Migrazione di un database SAP ASE locale su Amazon RDS for SQL Server

# AWS Linee guida prescrittive

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Linee guida prescrittive: Migrazione di un database SAP ASE locale su Amazon RDS for SQL Server

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

# Table of Contents

Introduzione ..... 1
Destinatari principali ..... 1
Obiettivi ..... 1
Migrazione ..... 2
Fasi di migrazione ..... 2
Fase 1: Preparazione ..... 2
Valuta il database di origine ..... 2
Valuta i problemi di sicurezza ..... 3
Consultazione AWS Partner ..... 3
Fase 2: Pianificazione ..... 3
Opzione di migrazione per Amazon RDS for SQL Server ..... 4
Fase 3: migrazione ..... 5
Conversione dei dati ..... 5
Convalida degli oggetti del database ..... 6
Migrazione dei dati utilizzando AWS DMS ..... 6
Migrazione dei dati offline ..... 7
Utilizzo di strumenti di terze parti ..... 7
Monitoraggio del database ..... 7
Convalida dei dati ..... 8
Test della migrazione ..... 10
Fase 4: Funzionamento e ottimizzazione ..... 11
Resources ..... 12
AWS documentazione ..... 12
Servizi AWS ..... 12
Post del blog ..... 12
Sybase ASE e SQL Server ..... 13
Cronologia dei documenti ..... 14
..... xv

# Migrazione di un database SAP ASE locale su Amazon RDS for SQL Server

Amit Kumar, Jitendra Kumar e Nishad Mankar, Amazon Web Services

Questa guida descrive le opzioni per la migrazione di un database SAP ASE ad Amazon Relational Database Service (Amazon RDS) per SQL Server. Amazon RDS for SQL Server è una scelta di migrazione comune per i carichi di lavoro SAP ASE perché supporta i comandi e le funzionalità Transact-SQL (T-SQL). Supporta anche Tabular Data Stream (TDS), che è il protocollo di comunicazione client-server nativo per SQL Server.

Organizations spesso migrano da SAP Adaptive Server Enterprise (SAP ASE) ad Amazon RDS for SQL Server per ridurre i costi di licenza e automatizzare la gestione dei database utilizzando. Servizi AWS Amazon RDS automatizza le attività di amministrazione come backup, patch software e funzionalità ad alta disponibilità, mantenendo al contempo prestazioni e sicurezza.

## Destinatari principali

Questa guida è destinata agli amministratori di database, agli ingegneri di database e ai tecnici della migrazione che intendono migrare un database SAP ASE verso. Cloud AWS Questa guida presuppone che l'utente conosca i seguenti servizi e concetti:

- Amazon RDS for SQL Server e tecnologie di database correlate
- AWS Database Migration Service (AWS DMS) caratteristiche, limitazioni e requisiti di sicurezza
- Utilizzo di SAP ASE come origine per AWS DMS, comprese le limitazioni, con SQL Server come destinazione
- Configurazione di rete per la AWS DMS replica
- Strumenti di conversione dello schema per SAP ASE e SQL Server

## Obiettivi

Questa guida descrive come migrare le applicazioni eseguite su database SAP ASE verso. AWS Scopri gli approcci di migrazione, le fasi di implementazione e i passaggi specifici per la ripiattaforma e il refactoring delle tue applicazioni. Scegliete una strategia di migrazione in linea con i requisiti aziendali e l'ambiente tecnico per ridurre i tempi di inattività e ottimizzare i costi.

# Migrazione

Quando scegli una strategia di migrazione, considera i requisiti aziendali, le risorse, il tempo e le finanze.

## Fasi di migrazione

Per fornire un unico framework per la migrazione di vari carichi di lavoro, come l'elaborazione delle transazioni online (OLTP) e l'elaborazione analitica online (OLAP), questa guida è organizzata in quattro fasi di migrazione che supportano la maggior parte delle applicazioni. Di seguito sono riportate le descrizioni delle fasi di migrazione più comuni.

