

Report - DataBase Project - IFEBY140

Table of contents

1	Introduction	3
2	Analyzis	4
2.1	Redudant Attributes	4
2.2	First Normal Form	4
2.3	Functional dependencies :	5
2.4	Constraints	5
2.5	Misc	5
3	Modelization	6
3.1	ER schema	6
3.2	Tables	6
4	Implementation	8
4.1	Execution	8
4.2	Code Details	8
4.2.1	Transport	8
4.2.2	Sub Event	8

1 Introduction

[Que Faire à Paris ?](#) is a french website that list events happing in Paris.

The event are archived at this [address](#) in multiple formats including csv.

In this project we propose a normalized database to store the event based on the archive format.

2 Analyzis

Initial Columns :

```
['ID' 'URL' 'Titre' 'Chapeau' 'Description' 'Date de début' 'Date de fin'
 'Occurrences' 'Description de la date' "URL de l'image"
 "Texte alternatif de l'image" "Crédit de l'image" 'Mots clés'
 'Nom du lieu' 'Adresse du lieu' 'Code postal' 'Ville'
 'Coordonnées géographiques' 'Accès PMR' 'Accès mal voyant'
 'Accès mal entendant' 'Transport' 'Url de contact' 'Téléphone de contact'
 'Email de contact' 'URL Facebook associée' 'URL Twitter associée'
 'Type de prix' 'Détail du prix' "Type d'accès" 'URL de réservation'
 'URL de réservation - Texte' 'Date de mise à jour' 'Image de couverture'
 'Programmes' 'En ligne - address_url' 'En ligne - address_url_text'
 'En ligne - address_text' 'title_event' 'audience' 'childrens' 'group']
```

2.1 Redudant Attributes

We will remove the following attributes : `description_de_la_date`

2.2 First Normal Form

Non atomic attributs :

- List
 - Occurences : separated by '_' (underscore)
 - Tags : separated by ','
 - Childrens : separated by ','
 - Transport : separated by '\n'
- Multiple attributes
 - Transport : `transport_type`, `transport_line`, `station`, `distance`
 - Geographic_Coordinates : `longitude`, `latitude`

2.3 Functional dependencies :

- $id \rightarrow url$
 - For simplicity sake, we consider that we cannot deduce `id` from `url`. Perhaps, we will use this property to insert identify `sub_events` in “childrens”
- $id \rightarrow titre, chapeau, description, date_de_debut, date_de_fin, occurrences, url_de_l_image, texte_alternatif_de_l_image, credit_de_l_image, mots_clefs, nom_du_lieu, adresse_du_lieu, code_postale, ville, url_du_contact, telephone_de_contact, email_de_contact, url_facebook_associee, url_twitter_associe, type_de_prix, detail_du_prix, type_d_acces, url_de_reservation, url_de_reservation_texte, date_de_mise_a_jour, image_de_couverture, programmes, en_ligne_address_url, en_ligne_address_url_text, en_ligne_address_text, title_event, audience, childrens, groupe, transport_type, transport_line, station, distance$
- $date_de_debut, date_de_fin \rightarrow description_de_la_date$
 - Removed attribute
- $adresse_du_lieu, ville, code_postale \rightarrow coordonnees_geographiques$
- $nom_du_lieu, adresse_du_lieu, ville, code_postale \rightarrow acces_pmr, acces_mal_voyant, acces_mal_entendant$

2.4 Constraints

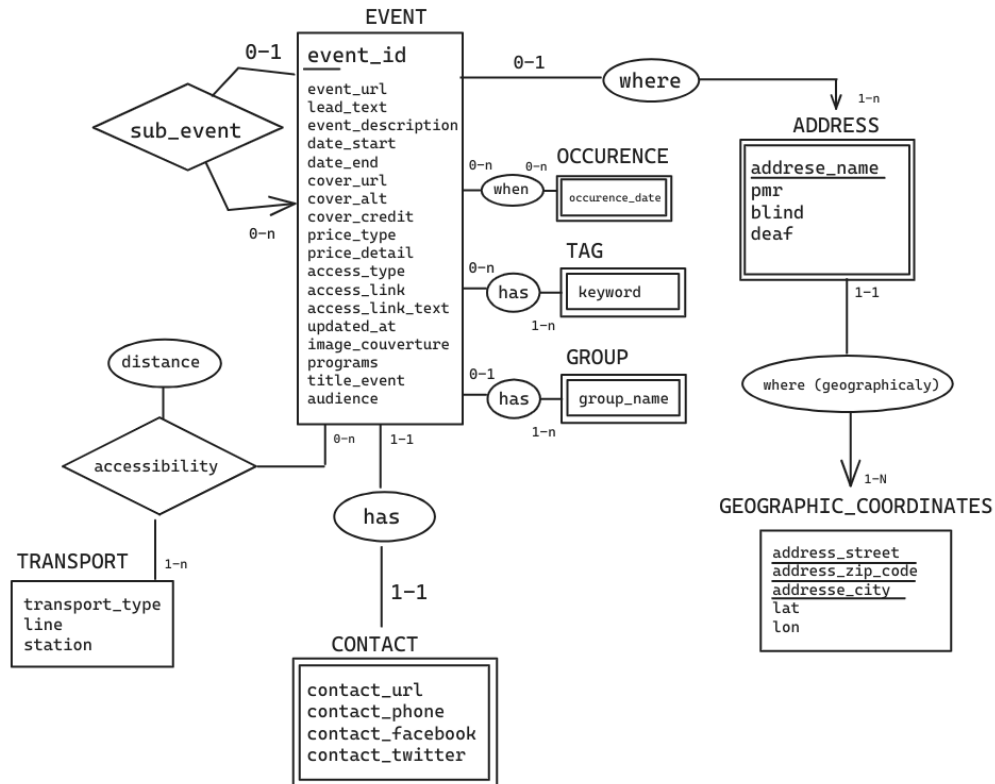
- Realistic implementation expectations
 - $date_end > date_start$
 - NOT NULL :
 - * `id`, `url`
 - * $(date_start \text{ AND } date_end) \text{ OR } (occurrences)$
- Unrealistic implementation expectations
 - `url` must finish with `id`
 - all `urls` must be valid
 - address must be valid
 - `phone_number` must be valid

2.5 Misc

We decide to translate everything to english.

3 Modelization

3.1 ER schema



The following decomposition, satisfies Boyce-codd normal form at the same time :

3.2 Tables

geographic_correspondance(address_street, address_zipcode, address_city | lat, lon)

address_table(address_name, address_street, address_zipcode, address_city | pmr, blind, deaf)

event_table (event_id | event_url, title, lead_text, event_description, date_start, date_end, cover_url, cover_alt, cover_credit, address_name, address_street, address_zipcode, address_city, price_type, price_detail, access_type, access_link, access_link_text, updated_at, image_couverture, programs, title_event, audience, contact_url, contact_phone, contact_mail,

contact_facebook, contact_twitter, address_url, address_url_text, address_text, keyword,
group_name, parent_event_id)

occurence(event_id, occurence_date)

tag(event_id | keyword)

transport(event_id, transport_type, transport_line, station | distance)

4 Implementation

4.1 Execution

Start psql in src folder.

```
cd src  
psql -d <database> username
```

Create tables and populate them with :

```
\i XXX_YYY_tables.sql
```

See the preview of the data with :

```
\i XXX_YYY_data.sql
```

4.2 Code Details

4.2.1 Transport

4.2.2 Sub Event