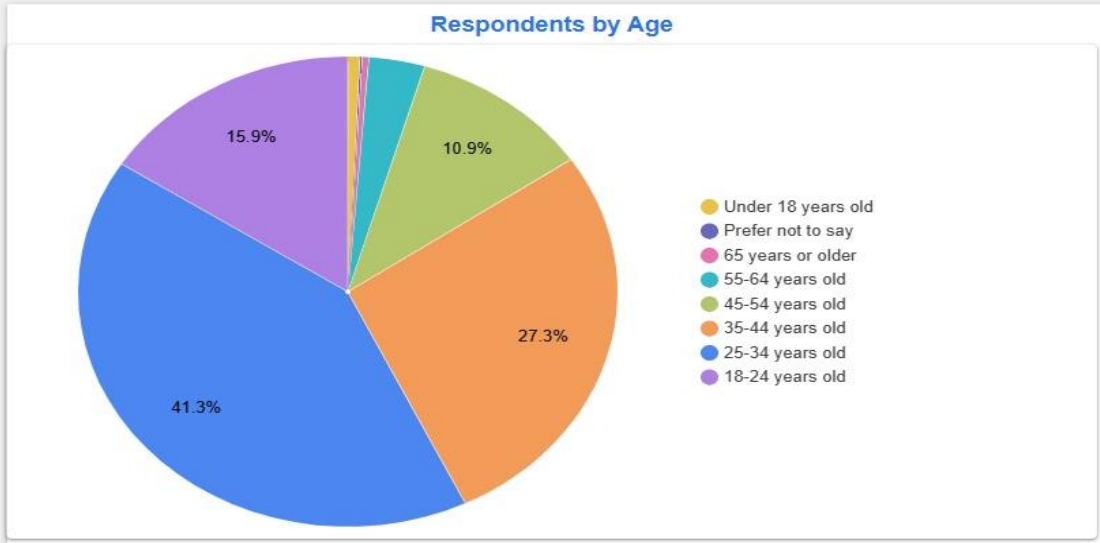
DASHBOARD TAB 1

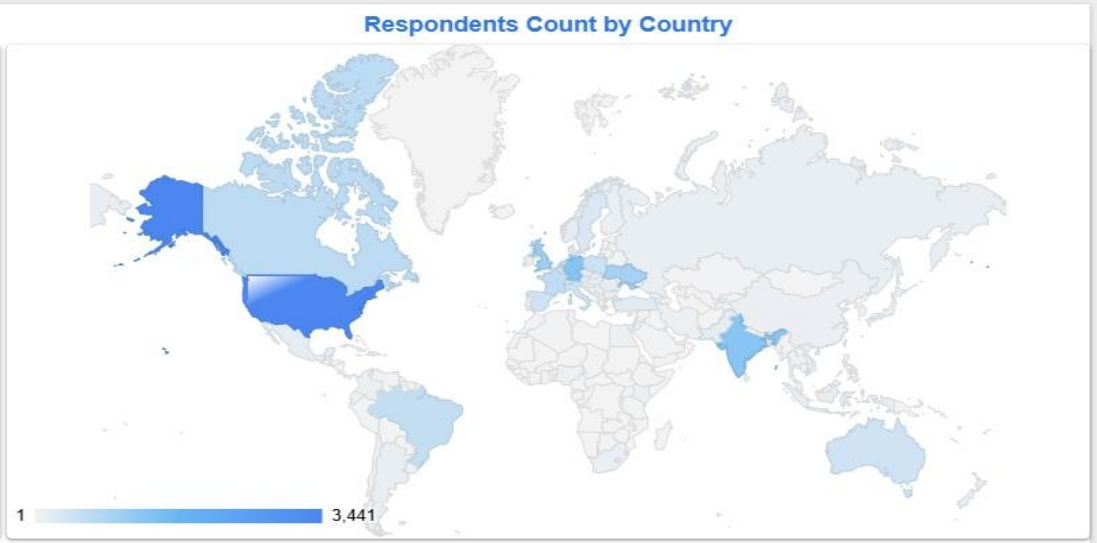


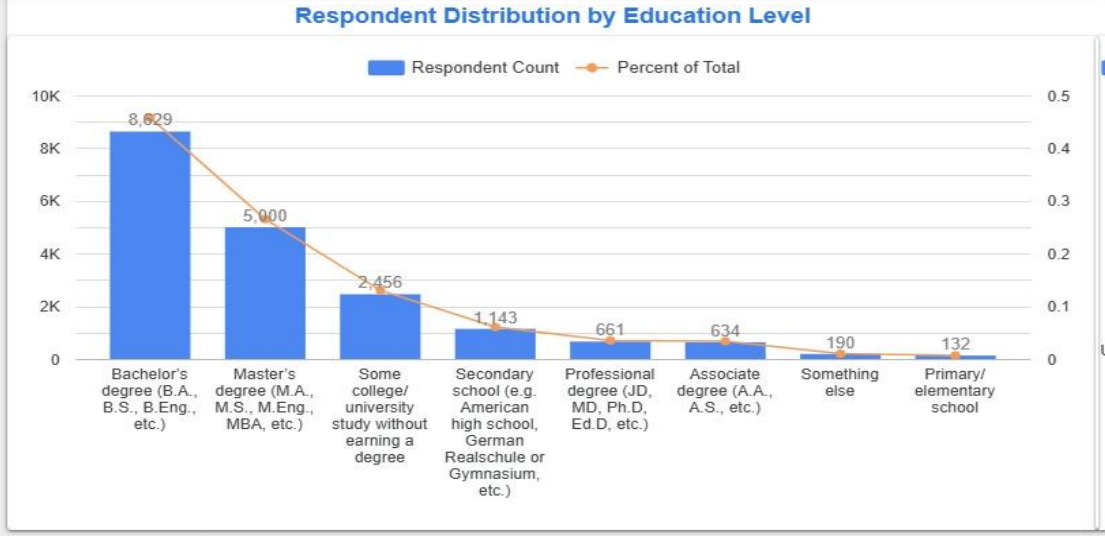
DASHBOARD TAB 2

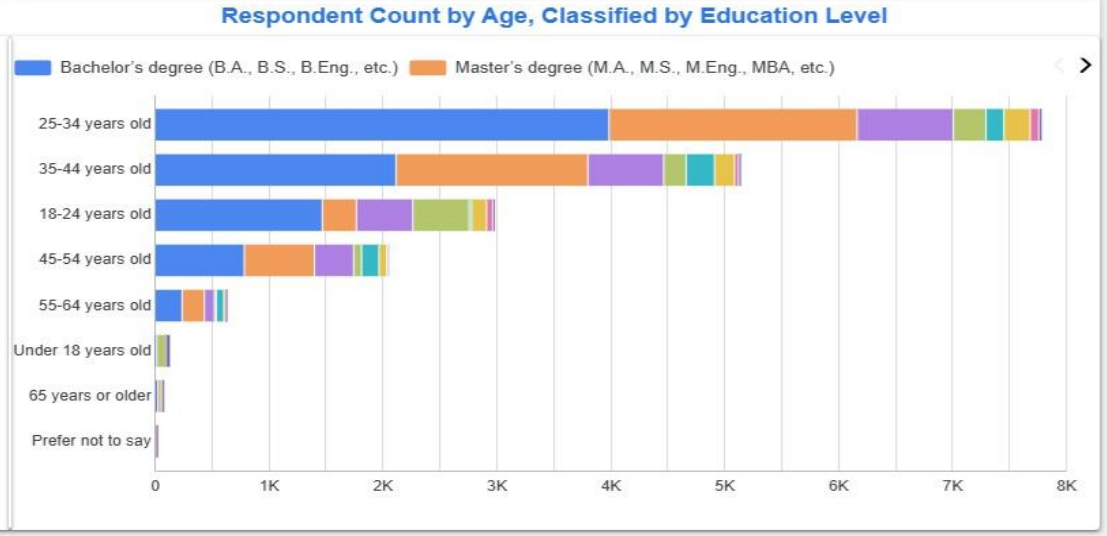


DASHBOARD TAB 3









Data Last Updated: 13/09/2025 04:08:41

DISCUSSION

📌 Summary in a Nutshell

1. Current Technology Usage

- Developers rely on **JavaScript, SQL, and Python** as the core languages, alongside **PostgreSQL/MySQL** for databases.
- **AWS, Azure, Google Cloud** dominate platforms, while **React/Node.js** lead web frameworks.
- Traditional technologies (**Java, PHP, Oracle**) remain in use but show less momentum.

