

# OBLIKOVANJE SUSTAVA ZA POTPORU ODLUČIVANJU U SEKTORU TURISTIČKOG SMJEŠTAJA

---

Vranić, Josipa

Master's thesis / Diplomski rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Dubrovnik / Sveučilište u Dubrovniku**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:155:136050>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-02**



SVEUČILIŠTE U DUBROVNIKU  
UNIVERSITY OF DUBROVNIK

Repository / Repozitorij:

[Repository of the University of Dubrovnik](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U DUBROVNIKU  
FAKULTET ELEKTOTEHNIKE I PRIMIJENJENOG RAČUNARSTVA

DIPLOMSKI RAD

OBLIKOVANJE SUSTAVA ZA POTPORU ODLUČIVANJU  
U SEKTORU TURISTIČKOG SMJEŠTAJA

JOSIPA VRANIĆ

Dubrovnik, ožujak 2025.

SVEUČILIŠTE U DUBROVNIKU  
FAKULTET ELEKTOTEHNIKE I PRIMIJENJENOG RAČUNARSTVA

DIPLOMSKI RAD

OBLIKOVANJE SUSTAVA ZA POTPORU ODLUČIVANJU  
U SEKTORU TURISTIČKOG SMJEŠTAJA

Studij: Primijenjeno/poslovno računarstvo

Kolegij: Skladišta podataka i poslovna inteligencija

Mentor: prof.dr.sc. Mario Miličević

Student: Josipa Vranić

Dubrovnik, ožujak 2025.

## SAŽETAK

U ovom diplomskom radu prikazan je koncept skladišta podataka i njegova primjena u sektoru turističkog smještaja radi potpore odlučivanju. Rad analizira ključne razlike između transakcijskih baza podataka i skladišta podataka te objašnjava njihovu svrhu i primjenu u poslovnim procesima. Poseban naglasak stavljen je na dimenzijsko modeliranje, uključujući tehnike poput zvjezdaste i pahuljaste sheme. Posebna pažnja posvećena je procesu koji omogućava ekstrakciju, transformaciju i učitavanje podataka (Extract, Transform, Load - ETL) u skladište podataka. Naglašena je važnost pravilne implementacije skladišta podataka za optimizaciju poslovnih procesa i podršku u donošenju odluka. Rad istražuje i mogućnosti korištenja alata poslovne inteligencije (Business Intelligence - BI) za vizualizaciju podataka i generiranje poslovnih izvještaja, čime se poboljšava kvaliteta i pravovremenost informacija dostupnih unutar organizacije.

U praktičnom dijelu, rad prikazuje proces izgradnje relacijske baze podataka, koristeći podatke o smještajnim jedinicama i ostvarenim noćenjima u razdoblju od 2019. do 2023. godine. U bazi su obuhvaćeni ključni podaci o rezervacijama, gostima, smještajnim jedinicama i financijskim segmentima poslovanja. Nadalje, primjenom tehnika dimenzijskog modeliranja prikazano je stvaranje skladišta podataka i ETL procesi za prijenos podataka iz relacijske baze u skladište podataka. Stvoreno skladište podataka uz pomoć alata poslovne inteligencije omogućit će stvaranje vizualizacija i izvještaja, koji će poslužiti za bolje uvide o poslovanju te donošenje bitnih odluka unutar organizacije.

Cilj rada je prikazati kako kvalitetno dizajnirano skladište podataka može osigurati pravovremene, točne i korisne informacije za podršku poslovnim procesima u sektoru turističkog smještaja.

Ključne riječi: skladišta podataka, relacijske baze podataka, dimenzijsko modeliranje, ETL, poslovna inteligencija.

## ABSTRACT

This thesis presents the concept of a data warehouse and its application in the tourist accommodation sector to support decision-making. The paper analyzes key differences between transactional databases and data warehouses and explains their purpose and application in business processes. Special emphasis is placed on dimensional modeling, including star and snowflake schema techniques and to the ETL (Extract, Transform, Load) process, which enables the extraction, transformation and loading of data into the data warehouse. The importance of proper data warehouse implementation for optimizing business processes and supporting decision-making is mentioned. The paper also explores the possibilities of using business intelligence (BI) tools for data visualization and business report generation, which improves the quality and timeliness of information available within the organization.

