

データモデル

1. 共通事項

データ管理の基本方針

年月の管理

- 年月はDATE型で管理し、月の1日を固定値として使用します
- 例: 2024年4月の場合は `2024-04-01` として保存
- これにより以下が容易になります：
 - 年月の範囲検索
 - 月次の集計
 - 会計年度ごとの集計
 - 前月・翌月の計算
- 売上計画と原価計画で同じ形式を使用することで、期間の突合が容易になります

NULL値の扱い

- 月別売上計画の売上計画金額、月別原価計画の人工と原価金額はNULL許容とします
- NULL値は「未入力」を表し、0は「計画値がゼロ」を表します
- これにより、未入力と計画値ゼロを区別して管理できます
- 集計時はNULL値を0として扱います

人工の管理

- 人工は月数単位で管理します（例: 0.5ヶ月、1.0ヶ月）
- 労務費の原価金額は、人工（月数）× 労務費レート（千円/月）で計算します

2. 組織構造

概要

要件定義の「[データ構造要件 > 採算管理群の階層構造](#)」に基づき、以下の制約でデータモデルを設計しています：

- 「採算管理郡」と「プロジェクト群」と「プロジェクト」を「組織」に統合

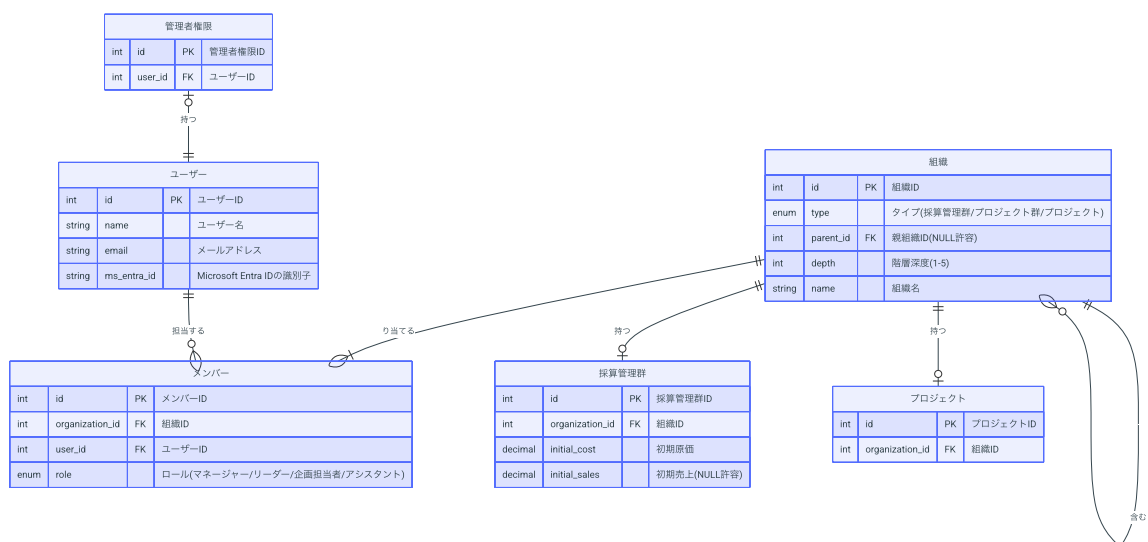
- ・「採算管理郡」の「階層深度」は1(トップレベル)固定
- ・「プロジェクト群」の親は「採算管理郡」または「プロジェクト群」
- ・「プロジェクト」の親は「採算管理郡」または「プロジェクト群」
- ・採算管理郡タイプの「組織」は必ず「採算管理郡」を持つ
- ・プロジェクトタイプの「組織」は必ず「プロジェクト」を持つ

制約

階層構造の制約

- ・採算管理郡の階層深度は常に1（トップレベル）
- ・プロジェクト群の階層深度は2-4（親の深度+1）
- ・プロジェクトの階層深度は2-5（親の深度+1）
- ・プロジェクトは必ずリーフノード（子を持たない）