DISCUSSION

📌 **Summary in a Nutshell**

2. Future Technology Trends

- Developers aspire to adopt **TypeScript, Rust, and Go**, highlighting a shift toward **safety, scalability, and modern design**.
- **PostgreSQL** continues its rise, while **MongoDB, Redis, Supabase** show strong desirability.
- Platforms: interest is diversifying — while **AWS/Azure/Google Cloud** stay relevant, newer entrants like **Cloudflare, Vercel, Supabase** are catching developer attention.
- Web frameworks: desire shifts to **Next.js, Vue.js, and Svelte**, showing demand for modern frontend ecosystems.

DISCUSSION

📌 Summary in a Nutshell

3. Demographics

- Majority are **25–34 years old**, early-to-mid career professionals.
- Most hold a **Bachelor's degree**, with Master's next in line.
- Respondents are globally distributed, but concentrated in **North America and Europe**.
- Coding experience varies, but many have **less than 10 years of professional experience**, showing a relatively young developer base.

DISCUSSION

🎯 **Overall Takeaway**

- **Current usage** shows stability around JavaScript, SQL, Python, PostgreSQL, and AWS.
- **Future aspirations** show clear momentum toward TypeScript, Rust, Go, PostgreSQL, and modern frameworks like Next.js.
- **Demographics** reveal a young, ambitious developer community driving this transition.

💡 **Implication:** Organizations that **support developers' desire for modern tools** (TypeScript, Rust, PostgreSQL, Next.js) while maintaining today's core stacks will be best positioned for talent attraction, retention, and innovation.

OVERALL FINDINGS & IMPLICATIONS

Findings

- **Current Usage**: Developers rely heavily on established technologies — **JavaScript, SQL, and Python** as core languages, **PostgreSQL/MySQL** for databases, **AWS/Azure/Google Cloud** for platforms, and **React/Node.js** for frameworks.
- **Future Trends**: There's strong aspiration toward **modern, safer, and more scalable tools** — **TypeScript, Rust, Go** (languages), **MongoDB, Redis, Supabase** (databases), **Cloudflare/Vercel** (platforms), and **Next.js/Vue.js/Svelte** (frameworks).
- **Demographics**: Respondents are largely **25–34 years old, bachelor's-educated, globally distributed**, and represent a young, ambitious developer community that's shaping adoption patterns.

Implications

Strategic Technology Investment

- Organizations should maintain strong support for today's mainstream stacks (JS, SQL, Python, AWS) while **actively piloting emerging tools** (TypeScript, Rust, Go, Supabase, Next.js).
- This dual approach ensures stability in delivery while future-proofing tech strategy.

Talent Attraction & Retention

- Developers want opportunities to work with **modern, aspirational technologies**.
- Offering projects in Rust, Go, TypeScript, or Next.js can **differentiate employers** in a competitive talent market.

Training & Upskilling

- Companies should invest in **transition training**: helping teams move from legacy stacks (Java, PHP, Oracle) to modern ecosystems.
- Upskilling in **data and cloud-native tech** (PostgreSQL, MongoDB, Redis, cloud platforms) will be key.

Market Positioning

- Vendors and stakeholders who align product offerings and developer experiences with these trends (e.g., simplifying PostgreSQL adoption, supporting modern frameworks) will be best positioned to capture growth.

