### **TEMA: L3. Analytical Integration Model: Implementare OLAP views**

### Indicativ\_echipa: **SIA\_02**

Descrieți modul de implementare pentru fiecare structură de integrare (view-de-consolidare, view-analitic, eventual elemente de logica afacerii - funcții și proceduri-utilizator stocate) într-un fișier sintetic trimis la adresa linus@uaic.ro (adăugați în mesaj toate celelalte fișiere de creare/configurare pe care le considerați necesare):

* **(2) Schema analitică ROLAP - dacă urmați strategia bazelor de date multidimensionale din cursul T2.2.FDB\_ORCL.Integration\_OLAP\_Views:**
  + **Tabele/view-uri cu date cantitative agregabile (Tabele de fapte - ROLAP Fact Views)**
    - Tabela\_de\_fapte\_1: Valoarea și greutatea totala a coletelor expiate în funcție de expeditori și data de expediere
      * Surse de date integrate
    - 5\_Facturi [JSON] + 6\_Detalii\_fact [XML] + 4\_Colete [Rest Mongo]
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_FACTURI\_EXPEDITORI;

CREATE OR REPLACE VIEW OLAP\_FACTURI\_EXPEDITORI AS

SELECT f.id\_expeditor, f.data\_tiparire, SUM(c.valoare) AS valoare\_totala, SUM(c.greutate) AS greutate\_totala

FROM FACTURI\_VIEW f

INNER JOIN DET\_FACT\_VIEW df

ON f.nr\_factura = df.nr\_factura

INNER JOIN COLETE\_VIEW c ON df.id\_colet=c.id\_colet

GROUP BY f.id\_expeditor, f.data\_tiparire ;

SELECT \* FROM OLAP\_FACTURI\_EXPEDITORI;

* + - Tabela\_de\_fapte\_2: Valoarea totala a coletelor livrate de către fiecare șofer
      * Surse de date integrate
      * 7\_Șoferi [CSV] + 10\_Transporturi [SQL] + 11\_Linii\_transport [XML] + 6\_Detalii\_fact [XML] + [Rest Mongo]
      * Definiție: fraza DDL SQL

CREATE OR REPLACE VIEW OLAP\_COLETE\_TRANSPORTURI AS

SELECT t.id\_sofer,t.id\_vehicul, id\_depozit, SUM(c.valoare) AS Valoare\_totala, SUM(c.greutate) AS Greutate\_totala, COUNT(c.id\_colet) AS Colete

FROM Transporturi\_view t

INNER JOIN det\_transp dt ON t.id\_transport=dt.id\_transport

INNER JOIN det\_fact\_view df ON dt.nr\_factura=df.nr\_factura

INNER JOIN COLETE\_VIEW c ON df.id\_colet=c.id\_colet

GROUP BY t.id\_sofer,t.id\_vehicul,id\_depozit;

SELECT \* FROM OLAP\_COLETE\_TRANSPORTURI;

* + - Tabela\_de\_fapte\_3: Valoarea, greutatea și numărul coletelor primite de către fiecare destinatar
      * Surse de date integrate
    - 5\_Facturi [JSON] + 6\_Detalii\_fact [XML] + 4\_Colete [Rest Mongo]
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_FACTURI\_DESTINATARI;

CREATE OR REPLACE VIEW OLAP\_FACTURI\_DESTINATARI AS

SELECT f.id\_destinatar, f.data\_tiparire, SUM(c.valoare) AS valoare\_totala, SUM(c.greutate) AS greutate\_totala, COUNT(f.id\_destinatar) as colete

FROM FACTURI\_VIEW f

INNER JOIN DET\_FACT\_VIEW df

ON f.nr\_factura = df.nr\_factura

INNER JOIN COLETE\_VIEW c ON df.id\_colet=c.id\_colet

GROUP BY f.id\_destinatar, f.data\_tiparire ;

SELECT \* FROM OLAP\_FACTURI\_DESTINATARI;

* + **Tabele cu criterii/categorii de agregare (Tabele/view-uri dimensionale - OLAP Dimensional Views )**
    - Tabela\_dimensionala\_1: Județul de proveniență al expeditorilor
      * Surse de date integrate
        + 2\_Expeditori [CSV] + 1\_Coduri\_postale [SQL]
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_JUDET\_EXPEDITORI;

CREATE OR REPLACE VIEW OLAP\_DIM\_JUDET\_EXPEDITORI AS

SELECT e.id\_expeditor,

e.nume||' '||e.prenume AS Expeditor,

e.mediu,

c.judet AS Judet

FROM expeditori e

INNER JOIN cpostale\_view c ON e.cod\_postal=c.cod\_postal;

