



UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
DEPARTMAN ZA MATEMATIKU I
INFORMATIKU



SEMINARSKI RAD

**kao deo praktičnog projekta
iz predmeta Softversko inženjerstvo za sisteme baza podataka**

prof. dr Jovana Vidaković

Nikola Vetnić 438/19 IT

Novi Sad, januar 2023.

SADRŽAJ

1 Uvod	5
2 Specifikacija poslovnih procesa	5
3 Online Transactional Processing baza podataka	5
4 Online Analytical Processing baza podataka	8
5 Izveštaji	14
6 Zaključak	16

1 Uvod

Praktični projekat iz predmeta Softversko inženjerstvo za sisteme baza podataka bavi se analizom poslovanja mreže fiktivnih turističkih agencija koje privređuju na teritoriji Republike Srbije i Republike Srpske u okviru Bosne i Hercegovine. Transakciona baza podataka usredsređena je na praćenje prodatih vaučera za putovanja u različite gradove Srbije i Republike Srpske, kao i različitih izleta u neposrednoj blizini ciljne destinacije koji se nude putnicima na mestu odmora.

Budući da je reč o agregiranim podacima velikog broja turističkih agencija, već u relativno kratkom periodu za koji su dostupni podaci (reč je o trogodišnjem intervalu sa početkom od 1. januara 2019. godine) uneto je oko 10,000 prodatih vaučera i nešto više izleta, sve to za oko 1,000 različitih kupaca. Zamišljeno nadležno telo zainteresovano je za aktuelne i precizne podatke vezane za popularnost i subjektivnu ocenu pojedinačnih destinacija, a takođe i za kvantitativne podatke poput ostvarnog profita i sličnih.

Navedeni razlozi upravo su poslužili kao osnovna motivacija za kreiranje skladišta podataka (eng. *data warehouse*), odnosno OLAP baze koja će pružiti fleksibilniji način manipulacije sakupljenim podacima sa ciljem kvalitetnijeg poslovanja u budućnosti.

2 Specifikacija poslovnih procesa

Kako je već rečeno, nadležno telo koje je pokrenulo analizu zainteresovano je pre svega za (ne)zainteresovanost klijenata za određene destinacije, njihovu subjektivnu ocenu ponude kao i njen objektivni obim, ostvaren profit, i konačno korelaciju navedenih parametara, u smislu postojanja određenih kontradikcija koje bi mogle da smetaju poslovnom uspehu (poput, siromašne ponude za popularne destinacije, ili pak preobimne ponude za lokacije za koje turisti nisu zainteresovani).

Dimenzije podataka koje će se posmatrati uključuju hijerarhijske geografske podatke, odnosno državu, regiju i grad ponude, zatim lične i demografske podatke o klijentima poput adrese stanovanja i uzrasta, te zbirne (količina, datum prve i poslednje ostvarene prodaje) i individualne (vreme prodaje, zaduženi agent) podatke o samim prodatim aranžmanima i kategorije ocenjivanja.

Glavna pitanja na koja naručilac analize traži odgovore su sledeća: koliko je putovanja (vaučera) prodato po gradovima, koji grad je najpopularnija destinacija (kao putovanje, i kao izlet), kakve su ocene (rejtinzi) po gradovima (kao putovanja, i kao izleta), koliki je profit ostvaren na aranžmanima po regionima, i kakav je odnos broja aranžmana po gradovima/regionima (sa ciljem utvrđivanja situacija gde potencijalno za popularne destinacije nema dovoljno ponuda).

Jedinstvenim izvorom podataka pri kreiranju i popunjavanju skladišta podataka figurira OLTP baza nadležnog tela koje i naručuje analizu.

