

IT 資格道場

DP-700 学習用テキスト

予備知識ゼロから合格ラインへ——読んで理解する1冊目

無料サンプル(第1章まで)

DP-700 出題範囲(Skills measured)2026-10-19版 準拠
2026年10月版

ご利用にあたって

- このテキストは「IT資格道場」(www.shikaku-doujou.com)の購入者限定テキストのうち、学習用テキストの第1章までを収録した**無料サンプル**です。どなたでもご覧いただけます。
- 無料サンプルも著作物です。転載・改変しての再配布・二次販売はできません。
- AI モデルの学習データとしてのご利用(学習目的の入力・データセット化を含む)はできません。
- 製品版(学習用テキスト・頻出度別ノート・暗記用ノートの3点)は、ご購入者を識別できる透かし入りのPDFとして提供されます。
- 本PDFはスマートフォン・タブレットでの閲覧に合わせたA5版面・文字サイズで組版しています。

目次

この教材の使い方

試験の基本情報

試験の設計図:3 分野と配点

構成

4 ステップ学習法

第1章 Implement and manage an analytics solution [1.1/1.2/1.3/1.4]

Fabric ワークスペース設定の構成 (Configure Microsoft Fabric workspace settings)
[1.1]

Fabric のライフサイクル管理の実装 (Implement lifecycle management in Fabric)
[1.2]

セキュリティとガバナンスの構成 (Configure security and governance) [1.3]

処理のオーケストレーション (Orchestrate processes) [1.4]

サンプルはここまでです

3冊の使い方は、ダウンロード欄の案内を参照してください。

SAMPLE

この教材の使い方

このテキストは、Microsoft の Associate 資格 **Microsoft Certified: Fabric Data Engineer Associate(試験コード DP-700、試験名 Implementing Data Engineering Solutions Using Microsoft Fabric)** に合格することを目的に作られました。DP-700 は、Microsoft Fabric 上でデータの取り込み・変換・分析の基盤を**データ エンジニアの視点で作って運用する**資格。製品の名前を知っているかではなく、「この要件なら、どの項目・どの設定・どの操作を選ぶか」を問われます。

Fabric には、Lakehouse、Warehouse、Eventhouse、パイプライン、Dataflow Gen2、ノートブックなど、似た役割の項目がいくつもあります。同じ「データを入れる」「権限を絞る」「遅さを直す」でも、項目によって使える機能と手順が違います。このテキストは、その違いを軸に、公式の出題範囲(3 分野)に沿って説明していきます。

公式の想定読者は、データの取り込み・変換・分析のソリューションを Fabric で設計して実装し、運用する人。

とはいえ、深い開発経験は要りません。**各項目の「何をする道具か」と「選ぶ条件・設定の勘どころ」を押さえれば合格ラインに届きます**。SQL・PySpark・KQL のコード例は出てきますが、**丸暗記は不要**で、「どのコードが何をするか」の対応が読めれば十分でしょう。

試験の基本情報

項目	内容
試験名	Microsoft Certified: Fabric Data Engineer Associate(DP-700)
出題数	Microsoft は公式に固定公表していません (Microsoft の認定試験は一般に 40～60 問程度とされます。このサイトのコースは 1 回 50 問と推定して作っています)
出題形式	択一・複数選択のほか、ドラッグ アンド ドロップ、ホットエリア、ケーススタディ、「問題と解決策」(同じ状況に対して解決策だけを差し替え、それぞれに「はい」「いいえ」で答える形式)が含まれることがあります。試験ごとの形式は事前に公表されません
試験時間	100 分(公式認定ページ記載)。ラボを含まない区分の試験時間です
合格ライン	1,000 点満点中 700 点 (スケールスコア)
受験料	受験する国・地域により異なります(日本での価格・税の扱いは申込時に確認してください)
受験方法	ピアソン VUE。テストセンターまたはオンライン監督試験
言語	日本語で受験可(英語ほか多言語提供)
資格の有効期限	取得後 12 か月(Microsoft の無料オンライン評価で更新)

分からない問題も、必ずどれかを選んで埋めてください。受験料や実施形式は、いつまでも同じとはかぎりません。申し込み時に、公式ページの最新情報

を必ず確認してください。

試験の設計図:3 分野と配点

このテキストが準拠する Skills measured は **2026 年 10 月 19 日版**で、前の版(2026 年 7 月 21 日版)との差は表記の 2 行だけ。2.3 の「Eventstreams」が「Eventstream」に、3.3 の「Eventstreams and Eventhouses」が「Eventstream and Eventhouse」に変わりました。

分野	配点	一言でいうと
① 分析ソリューションの実装と管理	30～35%	整える・守る・回す —— ワークスペース設定・バージョン管理・アクセス制御・オーケストレーション
② データの取り込みと変換	30～35%	入れる・変える —— 読み込みパターン・バッチ・ストリーミング
③ 分析ソリューションの監視と最適化	30～35%	見張って直す —— 監視・エラーの解決・性能の最適化

3 分野とも 30～35% で、配点の差はありません。どれか 1 分野を捨てる作戦は取れません。

構成

このテキストは 3 分野を 3 つの章に分ける構成。並べ方は**公式の分野順**そのままです。

- 第 1 章 … 分野①(分析ソリューションの実装と管理)
- 第 2 章 … 分野②(データの取り込みと変換)

- 第3章 … 分野③(分析ソリューションの監視と最適化)

見出しの末尾の [1.1] などの数字は、公式の出題範囲の番号です。

4 ステップ学習法

購入者限定テキストは3冊で1セット。次の順番で使うと最短です。

- **STEP 1(理解する)**: 「学習用テキスト」を通読し、Fabric の地図と「選ぶ条件・設定の勘どころ」を頭に入れる
- **STEP 2(解き方を掴む)**: 「頻出度別ノート」で頻出度 A の論点を解いて、解き方を掴む
- **STEP 3(腕試し)**: 本サイトの問題演習で出題形式に慣れる
- **STEP 4(再確認)**: 「頻出度別ノート」に戻って頻出度 A の論点を再確認する(試験直前もここ)

「暗記用ノート」は STEP には数えず、空き時間や用語の再確認に使ってください。

DP-700 は操作を問う試験なので、可能なら Fabric の試用版で Lakehouse やパイプラインを作って、実際に触ると定着が段違いです。時間がなければ、先に「頻出度別ノート」で頻出度 A の論点を確認し、それを含む章から読み始めても構いません。

第1章 Implement and manage an analytics solution [1.1/1.2/1.3/1.4]

分析ソリューションの実装と管理は、環境の設定、変更の管理、アクセスの制御、処理の自動実行の4つに分かれます。どの節も「誰が」「どの画面で」「どの値を」変えるかが問われるので、設定の場所と必要な権限を組にして覚えます。

Fabric ワークスペース設定の構成 (Configure Microsoft Fabric workspace settings) [1.1]

Spark ワークスペース設定 (Configure Spark workspace settings) [1.1.1]

Spark ワークスペース設定は、そのワークスペースで動く Spark ジョブの計算資源と動作を決める画面です。場所は「ワークスペース設定 > Data Engineering/Science > Spark コンピューティング」です。変更できるのはワークスペースの管理者ロールだけで、メンバーや共同作成者はできません。
[^c1-1]

ワークスペースを作ると、関連付けられたスターター プールが自動で作られます。ノードやマシンのサイズを選ぶ必要はありません。Spark セッションは5～10秒で開始します。ただし、これは事前に温めた容量を使う最善努力の仕組みで、速さは保証されません。温めた容量が空いていないときは、通常の起動になって時間がかかります。より細かい指定が要る場面では、カスタム Spark プールを作ります。

項目	スターター プール	カスタム Spark プール
中身	事前に温めたライブ プール。ノード サイズは Medium	ノード サイズ、オートスケール、エグゼキューターの動的割り当てを自分で決める
管理者ができること	最大ノード数とエグゼキューターを SKU の上限まで調整	新規プールの作成
前提	なし	容量管理者が容量設定の Spark コンピューティングで「カスタマイズされたワークスペース プール」を有効にしておく(容量レベルの制御は既定で有効)
セッション開始	速い	既定プールをカスタムに変えると長くなる(約 3 分)

最小ノード数を 1 にすると単一ノード クラスターになり、ドライバーとエグゼキューターが 1 ノードで動きます。ノード障害時に復元できる高可用性があり、小さな処理に向く構成。スターター プールの最大ノード数は容量 SKU で決まります。ノードサイズはどの SKU も Medium です。

SKU	既定の最大ノード数	上限
F2・F4	1	1
F8	2	2
F16	3	4
F32	8	8
F64 (試用容量も同じ)	10	16

SKU	既定の最大ノード数	上限
F128	10	32

環境 (Environment) は、計算のプロパティ、ランタイムの版、ライブラリ パッケージ、Spark プロパティをまとめた項目です。ワークスペースの既定の環境を 1 つ選べます (Environment のドロップダウンから新規作成もできます)。既定の環境を無効にすると、Fabric ランタイムの版をドロップダウンから選ぶ形になります。「項目ごとのコンピューティング構成のカスタマイズ」をオンにすると、環境を使って、ノートブックや Spark ジョブ定義ごとにドライバーとエグゼキューターのコアとメモリを変えられます。オフなら、全環境で既定プールの構成が使われます。

「ジョブ」の設定は、ジョブ受け入れの方式と寿命を決めます。

設定	既定と動き
楽観的ジョブ受け入れ (Optimistic Job Admission)	全ワークスペースで既定で有効
アクティブな Spark ジョブに最大コアを予約する	オンにすると楽観的受け入れを無効にし、最大コアを予約する
Spark セッション タイムアウト	対話型セッションの既定の有効期限は 20 分
最大ジョブ有効期間 (Set maximum job lifetime)	暴走ジョブを止めるガードレール。時間を指定すると、超えたジョブが自動でキャンセルされる

最大ジョブ有効期間の対象は、ユーザーが送信したジョブ。ノートブック (パイプライン経由を含む)、Spark ジョブ定義、Livy セッション (高同時実行を含む) が当たります。システム管理のジョブは対象外で、最後まで止まりません。

レイクハウスのテーブル メンテナンス (optimize、vacuum)、マテリアライズド レイク ビューの更新、プラットフォーム操作が当たります。

高同時実行モード (high concurrency) は、複数のノートブックが 1 つの Spark セッションを共有する仕組みです。設定は「Spark settings > High concurrency > For notebooks」で、既定で有効です。共有の条件は次の 5 つ。

