PLATEFORME NUMERIQUE DE GESTION   
DES ACTIVITES STATISTIQUES

CONTEXTE CLIENT & BESOINS

Le client final INSTAT souhaite proceder a la mise en place d’une plateforme web de gestion des activites de collecte, d’enquete, revue bilan. Diagnostic et reporting pour l’ensemble de la chaine de production statistique. Cette chaine contient les Macros Activites Suivants:

* Collecte
* Programme
* Bilan
* Diagnostic
* Evaluation
* Indicateurs
* Guide Utilisateur.doc

Liste des expressions des besoins fonctionnels attendus

Faire une plateforme web, en plusieurs modules, pour aider l’INSTAT et ses partenaires à gérer toute la chaîne de production statistique. Du début à la fin. Chaque service fait un peu ce qu’il veut, avec des outils différents, et après cela on pourra centraliser ou réutiliser de facon autonomes.

La plateforme, elle devrait être composée de cinq modules principaux — peut-être qu’on peut en ajouter un sixième plus tard, mais pour l’instant on reste sur cinq.

1. Métadonnées  
    Gérer les définitions, les classifications, les indicateurs, tout ce qui sert de base aux stats. Faut qu’on puisse les créer, les modifier, les valider, et surtout les réutiliser dans plusieurs enquêtes. Là maintenant, tout le monde redéfinit la même chose à chaque fois, c’est pas viable.
2. Formulaires  
   Un éditeur de formulaires, visuel, où on peut cliquer-déplacer pour créer un questionnaire. Avec des sauts conditionnels, des validations en temps réel, des listes déroulantes dynamiques… Et pouvoir sauvegarder des modèles, pour pas tout refaire à chaque campagne.
3. Collecte  
   L’interface de saisie. Doit marcher sur mobile, tablette, PC — responsive obligatoire. Et surtout, fonctionner hors ligne, parce que dans certaines zones, y a pas de réseau. Les données se synchronisent dès que la connexion revient. Idéalement, ça peut marcher aussi avec des agents qui collectent sur le terrain.
4. Traitement des données Generique ou Superset  
   Une fois qu’on a les données, il faut les nettoyer. Donc un module avec des règles de contrôle : cohérence, plausibilité, etc. Des alertes quand quelque chose cloche, et un outil pour corriger. A defaut la base donnees sera reliee a superset a travers ses datasets pour le dataming et l’analyse de donnees.
5. Diffusion avec Superset Egalement  
   Pour publier les résultats. Tableaux, graphiques, rapports PDF, mais aussi une API pour les partenaires techniques. Avec des droits d’accès : certains voient tout, d’autres seulement certaines parties. Et un historique des publications, pour suivre les mises à jour.

La plateforme permettra de gerer chaque macro-activite de l’enquete en fonction du contexte.

Spécification/Creation du Modèle de données pour la Gestion des Formulaires de l’INSTAT

Introduction

Développer un modèle de données structurées permettant de représenter de manière generique tous les types de formulaires utilisés par l’INSTAT ainsi que par ses PTF. Ce modèle doit servir de base a l’automatisation de la conception, la generation le déploiement et la gestion des formulaires d’enquêtes web responsive all-devices. Concevoir un modèle de données tabulaire intégrant des liens entre les valeurs des colonnes et des cellules, afin de représenter l’ensemble des types de formulaires et des données utilisés par l’INSTAT et ses partenaires.

Caractéristiques du modèle de données

* Le modèle repose sur une structure tabulaire normalisée, où chaque formulaire est représenté sous forme de tableaux (feuilles) composés de colonnes, lignes et cellules.
* Il intègre un mécanisme de liens sémantiques entre les valeurs de cellules et de colonnes (ex. : dépendances conditionnelles, références croisées, validations inter-champs), permettant de modéliser des logiques complexes de saisie (branchement, filtres dynamiques, calculs automatiques, etc.).
* Les métadonnées associées (types de champs, règles de validation, contraintes, libellés multilingues, etc.) sont intégrées au modèle de manière standardisée.
* Le modèle supporte la hiérarchie et la modularité (sections, sous-sections sous-formulaires, grilles répétables) afin de couvrir l’ensemble des cas d’usage, des formulaires simples aux enquêtes complexes.