データモデル図



3. 売上計画

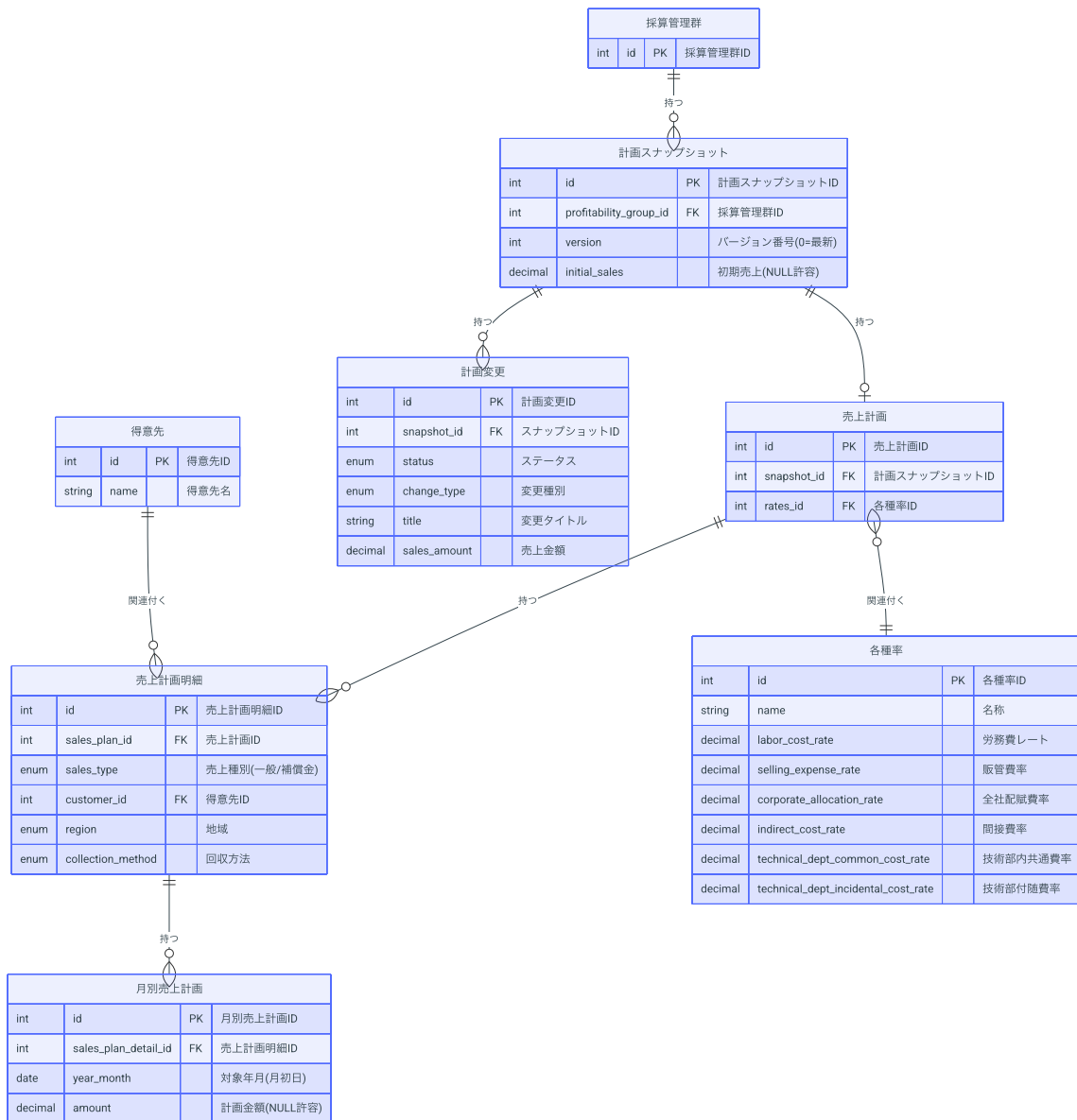
概要

要件定義の「[データ構造要件 > 計画データの構造](#)」に基づき、以下の要件でデータモデルを設計しています：

- ・売上計画は採算管理群単位で管理
- ・得意先別、地域別、回収方法別の明細管理に対応

- 年度・月別の計画金額管理を実現
- 履歴管理（設計変更、中止減額、補償金額）に対応

データモデル図



管理方針

売上の管理

- 売上総額 = 採算管理群の初期売上 + 承認済みの売上計画変更の売上金額の合計値
- 計画値合計 = 全売上計画明細の計画金額の合計値
- 計画値合計は売上総額と一致するように入力する

