**Raport Analiză**

Bocioroaga George

505

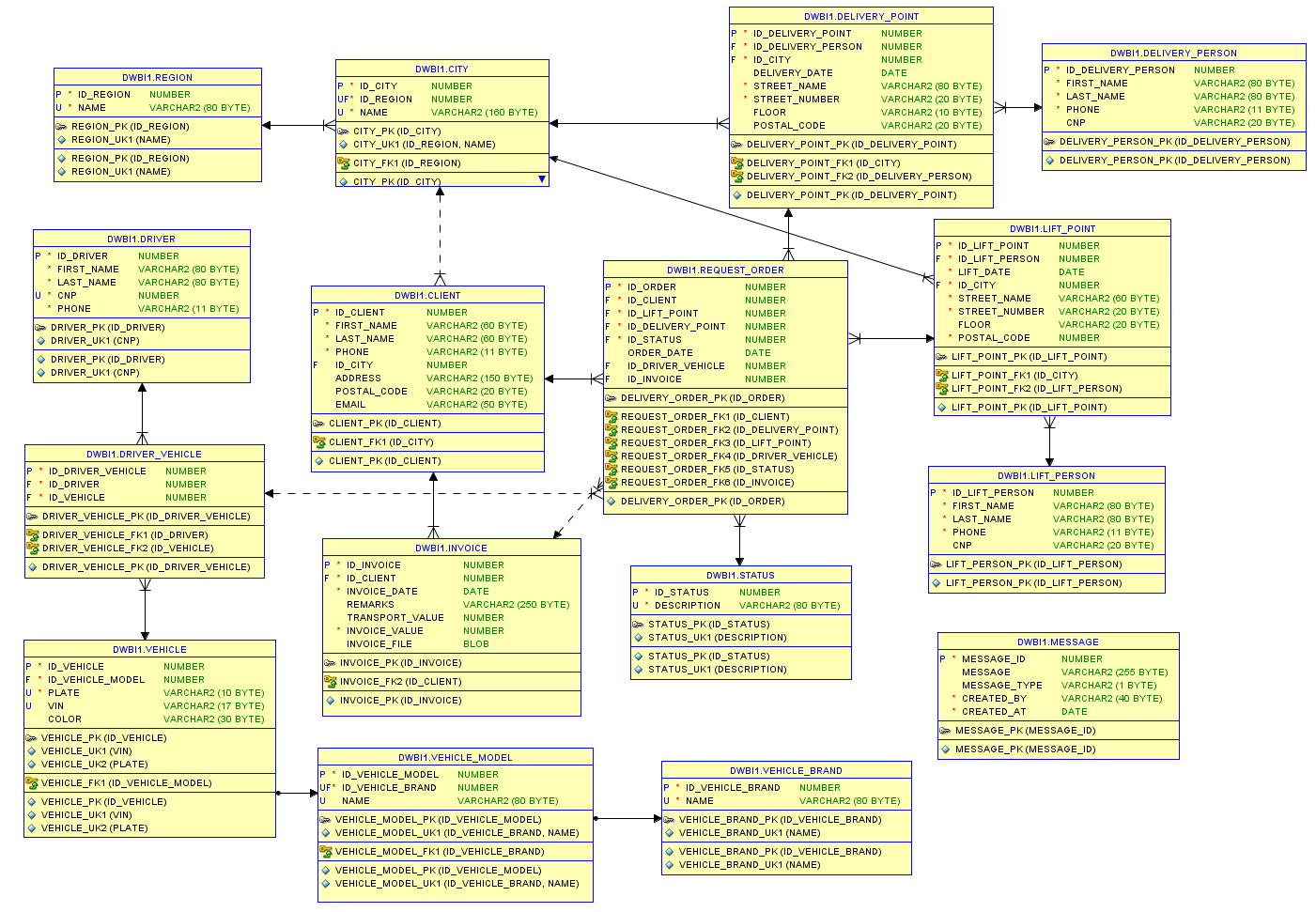
**1.Descrierea modelului ales şi a obiectivelor aplicaţiei**