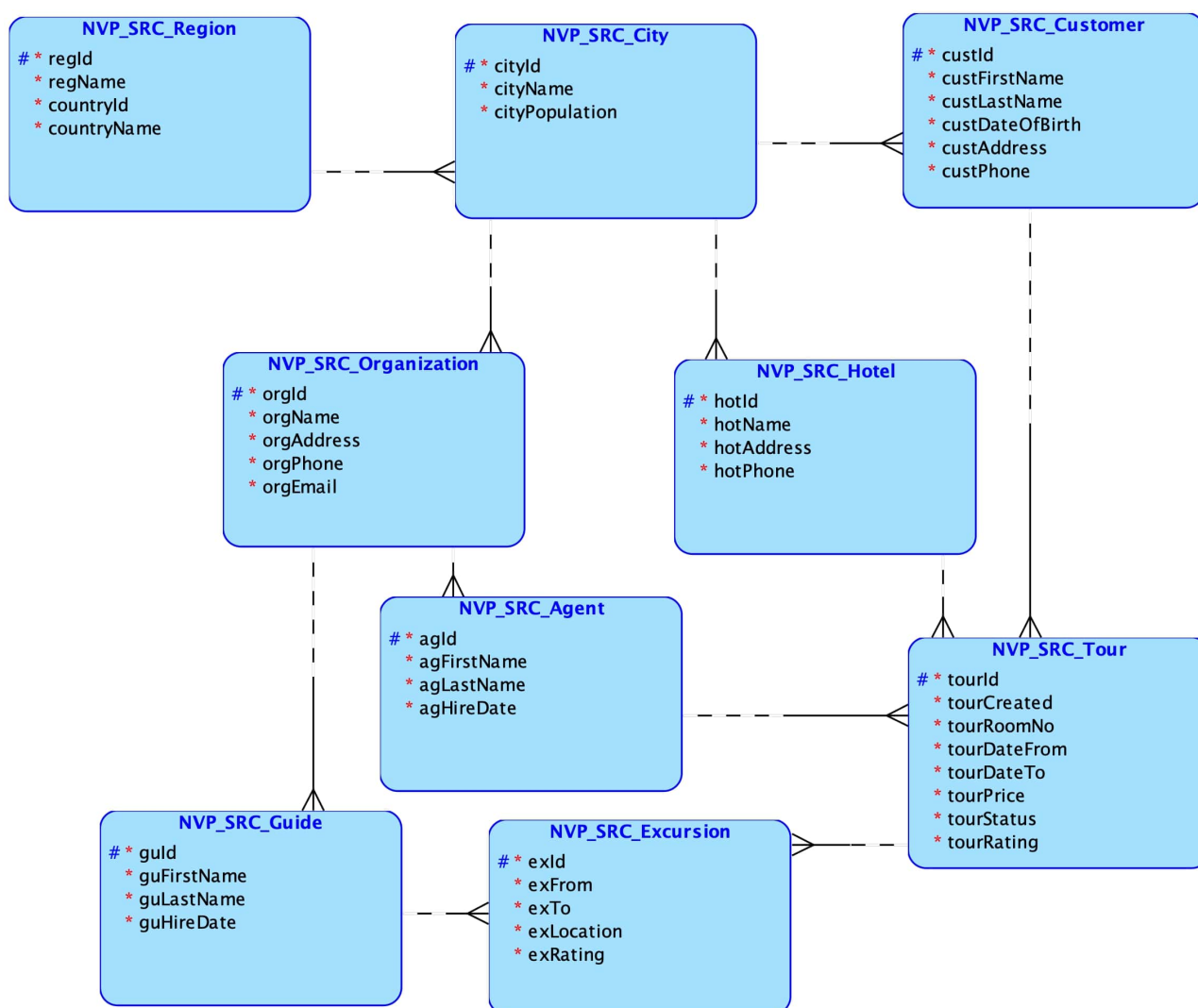
3 *Online Transactional Processing* baza podataka

Online Transactional Processing (OLTP) omogućava izvršavanje velikog broja transakcija u realnom vremenu od strane velikog broja ljudi. OLTP sistemi stoje iza mnogih svakodnevnih aktivnosti, od bankomata preko kupovine u prodavnicama do hotelskih rezervacija. OLTP takođe može da pokreće nefinansijske transakcije, uključujući promene lozinke i tekstualne poruke.

OLTP sistemi koriste relacionu bazu podataka koja omogućava obradu velikog broja relativno jednostavnih transakcija (umetanje, ažuriranje ili brisanje podataka), pristup za više korisnika istim podacima uz istovremeno osiguravanje njihovog integriteta, brzu obradu podataka (pri čemu je vreme odziva u milisekundama), indeksirane skupove podataka za brzo pretraživanje, pronalaženje i postavljanje upita, i drugo. Mnoge organizacije koriste OLTP sisteme za obezbeđivanje podataka za OLAP.

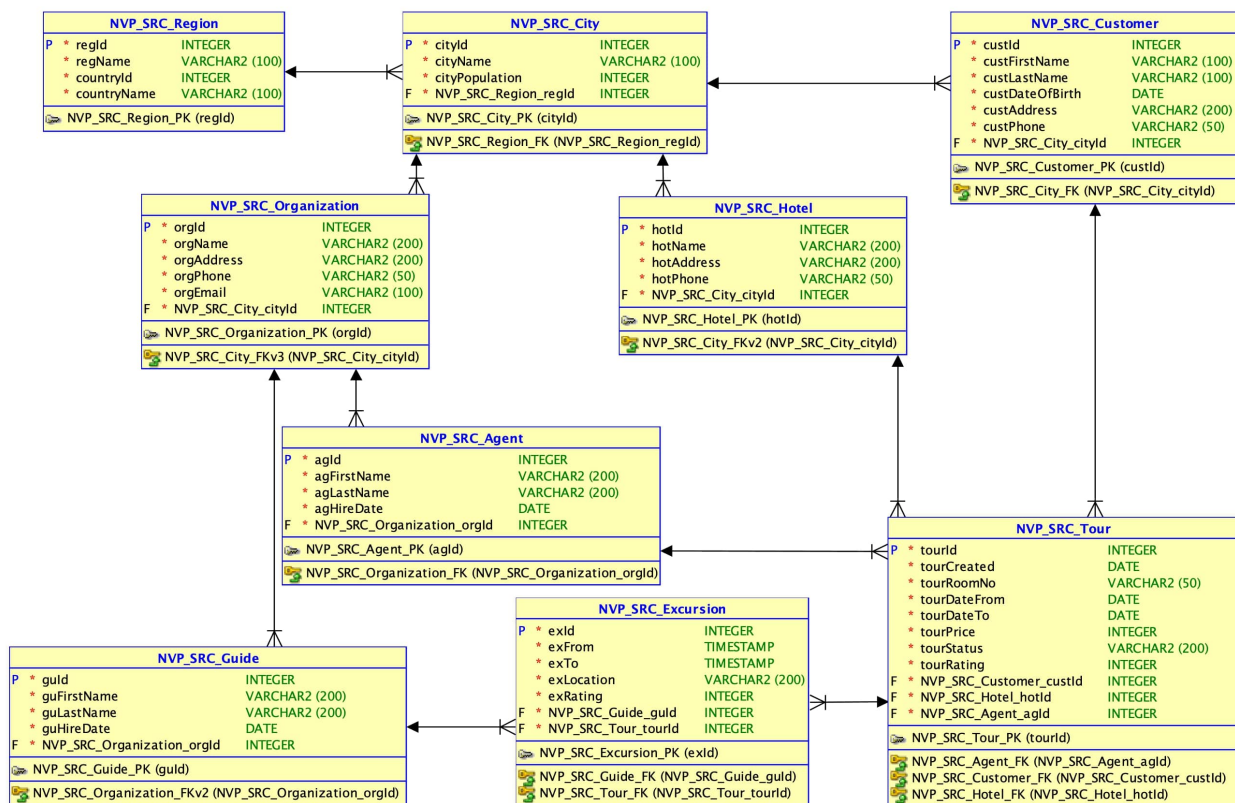
3.1 Logički i ER model OLTP baze podataka