The practical part shows the process of building a relational database, using data from accommodation units and reservations for the period between 2019. and 2023. The database includes key data on reservations, guests, accommodation units and financial aspects of the business. Furthermore, using dimensional modeling techniques, the creation of a data warehouse and ETL processes for data transfer from a relational database to a data warehouse are presented. The created data warehouse, supported by business intelligence tools, will enable the creation of visualizations and reports that will provide better insights into the business operations and aid in making important decisions within the organization.

The goal of this thesis is to demonstrate how a well-designed data warehouse can provide timely, accurate and useful information to support business processes in the tourist accommodation sector.

Key words: data warehouses, relational databases, dimensional modeling, ETL, BI.

# SADRŽAJ

1. UVOD.....	1
2. SUSTAVI ZA POTPORU ODLUČIVANJU .....	2
2.1.    Komponente sustava za potporu odlučivanju .....	2
2.2.    Tipovi sustava za potporu odlučivanju .....	2
2.3.    Prednosti i nedostaci korištenja sustava za potporu odlučivanju.....	3
3. PRIKUPLJANJE I PRIPREMA PODATAKA.....	5
3.1.    Rješavanje nedostajućih vrijednosti.....	6
4. TRANSAKCIJSKE BAZE PODATAKA – OLTP .....	12
4.1.    Ključna svojstva transakcijskih baza podataka.....	12
4.2.    ACID svojstva.....	13
4.3.    Tipovi transakcija u transakcijskim bazama podataka.....	13
4.4.    Izrada relacijske baze podataka i učitavanje podataka .....	14
5. SKLADIŠTA PODATAKA – OLAP .....	18
5.1.    Ciljevi izgradnje skladišta podataka (prema R. Kimballu):.....	19
5.2.    Komponente skladišta podataka .....	20
5.3.    Arhitekture skladišta podataka.....	21
5.4.    ETL procesi.....	22
5.5.    Metodologije izgradnje skladišta podataka.....	23
5.6.    Kvaliteta podataka u skladištima podataka.....	23
6. DIMENZIJSKO MODELIRANJE.....	25
6.1.    Zvezdasta shema ( <i>Star schema</i> ).....	25
6.2.    Činjenična tablica ( <i>Fact Table</i> ).....	25
6.3.    Dimenzijske tablice.....	26
6.4.    Pahuljasta shema ( <i>Snowflake schema</i> ).....	26
6.5.    Četiri koraka izgradnje dimenzijskog modela: .....	27
7. IZGRADNJA DIMENZIJSKOG MODELA.....	28
7.1.    Stvaranje skladišta podataka i tablica .....	29
7.1.1.    Dimenzijska tablica d_hotel.....	29
7.1.2.    Dimenzijska tablica d_guest .....	29
7.1.3.    Dimenzijska tablica d_room .....	29
7.1.4.    Dimenzijska tablica d_date .....	30
7.1.5.    Dimenzijska tablica d_reservationType .....	30
7.1.6.    Dimenzijska tablica d_bookingOption .....	31

7.1.7.	Dimenzijska tablica d_agency .....	31
7.1.8.	Činjenična tablica f_reservation .....	31
7.2.	ETL procesi.....	33
8.	POWER BI .....	43
8.1.	Učitavanje podataka – povezivanje sa skladištem podataka.....	43
8.2.	Izveštaj 1 – Pregled poslovanja smještajnih jedinica.....	44
8.3.	Izveštaj 2 – Pregled poslovanja Hotela Lero .....	45
8.4.	Izveštaj 3 – Pregled poslovanja Ville Orabelle .....	46
8.5.	Izveštaj 4 – Analiza soba hotela Lero .....	46
8.6.	Izveštaj 5 – Analiza soba Ville Orabelle .....	47
8.7.	Izveštaj 6 – Analiza gostiju hotela Lero.....	47
8.8.	Izveštaj 7 – Analiza gostiju Ville Orabelle .....	48
8.9.	Izveštaj 8 – Analiza prihoda po mjesecima .....	48
8.10.	Izveštaj 9 – Analiza prihoda po godinama i smještajnim jedinicama.....	49
8.11.	Izveštaj 10 – Usporedba broja rezervacija tijekom vikenda i radnih dana .....	49
8.12.	Izveštaj 11 – Broj rezervacija i suma prihoda u odnosu na blagdane .....	50
8.13.	Izveštaj 12 – Broj rezervacija tijekom sezona u godini .....	50
8.14.	Izveštaj 13 – Analiza opcija rezervacije i turističkih agencija .....	51
9.	ZAKLJUČAK.....	53
	POPIS SLIKA.....	54
	POPIS KODA.....	55
	LITERATURA.....	56

