

Requêtes SQL Avancées & Techniques ETL

Compétences Techniques Démonstrées

- Processus ETL avec jointures complexes
- Agrégations avancées et fonctions statistiques
- Opérations ensemblistes (UNION, INTERSECT)
- Requêtes analytiques avec sous-requêtes corrélées
 - Validation de données et contraintes métier
- Gestion de vues matérialisées et DML optimisé

Table des matières

1	Architecture et Approche Data Engineering	2
2	Phase d'Extraction (Extract)	2
2.1	Extraction de données complexes avec jointures multiples	2
2.2	Extraction avec validation de contraintes métier	2
2.3	Extraction de données référentielles	2
3	Phase de Transformation (Transform)	3
3.1	Agrégations et fonctions statistiques	3
3.2	Agrégations par dimension métier	3
3.3	Analyse comparative avec moyennes	3
3.4	Transformations avec GROUP BY et HAVING	4
4	Gestion des Relations et Substitutions	4
4.1	Système de remplacement avec contraintes	4
4.2	Gestion des substituts produits	4
5	Jointures Avancées et Analyse de Couverture	4
5.1	Jointures internes vs externes	4
6	Opérations Ensemblistes - Analyse Combinée	5
6.1	Union des données	5
6.2	Intersections pour validation	5
7	Requêtes Analytiques Avancées	5
7.1	Analyse avec EXISTS et algèbre relationnelle	5
8	Phase de Chargement (Load) - Vues et Tables	6
8.1	Création de vues matérialisées	6
9	Opérations DML - Maintenance des Données	6
9.1	Insertion contrôlée	6
9.2	Mise à jour avec critères	6
9.3	Suppression avec filtrage	7
10	Synthèse des Compétences Data Engineering	7
10.1	Processus ETL	7
10.2	Techniques Analytiques	7
10.3	Optimisation et Performance	7

1. Architecture et Approche Data Engineering

Ce portfolio présente des requêtes SQL développées selon les principes du Data Engineering, organisées en processus ETL (Extract, Transform, Load) pour optimiser le traitement et l'analyse des données de l'Atelier du Père Noël.

2. Phase d'Extraction (Extract)

2.1. Extraction de données complexes avec jointures multiples

ETL - Extract Transform Load

Extraction complète des données produit avec leurs relations : ateliers de production et matières premières utilisées.

```

1  -- Affiche les infos d'un jouet (J001), avec ses ateliers et matieres
   premieres associees
2  SELECT j.Id_Jouet, j.Nom_Jouet, a.Id_Atelier, a.Nom_Atelier, mp.Id_MP, mp.
   Nom_MP
3  FROM Jouet j
4  LEFT JOIN Produit p ON j.Id_Jouet = p.Id_Jouet
5  LEFT JOIN Atelier a ON p.Id_Atelier = a.Id_Atelier
6  LEFT JOIN Utilise u ON j.Id_Jouet = u.Id_Jouet
7  LEFT JOIN Matiere_Premiere mp ON u.Id_MP = mp.Id_MP
8  WHERE j.Id_Jouet = 'J001';

```

Listing 1 – Extraction master des données jouet avec relations

2.2. Extraction avec validation de contraintes métier

Data Quality & Validation

Validation des règles métier : vérification que la charge des traîneaux respecte leur capacité réelle.

```

1  -- Affiche les traîneaux dont la charge ne dépasse pas la capacite reelle (
   somme des rennes atteles)
2  SELECT t.Id_Traineau, rt.Capacite_Reelle, ct.Charge_Totale
3  FROM Traineau t
4  JOIN (
5      SELECT a.Id_Traineau, SUM(r.Capacite_Charge) AS Capacite_Reelle
6      FROM Attelage a
7      JOIN Renne r ON a.Id_Renne = r.Id_Renne
8      GROUP BY a.Id_Traineau
9  ) rt ON t.Id_Traineau = rt.Id_Traineau
10 JOIN (
11     SELECT Id_Traineau, SUM(Poids_Total) AS Charge_Totale
12     FROM Charge
13     GROUP BY Id_Traineau
14 ) ct ON t.Id_Traineau = ct.Id_Traineau
15 WHERE ct.Charge_Totale <= rt.Capacite_Reelle;

```

Listing 2 – Extraction avec validation des capacités de transport

2.3. Extraction de données référentielles

```

1  -- Liste des elfes et des traîneaux dont ils s'occupent
2  SELECT e.Id_Elfe, e.Nom_Elfe, st.Id_Traineau
3  FROM Elfe e
4  JOIN Soccupe_de_T st ON e.Id_Elfe = st.Id_Elfe;

```

Listing 3 – Extraction des équipes - Elfes et traîneaux

```

1  -- Liste des elfes et des rennes dont ils s'occupent
2  SELECT e.Id_Elfe, e.Nom_Elfe, sr.Id_Renne
3  FROM Elfe e
4  JOIN Soccupe_de_R sr ON e.Id_Elfe = sr.Id_Elfe;