Na slici 1 je prikazan logički model OLTP baze podataka koju nadležno telo održava sa ciljem praćenja poslovanja turističkih agencija za koje je zainteresovano.



Slika 1. Logički model OLTP baze podataka.

Na slici 2 prikazan je relacioni model iste baze podataka kreiran korišćenjem alata SQL Developer. U nastavku će ukratko biti prokomentarisana svaka od tabela.



Slika 2. ER model OLTP baze podataka.

Tabela NVP_SRC_Region sadrži podatke o regionu i državi i namerno je (poput primera sa nastave) denormalizovana kako bi se taj nedostatak otklonio pri popunjavanju skladišta podataka. NVP_SRC_City sadrži podatke o gradovima koje koriste sve tabele kojima su isti od značaja (svi entiteti koji u opisu imaju adresu). Obe tabele popunjene su realnim podacima o gradovima i okruzima u Srbiji, odnosno Republici Srpskoj.

Tabela NVP_SRC_Organization sadrži podatke o turističkim agencijama, dok se NVP_SRC_Hotel bavi hotelima i ostalim objektima za smeštanje turista na njihovim destinacijama. Tabele NVP_SRC_Guide, NVP_SRC_Agent i NVP_SRC_Customer sadrže podatke o osobama, gde prve dve pripadaju radnicima turističkih agencija, dok treća predstavlja klijente. Svih pet tabela popunjene su fiktivnim podacima generisanim preko servisa *Mockaroo*, sa određenim detaljima prilagođenim konkretnom slučaju.

Konačno, srž baze čine tabele NVP_SRC_Tour i NVP_SRC_Excursion koje čuvaju informacije o prodanim vaučerima i njihovom statusu, kao i izletima koji su vezani za određeni vaučer, a koje su klijenti ostvarili na mestu odmora.

3.2 Popunjavanje tabela NVP_SRC_Tour i NVP_SRC_Excursion

Kako je rečeno, tabele NVP_SRC_Tour i NVP_SRC_Excursion su od centralnog značaja za posmatrani projekat te su kao takve morale biti popunjene podacima koji su bar donekle smisleni. Sa tim ciljem napisan je jednostavan Java program koji generiše potrebne SQL naredbe, a čiji će delovi biti prokomentarisani u nastavku.

Osnovna ideja bila je kreirati smislene datume za putovanja, odnosno izlete. Jednostavnosti radi, svi datumi kreiranja vaučera su postavljeni u periodu od januara do marta tokom tri godine počevši od 2019. dok sva generisanja putovanja počinju nasumičnog dana između maja i septembra iste godine i traju 7, 14, 21 ili 28 dana.

Nakon kreiranja putovanja prelazi se na dodavanje izleta. Svako putovanje sadrži manje od četiri izleta, raspoređenih tako da se trajanje odmora podeli na $n + 1$ delova (pri čemu je n broj izleta), nakon čega se izleti raspoređuju tako da prvi počne m dana nakon početka odmora, drugi $2m$ nakon početka, i tako dalje, pri čemu je $m = \text{trajanje odmora} / (n + 1)$. Svaki izlet traje najviše dva dana.

Kreirane SQL naredbe se zatim upisuju u tekstualni fajl, nakon čega se kopiraju u SQL konzolu *SQL Developer* alata i izvršavaju. Primer generisanih naredbi je dat u listingu 1.

```
INSERT INTO NVP_SRC_Tour
    (tourId, tourCreated, tourRoomNo, tourDateFrom, tourDateTo, tourPrice,
    tourStatus, tourRating, NVP_SRC_Customer_custId, NVP_SRC_Hotel_hotId,
    NVP_SRC_Agent_agId)
VALUES
    (0, '12-Mar-2020', '41C', '09-Jul-2020', '06-Aug-2020', 41300,
    'canceled', 9, 255, 158, 187);
INSERT INTO NVP_SRC_Tour
    (tourId, tourCreated, tourRoomNo, tourDateFrom, tourDateTo, tourPrice,
    tourStatus, tourRating, NVP_SRC_Customer_custId, NVP_SRC_Hotel_hotId,
    NVP_SRC_Agent_agId)
VALUES
    (1, '14-Jan-2021', '114B', '16-Jun-2019', '07-Jul-2019', 52000,
    'canceled', 3, 262, 184, 77);

INSERT INTO NVP_SRC_Excursion
    (exId, exFrom, exTo, exLocation, exRating, NVP_SRC_Guide_guId,
    NVP_SRC_Tour_tourId)
VALUES
    (1, '17-Jul-2019', '17-Jul-2019', 'Excursion Site #4', 1, 3, 2);
```

Listing 1. Primer SQL naredbi za umetanje redova u tabele putovanja i izleta.