## 1. UVOD

U današnjem poslovnom okruženju, gdje organizacije svakodnevno prikupljaju i obrađuju velike količine podataka, učinkovito upravljanje informacijama ključno je za donošenje kvalitetnih poslovnih odluka. Sustavi za potporu odlučivanju razvili su se kao odgovor na složenost modernog poslovanja i potrebu za naprednim analizama podataka, a danas predstavljaju temelj poslovne inteligencije (*Business Intelligence* - BI) i ključni su alat za menadžere i analitičare. Jedan od ključnih elemenata poslovne inteligencije su skladišta podataka, koja omogućuju dugoročno pohranjivanje i analizu podataka iz različitih izvora. Razumijevanje razlika između transakcijskih baza podataka i skladišta podataka ključno je za donošenje odluka o tome kako pohranjivati, obrađivati i analizirati podatke. Iako su obje komponente bitne za pohranu podataka, služe različitim svrhama. Transakcijske baze podataka služe za upravljanje svakodnevnim operacijama, dok se skladišta podataka koriste za dugoročnu pohranu i analizu podataka. Razumijevanjem ključnih razlika između transakcijskih baza podataka i skladišta podataka mogu se donositi informirane odluke o tome koji alat će se koristiti na temelju specifičnih poslovnih potreba. Bez obzira radi se li se o transakcijama u stvarnom vremenu ili o dubinskoj analizi podataka, odabir prave arhitekture ključan je za postizanje željenih poslovnih ciljeva.

U ovom radu bit će prikazana važnost sustava za potporu odlučivanju, ključne razlike između transakcijskih baza podataka i skladišta podataka, njihove značajke i kada ih koristiti. Analizirat će se skladišta podataka, tehnike dimenzijskog modeliranja te različite arhitekture skladišta podataka. Prilikom definiranja skladišta podataka bit će prikazani stavovi dvaju velikana iz područja skladištenja podataka, W. Inmona i R. Kimballa. Fokus je na ključnim elementima skladišta podataka, a to su činjenične i dimenzijske tablice, zvjezdasta i pahuljasta shema, te koraci izgradnje dimenzijskog modela. Na konkretnom primjeru prikazat će se procesi za ekstrakciju, transformaciju i učitavanje (*Extract, Transform, Load* - ETL) podataka iz transakcijske baze u skladište podataka. Bitna je i kvaliteta podataka u skladištima, koja korisnicima osigurava pravovremene, točne i konzistentne informacije. Za kraj, uz pomoć alata poslovne inteligencije prikazat će se kreiranje poslovnih izvještaja i vizualizacija koji će se dalje koristiti za donošenje informiranih poslovnih odluka.



## 2. SUSTAVI ZA POTPORU ODLUČIVANJU

Sustavi za potporu odlučivanju (*Decision Support Systems - DSS*) predstavljaju interaktivne računalne sustave koji pomažu u procesima donošenja odluka u organizacijama. Korištenjem odgovarajućih alata i resursa pomažu pojedincima i grupama donositi informirane odluke, koje se zasnivaju na analiziranim podacima i generiranim uvidima. Za razliku od operacijskih programa koji služe za bilježenje poslovnih transakcija, sustavi za potporu odlučivanju predstavljaju informacijske programe koji korisnicima pružaju informacije temeljene na različitim izvorima podataka. Prilagodljivi su i namijenjeni za zadovoljavanje specifičnih potreba organizacije koja ga koristi.

Razvoj sustava za potporu odlučivanju započeo je 1960-ih godina, kada su organizacije počele koristiti računalne sustave za analizu podataka i podršku donošenju odluka. U početku su to bili jednostavni modeli koji su se temeljili na analizi podataka iz poslovnih transakcija. Prvi formalni DSS sustavi razvijeni su 1970-ih godina, a koristili su baze podataka i matematičke modele za optimizaciju poslovnih odluka. Tijekom 1980-ih i 1990-ih dolazi do napretka u računalnoj obradi podataka, što omogućuje razvoj složenijih sustava za potporu odlučivanju temeljenih na modelima i podacima. Tijekom 2000-ih godina dolaskom interneta DSS sustavi postali su sve dostupniji, koristeći poslovnu inteligenciju (*Business Intelligence - BI*), strojno učenje (*Machine Learning - ML*) i umjetnu inteligenciju (*Artificial Intelligence - AI*) za analizu velikih količina podataka. Danas su DSS sustavi ključni alati u poslovanju jer omogućuju donošenje bržih i preciznijih odluka.

