**Compte Rendu du TP : Gestion des données d’un aéroport en XML, DTD et XSD**

**1. Contexte**

Ce TP avait pour objectif de modéliser les données d’un aéroport en utilisant **XML**, de les organiser grâce à une **DTD** et un **XML Schema (XSD)**, et de pratiquer des requêtes **XPath** et **XQuery** pour interroger ces données de manière efficace. Les informations à gérer étaient :

* **Employés** : nom, prénom, âge, rôle.
* **Pistes** : numéro, longueur, localisation.
* **Compagnies aériennes** : nom, pays, code IATA.
* **Avions** : modèle, capacité, immatriculation.
* **Vols** : numéro, date, destination, associés à une compagnie, un avion, une piste et plusieurs employés.

L’objectif global était de structurer ces données pour qu’elles soient faciles à utiliser par toute application ou service tiers.

**2. Objectifs**

1. Créer un **fichier XML type** regroupant toutes les informations.
2. Définir une **DTD** pour assurer la bonne structure du XML.
3. Créer un **XML Schema (XSD)** pour une validation plus stricte et fiable.
4. Valider le fichier XML avec la DTD et le XSD.
5. Réaliser des requêtes **XPath** pour extraire rapidement certaines informations.
6. Réaliser des requêtes **XQuery** pour des besoins plus complexes.

**3. Méthodologie**

**3.1 Création du fichier XML type**

* La structure suit une hiérarchie logique : <aeroport> contient <compagnies>, <pistes>, <avions>, <employes> et <vols>.
* Chaque élément clé possède un **identifiant unique (@id)** permettant de relier les vols aux compagnies, avions, pistes et employés.
* Exemple d’un vol :

<vol id="v1">

<numero>AF100</numero>

<date>2023-02-15</date>

<destination>New York</destination>

<compagnieRef>c1</compagnieRef>

<avionRef>a1</avionRef>

<pisteRef>p1</pisteRef>

<employesRefs>e1 e2 e3</employesRefs>

</vol>

* Les références multiples (employés) sont listées par espace et gérées via xs:IDREFS dans le XSD.

**3.2 Création de la DTD (aeroport.dtd)**

* Définition des éléments principaux et de leurs sous-éléments.
* Les identifiants @id sont obligatoires pour tous les éléments qui sont référencés.
* Exemple :

<!ELEMENT aeroport (compagnies, pistes, avions, employes, vols)>

<!ELEMENT compagnies (compagnie+)>

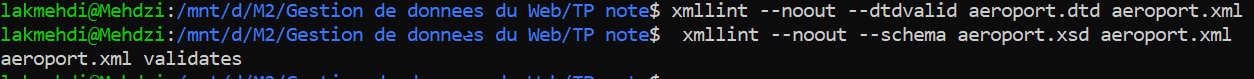
<!ELEMENT compagnie (nom, pays, codeIATA)>

<!ATTLIST compagnie id ID #REQUIRED>

**3.3 Création du XML Schema (aeroport.xsd)**

* Utilisation de **types complexes** pour chaque entité.
* Validation stricte avec :
  + xs:ID pour tous les éléments qui peuvent être référencés.
  + xs:IDREF et xs:IDREFS pour relier vols ↔ compagnies/pistes/avions/employés.
  + xs:date pour les dates de vols.