Čitav projekat se može pogledati na sledećem linku: https://github.com/NikolaVetnic/seds/tree/master/project/oltp_db/SedsFiller/src/app

4 Online Analytical Processing baza podataka

Online analytical processing (OLAP) je sistem za obavljanje višedimenzionalne analize velikom brzinom na velikim količinama podataka. Obično su ovi podaci iz skladišta podataka ili nekog drugog centralizovanog skladišta podataka. OLAP je idealan za rudarenje podataka (eng. *data mining*), potrebe poslovne inteligencije i složene analitičke proračune, kao i funkcije poslovnog izveštavanja kao što su finansijska analiza, budžetiranje i predviđanje prodaje.

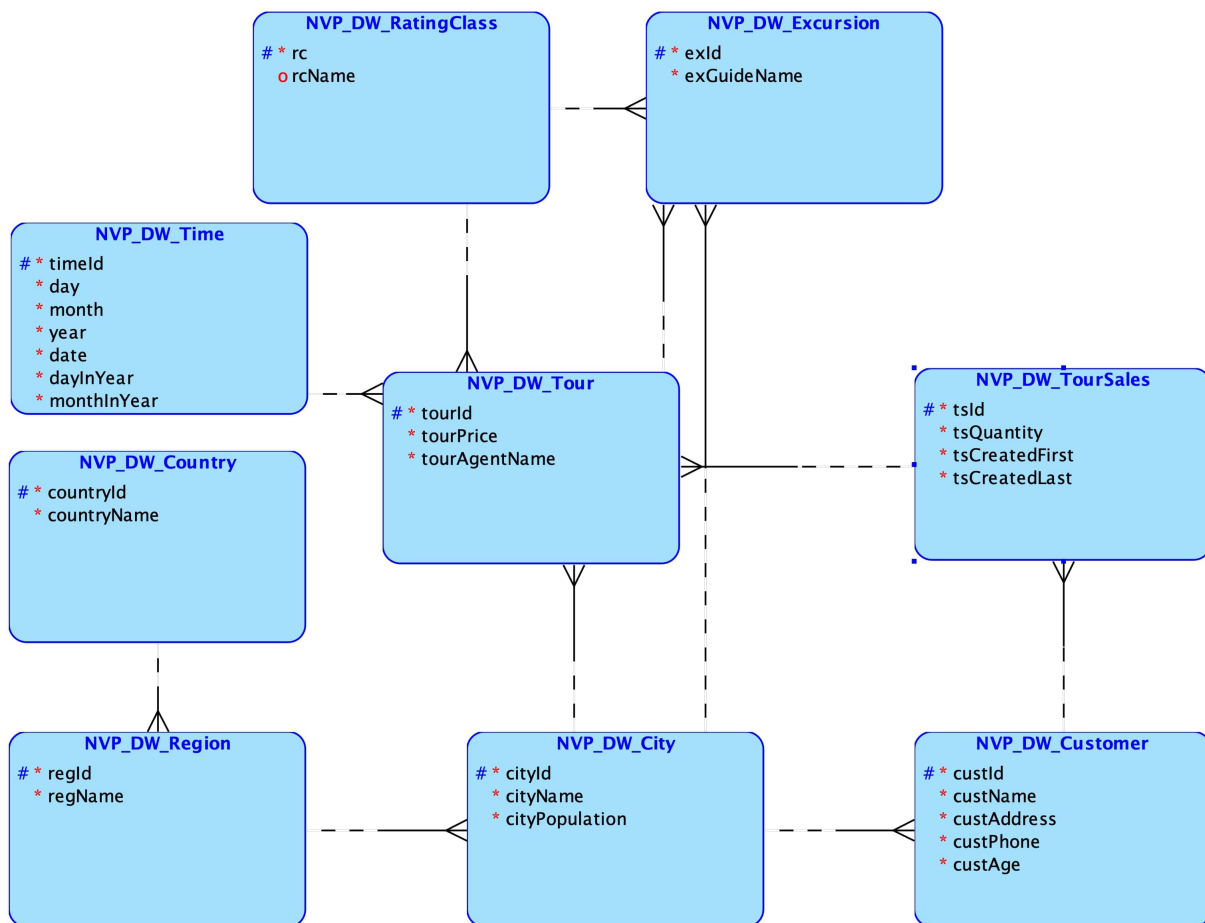
Srž većine OLAP baza podataka je OLAP kocka, koja omogućava brzo postavljanje upita, izveštavanje i analizu višedimenzionalnih podataka. Pod "dimenzijom podataka" se podrazumeva jednostavno jedan element određenog skupa podataka. Na primer, podaci o prodaji mogu imati nekoliko dimenzija vezanih za region, doba godine, modele proizvoda i još mnogo toga.

OLAP kocka proširuje *row-by-column* format tradicionalne šeme relacione baze i dodaje slojeve za druge dimenzije podataka. Na primer, dok gornji sloj kocke može da organizuje prodaju po regionima, takođe je moguće kretati se u dubinu, dosežući tako podatke o prodaji po državama ili regionima, gradovima ili pak određenim prodavnicama. Ovi istorijski, agregirani podaci za OLAP se obično čuvaju u zvezdastoj ili pahuljice šemi.