Aplicaţia pe care ne dorim să o dezvoltăm este un mini sistem de gestiune pentru o firmă de transport intern de exemplu Urgent Cargus, Fan Courier. Reperele principale ale modelului bazei de date sunt: order-ul, clientul, resursele alocate, ridicarea coletului şi livrarea coletului. Tabelele bazei de date OLTP sunt următoarele:

|  |  |
| --- | --- |
| **Denumire tabel** | **Descriere** |
| REQUEST\_ORDER | Reţine principalele date ale unui order |
| INVOICE | Reţine datele despre costurile unui transport |
| CLIENT | Reţine datele de bază ale clientului |
| CITY | Oraşele |
| REGION | Judeţele |
| DRIVER\_VEHICLE | Pentru fiecare order va reţine şoferul şi maşina cu care se va efectua livrarea |
| DRIVER | Şoferii |
| VEHICLE | Maşinile firmei |
| VEHICLE\_MODEL | Modele de maşini |
| VEHICLE\_BRAND | Brand-urile de maşini |
| LIFT\_POINT | Punctul de ridicare al coletului |
| LIFT\_PERSON | Persoana de la care se ridică coletul |
| DELIVERY\_POINT | Punctul la care se livrează coletul |
| DELIVERY\_PERSON | Persoana căreia i se livrează coletul |
| STATUS | Statusul unui order |
| MESSAGE | Tabela de logging |

