---

Projet: "Usine à Tests"

Date: "2508140422"

Version: "9.0"

---

### 1. Résumé des Échanges du Jour

La session d'aujourd'hui a transformé la structure de données naissante de la composition des e-mails en une architecture robuste, multilingue et prête pour des intégrations futures complexes comme l'IA.

[cite\_start]Partant de la création de la table `sys\_Composition\_Emails`[cite: 43], nous avons immédiatement identifié et corrigé des points cruciaux :

1. \*\*Gestion Multilingue :\*\* Vous avez soulevé une omission critique : l'absence de gestion de la langue. La solution a été d'intégrer une colonne `Code\_Langue` dans la table `sys\_Composition\_Emails`, la rendant capable de gérer plusieurs langues pour un même test.

2. \*\*Stratégie de Numérotation :\*\* Nous avons validé votre proposition d'utiliser des plages de numéros dans la colonne `Ordre` (ex: 1-19 pour les textes, 100+ pour les documents) comme une convention de travail explicite, améliorant la lisibilité et la flexibilité sans violer le principe de "Zéro Convention Implicite".

3. \*\*Intégration de l'Intelligence Artificielle :\*\* Anticipant une évolution majeure, nous avons conçu une architecture complète pour intégrer du contenu généré par IA de manière stable et traçable. La décision clé est de \*\*découpler la génération de la restitution\*\* :

\* Une nouvelle table, `log\_AI\_Generations`, stockera de manière permanente tout contenu (texte ou ID de document) généré par l'IA pour un utilisateur, évitant ainsi les appels multiples et les réponses inconsistantes.

\* La table `sys\_Composition\_Emails` utilisera des "placeholders" (ex: `{{AI:Texte}}`) pour indiquer au script où insérer ce contenu pré-généré et stocké.

### 2. État Actuel du Projet

Le projet a solidifié son architecture de restitution, la rendant significativement plus puissante et pérenne.

\* \*\*Tâches réalisées (Conception & Structuration) :\*\*

\* Création de la structure de base de la table `sys\_Composition\_Emails` dans le Google Sheets.

\* Ajout de la dimension multilingue (`Code\_Langue`), comblant une lacune majeure.

\* Conception d'une architecture complète et sécurisée pour l'intégration future de contenu généré par IA (`log\_AI\_Generations` et le pattern de découplage).

\* Validation d'une convention de nommage numérique pour la colonne `Ordre`.

\* \*\*Tâches en cours :\*\* Peuplement de la table `sys\_Composition\_Emails` avec les données d'un premier test pilote ("Couleurs") pour valider la nouvelle structure.

\* \*\*Prochaines étapes planifiées :\*\*

1. Mettre à jour la feuille `sys\_Composition\_Emails` en ajoutant la nouvelle colonne `Code\_Langue`.

2. Créer la nouvelle feuille `log\_AI\_Generations` avec la structure définie.

3. Poursuivre le peuplement des données du test pilote "Couleurs".

4. Développer le script qui lira ces nouvelles structures de données.

### 3. Contexte Historique du Projet

[cite\_start]Partant d'un système fonctionnel mais rigide, le projet a évolué vers une plateforme de création de tests entièrement pilotée par les données[cite: 78]. [cite\_start]Après avoir universalisé le calcul des scores [cite: 146] [cite\_start]et la composition des e-mails[cite: 67], la session actuelle a enrichi cette architecture en y ajoutant deux dimensions stratégiques : le \*\*support multilingue natif\*\* et une \*\*passerelle d'avenir pour l'IA\*\*, assurant que la plateforme reste évolutive.

### 4. Orientations Stratégiques

[cite\_start]La session renforce les principes directeurs du projet : \*\*"La Base de Données est la Source Unique de Vérité"\*\* et \*\*"Zéro Convention Implicite"\*\*[cite: 82].

\* L'ajout de `Code\_Langue` et la création de `log\_AI\_Generations` sont des applications directes du premier principe, déplaçant encore plus de logique métier dans la donnée plutôt que dans le code.

\* La discussion sur la colonne `Ordre` a clarifié comment créer une \*\*convention de travail explicite\*\* qui améliore l'efficacité sans introduire de logique cachée.