**3.4 Validation**

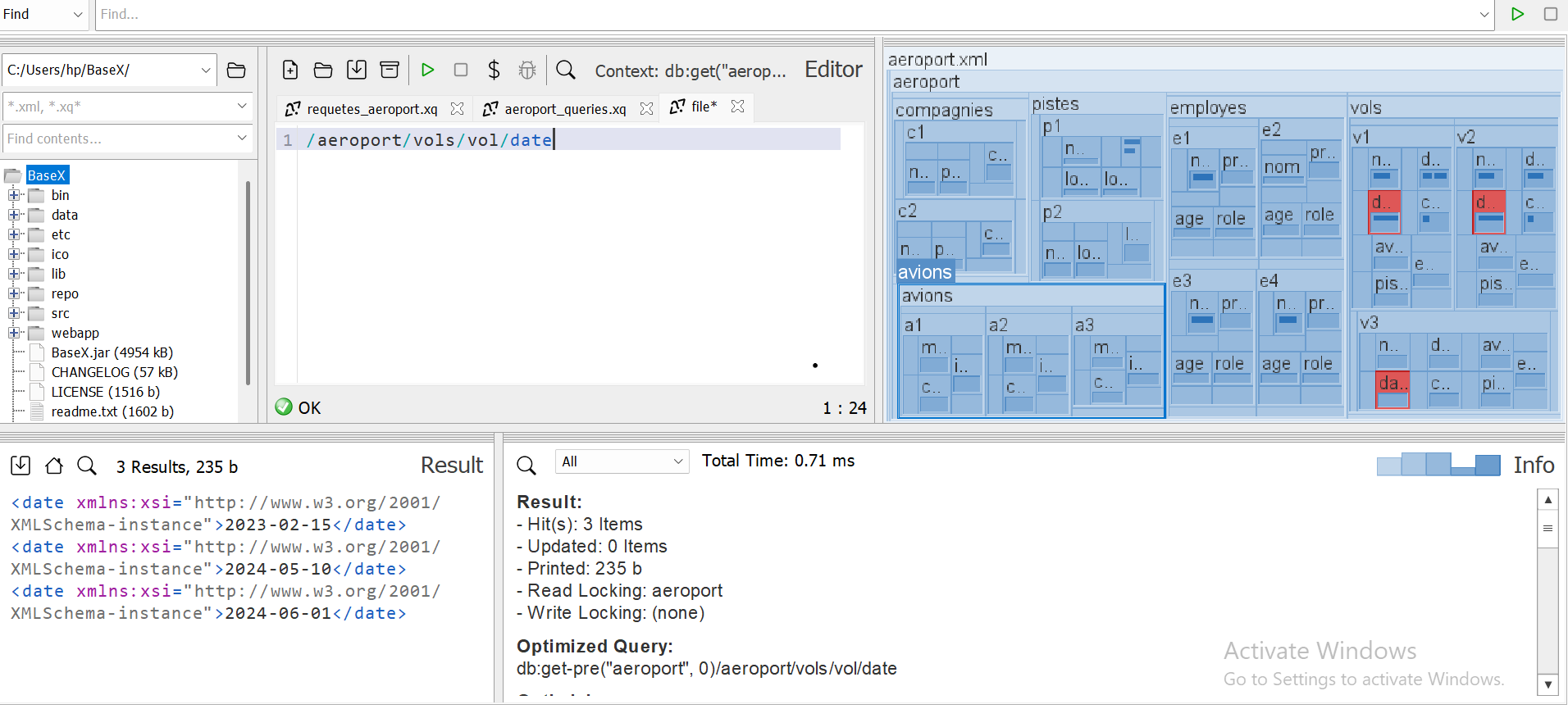
* DTD : xmllint --dtdvalid aeroport.dtd aeroport.xml et validation via PHP(voir script).
* XSD : xmllint --schema aeroport.xsd aeroport.xml et PHP(voir script).
* Les erreurs ont été corrigées jusqu’à ce que le fichier XML soit entièrement valide. 

