בסקייל עם Apache Iceberg Amazon S3 Tables

AWS Summit Tel Aviv 2026 ,סיכום סשן — TLV-DEM302

Yossi Chai (Senior Solutions Architect,-ו Dan Kolodny (Senior Solutions Architect, AWS Startups EMEA)
AWS Telco IBU EMEA)

10 בספטמבר 2026 | משך: כ-21 דקות | הופק מהקלטת הסשן

מסלול: AWS Demos | רמה: 300 — Advanced

על המסמך הזה

המסמך מסכם את הסשן TLV-DEM302 מתוך AWS Summit Tel Aviv 2026, בהתבסס על ההקלטה המלאה שפורסמה באתר הסאמיט. הוא נבנה מתמלול אוטומטי של ההקלטה ומהסליידים שחולצו מהווידאו עצמו, כך שאפשר לחזור לתוכן בלי לצפות שוב בהקלטה. חותמות הזמן בגוף המסמך מפנות לנקודה המדויקת בהקלטה.

איך לקרוא את זה

- **פרק 1 — תקציר מנהלים:** שלוש דקות קריאה. מה הוצג ומה המסקנה המעשית.
- **פרקים 2–5 — התיאוריה:** התרחיש העסקי, ארכיטקטורת הפייפליין, מבנה Apache Iceberg ושלושת אתגרי התחזוקה.
- **פרקים 6–9 — הדמו:** יצירת טבלה מהקונסול, הגדרות תחזוקה, השוואת S3 Tables מול S3 רגיל, ותחקור בשפה חופשית.
- **פרק 10 — מה לוקחים מכאן:** מסקנות ונקודות להמשך, ומילון מונחים.

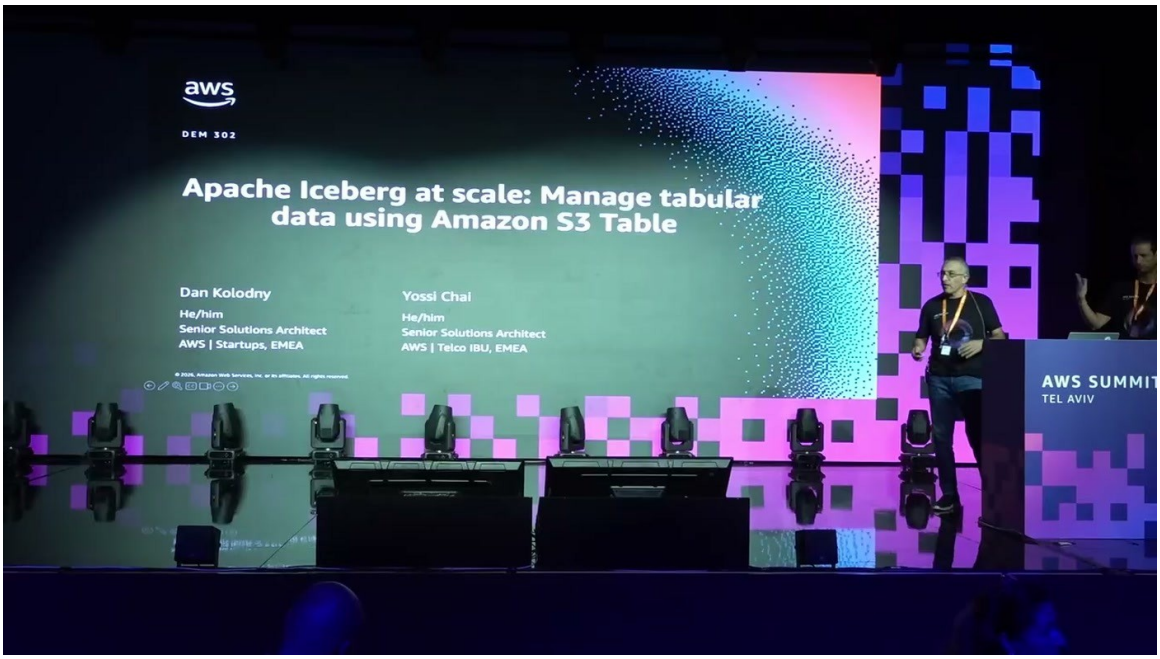
הערת מקור המסמך הופק מתמלול אוטומטי של ההקלטה. התמלול מדויק בקטעים המקצועיים, אך מונחים ושמות לועזיים מופיעים בו בשיבוש פונטי (למשל "קומפקשן" = compaction, "סנפשוט" = snapshot) — בגוף המסמך הם נורמלו למונח האנגלי, ובציטוטים הושארו כפי שנאמרו. הציטוטים אומתו מול התמלול בחותמת הזמן המצוינת. מספרים שמופיעים על הסליידים צוינו ככאלה.

1. תקציר מנהלים

שני Solutions Architects מ-AWS לקחו תרחיש אחד — פלטפורמת e-commerce שמייצרת עשרת אלפים אירועים בשנייה — והראו מה ההבדל בין להריץ Apache Iceberg בעצמך לבין להריץ אותו על Amazon S3 Tables. הסשן לא ניסה למכור את Iceberg; הוא הניח שכבר בחרתם בו, והתמקד במחיר התחזוקה שמגיע איתו ובשאלה מי משלם אותו.

- **הכאב הוא לא הפייפליין, הוא התחזוקה.** הדובר פתח בכך שבניית הצינור עצמו פשוטה יחסית, אבל "אתם מבזבזים שבועות על תשתית, על סנפשוטים, על סכם המנג'מנט ועל קומפקשן, במקום להתעסק באינסייטס" [0:00].
- **שלושה תהליכי תחזוקה, אפס ערך עסקי.** compaction, snapshot cleanup ו-orphan file cleanup — שלושתם cron jobs שצריך לכתוב, לתזמן, לנטר ולתקן. הסיכום שלו: "יש לנו שלושה אתגרים, שלושה תהליכים, אפס ערך עסקי, רק תחזוקה" [10:47].
- **המספרים שהוצגו לטובת S3 Tables.** "עד פי 3 בביצועי קריאה ועד פי 10 בטרנזקציות לשנייה בכתיבה למול S3 רגיל" [1:30]. המספרים הוצגו גם על הסלייד.
- **ההשוואה החיה היא הראיה החזקה בסשן.** אותו דאטה בדיוק נכתב פעמיים — פעם ל-table bucket ופעם ל-bucket רגיל. התוצאה על המסך: 20 קבצים בגודל ממוצע 7.24MB מול 8,849 קבצים בגודל ממוצע 24.8KB, בלי שאיש נגע בכלום.
- **השליטה נשארת בידיים שלכם.** כל שלוש פעולות התחזוקה ניתנות להפעלה, כיבוי וכיוונון מתוך הקונסול — target file size, אסטרטגיית compaction, ומספר ה-snapshots לשימור.
- **MCP מחבר את הטבלאות לעולם ה-AI.** החלק האחרון בדמו הראה שאילתות בשפה חופשית מול הטבלאות, כולל שאלה תפעולית על מתי רצה התחזוקה האחרונה לכל טבלה.

