5.

C1: Avem tabela de fapte FACT\_TRANZACTII având următoarele atribute: ID#, ID\_Tranzactie, ID\_Client, ID\_Cont, ID\_Comerciant, ID\_Canal\_Plata, ID\_Stare, ID\_Locatie, ID\_Data, Durata, Suma. Coloanele ID\_Client, ID\_Cont, ID\_Comerciant, ID\_Canal\_Plata, ID\_Stare, ID\_Locatie si ID\_Data identifică în mod unic o înregistrare din această tabelă. Un mod potrivit pentru un depozit de date de a defini aceasta constrângere de unicitate este prezentat in documentul „Scripturi”, punctul 5 (cod referință c1).

Justificare: Am ales aceasta abordare deoarece opțiunea DISABLE ne scutește de costurile unui index unic (care nu a fost creat), dar in același timp asigura unicitatea datelor. Având in vedere faptul ca tabela FACT\_TRANZACTII este actualizata printr-un proces ETL, perechea de constrângeri DISABLE VALIDATE nu incomodează: nu mai sunt permise operații LMD, constrângerea nu este activa, dar s-a realizat validarea datelor.

C2: Într-o schemă stea, constrângerile de cheie externă validează relațiile dintre tabela de fapte și tabelele dimensiune. Tabela de fapte FACT\_TRANZACTII conține atributele ID\_Client, ID\_Cont, ID\_Comerciant, ID\_Canal\_Plata, ID\_Stare si ID\_Locatie care referă atributele cu același nume din tabelele dimensiune DIM\_CLIENT, DIM\_DETALII\_PLATA, DIM\_COMERCIANT, DIM\_CANAL\_PLATA, DIM\_STARE, respectiv DIM\_LOCATIE. Pentru depozitele de date, o maniera potrivita de a defini constrângerile de cheie externa este exemplificata in documentul „Scripturi”, punctul 5 (cod referință c2-c7).

Justificare: Atunci când se încărca un depozit de date, de obicei nu este nevoie de validarea constrângerile, deoarece datele au fost încărcate în bulk dintr-o zonă de pregătire în care a fost efectuată toată validarea datelor. Astfel, am salvat timp prețios declarând constrângerile fără a le valida. Crearea unei constrângeri în NOVALIDATE este imediată și nu depinde de dimensiunea tabelului. Constrângerea garantează că nu se vor introduce rânduri in tabela de fapte fără un rând corespunzător în dimensiune. Cu toate acestea, deoarece Oracle nu a validat rezultatul în sine, putem seta atributul RELY. Pe lângă RELY NOVALIDATE, implicit, constrângerea este definită cu opțiunea ENABLE, adică constrângerea este activă și va determina verificarea integrității datelor în cazul operațiilor LDM realizate asupra tabelelor implicate.

C3: Tabela de fapte FACT\_TRANZACTII conține atributul ID\_Data care referă atributul cu același nume din tabela dimensiune DIM\_CALENDAR. Pentru depozitele de date, o alternativă de definire a constrângerii de cheie externă între cele două tabele se găsește in documentul „Scripturi”, punctul 5, sub codul de referința c8.

Justificare: Opțiunea RELY permite optimizatorului sa utilizeze constrângerea pentru a determina un plan optim, fără consum de resurse.

7.

Pentru a putea stabili care vor fi obiectele dimensiune pe care le vom crea, vom începe prin a menționa ce sunt acestea si la ce ne pot ajuta.

Dimensiunea este o structură care clasifică datele pentru a le permite utilizatorilor să răspundă la întrebările de afaceri. Dimensiunile utilizate în mod obișnuit sunt clienții, produsele și timpul. De exemplu, fiecare canal de vânzări al unui comerciant poate colecta și stoca date privind vânzările. Managementul lanțului de vânzare poate construi un depozit de date pentru a analiza vânzările produselor sale în toate magazinele de-a lungul timpului și pentru a răspunde la întrebări precum:

* Care este efectul promovării unui produs comparative cu unul care nu este promovat?
* Care sunt vânzările unui produs înainte și după o promoție?
* Cum afectează o promovare diferitele canale de distribuție?

În baza de date Oracle, informațiile dimensionale în sine sunt stocate într-un tabel de dimensiuni. În plus, obiectul dimensiune ajută la organizarea și gruparea informațiilor dimensionale în ierarhii. Aceasta reprezintă relații naturale 1:n între coloane sau grupuri de coloane (nivelurile unei ierarhii) care nu pot fi reprezentate cu condiții de constrângere. Urcarea unui nivel în ierarhie se numește acumularea datelor și coborârea la un nivel în ierarhie se numește detaliere a datelor.

Obiectele dimensiune nu trebuie definite neapărat. Cu toate acestea, crearea lor poate aduce beneficii semnificative, deoarece ajută la rescrierea interogărilor să efectueze tipuri mai complexe de rescriere.

In cazul acestui proiect au fost identificate 2 obiecte dimensiune: locație, timp. Ambele dimensiuni sunt normalizate(nu este stocata in mai mult de un table, sau altfel zis, nu are referințe la alte tabele).

Aceste obiecte vor avea următoarele nivele si ierarhie:

* Locație: locatie\_id -> strada -> oraș -> tara
* Timp: data\_id -> data -> anul

Pentru dimensiunea locație, clauza ATTRIBUTE ... DETERMINES se referă la a pune in legătura locatie\_id cu celelalte câmpuri, si anume: strada, oraș, tara si site. Aceasta este o determinare unidirecțională. Este doar garantat că pentru un anumit id, o sa fie găsita exact o valoare potrivită pentru strada, oraș, tara si site. Însă nu se poate determina un id pentru un anumit oraș spre exemplu.

Similar este si in cazul dimensiunii timp, un singur lucru diferă aici, si anume faptul ca putem crea mai multe relații prin atribute de aceasta data. Motivul pentru care putem face acest lucru este faptul ca atât data\_id cat si data sunt unice in tabelul dim\_calendar. Prin urmare, câmpurile: id\_data, data, ziua, luna, anul pot fi determinate in mod direct atât prin data\_id cat si prin data.

Nu in ultimul rând, informațiile unui obiect dimensiune sunt doar declarative și nu sunt impuse de baza de date. Dacă relațiile descrise de dimensiuni sunt incorecte, pot apărea rezultate incorecte. Prin urmare, ar trebui să fie verificate periodic relațiile specificate de CREATE DIMENSION utilizând procedura DBMS\_DIMENSION.VALIDATE\_DIMENSION.