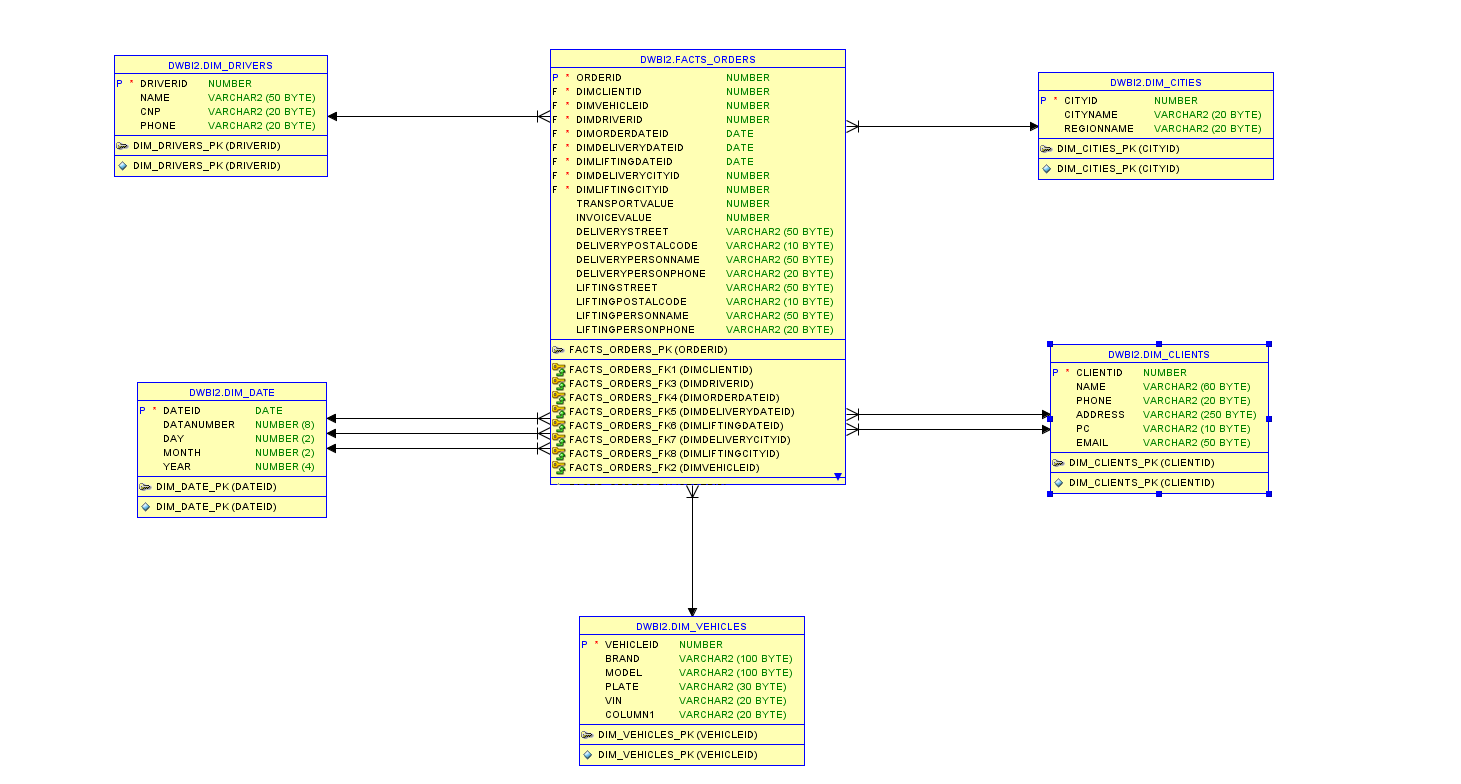
**2.Diagramele bazei de date OLTP**

*Diagrama conceptuală a bazei de date OLTP*



(adăugată şi în fişier pentru mai multă claritate)

**3.Diagrama stea/fulg a bazei de date depozit**



(adăugată şi în fişier pentru mai multă claritate)

**4.Descrierea câmpurilor necesare pentru fiecare tabel din baza de date depozit şi modul de populare al acestora cu informaţii din baza de date OLTP**

|  |  |
| --- | --- |
| **DimVehicles** | |
| VehicleId | Cheia primară a tabelei, va avea valori de la 1 |
| Brand | Valorea va fi populată din tabela VEHICLE\_BRAND |
| Model | Valore ava fi populată din tabela VEHICLE\_MODEL |
| Plate | Valore ava fi populată din tabela VEHICLE |
| Vin | Valore ava fi populată din tabela VEHICLE |
| Color | Valore ava fi populată din tabela VEHICLE |
| **DimDrivers** | |
| DriverId | Cheia primară a tabelei, va avea valori de la 1 |
| Name | Valorea va fi populată din tabela DRIVER |
| CNP | Valorea va fi populată din tabela DRIVER |
| Phone | Valorea va fi populată din tabela DRIVER |
| **DimDate** | |
| DateId | Cheia primară a tabelei, va fi de tip Date |
| DateNumber | Va fi de forma (YYYYMMDD) de exemplu (20200101, 20200210) |
| Day | Numărul zilei |
| Month | Numărul lunii |
| Quarter | Numărul trimestrului |
| Year | Numărul anului |
| **DimCities** | |
| CityId | Cheia primară a tabelei, va avea valori de la 1 |
| CityName | Numele oraşului din tabela Cities |
| RegionName | Numele judeţului din tabele Regions |
| **DimClients** | |
| ClientId | Cheia primară a tabelei, va avea valori de la 1 |
| Name | Numele clientului populat din tabela Clients |
| Phone | Telefonul clientului populat din tabela Clients |
| Adress | Adresa clientului care va îngloba date din (Adress, City, Region) din tabela Clients |
| PC | Codul poştal al clientului populat din table Clients |
| E-mail | E-mail-ul clientului populat din tabela Clients |
| **Facts\_Orders** | |
| FactId | Cheia primară a tabelei, va avea valori de la 1 |
| DimClientId | Cheie externă către DimClients, reprezintă clientul care a depus order-ul |
| DimVehicleId | Cheie externă către DimVehicles, reprezintă maşina cu care se va executa livrarea |
| DimDriverId | Cheie externă către DimDrivers, reprezintă şoferul care va executa livrarea |
| DimOrderDateId | Cheie externă către DimDate, reprezintă data la care s-a depus order-ul (va fi de tip Date) |
| DimDeliveryDateId | Cheie externă către DimDate, reprezintă data la care s-a finalizat livrarea (va fi de tip Date) |
| DimLiftingDateId | Cheie externă către DimDate, reprezintă data la care s-a ridicat coletul (va fi de tip Date) |
| DimDeliveryCityId | Cheie externă către DimCities, reprezintă oraşul unde se va livra coletul |
| DimLiftingCityId | Cheie externă către DimCities, reprezintă oraşul de unde se va ridica coletul |
| TransportValue | Valoarea transportului inserată de către client din tabela INVOICE |
| InvoiceValue | Valoarea transportului calculată în funcţie de numarul de kilometrii parcurşi şi valoarea transportului din tabela INVOICE |
| DeliveryStreet | Strada pe care se va livra coletul din tabela DELIVERY\_POINT |
| DeliveryPostalCode | Codul poştal la care se va livra coletul din tabela DELIVERY\_POINT |
| DeliveryPersonName | Persoana pe care se va livra coletul din tabela DELIVERY\_PERSON |
| DeliveryPersonPhone | Telefonul persoanei la care se va livra din tabela DELIVERY\_PERSON |
| LiftingStreet | Strada de unde se va ridica coletul din table LIFTING\_POINT |
| LiftingPostalCode | Codul postal de unde se va ridica coletul din tabela LIFTIN\_POINT |
| LiftingPersonName | Persoana de la care se va ridica coletul din tabela LIFTING\_PERSON |
| LiftingPersonPhone | Telefoanul persoanei de la care se va ridica coletul din tabela LIFTING\_PERSON |