2. התרחיש: צוות אנליטיקה שלא רוצה צוות אופס

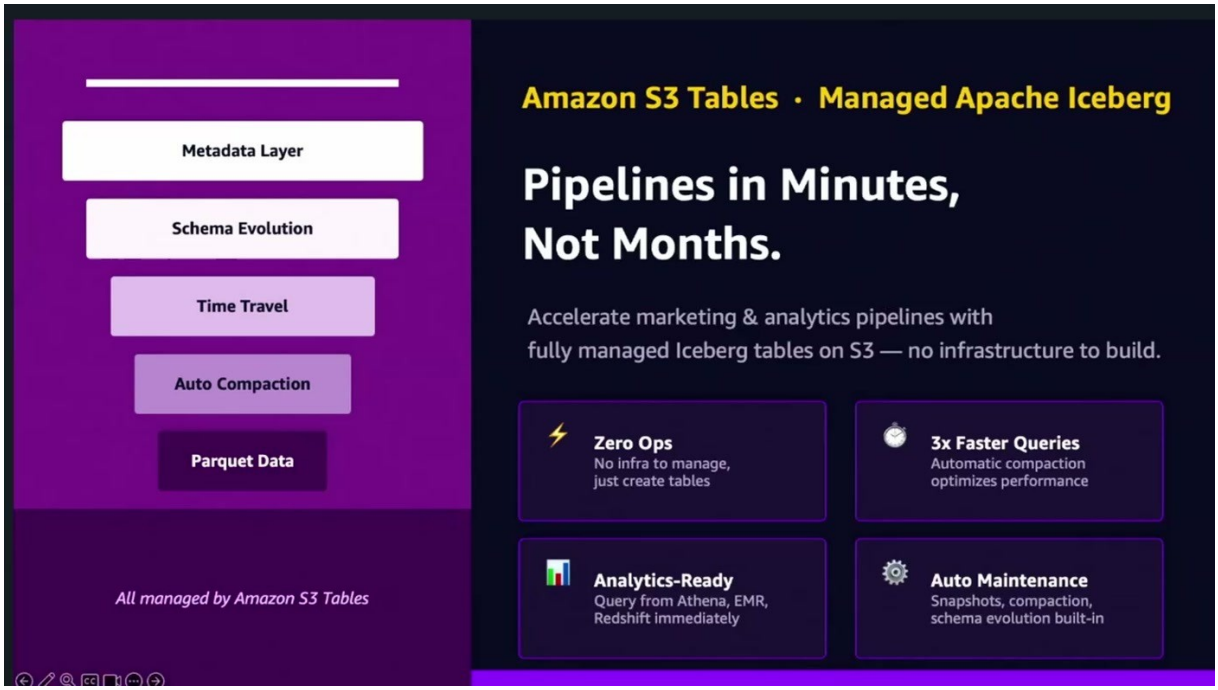


סלייד הפתיחה עם שמות שני הדוברים ותפקידיהם, ומספר הסשן DEM 302

הסשן נפתח בסיפור לקוח: חברת e-commerce שביקשה מערכת שתאפשר לצוות המרקטינג והאנליטיקה שלה לאסוף אירועים בקצב גבוה, לשמור אותם לאורך זמן ולתחקר אותם — "וכל זאת במינימום מינימום כוח אדם להקמה ותחזוקה" [0:00].

הפלטפורמה מייצרת "סדר גורל של 10,000 אירועים בשנייה" [1:30] — אירועי JSON קטנים, כל אחד מייצג מכירה עם מזהה, שם משתמש, עלות וכמות. הם קטנים, הם מגיעים במסות, והם לא מפסיקים. המטרה: להנחית אותם לטבלה שמוכנה מראש לאנליטיקה, בלי לבנות סביבה צוות תחזוקה.

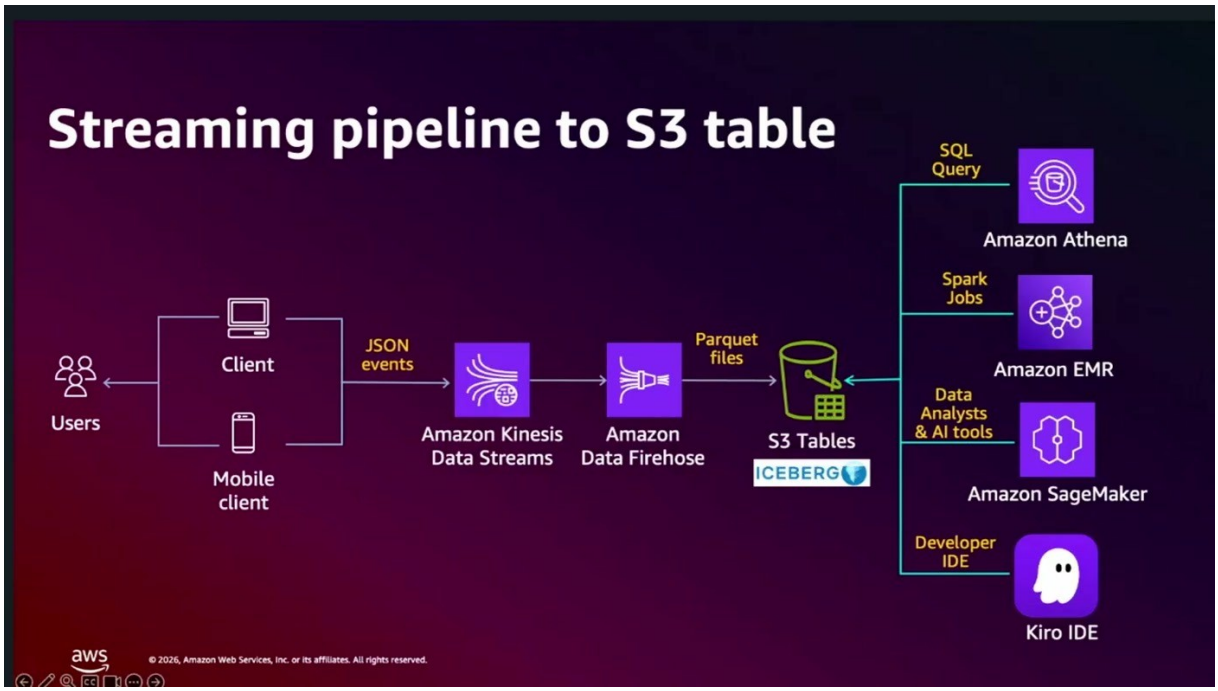
> **— [0:00]** די פשוט, אבל בפועל, אתם מבזבזים שבועות על תשתית, על סנפשוטים, על סכם המנג'מנט ועל קומפקשן, במקום להתעסק באינסייטס



הבטחת הערך של S3 Tables בארבעה רבעים: Zero Ops, ביצועי שאילתה, מוכנות לאנליטיקה ותחזוקה אוטומטית

הסלייד מתרגם את זה לארבע הבטחות: Zero Ops ("אין תשתית לנהל, רק צרו טבלאות"), שאילתות מהירות פי 3 בזכות compaction אוטומטי, מוכנות מיידית ל-Athena / EMR / Redshift, ותחזוקה מובנית של snapshots, compaction ו-schema evolution. בעמודה השמאלית: Metadata Layer, Schema Evolution, Time Travel, Auto Compaction ו-Parquet Data — כולם, כלשון הסלייד, "All managed by Amazon S3 Tables".

