Absolument. Voici la synthèse de notre session, prête à être sauvegardée. Elle formalise notre discussion et servira de base solide pour la prochaine étape de développement.

Markdown

---

Projet: "Usine à Tests"

Date: "2508131824"

Version: "8.0"

---

### 1. Résumé des Échanges du Jour

La session d'aujourd'hui a été consacrée à l'architecture de la composition des e-mails de résultats, transformant une idée conceptuelle en un plan technique détaillé et robuste.

Partant de votre proposition d'un système d'e-mails modulaire à plusieurs niveaux (N1, N2, N3), nous avons abordé et résolu plusieurs points critiques :

1. \*\*Gestion du Contenu Spécifique :\*\* Vous avez identifié une faille dans notre modèle initial : comment gérer le contenu (textes, documents) spécifique à un profil de résultat particulier (ex: INTP vs ESFJ). La solution adoptée est l'ajout d'une colonne `Code\_Profil` dans la table de composition des e-mails. Si cette colonne est vide, le contenu est générique ; si elle est remplie, le contenu n'est inclus que si le répondant obtient ce profil précis.

2. \*\*Viabilité à Long Terme :\*\* Nous avons confirmé que l'architecture de "pilotage par les données" que nous concevons n'est pas une simple convention, mais le cœur de la philosophie des plateformes no-code/low-code comme Bubble ou Airtable. Le travail de structuration actuel est donc une préparation idéale pour une future migration.

3. \*\*Stratégie de Déploiement :\*\* Face à la problématique de la coexistence des tests existants ("legacy") avec le nouveau système, nous avons validé une \*\*stratégie de migration progressive\*\*. Plutôt qu'un changement brutal, nous allons implémenter un "routeur" dans le code (piloté par une nouvelle colonne `Moteur\_Calcul`) qui dirigera chaque test vers l'ancien ou le nouveau moteur de calcul, garantissant ainsi qu'aucune fonctionnalité en production ne sera interrompue.

4. \*\*Centralisation de la Configuration :\*\* Nous avons clarifié que la nouvelle table `sys\_Tests\_Config` n'est pas une redondance, mais l'\*\*évolution directe\*\* de votre onglet "paramètres généraux" actuel. Elle centralisera toutes les règles, y compris la gestion des destinataires et des délais d'envoi, qui seront éditées via l'interface HTML existante.

### 2. État Actuel du Projet

Le projet a franchi une étape majeure de conception architecturale, passant de la logique de calcul à la logique de restitution des résultats.

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

\* Conception d'une architecture complète et scalable pour la composition modulaire des e-mails (`sys\_Composition\_Emails`).

\* Résolution du problème de la gestion de contenu spécifique aux profils (`Code\_Profil`).

\* Validation de l'alignement de l'architecture avec les standards no-code et la vision à long terme (Bubble).

\* Définition d'une stratégie de déploiement et de migration sécurisée ("pattern" du routeur).

\* Définition de la structure cible de l'onglet de configuration principal (`sys\_Tests\_Config`).

\* \*\*Tâches en cours :\*\* Aucune. Le projet est en attente de la mise en œuvre de la nouvelle architecture de données.

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

1. Créer la nouvelle structure de données dans le fichier Google Sheets `[BDD] Tests & Profils`.

2. Peupler les nouvelles tables avec les données d'un premier test "legacy" à migrer.

3. Développer le script du "routeur" et la nouvelle logique d'assemblage d'e-mails.

### 3. Contexte Historique du Projet

Partant d'un système fonctionnel mais rigide, le projet a évolué vers une plateforme de création de tests entièrement pilotée par les données. Après avoir rendu le calcul des scores universel (session précédente), la session actuelle s'est attaquée à la "dernière frontière" du code en dur : la composition et la restitution des résultats par e-mail. Cette étape est l'aboutissement logique de la vision "Usine à Tests", où toute la complexité réside dans les données de configuration, et non dans le code.

### 4. Orientations Stratégiques

La session renforce de manière spectaculaire les principes directeurs du projet, notamment \*\*"La Base de Données est la Source Unique de Vérité"\*\* et \*\*"Zéro Convention Implicite"\*\*.

\* La décision d'intégrer la gestion des profils spécifiques via une colonne `Code\_Profil` plutôt que de créer un mécanisme parallèle est une application directe de ces principes.