Baza de date de warehouse va fi populată cu ajutorul unui job care va prelua datele din baza de date OLTP si le va insera in WH. Acest job va fi o procedură stocată care va rula la un interval de timp stabilit (în jur de 5 minute).

**5.Identificarea constrângerilor specifice depozitelor de date ce trebuie definite, justificând alegerea făcută**

O bună observaţie ar fi că tabelele DimCities şi DimDate nu vor fi populate cu aceeaşi frecvenţă ca DimClient, DimDrivers şi DimVehicles, deoarece datele lor vor fi statice şi nu vor apărea modificari prea des. O altă menţiune este că tabela de Facts\_Orders va conţine doar order-uri care au fost încheiate cu succes şi care au fost plătite.

**Constrângeri de domeniu**

**UNIQUE**

|  |  |  |
| --- | --- | --- |
| **Tabel** | **Coloană** | **Justificare** |
| **DimVehicles** | Plate | Numărul de înmatriculare trebuie să fie unic (ENABLE, VALIDATE). |
| VIN | Seria de săşiu trebuie să fie unică (ENABLE, VALIDATE). |
| **DimDrivers** | CNP | CNP-ul unui individ trebuie să fie unic (ENABLE, VALIDATE) |

**NOT NULL**

În realizarea bazei de warehouse am considerat că absolut toate coloanele trebuie să aibe constrângerea de not null.

**CHECK**

|  |  |  |
| --- | --- | --- |
| **Tabel** | **Coloană** | **Justificare** |
| **DimVehicles** | VIN | Seria de şasiu trebuie să aibe un număr fix de caractere. |
| **DimDrivers** | CNP | CNP-ul trebuie sa aibe 13 cifre şi să înceapă cu 1 sau 2 |
| **DimClients** | E-mail | Trebuie să conţină @ |
| **Facts\_Orders** | InvoiceValue,TransportValue | Trebuie să fie mai mari ca 0 |