- 同じユーザーが実行している (共有は 1 ユーザーの範囲内)
- 同じワークスペースにある
- 既定のレイクハウスが同じ (既定のレイクハウスを持たないノートブック同士は共有できる)
- Spark コンピューティング構成が同じ
- ライブラリ パッケージが同じ (セル内のインライン インストールの違いは共有を妨げない)

1 つの高同時実行セッションで共有できるノートブックは、既定で最大 5 つです (使う環境の Spark プロパティ spark.highConcurrency.max で 2~50 に変えられます)。パイプラインのノートブック アクティビティで共有したい場合は、設定を 2 つ行います。ワークスペースの高同時実行モードで「パイプラインでの複数ノートブック実行」をオンにします。特定のノートブック同士をまとめたい場合は、ノートブック アクティビティの詳細設定で同じセッション タグを付けます。ログはノートブックごとに分かれ、監視画面の「関連ノートブック」タブでジョブの出どころを見分けられます。

場面の判断: 夜間に 3 本のノートブックを続けて動かし、毎回のセッション起動が遅い場合。パイプラインで動かす前提なら、「パイプラインでの複数ノートブック実行」をオンにして同じセッション タグを付けます。別々のユーザーのノートブック同士を相乗りさせる解決にはなりません。カスタム プールを作れない

いと相談された場合は、ワークスペース管理者の権限ではなく、容量管理者の設定が原因です。暴走ジョブだけを止めたいのに optimize が巻き込まれる心配があるなら、最大ジョブ有効期間で足ります。テーブル メンテナンスは対象外のため止まりません。

ドメイン ワークスペース設定 (Configure domain workspace settings) [1.1.2]

ドメインは、組織のデータを部門などの領域ごとに論理的にまとめる仕組みです。ワークスペースをドメインに関連付けると、その中の全項目にドメイン属性が付きます。データ メッシュ (分散型・連合型のガバナンス) を実現する部品で、テナント レベルで決めている設定の一部をドメインへ委任できます。[1-2]

ドメインの割り当ては、項目の可視性やアクセスに影響しません。見えるかどうかは、ワークスペース ロールと項目のアクセス許可で決まります。テナントの全ユーザーが、自分の役割に関係なく全ドメインを OneLake カタログのフィルターで見られます。「財務ドメインに入れたから財務部だけが見られる」という理解は成り立ちません。

ロール	できること	できないこと
Fabric 管理者	ドメインの作成・編集・削除、ドメイン管理者と共同作成者の指定、ワークスペースの割り当て	
ドメイン管理者	自分のドメインの説明の編集、共同作成者の指定、ワークスペースの割り当て、イメージの設定、委任された設定の上書き	ドメインの削除、名前の変更、ほかの管理者の追加と削除

ロール	できること	できないこと
ドメイン共同作成者	自分が管理者のワークスペースをドメインに割り当てる	管理ポータル上のドメインタブを見ること

ドメインを作れるのは Fabric 管理者です。サブドメインは Fabric 管理者かドメイン管理者が作れます。サブドメインは独自のドメイン管理者を持たず、親ドメインの管理者がそのまま管理者になります。設定は全般設定だけで、委任された設定などはありません。

ドメイン共同作成者がワークスペースを割り当てるには、そのワークスペースのワークスペース管理者 (Admin ロール) である必要があります。割り当てを行う場所は、ワークスペース自身の設定。共同作成者の指定は、組織の全員 (既定)、特定のユーザーとグループのみ、テナント管理者と特定のドメイン管理者のみ、の 3 通りから選びます。

ドメインのページで「ワークスペースの割り当て」を選ぶと、3 通りの方法が出ます。

方法	動き
ワークスペース名	名前前で検索し、複数を一度に選べる
ワークスペース管理者	ユーザーまたはセキュリティグループを指定。その人が管理者の全ワークスペースが対象 (My workspaces は除く)
容量	容量を指定。その容量の全ワークスペースが対象 (My workspaces は除く)

管理者と容量で選ぶ操作は、その時点の既存ワークスペースにだけ作用します。あとから作られたワークスペースは入りません。すでに別のドメインに関連付いているワークスペースには警告が出ますが、続行すれば上書きできます。

ただし、Fabric 管理者とドメイン管理者による割り当ての上書きが効くかどうかを決めるのは、テナント設定。「テナントおよびドメイン管理者によるワークスペース割り当ての上書きを許可する」(プレビュー)が有効な場合だけ、上書きできます。

既定のドメイン(Default domain)は、指定したユーザーやセキュリティグループのために使う仕組みです。その人が管理者を務める未割り当てのワークスペースと、その人が今後作る新しいワークスペースが、自動でそのドメインに入ります。すでにドメインが割り当てられたワークスペースは上書きしません。指定された人は、通常、割り当てられたワークスペースのドメイン共同作成者にも自動でなります。

ドメインの設定ペインには、全般設定(名前と説明)、イメージ、管理者、共同作成者、既定のドメイン、委任された設定のタブがあります。全般設定(名前と説明)のうち、ドメイン管理者が編集できるのは説明欄だけです。委任できるテナント設定は2つ。1つはドメインレベルの既定の秘密度ラベル(「情報保護」から選ぶ)。もう1つは認定(Certification)の設定で、有効と無効、認定者、ドキュメント URL を上書きできます。ドメインの作成・編集・削除は Fabric の監査ログに記録され、管理 REST API でもドメインを操作できます。

場面の判断: 営業部のワークスペース管理者がワークスペースを営業ドメインに入れたいのに、管理ポータルにドメイン タブが出ない場合。この人はドメイン共同作成者に指定され、自分のワークスペースの設定から割り当てます。部門ごとに認定者を変えたい場合は、認定の設定をドメインに委任し、ドメイン管理者が上書きします。

OneLake ワークスペース設定 (Configure OneLake workspace settings) [1.1.3]

OneLake は、テナントごとに自動で 1 つ作られる統合データレイク。削除も複数作成もできず、プロビジョニングも要りません。ワークスペースは OneLake の直下に置かれ、1 コピーのデータを複数のエンジンが使います。
[[^]c1-3]

ワークスペース設定の「OneLake」タブでは、ショートカットのキャッシュと診断を設定します。

ショートカットのキャッシュは、外部ショートカットの読み取りで取得したファイルを、ワークスペースのキャッシュに残します。クラウドをまたぐ読み取りの転送料金 (egress) の削減が目的。保持期間は 1～28 日で、ファイルにアクセスするたびにリセットされます。リモート側に新しい版があれば、リモートから返してキャッシュも更新します。保持期間内にアクセスがなければ、そのファイルはキャッシュから削除。1 GB を超えるファイルはキャッシュされません。対象は Google Cloud Storage、S3、S3 互換、オンプレミス データ ゲートウェイのショートカット。「キャッシュのリセット」ボタンでワークスペースのキャッシュを全部消せます。

OneLake 診断 (diagnostics) は、「誰がいつ何にどうアクセスしたか」を記録する機能です。ワークスペース レベルで有効にすると、データ アクセスのイベントが JSON ログとして、選んだレイクハウスに流れます。それを Delta テーブルに変換すれば分析できます。

項目	内容
有効化	ワークスペース設定で「診断イベントをレイクハウスに追加」をオンにし、レイクハウスを選ぶ。イベントが流れ始めるまで最大1時間
必要な権限	診断を有効にするワークスペースの管理者であり、保存先レイクハウスの共同作成者であること
保存先の条件	診断を有効にするワークスペースと同じ容量にあること。受信ネットワーク保護にプライベートリンクを使うなら同じ仮想ネットワーク内
記録の範囲	Fabric UI と Blob／ADLS API の操作は全部記録。Fabric ワークロード経由は一時的アクセス許可の付与が記録される。ショートカットのクロスワークスペース アクセスはソース側ワークスペースで記録される
不変性	保存期間中の JSON ファイルの改ざんと削除を禁止できる (Azure Blob の不変ストレージ=WORM)。期間は診断レイクハウスを含むワークスペースに設定し、一度設定すると変更できない。追加料金はないが保存容量は増える
期間内の削除	HTTP 409 Conflict、エラー コード BlobImmutableDueToPolicy で失敗する
保存先の変更	できる。既存のイベントは元のレイクハウスに残り、新しいイベントだけ新しい保存先に入る。無効にしても情報は保持され、再有効化で同じレイクハウスが使われる

診断と監査ログは役割が違い、同じものとは扱えません。監査ログは「誰がどの項目でどの操作をしたか」を残します。診断は OneLake のデータ アクセスを、レイクハウスに JSON で蓄積します。

テナント設定「ユーザーは Fabric 外のアプリで OneLake に保存されたデータにアクセスできる」(OneLake セクション) は、外部アプリからの入口を制御します。オンなら ADLS API や OneLake ファイル エクスプローラーを使えます。オフにしても、Spark、Data Engineering、Data Warehouse のような内部アプリからは使えます(外部アプリだけが止まります)。OneLake セキュリティでショートカットを使うには、この設定をオンにしておく必要があります。

データの保護も OneLake 側に備わっています。削除したファイルは 7 日間保持されるソフト削除で守られます。冗長性は、使える所ではゾーン冗長(ZRS)で3つの可用性ゾーンに同期コピーし、耐久性は 12 ナインです。ほかの所はローカル冗長(LRS)で 11 ナイン。容量管理ポータルで容量ごとに障害復旧(DR)を有効にすると、別リージョンへ非同期でレプリケートされます。セカンダリリージョンは Azure の標準ペアリングで決まり、変更できません。複製はファイルレベルで非同期です。

場面の判断: クラウドをまたぐ S3 ショートカットの転送料金を減らしたい場合はキャッシュです。誰がレイクハウスのファイルを読んだかを調べたい場合は診断を有効にします。診断の保存先を別の容量のレイクハウスにしたい要望は、同じ容量の条件を満たせないので実現できません。

Apache Airflow ワークスペース設定 (Configure Apache Airflow workspace settings) [1.1.4]

Apache Airflow ジョブは、Fabric の Data Factory に含まれるマネージド サービスです。Python ベースの DAG (有向非巡回グラフ) を実行します。Azure Data Factory の Workflow Orchestration Manager の次世代版です。コー