Fonctionnalités attendues   
Le modèle doit permettre la génération automatique avec ou sans IA de formulaires web interactifs avec les caracteristiques suivantes :

* Compatibilité multi-support : interfaces web responsive adaptées aux mobiles, tablettes et ordinateurs.
* Accessibilité : conformité aux standards d’accessibilité WCAG
* Intégration : interopérabilité avec les systèmes d’information existants (bases de données, API, outils de collecte CSPro CSWeb , SPSS. KOBOTOOLBOX etc.).

Resultats Attendus

* Uniformiser la conception et la gestion des formulaires de l’INSTAT & Partenaires.
* Faciliter la maintenance, l’évolution et la réutilisation des modèles de formulaires.
* Accroître la qualité, la cohérence et la fiabilité des données collectées.
* Réduire les délais de déploiement des campagnes d’enquête grâce à une génération automatisée des interfaces graphiques web.

Il serait utile egalement de mettre en place un pipeline de Big Data et d’intelligence artificielle, intégré à la plateforme, pour accompagner tout le cycle de production statistique.  
L’objectif ? Automatiser certaines étapes comme le nettoyage des données, la détection d’anomalies, ou même proposer des suggestions de correction.  
Ce système pourrait aller de la collecte jusqu’à la diffusion, en passant par le traitement et l’analyse.  
Cela permettrait de gagner en efficacité, en qualité des données, et de réduire la charge de travail manuel.

Expressions des besoins détaillées

Dans un souci de gain de délais, il a été décidé de concevoir le modèle en se concentrant sur les macro-activités suivantes :

* Programme
* Bilan
* Diagnostic

La modélisation présentée ci-après a été réalisée à partir des fichiers Word des formulaires d’enquête relatifs à ces trois macro-activités, puis transposée au format Excel.  
Ce document sera accompagné d’un fichier ZIP contenant, pour chaque macro-activité, les deux fichiers suivants :

1. Le modèle au format Word
2. Le modèle au format Excel associé

EB-000:

Application Web avec Base de donnees d’integration de generation et gestion des formulaires d’enquete parametrables a partir des donnees d’un fichier excel et word. Architecture MultiTiers Avec Java Python Php ou Go. Base de donnees PostgreSQL, Eventuellement Architecture No-Code de Type BaseRow Yao ou equivalent dokerise etc.

EB-001:

Creer une base donnees Postgresql qui doit contenir les 3 schemas des 3 formulaires d’enquetes correspondant aux 3 macros activites.

* Programme
* Bilan
* Diagnostique

Ces schémas constituent une version modifiée du modèle Entité-Relation (ERD) fourni dans le document de référence.

**EB-002 :**Développer une API de parsing des fichiers Excel modélisés pour les trois macro-activités, et insérer ou ventiler les résultats du parsing dans les différentes tables de chaque schéma de base de données.

**EB-003 :**Vérifier de manière aléatoire, en fonction des libellés ou des identifiants des entrées (Entry), la cohérence et la validité des différents chemins hiérarchiques (Path, Parent, DirectParent, Children, etc.).

**EB-004 :**Concevoir une application de génération de formulaires, basée sur une approche générative (IA), un langage spécifique de domaine (DSL) ou une API, capable d’alimenter les champs Responses et OptionalResponse de chaque schéma de base de données.

|  |  |
| --- | --- |
| **URL: Contexte** | [https://survey-bilan.instat.com/energie](https://survey-bilan.instat.com/entreprises-energie) |
| **Survey** | Bilan |
| **Context** | Energie |
| **Section** | Menu principal : Producteur |
| **Sous-Section** | Page active : *Producteur Energie-Propre* |
| **Question** | Nombre total de site de prodution |
| **Réponse** | Champ <input> obligatoire ou facultatif ( Rajouter \* ou Couleur etc. |
| **Réponse optionnelle** | Production Multi-Sites Indiquer les Sites <InoutText MultiLigne> |

EB-005:

Le formulaire Word doit etre generer par l’application par une API ou a partir de I’interface web ou par CLI et mise a la disposition pour telechargement avec Template.