### 2.1. Komponente sustava za potporu odlučivanju

Tipično se sustavi za potporu odlučivanju sastoje od tri dijela: baze znanja, softvera i korisničkog sučelja.

Baza znanja sastavni je dio baze podataka sustava za podršku odlučivanju, koja sadrži informacije iz unutarnjih i vanjskih izvora. To je biblioteka informacija koja se odnosi na određene teme i dio je DSS-a koji pohranjuje informacije koje koristi sustav za rasuđivanje kako bi odredio tijek djelovanja [1].

Softverski dio sustava predstavlja svu logiku i funkcionalnosti sustava za potporu odlučivanju, a uključuje modele, algoritme i metode koji analiziraju podatke i pretvaraju ih u vrijedne informacije i prijedloge za donošenje odluka. Modeli mogu biti jednostavni poput statističkih modela, do složenijih optimizacijskih modela, pa čak i alata za simulaciju.

Korisničko sučelje omogućuje jednostavnu i intuitivnu navigaciju sustavom. Primarni cilj korisničkog sučelja DSS-a je olakšati korisniku manipuliranje podacima koji su na njemu pohranjeni. Tvrtnke mogu koristiti sučelje za procjenu učinkovitosti DSS transakcija za krajnje korisnike. DSS sučelja uključuju jednostavne prozore, složena sučelja vođena izbornicima i sučelja naredbenog retka [1].

### 2.2. Tipovi sustava za potporu odlučivanju

Na temelju primarnih izvora informacija, sustavi za potporu odlučivanju dijele se u sljedećih pet kategorija: DSS vođen podacima, modelom vođen DSS, DSS vođen komunikacijom, DSS vođen znanjem i DSS vođen dokumentima.

DSS vođen podacima (*Data-driven DSS*) donosi odluke na temelju podataka iz internih ili eksternih baza podataka. Tipično se koriste tehnike rudarenje podataka kako bi se uočili trendovi i obrasci, što dalje omogućava predviđanje budućih događaja [1].

Modelom vođen DSS (*Model-driven DSS*) koristi matematičke i analitičke modele za donošenje odluka (npr. alate za financijsko predviđanje). Pomaže u financijskom planiranju, rasporedu resursa i optimizaciji poslovnih procesa.

DSS vođen komunikacijom (*Communication-driven DSS*) fokusira se na olakšavanje suradnje i donošenje odluka u grupama, a omogućuje timsku suradnju pomoću alata za komunikaciju (npr. videokonferencije, grupni softver), što za cilj ima povećanje učinkovitosti i suradnje u donošenju odluka.

DSS vođen znanjem (*Knowledge-driven DSS*) predstavlja sustav u kojemu se podaci koji pokreću sustav nalaze u bazi znanja koja se kontinuirano ažurira i održava sustav za upravljanje znanjem. DSS vođen znanjem korisnicima pruža informacije koje su u skladu s poslovnim procesima i znanjem tvrtke [1].

DSS vođen dokumentima (*Document-driven DSS*) koristi dokumente za dohvaćanje podataka, a korisnicima omogućuje pretraživanje informacija pomoću ključnih riječi u unutarnjim i vanjskim bazama podataka, uključujući i internet. Ovi sustavi probiru strukturirane i nestrukturirane podatke u dokumentima, a obično se nalaze online i u elektroničkim datotekama.

### 2.3. Prednosti i nedostaci korištenja sustava za potporu odlučivanju

Sustavi za potporu odlučivanju omogućuju donošenje odluka temeljenih na analitičkim podacima, smanjujući subjektivne procjene i povećavajući točnost odluka. Automatizacija analize podataka u DSS-u poboljšava operativnu učinkovitost i smanjuje vrijeme potrebno za donošenje odluka. Korištenjem sustava za potporu odlučivanju uzima se u obzir više izvora podataka, što omogućuje donošenje točnijih i relevantnijih odluka. Mogu se koristiti u različitim industrijama, kao što su zdravstvo, financije, marketing, turizam itd. Korištenjem ovih sustava mogu se rješavati i složeni problemi koje bi bilo teško analizirati ručno. DSS alati često uključuju i funkcije za suradnju i komunikaciju među zaposlenicima, što poboljšava suradnju unutar organizacije.