- Fase 1: preparazione: considera i carichi di lavoro del database per determinare se la migrazione del database è omogenea o eterogenea.
- Fase 2: pianificazione: valuta i dati per determinare una strategia di migrazione adeguata, ad esempio rehosting, ripiattaforma, riacquisto, rifattorizzazione, conservazione o ritiro.
- ???— Dopo aver scelto una categoria e una strategia di migrazione del database, create il database di destinazione e migrate il database di origine utilizzando la convalida e la replica dei dati.
- ???— Massimizza i vantaggi dell'hosting delle applicazioni, tra cui la riduzione dei costi AWS, delle prestazioni, della sicurezza e della resilienza delle applicazioni.

## Fase 1: Preparazione

Valuta l'architettura del database SAP ASE di origine per comprenderne le dipendenze e lo stato attuale.

### Valuta il database di origine

Per valutare il database di origine, ad esempio oggetti, utenti e autorizzazioni, puoi utilizzare strumenti di terze parti. Per ulteriori informazioni, consulta Preparazione e valutazione per la migrazione da SAP ASE ad Amazon Aurora MySQL.

- Analizza la conversione del codice elencando tutte le dipendenze e le interfacce dell'applicazione.
- Identifica le dipendenze delle applicazioni SAP supportate solo in SAP. Potrebbe essere necessario trovare un metodo alternativo per lavorare con Microsoft SQL.

- Identifica i protocolli di comunicazione.
- Identifica i requisiti di replica e l'elevata conformità al availability/disaster ripristino.
- Identifica i protocolli di sicurezza, conformità e crittografia.

Puoi utilizzare qualsiasi strumento di migrazione di terze parti per convertire il codice dello schema, come il database SAP ASE o Amazon RDS for SQL Server. È necessario convertire del codice manualmente, ad esempio SQL non ANSI, perché SQL Server potrebbe non supportarlo. Dopo aver convertito il codice, eseguite test unitari e funzionali per convalidare gli oggetti del database. Per ulteriori informazioni, consulta Migrare il database SAP ASE (Sybase ASE) su Amazon RDS for SQL Server.

## Valuta i problemi di sicurezza

SQL Server offre diverse opzioni di sicurezza basate su requisiti di conformità aziendali e casi d'uso unici. Per ulteriori informazioni, consulta i seguenti documenti sulla sicurezza di SQL Server:

- Autentica Microsoft SQL Server su Amazon EC2 utilizzando AWS Directory Service
- Abilita la crittografia trasparente dei dati in Amazon RDS for SQL Server

## Consultazione AWS Partner

Per ridurre i tempi di migrazione, prendi in considerazione la possibilità di consultare AWS Database Migration Service i partner. AWS I partner dispongono delle competenze necessarie per supportare migrazioni omogenee ed eterogenee tra diverse piattaforme di database, come SAP ASE verso Amazon Aurora MySQL o Microsoft SQL Server verso MySQL.

In base ai requisiti e alle preferenze dell'utente, i partner possono fornire strumenti e consigli su come gestire le migrazioni o risolvere gli aspetti relativi alla migrazione. Assicurati di seguire le AWS migliori pratiche per ospitare il tuo database in. Cloud AWS Per ulteriori informazioni, consulta Best practice for Amazon RDS.

# Fase 2: Pianificazione

In base al carico di lavoro e all'applicazione, scegli RDS personalizzato per SQL Server o RDS per Microsoft SQL Server. Per la maggior parte delle migrazioni di database, puoi eseguire il rehosting,

la ripiattaforma o il refactoring. Il principio guida dovrebbe essere come massimizzare i vantaggi della migrazione.

