Snowflake security audit - SF-LAT-20260427-004934

Account: titan-snowflake-prod-20260427
Generated: 2026-04-27T00:48:36+00:00
Mode: native

Total: 54 - Critical: 9 - High: 31 - Medium: 14

[CRITICAL] Users without MFA enrolled

Resource: TITANBAD_STALE

Citation: HIPAA 164.312(d) Person or Entity Authentication; CIS Snowflake 1.4

Recommendation: Enroll all human users in Duo or Snowflake-managed MFA. Service users should use RSA key-pair auth, not password+MFA.

Fix: `ALTER USER SET MINS_TO_BYPASS_MFA = 0; -- then have user enroll in MFA`

[CRITICAL] Users without MFA enrolled

Resource: TITANBAD_NOMFA2

Citation: HIPAA 164.312(d) Person or Entity Authentication; CIS Snowflake 1.4

Recommendation: Enroll all human users in Duo or Snowflake-managed MFA. Service users should use RSA key-pair auth, not password+MFA.

Fix: `ALTER USER SET MINS_TO_BYPASS_MFA = 0; -- then have user enroll in MFA`

[CRITICAL] Users without MFA enrolled

Resource: TITANBAD_NOMFA1

Citation: HIPAA 164.312(d) Person or Entity Authentication; CIS Snowflake 1.4

Recommendation: Enroll all human users in Duo or Snowflake-managed MFA. Service users should use RSA key-pair auth, not password+MFA.

Fix: `ALTER USER SET MINS_TO_BYPASS_MFA = 0; -- then have user enroll in MFA`

[CRITICAL] Users without MFA enrolled

Resource: TITANADMIN

Citation: HIPAA 164.312(d) Person or Entity Authentication; CIS Snowflake 1.4

Recommendation: Enroll all human users in Duo or Snowflake-managed MFA. Service users should use RSA key-pair auth, not password+MFA.

Fix: `ALTER USER SET MINS_TO_BYPASS_MFA = 0; -- then have user enroll in MFA`

[CRITICAL] Users without MFA enrolled

Resource: SVC_TITANBAD_BROAD

Citation: HIPAA 164.312(d) Person or Entity Authentication; CIS Snowflake 1.4

Recommendation: Enroll all human users in Duo or Snowflake-managed MFA. Service users should use RSA key-pair auth, not password+MFA.

Fix: `ALTER USER SET MINS_TO_BYPASS_MFA = 0; -- then have user enroll in MFA`

[HIGH] Users authenticating with password instead of RSA key

Resource: `TITANADMIN`

Citation: CIS Snowflake 1.6; NIST 800-53 IA-5

Recommendation: Move service accounts to RSA key-pair authentication. Passwords are weaker, harder to rotate, and not auditable to per-call granularity.

Fix: `ALTER USER SET RSA_PUBLIC_KEY=''; ALTER USER UNSET PASSWORD;`

[HIGH] Users authenticating with password instead of RSA key

Resource: `TITANBAD_NOMFA1`

Citation: CIS Snowflake 1.6; NIST 800-53 IA-5

Recommendation: Move service accounts to RSA key-pair authentication. Passwords are weaker, harder to rotate, and not auditable to per-call granularity.

Fix: `ALTER USER SET RSA_PUBLIC_KEY=''; ALTER USER UNSET PASSWORD;`

[HIGH] Users authenticating with password instead of RSA key

Resource: `TITANBAD_NOMFA2`

Citation: CIS Snowflake 1.6; NIST 800-53 IA-5

Recommendation: Move service accounts to RSA key-pair authentication. Passwords are weaker, harder to rotate, and not auditable to per-call granularity.

Fix: `ALTER USER SET RSA_PUBLIC_KEY=''; ALTER USER UNSET PASSWORD;`

[HIGH] Users authenticating with password instead of RSA key

Resource: `TITANBAD_STALE`

Citation: CIS Snowflake 1.6; NIST 800-53 IA-5

Recommendation: Move service accounts to RSA key-pair authentication. Passwords are weaker, harder to rotate, and not auditable to per-call granularity.