OLAP je tipično u suprotnosti sa OLTP-om (onlajn obrada transakcija), koji generalno karakterišu mnogo manje složeni upiti, u većem obimu i za obradu transakcija, a ne u svrhu poslovne inteligencije ili izveštavanja. Dok su OLAP sistemi uglavnom optimizovani za čitanje, OLTP mora da obrađuje sve vrste upita (čitanje, umetanje, ažuriranje i brisanje).

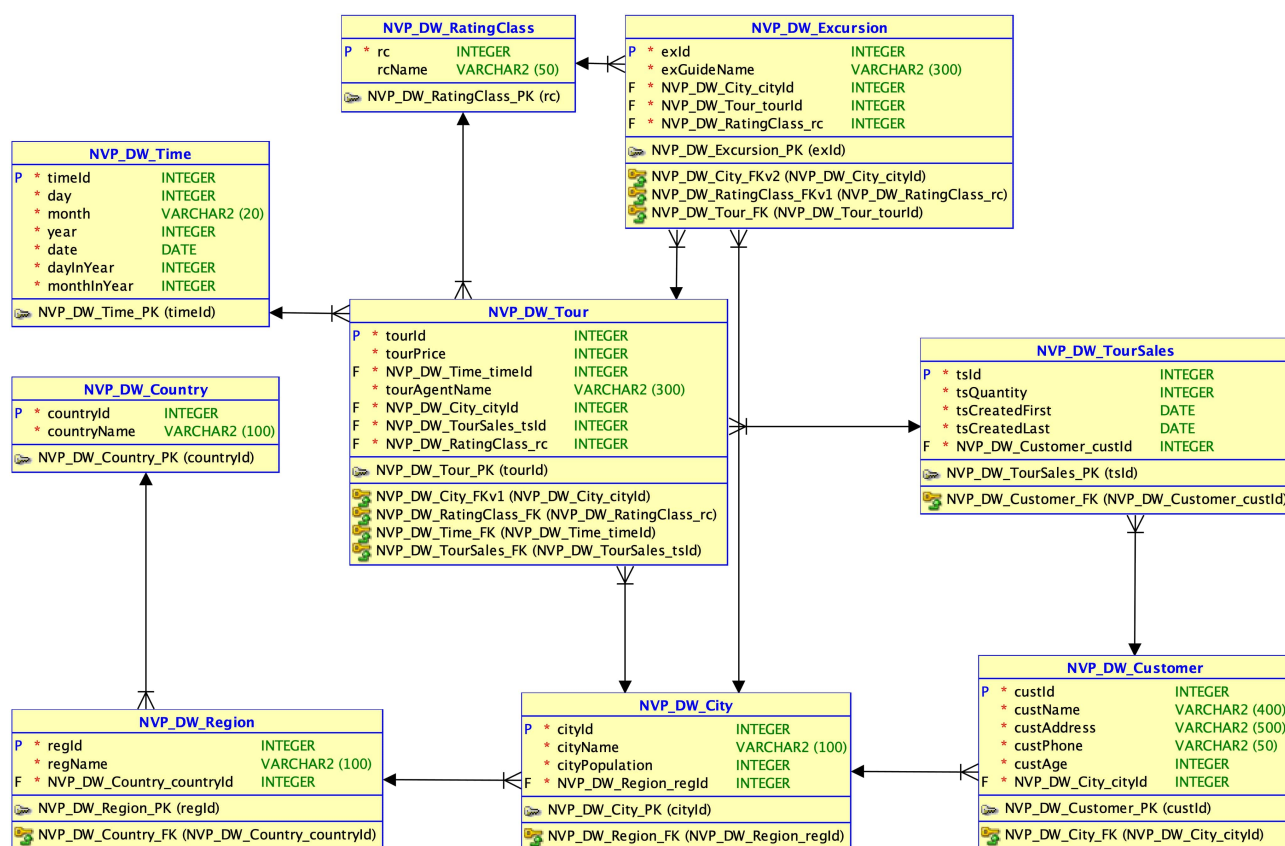
4.1 Logički i ER model OLAP baze podataka

Na slici 3 prikazan je logički model OLAP baze podataka koja se prevashodno koristi za dobijanje informacija zahtevanih od strane nadležnog tela.



Slika 3. Logički model OLAP baze podataka.

Na slici 4 prikazan je relacioni model OLAP podataka kreiran korišćenjem alata SQL Developer. Dalje sledi kratak komentar svake od tabela.



Slika 4. Relacioni model OLAP baze podataka.

OLAP baza podataka u ovom slučaju je zvezdastog oblika, pri čemu su glavne tabele činjenica NVP_DW_Tour i NVP_DW_Excursion okružene tabelama dimenzija poput NVP_DW_RatingClass, NVP_DW_City (koja je dalje hijerarhijski povezana sa NVP_DW_Region i NVP_DW_Country), i drugim.

Vremenska dimenzija je granulirana na jedinicu dana i predstavljena je unosima u tabeli NVP_DW_Time. Dimenzija ocene putovanja (rejtinga) je predstavljena je sa pet predefinisanih vrednosti datih u tabeli NVP_DW_RatingClass, a koje odgovaraju opsezima ocena počevši od nule sa inkrementom od 2.

Konačno, tabela NVP_DW_TourSales predstavlja agregirane kvantitativne i vremenske podatke o vaučerima kupljenim od strane jednog klijenta.