```

Listing 4 – Extraction des équipes - Elfes et rennes

3. Phase de Transformation (Transform)

3.1. Agrégations et fonctions statistiques

Analytics & Aggregation

Calcul d'indicateurs de performance avec fonctions MIN/MAX pour identifier les extrêmes de production.

```

1  -- Jouet(s) le(s) plus produit(s)
2  SELECT Id_Jouet, Quantite_Produite
3  FROM Produit
4  WHERE Quantite_Produite = (
5      SELECT MAX(Quantite_Produite) FROM Produit
6  );

```

Listing 5 – Identification des pics de production

```

1  -- Jouet(s) le(s) moins produit(s)
2  SELECT Id_Jouet, Quantite_Produite
3  FROM Produit
4  WHERE Quantite_Produite = (
5      SELECT MIN(Quantite_Produite) FROM Produit
6  );

```

Listing 6 – Identification des creux de production

3.2. Agrégations par dimension métier

```

1  -- Quantite totale de jouets produits par atelier
2  SELECT a.Nom_Atelier, SUM(p.Quantite_Produite) AS Total_Produits
3  FROM Produit p
4  JOIN Atelier a ON p.Id_Atelier = a.Id_Atelier
5  GROUP BY a.Nom_Atelier;

```

Listing 7 – Calcul de la production totale par atelier

3.3. Analyse comparative avec moyennes

Analytics & Aggregation

Utilisation de sous-requêtes pour identifier les anomalies par rapport à la moyenne.

```

1  -- Jouets dont le poids est superieur a la moyenne
2  SELECT *
3  FROM Jouet
4  WHERE Poids > (
5      SELECT AVG(Poids) FROM Jouet
6  );

```

Listing 8 – Identification des jouets au-dessus de la moyenne

3.4. Transformations avec GROUP BY et HAVING

```

1  -- Elfes ayant plus d'une specialite
2  SELECT e.Id_Elfe, e.Nom_Elfe, COUNT(ps.Id_Specialite) AS Nb_Specialites
3  FROM Elfe e
4  JOIN Possede_Spe ps ON e.Id_Elfe = ps.Id_Elfe
5  GROUP BY e.Id_Elfe, e.Nom_Elfe
6  HAVING COUNT(ps.Id_Specialite) > 1;

```

Listing 9 – Analyse des compétences multiples

4. Gestion des Relations et Substitutions

4.1. Système de remplacement avec contraintes

Data Quality & Validation

Requête complexe avec multiples contraintes pour assurer la validité des remplacements.

```

1  -- Liste des elfes valides pour remplacer chaque elfe absent
2  SELECT e1.Id_Elfe AS Elfe_Absent_ID, e1.Nom_Elfe AS Elfe_Absent_Nom,
3      e2.Id_Elfe AS Remplacant_ID, e2.Nom_Elfe AS Remplacant_Nom
4  FROM Elfe e1
5  JOIN Remplace r ON e1.Id_Elfe = r.Id_Elfe
6  JOIN Elfe e2 ON r.Id_Elfe_1 = e2.Id_Elfe
7  WHERE e1.Statut_Elfe = 'absent'
8      AND r.Equipe_Compatible = '1'
9      AND r.Spe_Compatible = '1';

```

Listing 10 – Calcul des remplaçants valides

4.2. Gestion des substituts produits

```

1  -- Liste des jouets non fabricables et leurs substituts
2  SELECT j.Nom_Jouet AS Jouet_Non_Fabricable, js.Nom_Jouet AS Substitut
3  FROM Jouet j
4  LEFT JOIN Peut_Substituer ps ON j.Id_Jouet = ps.Id_Jouet
5  LEFT JOIN Jouet js ON ps.Id_Jouet_1 = js.Id_Jouet
6  WHERE j.Statut_Jouet = 'non_fabricable';

```

Listing 11 – Mapping des substituts pour jouets non fabricables

5. Jointures Avancées et Analyse de Couverture

5.1. Jointures internes vs externes

```

1  -- Liste des jouets produits (avec leurs ateliers)
2  SELECT *
3  FROM Jouet j
4  INNER JOIN Produit p ON j.Id_Jouet = p.Id_Jouet;

```

Listing 12 – Jointure interne - Produits avec ateliers

```

1  -- Tous les ateliers, meme ceux qui n'ont produit aucun jouet
2  SELECT a.Id_Atelier, a.Nom_Atelier, p.Id_Jouet, p.Quantite_Produite
3  FROM Produit p
4  RIGHT JOIN Atelier a ON p.Id_Atelier = a.Id_Atelier;

```

Listing 13 – Jointure externe - Couverture complète des ateliers

6. Opérations Ensemblistes - Analyse Combinée

6.1. Union des données

Analytics & Aggregation

Consolidation de données provenant de sources multiples pour vue unifiée.

```

1  -- UNION : tous les jouets produits ou fournis
2  SELECT Id_Jouet FROM Produit
3  UNION
4  SELECT Id_Jouet FROM Fournit;