Fix: `ALTER USER SET RSA_PUBLIC_KEY=''; ALTER USER UNSET PASSWORD;`

[HIGH] Users authenticating with password instead of RSA key

Resource: `SVC_TITANBAD_BROAD`

Citation: CIS Snowflake 1.6; NIST 800-53 IA-5

Recommendation: Move service accounts to RSA key-pair authentication. Passwords are weaker, harder to rotate, and not auditable to per-call granularity.

Fix: `ALTER USER SET RSA_PUBLIC_KEY=''; ALTER USER UNSET PASSWORD;`

[MEDIUM] Stale users (no login in 90 days)

Resource: `SVC_TITANBAD_BROAD`

Citation: CIS Snowflake 1.10; NIST 800-53 AC-2(3)

Recommendation: Disable users with no login in 90 days. Service accounts that legitimately do not log in should be tagged so they are not flagged.

Fix: `ALTER USER SET DISABLED = TRUE;`

[MEDIUM] Stale users (no login in 90 days)

Resource: `TITANBAD_NOMFA2`

Citation: CIS Snowflake 1.10; NIST 800-53 AC-2(3)

Recommendation: Disable users with no login in 90 days. Service accounts that legitimately do not log in should be tagged so they are not flagged.

Fix: `ALTER USER SET DISABLED = TRUE;`

[MEDIUM] Stale users (no login in 90 days)

Resource: `TITANBAD_STALE`

Citation: CIS Snowflake 1.10; NIST 800-53 AC-2(3)

Recommendation: Disable users with no login in 90 days. Service accounts that legitimately do not log in should be tagged so they are not flagged.

Fix: `ALTER USER SET DISABLED = TRUE;`

[MEDIUM] Stale users (no login in 90 days)

Resource: `TITANBAD_NOMFA1`

Citation: CIS Snowflake 1.10; NIST 800-53 AC-2(3)

Recommendation: Disable users with no login in 90 days. Service accounts that legitimately do not log in should be tagged so they are not flagged.

Fix: `ALTER USER SET DISABLED = TRUE;`

[HIGH] Account-level network policy not configured

Resource: `account`

Citation: HIPAA 164.312(e)(1) Transmission Security; CIS Snowflake 1.13

Recommendation: Apply at least one account-level network policy that restricts logins to corporate egress IPs and approved cloud IP ranges.

Fix: `CREATE NETWORK POLICY corp_only ALLOWED_IP_LIST=('1.2.3.4/32', ...); ALTER ACCOUNT SET NETWORK_POLICY = corp_only;`

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.PATIENT_ID`

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;`

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT.MEMBER_PII_SSN

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.SSN

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.MEMBER_ID

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.DOB

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.MRN

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.PHONE

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT.MEMBER_PII_DOB

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[HIGH] Likely-PHI / PII columns without a masking policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.EMAIL

Citation: HIPAA 164.502 Uses and Disclosures of PHI; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values. Continuous re-scan flags new columns as schemas evolve.

Fix: CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SET MASKING POLICY phi_redact;

[MEDIUM] Sensitive tables without a row-access policy

Resource: SNOWFLAKE.TRUST_CENTER.ACCOUNT_NOTIFICATION_RECIPIENTS

Citation: HIPAA 164.312(a)(1); CIS Snowflake 4.6

Recommendation: Apply a row-access policy on PHI / patient / member / financial tables so each role sees only its in-scope rows.

Fix: CREATE ROW ACCESS POLICY tenant_scope AS (tenant STRING) RETURNS BOOLEAN -> tenant = CURRENT_ROLE(); ALTER TABLE ADD ROW ACCESS POLICY tenant_scope ON (tenant_col);

[MEDIUM] Sensitive tables without a row-access policy

Resource: TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT

Citation: HIPAA 164.312(a)(1); CIS Snowflake 4.6

Recommendation: Apply a row-access policy on PHI / patient / member / financial tables so each role sees only its in-scope rows.

