INFOCUS 2026 · DENVER · ERP SUITES

Access-Model Readiness Scorecard

Map-ready, or rebuild-bound? Score your JDE access model before a deal does it for you.

Eight yes/no questions. Circle Y or N for each, total your Y's, then read the score key. Every N is access you would have to rebuild user-by-user when you absorb (or get absorbed by) another company.

		YES	NO
1	Documented, process-aligned role catalog with a naming standard	Y	N
2	SoD analysis run BEFORE provisioning, not discovered after the fact	Y	N
3	*PUBLIC baseline designed on purpose: inquiry granted, everything beyond inquiry denied	Y	N
4	Security applied at the role level, not through user-level overrides	Y	N
5	Roles sequenced in function bands, so the resolving role is predictable before you provision	Y	N
6	Software license transferability and change-of-control mapped	Y	N
7	Security model documented, not held in one or two people's heads	Y	N
8	Recent orphan, duplicate, and privilege-creep cleanup completed	Y	N

SCORE (count your Y's)

7–8	Map-ready
4–6	Partial. Close the gaps before a deal lands.
0–3	Rebuild-bound. Start now.

Map onto a model, or rebuild for every user.

SCAN FOR THE FULL DECK & RESOURCES

PAGE 1 OF 6 · THREE SHEETS · SHEET 2 IS ORACLE, SHEET 3 IS SQL SERVER

Page 2: the six tables and what a bad answer looks like. Pages 3-4: the 19 queries in Oracle. Pages 5-6: the same 19 in SQL Server.

The Six Tables

What absorb-ready looks like at the table level. Run these against production.

Six tables tell you whether your access model can be mapped onto or has to be rebuilt. Write each number in the box. If you cannot produce these six on Monday, that is itself your answer to the scorecard on the other side.

TABLE	WHAT IT HOLDS	WHAT TO LOOK FOR	THE BAD ANSWER	YOUR NUMBER
F00950	Security Workbench	*PUBLIC grants by security type, and the count of user-level security records.	*PUBLIC granting add, change or delete. A large user-level record count.	
F00926	Role Sequence	Roles grouped by function, each function in its own band. Higher band wins.	Numbers that mean nothing, so nobody can predict which role wins.	
F95921	Role Relationships	Volume of role assignments, active against expired, and roles per user.	Expired assignments never removed. Users carrying roles nobody can justify.	
F0092	User & Role	The full inventory of users and roles. It will not tell you who is active.	Profiles nobody recognises, and service accounts set up as people.	
F0093	Environment List	Which roles can reach production.	Roles designated non-production that carry production access.	
F98OWSEC	Sign-on security	Active, disabled and locked-out users. Password change frequency and attempt limits.	SCSECFRQ = 0, so passwords never expire. SCATTEMPTS = 0, so accounts never lock.	

TWO TRAPS F95921 will never show a user with no role, because a user with no role has no row in it. Use F0092 for the population. F0092 shows users and roles but not whether an account is active or disabled. That answer is in F98OWSEC.

SEQUENCE ROLES BY FUNCTION, NOT BY ACCIDENT

Give each function its own sequence band. The higher band wins the conflict, so the most permissive role a person legitimately holds is the one that resolves. Predictable before you provision, rather than discovered afterward.

Add, Change, Delete	Add & Change	Add	Inquiry
4000 - 4999	3000 - 3999	2000 - 2999	1000 - 1999

On a baseline architecture for *PUBLIC and all roles, this eliminates roughly 90% of role conflicts.

IF YOU ONLY DO ONE THING

Count your user-level security records in F00950, then list your *PUBLIC records by security type. One number and one breakdown place you on the map-ready or rebuild-bound side of the line.