**CONSTRÂNGERI DE CHEIE EXTERNĂ**

|  |  |  |
| --- | --- | --- |
| **Tabel** | **Coloană** | **Justificare** |
| **DimCities** | RegionId | Această constrângere se va crea cu opţiunile implicite de ENABLE, VALIDATE şi NORELY, deoarce inserările vor fi foarte rare sau chiar inexistente. Toate oraşele şi judeţele lor vor fi încărcate o singură dată. |
| **Facts\_Orders** | DimClientId | Această constrângere se va crea cu opţiunile de ENABLE NOVALIDATE, deoarece această tabelă nu va fi populată cu aceeasi frecvenţă ca cea de fapte. |
| DimDriverId | Va avea aceleaşi opţiuni ca cea de DimClientId, deoarece în acest domeniu angajaţii se schimbă foarte des. |
| DimOrderDateId | Constrângerile se vor crea cu opţiunile implicite de ENABLE, VALIDATE şi RELY, deoarece aceste id-uri vor exista cu siguranţa în tabele de DimDate. |
| DimDeliveryDateId |
| DimLiftingDateId |
| DimDeliveryCityId | Constrângerile vor avea opţiunile implicite deoarece toate oraşele din ţară vor fi încărcate din prima. |
| DimLiftingCityId |

**6.Identificarea indecşilor specifici depozitelor de date ce trebuie definiți asupra modelului (minim 2 dacă echipa este formată din 4 persoane); formularea unei cereri în limbaj natural care va determina utilizarea indecșilor specificați şi va fi implementată în următoarea etapă**

|  |  |  |
| --- | --- | --- |
| **Index** | **Cod** | **Motivaţie** |
| Index bitmap join pe tabelele Facts si DimClients | Create bitmap index index\_facts\_clients on Facts\_Orders (DimVehicleId, DimDriverId) from Facts\_Orders, DimClients where fo.DimClientId = c.ClientId | Am ales să indexăm coloanele DimVehicleId si DimDriverId pentru că acestea vor avea cea mai mică cardinalitate şi pentru că am considerat că cele două vor avea o pondere de utilizare foarte mare în query-urile ce vor fi procesate la nivel de aplicaţie. |

**Cereri SQL:**

Să se afişeze numărul comenzilor, data, numele clientului si adresa de email pentru comenzile care au fost transportate cu vehiculele ce au id-urile: 153, 112, 95, 99 ,134 si de catre şoferii cu id-urile: 5, 52, 26, 44, 4

**7. Identificarea obiectelor de tip dimensiune ce trebuie definite asupra modelului (minim 2 dacă echipa este formată din 4 persoane)**

|  |  |  |
| --- | --- | --- |
| **Dimensiune** | **Ierarhie** | **Motivaţie** |
| Vom crea un obiect de tip dimensiune pentru tabela DimDate. | Day->Week->Month->Quarter->Year | Am mers pe presupunerea că multe dintre interogările de pe baza de date de WH vor dori să afişeze evoluţia comenzilor în timp. |

*CREATE DIMENSION times\_dim*

*LEVEL day IS DimDate.Day*

*LEVEL week IS DimDate.Week*

*LEVEL month IS DimDate.Month*

*LEVEL quarter IS DimDate.Quarter*

*LEVEL year IS DimDate.Year*

*HIERARCHY cal\_rollup(*

*day CHILD OF*

*week CHILD OF*

*month CHILD OF*

*quarter CHILD OF*

*year );*

**8. Identificarea tabelelor care vor fi partiţionate şi a tipului de partiţionare (minim 2 dacă echipa este formată din 4 persoane; formularea unei cereri în limbaj natural care va determina utilizarea lor şi va fi implementată în următoarea etapă**

Tabela care va fi partiţionată este tabela de fapte - FACTS\_ORDERS .