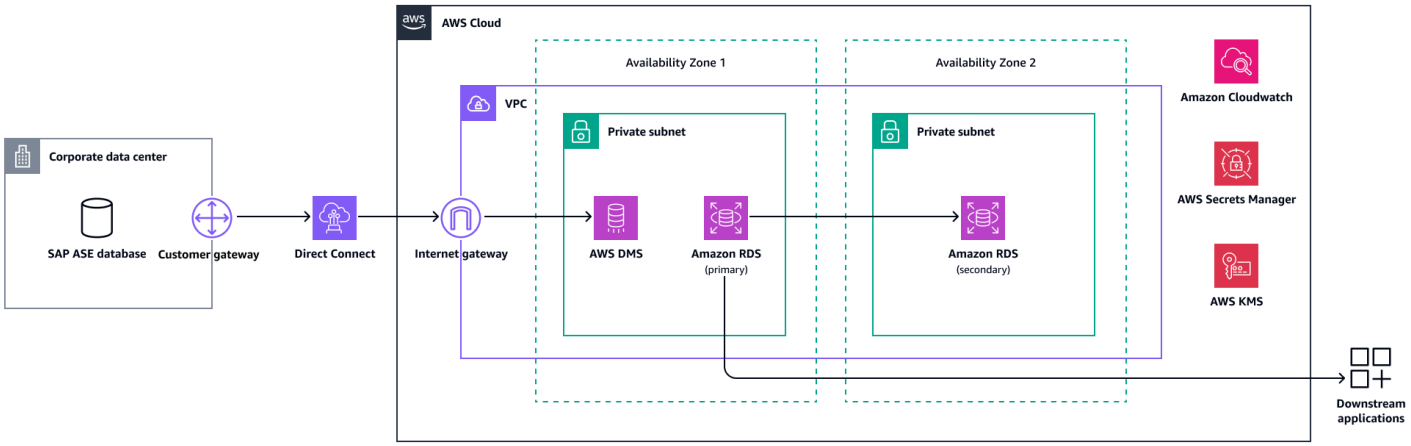
La scelta di rifattorizzare l'applicazione e migrare a un database nativo del cloud come Aurora può migliorare l'applicazione di database. A seconda della complessità del carico di lavoro, tuttavia, il refactoring di un database può richiedere molto tempo e risorse.

## Opzione di migrazione per Amazon RDS for SQL Server

Valuta se il database di origine è già configurato per sincronizzare i server di database primari e secondari, quindi decidi come sincronizzare i server. AWS Di seguito sono riportati tre approcci:

- Zone di disponibilità multiple
- Replica regionale
- Cross-Regionleggi le repliche

Nel seguente diagramma di architettura, AWS Direct Connect stabilisce una connessione di rete privata dedicata tra il data center locale e Amazon VPC. L'ambiente di database di destinazione ha un Amazon RDS for SQL Server primario distribuito in una zona di disponibilità, con una replica di lettura in una seconda zona di disponibilità per l'alta disponibilità e il disaster recovery. AWS Database Migration Service (AWS DMS) gestisce il processo di migrazione per ridurre al minimo i tempi di inattività.




Per una maggiore sicurezza, l'ambiente di destinazione utilizza sottoreti private. AWS Secrets Manager gestisce le credenziali dell'utente e AWS Key Management Service (AWS KMS) fornisce la crittografia inattiva per i dati sensibili. L'intera infrastruttura è monitorata da Amazon CloudWatch, che tiene traccia delle metriche delle prestazioni e abilita avvisi proattivi. Questo design aiuta a

ottimizzare le prestazioni risolvendo al contempo eventuali problemi che si presentano durante il processo di migrazione.

# Fase 3: migrazione

Un'attività principale delle migrazioni dei database consiste nel completare la migrazione con tempi di inattività minimi. Poiché entrambi i database devono utilizzare lo stesso linguaggio di programmazione e gli stessi protocolli di comunicazione, potrebbe essere necessario convertire il codice e lo schema per sintassi, procedure e funzioni di query simili. Durante la conversione di uno schema, considerate i seguenti aspetti:

- Modificate la connessione al database in base alle esigenze del nuovo motore.
- Correggi eventuali avvisi ed errori derivanti dalla conversione del codice.
- Modifica le mappature e il codice delle tabelle in base allo schema convertito.
- Identifica e rifattorizza qualsiasi funzionalità specifica del fornitore utilizzata dall'applicazione.

Puoi utilizzare qualsiasi strumento di migrazione di terze parti per la conversione del codice dello schema, come il database SAP ASE e Amazon RDS for SQL Server. Potrebbe essere necessario convertire del codice manualmente perché SQL non ANSI non è supportato in SQL Server.

