Noms et prénoms : LEMONNIER Etienne, ROYER Victor

Classe : M1

Module : Bases de données

Responsable du module : Benoit Lardeux

Titre du document : Rapport de projet - Inventaire d'une collection d'œuvres d'art

Date et heure d'envoi : 24/09/2021 23:00

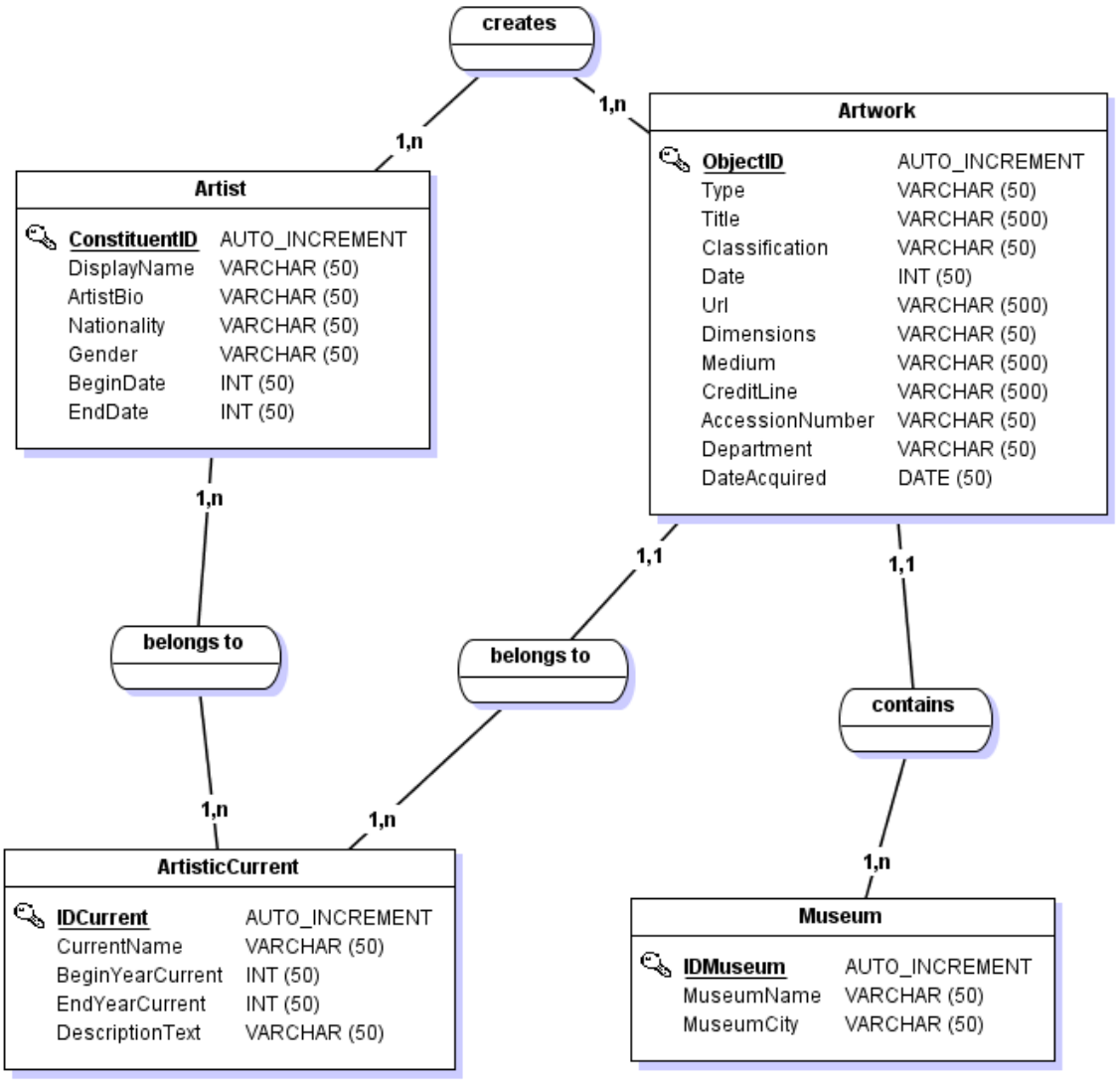
Nombre de mots : 1644

☑ En adressant ce document à l'enseignant, je certifie que ce travail est le mien et que j'ai pris connaissance des règles relatives au référencement et au plagiat.