**PUBLIC volume is not the metric: 500 to 1,000 records is normal, and every search, select and inquiry program belongs there. F00950.FSUSER mixes users, roles and *PUBLIC, so join to F0092 where ULUGRP <> '*GROUP' to isolate user-level records.*

SHEETS 2 AND 3 19 queries covering all six tables. Pages 3-4 Oracle, pages 5-6 SQL Server.
Same 19 queries either way. Take the sheet for your database; the two sheets are otherwise identical.

NOTES

ORACLE

The Queries, in SQL

15 queries here, 4 more on page 4. Oracle syntax. Substitute your own schema for SY920.

F95921 COLUMN KEY

RLFRROLE is the first column, labelled "Member" in the data dictionary, and it holds the ROLE.
RLTOROLE is the second, labelled "Member of", and it holds the JDE USER ID. The labels read backwards; the columns do not.

F0092 Population, and how many of those users can actually sign in

```
SELECT CASE WHEN RTRIM(ULUGRP) = '*GROUP' THEN 'Roles' ELSE 'Users' END AS PROFILE_TYPE,
       COUNT(*) AS PROFILES
  FROM SY920.F0092
 GROUP BY CASE WHEN RTRIM(ULUGRP) = '*GROUP' THEN 'Roles' ELSE 'Users' END;

-- Users by enabled state.  SCEUSER is UDC 98/EN, so read the codes, do not assume them.
SELECT s.SCEUSER AS ENABLED_CODE, COUNT(*) AS USERS
  FROM SY920.F0092 u JOIN SY920.F98OWSEC s ON RTRIM(s.SCUSER) = RTRIM(u.ULUSER)
 WHERE RTRIM(u.ULUGRP) <> '*GROUP' GROUP BY s.SCEUSER;
```

F00926 Every role, its description, and its sequence number

```
-- F0092 supplies the role itself; F00926 supplies the description and the sequence.
-- F00926 can carry one row per language (AULNGP). Filter it if you see duplicates.
SELECT a.AUUSER AS ROLE, a.AUROLEDESC AS DESCRIPTION, a.AUSEQNO AS SEQ
  FROM SY920.F00926 a JOIN SY920.F0092 u ON RTRIM(u.ULUSER) = RTRIM(a.AUUSER)
 WHERE RTRIM(u.ULUGRP) = '*GROUP'
 ORDER BY a.AUSEQNO;      -- read it as bands: 1000s inquiry ... 4000s add/change/delete
```

F0093 Which roles reach which environments

```
-- LLLL is the library list, i.e. the environment. Scan for non-production roles on PD.
SELECT e.LLUSER AS ROLE, e.LLLL AS ENVIRONMENT, e.LLSEQ AS SEQ
  FROM SY920.F0093 e JOIN SY920.F0092 u ON RTRIM(u.ULUSER) = RTRIM(e.LLUSER)
 WHERE RTRIM(u.ULUGRP) = '*GROUP'
 ORDER BY e.LLUSER, e.LLSEQ;
```

F95921 Assignment volume, active against expired, and roles per user

```
-- Oracle Julian for today is written out in full below, not abbreviated.
SELECT SUM(CASE WHEN RLEXPIRDATE = 0
             OR RLEXPIRDATE >= (TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-1900)*1000 + TO_NUMBER(TO_CHAR(SYSDATE,'DDD'))
            THEN 1 ELSE 0 END) AS ACTIVE,
       SUM(CASE WHEN RLEXPIRDATE <> 0
            AND RLEXPIRDATE <  (TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-1900)*1000 + TO_NUMBER(TO_CHAR(SYSDATE,'DDD'))
            THEN 1 ELSE 0 END) AS EXPIRED
  FROM SY920.F95921;

-- Active roles per user.  RLTOROLE is the user, see the column key above.
SELECT RLTOROLE AS USER_ID, COUNT(*) AS ACTIVE_ROLES
  FROM SY920.F95921
 WHERE RLEXPIRDATE = 0
    OR RLEXPIRDATE >= (TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-1900)*1000 + TO_NUMBER(TO_CHAR(SYSDATE,'DDD'))
 GROUP BY RLTOROLE ORDER BY 2 DESC;
```

