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

### Indicativ\_echipa: **SIA\_11**

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):

* **(1) Nivel CONSOLIDARE date - dacă este cazul, de exemplu joncționare(JOIN) a structurilor de access la surse de date diferite (view-uri, tabele externe)**
  + **View\_Consolidare\_1**: Nume/descriere
    - Surse de date integrate
      * DS\_x + DS\_y + DS\_z
    - Definiție: fraza DDL SQL
  + **View\_Consolidare\_2**: Nume/descriere
    - Surse de date integrate
      * DS\_x + DS\_y + DS\_z
    - Definiție: fraza DDL SQL
  + **View\_Consolidare\_n…**
* **(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: OLAP\_FACT\_CATEGORY\_AMOUNT --Total amount of rental per car category
      * Surse de date integrate
        + DS\_x + DS\_y + DS\_z
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_FACT\_CATEGORY\_AMOUNT;

CREATE OR REPLACE VIEW OLAP\_FACT\_CATEGORY\_AMOUNT AS

SELECT cc.Label,COUNT(r.Car\_ID) AS Total\_rent, SUM(r.Amount) AS Total\_amount

FROM rentals\_view r

INNER JOIN cars c ON r.Car\_ID = c.Car\_ID

INNER JOIN car\_category\_view cc ON c.Category\_ID = cc.Category\_ID

GROUP BY cc.Label;

SELECT \* FROM OLAP\_FACT\_CATEGORY\_AMOUNT;

* + - Tabela\_de\_fapte\_2: OLAP\_FACT\_CAT\_CAR\_AMOUNT
      * Surse de date integrate
        + DS\_x + DS\_y + DS\_z
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_FACT\_CAT\_CAR\_AMOUNT;

CREATE OR REPLACE VIEW OLAP\_FACT\_CAT\_CAR\_AMOUNT AS

SELECT cc.Label,r.Car\_ID, COUNT(r.Car\_ID) AS Total\_rent, SUM(r.Amount) AS Total\_amount

FROM rentals\_view r

INNER JOIN cars c ON r.Car\_ID = c.Car\_ID

INNER JOIN car\_category\_view cc ON c.Category\_ID = cc.Category\_ID

GROUP BY cc.Label, r.Car\_ID

ORDER BY cc.Label;

SELECT \* FROM OLAP\_FACT\_CAT\_CAR\_AMOUNT;

* + **Tabele cu criterii/categorii de agregare (Tabele/view-uri dimensionale - OLAP Dimensional Views )**
    - Tabela\_dimensionala\_1: OLAP\_FACT\_CAT\_CAR\_AMOUNT -- Customer Car Location
      * Surse de date integrate
        + DS\_x + DS\_y + DS\_z
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_CUSTS\_CAR\_LOC;

CREATE OR REPLACE VIEW OLAP\_DIM\_CUSTS\_CAR\_LOC AS

SELECT

c."Customer\_ID" as customer\_id, c."First\_Name" as customer\_name, -- L1

cr.Car\_ID as car\_id, cr.Brand as car\_brand, cr.Model as car\_model, -- L2

l."Location\_ID" as location\_id, l."City" as city\_name -- L3

FROM customers\_view c

INNER JOIN rentals\_view r ON c."Customer\_ID" = r.Customer\_ID

INNER JOIN Cars cr ON cr.Car\_ID = r.Car\_ID

INNER JOIN Location\_view l ON l."Location\_ID" = r.Pick\_up\_location

ORDER BY c."Customer\_ID";

SELECT \* FROM OLAP\_DIM\_CUSTS\_CAR\_LOC;

* + - Tabela\_dimensionala\_2: OLAP\_DIM\_CAT\_CAR\_LOC -- Car\_category Car Location
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_DIM\_CAT\_CAR\_LOC;

CREATE OR REPLACE VIEW OLAP\_DIM\_CAT\_CAR\_LOC AS

SELECT

cc.Category\_ID as category\_id, cc.Label as categorie, -- L1

cr.Car\_ID as car\_id, cr.Brand as car\_brand, cr.Model as car\_model, -- L2

l."Location\_ID" as location\_id, l."City" as city\_name, l."State" as state\_name -- L3