SELECT \* FROM OLAP\_DIM\_JUDET\_EXPEDITORI;

* + - Tabela\_dimensionala\_2: Județul de proveniență al șoferilor și genul acestora
      * Surse de date integrate
        + 7\_Șoferi [CSV] + 1\_Coduri\_postale [SQL]
      * Definiție: fraza DDL SQL

CREATE OR REPLACE VIEW OLAP\_DIM\_SOFERI\_JUDET AS

SELECT s.id\_sofer,

s.nume||' '||s.prenume AS Sofer,

gen,

probleme\_sanatate,

c.judet AS Judet

FROM soferi s

INNER JOIN cpostale\_view c ON s.cod\_postal=c.cod\_postal;

SELECT \* FROM OLAP\_DIM\_SOFERI\_JUDET;

* + - Tabela\_dimensionala\_3: Categoria de permis și situația problemelor de sanatate a șoferilor cu vârsta cuprinsă între 35 și 50 de ani.
      * Surse de date integrate
        + 7\_Șoferi [CSV] + 10\_Transporturi [SQL] + 9\_Depozite [JSON].
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_SOFERI\_DEPOZITE;

CREATE OR REPLACE VIEW OLAP\_DIM\_SOFERI\_DEPOZITE AS

SELECT s.id\_sofer,

s.nume||' '||s.prenume AS sofer,

s.categorie\_permis,

probleme\_sanatate,

d.id\_depozit

FROM soferi s

INNER JOIN transporturi\_view t ON s.id\_sofer=t.id\_sofer

INNER JOIN depozite\_view d ON t.id\_depozit=d.id\_depozit

WHERE s.varsta BETWEEN 35 AND 50;

SELECT \* FROM OLAP\_DIM\_SOFERI\_DEPOZITE;

* + - Tabela\_dimensionala\_4: Detalii privind marca, carburantul utilizat și tipul de transmisie al vehiculelor cu peste 250000 de km utilizate de șoferi.
      * Surse de date integrate
        + 7\_Șoferi [CSV] + 8\_Vehicule [XLSX]
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_SOFERI\_VEHICULE;

CREATE OR REPLACE VIEW OLAP\_DIM\_SOFERI\_VEHICULE AS

SELECT s.id\_sofer,

v."marca" as marca,

v."istoric\_accident" as istoric\_accident,

v."carburant" as carburant,

v."probleme\_tehnice" as probleme\_tehnice,

v."tip\_transmisie" as tip\_transmisie

FROM soferi s

INNER JOIN transporturi\_view t ON s.id\_sofer=t.id\_sofer

INNER JOIN courier\_vehicule v ON t.id\_vehicul=v."id\_vehicul"

WHERE v."km">=250000;

SELECT \* FROM OLAP\_DIM\_SOFERI\_VEHICULE;

* + - Tabela\_dimensionala\_5: Detalii privind vehiculele ce au cutie de transmisie manuală și sunt conduse de către șoferi.
      * Surse de date integrate
        + 7\_Șoferi [CSV] + 8\_Vehicule [XLSX]
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_SOFERI\_VEH\_DETALIAT;

CREATE OR REPLACE VIEW OLAP\_DIM\_SOFERI\_VEH\_DETALIAT AS

SELECT s.id\_sofer,

s.nume||' '||s.prenume as sofer,

s.nivel\_educational,

s.gen,

v."marca" as marca,

v."categorie\_vehicul" as categorie\_vehicul,

v."carburant" as carburant,

v."probleme\_tehnice" as probleme\_tehnice,

v."tip\_transmisie" as tip\_transmisie

FROM soferi s

INNER JOIN transporturi\_view t ON s.id\_sofer=t.id\_sofer

INNER JOIN courier\_vehicule v ON t.id\_vehicul=v."id\_vehicul"

WHERE v."tip\_transmisie" LIKE 'M';

SELECT \* FROM OLAP\_DIM\_SOFERI\_VEH\_DETALIAT;

* + - Tabela\_dimensionala\_6: Județul de proveniență și localitatea destinatarilor
      * Surse de date integrate
        + 3\_Destinatari [XLSX] + 1\_Coduri\_postale [SQL]
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_LOC\_DESTINATARI;

CREATE OR REPLACE VIEW OLAP\_DIM\_LOC\_DESTINATARI AS

SELECT d."id\_destinatar" as id\_destinatar,

d."nume"||' '||d."prenume" AS destinatar,

d."mediu" as mediu,

cp.localitate,

cp.judet

FROM courier\_destinatari d

INNER JOIN cpostale\_view cp ON d."cod\_postal"=cp.cod\_postal;