3. הארכיטקטורה: מ-Kinesis ועד הטבלה



זרימת האירועים: לקוחות → Kinesis Data Streams → Data Firehose → S3 Tables, וארבעה צרכנים בצד הקריאה

הפייפליין שהוצג מורכב משלושה שלבים. ראשית Amazon Kinesis Data Streams קולט את זרם האירועים ממקורות שונים ומבצע סינון ראשוני. אחריו Amazon Data Firehose מקבץ אירועים לקובץ אחד וממיר אותם מ-JSON לפורמט עמודתי דחוס — במקרה הזה Parquet. הדובר הדגיש שהבחירה בכלים אינה מחייבת: "אפשר להשתמש כמובן באפצ'י קפקא, אפשר להשתמש באפצ'י פלינק ובכל ספרק סטרימינג שהוא והעיקרון שם יהיה זהה" [3:01].

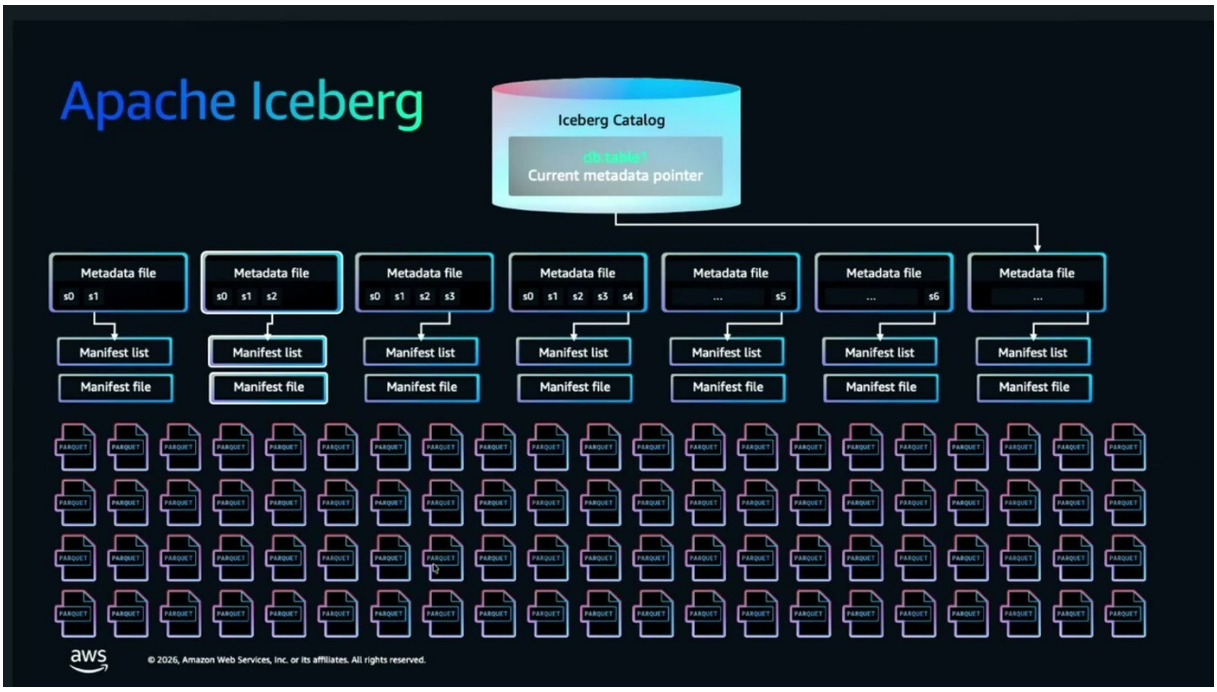
השלב השלישי הוא היעד. במקום bucket שמחזיק ערימת קבצים, S3 Tables הוא שכבת אחסון טבלאית עם Apache Iceberg מובנה — הנתונים נוחתים ומנוהלים כטבלה.

> **[3:01]** קומפק שניקוי סנפשוטים, מחיקת קבצים נתומים, הכל אוטומטי. בלי קרוניוב, בלי ספרק ג'וב, בלי לגעת.

בצד הקריאה הוצגו ארבעה סוגי צרכנים: Amazon Athena כמנוע SQL סרברלס לשאילתות ישירות, Amazon EMR להרצת Spark jobs, Amazon SageMaker לאנליטיקה ולכלי AI, וסביבת פיתוח — על הסלייד Kiro IDE. הדובר הבהיר שזה לא רשימה סגורה ולא בלעדית ל-AWS: אפשר להתחבר גם בכלי third party, וכל agent שבניתם יכול לתחקר את הטבלאות "באמצעות חיבור של MCP למנוע" [4:37].

4. מה זה בעצם Apache Iceberg

לפני שדיברו על הפתרון, הדובר הקדיש כמה דקות למבנה הפנימי — כי ממנו נגזרים שלושת האתגרים. Iceberg הוא, בהגדרתו, "פורמט טבלות פתוח לניהול כמויות ענק של מידע שמאפשר לעבוד איתם כמו טבלת SQL רגיל" [4:37], והוא בנוי משלוש שכבות.



שלוש השכבות: catalog עם מצביע ל-metadata נוכחי, שרשרת metadata files ו-manifests, ותחתיהן קובצי Parquet

- **Iceberg Catalog.** נקודת הכניסה לכל שאילתה או כתיבה. הוא מצביע על ה-metadata הנוכחי, ומספק תכונות ACID "כמו שאנחנו מצפים מכל Data Bay שאנחנו עובדים איתו" [6:12].
- **שכבת ה-Metadata.** "השכבה השנייה, שריבת המטה-Data, מורכבת למעשה משלושה קבצים" [6:12]: metadata file ראשי שמחזיק סכמות, הגדרות, partitions ורשימת manifest list; snapshots שמחזיקה את תמונת המצב הנוכחית וסטטיסטיקות ברמת partition; ו-manifest file שמצביע על הקבצים עצמם ושומר סטטיסטיקה ברמת עמודה — min ו-max לכל עמודה — כדי לדלג על קבצים לא רלוונטיים.
- **שכבת הדאטה.** קובצי ה-Parquet עצמם, כל אחד עומד בפני עצמו. הדובר ציין שאפשר גם Avro או ORC.

מה המבנה הזה קונה? schema evolution פשוט, time travel ("לראות את הדלבס כמו שהוא היה אתמול ושלשום" [7:43]), ו-partition pruning. אבל הוא גם מייצר שלוש בעיות תפעוליות.