```

Listing 14 – Union des jouets produits et fournis

6.2. Intersections pour validation

```

1  -- INTERSECT : jouets presents ET produits
2  SELECT Id_Jouet FROM Jouet
3  INTERSECT
4  SELECT Id_Jouet FROM Produit;

```

Listing 15 – Intersection - Jouets presents ET produits

```

1  -- INTERSECT : jouets presents ET fournis
2  SELECT Id_Jouet FROM Jouet
3  INTERSECT
4  SELECT Id_Jouet FROM Fournit;

```

Listing 16 – Intersection - Jouets presents ET fournis

7. Requêtes Analytiques Avancées

7.1. Analyse avec EXISTS et algèbre relationnelle

Analytics & Aggregation

Requête complexe utilisant EXISTS et MINUS pour identifier les elfes experts.

```

1  -- Elfes qui possèdent toutes les specialites secondaires
2  SELECT DISTINCT ps.Id_Elfe
3  FROM Possede_Spe ps
4  WHERE NOT EXISTS (
5      SELECT Id_Specialite
6      FROM Specialite
7      WHERE Type_Specialite = 'secondaire'
8      MINUS
9      SELECT ps2.Id_Specialite
10     FROM Possede_Spe ps2
11     WHERE ps2.Id_Elfe = ps.Id_Elfe
12 );

```

Listing 17 – Identification des elfes maîtrisant toutes spécialités secondaires

8. Phase de Chargement (Load) - Vues et Tables

8.1. Création de vues matérialisées

ETL - Extract Transform Load

Création de vues optimisées pour l'accès récurrent aux données agrégées.

```

1  -- Vue 1 : quantite totale de jouets produits par atelier
2  CREATE VIEW vue_quantite_par_atelier AS
3  SELECT a.Id_Atelier, a.Nom_Atelier, SUM(p.Quantite_Produite) AS
4      Total_Produits
5  FROM Atelier a, Produit p
6  WHERE a.Id_Atelier = p.Id_Atelier
7  GROUP BY a.Id_Atelier, a.Nom_Atelier;

```

Listing 18 – Vue matérialisée - Agrégation par atelier

```

1  -- Vue 2 : specialites possedees par chaque elfe
2  CREATE VIEW vue_elfe_specialite AS
3  SELECT e.Id_Elfe, e.Nom_Elfe, s.Nom_Specialite, s.Type_Specialite
4  FROM Elfe e, Possede_Spe ps, Specialite s
5  WHERE e.Id_Elfe = ps.Id_Elfe
6  AND ps.Id_Specialite = s.Id_Specialite;

```

Listing 19 – Vue relationnelle - Compétences des elfes

9. Opérations DML - Maintenance des Données

9.1. Insertion contrôlée

```

1  -- Ajout d'un jouet (ex. pour test)
2  INSERT INTO Jouet (Id_Jouet, Nom_Jouet, Statut_Jouet, Poids)
3  VALUES ('J900', 'Pokemon', 'fabricable', 180);

```

Listing 20 – Insertion de nouvelles données

9.2. Mise à jour avec critères

```

1  -- Mise a jour : augmente de 10% le poids des jouets Pokemon
2  UPDATE Jouet
3  SET Poids = Poids * 1.1

```

```
4 WHERE UPPER(Nom_Jouet) LIKE '%POKEMON%';
```

Listing 21 – Mise à jour conditionnelle avec calcul

9.3. Suppression avec filtrage

```
1 -- Suppression : supprime les jouets liés à Pokemon
2 DELETE FROM Jouet
3 WHERE UPPER(Nom_Jouet) LIKE '%POKEMON%';
```

Listing 22 – Suppression conditionnelle

10. Synthèse des Compétences Data Engineering

Ce portfolio démontre la maîtrise des techniques essentielles du Data Engineering avec SQL :

10.1. Processus ETL

- **Extract** : Jointures complexes, filtrage métier, validation de contraintes
- **Transform** : Agrégations avancées, fonctions statistiques, calculs métier
- **Load** : Vues matérialisées, optimisation d'accès, structures dénormalisées

10.2. Techniques Analytiques

- **Agrégations** : SUM, COUNT, MAX, MIN, AVG avec GROUP BY/HAVING
- **Sous-requêtes** : Corrélées et non corrélées pour comparaisons
- **Opérations ensemblistes** : UNION, INTERSECT, MINUS
- **Fonctions avancées** : EXISTS, NOT EXISTS pour logique complexe

10.3. Optimisation et Performance

- **Jointures optimisées** : LEFT/RIGHT/INNER selon les besoins
- **Indexation implicite** : Requêtes structurées pour performance
- **Vues matérialisées** : Pré-calcul des agrégations coûteuses
- **DML optimisé** : INSERT/UPDATE/DELETE avec critères précis