**3.5 Requêtes XPath et XQuery**

**Exemples XPath :**

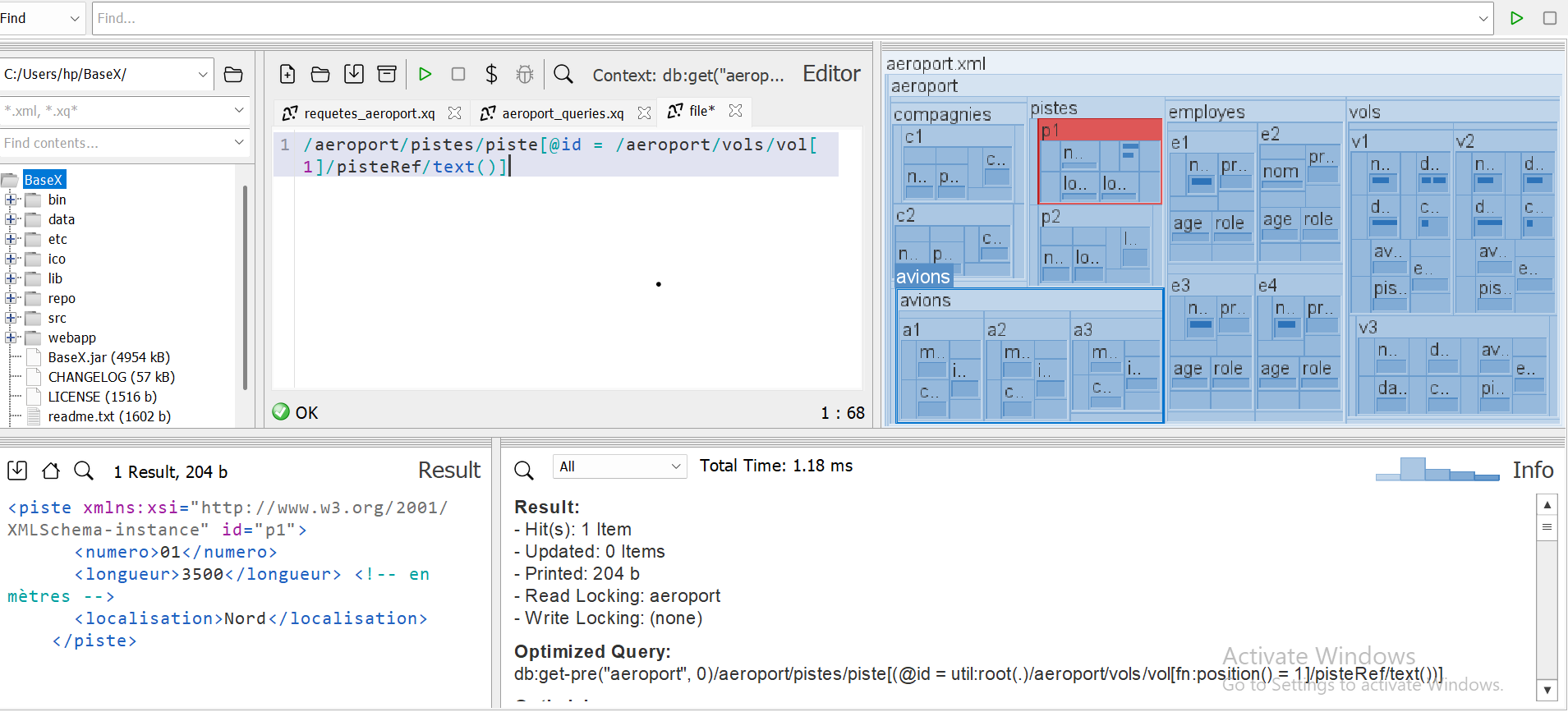
1. Lister les dates de tous les vols :