Partie 1.2 : Modélisation

Modèle conceptuel de données (MCD) :

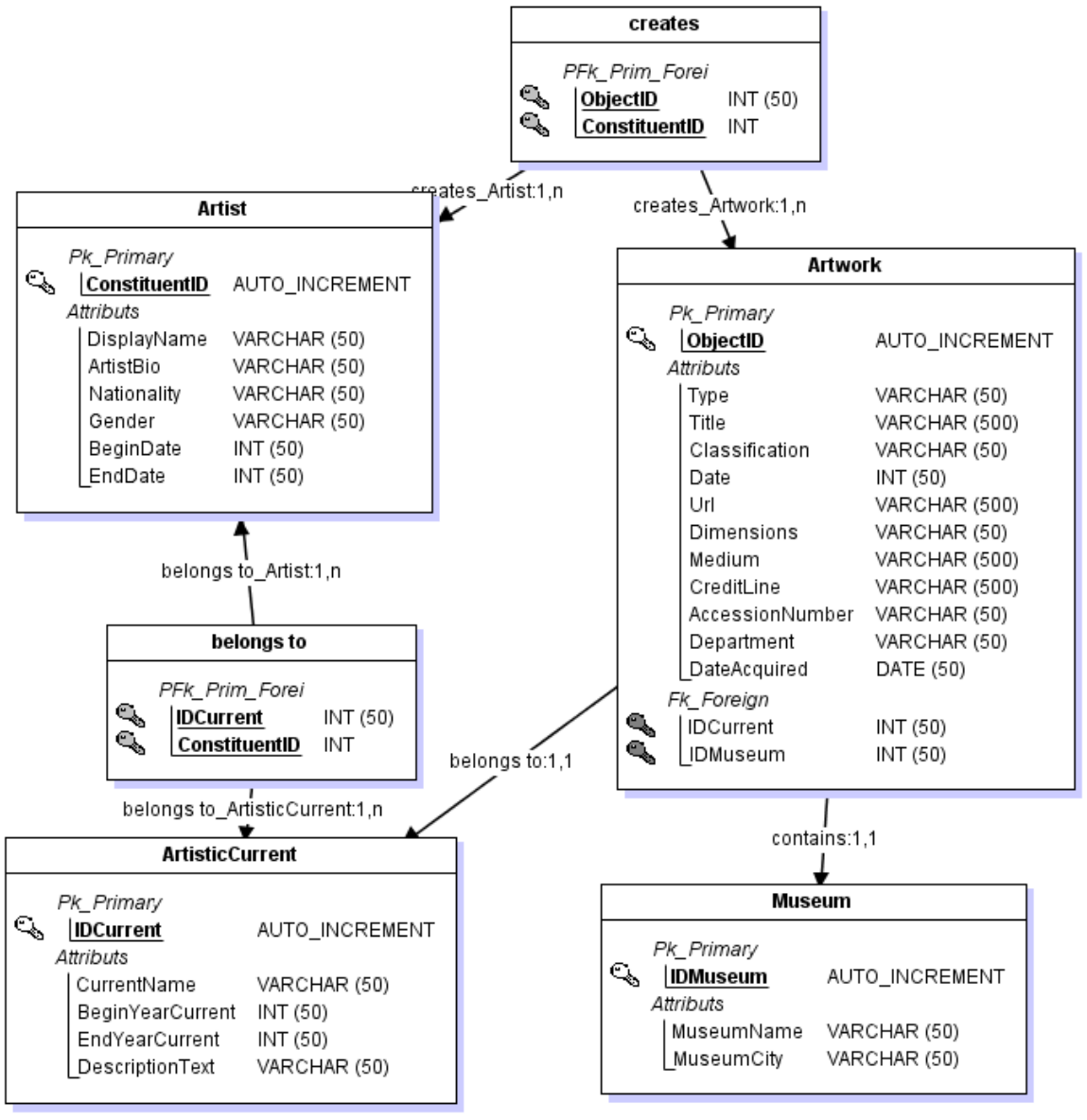
Voici notre modèle :