Iako sustavi za potporu odlučivanju nude brojne prednosti, također imaju i značajne nedostatke. Implementacija i održavanje sustava zahtijeva značajna financijska ulaganja, posebno za prilagođena rješenja, što ih za manje organizacije čini i manje dostupnim. DSS sustavi moraju osigurati zaštitu osjetljivih podataka, što zahtijeva napredne sigurnosne mjere. Dizajn i implementacija sustava za potporu odlučivanju zahtijevaju naprednu analitiku i integraciju podataka. Pretjerana ovisnost o sustavu može smanjiti subjektivnost i kreativnost prilikom donošenja odluka. Također, može doći i do otpora zaposlenika koji se možda ne žele voditi preporukama sustava, što može usporiti prihvaćanje novih tehnologija.

Sustavi za potporu odlučivanju igraju ključnu ulogu u modernim organizacijama unaprjeđujući donošenje odluka kroz uvide temeljene na podacima, analitičke modele i alate jednostavne za korištenje. Primjena ovih sustava obuhvaća različite industrije, od poslovanja do zdravstva, što ih čini nezamjenjivima u današnjem svijetu usredotočenom na podatke. Sve ovo za cilj ima pružiti donositeljima odluka sveobuhvatne informacije koje se mogu koristiti za donošenje

bržih i točnijih odluka. Međutim, njihova implementacija zahtijeva značajna ulaganja, a izazovi poput sigurnosti i prilagodbe zaposlenika mogu utjecati na uspješnost korištenja sustava. Osim visokih početnih troškova često su potrebne i dodatne edukacije korisnika, kako bi zaposlenici mogli iskoristiti puni potencijal sustava. Također, potrebno je osigurati pravilnu integraciju s postojećom infrastrukturom, kako bi podaci mogli nesmetano prolaziti između različitih poslovnih aplikacija. Ako sustav ne raspolaže kvalitetnim, točnim i ažurnim podacima, rezultati mogu dovesti do pogrešnih poslovnih odluka. Unatoč svim izazovima, prednosti sustava za potporu odlučivanju nadmašuju početne prepreke, čime organizacijama omogućuju da bolje razumiju tržišne trendove i unaprijede poslovne procese. Uz pravilan pristup implementaciji, edukaciji zaposlenika i održavanju sustava, organizacije mogu postići znatna poboljšanja u procesu donošenja odluka.