/aeroport/vols/vol/date



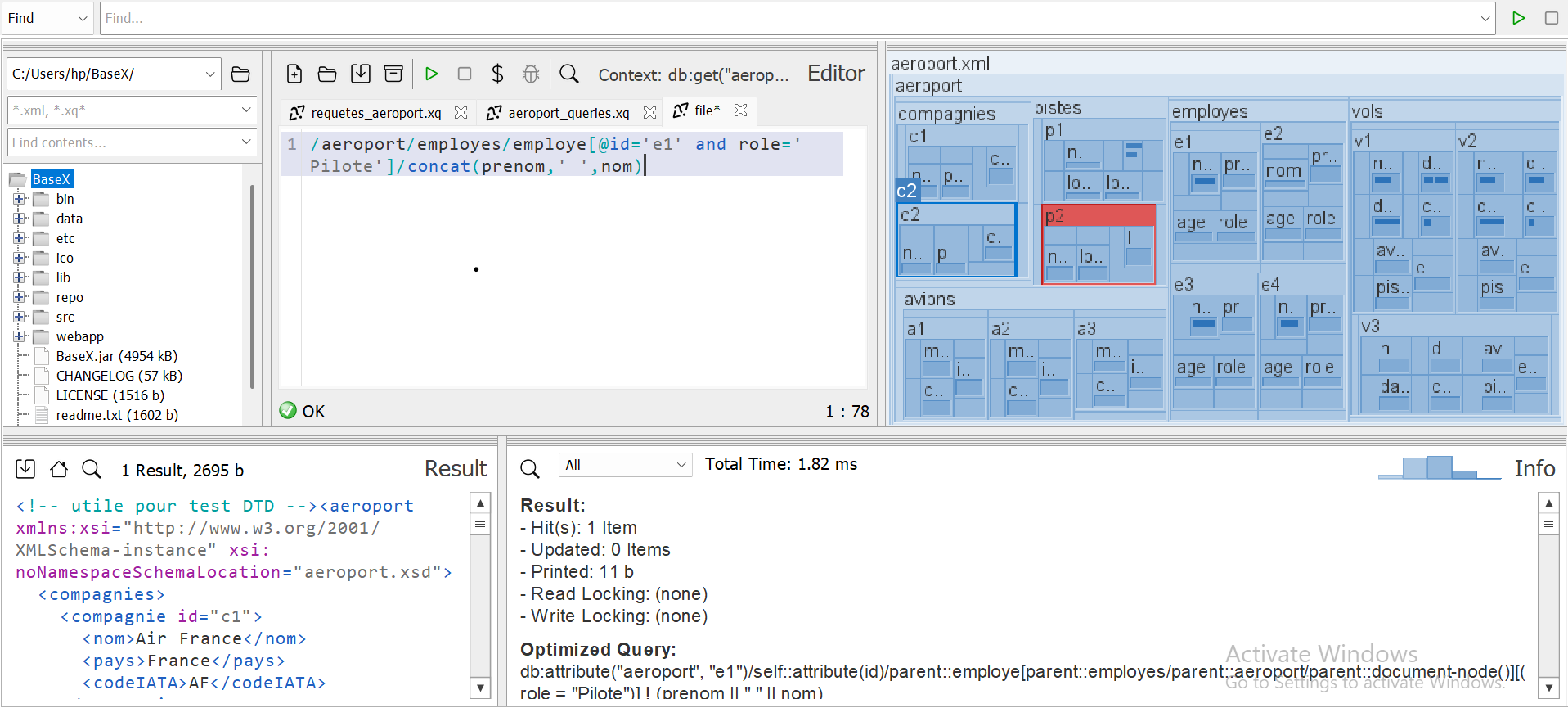
1. Identifier la piste du premier vol :

/aeroport/pistes/piste[@id = /aeroport/vols/vol[1]/pisteRef/text()]