Fix: CREATE ROW ACCESS POLICY tenant_scope AS (tenant STRING) RETURNS BOOLEAN -> tenant = CURRENT_ROLE(); ALTER TABLE ADD ROW ACCESS POLICY tenant_scope ON (tenant_col);

[MEDIUM] Sensitive tables without a row-access policy

Resource: SNOWFLAKE.TRUST_CENTER_STATE.ACCOUNT_NOTIFICATION_METADATA

Citation: HIPAA 164.312(a)(1); CIS Snowflake 4.6

Recommendation: Apply a row-access policy on PHI / patient / member / financial tables so each role sees only its in-scope rows.

```
Fix: CREATE ROW ACCESS POLICY tenant_scope AS (tenant STRING) RETURNS BOOLEAN -> tenant = CURRENT_ROLE();
ALTER TABLE ADD ROW ACCESS POLICY tenant_scope ON (tenant_col);
```

[MEDIUM] Sensitive tables without a row-access policy

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI

Citation: HIPAA 164.312(a)(1); CIS Snowflake 4.6

Recommendation: Apply a row-access policy on PHI / patient / member / financial tables so each role sees only its in-scope rows.

```
Fix: CREATE ROW ACCESS POLICY tenant_scope AS (tenant STRING) RETURNS BOOLEAN -> tenant = CURRENT_ROLE();
ALTER TABLE ADD ROW ACCESS POLICY tenant_scope ON (tenant_col);
```

[MEDIUM] Sensitive tables without a row-access policy

Resource: SNOWFLAKE.TRUST_CENTER_STATE.ACCOUNT_NOTIFICATION_HISTORY

Citation: HIPAA 164.312(a)(1); CIS Snowflake 4.6

Recommendation: Apply a row-access policy on PHI / patient / member / financial tables so each role sees only its in-scope rows.

```
Fix: CREATE ROW ACCESS POLICY tenant_scope AS (tenant STRING) RETURNS BOOLEAN -> tenant = CURRENT_ROLE();
ALTER TABLE ADD ROW ACCESS POLICY tenant_scope ON (tenant_col);
```

[HIGH] Authentication policy not configured at account level

Resource: account

Citation: PCI-DSS 8.2; NIST 800-53 IA-2; HIPAA 164.308(a)(5)(ii)(D)

Recommendation: Apply an authentication policy at account level that requires MFA, restricts auth methods to PASSWORD+MFA / KEYPAIR / SAML, and disables legacy paths.

```
Fix: CREATE AUTHENTICATION POLICY require_mfa ALLOWED_AUTHENTICATION_METHODS=('PASSWORD','KEYPAIR','SAML')
MFA_AUTHENTICATION_METHODS=('PASSWORD'); ALTER ACCOUNT SET AUTHENTICATION POLICY require_mfa;
```

[MEDIUM] Session policy missing or idle timeout above 60 minutes

Resource: no_session_policy

Citation: PCI-DSS 8.1.8; NIST 800-53 AC-11; HIPAA 164.312(a)(2)(iii)

Recommendation: Apply a session policy with idle timeout no greater than 30 minutes (15 for healthcare). PCI requires 15 minutes for cardholder-data environments.

```
Fix: CREATE SESSION POLICY tight_idle SESSION_IDLE_TIMEOUT_MINS=15; ALTER ACCOUNT SET SESSION POLICY
tight_idle;
```

[MEDIUM] Sensitive tables with Time Travel retention below 7 days

Resource: TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI

Citation: HIPAA 164.316(b)(2)(i); PCI-DSS 10.5.3; SOC 2 CC7.3

Recommendation: Sensitive tables (PHI, transactions, audit logs) should hold at least 7 days of Time Travel for incident reconstruction. Compliance frameworks frequently require longer log retention.