*CREATE TABLE FACTS\_ORDERS\_ORD\_NUMBER*

*(ORDERID NUMBER,*

*DIMCLIENTID NUMBER,*

*DIMVEHICLEID NUMBER,*

*DIMDRIVERID NUMBER,*

*DIMORDERDATEID NUMBER)*

*PARTITION BY RANGE(DIMORDERDATEID)*

*(*

*PARTITION ORDERS\_2015*

*VALUES LESS THAN(2457389),*

*-- 2457389 = select TO\_CHAR(TO\_DATE('01/01/2016','DD/MM/YYYY'),'J') from dual*

*PARTITION ORDERS\_2016*

*VALUES LESS THAN(2457755),*

*-- 2457755 = select TO\_CHAR(TO\_DATE('01/01/2017','DD/MM/YYYY'),'J') from dual*

*PARTITION ORDERS\_2017*

*VALUES LESS THAN(2458120),*

*-- 2458120 = select TO\_CHAR(TO\_DATE('01/01/2018','DD/MM/YYYY'),'J') from dual*

*PARTITION ORDERS\_2018*

*VALUES LESS THAN(2458485),*

*-- 2458485 = select TO\_CHAR(TO\_DATE('01/01/2019','DD/MM/YYYY'),'J') from dual*

*PARTITION ORDERS\_2019*

*VALUES LESS THAN(2458850),*

*-- 2458850 = select TO\_CHAR(TO\_DATE('01/01/2020','DD/MM/YYYY'),'J') from dual*

*PARTITION ORDERS\_RESTVALUES LESS THAN (MAXVALUE));*

*insert into FACTS\_ORDERS\_ORD\_NUMBER*

*select orderid, dimclientid, dimvehicleid, dimdriverid,*

*to\_char(dimorderdateid, 'j')*

*from FACTS\_ORDERS;*

**Exemplu de cerere care utilizează o submulţime a partitiei:**

*SELECT \**

*FROM facts\_orders\_ord PARTITION (orders\_2019);*

Un alt tip de partiţionare pe care o putem face tabelei de fapte este partionarea te tip LISTă, având cheia de partiţionare coloana LiftingCity\_id. Astfel, vom putea crea tabela facts\_orders\_list

*CREATE TABLE facts\_orders\_LIST*

*(factid NUMBER,*

*client\_id NUMBER,*

*vehicle\_id NUMBER,*

*driver\_id NUMBER,*

*orderDate\_id DATE,*

*deliveryCityId NUMBER,*

*liftingCityId NUMBER*

*)*

*PARTITION BY LIST(liftingCityId)*

*(*

*PARTITION* *orders\_cities\_rang1*

*VALUES (1,2,3,4,5,6),*

*-- 1 = Bucuresti, 2 = Cluj, 3 = CONSTANTA, 4 = TIMISOARA, 5 = Iasi, ….*

*PARTITION orders\_cities\_rang2*

*VALUES (21,22,23,24,25),*

*-- 21 = PASCANI, 22 = Targoviste, 23 = Botosani, 24 = Medgidia, 25 = SIGHISOARA*

*PARTITION orders\_cities\_rang3*

*VALUES (51,52,53,54),*

*--51 = VIESUL DE SUS, 52 = OGNA MURES, 53 = GURA HUMORULUI, 54 = ORSOVA*

*PARTITION orders\_cities\_rang4*

*VALUES(DEFAULT)*

*);*

În cazul acestei partiţionări am utilizat submulţimi predefinite astfel incât înregistrarile care nu se încadrează în nicio submulţime definite să nu genereze erori.

Exemplu de cerere care utilizeaza doar o submultime a partitiei.