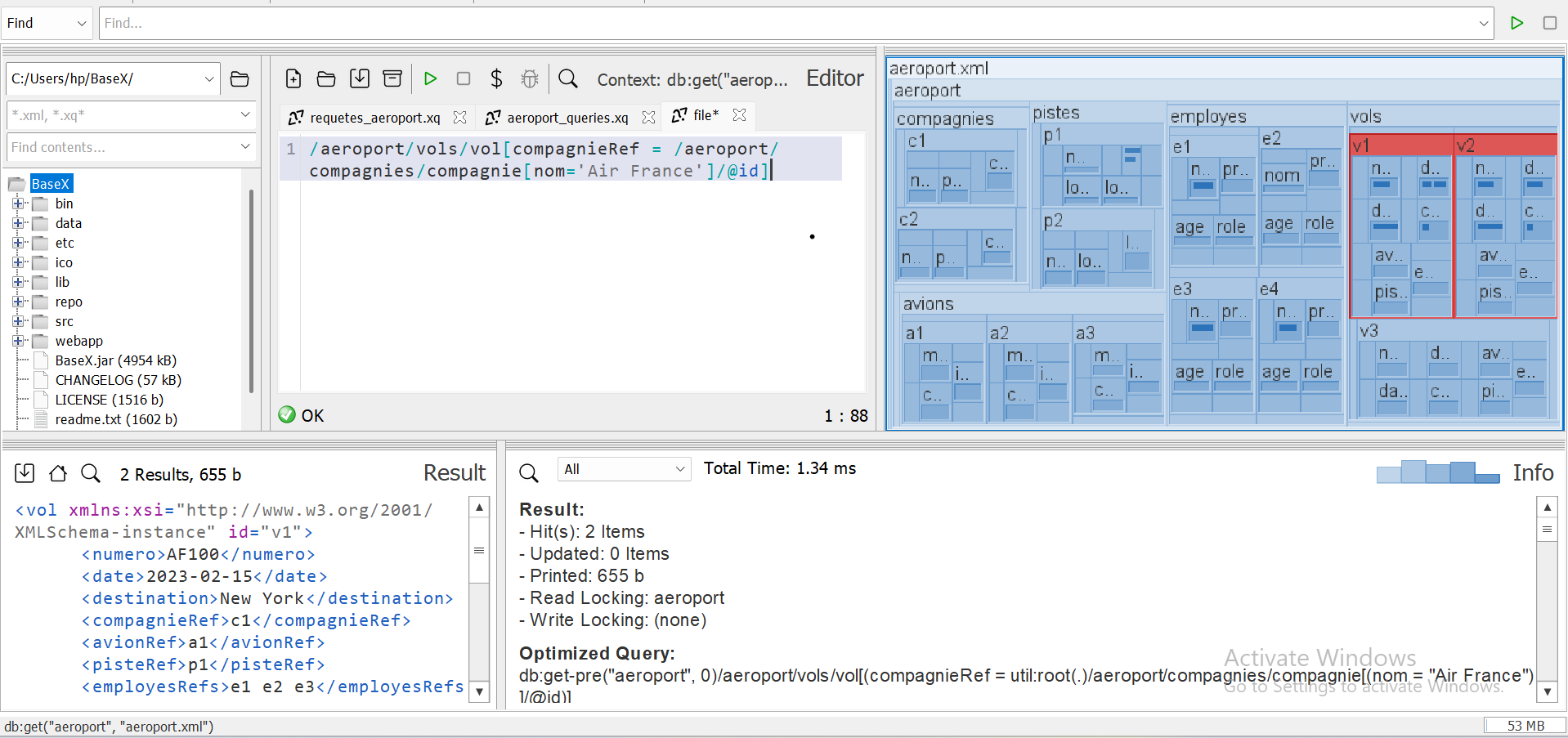
1. Nom du pilote pour chaque vol :

/aeroport/employes/employe[@id='e1' and role='Pilote']/concat(prenom,' ',nom)