F98OWSEC Totals, lockout state, no-expiry passwords, and users with no record at all

```
SELECT COUNT(*) AS TOTAL_RECORDS FROM SY920.F98OWSEC;
SELECT SCEUSER AS ENABLED_CODE, COUNT(*) AS RECORDS FROM SY920.F98OWSEC GROUP BY SCEUSER;

-- Cannot sign in: retries have reached the limit.  SCATTEMPTS > 0 matters, see below.
SELECT COUNT(*) AS LOCKED_OUT FROM SY920.F98OWSEC WHERE SCATTEMPTS > 0 AND SCRETRY >= SCATTEMPTS;

-- Users with no signon record at all.  These are invisible to any F98OWSEC-only review.
SELECT COUNT(*) AS USERS_WITHOUT_SIGNON FROM SY920.F0092 u
 WHERE RTRIM(u.ULUGRP) <> '*GROUP'
   AND NOT EXISTS (SELECT 1 FROM SY920.F98OWSEC s WHERE RTRIM(s.SCUSER) = RTRIM(u.ULUSER));

-- SCSECFRQ = 0 means the password never has to change.  SCATTEMPTS = 0 means unlimited
-- sign-in attempts, so the account never locks itself.  Both are findings, not settings.
SELECT COUNT(*) AS PWD_NEVER_EXPIRES  FROM SY920.F98OWSEC WHERE SCSECFRQ   = 0;
SELECT COUNT(*) AS UNLIMITED_ATTEMPTS FROM SY920.F98OWSEC WHERE SCATTEMPTS = 0;
SELECT SCUSER, SCSECFRQ, SCATTEMPTS FROM SY920.F98OWSEC
 WHERE SCSECFRQ = 0 OR SCATTEMPTS = 0 ORDER BY SCUSER;   -- name them, do not just count
```

F00950 *PUBLIC composition, and the user-level records

```
-- Volume is not the metric. Composition is. Anything beyond inquiry is the question.
SELECT FSSETY, COUNT(*) AS ROW_COUNT FROM SY920.F00950 WHERE FSUSER = '*PUBLIC'
 GROUP BY FSSETY ORDER BY 2 DESC;

-- FSUSER mixes users, roles and *PUBLIC, so F0092 is what isolates the user-level rows.
SELECT COUNT(*) AS USER_LEVEL_ROWS FROM SY920.F00950 s
 WHERE EXISTS (SELECT 1 FROM SY920.F0092 u
               WHERE RTRIM(u.ULUSER) = RTRIM(s.FSUSER) AND u.ULUGRP <> '*GROUP');
```

ORACLE

Monday Morning: Two Numbers

Non-unique users, and orphaned users. Four queries here, fifteen more on page 3. Oracle syntax.

WHY THESE TWO

These are the two failures the FTC named in its October 2024 complaint against Marriott and Starwood, at paragraph 27(d): accounts of former employees not terminated, and separate unique accounts not created.

1. Non-unique users

One login, more than one human behind it.

```
-- A real person has an address book record. A profile with ULAN8 = 0 usually does not
-- have a person behind it: generic logins, service accounts, shared credentials.
SELECT COUNT(*) AS NO_ADDRESS_BOOK FROM SY920.F0092
 WHERE RTRIM(ULUGRP) <> '*GROUP' AND ULAN8 = 0;

-- One person holding several user IDs: the same address book number more than once.
SELECT ULAN8, COUNT(*) AS PROFILES FROM SY920.F0092
 WHERE RTRIM(ULUGRP) <> '*GROUP' AND ULAN8 <> 0
 GROUP BY ULAN8 HAVING COUNT(*) > 1 ORDER BY 2 DESC;
```