Dopo aver convertito il codice, converti il codice dell'applicazione o l'applicazione SQL dinamico, quindi esegui test unitari e funzionali. Per ulteriori informazioni, vedere Testing Migrated Database Objects (SybaseToSQL).

## Conversione dei dati

Trasforma i dati grezzi per renderli più utili pulendoli, standardizzandoli, verificandoli e ordinandoli. Nelle migrazioni dei database, i processi di estrazione, trasformazione e caricamento (ETL) vengono utilizzati nei seguenti modi:

- All'interno del database
- Con script esterni
- Utilizzo di strumenti di terze parti

Alcuni esempi di strumenti ETL includono AWS Glue Informatica e Talend. Per le migrazioni da SAP ASE a SQL Server, alcuni strumenti gratuiti possono convertire automaticamente le procedure e le funzioni archiviate.

## Convalida degli oggetti del database

La convalida del database aiuta a prevenire problemi nelle fasi successive della migrazione. Dopo la conversione del codice, convalida lo schema del database confrontando i seguenti elementi tra SAP ASE e RDS SQL Server:

- Schemi
- Tabelle
- Visualizzazioni
- Funzioni
- Indici delle procedure memorizzate
- Triggers
- Vincoli (ad esempio, chiavi primarie, chiavi esterne, controlli e impostazioni predefinite)

Verificate che ogni oggetto sia stato migrato correttamente. Se trovi delle differenze, identifica il motivo dell'errore. Potrebbe essere necessario creare manualmente gli oggetti mancanti nel database di destinazione o convertire il Transact-SQL codice. Per ulteriori informazioni, consulta Convalida degli oggetti del database dopo la migrazione da SAP ASE ad Amazon RDS for SQL Server o Microsoft SQL Server.

## Migrazione dei dati utilizzando AWS DMS

Se hai più utenti del database, potrebbe essere necessario migrare l'applicazione in base a una pianificazione. A seconda delle dimensioni del database e della finestra di migrazione, tali migrazioni di dati richiedono la conoscenza dei carichi completi e incrementali. Per questo motivo, AWS DMS può connettere i database di origine e di destinazione per replicare il contenuto del database secondo i seguenti processi:

- Creazione di un server di replica.
- Crea endpoint di origine e destinazione che descrivono le connessioni ai data store.
- Crea una o più attività di migrazione per migrare i dati tra gli archivi dati di origine e di destinazione.

- Replica continua da SAP ASE a SQL Server
- (Facoltativo) Migrazione completa dei dati da SAP ASE a SQL Server con acquisizione dei dati di modifica

Potrebbe essere necessario ottimizzare la gestione AWS DMS di determinati tipi di dati. Per ulteriori informazioni, vedere Utilizzo di un database SAP ASE come fonte per AWS DMS.

## Migrazione dei dati offline

Puoi utilizzarlo Gateway di archiviazione AWS per integrare il tuo database SAP ASE con Amazon Simple Storage Service (Amazon S3), che fornisce uno storage conveniente, scalabile e sicuro per i backup di database SAP ASE locali. Per ulteriori informazioni, consulta Integrare un database SAP ASE con Amazon Gateway di archiviazione AWS S3 utilizzando.

## Utilizzo di strumenti di terze parti

Alcune applicazioni fungono da punto di contatto unico (SPOC) che si interfaccia con altre applicazioni. Durante la migrazione a una piattaforma di database SQL Server, queste interconnessioni potrebbero risentirne e il monitoraggio del database potrebbe richiedere strumenti nativi o di terze parti che utilizzano protocolli di comunicazione specifici del server. È importante valutare se queste applicazioni e strumenti dipendenti supportano già SQL Server o se necessitano di modifiche per funzionare correttamente.

Per le applicazioni in pacchetto, consulta i fornitori per determinare se supportano Amazon RDS for SQL Server. Per le applicazioni personalizzate, potrebbe essere necessario modificare il codice per garantire la compatibilità con il database migrato.

## Monitoraggio del database

Indipendentemente dal percorso di migrazione scelto, Amazon CloudWatch svolge un ruolo nella raccolta di parametri, come il tipo di CPU, la memoria e I/O le funzioni. È anche in grado di impostare soglie di metriche e avviare azioni quando vengono attivate le soglie.