ドを書く人、または Apache Airflow に慣れた人にはこちらが向き、コードを書かずにオーケストレーションしたいならパイプライン(ノーコード)を選びます。
[^c1-4]

サポートされるのは Apache Airflow 2.10.5 と Python 3.12 です。既存の Airflow ジョブの Airflow バージョンは変更できません。別のバージョンが要るなら、新しいジョブを作ります。プライベート ネットワークと仮想ネットワークは Airflow ジョブでは未対応です。

Workflow Orchestration Manager と比べると、Git 同期、Azure Key Vault をバックエンドにする設定、プライベート パッケージの要件へのインストールは両方で使えます。高可用性、遅延可能オペレーター、一時停止と再開の TTL は Fabric 側だけにあり、オートスケールは Fabric が対応・Workflow Orchestration Manager は一部対応です。診断ログとメトリクス、Blob Storage は Fabric では使えません。

ランタイムの設定は、ワークスペース設定の Data Factory セクション > Apache Airflow ランタイム設定にあります。

項目	スターター プール(既定)	カスタム プール
サイズ	Large	自分で選ぶ
起動	即時	停止状態から始まる
再開	最大 5 分	最大 5 分
停止の挙動	20 分間操作がないと停止	手動で一時停止するまで常時オン
向く環境	開発者向け	本番向け

カスタム プールは、既定のデータ ワークフロー設定のドロップダウンから「新しいプール」を選んで作ります。設定項目は、名前、計算ノード サイズ、オートスケールの有効化、追加ノード (Extra nodes) です。ノード サイズは Large が複雑な DAG と本番向け、Small が単純な DAG 向け。追加ノード 1 つにつきワーカー 3 つ分の同時実行容量が増えるので、追加ノード 4 つなら 12 ワーカー分増えます (3 × 4)。項目ごとのコンピューティング構成のカスタマイズ設定が無効だと、ワークスペース内の全環境でスターター プールが使われます。

DAG は Airflow ジョブの項目の中でファイルとして作ります。「+ 新しい項目 > Data Factory > Apache Airflow ジョブ」で項目を作り、「New DAG file」でボーイプレート の DAG が出ます。保存すると DAG が Apache Airflow UI に読み込まれ、実行状況を見られます。ジョブは Git 同期にも対応します (機能の一覧で Git sync は Yes です)。Fabric の項目を DAG から動かすには、Fabric プラグインの `FabricRunItemOperator` を使います。引数の `job_type` は大文字小文字を区別します。ノートブックは "RunNotebook"、Spark ジョブ定義は "sparkjob"、パイプラインは "Pipeline" です。認証には、サービス プリンシパル (テナント ID・クライアント ID・シークレット) を登録した Fabric 接続を使います。サービス プリンシパルは、実行対象の項目があるワークスペースの共同作成者にしておかないと実行できません。 `wait_for_termination` は項目の完了を待つ指定、 `deferrable` は待つ間にリソースを解放する指定です。

Fabric のライフサイクル管理の実装 (Implement lifecycle management in Fabric) [1.2]

バージョン コントロール (Configure version control) [1.2.1]

Fabric の Git 統合は、ワークスペース単位でバージョン管理を行う機能 (ALM) です。ブランチで共同作業でき、サブフォルダーを含むワークスペースの構造が Git リポジトリに保たれます。[^c1-5]

項目	内容
対応するプロバイダー	Azure DevOps と GitHub (クラウド版の GitHub Enterprise を含む)。Azure DevOps の組織が Fabric と別テナントでもよい
テナント設定	「ユーザーは Fabric 項目を作成できる」「ユーザーはワークスペース項目を Git リポジトリと同期できる」。GitHub は「GitHub リポジトリと同期できる」。新しいワークスペースへ分岐するなら「ワークスペースを作成する」も必要
接続できるロール	ワークスペース管理者だけ。接続後は権限のあるユーザーが作業できる
GitHub の認証	個人用アクセストークン。細粒度トークン (推奨) はリポジトリ権限 Contents の読み取りと書き込み、クラシックトークンは repo スコープ
接続時の指定	ブランチ (既存を選ぶか新規作成) とフォルダー (空白ならルート)。どちらも一度に 1 つだけ
Azure DevOps の URL	https://dev.azure.com/{organization}/{project}/_git/{repository}

接続は、ユーザーが項目を Git と同期できるテナント設定を確かめ、ワークスペース管理者が Git プロバイダーに接続し、ブランチとフォルダーを指定してから、初回の同期を行う順に進めます。初回の同期は、ワークスペースか Git ブランチの一方が空なら、内容のある側から空の側へコピーされます。両方に内容がある場合は、どちらを正とするか同期の方向を選ばなければなりません。

コミットは「ソース管理」の「変更」から行い、項目を選べます。Git 側の更新を取り込む「更新」は項目を選べず、常にブランチ全体を最新コミットまで同期します。Git 側に更新があるとコミットは無効になり、先に更新しないとコミットできません。同じ項目がワークスペースと Git の両方で変わっていると、状態がコンフリクトになり、コミットは無効です。この状態で「更新」を選ぶと「コンフリクトを解決」が出ます。残す版を選ぶなどして解決するまで、更新は終わられません。

ブランチの操作はロールで変わります。既定では、ブランチの切り替えと新ブランチへのチェックアウトにはワークスペース管理者が必要です。ワークスペース設定「少なくとも共同作成者のロールを持つユーザーによる Git ブランチ変更を許可」を有効にします。すると、全項目に書き込み権限のある共同作成者とメンバーも可能になります。この設定はワークスペース単位で、テナント側のスイッチはありません。

操作	動き
Checkout new branch	ワークスペース管理者が実行。最後に同期したコミットから新ブランチを作り、接続を変える。ワークスペースの内容は変えない
Branch out to another workspace	共同作成者以上が実行。最終コミットを元に新しいワークスペースを作る、または既存のブランチ ワークスペースを新ブランチに切り替える

ブランチ ポリシーなどで接続中のブランチに直接変更できない場合は、Checkout Branch を使います。機能ごとに別ワークスペースへ Branch out して開発し、プル リクエストで本流へ戻す流れが基本です。

上限	値
ワークスペースの項目数	最大 1,000。Git ブランチに 1,000 超の項目があると同期は失敗する
コミット サイズ	Azure DevOps: サービス プリンシパル は 25 MB、既定の SSO (Entra ID ユーザー) は 125 MB。GitHub: 一度にコミットするファイルの合計は 50 MB まで
ブランチ名	最大 244 文字
ファイル名を含むフルパス	最大 250 文字
フォルダー構造	10 階層まで維持

サポート外の項目があってもワークスペースは接続できますが、その項目は無視されます。保存も同期も削除もされず、ソース管理に表示されてもコミットと更新はできません。2026 年 12 月 1 日以降は、ワークスペースの項目に読み書き権限のないユーザーは Git 統合を使えなくなります (秘密度ラベルや保護ポリシーの付いた項目へのアクセスを失う原因になりえます)。

場面の判断: GitHub に接続したワークスペースで 120 MB の変更を一度にコミットしようとして失敗した場合は、GitHub の合計 50 MB の上限があるので 3 回以上に分けます。共同作成者が自分のブランチへ切り替えられないなら、ワークスペース単位の設定を管理者が有効にします。

データベース プロジェクト (Implement database projects) [1.2.2]
データベース プロジェクトは、スキーマ オブジェクトをコードとして管理するプロジェクト。Fabric Data Warehouse では、VS Code の SQL Database Projects 拡張機能で作成、ビルド、発行します。[^c1-6]

拡張機能は DacFx (Data-Tier Application Framework) パッケージで動きます。プロジェクトのビルドと発行、スキーマ比較、変更のスクリプト化、.dacpac の抽出と展開を行います。

項目	内容
プロジェクトの形式	SDK スタイルのプロジェクトだけ対応
作成時の選択	プロジェクトの種類は SQL Server Database、ターゲット プラットフォームは「Synapse Data Warehouse in Microsoft Fabric」
前提	既存のウェアハウスに共同作成者以上でアクセスできること。.NET SDK、VS Code、拡張機能の SQL Database Projects と SQL Server (mssql)

既存のウェアハウスからプロジェクトを作る手順は次のとおりです。

1. mssql で接続プロファイルを作る。サーバー名はウェアハウス設定の SQL 接続文字列 (<server unique ID>.datawarehouse.fabric.microsoft.com) で、SQL 分析エンドポイントとは別物
2. 認証の種類に Microsoft Entra ID - Universal with MFA を選ぶ
3. Database Projects の「Create Project From Database」を実行する
4. フォルダー構造は Schema/Object Type (推奨) を選ぶ

ビルドすると新規と更新のオブジェクトが検証され、dacpac ファイルができます。発行は、プロジェクトとウェアハウスの差を解消するスクリプトを作って実行する操作です。発行先は「Publish to an existing SQL server」で選び、初回は「Generate Script」で内容を確認してから発行します。スクリプトの生成は

対象を変更しません。発行オプションは publish プロファイル (.publish.xml) に保存して使い回せます。

発行の設定	動きと勧め
BlockOnPossibleDataLoss	データ損失の恐れがあるとデプロイを止める。本番では True
DropObjectsNotInSource	プロジェクトに無いオブジェクトを対象から削除する。開発・テスト向けで、本番では注意
PreDeploy / PostDeploy	スキーマデプロイの前後にカスタム SQL を実行する (PreDeploy の例: 表を削除する前にデータを退避・バックアップする)。何度流しても問題の出ない (冪等な) 書き方にする

ウェアハウスの開発モードは 2 つあります。ライブ開発はポータルで直接変更でき、即時反映されますがバージョン管理されません。Git 統合モードは Azure DevOps か GitHub に接続し、ブランチ、同期、機能ワークスペースへの Branch out ができます。Git 統合はワークスペース レベルの双方向同期で、Fabric REST API でコミット、同期、ブランチ操作を自動化できます。

デプロイの選択肢は 2 系統。1 つは Fabric のデプロイ パイプライン (dev → test → prod の昇格) です。もう 1 つは外部 CI/CD で、DacFx タスク、SqlPackage CLI、REST API を Azure DevOps や GitHub から実行します。クライアント ツールは 2 つ。VS Code (MSSQL 拡張) はデータベース プロジェクトによる開発、スキーマ比較、CI 向けです。SSMS は対話的なクエリ、パフォーマンス チューニング、セキュリティ ロールの管理向けです。