\* Le "Journal des Décisions d'Architecture" s'enrichit d'un nouveau principe majeur :

\* \*\*Découplage Stabilité/Innovation (Pattern IA) :\*\* "Pour intégrer une fonctionnalité non déterministe ou volatile (comme l'IA), on la découple en deux processus. Une phase de 'Génération' se produit une seule fois et stocke le résultat stable dans un log. Une phase de 'Restitution' lit ce résultat stable. Le système principal n'appelle jamais directement le service instable, garantissant performance et cohérence."

### 5. Architecture Technique & Ressources Clés

L'architecture de données a été finalisée pour la composition des e-mails et étendue pour les futures fonctionnalités.

#### 5.1. État des Scripts et Fichiers

Aucun script n'a été modifié durant cette session. Toutes les décisions ont porté sur l'architecture des données qui précède le développement.

#### 5.2. Architecture Détaillée et Traçabilité

Fichier : \*\*`[BDD] Tests & Profils`\*\*

\* \*\*MODIFIÉ PROFONDÉMENT :\*\*

\* L'onglet \*\*`sys\_Composition\_Emails`\*\* (nouvellement créé) a été enrichi d'une colonne critique (`Code\_Langue`) pour le rendre multilingue.

\* \*\*AJOUTÉ (Conception) :\*\*

\* Un nouvel onglet \*\*`log\_AI\_Generations`\*\* a été conçu pour stocker les résultats des appels à l'IA.

#### 5.3. Nomenclature des Variables

\* \*\*`Code\_Langue` (String) :\*\*

\* \*\*Origine :\*\* Colonne de l'onglet `sys\_Composition\_Emails`.

\* \*\*Rôle :\*\* Spécifie la langue (ex: "FR", "EN") de la brique de contenu. Agit comme une clé pour récupérer le contenu approprié.

\* \*\*`{{AI:Texte}}` / `{{AI:Document}}` (String) :\*\*

\* \*\*Origine :\*\* Valeur spéciale dans la colonne `Contenu / ID\_Document` de l'onglet `sys\_Composition\_Emails`.

\* \*\*Rôle :\*\* Agit comme un "placeholder" ou une instruction. Indique au script de ne pas utiliser cette valeur directement, mais d'aller chercher le contenu correspondant dans la table `log\_AI\_Generations`.

\* \*\*Colonnes du log IA :\*\*

\* `ID\_Generation` (String) : Identifiant unique de la transaction IA.

\* `ID\_Repondant` (String) : Identifiant de l'utilisateur concerné.

\* `Type\_Contenu` (String) : 'Texte' ou 'Document'.

\* `Contenu\_Texte` (String) : Le texte généré par l'IA.

\* `ID\_Document` (String) : L'ID du fichier Google Drive créé par le processus IA.

#### 5.4. Structure des Données (Feuilles et Colonnes)

\* \*\*Onglet à mettre à jour : `sys\_Composition\_Emails`\*\*

\* \*\*Colonnes critiques :\*\* `Type\_Test`, `Code\_Langue`, `Code\_Niveau\_Email`, `Code\_Profil`, `Element`, `Ordre`, `Contenu / ID\_Document`.

\* \*\*Nouvel Onglet à créer : `log\_AI\_Generations`\*\*

\* \*\*Colonnes critiques :\*\* `ID\_Generation`, `ID\_Repondant`, `Type\_Test`, `Profil\_Obtenu`, `Timestamp\_Generation`, `Type\_Contenu`, `Contenu\_Texte`, `ID\_Document`.

### 6. Actions Recommandées / Prochaines Étapes

1. \*\*Action Immédiate (Pour vous) :\*\* Dans le Google Sheets `[BDD] Tests & Profils` :

\* a. Ajouter la colonne \*\*`Code\_Langue`\*\* à l'onglet `sys\_Composition\_Emails`.

\* b. Créer le nouvel onglet \*\*`log\_AI\_Generations`\*\* avec les colonnes définies en 5.4.

2. \*\*Action Suivante (Pour vous) :\*\* Poursuivre la "traduction" du test "Couleurs" dans la table `sys\_Composition\_Emails` (maintenant multilingue) pour avoir un cas pratique complet.

3. \*\*Question pour notre prochaine session :\*\* Une fois ces structures finalisées et un cas pratique peuplé, la question sera : \*\*Par quel bout du développement du script souhaitez-vous commencer ? La lecture et l'assemblage de la logique multilingue de `sys\_Composition\_Emails` ou l'ébauche du script de la Phase 1 (génération et stockage IA) ?\*\*