Fix: ALTER TABLE SET DATA_RETENTION_TIME_IN_DAYS = 30;

[HIGH] Customer-managed encryption keys (Tri-Secret Secure) not enabled

Resource: account

Citation: HIPAA 164.312(a)(2)(iv); FedRAMP SC-12; PCI-DSS 3.5.2

Recommendation: Enable Tri-Secret Secure so encryption requires a customer-controlled KMS key. This is a Business Critical / Enterprise-tier feature; if your account is on a lower tier and you handle PHI / cardholder data, request the upgrade.

Fix: Contact Snowflake support to enable Tri-Secret Secure on the account; provide your AWS KMS / Azure Key Vault / GCP KMS key reference.

[MEDIUM] No replication or failover groups for HIPAA / PCI workloads

Resource: account

Citation: HIPAA 164.308(a)(7) Contingency; PCI-DSS 12.10.1; SOC 2 A1.2

Recommendation: Configure at least one failover group covering the regulated databases. RPO and RTO requirements (HIPAA: 24h backup target) are not satisfied by Time Travel alone.

Fix: CREATE FAILOVER GROUP regulated_data OBJECT_TYPES=(DATABASES, ROLES) ALLOWED_DATABASES=() ALLOWED_ACCOUNTS=() REPLICATION_SCHEDULE='10 MINUTE';

[MEDIUM] Recent Cortex AI calls invoked from likely-PHI tables

Resource: TITANADMIN

Citation: HIPAA 164.502; NIST AI RMF GV-2

Recommendation: Cortex AI calls send data to Snowflake's hosted LLM. For PHI workloads, confirm the BAA covers Cortex inference and that PHI is masked or tokenized before the call.

Fix: Apply masking policy to PHI columns before they can be passed to CORTEX functions; add a row-access policy that hides PHI from the role used by Cortex callers.

[HIGH] Users without MFA enrolled

Resource: SVC_TITANBAD_BROAD

Citation: HIPAA 164.312(d); CIS Snowflake 1.4

Recommendation: Enrol the user in Duo or Snowflake-managed MFA. Service users must use RSA key-pair auth instead.

Fix: ALTER USER SVC_TITANBAD_BROAD SET MINS_TO_BYPASS_MFA = 0; -- then enrol in MFA

[CRITICAL] Users without MFA enrolled

Resource: TITANADMIN

Citation: HIPAA 164.312(d); CIS Snowflake 1.4

Recommendation: Enrol the user in Duo or Snowflake-managed MFA. Service users must use RSA key-pair auth instead.

Fix: `ALTER USER TITANADMIN SET MINS_TO_BYPASS_MFA = 0; -- then enrol in MFA`

[HIGH] Users without MFA enrolled

Resource: `TITANBAD_NOMFA1`

Citation: HIPAA 164.312(d); CIS Snowflake 1.4

Recommendation: Enrol the user in Duo or Snowflake-managed MFA. Service users must use RSA key-pair auth instead.

Fix: `ALTER USER TITANBAD_NOMFA1 SET MINS_TO_BYPASS_MFA = 0; -- then enrol in MFA`

[HIGH] Users without MFA enrolled

Resource: `TITANBAD_NOMFA2`

Citation: HIPAA 164.312(d); CIS Snowflake 1.4

Recommendation: Enrol the user in Duo or Snowflake-managed MFA. Service users must use RSA key-pair auth instead.

Fix: `ALTER USER TITANBAD_NOMFA2 SET MINS_TO_BYPASS_MFA = 0; -- then enrol in MFA`

[HIGH] Users without MFA enrolled

Resource: `TITANBAD_STALE`

Citation: HIPAA 164.312(d); CIS Snowflake 1.4

Recommendation: Enrol the user in Duo or Snowflake-managed MFA. Service users must use RSA key-pair auth instead.