場面の判断: ポータルで直接直した変更が履歴に残らず困っている場合は、ライブ開発のままでは追えません。Git 統合モードにするか、プロジェクトを抽出

して Git で管理します。本番の発行でデータ損失を避けたいなら、BlockOnPossibleDataLoss を True にします。

デプロイ パイプライン(Create and configure deployment pipelines) [1.2.3]

デプロイ パイプラインは、開発から本番へコンテンツを段階的に昇格させる仕組み。ステージにワークスペースを 1 つずつ割り当てます。[^c1-7]

```
Development --deploy--> Test --deploy--> Production
(Workspace A)           (Workspace B)       (Workspace C)
```

ステージ数は 2～10 で、既定は 3 (Development、Test、Production) です。ステージ数と名前は、作成後に変更できません。最終ステージは既定で公開 (public) 扱いで、パイプラインへのアクセスがない利用者には普通のワークスペースに見えます。

ペアリング (pairing) は、あるステージの項目を隣のステージの同じ項目と関連付ける処理です。ワークスペースをステージに割り当てた時と、ペアのない内容をクリーン デプロイした時に行われます。ペアになった項目は、名前を変えてもペアのままです。名前、種類、フォルダーが同じでも、ペアになっていなければ上書きされません。重複コピーが作られ、それがペアになります。ワークスペースの割り当て後に追加された項目は、自動ではペアになりません。

デプロイでは、コピー元の項目が、ペアのターゲット項目を上書きします。コピー元に無い項目は、ターゲットにそのまま残ります。デプロイでは、少なくとも 1 項目を選ばなければなりません。メタデータだけがコピーされるので、デプロイ後にセマンティック モデルを更新してから使います。

コピーされないもの
データ、URL、ID、ワークスペースと項目のアクセス許可、ワークスペース設定、アプリのコンテンツと設定、個人用ブックマーク
セマンティック モデルのロール割り当て、更新スケジュール、データ ソースの資格情報、クエリ キャッシュ設定、承認設定

デプロイ規則は、ターゲット ステージで項目ごとに定義します。次にデプロイするまで適用されません。種類は 3 つです。

規則	設定できる項目
データ ソース規則	データフロー Gen1、セマンティック モデル、ページ分割レポート、ミラーリング データベース
パラメーター規則	データフロー Gen1、セマンティック モデル (レポートには使えない)
既定のレイクハウス規則	ノートブックだけ

規則を作るには項目の所有者である必要があります。データ ソース規則は同じ種類のデータ ソースにしか変えられず、同じデータ ソースを複数の規則で使えません。

オートバインド (autobinding) は、デプロイした項目の依存先がデプロイ先ステージにあれば、そのステージ内の項目へ自動で接続する動作です。パイプラインをまたぐ場合は、両パイプラインのステージ数が同じである必要があります。パラメーターで接続を制御している場合は、オートバインドされません。依存先がデプロイ先に存在しなければデプロイは失敗します。たとえばセマンティック モデルのないレポートだけをデプロイしても、そのモデルがターゲッ

トに無ければ成功しません。選択式デプロイの「Select related」で依存先も一緒に選べます。

デプロイの方法	内容
すべてのコンテンツ	隣のステージへ全部
選択的デプロイ	選んだ項目だけ隣のステージへ
逆方向のデプロイ	後ろのステージから前へ。Get started のページは「対象のステージが空 (ワークスペース未割り当て) の時だけ」とし、デプロイプロセスのページは「隣のステージへはどちら向きにもデプロイできる」と書く (文書によって書き方が違う)

ステージの比較は、パイプライン ホームの比較アイコンで見ます。緑はメタデータが同じ、オレンジは項目が変更・追加・削除されています。項目のラベルは「Not in source」(ソースに無い。選べず、デプロイの影響も受けない)、 「Different from source」(ペアと差がある。デプロイでソースが上書きする)、 「Only in source」(新規。デプロイで複製される)、 「Same as source」です。設定した規則が未適用の項目も「Different」と出ます。

権限は 2 層です。パイプライン自体の権限は Admin だけで、共有、編集、削除に必要です。Microsoft 365 グループはパイプライン管理者にできません。デプロイするには、パイプライン管理者である必要があります。かつ、デプロイ元とデプロイ先の両方のワークスペースで、共同作成者以上 (共同作成者、メンバー、管理者) であることも必要です。パイプライン管理者だけでは、コンテンツの操作はできません。

既定では、Fabric のリネージが依存関係とデプロイ順序を決めます。明示的な順序や前後のアクションが要る場合は、オプションのデプロイプラン(プレ

ビュー)を添付します。増分更新のポリシーを保持してデプロイできるのは、拡張セマンティック モデル メタデータを使うモデルだけ。増分更新テーブルの名前変更などの再発行は、データ損失の恐れがありデプロイが失敗します。

Git 統合とは役割を分けて組み合わせます。開発ステージのワークスペースを Git に接続してコミットで保管し、昇格はデプロイ パイプラインで行います。デプロイは Power BI または Fabric のデプロイ パイプライン REST API から実行でき、パイプラインの自動化に使えます。

場面の判断: パイプライン管理者が本番へデプロイしようとして失敗する場合は、デプロイ元とデプロイ先のワークスペースの共同作成者以上のロールを持っていません。テスト ステージの接続先データベースだけを本番用に切り替えたいなら、データ ソース規則か、パラメーターで管理しているならパラメーター規則を、本番ステージで定義します。同じ名前のレポートが上書きされず 2 つに増えたなら、ペアリングされていなかった結果です。

セキュリティとガバナンスの構成 (Configure security and governance) [1.3]

ワークスペース レベルのアクセス制御 (Implement workspace-level access controls) [1.3.1]

ワークスペース ロールは、そのワークスペースの項目全体への権限を決める最も広い層です。ロールは該当ワークスペースだけで有効で、ほかのワークスペース、容量、テナントには及びません。ロールを持たないユーザーはワークスペースにアクセスできません。[^c1-8]

できること	Admin	Member	Contributor	Viewer
ワークスペースの更新・削除	○			

できること	Admin	Member	Contributor	Viewer
他の管理者を含む人の追加・削除	☐			
Member 以下の人の追加、他者の再共有の許可	☐	☐		
ワークスペース ID の作成、Git リポジトリへの接続	☐			
ウェアハウスなどの項目の作成・変更	☐	☐	☐	
パイプライン、ノートブックなどの書き込み・実行	☐	☐	☐	
OneLake API と Spark でのデータ読み取り (ReadAll)	☐	☐	☐	
OneLake イベントの購読	☐	☐	☐	

できること	Admin	Member	Contributor	Viewer
内容と実行出力の表示、SQL 分析エンドポイントへの接続、T-SQL での読み取り (ReadData)	○	○	○	○

Viewer は内容と実行出力を見て T-SQL で読めますが、ReadAll (OneLake API と Spark)、書き込み、実行はできません。Contributor と Viewer も、再共有 (Reshare) の権限があれば項目を共有できます。

ロールの割り当て先は、個人、セキュリティ グループ、Microsoft 365 グループ、配布リスト。Entra サービス プリンシパルも割り当てられ、API 操作で同じ権限を継承します。ノートブックの作成・変更・書き込み・削除は Admin、Member、Contributor が Items REST API でもできます。Job Scheduler API では、この 3 ロールが実行の開始とキャンセルをでき、Viewer を含む全ロールが実行状態の監視と出力の確認をできます。

複数のグループに属する場合の権限は、割り当てられたロールのうち最も高いもの。入れ子のグループにロールを割り当てると、含まれる全ユーザーに適用されます。管理は個人ではなくセキュリティ グループへの割り当てで行い、必要最小限のロールだけを付けます (最小特権)。

項目を共有してもワークスペース ロールは変わりません。項目のアクセス許可を削除しても、ワークスペースの Viewer ロールがあれば、その項目を引き続き見られます。見せたくないなら、ワークスペースのロールも削除します。Admin、Member、Contributor は書き込み権限で全データを読み書きでき

るので、OneLake セキュリティ ロールの影響を受けません。OneLake セキュリティで制限できるのは、主に Viewer です。

パイプラインのノートブック アクティビティでワークスペース ID を使うには、準備が 2 つ。テナント設定「サービス プリンシパルは Fabric パブリック API を呼び出せる」(既定は無効)を有効にし、ワークスペース ID に共同作成者以上を付与します。

場面の判断: 分析担当者にデータ エンジニアリング用のワークスペースのデータを T-SQL で読ませたいが、ノートブックの編集はさせたくない場合です。最小のロールは Viewer です。Viewer なら ReadData で T-SQL の読み取りができ、書き込みと実行はできません。メンバーを増やす権限だけを部門長に渡したいなら、Member を付けます。Admin までは要りません。

項目レベルのアクセス制御 (Implement item-level access controls)
[1.3.2]

項目のアクセス許可は、特定の項目だけに有効で、ほかの項目には及びません。ワークスペースにアクセスできないユーザーに、1 つの項目だけを使わせる用途に使います。[^c1-9]

項目を共有すると、既定で Read が付きます。Read はメタデータと関連レポートの表示ができますが、SQL や OneLake の基になるデータへのアクセスは含みません。ウェアハウスを共有できるのはワークスペースの Admin か Member で、共有された受信者は再共有できません。

許可	中身
Read (追加選択なし)	SQL 分析エンドポイントへの接続だけ (SQL Server の CONNECT 相当)。T-SQL の GRANT でオブジェクト権限を与えない限り、テーブルやビューをクエリできない

許可	中身
ReadData	「SQL ですべてのデータを読み取る」。全テーブルとビューを読み取り専用で読める (db_datareader 相当)。T-SQL でさらに GRANT/REVOKE/DENY で絞れる
ReadAll	「Apache Spark ですべてのデータを読み取る」。OneLake のファイルを Spark などで読む
SubscribeOneLakeEvents	ウェアハウスの OneLake イベントをリアルタイム ハブで購読する
Monitor	DMV (sys.dm_exec_requests など)、クエリテキスト、接続、セッションの照会
Audit	ウェアハウスの監査ログの有効化、構成、照会