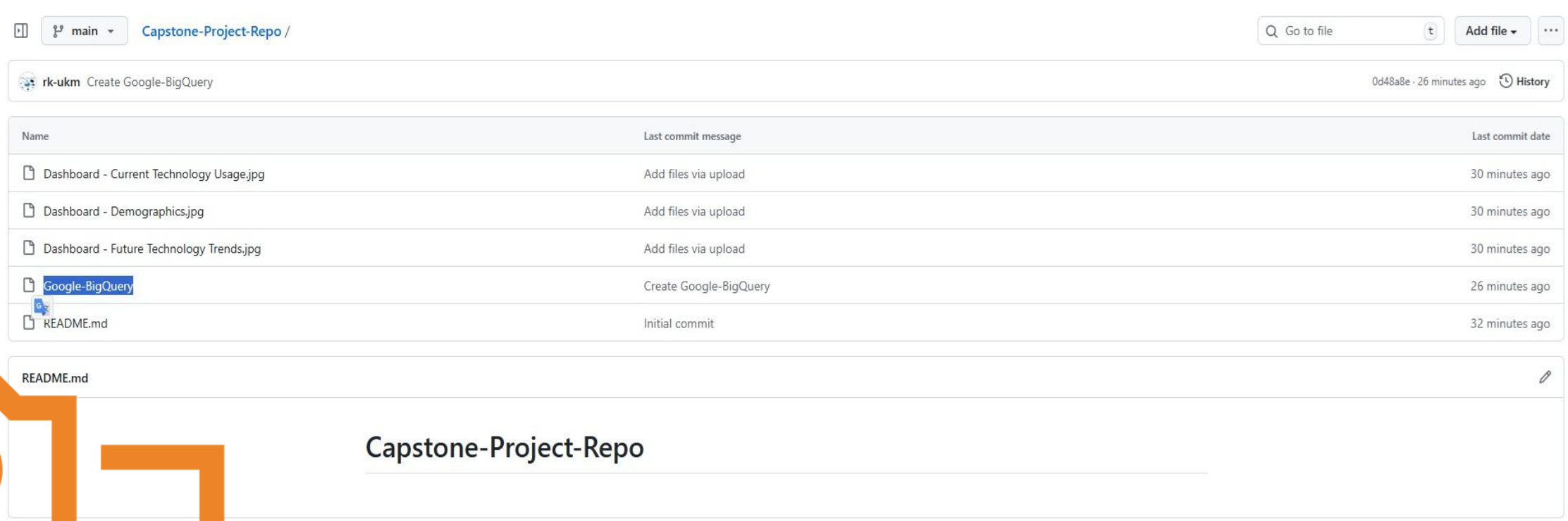
CONCLUSION

- Today’s developer landscape is built on **stable, proven technologies** like JavaScript, SQL, Python, PostgreSQL, and AWS — but tomorrow’s momentum is clearly shifting toward **modern, safer, and more developer-friendly tools** such as TypeScript, Rust, Go, PostgreSQL (again, even stronger), and frameworks like Next.js.

- The community driving this shift is **young, globally distributed, and eager to adopt new technologies**.

👉 **In short:** organizations that balance **current stability** with **future aspirations** will be best positioned to innovate, attract top talent, and stay competitive in a rapidly evolving tech ecosystem.