Nous avons choisi de diviser notre modèle en quatre tables différentes, possédant chacune un identifiant unique pour éviter les doublons dans les tables. Les variables ont été choisies de telle sorte qu'elles ne puissent chacune figurer que dans une seule table. Des explications à propos des choix des cardinalités figurent dans la partie 1.3 d).

Modèle physique de données (MPD) :

Voici le MPD généré à l'aide du logiciel JMerise :



NB : Dans notre code, nous avons utilisé une majorité de VARCHAR à la place des types cités ci-dessus, pour faciliter la gestion des données et éviter certains problèmes de compatibilité (voir choix de types dans le dossier projet > Scripts SQL > createTables.sql).

Partie 1.3 : Insertion d'un jeu de données partiel

a) *Veuillez explorer ces fichiers et indiquez les données manquantes dans les fichiers d'entrée par rapport à la base de données décrite auparavant*

Les éléments non présents dans les fichiers csv sont : le type d'œuvre, les principaux éléments des courants artistiques comme le nom du courant, la période (année de début, année de fin) et le texte descriptif. Les éléments manquants pour l'artiste sont : son texte descriptif, ses date de naissance et de décès (seule l'année est renseignée) et son courant artistique. Les informations sur les musées (nom, ville) sont également manquantes.

b) *Pour chacun des champs dans ces deux fichiers, indiquez la valeur ayant le maximum d'occurrences, la quantité et la proportion que celles-ci représentent*

Afin de déterminer les valeurs les plus présentes sur les fichiers d'entrée, nous avons créé deux tables temporaires "Artist" et "Artwork", où nous avons directement téléversé les fichiers csv. Voici le script SQL permettant de prendre la valeur ayant le plus d'occurrences par colonne, et indiquant son nombre d'occurrences (ici le cas de "Nationality" pour la table Artist) :

```
SELECT Nationality, COUNT(*)
FROM Artist
GROUP BY Nationality
ORDER BY COUNT(*)
DESC
LIMIT 1;
```

Voici un récapitulatif des valeurs trouvées :

Colonne	Valeur la plus présente	Nombre	Proportion (%)	Description (si nécessaire)
Artists.csv				
ConstituentID	1	1	0,01	Chaque identifiant est unique
DisplayName	Unknown Designer	22	0,14	22 artistes sont inconnus
ArtistBio		2133	14,01	2133 artistes n'ont pas de biographie
Nationality	American	5171	33,97	
Gender	Male	9750	64,05	
BeginDate	0	3597	23,63	3597 artistes ont une date de naissance inconnue
EndDate	0	10084	66,24	10084 artistes sont encore en vie ou date de décès inconnue
Wiki_QID		11856	77,88	11856 artistes n'ont pas de code QID
ULAN		12177	79,99	12177 artistes n'ont pas de code ULAN
Artworks.csv				
Title	Untitled	8387	5,57	8387 oeuvres n'ont pas de nom
DisplayName	Eugène Atget	5050	3,35	5050 oeuvres ont été créées par Eugène Atget
ConstituentID	229	5050	3,35	
ArtistBio		5925	3,93	5925 artistes n'ont pas de biographie
Nationality	American	56963	37,80	
BeginDate	0	8115	5,39	3597 artistes ont une date de naissance inconnue
EndDate	0	46071	30,57	10084 artistes sont encore en vie ou date de décès inconnue
Gender	Male	104140	69,11	
Date		2198	1,46	2198 oeuvres n'ont pas de date connue
Medium	Gelatin silver print	13235	8,78	
Dimensions	Approx. 8 11/16 × 7 1/16" (22 × 18 cm)	1296	0,86	
CreditLine	The Louis E. Stern Collection	11258	7,47	
AccessionNumber	107.1956.1	1	0,00	Chaque valeur est unique
Classification	Photograph	33025	21,92	
Department	Drawings & Prints	76117	50,51	
DateAcquired	1964-10-06	11520	7.65	
Cataloged	Y	87184	57,86	87184 oeuvres sont cataloguées contre 50967 non cataloguées
ObjectID	100	1	0,00	Chaque identifiant est unique
URL		50967	33,82	50967 n'ont pas d'url, les autres sont uniques
ThumbnailURL		61742	40,98	
Circumference (cm)		138141	91,68	
Depth (cm)		124312	82,50	
Diameter (cm)		136689	90,71	
Height (cm)		17796	11,81	
Length (cm)		137409	91,19	
Weight (kg)		137861	91,49	
Width (cm)		18717	12,42	
Seat Height (cm)		138151	91,68	
Duration (sec)		136011	90,26	