4.2 Transformacija podataka iz OLTP baze podataka

Transformacija podataka iz OLTP baze podataka izvršena je korišćenjem alata *Spoon* iz programskog paketa Pentaho Data Integration, i to u većini slučajeva pisanjem (složenijeg) SQL upita koji bi podatke i nazive kolona pripremio unapred, da bi se nakon toga podaci jednostavno prebacili u ciljanu OLAP tabelu putem *Table Output* čvora.

Tabela NVP_DW_Country popunjena je na osnovu podataka iz NVP_SRC_Region tabele, budući da je ona namerno kreirana kao denormalizovana kako bi se taj problem demonstrativno uklonio u ovoj fazi. Listing 2 prikazuje SQL upit kojim su podaci pripremljeni u alatu *Spoon*.

```
SELECT DISTINCT countryId, countryName
FROM NVP_SRC_Region
ORDER BY countryId
```

Listing 2. SQL upit za popunjavanje tabele NVP_DW_Country.

Tabele NVP_DW_Region i NVP_DW_City ne zaslužuju poseban komentar, budući da je reč o 1-na-1 preslikavanju analognih podataka iz tabela NVP_SRC_Region i NVP_SRC_City. Sam upit korišćen u *Spoon*-u je gotovo identičan onom prikazanom u listingu 2.

Tabela NVP_DW_Customer popunjena je na osnovu podataka iz NVP_SRC_Customer tabele, pri čemu je reč o 1-na-1 preslikavanju kod svih kolona osim custAge, gde se korišćenjem *Oracle*-ove TRUNC funkcije koju, srećom, prepoznaje i *Spoon*.

```
SELECT
    cust.custId,
    cust.custFirstName || ' ' || cust.custLastName AS "custName",
    cust.custAddress,
    cust.custPhone,
    TRUNC(months_between(sysdate, cust.custDateOfBirth) / 12) AS "custAge",
    cust.NVP_SRC_City_cityId AS "NVP_DW_City_cityId"
FROM NVP_SRC_Customer cust
ORDER BY cust.custId

INSERT INTO NVP_DW_Customer
    (custId, custName, custAddress, custPhone, custAge,
     NVP_DW_City_cityId)
SELECT
    cust.custId,
    cust.custFirstName || ' ' || cust.custLastName AS "custName",
    cust.custAddress,
    cust.custPhone,
    TRUNC(months_between(sysdate, cust.custDateOfBirth) / 12) AS "custAge",
    cust.NVP_SRC_City_cityId AS "NVP_DW_City_cityId"
FROM NVP_SRC_Customer cust;
```

Listing 3. SQL upit za popunjavanje tabele NVP_DW_Customer.

Budući da je u nekom momentu došlo do problema u radu sa alatom *Spoon*, za popunjavanje tabele klijenata korišćen je u nekim slučajevima i direktan SQL upit pokrenut u okviru *SQL Developer*-a. Obe varijante prikazane su u listingu 3.

```

SELECT DISTINCT
CASE
    WHEN tourRating > 8 THEN 'magnificent'
    WHEN tourRating > 6 THEN 'exceptional'
    WHEN tourRating > 4 THEN 'decent'
    WHEN tourRating > 2 THEN 'correct'
    ELSE 'appalling'
END rcName
FROM NVP_SRC_Tour
ORDER BY rcName

INSERT INTO NVP_DW_RatingClass (rc, rcName) VALUES (1, 'appalling');
INSERT INTO NVP_DW_RatingClass (rc, rcName) VALUES (2, 'correct');
INSERT INTO NVP_DW_RatingClass (rc, rcName) VALUES (3, 'decent');
INSERT INTO NVP_DW_RatingClass (rc, rcName) VALUES (4, 'exceptional');
INSERT INTO NVP_DW_RatingClass (rc, rcName) VALUES (5, 'magnificent');

```

Listing 4. SQL upit za popunjavanje tabele NVP_DW_RatingClass.

Tabela NVP_DW_RatingClass popunjena je korišćenjem proste CASE funkcije u okviru alata *Spoon*, odnosno direktnim unosom potrebnih vrednosti u slučaju kada se unos izvršavao iz *SQL Developer*-a. Kôd je prikazan na listingu 4.



Slika 5. Lanac transformacije korišćen za popunjavanje tabele NVP_DW_Time.

Popunjavanje tabele NVP_DW_Time predstavlja najsloženiju transformaciju realizovanu unutar alata *Spoon*. Prvi korak, odnosno čvor *Generate rows*, kreira 4000 datuma 31-12-2019 i prosleđuje ih čvoru *Add sequence* koji tako kreira sekvencu celobrojnih vrednosti počevši od 1.

```

SELECT
    COUNT(*) AS "tsQuantity",
    NVP_SRC_Customer_custId AS "NVP_DW_Customer_custId",
    MIN(tourCreated) AS "tsCreatedFirst",
    MAX(tourCreated) AS "tsCreatedLast"
FROM NVP_SRC_Tour
GROUP BY NVP_SRC_Customer_custId
ORDER BY NVP_SRC_Customer_custId

```

Listing 5. SQL upit za popunjavanje tabele NVP_DW_TourSales.