ReadData と ReadAll は別の権限で、重複しません。ReadData は T-SQL、ReadAll は OneLake と SQL 分析エンドポイントで使います。既定の付与は、Read と ReadData が Admin、Member、Contributor、Viewer です。ReadAll は Admin、Member、Contributor に付き、Viewer には付きません。Monitor は Admin、Member、Contributor に、Audit は Admin に既定で付きます。Monitor と Audit を共有で付けられるかは、文書によって書き方が違います (共有の文書の表は「単独では付けられない」、SQL 監査ログの文書は「管理者が共有のダイアログで Audit を付けられる」)。この 2 つは、何ができる許可かで覚えます。

特定のテーブルだけに権限を与えたい場合は、追加権限なしで共有 (Read のみ) してから、T-SQL の GRANT で特定のオブジェクトへ権限を与えます。

共有したウェアハウスのレポートを他人が見る場合は、セマンティック モデルの接続モードに必要な権限が変わります。DirectQuery モードと Direct Lake モードでは、ウェアハウスの ReadData (または特定のテーブルやビューへの細かい権限) がない閲覧者は見られません。インポート モードなら、追加の権限は要りません。

Manage permissions 画面の「Remove access」は、項目権限をすべて削除します。「Remove ReadData」と「Remove ReadAll」は、それぞれの権限だけを削除します。付与や削除の伝達には最大 2 時間かかることがあります。画面には即時に反映され、SQL 分析エンドポイントへ反映するには再サインインします。共有の受信者は所有者の ID (委任モード) でウェアハウスにアクセスするので、所有者をワークスペースから外してはいけません。受信者は共有されたウェアハウスにだけアクセスでき、同じワークスペースの他項目は見えません。

ワークスペース ロールと項目の共有は、項目に対する操作を決めるコントロール プレーンです。どのデータを見られるかを決めるデータ プレーンは、OneLake セキュリティ (1.3.8) が担います。

場面の判断: ワークスペースには入れずに、ウェアハウスの売上テーブルだけを読ませたいなら、Read のみで共有し、そのテーブルへ GRANT SELECT します。全テーブルを読ませるなら ReadData を付けて共有します。Direct Lake のレポートを共有したのに閲覧者が開けないなら、ウェアハウスの ReadData が足りません。

行・列・オブジェクト・フォルダー/ファイル レベルのアクセス制御 (Implement row-level, column-level, object-level, and folder/file-level access controls) [1.3.3]

データの絞り込みには 4 つの層があり、ウェアハウスと OneLake で道具が違います。[^c1-10]

行レベル セキュリティ(RLS)

RLS は、インライン テーブル値関数で述語を作り、CREATE SECURITY POLICY で適用します。対応するのは、Fabric Data Warehouse と SQL 分析 エンドポイント。SQL 分析エンドポイントでは CREATE TABLE はできませんが、CREATE SCHEMA、CREATE FUNCTION、CREATE SECURITY POLICY はできます。

```
CREATE FUNCTION Security.tvf_securitypredicate(@SalesRep AS
nvarchar(50))
RETURNS TABLE
WITH SCHEMABINDING
AS
RETURN SELECT 1 AS tvf_securitypredicate_result
WHERE @SalesRep = USER_NAME() OR USER_NAME() = 'manager@contoso.com';

CREATE SECURITY POLICY SalesFilter
ADD FILTER PREDICATE Security.tvf_securitypredicate(SalesRep)
ON sales.Orders
WITH (STATE = ON);
```

sales.Orders が 10 行 (Sales1 が 5 行・合計 1,700、Sales2 が 5 行・合計 2,950) だとします。Sales1 でサインインした人は 5 行、manager は 10 行 (合計 4,650)、どちらでもない人は 0 行を見ます。関数を変更するには、先にポリシーを DROP しておかなければなりません。

- フィルター述語は、SELECT、UPDATE、DELETE で該当しない行を黙って除外します。全行が除外されれば空の結果が返り、アプリは除外に気づきません。INSERT は可能です。
- 述語はテーブルごとに個別に定義します。ポリシーのないテーブルはフィルターされません。
- セキュリティ ポリシーは dbo や db_owner のメンバー、テーブル所有者を含む全ユーザーに適用されます。Admin、Member、Contributor が全行を見る必要があるなら、述語の側でそれを許可する書き方にします (上の manager の条件が例です)。
- SCHEMABINDING は既定で ON です。ON なら述語関数への SELECT と EXECUTE の権限チェックは省かれます。OFF の場合は、述語関数と関連オブジェクトへの SELECT か EXECUTE の権限が要ります。
- ポリシーの作成、変更、削除には ALTER ANY SECURITY POLICY 権限が必要です。作成と削除には、スキーマの ALTER 権限も欠かせません。述語を追加するには、関数への SELECT と REFERENCES、対象テーブルへの REFERENCES、使う列への REFERENCES が要ります。

列レベル セキュリティ (CLS) とオブジェクト レベル セキュリティ

CLS は、T-SQL の GRANT で列を指定して SELECT を付与します。除外した列を含むクエリは、SELECT 権限拒否のエラーになります。

```
GRANT SELECT ON Customers(CustomerID, FirstName, LastName, Phone,  
Email) TO [Charlie@contoso.com];
```

CLS は Microsoft Entra 認証だけに対応し、SQL 認証は使えません。ウェアハウスに対する Direct Lake モードの Power BI クエリは、CLS を守るために DirectQuery へフォールバックします。オブジェクト レベル セキュリティ (SQL 粒度のアクセス許可) は、GRANT、REVOKE、DENY で管理します。ユーザーを割り当てる先は、ユーザー定義のデータベース ロールや組み込みロール。

セマンティック モデルとウェアハウスの違い

ウェアハウスの RLS と CLS は T-SQL で書き、そのウェアハウスへのクエリに効きます。Direct Lake モードの Power BI クエリも、この RLS・CLS を守るために DirectQuery へフォールバックします。セマンティック モデルの RLS と OLS は、セマンティック モデルのロールとして定義し、そのモデルを読むレポートに効きます。モデルのロールに書いた制限は、ウェアハウスへ直接 SQL で読む人には効きません。

フォルダー/ファイル レベル (OneLake のデータ アクセス ロール)

OneLake セキュリティ ロールは、4 つの要素の組。権限 (Read または ReadWrite)、種類 (Grant ロールだけで Deny ロールは未対応)、ロール内のデータ (テーブル、フォルダー、スキーマ)、ロールのメンバーです。メンバーには Entra のユーザー、グループ、非ユーザー ID を入れられます。Read は読み取りとメタデータの表示で、SQL の VIEW_DEFINITION と SELECT に当たります。ReadWrite は読み書きで、ALTER、DROP、UPDATE、INSERT に当たります。ReadWrite は Viewer に書き込みを許す用途で、RLS や CLS の制約を含められません。

既定は拒否 (deny-by-default) で、ロールが明示的に許可しない限りデータにアクセスできません。フォルダーの権限は下位のファイルとサブフォルダーに継承されます。子へのアクセス権があれば、親フォルダーの一覧表示と走査もできますが、兄弟のファイルやフォルダーは見えません。

CLS で絞ったテーブルを select * すると、Spark は許可された列だけを返して成功しますが、SQL 分析エンドポイントとセマンティック モデルはエラーになります。RLS ロールは動的クエリと複数テーブルのクエリをサポートします。RLS と CLS を併用するなら 1 つのロールの中で定義します。RLS を含むロールと CLS を含む別ロールの組み合わせは未対応で、クエリ エラーとなり、使えません。

上限	値
1 項目あたりのロール数	250 (Azure サポートで 1,000 まで増やせる)
1 ロールあたりのメンバー	500 ユーザーまたはグループ
1 ロールあたりの権限	500

場面の判断: ウェアハウスで、営業担当には自分の行だけを、マネージャーには全行を見せたい場合です。述語に担当者の列の一致とマネージャーの例外を書き、RLS ポリシーを作ります。社員表の給与の列だけを隠したいなら、CLS (許可する列だけに GRANT SELECT) が当たりです。レイクハウスの Files の特定フォルダーだけを分析チームの Viewer に読ませたいなら、そのフォルダーに Read を付けた OneLake セキュリティ ロールを作ります。

動的データ マスク (Implement dynamic data masking) [1.3.4]

動的データ マスク (DDM) は、クエリ結果で、権限のないユーザーに対してデータをマスクして見せる機能。データベース内のデータは変わらず、アプリを

改修せずに使えます。ウェアハウスと SQL 分析エンドポイントで対応します。
[^c1-11]

関数	動き
default()	データ型に応じた完全マスク。文字列は XXXX (4 文字未満なら短く)、数値型は 0、日付と時刻型は 1900-01-01 00:00:00.0000000
email()	先頭 1 文字と固定の .com を残し、 aXXX@XXX.com の形にする
random(start, end)	数値型を、指定範囲のランダムな値に置き換える
partial(prefix,[padding],suffix)	先頭と末尾を出し、間に任意の文字列を入れる

列定義で付けるか、既存の列に追加します。

```
Phone# varchar(12) MASKED WITH (FUNCTION = 'default()') NULL
ALTER TABLE ... ALTER COLUMN [Phone Number] ADD MASKED WITH (FUNCTION = 'partial(1,"XXXXXXX",0)')
```

partial(1,"XXXXXXX",0) は 555.123.1234 を 5XXXXXXX にします (random の例は 'random(1, 12)').