5. שלושת אתגרי התחזוקה

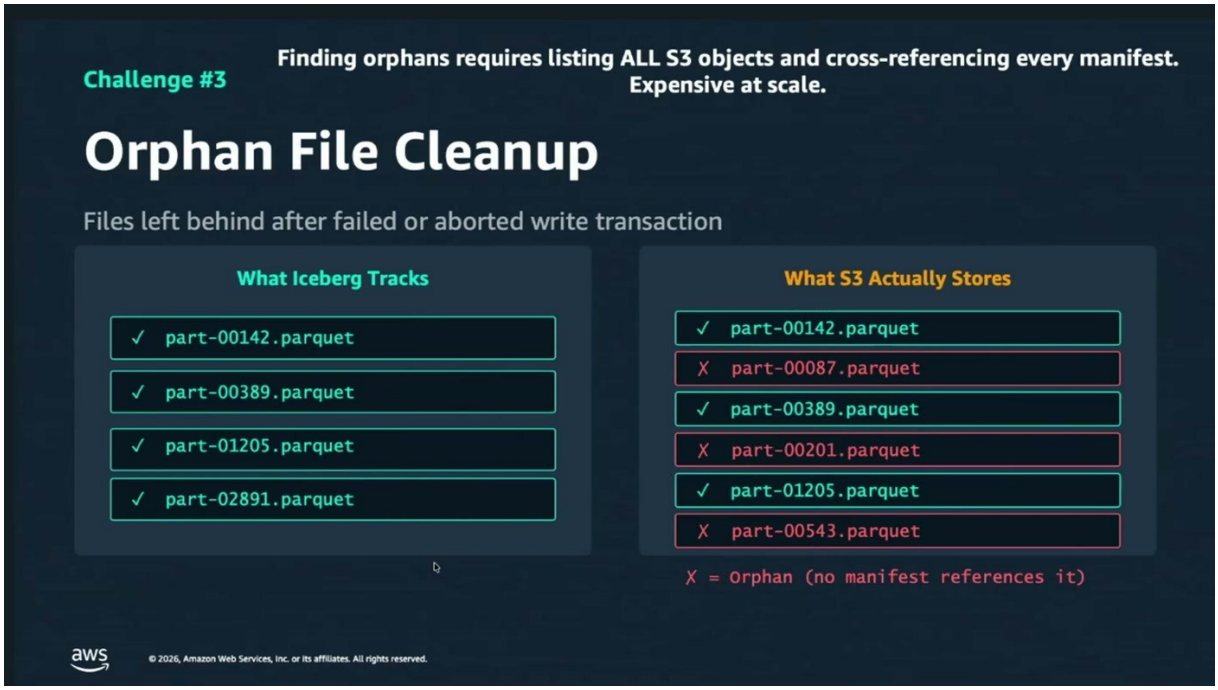
אתגר 1 — Compaction

אם כותבים קובץ כל שנייה, אחרי עשרה ימים יש מאות אלפי קבצים קטנים. הדובר אמר בהקלטה "עדיין אחרי עשרה ימים, אוקיי, יש לו מעל שמונה מאות אלף קבצים" [7:43], בעוד שסלייד הסיכום נוקב במספר של 67K+ small files — שתי המחשות לאותה תופעה, בסדרי גודל שונים. התוצאה זהה בשני המקרים: "הביצועים שלנו בקירה הלכו והידרדרו" [7:43]. הפתרון המסורתי הוא Spark job מתוזמן שמאחד קבצים קטנים לגדולים — וגם מתחזק את ה-manifests שמעליהם.

אתגר 2 — Snapshot Cleanup

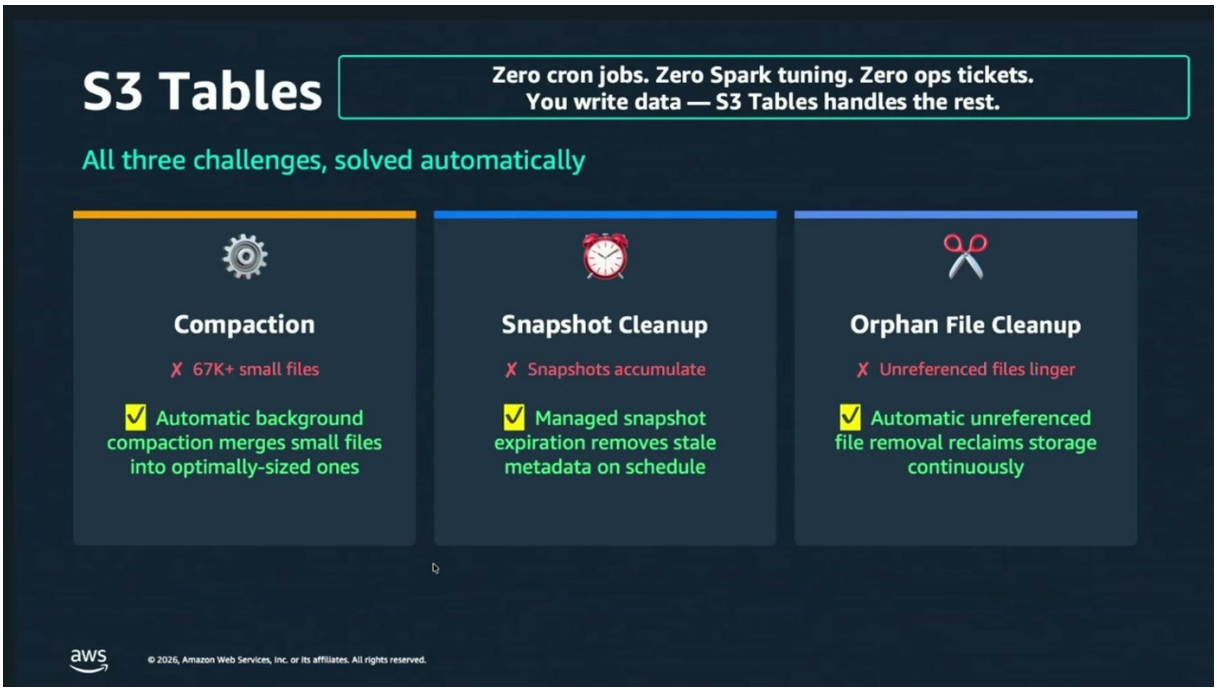
כל כתיבה יוצרת snapshot. אחרי שהארגון מחליט כמה snapshots לשמור אחורה, צריך מנגנון שלא רק מנתק אותם מהקטלוג אלא גם מוחק אותם בפועל — ומטפל בכישלון באמצע התהליך, כולל retries. "בגישה המפוררותית, זה עוד קרון ג'וב שצריך לתחזק" [9:15].

אתגר 3 — Orphan File Cleanup



מה Iceberg עוקב אחריו מול מה ש-S3 באמת מחזיק — ההפרש הוא קבצים יתומים שאף manifest לא מצביע עליהם

קבצים נותרים ב-S3 בלי שאף metadata מצביע עליהם — אחרי כתיבה שנכשלה או הופסקה, אחרי compaction שנקטע, או אחרי snapshot expiry. הם לא פוגעים בנכונות התוצאות: "הם לא גורמים לבעיה, אוקיי? מבחינת עיבוד מידע, הם פשוט נופסים מקום" [9:15]. הבעיה היא באיתור שלהם — צריך להשוות את כל האובייקטים ב-S3 מול כל ה-manifests, ו"בתבלאות גדולות מאוד, ה-list operations הזה מאוד יקר" [10:47].