Ad esempio, puoi creare allarmi per parametri, notifiche e azioni del cluster Amazon RDS per rilevare e chiudere le istanze di lettura inutilizzate o sottoutilizzate. L'impostazione di allarmi su parametri ed eventi può aiutare a ridurre al minimo i tempi di inattività e gli impatti aziendali. Servizi AWS come Amazon S3, Amazon RDS Performance Insights, Amazon CloudWatch, e AWS CloudTrail sono già integrati con la piattaforma di database RDS e li consigliamo per monitorare le prestazioni.

# Convalida dei dati

Una volta completata la migrazione dei dati da SAP ASE ad Amazon RDS for SQL Server, convalida i dati per garantire precisione e coerenza. Utilizza le seguenti query SQL per generare istruzioni di metadati per ogni tabella del database.

## Passaggio 1: generare istruzioni di metadati ed elenchi di colonne

```
SELECT dt.schema_name, dt.table_name,
    STRING_AGG(dt.column_name, ',') AS column_name,
    STRING_AGG(dt.cname, ',') AS column_order
FROM (
    SELECT
        object_name(a.id) AS table_name,
        a.name colname,
        c.name col_type,
        a.isnullable,
        a.name AS cname,
        schema_name(b.uid) AS schema_name,
        CASE
            WHEN a.isnullable = 1 THEN
                CASE
                    WHEN c.name LIKE '%char%'
                        THEN 'coalesce(ltrim(rtrim('+a.name+')),''X'') as '+a.name
                    WHEN (c.name LIKE '%int%' OR c.name = 'numeric')
                        THEN 'coalesce('+a.name+',0) as '+a.name
                    WHEN c.name IN ('decimal','float','money')
                        THEN 'coalesce('+a.name+',0.0) as '+a.name
                    WHEN c.name LIKE 'datetime%'
                        THEN 'coalesce(convert(nvarchar(30),'+a.name
+',112),''99991231'') as '+a.name
                    ELSE a.name
                END
            WHEN c.name LIKE 'datetime%'
                THEN 'coalesce(convert(nvarchar(30),'+a.name+',112),''99991231'') as
 '+a.name
            WHEN c.name LIKE '%char%'
                THEN 'coalesce(ltrim(rtrim('+a.name+')),''X'') as '+a.name
            ELSE a.name
        END AS column_name
    FROM syscolumns a
    INNER JOIN sysobjects b
```

```
        ON a.id = b.id
        AND b.type = 'U'
    INNER JOIN systypes c
        ON a.usertype = c.usertype
        AND a.xusertype = c.xusertype
        AND c.name != 'varbinary'
    INNER JOIN (
        SELECT
            OBJECT_NAME(ic.OBJECT_ID) AS table_name,
            COL_NAME(ic.OBJECT_ID, ic.column_id) AS column_name
        FROM sys.indexes AS i
        INNER JOIN sys.index_columns AS ic
            ON i.OBJECT_ID = ic.OBJECT_ID
            AND i.index_id = ic.index_id
            AND i.is_primary_key = 1
    ) pk
        ON pk.table_name = object_name(a.id)
        AND pk.column_name = a.name
) dt
GROUP BY dt.schema_name, dt.table_name;
```

La tabella seguente elenca esempi di output:

| Nome dello schema | Nome della tabella | Nome della colonna | Ordine delle colonne |
|---|---|---|---|
| Person | Indirizzo | ID dell'indirizzo | ID indirizzo |
| Person | AddressType | AddressTypeID | AddressTypeID |
| Person | BusinessEntity | BusinessEntityID | BusinessEntityID |
| Person | BusinessEntityAddress | BusinessEntityID, ID indirizzo, ID AddressType | BusinessEntityID, ID indirizzo, ID AddressType |

## Fase 2: Generazione di query di confronto utilizzando i risultati dei metadati e creazione di istruzioni SELECT

```
SELECT <column_name>
FROM [schema_name].[table_name]
```

```
ORDER BY <column_order>;
```

Di seguito è riportato un esempio di query generata:

```
SELECT BusinessEntityID, AddressID, AddressTypeID
FROM [Person].[BusinessEntityAddress]
ORDER BY BusinessEntityID, AddressID, AddressTypeID;
```

### Fase 3: Convalida

1. Esegui la query sui metadati in entrambi i database.
2. Genera ed esegui `SELECT` istruzioni per ogni tabella.
3. Confronta i risultati tra i database di origine e di destinazione:
   - Il numero di righe deve corrispondere.
   - I valori dei dati devono essere identici.
   - Verifica la presenza di problemi di conversione dei tipi di dati.