Colonne	Deuxième valeur la plus citée (si première absente)	Nombre	Proportion (%)
Table Artist			
DisplayName	John Wood	3	0,00
ArtistBio	American	387	0,03
BeginDate	1942	188	0,01
EndDate	1991	86	0,01
Wiki_QID	Q1994938	2	0,00
ULAN	500055849	2	0,00
Table Artwork			
Title	Poster of the French National Railroad	376	0,00
ArtistBio	French, 1857-1927	5050	0,03
BeginDate	1857	5111	0,03
EndDate	1927	5137	0,03
Date	1967	1846	0,01
URL	valeur trop longue	1	0,00
ThumbnailURL	valeur trop longue	1	0,00
Circumference (cm)	23.5	2	0,00
Depth (cm)	0.0	6785	0,05
Diameter (cm)	10.16	22	0,00
Height (cm)	0.0	2533	0,02
Length (cm)	0.0	59	0,00
Weight (kg)	4.536	6	0,00
Width (cm)	0.0	2385	0,02
Seat Height (cm)	valeur trop longue	1	0,00
Duration (sec)	120.0	239	0,00

c) *Dans Postgresql, créez la base de données, un premier rôle ayant pour fonction d'administrer la base de données, un deuxième rôle n'ayant que des droits de consultations ainsi que les tables telles que vous les avez modélisées dans la première partie de ce projet*

Script pour créer un rôle permettant d'administrer une base de données :

```
CREATE ROLE Admin
SUPERUSER
LOGIN;
```

Script pour créer un rôle n'ayant que des droits de consultations ainsi que les tables :

```
CREATE ROLE Visitor
NOCREATEDB
NOCREATEROLE
LOGIN;
```

d) *Expliquez les contraintes d'intégrité référentielle que vous avez créées*

Nous avons utilisé, pour chaque table, un Serial (Int Auto_Increment) en clé primaire, afin de ne pas avoir à redéfinir une clé unique manuellement à chaque fois. Nous avons choisi la cardinalité 1,n de Artist vers Artwork et Artist vers ArtisticCurrent car un artiste peut créer une à plusieurs œuvres, et peut appartenir à un ou plusieurs courants artistiques. Pour les cardinalités partant de Artwork, nous avons 1,1 pour les relations vers ArtisticCurrent et Museum car une œuvre ne peut appartenir qu'à un seul courant artistique et n'être disponible que dans un seul musée, et la cardinalité 1,n vers Artist car une oeuvre peut être créée par plusieurs artistes. Ensuite, pour les relations partant de ArtisticCurrent, nous avons choisi 1,n vers Artist et Artwork car à un artiste ou une œuvre, peut contenir plusieurs courants artisiques. Enfin, on a choisi la cardinalité 1,n de Museum vers Artwork car le musée doit contenir une ou plusieurs œuvres. Ces cardinalités ont impliqué la migration de clés et la création des tables associatives "Belongs to" et "Creates".

e) *Développez et exécutez les scripts en Python permettant de remplir les tables avec les informations disponibles dans ces deux fichiers*