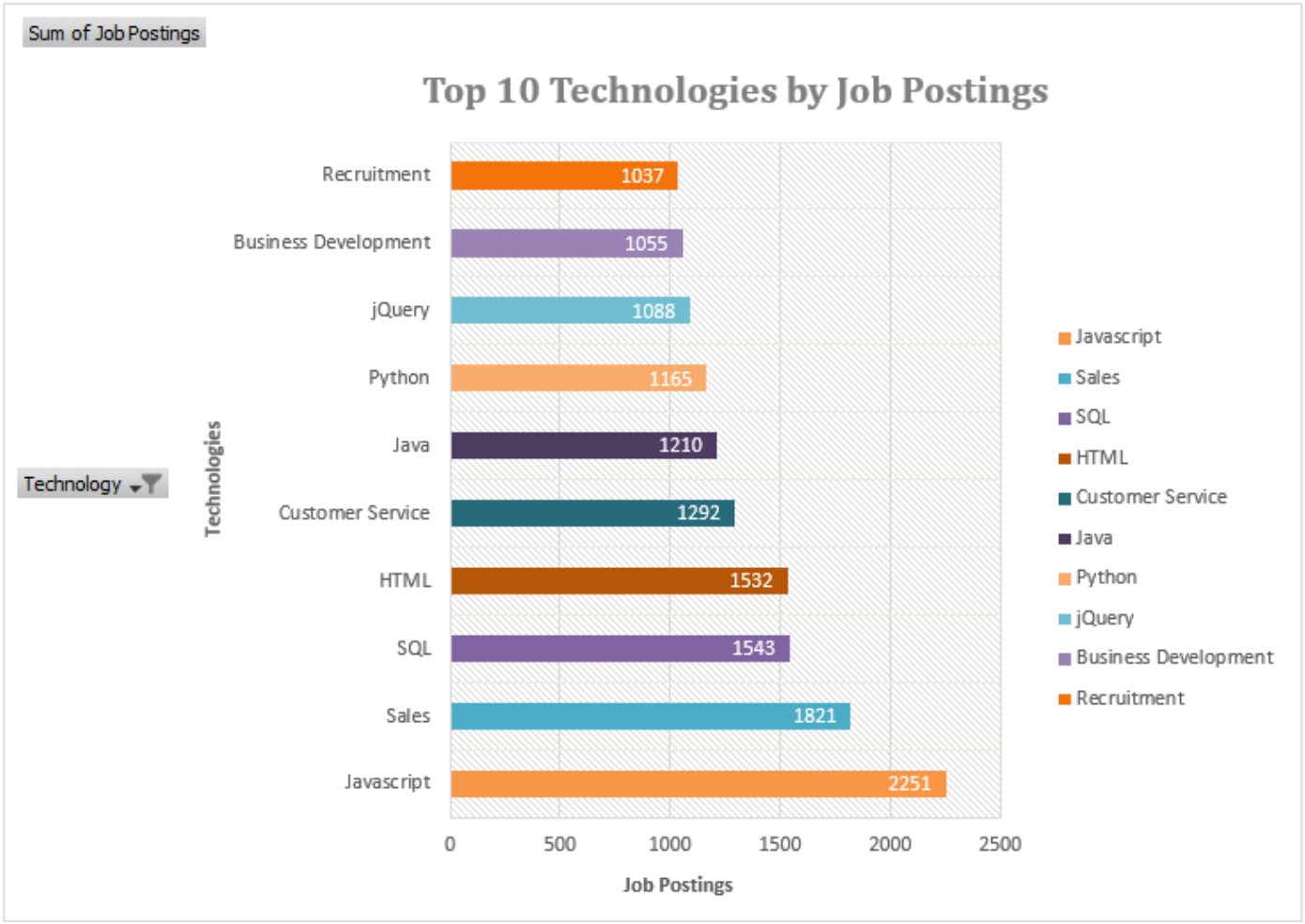
APPENDIX



- Please find my **BigQuery** Commands within my GitHub Repo to better understand the tables, views, relations, filters etc. used in survey data analysis.