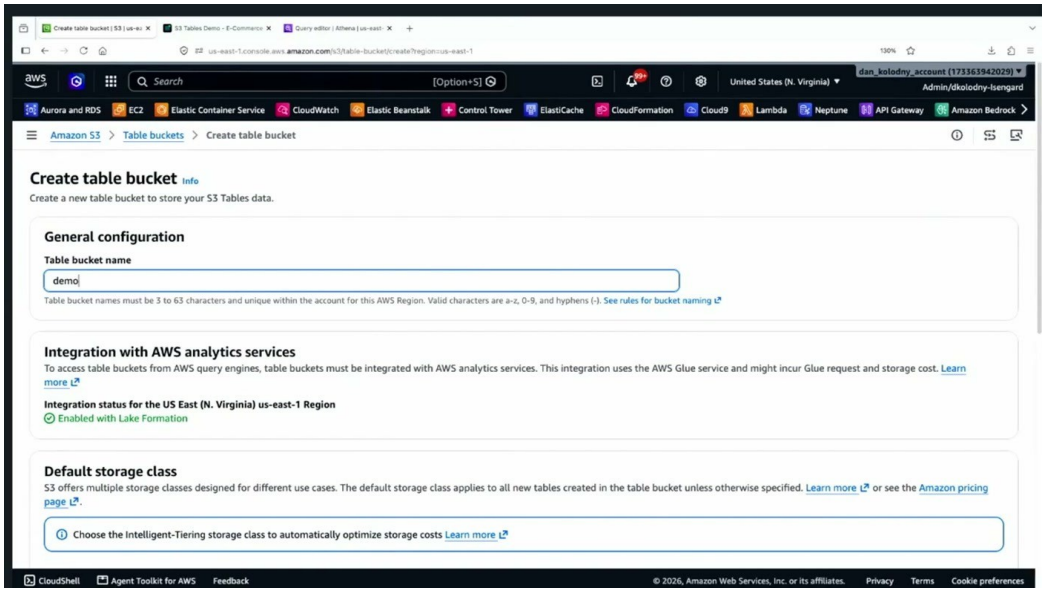


שלוש הבעיות ומול כל אחת מנגנון מנוהל: "Zero cron jobs. Zero Spark tuning. Zero ops tickets".

[10:47] — יש לנו שלושה אתגרים, שלושה תהליכים, אפס ערך עסקי, רק תחזוקה, וזה בדיוק מה ש-Amazon S3 Table פותר עבורכם אוטומטית

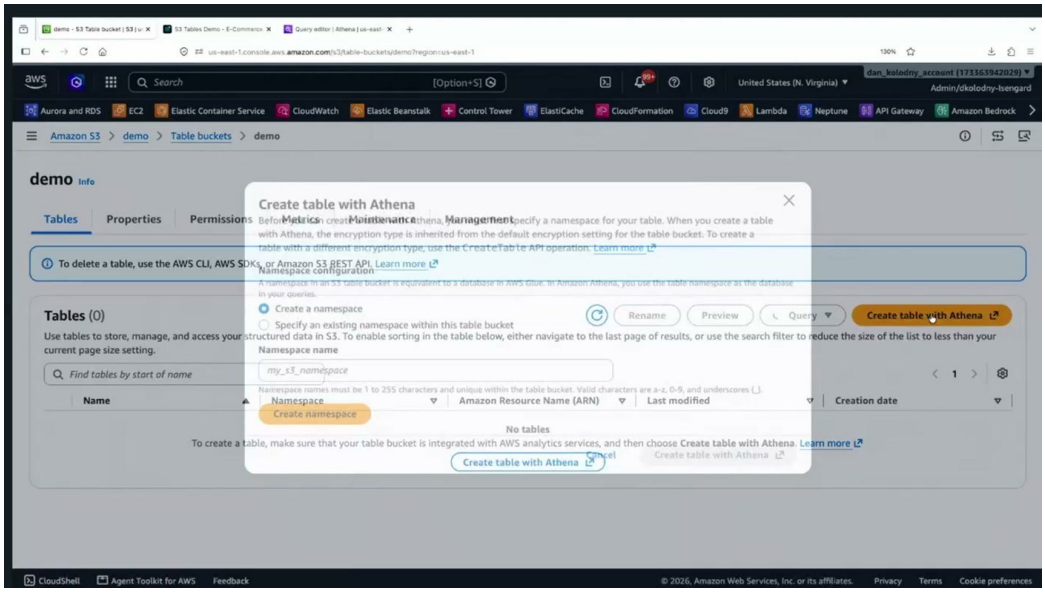
6. דמו, חלק א׳: יצירת טבלת Iceberg מהקונסול

הדמו חולק לשלושה חלקים: בנייה של טבלת Iceberg ב-S3 Tables מהקונסול; השוואה בין אותו דאטה שנכתב פעם ל-table bucket ופעם ל-bucket רגיל; ותחקור הטבלאות בשפה חופשית.



מסך Create table bucket: שם הדלי, סטטוס האינטגרציה עם שירותי האנליטיקה, ובחירת storage class

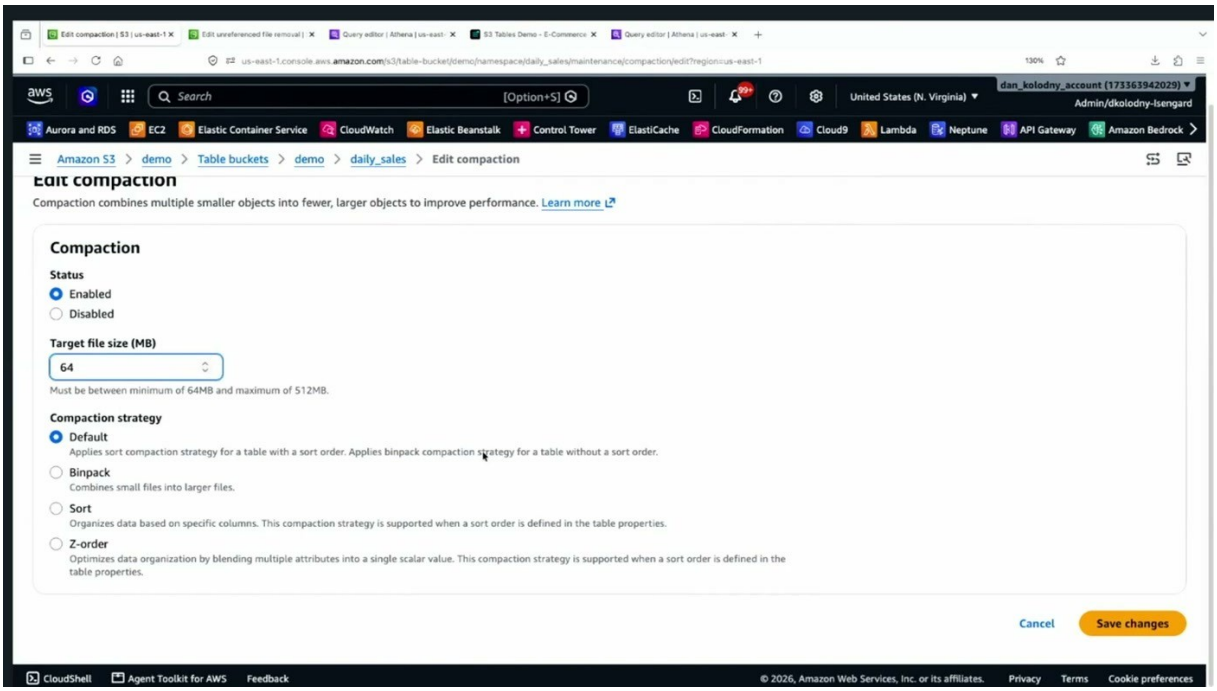
שלב ראשון, יצירת table bucket בשם demo. הקונסול מציע Standard או Intelligent-Tiering, והמלצת הדובר הייתה חד-משמעית: "תמיד כדאי לבחור Intelligent Tiring שעובדים עם S3 Tables" [12:18].