Application securiser avec gestion des droits: Admin, Manager, DataScientiste, ReadOnly, Write,

## SurveyEntry : Table/Entity Data Classe

La prestation consiste a créer une plateforme web multi module qui va permettre a l’INSTAT & ses partenaires de gérer les activités de la chaîne de production statistique. La plateforme est constituée de 5 modules contextuels:

1. Bilan

2. Collecte

3. Diagnostic

4. Evaluation

5. Indicateurs

La plateforme permettra de gerer chaque activite de l’enquete en fonction du contexte.

## SurveyEntry : Table/Entity Data Classe

+ id : integer

+ entryLabel : str

+ entryName : EnumElem {

Survey,

Context,

Section : 2,

SubSection : 3,

Question : 4,

OptionalQuestion : 5,

Response : 5,

OptionalResponse : 5

}

+ entryParentIndex : integer

(Index of the parent entry of this entry)

+ entryIndex : integer

(Index from 1 of this entry among the children of its parent entry)

+ directParentId : integer

(Reference to the primary ID of the direct parent of this entry)

+ entryFullPath : str (computed via getEntryFullPath)

(Full path from the top-level Context entry to this entry)

+ entryDescription : str

(Description of the entry)

+ entryAnnotation : str

(Supplementary information providing further context on the entry’s role and meaning)

+ caution : str

(Warning or cautionary text related to the entry)

+ existingConditions : str

(Conditions under which this entry should exist)

+ jumpTo : str

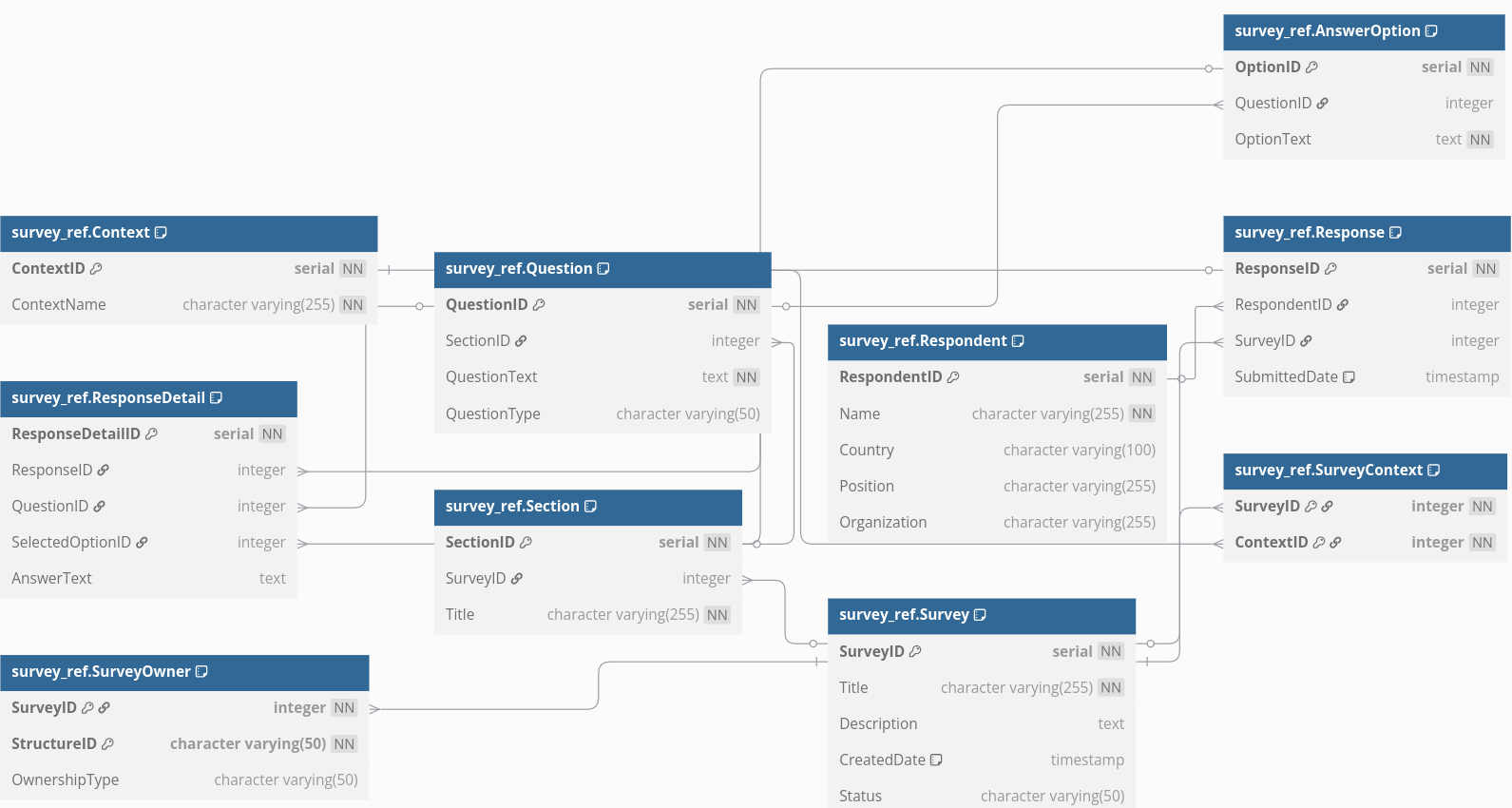
(Condition under which the value of this entry leads to another entry. Typically used for conditional navigation.)