操作	必要な権限
マスク付きの列を持つテーブルを作る	CREATE TABLE とスキーマの ALTER
マスクの追加、置換、削除	ALTER ANY MASK とテーブルの ALTER
マスクされたデータを見る	テーブルの SELECT

操作	必要な権限
マスクなしのデータを見る	列の UNMASK、またはデータベースの CONTROL

CONTROL は ALTER ANY MASK と UNMASK の両方を含みます。ワークスペースの Admin、Member、Contributor、またはウェアハウスで昇格した権限を持つ人は、CONTROL を既定で持ちます。そのためマスクなしで見えます。それ以外の人が見るのは、マスクされた値。セキュリティ担当者には ALTER ANY MASK を付与します。

DDM は悪意のあるユーザーを防ぐ機能ではありません。クエリ権限のある人は、範囲クエリで推測したり総当たりしたりして、マスクされた値を推定できます。偶発的な露出の防止用で、CLS、RLS、SQL 粒度の権限と併用して最小権限にします。

3 つの違いは、CLS が列そのものを見せない(アクセス拒否)、RLS が行を見せない(WHERE 述語)、DDM が列の値をマスクして見せる、です。SQL 分析エンドポイントがユーザー ID モード(1.3.8)の間は、DDM は OneLake セキュリティでサポートされません。

場面の判断: コールセンターの担当者にメールアドレスの先頭 1 文字だけを見せたいなら、email() のマスクを付けます。あるアナリストにだけ全文を見せたい場合は、その人に列の UNMASK を付与します。メンバー ロールのユーザーがマスクを見られない、という相談は成り立ちません。Member は CONTROL を持つので、マスクなしで見えます。

秘密度ラベル(Apply sensitivity labels to items) [1.3.5]

秘密度ラベルは、Fabric と Power BI のデータを分類し保護する仕組みです。使うのは、Microsoft Purview Information Protection のラベルとポリシー。

利用には Purview Information Protection の適切なライセンスが要ります。Azure Information Protection のラベルを使っているなら、Purview の統合ラベル付けプラットフォームへの移行が必要です。[^c1-12]

ラベルを適用できるユーザーは、テナントで秘密度ラベルを有効にする時に指定します。それだけでは足りず、Purview のラベル ポリシーでそのユーザーにラベルが公開されている必要があります。Power BI 項目には、Power BI Pro か PPU のライセンスも欠かせません。

機能	Fabric での扱い
手動ラベル付け	全 Fabric 項目で可能
既定のラベル	項目の作成と編集時、ユーザーが選ばなければ付く。Power BI 以外は、作成ダイアログが明確な場合と、名前や説明などの一部の属性の編集時に限る
必須ラベル付け	完全にサポートされるのは Power BI 項目だけ。レイクハウス、パイプライン、ウェアハウスでは強制されず、ラベルなしで保存できる
プログラムによるラベル付け	Power BI 管理 REST API で追加、変更、削除。全 Fabric 項目で可能
ダウンストリーム継承	ラベルが依存する下流の項目へ伝わる。既定でオン
作成時の継承	既存項目から作った項目がラベルを継承する
データソースからの継承	セマンティック モデルだけ

ダウンストリーム継承は、Power BI → Power BI、Fabric → Fabric、Fabric → Power BI に対応します。Power BI → Fabric は非対応です。レイクハウスやウ

エアハウスから自動生成された項目 (SQL 分析エンドポイントなど) は、親のラベルを取り、さらに上流の項目からは継承しません。作成時の継承は、レイクハウスから作ったパイプラインやノートブックが親のラベルを継ぐ形です。新しいデータソースへの接続やリネージの追加でも、ラベルは再評価されて更新されます。

ドメインの委任設定の「情報保護」で、ドメイン内のワークスペースの項目に既定で付けるラベルを指定できます (Fabric 管理者かドメイン管理者)。

ラベルに Purview の保護ポリシー (暗号化) が紐づく、アクセス制御が働きます。効くのは 3 つの場面です。ラベルを付けたテナント内で、保護ポリシーのある項目へアクセスする時。Power BI Desktop の .pbix ファイルを開く時。対応するエクスポート経路で作ったファイルを開く時です。後ろ 2 つは Purview の公開ポリシーに紐づくラベルが前提です。対応するエクスポートは、Excel、PDF、PowerPoint への出力、Excel での分析、.pbix のダウンロードで、ラベルと保護が引き継がれます。.csv や .txt へのエクスポートと、クロステナントのデータ共有では、アクセス制御は効きません。Fabric 項目の多くは、ラベル付きでエクスポートしても警告が出るだけで、ラベルは移りません。

2026 年 12 月 1 日以降は、変更が 2 つ入ります。デプロイ パイプラインでは、保護ポリシー付きの秘密度ラベルで守られた項目を含むワークスペースに限り、全項目の読み書き権限を持たないユーザーは、そのワークスペースへのデプロイと一部のステージへの割り当てができなくなります。Git 統合では、ワークスペースの項目に読み書き権限を持たないユーザーは Git 統合を使えなくなります。

場面の判断: レイクハウスにラベルを付けた後で、SQL 分析エンドポイントのラベルを付け直す必要はありません。自動生成された項目は親のラベルを取ります。すべての項目にラベルを強制したい要望は、Power BI 項目以外では

実現できません。ラベル付きのセマンティック モデルを .csv に出力しても、保護は移りません。

承認(Endorse items) [1.3.6]

承認(Endorsement) は、信頼できる高品質のコンテンツを見つけやすくするバッジです。UI に表示され、一部の一覧で優先して並びます。秘密度ラベルがデータの機密性を分類して保護する機能なのとは目的が違います。[^c1-13]

バッジ	意味	付けられる人	対象
Promoted (昇格)	作成者が共有と再利用の準備ができたと考える	項目の所有者と、書き込み権限のあるユーザー	Power BI ダッシュボードを除く全項目
Certified (認定)	組織が認めた確認者が、品質基準を満たすと認定した	Fabric 管理者が指定した認定者だけ (認定者にも書き込み権限が要る)。誰でも認定を依頼できる	Power BI ダッシュボードを除く全項目
Master data (マスター データ)	組織の中核となる、信頼できる唯一のデータ	Fabric 管理者が指定した認可済みレビュー担当者だけ	レイクハウスやセマンティック モデルなど、データを含む項目

認定とマスター データの承認は、Fabric 管理者がテナント設定で有効にした時だけ使えます。認定を有効にする設定は、エクスポートと共有の設定 > 認定にあります。認定者はセキュリティ グループで指定し、ユーザーを直接は入力できません。認定権を与えたくないサブグループは、除外するオプションで外します。組織の認定プロセスの文書 URL を設定すると、承認設定ダイアログの「詳細情報」リンクになります。未設定なら、案内されるのは Fabric 管理者への問い合わせ。

認定の設定を委任すると、ドメイン管理者が自分のドメインで、有効と無効や認定者などのテナント設定を上書きできます。上書きできるのは、「ドメイン管理者が有効/無効にできる」チェックボックスを入れた場合で、有効と無効以外の設定も含めて上書きできます。

Promoted にするには、項目の設定を開き、承認セクションで Promoted を選びます。Power BI セマンティック モデルには、「検出可能にする」チェックボックスが出ることがあります（アクセス権のない人にも見つけてもらう設定）。Certified は、ワークスペースで項目の設定を開き、承認セクションで Certified を選びます。認定者でなければボタンがグレーになり、認定の依頼しかできません。

場面の判断: データ エンジニアが自作のレイクハウスを全社の標準データとして示したいなら、マスター データを依頼します。自分で付けられるのは Promoted までで、Certified と Master data は付けられません。認定を付けられないという相談では、認定者のセキュリティグループに入っているか、テナントの認定が有効かを確認めます。

Fabric 監査ログ (Implement and use Microsoft Fabric audit logs)

[1.3.7]

監査ログは、誰がどの項目でどの操作をしたかを記録します。規制対応や記録管理の要件を満たすために使います。[^c1-14]

アクセスには、Exchange Online の Audit Logs ロールが要り、持たないユーザーは検索できません。既定では、Compliance Management と Organization Management のロール グループにこのロールが割り当てられています。検索は Microsoft Purview ポータルで行います。

項目	内容
日付範囲	開始日と終了日で指定。既定は過去 7 日間。表示は UTC
フィルターの種類	日付と時刻の範囲、アクティビティ、ユーザー、ファイル／フォルダー／サイト
ユーザーの指定	メール形式。空欄なら全ユーザーとサービス アカウント
複数フィルター	すべての条件に一致する結果だけが出る (AND)

Exchange Online PowerShell に接続して、監査ログを取得することもできます。操作の種類は「操作の一覧」にまとまっており、ドメインの作成・編集・削除も、統合監査ログか Fabric アクティビティ ログで追えます。容量 ID と容量名が監査ログに無い時は、Microsoft Fabric Capacity Metrics アプリで確認します (監査ログには出ないことがあります)。保持期間は Purview の監査のライセンスで決まります。Audit (Standard) は 180 日です。Audit (Premium) でも、Entra ID・Exchange・OneDrive・SharePoint 以外の操作の記録は既定で 180 日で、監査ログの保持ポリシーを作れば最大 1 年まで延ばせます (10 年の保持にはアドオンのライセンスが要ります)。

ウェアハウスの Audit 権限は、ウェアハウスの SQL 監査ログの有効化、構成、照会に使うもので、Purview の監査ログとは別物。OneLake のデータ アクセスの記録は OneLake 診断 (1.1.3)、項目の操作の記録は監査ログ。OneLake の制御プレーンの操作 (セキュリティ ロールの変更、ショートカット、共有など) は、Fabric の監査ログで追います。

場面の判断: 誰がセキュリティ ロールを変更したかを調べたいなら、監査ログです。誰がどのファイルを読んだかを調べたいなら、OneLake 診断です。監査

ログの検索で結果が出ないと言われたら、既定の検索範囲が過去 7 日であること、Audit Logs ロールを持っていることを確かめます。

OneLake セキュリティ (Configure and implement OneLake security) [1.3.8]

OneLake セキュリティは、OneLake のデータ プレーンのセキュリティ。項目に定義したロールで、フォルダー、テーブル、スキーマ、行、列まで絞り、すべての Fabric コンピュート エンジンで一貫して適用します。[^c1-15]

軸	コントロール プレーン	データ プレーン
何を決めるか	項目への操作 (作成、構成、共有、削除)	データそのものへのアクセス (読み書き)
設定の場所	ワークスペース ロール、項目のアクセス許可	項目に定義した OneLake セキュリティ ロール
影響する人	ワークスペース内の全員	主に Viewer、または項目の Read 権限を持つ人

OneLake セキュリティ ロールを作れるのは、ワークスペースの Admin か Member です。ロールが効くのは Viewer ロール、または項目の Read 権限を持つユーザーです。Admin、Member、Contributor は書き込み権限で全データにアクセスできるので、影響を受けません。

レイクハウスには DefaultReader ロールが自動で作られ、ReadAll 権限を持つ全員に Read を与えます。OneLake セキュリティが有効な項目では、ReadAll 単独ではデータ アクセスを与えず、DefaultReader ロール経由で与えます。DefaultReader を削除すると、それに頼っていたユーザーはデータに入れなくなります。編集や削除で、既定のアクセスを絞れます。