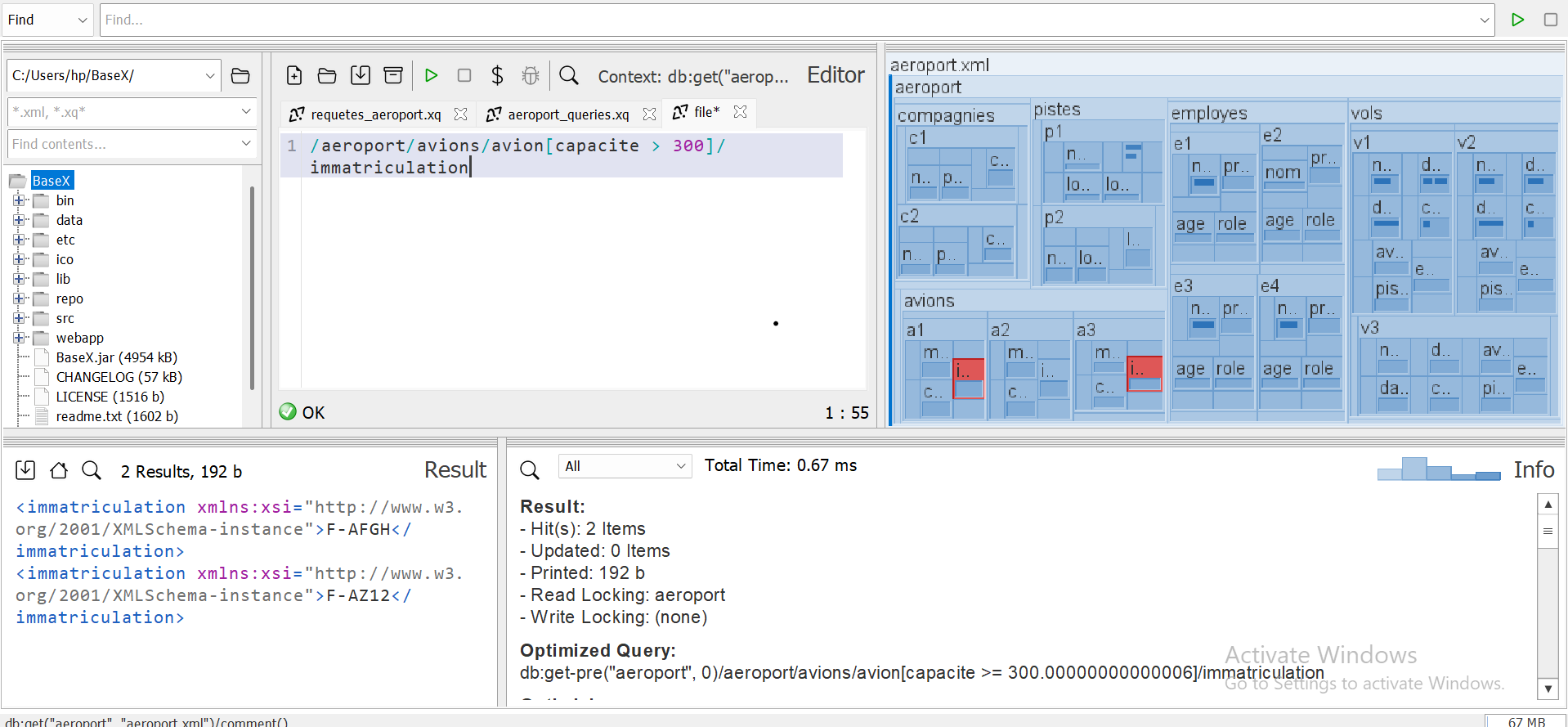
1. Vols opérés par Air France :

/aeroport/vols/vol[compagnieRef = /aeroport/compagnies/compagnie[nom='Air France']/@id]