### Fase 4: Convalida del conteggio delle righe

```
SELECT COUNT(1) AS total_rows
FROM [schema_name].[table_name];
```

### Passaggio 5: aggiorna la configurazione dell'applicazione in modo che punti al nuovo database

1. Aggiorna il gruppo di sicurezza.
2. Modifica la stringa di connessione DNS in base alle esigenze per connetterti al database di destinazione.

## Test della migrazione

Il processo di test può aiutarti a identificare i problemi trascurati durante lo sviluppo, come le query convertite in modo errato o gli indici mancanti. Inoltre, potrebbe rivelare la necessità di ottimizzare il motore di database o modificare le query in base alle prestazioni del carico di lavoro.

I test funzionali, che includono test unitari per i flussi di lavoro delle applicazioni, aiutano a garantire una perfetta integrazione con il nuovo database. I test delle prestazioni aiutano a ottimizzare il database verificando i tempi di risposta accettabili e identificando i punti deboli.

Sebbene esistano metodi di test manuali e automatizzati, consigliamo un metodo automatizzato perché è più efficiente, soprattutto per cicli di test aggiuntivi.

## Fase 4: Funzionamento e ottimizzazione

Ottimizzate il database valutando lo stack di applicazioni in termini di costi, prestazioni, sicurezza e resilienza. Ad esempio, utilizzate le funzionalità di scalabilità automatica per aggiungere altre repliche di lettura durante le ore di punta e quindi rimuoverle nelle ore non di punta quando i costi sono inferiori.

Puoi anche Servizi AWS utilizzarlo per integrarlo con i database Amazon RDS, ad esempio indirizzando i log del tuo motore di database ad Amazon CloudWatch Logs per l'analisi, e utilizzare Performance Insights per aiutarti ad analizzare e visualizzare il tuo database. Per ulteriori informazioni, consulta Best practice per la configurazione dei parametri prestazionali per Amazon RDS for SQL Server.

# Resources

## AWS documentazione

- Autentica Microsoft SQL Server su Amazon EC2 utilizzando AWS Directory Service
- Abilita la crittografia trasparente dei dati in Amazon RDS for SQL Server
- Preparazione e valutazione per la migrazione da SAP ASE ad Amazon Aurora MySQL
- Configurazione di una rete per un'istanza di replica
- Utilizzo di un database Microsoft SQL Server come destinazione per AWS Database Migration Service

## Servizi AWS

- Amazon CloudWatch
- Amazon RDS per SQL Server
- Approfondimenti sulle prestazioni di Amazon RDS
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Storage Service (Amazon S3)
- AWS CloudTrail
- AWS Database Migration Service (AWS DMS)
- Gateway di archiviazione AWS

## Post del blog

- Best practice per la configurazione dei parametri prestazionali per Amazon RDS for SQL Server
- Integra un database SAP ASE in Amazon S3 utilizzando Gateway di archiviazione AWS
- Esegui la migrazione del tuo database SAP ASE (Sybase ASE) su Amazon RDS for SQL Server
- Convalida gli oggetti del database dopo la migrazione da SAP ASE ad Amazon RDS for SQL Server o Microsoft SQL Server

# Sybase ASE e SQL Server

- Le migliori pratiche per lavorare con SQL Server
- SAP Adaptive Server Enterprise (SAP ASE)
- Transact-SQL
- Flusso di dati tabulari (TDS)

# Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

| Modifica | Descrizione | Data |
| --- | --- | --- |
| Pubblicazione iniziale | — | 16 luglio 2025 |

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.