Čvor *Calculator* dodaje celobrojne vrednosti na datum 31-12-2019 u vidu dana i dobijene datume prosleđuje dalje. *Select values* grupiše vrednosti kao kolonu sa nazivom *tour_date*, dok *Calculator 2* zatim izvodi ostale vrednosti neophodne za popunjavanje reda tabele NVP_DW_Time. Konačno, rezultati se upisuju u tabelu od strane čvora *Table output*, kao što je prikazano na slici 5.

```

SELECT
    tourId AS "tourId",
    tourPrice AS "tourPrice",
    timeId AS "NVP_DW_Time_timeId",
    agFirstName || ' ' || agLastName AS "tourAgentName",
    NVP_SRC_City_cityId AS "NVP_DW_City_cityId",
    tsId AS "NVP_DW_TourSales_tsId",
    CASE
        WHEN tourRating > 8 THEN 5
        WHEN tourRating > 6 THEN 4
        WHEN tourRating > 4 THEN 3
        WHEN tourRating > 2 THEN 2
        ELSE 1
    END "NVP_DW_RatingClass_rc"
FROM NVP_SRC_Tour
LEFT JOIN NVP_SRC_Agent ON NVP_SRC_Agent_agId = agId
LEFT JOIN NVP_SRC_Hotel ON NVP_SRC_Hotel_hotId = hotId
LEFT JOIN NVP_DW_Time ON tourCreated = "date"
LEFT JOIN NVP_DW_TourSales ON NVP_SRC_Customer_custId = NVP_DW_Customer_custId

```

Listing 6. SQL upit za popunjavanje tabele NVP_DW_Tour.

Popunjavanje tabele NVP_DW_TourSales izvedeno je transformacijom podataka iz tabele NVP_SRC_Tour korišćenjem najjednostavnijeg mogućeg lanca u alatu *Spoon*, a koji se sastoji iz *Table input* i *Table output* čvorova. Korišćeni SQL upit je dat u listingu 5.

Na analogan način se popunjavaju i tabele NVP_DW_Tour i NVP_DW_Excursion, i to upitom prikazanim u listingu 6, odnosno 7.

```

SELECT
    exId AS "exId",
    guFirstName || ' ' || guLastName AS "exGuideName",
    NVP_SRC_City_cityId AS "NVP_DW_City_cityId",
    tourId AS "NVP_DW_Tour_tourId",
    CASE
        WHEN exRating > 8 THEN 5
        WHEN exRating > 6 THEN 4
        WHEN exRating > 4 THEN 3
        WHEN exRating > 2 THEN 2
        ELSE 1
    END "NVP_DW_RatingClass_rc"
FROM NVP_SRC_Excursion
LEFT JOIN NVP_SRC_Tour ON NVP_SRC_Tour_tourId = tourId
LEFT JOIN NVP_SRC_Hotel ON NVP_SRC_Hotel_hotId = hotId
LEFT JOIN NVP_SRC_Guide ON NVP_SRC_Guide_guId = guId

```

Listing 7. SQL upit za popunjavanje tabele NVP_DW_Excursion.

5 Izveštaji

Poslovni procesi za koje je zainteresovano nadležno telo zaduženo za turističke agencije čije se poslovanje prati u OLTP bazi skoncentrisani su na različite aspekte prodaje vaučera, ukupnu ostvarenu zaradu, te popularnosti različitih destinacija. Konkretno, naručilac posla se interesuje pre svega za odgovore na sledeća pitanja, koja su zauzvrat predmet odgovarajućih izveštaja:

1. koliko je putovanja (vaučera) prodato po gradovima,
2. koji grad je najpopularnija destinacija - kao putovanje, i kao izlet,
3. kakve su ocene (rejtinzi) po gradovima - kao putovanja, i kao izleta,
4. koliko se može zaraditi na aranžmanima po regionima,

```
SELECT
    SUM(ts.tsQuantity) AS "toursSold",
    (SELECT cityName FROM NVP_DW_City
     WHERE cityId = cust.NVP_DW_City_cityId
    ) AS "city",
    (SELECT regName FROM NVP_DW_City, NVP_DW_Region
     WHERE cityId = cust.NVP_DW_City_cityId AND NVP_DW_Region_regId = regId
    ) AS "region",
    (SELECT countryName
     FROM NVP_DW_City, NVP_DW_Region, NVP_DW_Country
     WHERE cityId = cust.NVP_DW_City_cityId AND
           NVP_DW_Region_regId = regId AND NVP_DW_Country_countryId = countryId
    ) AS "country"
FROM NVP_DW_TourSales ts
FULL OUTER JOIN NVP_DW_Customer cust ON ts.NVP_DW_Customer_custId = cust.custId
GROUP BY cust.NVP_DW_City_cityId
ORDER BY "toursSold" DESC;
```

Listing 8. SQL upit za izveštaj #1.

```
SELECT
    SUM(NVP_DW_City_cityId) AS "toursSold",
    (SELECT cityName
     FROM NVP_DW_City
     WHERE cityId = NVP_DW_City_cityId
    ) AS "city"
FROM NVP_DW_Tour GROUP BY NVP_DW_City_cityId ORDER BY "toursSold" DESC;

SELECT
    SUM(NVP_DW_City_cityId) AS "excursionsSold",
    (SELECT cityName
     FROM NVP_DW_City
     WHERE cityId = NVP_DW_City_cityId
    ) AS "city"
FROM NVP_DW_Excursion GROUP BY NVP_DW_City_cityId ORDER BY "excursionsSold" DESC;
```