FROM car\_category\_view cc

INNER JOIN rentals\_view r ON cc.Category\_ID = r.Customer\_ID

INNER JOIN Cars cr ON cr.Car\_ID = r.Car\_ID

INNER JOIN Location\_view l ON l."Location\_ID" = r.Pick\_up\_location;

SELECT \* FROM OLAP\_DIM\_CUSTS\_CAR\_LOC;

* + **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: OLAP\_VIEW\_AMOUNT\_CUSTS\_CAR\_LOC – sales amount model car / city
      * Tip procesare analitică: operator-analitic/funcție de agregare etc.
      * Definiție: fraza DDL SQL

DROP VIEW OLAP\_VIEW\_AMOUNT\_CUSTS\_CAR\_LOC;

CREATE OR REPLACE VIEW OLAP\_VIEW\_AMOUNT\_CUSTS\_CAR\_LOC AS

SELECT

CASE

WHEN GROUPING(d.city\_name ) = 1 THEN '{Total General}'

ELSE d.city\_name END AS City\_name,

CASE

WHEN GROUPING(d.city\_name) = 1 THEN ' '

WHEN GROUPING(d.car\_brand) = 1 THEN 'subtotal city ' || d.car\_brand

ELSE d.car\_brand END AS Car\_brand,

SUM(NVL(f.total\_amount, 0)) as SALES\_AMOUNT

FROM OLAP\_DIM\_CUSTS\_CAR\_LOC D

INNER JOIN OLAP\_FACT\_CAT\_CAR\_AMOUNT F ON d.car\_id = F.Car\_Id

GROUP BY ROLLUP (d.city\_name, d.car\_brand)

ORDER BY d.city\_name, d.car\_brand;

SELECT \* FROM OLAP\_VIEW\_AMOUNT\_CUSTS\_CAR\_LOC;

* + - View\_Analitic\_OLAP\_2: OLAP\_VIEW\_RENT\_CAT\_CAR\_LOC
      * Definitie: fraza DDL SQL

CREATE OR REPLACE VIEW OLAP\_VIEW\_RENT\_CAT\_CAR\_LOC AS

SELECT

CASE

WHEN GROUPING(d.state\_name ) = 1 THEN '{Total General}'

ELSE d.state\_name END AS State\_name,

CASE

WHEN GROUPING(d.State\_name) = 1 THEN ' '

WHEN GROUPING(d.categorie) = 1 THEN 'subtotal State ' || d.categorie

ELSE d.categorie END AS Car\_category,

CASE

WHEN GROUPING(d.State\_name) = 1 THEN ' '

WHEN GROUPING(d.categorie) = 1 THEN ' '

WHEN GROUPING(d.car\_brand) = 1 THEN 'subtotal Categorie ' || d.car\_brand

ELSE d.car\_brand END AS Car\_brand,

SUM(NVL(f.total\_rent, 0)) as TOTAL\_RENT

FROM OLAP\_DIM\_CAT\_CAR\_LOC D

INNER JOIN OLAP\_FACT\_CAT\_CAR\_AMOUNT F ON d.car\_id = F.Car\_Id

GROUP BY ROLLUP (d.state\_name,d.categorie, d.car\_brand)

ORDER BY d.state\_name,d.categorie ,d.car\_brand;

* + - View\_Analitic\_OLAP\_3...

**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.

CREATE OR REPLACE VIEW View\_StateCount AS

Select l."State" as state, cc.Label as label, count(r.Reservation\_ID) as total\_rent

from rentals\_view r

inner join location\_view l on l."Location\_ID" = r.Pick\_up\_location

inner join cars c on r.car\_id = c.car\_id

inner join car\_category\_view cc on c.Category\_ID = cc.Category\_ID

group by l."State", cc.Label

order by l."State"

Select state, max(total\_rent) from View\_StateCount group by state;