No address book link

Address book numbers with 2+ IDs

2. Orphaned users

A login nobody owns. Three separate counts, not one.

```
-- (a) Enabled, but holding no unexpired role. It can sign in and nobody owns it.
--     Grouped by SCEUSER so you do not have to assume which 98/EN code means enabled.
SELECT s.SCEUSER AS ENABLED_CODE, COUNT(*) AS NO_ACTIVE_ROLE
  FROM SY920.F0092 u JOIN SY920.F98OWSEC s ON RTRIM(s.SCUSER) = RTRIM(u.ULUSER)
 WHERE RTRIM(u.ULUGRP) <> '*GROUP'
   AND NOT EXISTS (SELECT 1 FROM SY920.F95921 r
                   WHERE RTRIM(r.RLTOROLE) = RTRIM(u.ULUSER)
                     AND (r.RLEXPIRDATE = 0
                      OR r.RLEXPIRDATE >= (TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-1900)*1000
                                        + TO_NUMBER(TO_CHAR(SYSDATE,'DDD'))))
 GROUP BY s.SCEUSER;

-- (c) No signon record at all. Invisible to any review that starts from F98OWSEC.
SELECT COUNT(*) AS NO_SIGNON_RECORD FROM SY920.F0092 u
 WHERE RTRIM(u.ULUGRP) <> '*GROUP'
   AND NOT EXISTS (SELECT 1 FROM SY920.F98OWSEC s WHERE RTRIM(s.SCUSER) = RTRIM(u.ULUSER));
```

(a) Enabled, no active role

(b) No sign-in 180+ days

(c) No F98OWSEC record

(b) NO SIGN-IN FOR 180+ DAYS IS NOT IN THESE SIX TABLES

Last sign-in lives in the sign-on audit table F9312, and only if sign-on security history is switched on. If it is off, the count is not zero, it is unknowable, and turning history on is the first fix. No query is printed here because it would be guesswork until the F9312 layout is confirmed for your release.

THESE ARE ROWS 4 AND 8 ON THE FRONT

Role-level rather than user-level security, and recent orphan cleanup. If either number surprises you, that is your answer.

NOTES

The Queries, in SQL

15 queries here, 4 more on page 6. Microsoft SQL Server syntax. Substitute your own schema for SY920.
Run `USE JDE920` · `GO` · first. Each block below is its own batch. Substitute your own database name for JDE920.

F95921 COLUMN KEY

RLFRROLE is the first column, labelled "Member" in the data dictionary, and it holds the ROLE.
RLTOROLE is the second, labelled "Member of", and it holds the JDE USER ID. The labels read backwards; the columns do not.

F0092 — Population, and how many of those users can actually sign in

```
SELECT CASE WHEN RTRIM(ULUGRP) = '*GROUP' THEN 'Roles' ELSE 'Users' END AS PROFILE_TYPE,
       COUNT(*) AS PROFILES
  FROM SY920.F0092
 GROUP BY CASE WHEN RTRIM(ULUGRP) = '*GROUP' THEN 'Roles' ELSE 'Users' END;

-- Users by enabled state.  SCEUSER is UDC 98/EN, so read the codes, do not assume them.
SELECT s.SCEUSER AS ENABLED_CODE, COUNT(*) AS USERS
  FROM SY920.F0092 u JOIN SY920.F98OWSEC s ON RTRIM(s.SCUSER) = RTRIM(u.ULUSER)
 WHERE RTRIM(u.ULUGRP) <> '*GROUP' GROUP BY s.SCEUSER;
```

F00926 — Every role, its description, and its sequence number

```
-- F0092 supplies the role itself; F00926 supplies the description and the sequence.
-- F00926 can carry one row per language (AULNGP). Filter it if you see duplicates.
SELECT a.AUUSER AS ROLE, a.AUROLEDESC AS DESCRIPTION, a.AUSEQNO AS SEQ
  FROM SY920.F00926 a JOIN SY920.F0092 u ON RTRIM(u.ULUSER) = RTRIM(a.AUUSER)
 WHERE RTRIM(u.ULUGRP) = '*GROUP'
 ORDER BY a.AUSEQNO;      -- read it as bands: 1000s inquiry ... 4000s add/change/delete
```