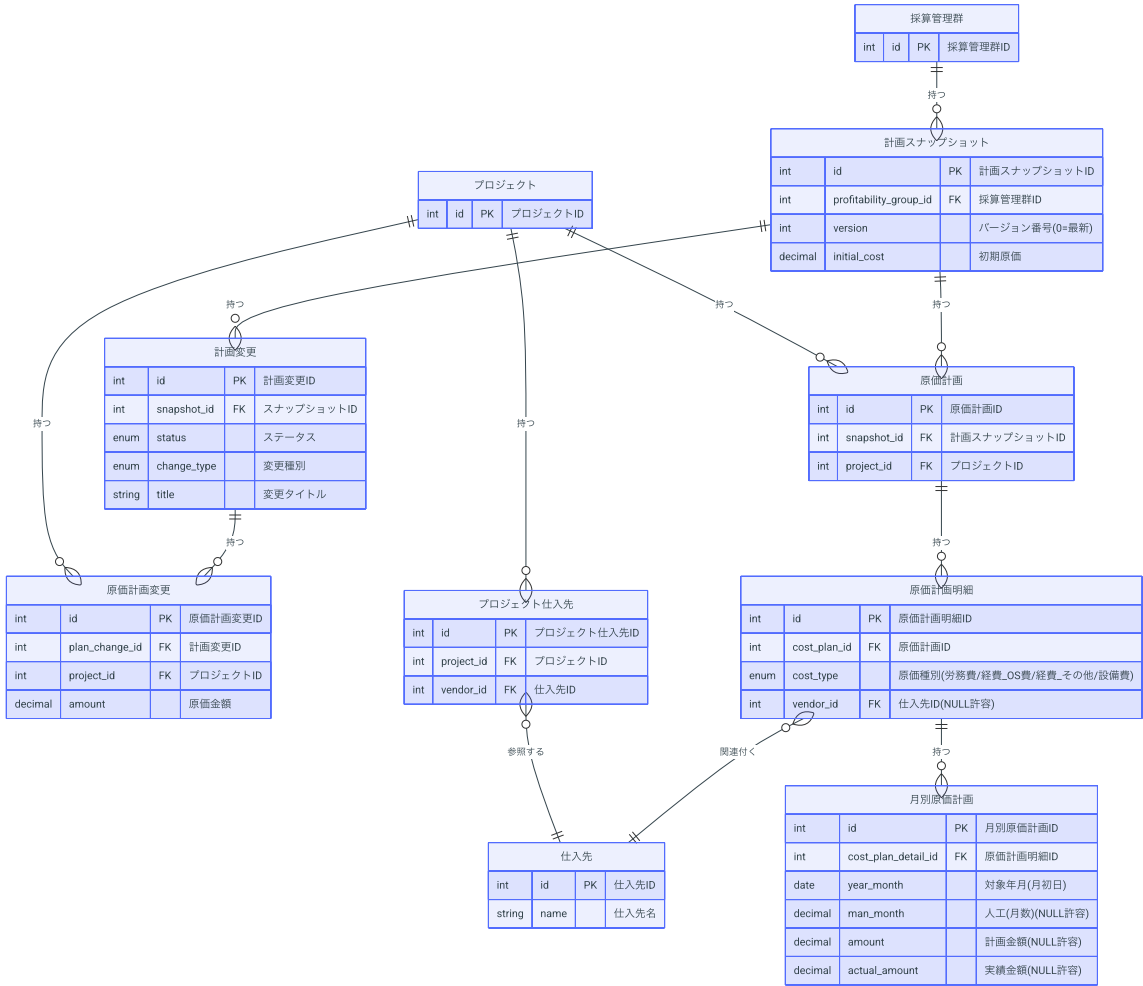
4. 原価計画

概要

以下の要件でデータモデルを設計しています。詳細はデータ構造要件 > 原価計画を参照してください：

- 原価計画はプロジェクト単位で管理
- 原価種別（労務費/経費_OS費/経費_その他/設備費）ごとの管理
- 年度・月別の計画金額管理を実現
- 履歴管理（追加予算、設計変更、中止減額）に対応