יצירת namespace והפעלת Create table with Athena מתוך ה-table bucket הריק

בתוך הדלי מגדירים namespace — "ניימספייס זה כמו עוד פולדר, עוד איזה קונטיינר לוגי שבתוכו אני כותב את הטבלעות" [12:18] — ואז הקונסול מציע ליצור טבלה ישירות מ-Athena. בדמו נוצרה טבלה בשם daily_sales עם שלוש עמודות, הורצה פקודת INSERT INTO עם תשע רשומות, ושאילתה אגרגטיבית החזירה את התוצאה — "שיש לנו שני לפטופים ושני מוניטורים וקיבורד אחד" [13:50].

7. דמו, חלק ב׳: הידיות שנשארות בשליטתכם



מסך Edit compaction: סטטוס, target file size ובחירה בין Default, Binpack, Sort ו-Z-order

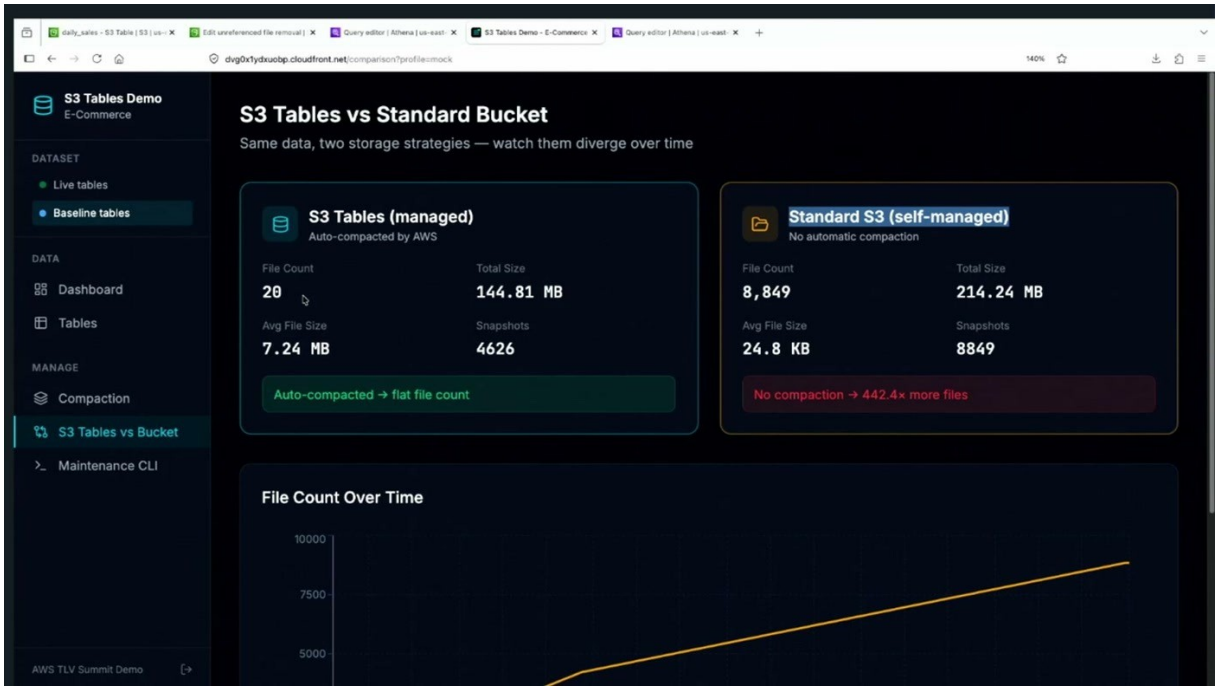
בטאב ה-Maintenance של הטבלה יושבות שלוש הפעולות שיוסי תיאר: unreferenced file removal, compaction ו-snapshot management. הדובר הדגיש שההגדרות הן ברמת ה-bucket ולא ברמת הטבלה הבודדת, ושכל אחת מהן ניתנת לכיבוי — אם כי, בלשונו, "סביר מאוד להניח שאני כן ארצה שהוא יעשה" [13:50] compaction.

ההגדרות שהוצגו:

- **מחיקת קבצים לא מוצבעים.** אחרי כמה ימים קובץ שאין לו מצביעים הופך מועמד למחיקה — בדמו הוצגו ערכים של 3 ו-10 ימים.
- **Target file size.** בקונסול נראה ערך של 64MB, עם הערה שהטווח המותר הוא בין 64MB ל-512MB.
- **אסטרטגיית compaction.** "יש לי פה או binpac או sort או Z-order, זה אומר באיזה אסטרטגיה הוא יקבץ את הקבצים ביחד" [13:50]. בדמו נבחר binpack לשם הפשטות.
- **Snapshot expiration.** "הדיפולט זה 120 שעות, כלומר הוא ישמור לחמישה ימים אחורה את ה-version של הטבלה" [15:22], ולצדו מספר מינימלי של snapshots לשימור — פרמטר שגובר על חלון הזמן.

8. דמו, חלק ג׳: אותו דאטה, שתי אסטרטגיות אחסון

זהו החלק המשכנע ביותר בסשן. אותו זרם נתונים בדיוק נכתב במקביל לשני יעדים — table bucket מסוג S3 Tables ו-bucket סטנדרטי — ואיש לא נגע באף אחד מהם לאחר מכן.



לוח ההשוואה: מספר קבצים, נפח, גודל קובץ ממוצע ומספר snapshots בכל אחת משתי האסטרטגיות

מדד	S3 Tables (מנוהל)	S3 סטנדרטי (ניהול עצמי)
מספר קבצים	20	8,849
נפח כולל	MB 144.81	MB 214.24
גודל קובץ ממוצע	MB 7.24	KB 24.8
מספר snapshots	4626	8849