Grant ロールだけで Deny ロールは無いので、別のロールやアクセス許可モデルで与えられたアクセスを、OneLake セキュリティで取り消すことはできません。1 人が複数のロールに属すると、アクセスは和集合（最も緩い側）で合成されます。ロール内では OLS と RLS と CLS の積集合です。複数のロールが同じテーブルに RLS を付けると、述語は OR で結合されます。ただし SQL 分析エンドポイントの CLS は厳しい側に倒れ、1 つでも列を隠すロールがあればその列は見えません。

条件・制約	内容
テーブルとして扱われる条件	Tables/ 配下で、_delta_log フォルダー（JSON メタデータ）を持ち、子ショートカットを含まないフォルダー。条件外のフォルダーに RLS/CLS を設定するとアクセスが拒否される
配布リスト	ロールに追加すると、SQL 分析エンドポイントはメンバーを解決できず、ユーザーがメンバーに見えない（Direct Lake on SQL セマンティック モデルも同様）
非スキーマレイクハウス	RLS/CLS で保護されたテーブルのデータプレビューを非サポート。スキーマ対応レイクハウスを使う
安全側の動作	一部のストレージ操作は RLS/CLS を強制できないため、安全にフィルターできないクエリは全体がブロックされる
ReadWrite	項目の Read 権限を持つユーザー（Viewer など）にだけ意味がある。書き込みは OneLake 経由（Spark ノートブック、OneLake ファイル エクスプローラー、API）で行う

条件・制約	内容
テナント設定	「ユーザーは Fabric 外のアプリで OneLake に保存されたデータにアクセスできる」をオフにすると内部アプリ (Spark、Data Engineering、Data Warehouse) のみ。ショートカットで OneLake セキュリティを使うならオンが必要

ロール定義の変更が反映されるまでは約 5 分、ユーザー グループの変更の反映は約 1 時間かかります。エンジンによっては、さらに 1 時間のキャッシュを挟みます。

SQL 分析エンドポイントには 2 つのアクセス モードがあり、新しく作ったエンドポイントは委任 (Delegated) モードで始まります。

対象	ユーザー ID モード	委任された ID モード
認証の主体	サインインしたユーザーの ID を OneLake に渡す	ワークスペースまたは項目の所有者の ID
テーブル	OneLake セキュリティ ロールが制御。SQL の GRANT/REVOKE は使えない (無視される)	SQL の GRANT/REVOKE で全面的に制御
RLS・CLS	OneLake セキュリティ ロールの一部として定義	CREATE SECURITY POLICY と GRANT SELECT (列指定)
動的データ マスク	未サポート	ALTER TABLE の MASKED で定義
ビュー・ストアド プロシージャ・関数	SQL の GRANT で付与できる	同じ

OneLake セキュリティを使うには、Admin か Member がエンドポイントを一度ユーザー ID モードへ切り替えなければなりません。切り替えるとワークスペース内の全 SQL 分析エンドポイントが一時的に使えなくなり、実行中とキューの待ちのクエリは全部キャンセルされます。ユーザー ID モードでは、SQL 分析エンドポイントへ書き込みできず、書き込みはレイクハウス側でワークスペース ロールに従います。

ショートカットの認証モードは 2 つ。パススルー (Passthrough) は、クエリするユーザーの ID でターゲットにアクセスし、OneLake から OneLake へのショートカットの既定です。委任 (Delegated) は、設定した接続の ID や資格情報でアクセスします。外部システム (ADLS、Amazon S3、Dataverse など) へのショートカットは常に委任です。OneLake 間のショートカットには、OneLake セキュリティの権限を直接は定義できません。ショートカットがあるフォルダーの権限と、ターゲット側の権限が組み合わさります。ターゲットが OneLake セキュリティに対応していればそのロール、対応していなければターゲット項目の ReadAll が要ります。委任の OneLake ショートカットで見えるのは、ショートカット側のアクセスと接続 ID の対象側アクセスの積集合。Power BI の Direct Lake over SQL や委任モードの SQL 分析エンドポイントは、所有者の ID を使います。ユーザー ID を最後まで通したいなら、Direct Lake over OneLake を使うか、SQL 分析エンドポイントをユーザー ID モードにします。

場面の判断: 分析チームの Viewer に、レイクハウスの売上テーブルのうち自分の地域の行だけを見せたいなら、Viewer ロールのままで足ります。その行条件を持つ OneLake セキュリティ ロールを作り、ワークスペース ロールは変えません。さらに SQL 分析エンドポイントからも同じ制限を効かせるなら、エンドポイントをユーザー ID モードに切り替えます。DefaultReader を残したままでは、ReadAll のユーザーがレイクハウス全体を読めます。DefaultReader を編集するか削除して、必要なロールだけに入れます。

処理のオーケストレーション(Orchestrate processes) [1.4]

Dataflow Gen2、パイプライン、ノートブックの使い分け(Choose between Dataflow gen 2, a pipeline and a notebook) [1.4.1]

選ぶ軸は、コードを書くか、変換の複雑さ、データ量、チームのスキル、取り込みかオーケストレーションか、の5つです。[^c1-16]

項目	コピー アクティビティ (パイプライン)	Copy ジョブ	Dataflow Gen2	Eventstream	Spark (ノートブック)
主な用途	データレイクとDWHの移行、取り込み、軽量変換	取り込み、増分コピー、レプリケーション	取り込み、変換、データラングリング、プロファイリング	イベントデータの取り込みと変換	取り込み、変換、処理、プロファイリング
コード	ノーコード／ローコード	ノーコード／ローコード	ノーコード／ローコード	ノーコード／ローコード	コードを書く
開発インターフェイス	ウィザードとキャンバス	ウィザードとキャンバス	Power Query	キャンバス	ノートブックとSparkジョブ定義
主なスキル	ETL、SQL、JSON	ETL、SQL、JSON	ETL、M、SQL	SQL、JSON、メッセージング	Spark (Scala、Python、Spark SQL、R)

項目	コピー アクティビティ (パイプライン)	Copy ジョブ	Dataflow Gen2	Eventstream	Spark (ノートブック)
ソース	50 以上のコネクタ	50 以上のコネクタ	150 以上のコネクタ	CDC 対応データベース、Kafka、パブリッシュ/サブスクライブ型メッセージング	数百の Spark ライブラリ
宛先	40 以上のコネクタ	40 以上のコネクタ	レイクハウス、Azure SQL Database、Azure Data Explorer、Azure Synapse Analytics	Eventhous e、レイクハウス、Activator アラート、派生ストリーム、カスタムエンドポイント	数百の Spark ライブラリ
変換の複雑さ	軽量(型変換、列マッピング、ファイルの結合と分割、階層のフラット化)	軽量	低～高。300 以上の変換関数	軽量	低～高。ネイティブ Spark とオープンソース ライブラリ
データ量	低～高	低～高	低～高	中～高	低～高

パイプラインは、取り込みとオーケストレーション (アクティビティの順序制御) の道具です。Copy ジョブの典型は、オンプレミス SQL Server の CDC 有効テーブルを Azure SQL Database へ増分ロードする用途です。ウィザードでスケ

ジュールまで組めるので、コードも複雑なオーケストレーションも要りません（オンプレミス ゲートウェイ対応）。Dataflow Gen2 は、Power Query に慣れた担当者が、低～中程度のデータ量を、ノーコードかローコードで変換して複数の宛先へ出力する場面に向きます。Spark は、大量データを分散処理し、Python や Scala で構造化、半構造化、非構造化データを読み、独自ロジック（テキスト分析など）を書く場面に向きます。

コードで DAG を書くなら Airflow ジョブ (1.1.4)、ノーコードのオーケストレーションならパイプライン。

場面の判断: ビジネス部門の担当者が Power Query で整形し、複数の宛先へ出す場合は Dataflow Gen2 です。数百 GB の JSON を読んで独自のテキスト解析をかける場合は、ノートブックを選びます。オンプレミスの CDC を増分で寄せるだけなら Copy ジョブです。夜間に取り込み、変換、通知を順に動かす全体の流れは、パイプラインが担います。

スケジュールとイベント トリガー (Design and implement schedules and event-based triggers) [1.4.2]

パイプラインの開始方法は 3 つ。オンデマンド実行 (Run)、スケジュール実行、イベント ベース実行 (イベントトリガー) です。既定ではスケジュールは設定されていません。[^c1-17]

スケジュールの設定項目は、スケジュールの種類、頻度、開始と終了の日時、タイムゾーンです。開始日と終了日の両方が必須で、終了日なしの無期限は指定できません。長期に動かすなら、遠い将来 (例: 01/01/2099 12:00 AM) にします。1 つのパイプラインに最大 20 個のスケジュールを持て、頻度や開始と終了をそれぞれ変えられます。

種類	内容
Fixed	通常の固定スケジュール。有効化、無効化、編集ができる
Interval-based (プレビュー)	固定の重ならない間隔で実行する。式ビルダーのトリガー パラメーターに Window start time と Window end time が追加される。有効化、無効化、編集はできず、削除して再作成する。時間スライスの監視とバックフィルは未対応

スケジュールからパラメーターを渡せます。スケジュールで指定するパラメーター名は、パイプラインのパラメーター名と完全に一致する必要があり、不一致は実行時に無視され、値が渡りません。値は直接値か、変数ライブラリから指定します。失敗時のメール通知（ユーザーやグループを追加）はスケジュール実行で設定でき、オンデマンド実行では通知されません。

イベントトリガーは、Fabric のイベントストリームと Data Activator アラートを基盤にします。トリガーは Reflex (Activator) 項目としてワークスペースに作られます。ファイル イベント、ジョブ イベント、ワークスペース イベントでパイプラインを開始できます。

1. パイプラインのホームで「トリガー」を選ぶ
2. イベントの種類を選ぶ (OneLake イベント、Azure Blob イベントなど)
3. ソースを選ぶ。Azure Blob ならサブスクリプションとストレージ アカウントを選ぶと、ワークスペースに eventstream が作られる
4. イベントの種類を選ぶ。ファイルの作成と削除のほかにも多数ある
5. Subject フィールドで、フォルダー名、ファイル名、ファイルの種類、コンテナーを絞る

6. トリガーを保存するワークスペース、パイプライン、アクション、Reflex 項目としての名前を決めて作成する

式ビルダーの「トリガー パラメーター」タブでは、イベントの Subject と Topic から自動で解析されたファイル名とフォルダー名を使えます。たとえば `@pipeline()?.TriggerEvent?.FileName`。手動テストではこの値が NULL になるので、`pipeline()?` のように `?` を付けて NULL を許します。