Listing 9. SQL upit za izveštaj #2.

```

SELECT
    ROUND(TO_CHAR(AVG(tourRating)), 2) AS "avgRating",
    (SELECT cityName
     FROM NVP_SRC_City
     WHERE cityId = NVP_SRC_City_cityId
    ) AS "city",
    (SELECT regName
     FROM NVP_SRC_City, NVP_SRC_Region
     WHERE cityId = NVP_SRC_City_cityId AND NVP_SRC_Region_regId = regId
    ) AS "region",
    (SELECT countryName
     FROM NVP_SRC_City, NVP_DW_Region, NVP_DW_Country
     WHERE cityId = NVP_SRC_City_cityId AND NVP_SRC_Region_regId = regId AND
NVP_DW_Country_countryId = countryId
    ) AS "country"
FROM NVP_SRC_Tour, NVP_SRC_Hotel
WHERE NVP_SRC_Hotel_hotId = hotId
GROUP BY NVP_SRC_City_cityId ORDER BY "avgRating" DESC;

```

```

SELECT
    ROUND(TO_CHAR(AVG(exRating)), 2) AS "avgRating",
    (SELECT cityName
     FROM NVP_SRC_City
     WHERE cityId = NVP_SRC_City_cityId
    ) AS "city",
    (SELECT regName
     FROM NVP_SRC_City, NVP_SRC_Region
     WHERE cityId = NVP_SRC_City_cityId AND NVP_SRC_Region_regId = regId
    ) AS "region",
    (SELECT countryName
     FROM NVP_SRC_City, NVP_DW_Region, NVP_DW_Country
     WHERE cityId = NVP_SRC_City_cityId AND NVP_SRC_Region_regId = regId AND
NVP_DW_Country_countryId = countryId
    ) AS "country"
FROM NVP_SRC_Excursion, NVP_SRC_Tour, NVP_SRC_Hotel
WHERE NVP_SRC_Tour_tourId = tourId AND NVP_SRC_Hotel_hotId = hotId
GROUP BY NVP_SRC_City_cityId ORDER BY "avgRating" DESC;

```

Listing 10. SQL upit za izveštaj #3.

5. kakav je odnos broja aranžmana po gradovima/regijama i prosečne ocene?

Dok se prva dva upita oslanjaju isključivo na tabele OLAP baze, u preostala tri se određeni podaci dobivaju iz tabela izvorne baze. Reč je o geografskim podacima, odnosno o podacima vezanim za gradove, regije i države, koje se čak ni u OLTP bazi ne ažuriraju suviše često te uključivanje istih u upite ne obara performanse izveštaja u značajnom stepenu.

Takođe, izveštaji su realizovani kao "čisti" SQL upiti, u tom smislu da su izvršavani iz *SQL Developer*-a umesto *Pentaho Report Designer*-a. Razlog tome je nemogućnost pokretanja programa na računaru sa Apple M1 procesorom, te je stoga izostavljena grafička obrada tabelarnih prikaza, kao i različite napredne mogućnosti koje alat *Report Designer* nudi.

```

SELECT
    SUM(tourPrice) AS "totalProfit",
    (SELECT regName
     FROM NVP_DW_Region reg1
     WHERE reg0.regId = reg1.regId
    ) AS "region",
    (SELECT countryName
     FROM NVP_DW_Region reg1, NVP_DW_Country co1
     WHERE reg0.regId = reg1.regId AND reg1.NVP_DW_Country_countryId = co1.countryId
    ) AS "country"
FROM NVP_DW_Tour tour0, NVP_DW_City city0, NVP_DW_Region reg0
WHERE tour0.NVP_DW_City_cityId = city0.cityId
AND city0.NVP_DW_Region_regId = reg0.regId
GROUP BY regId ORDER BY "totalProfit";

```

Listing 11. SQL upit za izveštaj #4.

```

SELECT
    COUNT(tour0.tourId) AS "numTours",
    ROUND(TO_CHAR(AVG(tour0.tourRating)), 2) AS "avgRating",
    (SELECT regName
     FROM NVP_DW_Region reg1
     WHERE reg0.regId = reg1.regId
    ) AS "region"
FROM NVP_SRC_Tour tour0, NVP_SRC_Hotel hot0, NVP_SRC_City city0,
NVP_SRC_Region reg0, NVP_DW_Country co0
WHERE
    tour0.NVP_SRC_Hotel_hotId = hot0.hotId AND
    hot0.NVP_SRC_City_cityId = city0.cityId AND
    city0.NVP_SRC_Region_regId = reg0.regId
GROUP BY reg0.regId ORDER BY "avgRating" DESC;

```

Listing 12. SQL upit za izveštaj #5.

Rezultati svih upita dostupni su preko sledećeg linka: https://github.com/NikolaVetnic/seds/tree/master/project/olap_db/results

6 Zaključak

Gotovo da ne postoji oblast ljudske delatnosti koja danas ne izvlači korist iz sakupljanja, analitičke obrade i rudarenja podataka, u kom smislu OLAP baze pružaju očigledne prednosti. U tom smislu, ovaj praktični projekat bio je zanimljiva i korisna mogućnost da se u ograničenom obimu čitav proces prođe od početka do kraja.

Takođe, jasno je da je za kreiranje zaista korisne OLAP baze i upita koji pružaju najvrednije poslovne informacije neophodno iskustvo i poznavanje materije, te je kao takav ovaj smer izuzetno značajan za budući rad. (Samo)kritički osvrt na urađeno otkriva da su podaci OLAP bazi ovog projekta veoma slični onima u OLTP izvorniku, te je to svakako nešto na čemu bi ubuduće trebalo

poraditi. S druge strane, u okolnostima malog studentskog projekta veliko je pitanje u kom obimu su, i da li uopšte takvi nedostaci mogli biti otklonjeni.