1. Immatriculations des avions de plus de 300 places :

/aeroport/avions/avion[capacite > 300]/immatriculation



**Exemples XQuery :**

1. Vols Air France après le 1er janvier 2023 :

let $root := doc("aeroport.xml")/aeroport

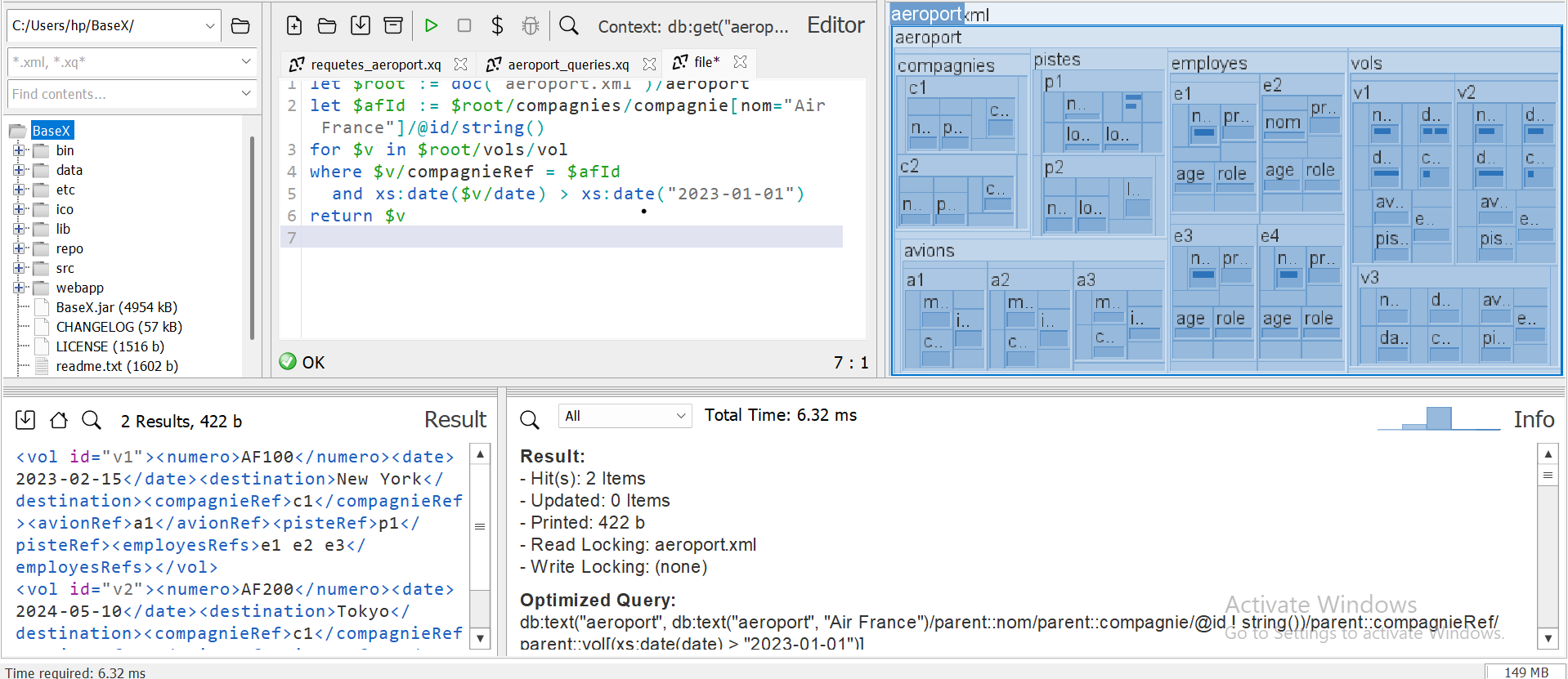
let $afId := $root/compagnies/compagnie[nom="Air France"]/@id/string()

for $v in $root/vols/vol

where $v/compagnieRef = $afId

and xs:date($v/date) > xs:date("2023-01-01")

return $v



1. Employés affectés à plus d’un vol :

let $root := doc("aeroport.xml")/aeroport

for $e in $root/employes/employe

let $id := $e/@id/string()