SELECT \* FROM OLAP\_DIM\_LOC\_DESTINATARI;

* + **Tabele/view-uri cu agregări analitice (Tabele/view-uri dimensionale cu funcții de agregare, statistice, clauze ROLLUP, CUBE etc. - OLAP Analytical Views**)
    - View\_Analitic\_OLAP\_1: Situația coletelor expediate din fiecare județ în parte
      * Tip procesare analitică: clauza ROLLUP.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_COLETE\_EXPEDITORI;

CREATE OR REPLACE VIEW OLAP\_VIEW\_COLETE\_EXPEDITORI AS

SELECT

CASE

WHEN GROUPING(j.judet) = 1 THEN '{Total General}'

ELSE j.judet END AS Judet,

CASE

WHEN GROUPING(j.judet) = 1 THEN ' '

WHEN GROUPING(j.expeditor) = 1 THEN 'Subtotal Judet ' || j.judet

ELSE to\_char(j.expeditor) END AS Expeditor,

SUM(NVL(f.valoare\_totala, 0)) as valoare\_colete

FROM OLAP\_DIM\_JUDET\_EXPEDITORI J

INNER JOIN OLAP\_FACTURI\_EXPEDITORI F ON j.id\_expeditor=f.id\_expeditor

GROUP BY ROLLUP (j.judet, j.expeditor)

ORDER BY j.judet, j.expeditor;

SELECT \* FROM OLAP\_VIEW\_COLETE\_EXPEDITORI;

* + - View\_Analitic\_OLAP\_2: Situația coletelor livrate de către fiecare șofer în parte în funcție de genul și județul de proveniență al acestora.
      * Tip procesare analitică: clauza ROLLUP.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_COLETE\_SOFERI;

CREATE OR REPLACE VIEW OLAP\_VIEW\_COLETE\_SOFERI AS

SELECT

CASE

WHEN GROUPING(j.gen) = 1 THEN '{Total General}'

ELSE j.gen END AS Gen,

CASE

WHEN GROUPING(j.gen) = 1 THEN ' '

WHEN GROUPING(j.judet) = 1 THEN 'Subtotal Gen ' || j.gen

ELSE j.judet END AS Judet,

CASE

WHEN GROUPING(j.gen) = 1 THEN ' '

WHEN GROUPING(j.judet) = 1 THEN ' '

WHEN GROUPING(j.sofer) = 1 THEN 'Subtotal ' || j.judet

ELSE to\_char(j.sofer) END AS Sofer,

SUM(NVL(f.valoare\_totala, 0)) as valoare\_colete

FROM OLAP\_DIM\_SOFERI\_JUDET J

INNER JOIN OLAP\_COLETE\_TRANSPORTURI F ON j.id\_sofer=f.id\_sofer

GROUP BY ROLLUP (j.gen, j.judet,j.sofer)

ORDER BY j.gen, j.judet,j.sofer;

SELECT \* FROM OLAP\_VIEW\_COLETE\_SOFERI;

* + - View\_Analitic\_OLAP\_3: Numărul de colete ce a plecat din fiecare depozit cu șoferi ce au vârsta cuprinsă între 35 – 50 de ani în funcție de problemele de sănătate și de categoria de permis.
      * Tip procesare analitică: clauza ROLLUP.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_SOFERI\_DEPOZIT;

CREATE OR REPLACE VIEW OLAP\_VIEW\_SOFERI\_DEPOZIT AS

SELECT

CASE

WHEN GROUPING(s.id\_depozit) = 1 THEN '{Total General}'

ELSE s.id\_depozit END AS Id\_depozit,

CASE

WHEN GROUPING(s.id\_depozit) = 1 THEN ' '

WHEN GROUPING(s.categorie\_permis) = 1 THEN 'Subtotal Depozit ' || s.id\_depozit

ELSE s.categorie\_permis END AS Categorie\_permis ,

CASE

WHEN GROUPING(s.id\_depozit) = 1 THEN ' '

WHEN GROUPING(s.categorie\_permis) = 1 THEN ' '

WHEN GROUPING(s.probleme\_sanatate) = 1 THEN 'Subtotal ' || s.categorie\_permis

ELSE to\_char(s.probleme\_sanatate) END AS Probleme\_sanatate,

SUM(NVL(f.colete, 0)) as valoare\_colete

FROM OLAP\_DIM\_SOFERI\_DEPOZITE s

INNER JOIN OLAP\_COLETE\_TRANSPORTURI F ON s.id\_sofer=f.id\_sofer

GROUP BY ROLLUP (s.id\_depozit,categorie\_permis,probleme\_sanatate)