実行条件 (Run conditions、プレビュー) は、上流のパイプラインの実行、必要なデータ、時間枠の充足を待ってから実行する依存条件です。スケジュールやイベントトリガーの代わりにはなりません。併用します。

場面の判断: 取引先が Blob にファイルを置いた時点で処理を始めたいなら、Azure Blob イベントのトリガーです。毎朝 6 時に動かし、失敗時だけ担当者に知らせたいなら、スケジュールに失敗通知を設定します。手動でテスト実行した式が失敗する場合は、トリガー パラメーターが NULL なので、`?` を付けないと式が評価できません。毎日の増分範囲を取りたいなら、Interval-based の Window start time と Window end time を使います。ただしプレビューで、変更は削除と再作成です。

パラメーターと動的な式によるオーケストレーション (Implement orchestration patterns with notebooks and pipelines, including parameters and dynamic expressions) [1.4.3]

パラメーターと式

パラメーターは、パイプラインに外部の値を渡す仕組みで、実行中は値が変わりません。同じパイプラインを異なる値で再利用できます。パイプライン キャンバスの背景を選び、パラメーター タブで「+ New」から名前、データ型、既定値

を指定します。実行中に値を入れ替えたい時は、変数 (variables) を使い、Set Variable アクティビティで String、Bool、Array 型の値を設定します。[^c1-18]

式は @ で始まり、@ で始まる文字列そのものを使うなら @@ と書きます。文字列補間は @{ ... } で、常に文字列を返します。

書き方	結果 (myNumber = 42、myString = foo)
@pipeline().parameters.myNumber	数値の 42
@{pipeline().parameters.myNumber}	文字列の 42
Answer is: @{pipeline().parameters.myNumber}	文字列 Answer is: 42
First Name: @{pipeline().parameters.firstName}	firstName の値を埋め込んだ文字列
Answer is: @@{pipeline().parameters.myNumber}	文字列 Answer is: @{pipeline().parameters.myNumber}

参照の形は 3 つ。パラメーターは @pipeline().parameters.<名前>、アクティビティの出力は @activity('名前').output.<サブフィールド>、変数は @variables('名前') です。サブフィールドがパラメーターで決まる時の書き方は、ドットではなく [] 記法。

```
@concat('raw/', formatDateTime(utcNow(), 'yyyy/MM/dd'), '/',  
pipeline().parameters.fileName)
```

2026-10-26 に fileName が orders.csv で実行されると、raw/2026/10/26/orders.csv になります。utcNow() は UTC の時刻を返します。システム変数には、@pipeline().RunId (実行 ID) と @pipeline().TriggerName

(トリガー名)があります。`@pipeline().TriggerTime` は、トリガーが実際に起動した時刻です。日時系は UTC の ISO 8601 形式。Execute Pipeline から起動された時の親パイプライン名と実行ID (`@pipeline()?.TriggeredByPipelineName` など) は、それ以外では Null になります。そのため `?` なしでは参照できません。接続のパラメーター化には、接続の GUID (Connection ID) を文字列パラメーターに入れます。

Dataflow Gen2 の公開パラメーター (実行時に外から値を渡す機能)。前提として、パラメーターを先に Dataflow の中に作っておきます。そのうえで、Dataflow のオプション > Parameters で、実行のためのパラメーターの検出とオーバーライドを有効にします。すると、Fabric REST API や Fabric の体験から実行時に値を渡せます。必須パラメーターは値がないと実行が失敗し、任意のパラメーターは現在値が使われます。

制御フローのアクティビティ

アクティビティ	役割と注意
ForEach	一覧を繰り返す。Sequential (順次) か並列かを選び、並列なら Batch count で同時処理の最大数を指定する。Items にはカンマ区切りの一覧、パラメーター、動的コンテンツを使う。子アクティビティは、項目ごとに順番に処理される
If Condition	真偽の式で、True と False のどちらかの枝を実行する
Switch	式の値を case に突き合わせて実行する。一致がなければ Default の case
Until	条件が true になるまで繰り返す。内側のアクティビティが失敗しても止まらず、タイムアウトを指定できる

アクティビティ	役割と注意
Lookup	最大 5,000 行(超過分は先頭 5,000 行)、出力は最大 4 MB(超えると失敗)、タイムアウトは最長 24 時間
Get Metadata	データのメタデータ(列数、exists、構造など)を取り、条件式で検証に使う
Set Variable	パイプライン変数やパイプラインの戻り値を設定する

典型的な組み合わせは、Lookup で一覧を取り、ForEach で 1 件ずつ Copy を動かす形。Lookup の 5,000 行の上限を超える一覧は、先頭の 5,000 行で切れます。

アクティビティの依存関係には、UponSuccess (成功時)、UponFailure (失敗時)、UponCompletion (完了時)、UponSkip (スキップ時) の 4 つ。失敗時の通知は UponFailure の枝に置き、後始末は UponCompletion の枝に置きます。アクティビティの全般設定では、タイムアウトの既定が 12 時間、再試行の間隔の既定が 30 秒(回数の既定は 1)で、再試行を有効にして回数を指定できます。再試行 3 回なら、最初の実行と合わせて最大 4 回動き、各再試行の前に間隔(再試行と再試行の間に待つ秒数)の分だけ待つ、という数え方です。タイムアウトの最大は、Activity overview が 7 日、パイプラインの制限の表が 24 時間と書き、文書によって値が違います。1 パイプラインに置けるアクティビティは既定で 120 個を超えられません。

パイプラインからノートブックを呼ぶ

ノートブック アクティビティは、接続(認証方法)、ノートブック、ベース パラメーターを指定してノートブックを実行します。ノートブック側でパラメーターにしたいセルに「Mark cell as parameters」を設定します。そのセルの変数が既定

値となり、パイプラインのベース パラメーターや親ノートブックの引数で上書きされます。ベース パラメーターの自動入力が検出するのは、このセルの変数だけです。パラメーター セルが無ければ何も入りません。

セッション タグ(任意の文字列)を設定すると、既存の Spark セッションを再利用して起動を短縮できます。利用には、ワークスペースの高同時実行モードの「パイプラインでの複数ノートブック実行」オプションをオンにします。ワークスペース ID で実行する条件は 1.3.1 のとおりです。ノートブック アクティビティは、リージョンをまたぐ呼び出しができません。

SAMPLE

ノートブックから別のノートブックを呼ぶ

1 本を実行して終了値を受け取る

```
result = notebookutils.notebook.run("NB_clean", 90, {"run_date": "2026-10-26"})
```

依存順序つきで並列実行 (DAG)

```
dag = {  
  "activities": [  
    {"name": "load_raw", "path": "NB_load_raw", "args": {"run_date":  
"2026-10-26"}},  
    {"name": "clean", "path": "NB_clean", "dependencies":  
["load_raw"]},  
    {"name": "dim", "path": "NB_dim", "dependencies": ["clean"]},  
    {"name": "fact", "path": "NB_fact", "dependencies": ["clean"]},  
    {"name": "publish", "path": "NB_publish", "dependencies": ["dim",  
"fact"]}  
  ],  
  "concurrency": 2,  
  "timeoutInSeconds": 43200  
}  
notebookutils.notebook.runMultiple(dag)
```

この DAG は load_raw、clean、dim と fact の同時実行、publish の順に動きます。

項目	run	runMultiple
形	<code>run(path, timeout_seconds=90, arguments=None, workspace="")</code>	<code>runMultiple(dag, config=None)</code> 。config は Python のみ
戻り値	子の exit 値 (文字列)。exit が無ければ空文字列	活動名をキーにした辞書。exitVal と exception を持つ
並列	しない (1 本ずつ)	する。依存順序は dependencies で指定
既定値	timeout_seconds は 90	concurrency は CPU コア数の 3 倍 (0 は無制限)、DAG 全体の timeoutInSeconds は 43200 (12 時間)、timeoutPerCellInSeconds は 90

runMultiple には retry と retryIntervallInSeconds も指定できます。クロスワークスペースの参照は、run では workspace 引数にワークスペース ID を、runMultiple では各活動の workspace に名前か ID を指定します (ランタイム 1.2 以上)。exit() は文字列だけを返し、try-catch の中で呼ぶと効きません。子ノートブックの既定レイクハウスが親と違うと、実行がブロックされます。回避には引数で useRootDefaultLakehouse: True を指定します。

パイプラインからパイプラインを呼ぶ

Invoke pipeline には新旧があります。Legacy は同じワークスペースの Fabric パイプラインだけを呼べ、監視できるのは親だけ。新しい Invoke

pipeline は、ワークスペースをまたぐ呼び出しと、ADF や Synapse のパイプラインの呼び出しができ、子も監視できます。

場面の判断: 毎日 200 のテーブルを同じ手順で取り込むなら、Lookup でテーブル名の一覧を取ります。ForEach を並列にして Batch count を決め、Copy にテーブル名を `@item()` で渡す構成です。一覧が 5,000 行を超える場合、Lookup は先頭の 5,000 行しか返しません。分割して取ります。コピーが失敗した時だけ通知を出すなら、Copy の UponFailure に通知を置きます。ノートブックの結果を次の工程の分岐に使うなら、子ノートブックで exit に値を渡し、パイプラインか親ノートブックで受けます。

サンプルはここまでです

続き(第2章～巻末)は、DP-700 問題集のご購入で、頻出度別ノート・暗記用ノートと合わせて3点すべてダウンロードできます。

SAMPLE

DP-700 学習用テキスト(無料サンプル)

版

2026年10月版(DP-700 出題範囲(Skills measured)2026-10-19版 準拠)

発行

IT資格道場(<https://www.shikaku-doujou.com>)

お問い合わせ

info@shikaku-doujou.com

Microsoft、Microsoft Fabric、Azure および Microsoft の製品名・サービス名は、Microsoft Corporation の米国およびその他の国における商標または登録商標です。記載されているその他の会社名・製品名・サービス名は、各社の商標または登録商標です。

本教材は公式シラバスの内容を基に作成したもので、Microsoft Corporation との提携・承認関係はありません。

本文・図表・設問はすべてオリジナルの著作物です。無断での転載・複製・再配布・二次販売はできません。お気づきの点は info@shikaku-doujou.com までご連絡ください。