データモデル図



管理方針

原価の管理

- 開発費総額 = 採算管理群の初期原価 + 承認済みの計画変更の原価金額の合計値
- 原価計画明細の計画金額 = 関連する月別原価計画の計画金額の合計値
- 計画値合計は開発費総額を超過しないことが望ましい
- 計画値合計が開発費総額を超過した場合は、プロジェクトマネージャーに通知する仕組みを検討
- 実績値はSCOPEから定期的に取得し、月別原価計画の実績金額に設定します
 - 設備費の実績値はSCOPEから取得できないため、システムに入力する必要があります

仕入先の管理

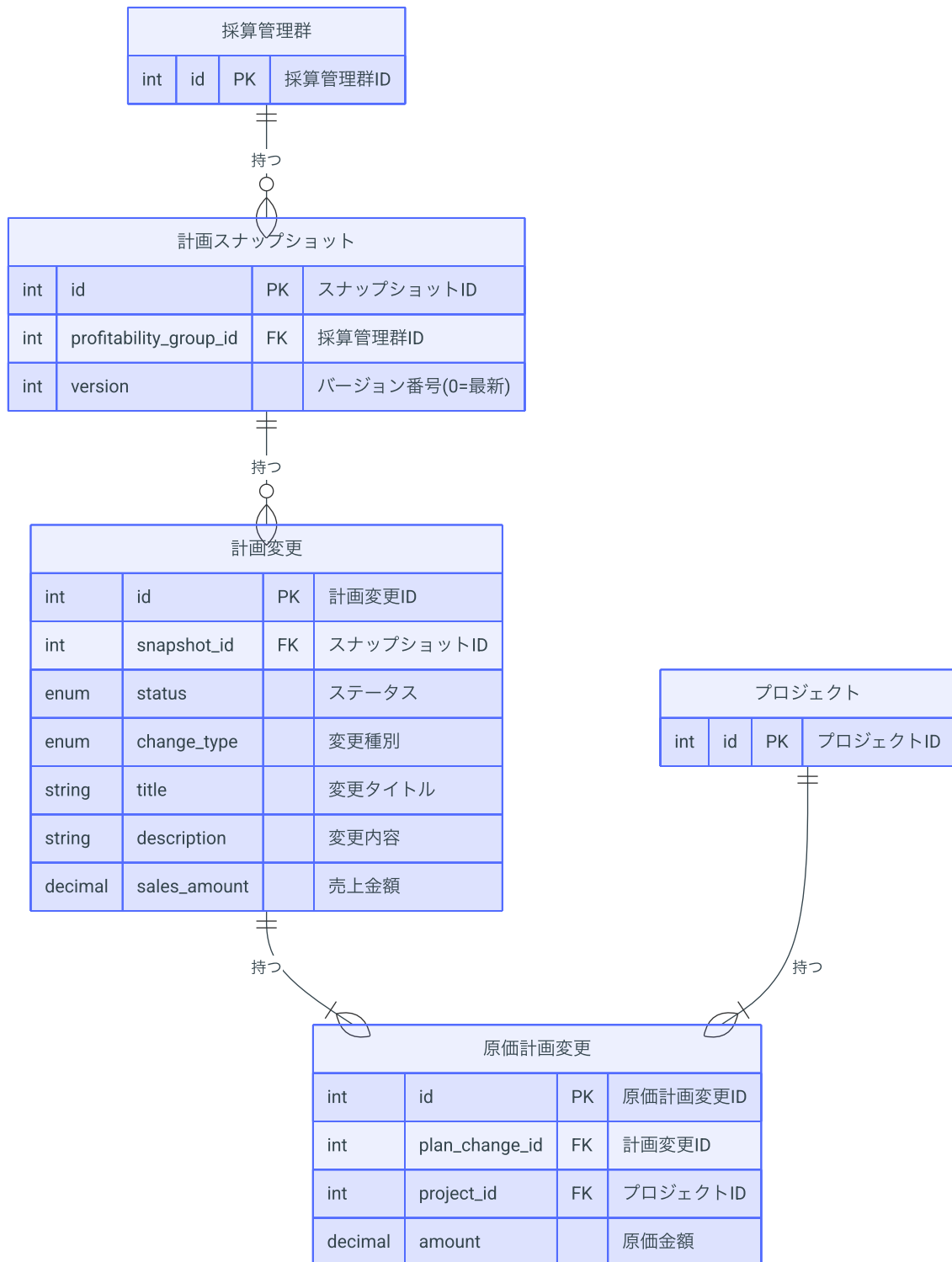
- プロジェクトは複数の仕入先と関連付けることができます（プロジェクト仕入先）
- 原価計画明細は直接仕入先と関連付けることができます
- 経費_OS費の原価計画明細には、必ず仕入先を設定する必要があります
- その他の原価種別の場合、仕入先の設定は任意です
- 仕入先との関連付けにより、以下が可能になります：
 - 仕入先ごとの原価計画の集計
 - 仕入先ごとの実績管理
 - プロジェクト全体での仕入先の把握