### 3. PRIKUPLJANJE I PRIPREMA PODATAKA

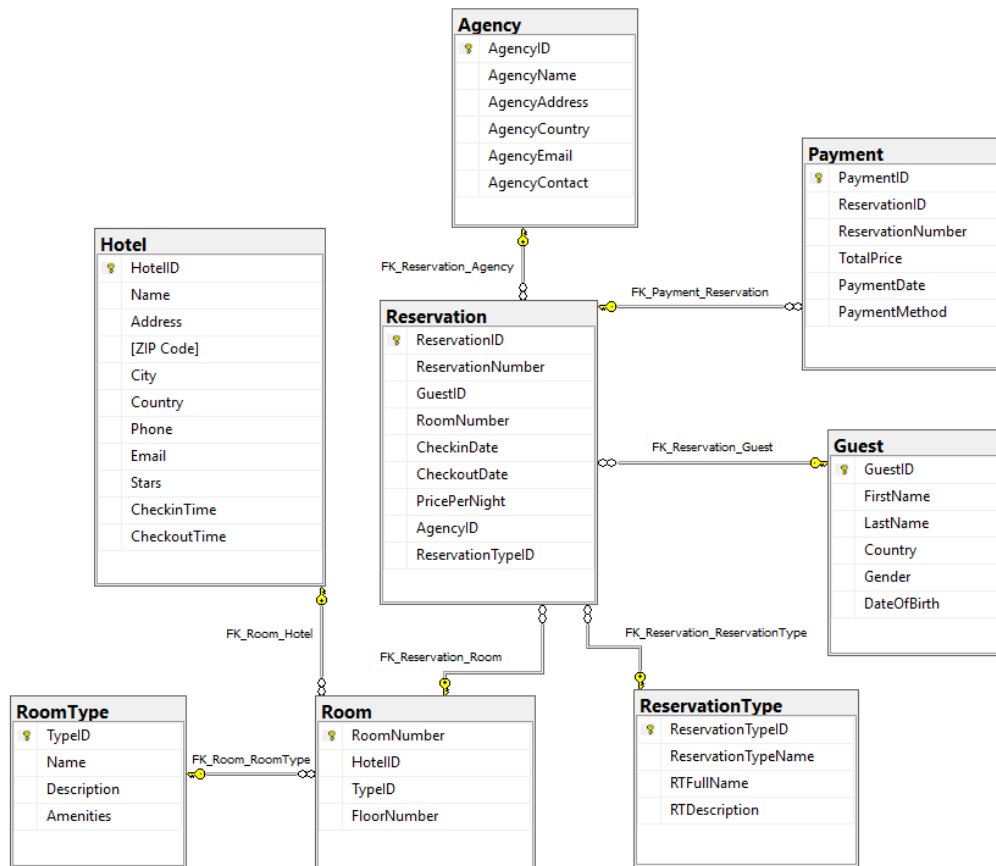
U praktičnom dijelu rada bit će korišteni podaci hotelskog rezervacijskog sustava dviju smještajnih jedinica na području grada Dubrovnika za razdoblje od 2019. godine do 2023. godine. Podaci dobiveni na korištenje nalaze se u Excel datotekama, a u nastavku će biti prikazana priprema i čišćenje podataka, koji će se u idućem koraku učitati u relacijsku bazu podataka. Na slici 1 prikazan je primjer podataka dobivenih na korištenje.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	101	JELENKO, ALOJZ	JELENKO	ALOJZ		SAJKO TURIZEM d.o.o.	30/12/2018	01/01/2019	HBG	34.40	EUR	F	2018/1018E	SLOVENIJA
2	101	JELENKO, VALERIJA	JELENKO	VALERIJA		SAJKO TURIZEM d.o.o.	30/12/2018	01/01/2019	HBG	34.40	EUR	F	2018/1018E	SLOVENIJA
3	101	JELENKO, PETRA	JELENKO	PETRA		SAJKO TURIZEM d.o.o.	30/12/2018	01/01/2019	HBG	34.40	EUR	F	2018/1018E	SLOVENIJA
4	101	SALINAS, IZAAK	SALINAS	IZAAK		HAPPY TOURSDruzina K	21/03/2019	23/03/2019	BBG	30.00	EUR	F	2019/100	SJEDINJENE AMERIČKE DRŽAVE
5	101	MANSOUR, ANDREW	MANSOUR	ANDREW		HAPPY TOURSDruzina K	21/03/2019	23/03/2019	BBG	30.00	EUR	F	2019/100	SJEDINJENE AMERIČKE DRŽAVE
6	101	DE LA ROSA, MATTHEW	DE LA ROSA	MATTHEW		HAPPY TOURSDruzina K	21/03/2019	23/03/2019	BBG	30.00	EUR	F	2019/100	SJEDINJENE AMERIČKE DRŽAVE
7	101	CHENG, SHENG CHIH	CHENG	SHENG CHIH		SUNSHINE INTERNATIO	24/03/2019	25/03/2019	BBG	28.00	EUR	G	2019/280	KINA
8	101	OU, MEI HSUAN	OU	MEI HSUAN		SUNSHINE INTERNATIO	24/03/2019	25/03/2019	BBG	28.00	EUR	G	2019/280	KINA
9	101	KANERVA, LEENA ANNELI	KANERVA	LEENA ANNELI		GRAND HOTEL PARK	26/03/2019	27/03/2019	HBB	24.00	EUR	F	2019/2729	FINSKA
10	101	AKREN, TUJUA INGRID	AKREN	TUJUA INGRID		GRAND HOTEL PARK	26/03/2019	27/03/2019	HBB	24.00	EUR	F	2019/2729	FINSKA
11	101	RAJENDRAN, KAMALINI	RAJENDRAN	KAMALINI		MEETING POINT CROAT	02/04/2019	09/04/2019	HB	40.00	EUR	F	2019/228	VELIKA BRITANIJA
12	101	RAJENDRAN, KANAGARATNAM	RAJENDRAN	KANAGARATNAM		MEETING POINT CROAT	02/04/2019	09/04/2019	HB	40.00	EUR	F	2019/228	VELIKA BRITANIJA
13	101	RAJENDRAN, YALINI	RAJENDRAN	YALINI		MEETING POINT CROAT	02/04/2019	09/04/2019	HB	32.00	EUR	F	2019/228	VELIKA BRITANIJA
14	101	JURIĆ, MARTA	JURIĆ	MARTA		EŠKO d.o.o. za trgovinu, t	09/04/2019	12/04/2019	HB	40.67	EUR	G	2019/108	HRVATSKA
15	101	HORVAT, LANA	HORVAT	LANA		EŠKO d.o.o. za trgovinu, t	09/04/2019	12/04/2019	HB	40.67	EUR	G	2019/108	HRVATSKA