Pré-processing :
Le fichier d'entrée Artworks.csv n'est pas rigoureusement homogène. Par exemple, si nous ouvrons le fichier Artworks.csv sous un éditeur comme vim, on peut se rendre compte que ce fichier possède, en plein milieu des lignes, des "Carriage Return Line Feed" (CRLF), visibles sous forme de ^M. Nous avons alors saisi ces commandes sur vim afin d'homogénéiser le fichier :

```
vim Artworks.csv
:%s/^M\n//g
:%s/(\n\n/(/g
:%s/^M //g
:%s/>\n/>/g
:%s/''//g
:%s/'//g
:%s/|//g
```

Ces commandes prennent en compte une expression régulière entre s/ et le deuxième slash, et cette expression sera remplacée par celle entre le deuxième slash et le /g. Une phase de post-processing est également nécessaire sur le fichier fillCreates.sql, pour plus d'informations, consulter le fichier "readme.md" présent dans le dossier du projet.

Le script Python est disponible dans le dossier du projet > Script Python.

Partie 1.4 : Requêtes SQL

Ces requêtes sont présentes dans le dossier du projet > Scripts SQL > Requêtes.

a) *Afficher la liste des œuvres des artistes nés après la seconde guerre mondiale*

```
SELECT c.Title
FROM Artist a, Creates b, Artwork c
WHERE a.ConstituentID = b.ConstituentID
AND b.ObjectID = c.ObjectID
AND a.BeginDate > 1945;
```

b) *Afficher le nom et le nombre d'œuvres pour chaque artiste ayant effectué des Gelatin silver print. Seuls les 10 premiers artistes au maximum sont affichés dans l'ordre décroissant du nombre de fois.*

```
SELECT a.DisplayName, COUNT(*)
FROM Artist a, Creates b, Artwork c
Where a.ConstituentID = b.ConstituentID
AND b.ObjectID = c.ObjectID
And position('gelatin silver print' in c.Medium) != 0
AND a.DisplayName != 'Unknown photographer'
GROUP BY a.DisplayName
ORDER BY COUNT
DESC
LIMIT 10;
```

c) *Créez une vue matérialisée permettant d'afficher le nombre d'œuvres répertoriées pour chacun des artistes*

Script SQL permettant de créer une vue :

```
CREATE VIEW View
AS SELECT a.DisplayName, COUNT(*)
FROM Artist a, Creates b, Artwork c
Where a.ConstituentID = b.ConstituentID
AND b.ObjectID = c.ObjectID
GROUP BY a.DisplayName;
```

Script SQL permettant d'afficher cette vue :

```
SELECT *
FROM View;
```

d) Créez un déclencheur (Trigger) qui produit un message d'erreur dans le SGBD si la date de naissance d'un des artistes est inférieur à l'an 1000

Création d'une fonction à appeler dans la création du trigger :

```
CREATE FUNCTION trigger_call()
    RETURNS TRIGGER
    LANGUAGE PLPGSQL
    AS
$$
BEGIN
    RAISE NOTICE 'ERROR: BeginDate is too low (<1000) !';
    RETURN NEW;
END;
$$
```

Script SQL pour créer le trigger :

```
CREATE TRIGGER trigger_artist
BEFORE INSERT ON Artist
FOR EACH ROW
WHEN (NEW.BeginDate < 1000)
EXECUTE PROCEDURE trigger_call();
```

Script SQL pour tester le trigger :

```
INSERT INTO Artist (ConstituentID, BeginDate)
VALUES ('1000000', '999')
# -> affichage de la notice "ERROR: BeginDate is too low (<1000) !"
INSERT INTO Artist (ConstituentID, BeginDate)
VALUES ('1000001', '1000'')
# -> pas d'affichage
```

e) Ajoutez un indexe à la table ayant le plus de données et montrez les différences de performances liées à la création de cet indexe

Script SQL permettant de créer l'index :

```
CREATE INDEX idxDateAcquired
ON Artwork(DateAcquired);
```

Après avoir saisi la commande "\timing on" dans l'invite de commandes, on peut à présent consulter le temps d'exécution de chaque requête.