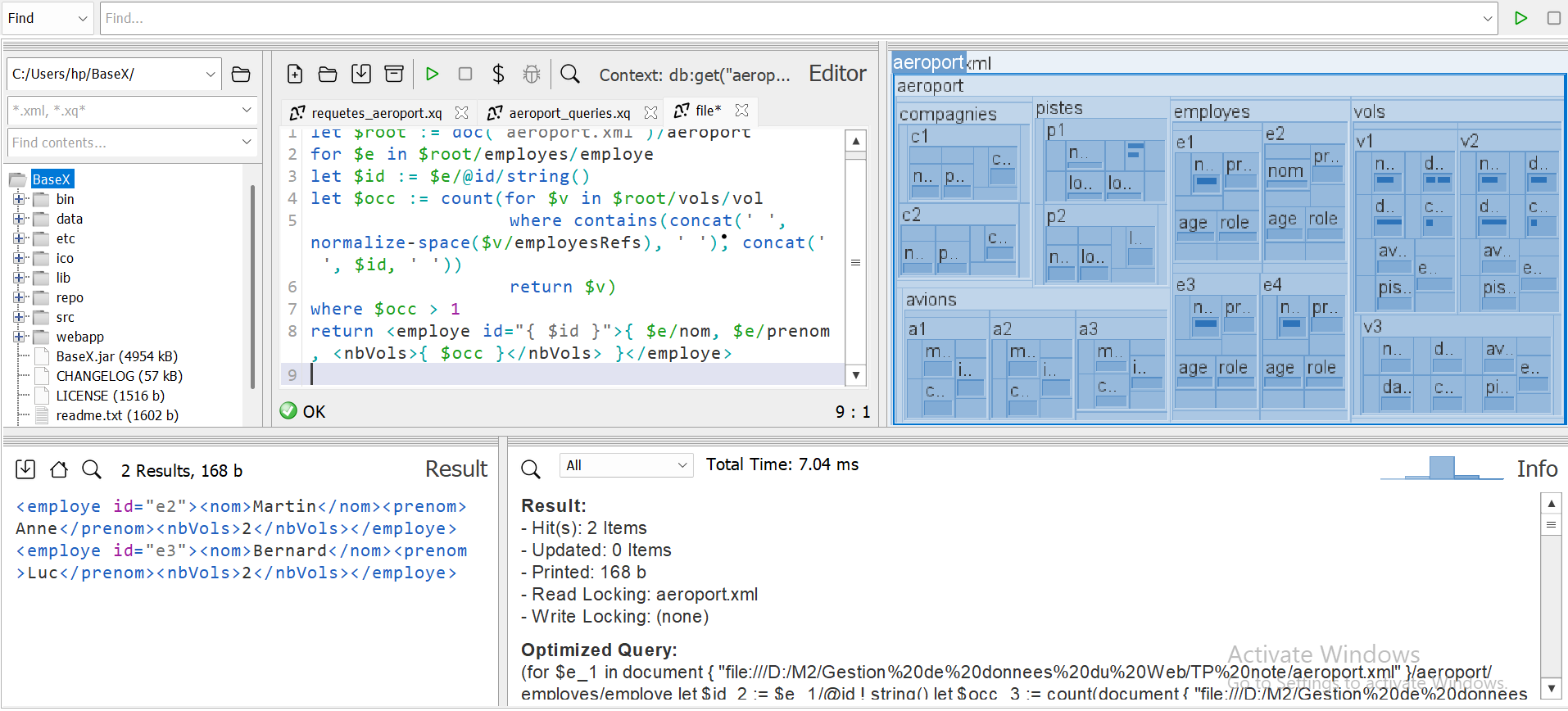
let $occ := count(for $v in $root/vols/vol

where contains(concat(' ', normalize-space($v/employesRefs), ' '), concat(' ', $id, ' '))

return $v)

where $occ > 1

return <employe id="{ $id }">{ $e/nom, $e/prenom, <nbVols>{ $occ }</nbVols> }</employe>



**4. Résultats**

* Le fichier XML est **valide** avec DTD et XSD.
* Les requêtes XPath permettent de récupérer rapidement toutes les informations importantes : dates des vols, pistes, pilotes, compagnies, avions de grande capacité.
* Les requêtes XQuery permettent de filtrer des cas complexes comme les vols récents et les employés impliqués dans plusieurs vols.

**5. Analyse**

* L’utilisation des **ID et IDREF(S)** assure la cohérence des données et simplifie les requêtes.
* La combinaison DTD/XSD permet de vérifier à la fois la structure et la validité stricte du fichier.
* Les XQuery offrent plus de puissance que XPath seul, notamment pour le filtrage conditionnel et les calculs.
* Le modèle reste **facilement extensible** pour l’ajout de nouvelles compagnies, avions ou vols.

**6. Conclusion**

* Ce TP a permis de :
  1. Créer un **modèle XML clair et cohérent** pour l’aéroport.
  2. Appliquer une **DTD et un XSD** pour une validation fiable.
  3. Tester et exécuter des **requêtes XPath et XQuery** pour exploiter les données.
* Le modèle est prêt pour une **interopérabilité** avec d’autres systèmes (gestion des vols, planning des employés, suivi des avions).
* Compétences acquises : XML hiérarchique, DTD vs XSD, ID et références, XPath, XQuery, structuration et interopérabilité des données.