Slika 1. Prikaz izvornih podataka

Podaci prikazuju broj sobe, imena gostiju, agenciju preko koje je odrađena rezervacija, datume dolaska i odlaska, vrstu rezerviranog smještaja, cijenu noćenja, broj rezervacije i državu iz koje gost dolazi. Budući da iz prve slike prikaza podataka nije odmah jasno značenje samih podataka, u nastavku će biti prikazan proces pripreme i čišćenja podataka i izgled podataka nakon. Za pregled podataka i pripremu istih, koristit će se Pandas biblioteka. To je open-source Python biblioteka koja se koristi za manipulaciju i analizu podataka.

Na osnovu dostupnih podataka stvorit će se relacijska baza podataka prikazana na slici 2. Baza podataka sastojat će se od sljedećih tablica: *Hotel*, *Room*, *RoomType*, *Reservation*, *ReservationType*, *Guest*, *Agency* i *Payment*. Tablica *Hotel* koristit će se za pohranu podataka o smještajnim jedinicama, a prikazat će ID hotela, naziv, adresu, poštanski broj, grad, državu, kontakt podatke, broja zvjezdica, vrijeme prijave u hotel i vrijeme odjave iz hotela. Tablica *RoomType* pohranit će sve tipove soba na osnovu naziva tipa sobe, opisa i sadržaja. Tablica *Room* omogućit će pohranu podataka o svim sobama, korištenjem broja sobe, kata na kojem se soba nalazi, te stranim ključevima prema tablicama *Hotel* i *RoomType*. Tablica *Guest* pohranit će podatke o gostima, uključujući ime, prezime, godinu rođenja, spol i državu iz koje gost dolazi. Tablica *ReservationType* koristit će se za prikaz svih tipova rezervacija i njihovih opisa npr. noćenje s doručkom, polupansion, puni pansion, itd. Tablica *Agency* prikazuje agencije putem kojih su napravljene rezervacije, a sadrži naziv, adresu, državu i kontakt podatke. Tablica *Reservation* pohranit će sve podatke o rezervacijama korištenjem ID-a rezervacije, broja rezervacije, datuma prijave i odjave, cijene noćenja, te stranih ključeva prema tablicama *Room*, *ReservationType*, *Guest*, *Agency* i *Payment*. Tablica *Payment* pohranjuje podatke o ukupnoj cijeni, datumu plaćanja i načinu plaćanja rezervacije. Ovaj relacijski model omogućuje fleksibilnost i normalizaciju podataka, čime se osigurava konzistentnost i smanjuje redundantnost informacija.



Slika 2. ER dijagram relacijske baze podataka

Budući da u podacima o cijenama noćenja postoje vrijednosti u valutama kuna i eura, u prvom koraku je prikazan proces pretvorbe svih vrijednosti sa oznakom HRK u EUR. Kod 1 prikazuje skriptu korištenu za navedeni proces.

```

import pandas as pd

df = pd.read_excel('C:\\Users\\Josipa\\Desktop\\LERO2023L-Copy.xlsx')

hrk_rows = df[df.iloc[:, 10] == 'HRK'].index

df.iloc[hrk_rows, 9] = df.iloc[hrk_rows, 9] / 7.53450

df.to_excel('C:\\Users\\Josipa\\Desktop\\LERO2023L-Copy.xlsx', index = False)
print('Podaci uspješno ažurirani.')

```

Kod 1. Promjena valute HRK u EUR

### 3.1. Rješavanje nedostajućih vrijednosti

Nedostajuće vrijednosti u podacima često mogu uzrokovati netočnosti, iskrivljene analize i donošenje loših odluka na temelju netočnih podataka. Zbog kvalitete podataka i točnosti analiza u nastavku će biti prikazani primjeri rješavanja nedostajućih vrijednosti.