F0093 — Which roles reach which environments

```
-- LLLL is the library list, i.e. the environment. Scan for non-production roles on PD.
SELECT e.LLUSER AS ROLE, e.LLLL AS ENVIRONMENT, e.LLSEQ AS SEQ
  FROM SY920.F0093 e JOIN SY920.F0092 u ON RTRIM(u.ULUSER) = RTRIM(e.LLUSER)
 WHERE RTRIM(u.ULUGRP) = '*GROUP'
 ORDER BY e.LLUSER, e.LLSEQ;
```

F95921 — Assignment volume, active against expired, and roles per user

```
-- SQL Server Julian for today is written out in full below, not abbreviated.
SELECT SUM(CASE WHEN RLEXPIRDATE = 0
             OR RLEXPIRDATE >= (YEAR(GETDATE())-1900)*1000 + DATEPART(DY, GETDATE())
            THEN 1 ELSE 0 END) AS ACTIVE,
       SUM(CASE WHEN RLEXPIRDATE <> 0
            AND RLEXPIRDATE <  (YEAR(GETDATE())-1900)*1000 + DATEPART(DY, GETDATE())
            THEN 1 ELSE 0 END) AS EXPIRED
  FROM SY920.F95921;

-- Active roles per user.  RLTOROLE is the user, see the column key above.
SELECT RLTOROLE AS USER_ID, COUNT(*) AS ACTIVE_ROLES
  FROM SY920.F95921
 WHERE RLEXPIRDATE = 0
    OR RLEXPIRDATE >= (YEAR(GETDATE())-1900)*1000 + DATEPART(DY, GETDATE())
 GROUP BY RLTOROLE ORDER BY 2 DESC;
```

F98OWSEC — Totals, lockout state, no-expiry passwords, and users with no record at all

```
SELECT COUNT(*) AS TOTAL_RECORDS FROM SY920.F98OWSEC;
SELECT SCEUSER AS ENABLED_CODE, COUNT(*) AS RECORDS FROM SY920.F98OWSEC GROUP BY SCEUSER;

-- Cannot sign in: retries have reached the limit.  SCATTEMPTS > 0 matters, see below.
SELECT COUNT(*) AS LOCKED_OUT FROM SY920.F98OWSEC WHERE SCATTEMPTS > 0 AND SCRETRY >= SCATTEMPTS;

-- Users with no signon record at all.  These are invisible to any F98OWSEC-only review.
SELECT COUNT(*) AS USERS_WITHOUT_SIGNON FROM SY920.F0092 u
 WHERE RTRIM(u.ULUGRP) <> '*GROUP'
   AND NOT EXISTS (SELECT 1 FROM SY920.F98OWSEC s WHERE RTRIM(s.SCUSER) = RTRIM(u.ULUSER));

-- SCSECFRQ = 0 means the password never has to change.  SCATTEMPTS = 0 means unlimited
-- sign-in attempts, so the account never locks itself.  Both are findings, not settings.
SELECT COUNT(*) AS PWD_NEVER_EXPIRES  FROM SY920.F98OWSEC WHERE SCSECFRQ   = 0;
SELECT COUNT(*) AS UNLIMITED_ATTEMPTS FROM SY920.F98OWSEC WHERE SCATTEMPTS = 0;
SELECT SCUSER, SCSECFRQ, SCATTEMPTS FROM SY920.F98OWSEC
 WHERE SCSECFRQ = 0 OR SCATTEMPTS = 0 ORDER BY SCUSER;   -- name them, do not just count
```