ORDER BY s.id\_depozit,categorie\_permis,probleme\_sanatate;

SELECT \* FROM OLAP\_VIEW\_SOFERI\_DEPOZIT;

* + - View\_Analitic\_OLAP\_4: Numărul de colete livrate de cu vehiculele ce au peste 250.000 de km, în funcție de carburantul, marca și tipul transmisiei
      * Tip procesare analitică: clauza ROLLUP.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_SOFERI\_VEHICULE;

CREATE OR REPLACE VIEW OLAP\_VIEW\_SOFERI\_VEHICULE AS

SELECT

CASE

WHEN GROUPING(s.carburant) = 1 THEN '{Total General}'

ELSE s.carburant END AS Carburant,

CASE

WHEN GROUPING(s.carburant) = 1 THEN ' '

WHEN GROUPING(s.marca) = 1 THEN 'Subtotal ' || s.carburant

ELSE s.marca END AS Marca ,

CASE

WHEN GROUPING(s.carburant) = 1 THEN ' '

WHEN GROUPING(s.marca) = 1 THEN ' '

WHEN GROUPING(s.tip\_transmisie) = 1 THEN 'Subtotal ' || s.Marca

ELSE to\_char(s.tip\_transmisie) END AS Tip\_transmisie,

SUM(NVL(f.colete, 0)) as numar\_colete

FROM OLAP\_DIM\_SOFERI\_VEHICULE s

INNER JOIN OLAP\_COLETE\_TRANSPORTURI F ON s.id\_sofer=f.id\_sofer

GROUP BY ROLLUP (carburant, marca, tip\_transmisie)

ORDER BY carburant, marca, tip\_transmisie;

SELECT \* FROM OLAP\_VIEW\_SOFERI\_VEHICULE;

* + - View\_Analitic\_OLAP\_5: Greutatea coletelor livrate cu vehicule ce au cutie de transmisie Manuală în funcție de genul șoferilor, nivelul educațional și problemele tehnice ale vehiculelor
      * Tip procesare analitică: clauza ROLLUP.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_SOFERI\_VEHICULE\_DET;

CREATE OR REPLACE VIEW OLAP\_VIEW\_SOFERI\_VEHICULE\_DET AS

SELECT

CASE

WHEN GROUPING(s.gen) = 1 THEN '{Total General}'

ELSE s.gen END AS Gen,

CASE

WHEN GROUPING(s.gen) = 1 THEN ' '

WHEN GROUPING(s.nivel\_educational) = 1 THEN 'Subtotal ' || s.gen

ELSE s.nivel\_educational END AS nivel\_educational ,

CASE

WHEN GROUPING(s.gen) = 1 THEN ' '

WHEN GROUPING(s.nivel\_educational) = 1 THEN ' '

WHEN GROUPING(s.sofer) = 1 THEN 'Subtotal ' || s.nivel\_educational

ELSE to\_char(s.sofer) END AS Sofer,

SUM(NVL(f.greutate\_totala, 0)) as greutate\_colete

FROM OLAP\_DIM\_SOFERI\_VEH\_DETALIAT s

INNER JOIN OLAP\_COLETE\_TRANSPORTURI F ON s.id\_sofer=f.id\_sofer

GROUP BY ROLLUP (gen, nivel\_educational,sofer)

ORDER BY gen, nivel\_educational,sofer;

SELECT \* FROM OLAP\_VIEW\_SOFERI\_VEHICULE\_DET;

* + - View\_Analitic\_OLAP\_6: Numarul coletelor primite de către destinatarii din fiecare județ
      * Tip procesare analitică: clauza ROLLUP.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_COLETE\_DESTINATARI

CREATE OR REPLACE VIEW OLAP\_VIEW\_COLETE\_DESTINATARI AS

SELECT

CASE

WHEN GROUPING(j.judet) = 1 THEN '{Total General}'

ELSE j.judet END AS Judet,

CASE

WHEN GROUPING(j.judet) = 1 THEN ' '

WHEN GROUPING(j.localitate) = 1 THEN 'Subtotal Judet ' || j.judet

ELSE j.localitate END AS Localitate,

CASE

WHEN GROUPING(j.judet) = 1 THEN ' '

WHEN GROUPING(j.localitate) = 1 THEN ' '

WHEN GROUPING(j.destinatar) = 1 THEN 'Subtotal ' || j.localitate

ELSE to\_char(j.destinatar) END AS Destinatar,

SUM(NVL(f.colete, 0)) as colete

FROM OLAP\_DIM\_LOC\_DESTINATARI J

INNER JOIN OLAP\_FACTURI\_DESTINATARI F ON j.id\_destinatar=f.id\_destinatar

GROUP BY ROLLUP (j.judet,j.localitate,j.destinatar)

ORDER BY j.judet,j.localitate,j.destinatar;

SELECT \* FROM OLAP\_VIEW\_COLETE\_DESTINATARI;

**Observație**: Funcție de strategia pe care ați decis să o urmați, puteți construi un model analitc simplu - fără valențe ROLAP **SAU** (pentru un punctaj de evaluare mai bun) puteți alege un model dimensional mai complex.