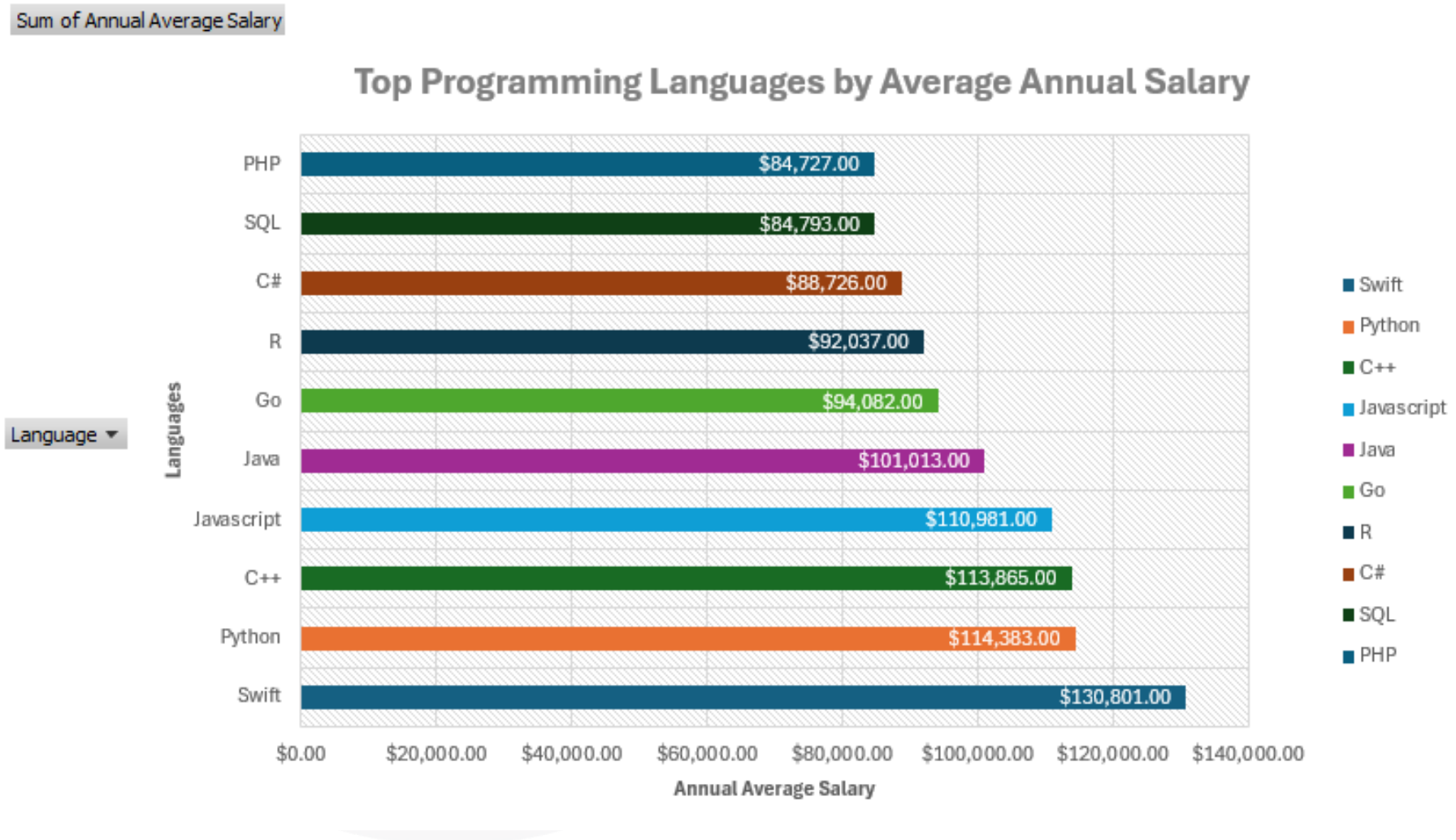
JOB POSTINGS

Row Labels	Sum of Job Postings
Javascript	2251
Sales	1821
SQL	1543
HTML	1532
Customer Service	1292
Java	1210
Python	1165
jQuery	1088
Business Development	1055
Recruitment	1037
Grand Total	**13994**



POPULAR LANGUAGES

Row Labels	Sum of Annual Average Salary
Swift	$130,801.00
Python	$114,383.00
C++	$113,865.00
Javascript	$110,981.00
Java	$101,013.00
Go	$94,082.00
R	$92,037.00
C#	$88,726.00
SQL	$84,793.00
PHP	$84,727.00
Grand Total	**$1,015,408.00**



SUMMARY

Chart 1: Top Programming Languages by Average Annual Salary

- **Swift** commands the highest average salary (~$130K), followed by **Python**, **C++**, and **JavaScript** (~$111K).
- The more “enterprise” languages (**C#, SQL, PHP**) sit lower in the salary range (~$85K).

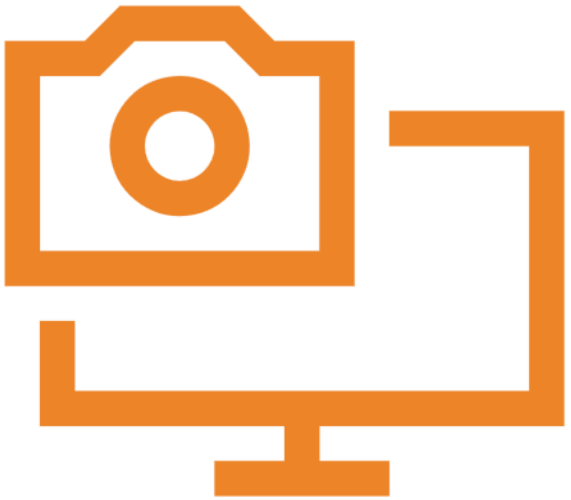
➞ **Implication:** Niche or specialized languages (Swift, C++) tend to attract higher pay, while ubiquitous technologies (SQL, PHP) are essential but less differentiated in salary.

Chart 2: Top 10 Technologies by Job Postings

- **JavaScript** leads strongly in job demand (~2250 postings), followed by **Sales, SQL, and HTML**.
- There’s also high demand for general business/soft-skill roles like **Customer Service, Recruitment, and Business Development** alongside technical skills.