5. 計画変更

概要

売上計画と原価計画の変更を統合的に管理します。変更のステータスを管理し、承認された変更は計画スナップショットとして記録します。

データモデル図



管理方針

計画変更の金額

- 計画変更の売上金額はモデルをシンプルにするためエンティティを分けずに計画変更の売上金額として管理します

- ・ 計画変更の原価金額はプロジェクト毎に管理するため別エンティティとして管理します

ステータス管理

- ・ 計画変更は以下のステータスを管理します：
 - ・ 見積中：変更内容の検討段階
 - ・ 顧客承認：変更内容が承認され、計画に反映される段階
 - ・ 中止：変更が中止された段階
- ・ 「顧客承認」以外のステータスの金額は、実際の計画金額の計算には使用しません
- ・ 承認された変更は必ずスナップショットとして記録します

変更種別

計画変更は以下の種別を持ちます：

- ・ 設計変更：仕様や設計の変更に伴う計画の変更
- ・ 中止減額：プロジェクトの中止や規模縮小に伴う減額
- ・ 補償金額：補償金に関する変更
- ・ 追加予算：予算の追加による変更

金額の管理

- ・ 売上計画変更は採算管理群単位で金額を管理します
- ・ 原価計画変更はプロジェクト単位で金額を管理します
- ・ 計画変更の総額（売上・原価）は明細の合計値として算出します