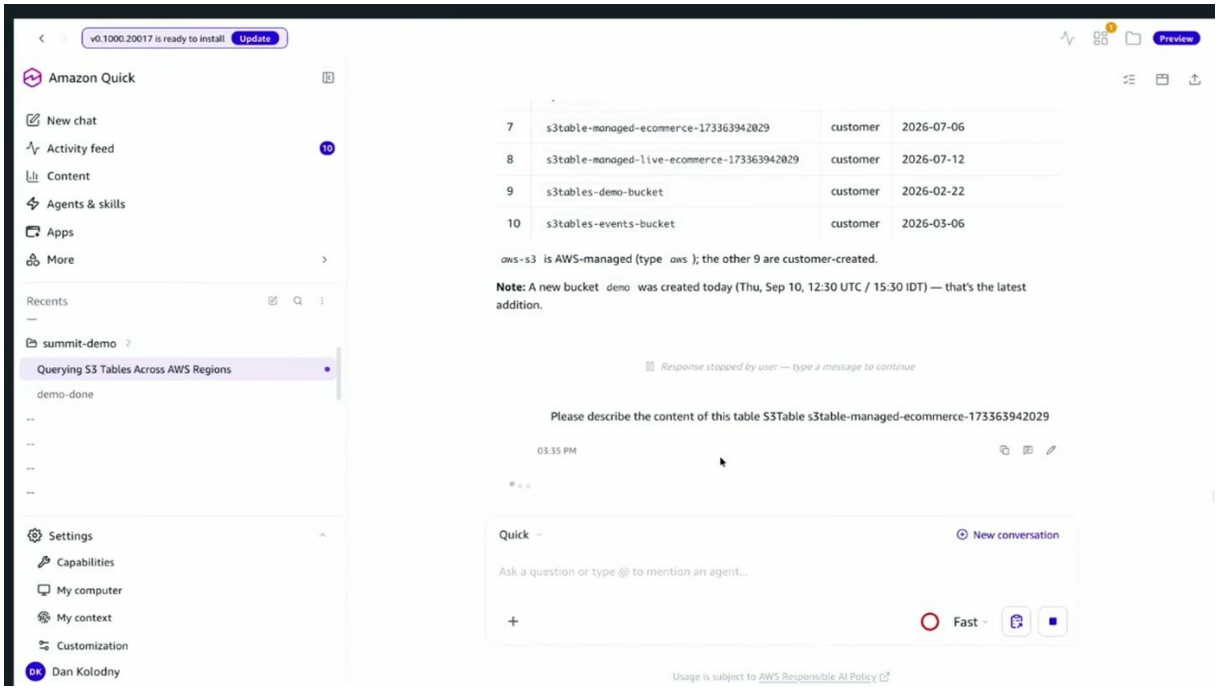
הפער שהלוח מציג מסוכם על המסך כ-"442.4× more files" בצד הלא מנוהל. הדובר הקריא את המספרים מהמסך: "שימו לב פה יש לנו 20 קבצים פה יש לנו 8850 כמעט" [15:22], ובסיכום — "אתם יכולים לראות פה יש כמעט 9,000 קבצים אל מול 20" [15:22].

גרף File Count Over Time מתחת ללוח ממחיש את ההתנהגות לאורך זמן: הקו הצהוב, של ה-bucket הרגיל, מטפס באופן מונוטוני; הקו של ה-table bucket נשאר שטוח, ויורד בכל פעם שרץ compaction. "בשתי המקרים לא נגעתי. זה עושה את זה לבד, וזה אחד מהיתרונות המאוד משמעותיים של ה-[15:22] "S3 Table.

נקודה שקל לפספס שימו לב שגם הנפח הכולל קטן יותר בצד המנוהל (144.81MB מול 214.24MB) — כלומר החיסכון אינו רק בביצועי קריאה אלא גם בעלות אחסון, בזכות דחיסה יעילה יותר בקבצים גדולים ומחיקת קבצים לא מוצבעים.

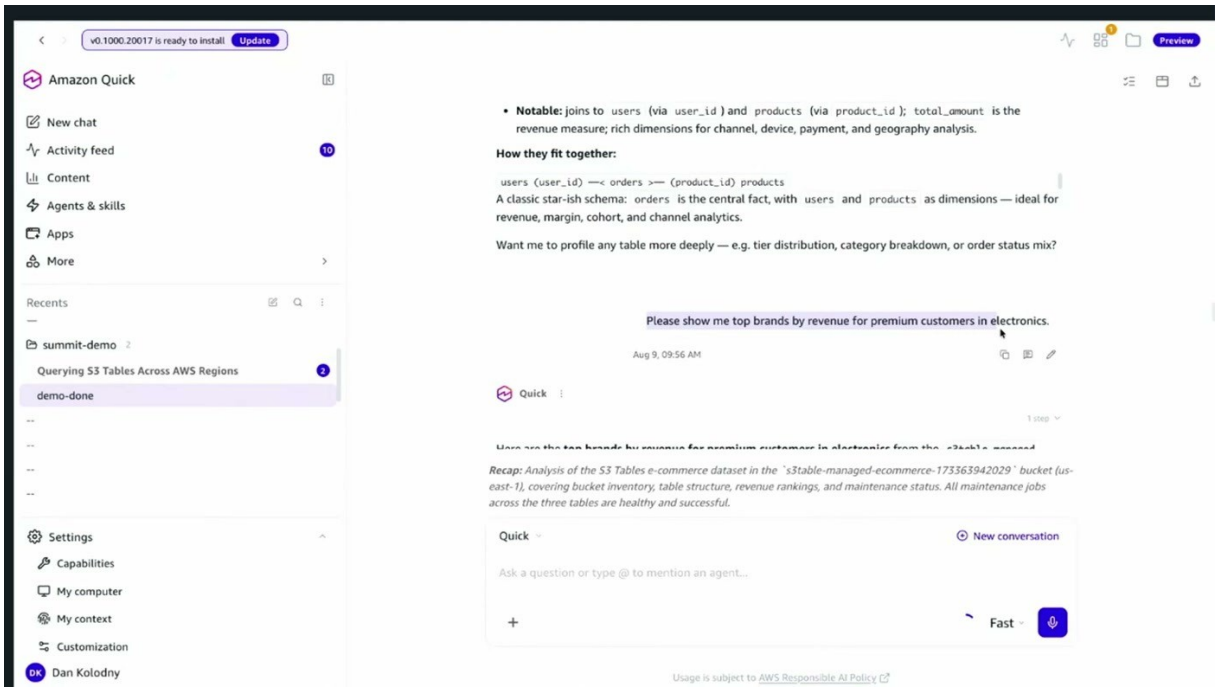
9. דמו, חלק ד׳: תחקור בשפה חופשית

החלק האחרון הראה חיבור של עוזר Amazon Quick — AI, שהוצג כ-"ה-Gen AI Assistance שלנו ב-"AWS [10:47] — ישירות מול הטבלאות, בשפה חופשית.



תשובת העוזר לשאלת מיפוי: רשימת ה-table buckets באזור, מי מהם מנוהל על ידי AWS ומי נוצר על ידי הלקוח

השאלה הראשונה הייתה מיפוי: אילו table buckets קיימים ב-us-east-1. התשובה כללה עשרה דליים עם תאריכי יצירה, כולל הערה שהדלי demo נוצר באותו יום. השאלה השנייה ביקשה לתאר את תוכן דלי ה-e-commerce, והתשובה פירטה שלוש טבלאות — users, products ו-orders — עם העמודות של כל אחת.