Fix: `ALTER USER TITANBAD_STALE SET MINS_TO_BYPASS_MFA = 0; -- then enrol in MFA`

[CRITICAL] PUBLIC role granted on regulated data object

Resource: `TITAN_DEMO.PUBLIC_BAD`

Citation: HIPAA 164.502; PCI-DSS 7.1; CIS Snowflake 2.1

Recommendation: Revoke the privilege from PUBLIC. PUBLIC is granted to every role automatically; non-default grants on PUBLIC make data world-readable inside the account.

Fix: `REVOKE USAGE ON SCHEMA TITAN_DEMO.PUBLIC_BAD FROM ROLE PUBLIC;`

[CRITICAL] PUBLIC role granted on regulated data object

Resource: `TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT`

Citation: HIPAA 164.502; PCI-DSS 7.1; CIS Snowflake 2.1

Recommendation: Revoke the privilege from PUBLIC. PUBLIC is granted to every role automatically; non-default grants on PUBLIC make data world-readable inside the account.

Fix: `REVOKE SELECT ON TABLE TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT FROM ROLE PUBLIC;`

[CRITICAL] PUBLIC role granted on regulated data object

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI`

Citation: HIPAA 164.502; PCI-DSS 7.1; CIS Snowflake 2.1

Recommendation: Revoke the privilege from PUBLIC. PUBLIC is granted to every role automatically; non-default grants on PUBLIC make data world-readable inside the account.

Fix: `REVOKE SELECT ON TABLE TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI FROM ROLE PUBLIC;`

[MEDIUM] Role holds 10 distinct privileges

Resource: `R_TITANBAD_EXPLOSION`

Citation: NIST 800-53 AC-6 Least Privilege; CIS Snowflake 2.5

Recommendation: Split this role into purpose-specific sub-roles. A single role with many privileges is hard to audit and tends to accumulate over-grant.

Fix: `Identify each functional purpose; CREATE sub-roles; GRANT specific privileges; over time REVOKE direct grants.`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.EMAIL`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN EMAIL SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.MEMBER_ID`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN MEMBER_ID SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.DOB`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN DOB SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.SSN`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN SSN SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.PHONE`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN PHONE SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT.MEMBER_PII_DOB`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN MEMBER_PII_DOB SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.PATIENT_ID`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN PATIENT_ID SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.MEMBER_ACCOUNT.MEMBER_PII_SSN`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN MEMBER_PII_SSN SET MASKING POLICY phi_redact;`

[HIGH] PHI / PII column without masking policy

Resource: `TITAN_DEMO.PUBLIC_BAD.PATIENT_PHI.MRN`

Citation: HIPAA 164.502; CIS Snowflake 4.5

Recommendation: Apply a masking policy on every column whose name suggests PHI / PII so non-clinician roles see hashed or redacted values.

Fix: `CREATE MASKING POLICY phi_redact AS (val STRING) RETURNS STRING -> CASE WHEN IS_ROLE_IN_SESSION('CLINICIAN_RO') THEN val ELSE 'REDACTED' END; ALTER TABLE MODIFY COLUMN MRN SET MASKING POLICY phi_redact;`

[HIGH] External stage without server-side encryption

Resource: `TITAN_DEMO.PUBLIC_BAD.BAD_EXT_STAGE`

Citation: HIPAA 164.312(a)(2)(iv); CIS Snowflake 4.2

Recommendation: Configure server-side encryption (AWS-KMS or Azure-KV or GCS-CMEK) on every external stage carrying regulated data.

Fix: `CREATE OR REPLACE STAGE TITAN_DEMO.PUBLIC_BAD.BAD_EXT_STAGE URL='s3://titan-demo-fake-bucket/' STORAGE_INTEGRATION= ENCRYPTION=(TYPE='AWS_SSE_KMS' KMS_KEY_ID='');`