データの整合性

- ・ 原価計画変更は必ずプロジェクトと紐づきます
- ・ 承認された計画変更は必ずスナップショットと紐づきます

履歴の保持

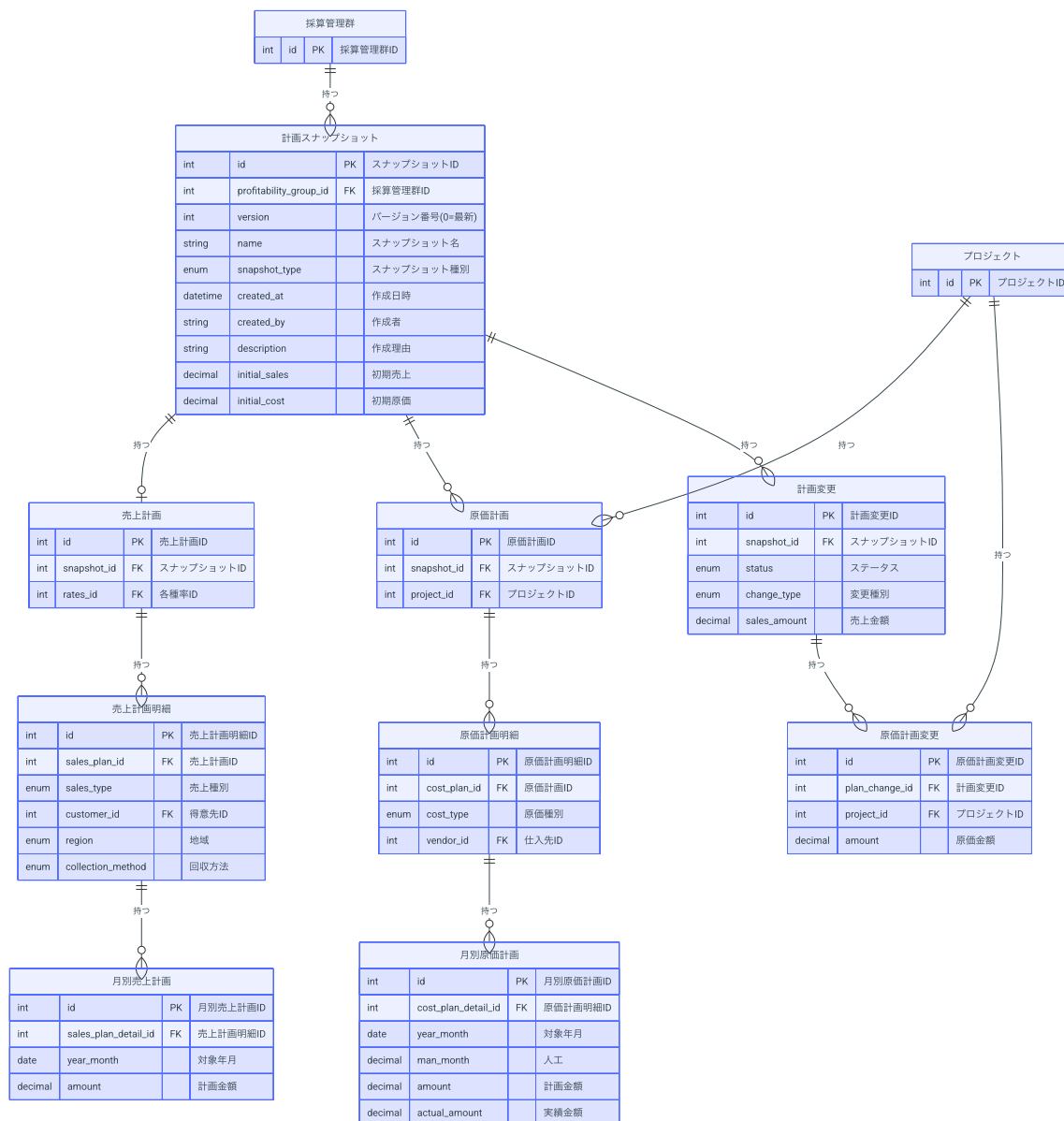
- ・ 中止された変更も含めて全ての計画変更を保持します
- ・ 変更履歴は計画変更として永続的に保持します
- ・ 承認された変更はスナップショットとしても保持されます

6. 計画スナップショット

概要

採算管理群の売上計画と原価計画の状態を任意のタイミングで保存し、後から参照できるようにします。バージョン管理方式を採用し、最新データ（バージョン=0）と過去のスナップショットを同一テーブルで管理します。

データモデル図



管理方針

バージョン管理

- バージョン=0を最新データとして扱います
- スナップショット作成時は、バージョン番号を1から連番で採番します
- 最新データの更新時は、必要に応じてスナップショットを作成します

データの完全性

- スナップショット作成時は、関連する全てのデータを同時に保存します
- 下記のマスタデータは常に最新のデータを参照します：
 - プロジェクト
 - 各種率
 - 仕入先
 - 得意先

検討事項

スナップショット種別

分析のためにスナップショット種別を定義するか要検討：

- 最新データ
- 初期計画確定
- 設計変更
- プロジェクト中止
- その他

各種率のスナップショット

- 各種率は利益率の計算に使用するため、売上計画の項目として保存するか要検討

代替案：スナップショット専用テーブル方式

最新データと過去のスナップショットを別テーブルで管理する方式：

メリット：

- 最新データと過去データの分離が明確
- スナップショットデータの構造変更が容易
- 最新データへのアクセス性能が高い

デメリット：

- テーブル数が多くなる
- データの2重管理による整合性リスク

この方式は以下の場合に適しています：

- 最新データと過去データの分離が重要な場合

- スナップショットの構造を頻繁に変更する必要がある場合
- 最新データへのアクセスが非常に頻繁な場合

ただし、本システムでは以下の理由から採用しません：

- データの整合性を重視
- シンプルな構造による保守性を重視
- 想定される検索パターンで十分なパフォーマンスを確保可能
- 将来的な機能拡張への対応を考慮