Ensuite, on crée un index sur la colonne DateAcquired, nous permettant de diminuer les temps de requête sur cette colonne :

```
postgres=# CREATE INDEX idxDateAcquired ON Artwork(DateAcquired);
CREATE INDEX
Time: 317.283 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-06';
Time: 124.442 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-07';
Time: 0.791 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-08';
Time: 0.641 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-09';
Time: 0.706 ms
postgres=# DROP INDEX idxDateAcquired;
DROP INDEX
Time: 6.369 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-06';
Time: 228.264 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-07';
Time: 131.800 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-08';
Time: 126.215 ms
postgres=# SELECT * FROM Artwork WHERE DateAcquired = '1964-10-09';
Time: 130.989 ms
```

La création d'index prend beaucoup de temps, mais elle devient rentable à partir de 3 requêtes sur la colonne DateAcquired (317+124+1 = 442 ms < 228+131+126 = 485 ms).
Les temps de requête des 7, 8 et 9 octobre 1964 sont très courts pour les requêtes avec index, car le SELECT n'affiche rien (pas d'œuvre créée à ces dates), ce temps reste quand même élevé sans index.
Cependant, plus la table sera grande, plus l'index mettra de temps à se créer, et il ne sera rentable seulement si le nombre de requêtes effectuées est élevé.

Annexe - Tables générées non vides (premières lignes) :

*SELECT * FROM Artist*

ConstituentID | DisplayName | ArtistBio | Nationality | Gender | BeginDate | EndDate | idCurrent
---
1 | Robert Arneson | American, 1930–1992 | American | Male | 1930 | 1992 |
2 | Doroteo Arnaiz | Spanish, born 1936 | Spanish | Male | 1936 | 0 |
3 | Bill Arnold | American, born 1941 | American | Male | 1941 | 0 |
4 | Charles Arnoldi | American, born 1946 | American | Male | 1946 | 0 |
5 | Per Arnoldi | Danish, born 1941 | Danish | Male | 1941 | 0 |
6 | Danilo Aroldi | Italian, born 1925 | Italian | Male | 1925 | 0 |
7 | Bill Aron | American, born 1941 | American | Male | 1941 | 0 |
9 | David Aronson | American, born Lithuania 1923 | American | Male | 1923 | 0 |
10 | Irene Aronson | American, born Germany 1918 | American | Female | 1918 | 0 |
11 | Jean (Hans) Arp | French, born Germany (Alsace) 1886–1966 | French | Male | 1886 | 1966|
12 | Jüri Arrak | Estonian, born 1936 | Estonian | Male | 1936 | 0 |

*SELECT * FROM Artwork*

ObjectID | Type | Title | Classification | Date | URL | Dimensions | Medium | CreditLine | AccessionNumber | Department | DateAcquired | IDCurrent | IDMuseum
---
2 | | Ferdinandsbrücke Project, Vienna, Austria (Elevation, preliminary version) | Architecture | 1896 | http://www.moma.org/collection/works/2 | 19 1/8 x 66 1/2" (48.6 x 168.9 cm) | Ink and cut-and-pasted painted pages on paper | Fractional and promised gift of Jo Carole and Ronald S. Lauder | 885.1996 | Architecture & Design | 1996-04-09 | |
3 | | City of Music, National Superior Conservatory of Music and Dance, Paris, France, View from interior courtyard | Architecture | 1987 | http://www.moma.org/collection/works/3 | 16 x 11 3/4" (40.6 x 29.8 cm) | Paint and colored pencil on print | Gift of the architect in honor of Lily Auchincloss | 1.1995 | Architecture & Design | 1995-01-17 | |
4 | | Villa near Vienna Project, Outside Vienna, Austria, Elevation | Architecture | 1903 | http://www.moma.org/collection/works/4 | 13 1/2 x 12 1/2" (34.3 x 31.8 cm)| Graphite, pen, color pencil, ink, and gouache on tracing paper | Gift of Jo Carole and Ronald S. Lauder | 1.1997 | Architecture & Design | 1997-01-15 | |

*SELECT * FROM Creates*

ObjectID | ConstituentID
---
2 | 6210
3 | 7470
4 | 7605
5 | 7056
6 | 7605
7 | 7056
8 | 7056
9 | 7056