F00950 — *PUBLIC composition, and the user-level records

```
-- Volume is not the metric. Composition is. Anything beyond inquiry is the question.
SELECT FSSETY, COUNT(*) AS ROW_COUNT FROM SY920.F00950 WHERE FSUSER = '*PUBLIC'
 GROUP BY FSSETY ORDER BY 2 DESC;

-- FSUSER mixes users, roles and *PUBLIC, so F0092 is what isolates the user-level rows.
SELECT COUNT(*) AS USER_LEVEL_ROWS FROM SY920.F00950 s
 WHERE EXISTS (SELECT 1 FROM SY920.F0092 u
               WHERE RTRIM(u.ULUSER) = RTRIM(s.FSUSER) AND u.ULUGRP <> '*GROUP');
```

Monday Morning: Two Numbers

Non-unique users, and orphaned users. Four queries here, fifteen more on page 5. Microsoft SQL Server syntax.

WHY THESE TWO

These are the two failures the FTC named in its October 2024 complaint against Marriott and Starwood, at paragraph 27(d): accounts of former employees not terminated, and separate unique accounts not created.

1. Non-unique users

One login, more than one human behind it.

```
-- A real person has an address book record. A profile with ULAN8 = 0 usually does not
-- have a person behind it: generic logins, service accounts, shared credentials.
SELECT COUNT(*) AS NO_ADDRESS_BOOK FROM SY920.F0092
 WHERE RTRIM(ULUGRP) <> '*GROUP' AND ULAN8 = 0;

-- One person holding several user IDs: the same address book number more than once.
SELECT ULAN8, COUNT(*) AS PROFILES FROM SY920.F0092
 WHERE RTRIM(ULUGRP) <> '*GROUP' AND ULAN8 <> 0
 GROUP BY ULAN8 HAVING COUNT(*) > 1 ORDER BY 2 DESC;
```

No address book link

Address book numbers with 2+ IDs

2. Orphaned users

A login nobody owns. Three separate counts, not one.

```
-- (a) Enabled, but holding no unexpired role. It can sign in and nobody owns it.
--     Grouped by SCEUSER so you do not have to assume which 98/EN code means enabled.
SELECT s.SCEUSER AS ENABLED_CODE, COUNT(*) AS NO_ACTIVE_ROLE
  FROM SY920.F0092 u JOIN SY920.F98OWSEC s ON RTRIM(s.SCUSER) = RTRIM(u.ULUSER)
 WHERE RTRIM(u.ULUGRP) <> '*GROUP'
   AND NOT EXISTS (SELECT 1 FROM SY920.F95921 r
                    WHERE RTRIM(r.RLTOROLE) = RTRIM(u.ULUSER)
                      AND (r.RLEXPIRDATE = 0
                       OR r.RLEXPIRDATE >= (YEAR(GETDATE())-1900)*1000 + DATEPART(DY, GETDATE())))
 GROUP BY s.SCEUSER;

-- (c) No signon record at all. Invisible to any review that starts from F98OWSEC.
SELECT COUNT(*) AS NO_SIGNON_RECORD FROM SY920.F0092 u
 WHERE RTRIM(u.ULUGRP) <> '*GROUP'
   AND NOT EXISTS (SELECT 1 FROM SY920.F98OWSEC s WHERE RTRIM(s.SCUSER) = RTRIM(u.ULUSER));
```

(a) Enabled, no active role

(b) No sign-in 180+ days

(c) No F98OWSEC record

(b) NO SIGN-IN FOR 180+ DAYS IS NOT IN THESE SIX TABLES

Last sign-in lives in the sign-on audit table F9312, and only if sign-on security history is switched on. If it is off, the count is not zero, it is unknowable, and turning history on is the first fix. No query is printed here because it would be guesswork until the F9312 layout is confirmed for your release.

THESE ARE ROWS 4 AND 8 ON THE FRONT

Role-level rather than user-level security, and recent orphan cleanup. If either number surprises you, that is your answer.

NOTES