## SurveyEntryTypeDepth : Enum Entity /Table

entryTypeDepth = Enum('EntryTypeDepth', [('Survey ', 0), ('Context', 1), ('Section', 2),('SubSection', 3), ('Question', 4),('OptionalQuestion', 5), ('Response', 5), ('OptionalResponse', 6)])

Création d’un schéma de base de données pour la Gestion des formulaires et les documents d’enquêtés de l’INSTAT.

## Il faut s’en inspirer, mais y apporter les modifications nécessaires. Par exemple, chaque type de macro-activité doit disposer d’un schéma de base de données dédié. L’ensemble des schemas doit être regroupé dans une même instance de base de données.



Le DDL associe

BEGIN;

CREATE TABLE IF NOT EXISTS survey\_ref."AnswerOption"

(

"OptionID" serial NOT NULL,

"QuestionID" integer,

"OptionText" text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "AnswerOption\_pkey" PRIMARY KEY ("OptionID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."Context"

(

"ContextID" serial NOT NULL,

"ContextName" character varying(255) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Context\_pkey" PRIMARY KEY ("ContextID"),

CONSTRAINT "Context\_ContextName\_key" UNIQUE ("ContextName")

);

CREATE TABLE IF NOT EXISTS survey\_ref."Question"

(

"QuestionID" serial NOT NULL,

"SectionID" integer,

"QuestionText" text COLLATE pg\_catalog."default" NOT NULL,

"QuestionType" character varying(50) COLLATE pg\_catalog."default",

CONSTRAINT "Question\_pkey" PRIMARY KEY ("QuestionID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."Respondent"

(

"RespondentID" serial NOT NULL,

"Name" character varying(255) COLLATE pg\_catalog."default" NOT NULL,

"Country" character varying(100) COLLATE pg\_catalog."default",

"Position" character varying(255) COLLATE pg\_catalog."default",

"Organization" character varying(255) COLLATE pg\_catalog."default",

CONSTRAINT "Respondent\_pkey" PRIMARY KEY ("RespondentID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."Response"

(

"ResponseID" serial NOT NULL,

"RespondentID" integer,

"SurveyID" integer,

"SubmittedDate" timestamp without time zone DEFAULT CURRENT\_TIMESTAMP,

CONSTRAINT "Response\_pkey" PRIMARY KEY ("ResponseID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."ResponseDetail"