*SELECT \**

*FROM FACTS\_ORDERS\_LIST PARTITION (orders\_cities\_rang1);-- i.e. LIFTINGCITYID IN (1,2,3,4,5);*

**9. Formularea în limbaj natural a unei cereri SQL complexe care va fi optimizată în următoarea etapă, folosind tehnici specifice bazelor de date depozit. Precizarea tehnicilor de optimizare ce ar putea fi utilizate pentru această cerere particulară (avantaje/dezavantaje de utilizare pentru o anumită tehnică)**

**Să se afişeze top 3 cele mai utilizate vehicule precum şi oraşele în care au efectuat acestea livrări**.

*select*

*rownum as TOP\_NUMBER, plate, brandmodel, cities, veh\_count, vehicleid*

*from (*

*select*

*sel.veh\_count, sel.vehicleid, sel.brandmodel, sel.plate*

*, listagg(sel.cityname, ', ') within group (order by sel.cityname) cities*

*from (*

*with top\_veh as (*

*select \* from (*

*select count(orderid) veh\_count, dimvehicleid vehid*

*from facts\_orders*

*group by dimvehicleid*

*order by count(orderid) desc*

*) where rownum <=3 )*

*select*

*distinct veh\_count, v.vehicleid, v.brand || ' ' || v.model as brandModel, v.plate, c.cityname*

*from*

*facts\_orders f, dim\_vehicles v, dim\_cities c, top\_veh*

*where*

*f.dimdeliverycityid = c.cityid*

*and f.dimvehicleid = v.vehicleid*

*and top\_veh.vehid = f.dimvehicleid) sel*

*group by*

*sel.veh\_count, sel.vehicleid, sel.brandmodel, sel.plate*

*order by veh\_count desc, vehicleid*

**10. Formularea în limbaj natural a cel puţin 5 cereri cu grad de complexitate diferit, concretizate în rapoarte (grafice) ce vor fi create în următoarele etape**

1. Să se afişeze topul clientilor in functie de valoarea facturilor emise pentru acestia. Selectul va returna locul din top, codul clientului, numele acestuia şi suma valorii facturilor emise pentru acesta.

*select rownum as TOP#, a.\* from*

*( select dimclientid as client\_code, c.name as client\_name, sum(invoicevalue) as sum\_of\_invoices*

*from facts\_orders f, dim\_clients c*

*where f.dimclientid = c.clientid*

*group by dimclientid, name*

*order by sum(invoicevalue) desc, c.name asc) a;*

1. Să se afişeze top 5 oraşe în care au fost livrate cele mai multe colete, precum şi numărul de colete.

*select \* from*

*(*

*select*

*c.cityname, count(f.orderid) packages\_#*

*from*

*facts\_orders f, dim\_cities c*

*where*

*f.dimdeliverycityid = c.cityid*

*group by*

*c.cityname*

*order by*

*count(f.orderid) desc*

*)a*

*where rownum < 6;*

1. Să se afişeze şoferii care nu au livrat niciun pachet în anul 2015.

*select \**

*from*

*dim\_drivers d*

*where*

*d.driverid not in (*

*select f.dimdriverid*

*from*

*facts\_orders f*

*where*

*f.dimorderdateid between to\_date('01/01/2015','dd/mm/yyyy')*

*and to\_date('31/12/2015', 'dd/mm/yyyy')*

*)*

*order by d.name;*

***VS***

*select \**

*from dim\_drivers d*

*where*

*d.driverid not in (*

*select f.dimdriverid*

*from*

*facts\_orders\_ord\_number partition(orders\_2015) f*

*) order by d.name;*

1. Să se afişeze ziua cea mai profitabilă din sistem (doar transport value va fi considerat).

*select dimorderdateid as the\_day, transport\_value\_day most\_valuable\_day*

*from (*

*select*

*sum(f.transportvalue) transport\_value\_day, f.dimorderdateid*

*from*

*facts\_orders f*

*group by f.dimorderdateid*

*order by sum(f.transportvalue) desc*

*)*

*where rownum =1;*

1. Să se afişeze numărul mediu de comenzi pe fiecare quarter înregistrat în sistem.

*select*

*d.quarter quarter, round(avg(orderid),0) medie\_comenzi*

*from*

*facts\_orders f, dim\_date d*

*where*

*f.dimorderdateid = d.dateid*

*group by*

*d.quarter order by avg(orderid);*