* **Interogări analitice simple:**
* **Clasamentul destinatarilor dupa valoarea coletelor primite**

**Surse de date integrate:** 3\_Destinatari [XLSX] + 5\_Facturi [JSON] + 6\_Detalii\_fact [XML] + 4\_Colete [REST JSON]

**Tip procesare analitica:** functie analitica RANK()

**DDL SQL:**

SELECT d."nume"||' '||d."prenume" AS Destinatar,

SUM(c.valoare) AS valoare\_totala,

RANK() OVER(ORDER BY SUM(c.valoare) DESC) AS POZ

FROM courier\_destinatari d

INNER JOIN FACTURI\_VIEW f ON d."id\_destinatar"=f.id\_destinatar

INNER JOIN det\_fact\_view df ON f.nr\_factura=df.nr\_factura

INNER JOIN colete\_view c ON df.id\_colet=c.id\_colet

GROUP BY d."nume"||' '||d."prenume";

* **Valoarea coletelor expediate din fiecare judet și mediu**

**Surse de date integrate:** 1\_Coduri\_postale [SQL] + 2\_Expeditori [CSV] + 5\_Facturi [JSON] + 6\_Detalii\_fact [XML] + 4\_Colete [REST JSON]

**Tip procesare analitica:** functie analitica RANK()

**DDL SQL:**

SELECT cp.judet,

e.mediu, SUM(c.valoare) AS valoare\_totala

FROM cpostale\_view cp

INNER JOIN expeditori e ON cp.cod\_postal=e.cod\_postal

INNER JOIN FACTURI\_VIEW f ON e.id\_expeditor=f.id\_expeditor

INNER JOIN det\_fact\_view df ON f.nr\_factura=df.nr\_factura

INNER JOIN colete\_view c ON df.id\_colet=c.id\_colet

GROUP BY ROLLUP(cp.judet,e.mediu)

ORDER BY 1,2;

* **Clasamentul soferilor pe baza greutatii coletelor livrate**

**Surse de date integrate:** 7\_Soferi [CSV] + 10\_Transporturi [SQL] + 11\_Linii\_transport [XML] + 6\_Detalii\_fact [XML] + 4\_Colete [REST JSON]

**Tip procesare analitica:** clauza GROUP BY

**DDL SQL:**

SELECT s.nume||' '||s.prenume AS Sofer, SUM(c.greutate) as greutate\_totala

FROM Soferi s

INNER JOIN Transporturi\_view t ON s.id\_sofer=t.id\_sofer

INNER JOIN DET\_TRANSP dt ON t.id\_transport=dt.id\_transport

INNER JOIN DET\_FACT\_VIEW df ON dt.nr\_factura=df.nr\_factura

INNER JOIN COLETE\_VIEW c ON df.id\_colet=c.id\_colet

GROUP BY s.nume||' '||s.prenume

ORDER BY SUM(c.greutate) DESC;

* **Raport pe baza valorii totale a coletelor primite de destinatarii din fiecare judet și mediu**

**Surse de date integrate:** 1\_Coduri\_postale [SQL] + 3\_Destinatari [XLSX] + 5\_Facturi [JSON] + 6\_Detalii\_fact [XML] + 4\_Colete [REST JSON]

**Tip procesare analitica:** operatorul CUBE

**DDL SQL:**

SELECT

cp.judet,

cp.localitate,

d."mediu" as mediu,

SUM(c.valoare) AS valoare\_totala

FROM cpostale\_view cp

INNER JOIN courier\_destinatari d ON cp.cod\_postal=d."cod\_postal"

INNER JOIN FACTURI\_VIEW f ON d."id\_destinatar"=f.id\_destinatar

INNER JOIN det\_fact\_view df ON f.nr\_factura=df.nr\_factura

INNER JOIN colete\_view c ON df.id\_colet=c.id\_colet

GROUP BY CUBE(cp.judet,cp.localitate,d."mediu")

ORDER BY 1,2

FETCH NEXT 90 ROWS ONLY;;