(

"ResponseDetailID" serial NOT NULL,

"ResponseID" integer,

"QuestionID" integer,

"SelectedOptionID" integer,

"AnswerText" text COLLATE pg\_catalog."default",

CONSTRAINT "ResponseDetail\_pkey" PRIMARY KEY ("ResponseDetailID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."Section"

(

"SectionID" serial NOT NULL,

"SurveyID" integer,

"Title" character varying(255) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Section\_pkey" PRIMARY KEY ("SectionID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."Survey"

(

"SurveyID" serial NOT NULL,

"Title" character varying(255) COLLATE pg\_catalog."default" NOT NULL,

"Description" text COLLATE pg\_catalog."default",

"CreatedDate" timestamp without time zone DEFAULT CURRENT\_TIMESTAMP,

"Status" character varying(50) COLLATE pg\_catalog."default",

CONSTRAINT "Survey\_pkey" PRIMARY KEY ("SurveyID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."SurveyContext"

(

"SurveyID" integer NOT NULL,

"ContextID" integer NOT NULL,

CONSTRAINT "SurveyContext\_pkey" PRIMARY KEY ("SurveyID", "ContextID")

);

CREATE TABLE IF NOT EXISTS survey\_ref."SurveyOwner"

(

"SurveyID" integer NOT NULL,

"StructureID" character varying(50) COLLATE pg\_catalog."default" NOT NULL,

"OwnershipType" character varying(50) COLLATE pg\_catalog."default",

CONSTRAINT "SurveyOwner\_pkey" PRIMARY KEY ("SurveyID", "StructureID")

);

ALTER TABLE IF EXISTS survey\_ref."AnswerOption"

ADD CONSTRAINT "AnswerOption\_QuestionID\_fkey" FOREIGN KEY ("QuestionID")

REFERENCES survey\_ref."Question" ("QuestionID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."Question"

ADD CONSTRAINT "Question\_SectionID\_fkey" FOREIGN KEY ("SectionID")

REFERENCES survey\_ref."Section" ("SectionID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."Response"

ADD CONSTRAINT "Response\_RespondentID\_fkey" FOREIGN KEY ("RespondentID")

REFERENCES survey\_ref."Respondent" ("RespondentID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."Response"

ADD CONSTRAINT "Response\_SurveyID\_fkey" FOREIGN KEY ("SurveyID")

REFERENCES survey\_ref."Survey" ("SurveyID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."ResponseDetail"

ADD CONSTRAINT "ResponseDetail\_QuestionID\_fkey" FOREIGN KEY ("QuestionID")

REFERENCES survey\_ref."Question" ("QuestionID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."ResponseDetail"

ADD CONSTRAINT "ResponseDetail\_ResponseID\_fkey" FOREIGN KEY ("ResponseID")

REFERENCES survey\_ref."Response" ("ResponseID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."ResponseDetail"

ADD CONSTRAINT "ResponseDetail\_SelectedOptionID\_fkey" FOREIGN KEY ("SelectedOptionID")

REFERENCES survey\_ref."AnswerOption" ("OptionID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE SET NULL;

ALTER TABLE IF EXISTS survey\_ref."Section"

ADD CONSTRAINT "Section\_SurveyID\_fkey" FOREIGN KEY ("SurveyID")

REFERENCES survey\_ref."Survey" ("SurveyID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."SurveyContext"

ADD CONSTRAINT "SurveyContext\_ContextID\_fkey" FOREIGN KEY ("ContextID")

REFERENCES survey\_ref."Context" ("ContextID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."SurveyContext"

ADD CONSTRAINT "SurveyContext\_SurveyID\_fkey" FOREIGN KEY ("SurveyID")

REFERENCES survey\_ref."Survey" ("SurveyID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

ALTER TABLE IF EXISTS survey\_ref."SurveyOwner"

ADD CONSTRAINT "SurveyOwner\_SurveyID\_fkey" FOREIGN KEY ("SurveyID")

REFERENCES survey\_ref."Survey" ("SurveyID") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE;

END;

***Il faut rajouter les Entry de type “Sous-section” qui n’y est pas. Une Section peut contenir 0 ou plusieurs Sous-Sections.***

***Les propositions de modelisation BD sont les bienvenus.***