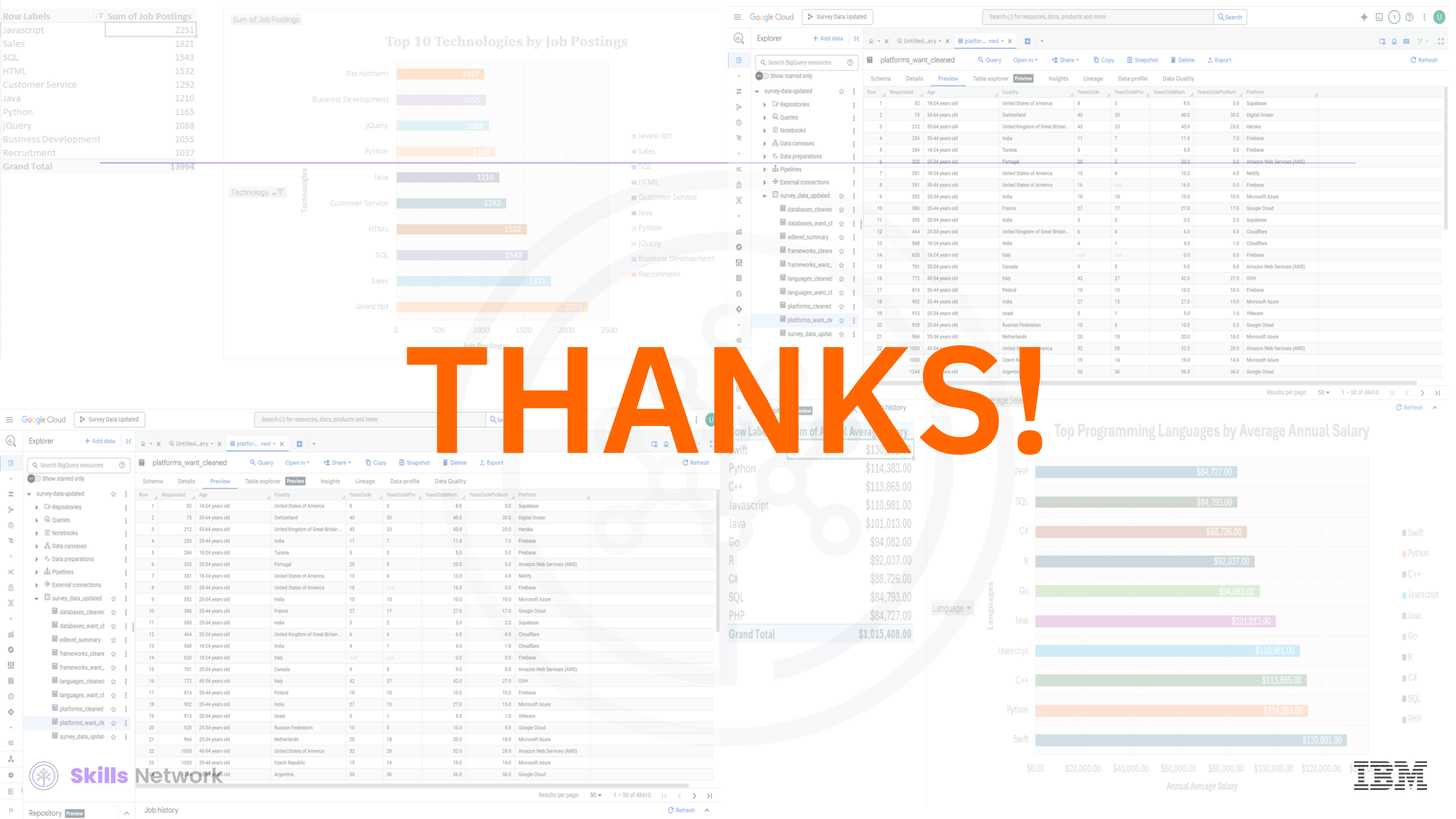
➞ **Implication:** While JavaScript dominates tech hiring, companies are also balancing demand with business-facing roles. Tech stacks remain centred around web technologies (JavaScript, SQL, HTML).

END LINE



Takeaway for Stakeholders

- **High Salaries ≠ High Demand:** Swift and C++ offer premium salaries but aren't as in-demand as JavaScript or SQL.
- **High Demand, Stable Salaries:** JavaScript and SQL dominate job postings but offer mid-range pay — essential skills that every developer is expected to know.
- **Strategic Action:**
  - Upskill teams in **JavaScript + SQL** to meet current demand.
  - Invest in niche expertise (Swift, C++) for high-value projects.
  - Balance hiring between **technical specialists** and **business enablers** (sales, customer service, recruitment).


Row Labels
Sum of Job Postings
Javascript 2251
Sales 1821
SQL 1543
HTML 1532
Customer Service 1292
Java 1210
Python 1165
jQuery 1088
Business Development 1055
Recruitment 1037
Grand Total 13994
Top 10 Technologies by Job Postings
Technology
Job Postings
Recruitment
Business Development
jQuery
Python
Java
Customer Service
HTML
SQL
Sales
Javascript
THANKS!
Top Programming Languages by Average Annual Salary
Annual Average Salary
Skills Network
IBM