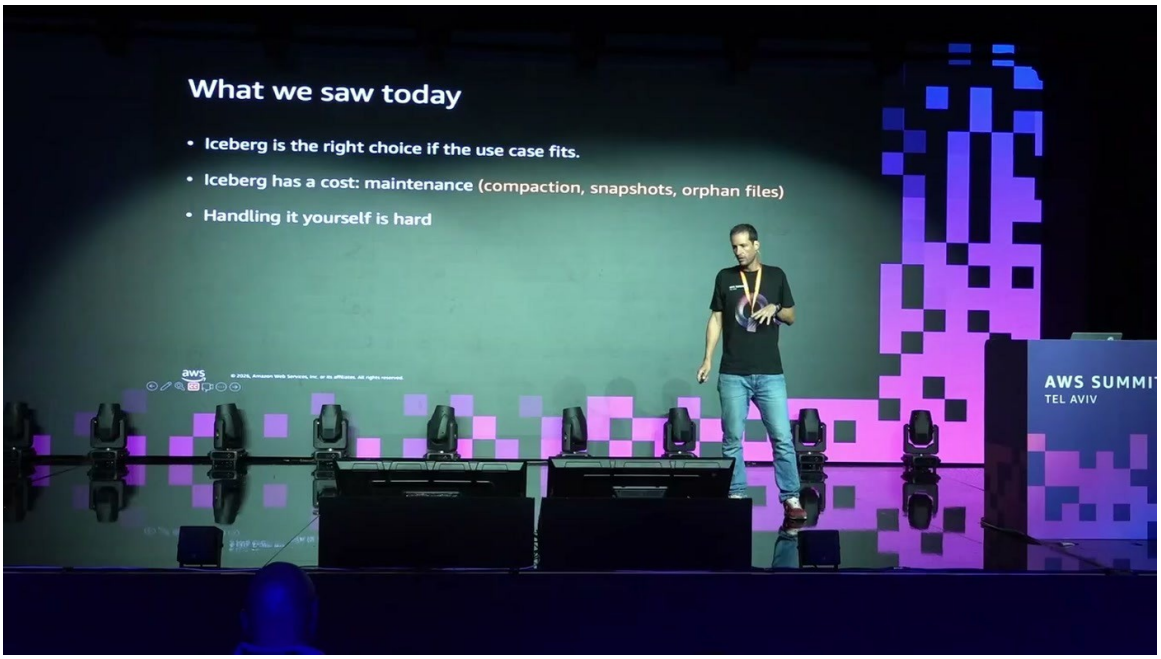
מיפוי הקשרים בין הטבלאות (star schema) ולאחריו שאלה עסקית על מותגים מובילים לפי הכנסה

משם עברו לשאלות עסקיות — המוצרים הנמכרים ביותר ללקוחות פרימיום. הנקודה שהדובר הדגיש היא השקיפות: "ותראו אני גם יכול לבקש, הוא יראה לי איזה שילטה הוא יריץ בדיוק, ככה שזה לא בלקבוקס" [18:34] — אפשר לראות את ה-SQL שנוצר ולאמת אותו.

השאלה האחרונה הייתה תפעולית ולא עסקית, וזו שהמחישה לדעתו את עוצמת S3 Tables: "please show me the last maintenance on this bucket per table" [18:34]. התשובה הציגה, לכל טבלה בנפרד, מתי רץ ה-compaction, מתי רצה הסרת הקבצים ומתי טופלו ה-snapshots — התאריכים שהוצגו היו בתשעה ובחמישה באוגוסט.

> **— [18:34]** בדמו אם זה תמיד נראה קל כי תמיד חושבי שיש טבלה אחת או שתיים או שלוש אבל בעולם האמיתי יש לנו עשרות או מאות טבלות, ולנהל את כל התהליכים האלה עבור כל טבלה לבד זה מאוד קשה

10. מה לוקחים מכאן



שלוש שורות הסיכום שהדובר הציג בסוף: מתי Iceberg מתאים, מה המחיר שלו, ולמה הניהול העצמי קשה

- **Iceberg הוא בחירה טובה — אם ה-use case מתאים.** "אייסברג הוא באמת יכול להיות הבחירה הטובה אם ה-use case מתאים לנו" [18:34]. הסשן לא טען שזה פורמט לכל מקרה.
- **למחיר התחזוקה יש שם ומספר.** compaction, snapshot expiry ו-orphan file cleanup. אם אתם מריצים Iceberg בעצמכם היום, שאלו מי כותב ומנטר את שלושת ה-jobs האלה ומה קורה כשאחד מהם נכשל באמצע.
- **S3 Tables מסיר את שלושת התהליכים בלי לוותר על שליטה.** ההגדרות נשארות חשופות בקונסול — הפעלה, כיבוי, target file size, אסטרטגיה ומדיניות snapshots.
- **הכאב מתעצם ביניארית עם מספר הטבלאות.** על טבלה אחת אפשר לחיות עם cron job; על עשרות או מאות, זה הופך לעבודה קבועה.
- **בדיקת היתכנות פשוטה.** ההמלצה הסוגרת של הדוברים הייתה לקחת workload קיים שרץ על Iceberg, להריץ אותו על S3 Tables ולהשוות — בדיוק כמו ההשוואה שהוצגה על הבמה [20:11].
- **ה-REST Catalog ו-MCP הם נקודת החיבור לכלים אחרים.** כל לקוח תואם Iceberg יכול לגשת לאותה טבלה, וכל agent יכול לתחקר אותה דרך MCP — בלי שכפול נתונים.

11. מילון מונחים

מונח	פירוש
Apache Iceberg	פורמט טבלאות פתוח לכמויות מידע גדולות, שמאפשר לעבוד מעל אחסון אובייקטים כמו מול טבלת SQL רגילה, כולל ACID.
S3 Table bucket	סוג דלי ייעודי ב-Amazon S3 שמחזיק טבלאות Iceberg מנוהלות במקום אובייקטים חופשיים.
Namespace	מכל לוגי בתוך table bucket, מקביל ל-database ב-AWS Glue, שבתוכו יושבות הטבלאות.
Compaction	איחוד קבצים קטנים לקבצים גדולים יותר כדי לשפר ביצועי קריאה. ב-S3 Tables רץ ברקע אוטומטית.
Binpack / Sort / Z-order	שלוש אסטרטגיות לאיחוד קבצים: איחוד פשוט לפי גודל, סידור לפי עמודות, ומיזוג כמה מאפיינים לערך סקלרי אחד.
Snapshot	תמונת מצב של הטבלה שנוצרת בכל כתיבה, ומאפשרת time travel לגרסאות קודמות.
Orphan file	קובץ שנותר ב-S3 אחרי כתיבה או compaction שנכשלו, שאף manifest אינו מצביע עליו. תופס מקום ועולה כסף.

מונח	פירוש
Manifest list / Manifest file	שכבות ה-metadata של Iceberg: הראשונה מחזיקה את תמונת המצב וסטטיסטיקות partition, השנייה מצביעה על קובצי הדאטה ושומרת סטטיסטיקה ברמת עמודה.
Iceberg REST Catalog	תקן פתוח לגישה לקטלוג, שמאפשר לכל לקוח תואם להתחבר לאותן טבלאות.
MCP	פרוטוקול שמאפשר לעוזרי AI ולסוכנים להתחבר למקורות מידע וכלים — כאן, לתחקר את טבלאות ה-Iceberg.
Intelligent-Tiering	מחלקת אחסון שמעבירה אוטומטית אובייקטים שלא נעשה בהם שימוש לשכבה זולה יותר.