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

\* \*\*Migration Progressive par Routeur :\*\* "Face à la nécessité de faire évoluer un système en production, on ne remplace pas l'ancien système d'un coup. On introduit un 'routeur' logiciel qui permet aux deux systèmes (legacy et nouveau) de coexister. La bascule se fait ensuite de manière unitaire, par configuration, garantissant stabilité et contrôle."

La discussion sur la compatibilité avec Bubble valide que l'investissement actuel dans une architecture de données propre est un accélérateur majeur pour toute évolution future sur des plateformes plus puissantes.

### 5. Architecture Technique & Ressources Clés

L'architecture de données a été profondément repensée pour accommoder la nouvelle logique de restitution.

#### 5.1. État des Scripts et Fichiers

Aucun script n'a encore été modifié. La discussion a porté sur la conception qui précède le codage. Les prochains développements impacteront principalement le script qui déclenche les calculs et envois, ainsi qu'une potentielle nouvelle librairie de `CompositionEmail.gs`.

#### 5.2. Architecture Détaillée et Traçabilité

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

\* \*\*AJOUTÉ :\*\*

\* Un nouvel onglet \*\*`sys\_Composition\_Emails`\*\*, qui devient le cœur de la restitution des résultats.

\* \*\*MODIFIÉ PROFONDÉMENT :\*\*

\* L'onglet \*\*`paramètres généraux`\*\* (nom actuel) est destiné à être restructuré pour devenir \*\*`sys\_Tests\_Config`\*\*. Il intégrera la nouvelle logique de configuration.

#### 5.3. Nomenclature des Variables

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

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

\* \*\*Contenu :\*\* Le code exact d'un profil de résultat (ex: "INTP", "Rouge").

\* \*\*Rôle :\*\* Permet de lier une brique de contenu (texte ou document) à un résultat de test spécifique. Si la cellule est vide, la brique est considérée comme générique.

\* \*\*`Moteur\_Calcul` (String) :\*\*

\* \*\*Origine :\*\* Colonne du futur onglet `sys\_Tests\_Config`.

\* \*\*Contenu :\*\* "Legacy" ou "Universel".

\* \*\*Rôle :\*\* Agit comme un drapeau pour le "routeur" logiciel, afin de diriger un test vers l'ancien ou le nouveau moteur de traitement.

\* \*\*`Config\_Destinataires` (String/JSON) :\*\*

\* \*\*Origine :\*\* Colonne du futur onglet `sys\_Tests\_Config`.

\* \*\*Contenu :\*\* Une chaîne de caractères au format JSON décrivant la liste des destinataires, leur adresse e-mail, le niveau d'e-mail à envoyer, et le délai. Exemple : `[{"dest": "Manager", "level": "N3", "delai\_h": 24, "email": "manager@email.com"}]`

\* \*\*Rôle :\*\* Centralise toute la configuration d'envoi pour un test, rendant le système scalable.

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

\* \*\*Nouvel Onglet : `sys\_Composition\_Emails`\*\*

\* \*\*Colonnes critiques :\*\* `Type\_Test`, `Code\_Niveau\_Email`, `Code\_Profil`, `Element` (valeurs : Introduction, Corps\_Texte, Document), `Ordre` (pour l'affichage), `Contenu / ID\_Document`.

\* \*\*Onglet à restructurer : `sys\_Tests\_Config`\*\* (évolution de "paramètres généraux")

\* \*\*Colonnes critiques :\*\* `ID\_Test`, `Type\_Test`, `Statut`, `Moteur\_Calcul` (Legacy/Universel), `Config\_Destinataires` (JSON).

### 6. Actions Recommandées / Prochaines Étapes

1. \*\*Action Immédiate (Pour vous) :\*\* Dans le Google Sheets `[BDD] Tests & Profils`, créer le nouvel onglet \*\*`sys\_Composition\_Emails`\*\* avec les colonnes définies en 5.4.

2. \*\*Action Suivante (Pour vous) :\*\* Choisir un premier test "legacy" simple (ex: Couleurs, Ancres) et commencer à "traduire" son contenu de résultat dans la nouvelle table `sys\_Composition\_Emails`. Cela constituera notre cas pratique pour le développement.

3. \*\*Question pour notre prochaine session :\*\* Une fois les données du premier test migrées, nous pourrons développer la partie script. La question sera donc : \*\*Quel test simple (Couleurs, Ancres...) souhaitez-vous que nous utilisions comme premier cas pratique pour la migration et le développement du nouveau moteur de script ?\*\*