

解答例

テーブル名/ビュー名 (論理名)	経費情報明細
テーブル名/ビュー名 (物理名)	TT_EXPENSES_DETAIL

No.	項目名 (物理名)	項目名 (論理名)	PK	NOT NULL	データ型	桁数	ユニーク キー	ユニーク キー	初期値	内容
1	EXPENSE_APPLY_NO	経費申請No	P	NOT NULL	NUMBER	8				
2	EXPENSE_CODE	経費コード	P	NOT NULL	NVARCHAR2	12				
3	AMMOUNT	金額		NOT NULL	NUMBER	8				

↑桁数については、大抵は余裕を持たせるため、上記以上になっていれば正答。

桁数検討について (判断基準例)			
◆経費申請Noの考慮 (発生量)			
1年	100万件	1,000,000件	7桁
30年	3000万件	30,000,000件	8桁
◆金額の考慮			
	1万円	10,000円	5桁
	100万円	1,000,000円	7桁
	1000万円	10,000,000円	8桁

PK検討について	
演習難易度の都合上、「経費明細No」のようなものは用意していないが、 (1つの申請上で、同一経費を合算で申請させたくないなどの要件はないこととしているが) もし「経費明細No」のようなものを用意するのであれば、以下ようになる。	
項目名 (論理名)	PK
経費申請No	P
経費明細No	P
経費コード	
金額	

参考：ワンステップアップ（よく使用される項目）

標準管理項目として、すべてのテーブルに、以下のような項目を用意するルールがあります。

No.	項目名（物理名）	項目名（論理名）	PK	NOT NULL	データ型	桁数	ユニークキー	ユニークキー	初期値	内容
-	DEL_FLG	削除フラグ		NOT NULL	NUMBER	1			0	
-	CREATE_BY	作成者			NVARCHAR2	36				
-	CREATE_TIME	作成日時			TIMESTAMP					
-	UPDATE_BY	更新者			NVARCHAR2	36				
-	UPDATE_TIME	更新日時			TIMESTAMP					
-	VERSION	バージョン			NUMBER	5,0			1	

標準管理項目	
削除フラグ	「データの削除」を行った際、物理的には削除させたくないケースがあり、その場合に使用します。
作成者～更新日時	最後にデータを更新したのは誰なのか？など、データの「追跡性」などのため、管理情報として保存しておきます。
バージョン	データの更新時、「排他制御」を行いたいケースがありますが、楽観排他方式による排他を行うケースなどで使用します。 楽観排他の利用例：「対象のデータは他のユーザによって更新されました。再度画面を表示し、確認の上、改めて更新を行ってください」 (ユーザAとユーザBがバージョン1のデータを参照し、ユーザAが更新⇒バージョン2になったとき、他のユーザ（ユーザB）の更新を許可しない。 UPDATE文のWHERE句に「データ取得時のバージョン」を加えることで、他の人が触っていないデータしか更新ができなくなる仕組み)

参考：ワンステップアップ（よく使用される項目）

例えば、組織マスタや、製品マスタなどは、情報を履歴で持たせたいことがあるため、有効開始・有効終了の項目を用意することもあります。

No.	項目名（物理名）	項目名（論理名）	PK	NOT NULL	データ型	桁数	ユニークキー	ユニークキー	初期値	内容
-	VALID_START_DT	有効開始日時		NOT NULL	TIMESTAMP				1900/1/1	有効開始日時
-	VALID_END_DT	有効終了日時		NOT NULL	TIMESTAMP				2999/12/31	有効終了日時