Prvi korak je rješavanje nedostajućih vrijednosti u stupcu koji označava broj rezervacije. Ova nedostajuća vrijednost u podacima je prikazana oznakom '/', a slučajevi su da broj rezervacije nije uopće upisan ili kod jedne rezervacije sa više osoba, za pojedine retke broj nije upisan.

Rješavanje ovih nedostajućih vrijednosti važno je za daljnje praćenje brojeva rezervacija i povezivanje gostiju pod istim brojem rezervacije. Kod 2 prikazuje skriptu u kojoj se provjerava postoji li veza između prethodnog retka i retka koji ima nedostajuću vrijednost. Ako se podaci o broju sobe i vremenu boravka podudaraju, zaključit će se da je to ista rezervacija i bit će joj pridijeljena ista vrijednost koju ima prethodni redak. Ako redovi nisu povezani, onda se dodjeljuje novi broj rezervacije.

```
import pandas as pd

file_path = 'C:\\Users\\Josipa\\Desktop\\Book2.xlsx'
df = pd.read_excel(file_path)

df['OriginalOrder'] = df.index
if 'OriginalOrder' not in df.columns:
    raise ValueError("OriginalOrder stupac nije kreiran!")

df = df.sort_values(by=['RoomNumber', 'CheckinDate', 'CheckoutDate'])

reservation_number_counter = 16000
previous_reservation_number = None
previous_room = None
previous_checkin = None
previous_checkout = None

for index, row in df.iterrows():
    if row['ReservationID'] == '/': # Provjera nedostajućih vrijednosti broja rezervacija
        # Provjera je li red povezan sa prethodnim retkom (ista soba i isti datumi boravka)
        if (row['RoomNumber'] == previous_room and
            row['CheckinDate'] == previous_checkin and
            row['CheckoutDate'] == previous_checkout):
            # Dodijeliti isti broj rezervacije kao i prethodnom retku
            df.at[index, 'ReservationID'] = previous_reservation_number
        else:
            # Dodijeliti novi broj rezervacije u formatu '2023/xxxxx'
            new_reservation_number = f'2023/{reservation_number_counter}'
            df.at[index, 'ReservationID'] = new_reservation_number
            # Ažurirati brojač i postaviti prethodne vrijednosti za sljedeću iteraciju
            previous_reservation_number = new_reservation_number
            reservation_number_counter += 1
        else:
            # Ako broj rezervacije već postoji, pohraniti ga kao prethodni
            previous_reservation_number = row['ReservationID']

            # Ažurirati vrijednosti prethodnog retka za usporedbu
            previous_room = row['RoomNumber']
            previous_checkin = row['CheckinDate']
            previous_checkout = row['CheckoutDate']

# Provjeriti da stupac 'OriginalOrder' još postoji prije sortiranja
if 'OriginalOrder' not in df.columns:
    raise ValueError("OriginalOrder stupac nedostaje!")

# Sortirati podatke natrag u izvorni poredak
df = df.sort_values(by='OriginalOrder')

# Brisanje privremenog 'OriginalOrder' stupca
df = df.drop(columns=['OriginalOrder'])

# Spremiti ažurirane podatke natrag u Excel
output_file = 'C:\\Users\\Josipa\\Desktop\\Book2-UPDATED.xlsx'
df.to_excel(output_file, index=False)

print(f"File saved to {output_file}")
```

File saved to C:\Users\Josipa\Desktop\Book2-UPDATED.xlsx

Kod 2. Rješavanje nedostajućih vrijednosti broja rezervacije

Zbog zaštite osobnih podataka, neke podatke poput godine rođenja i spola gostiju nije bilo moguće dobiti na korištenje. Zbog toga su podaci o spolu generirani korištenjem `gender-guesser` biblioteke prikazane na sljedećoj slici (Kod 3). Korištena imena gostiju poslužit će za generiranje navedenih vrijednosti, a tijekom prikaza analiza podataka izvan organizacije imena će se sakriti, te na taj način spriječiti da osobni podaci o gostima budu javno dostupni.

```
import pandas as pd
import gender_guesser.detector as gender

def get_first_name(first_name):
    # Ako postoje dupla imena razdvaja ih i uzima prvo ime
    first = first_name.split()[0].capitalize()
    return first

def get_gender(first_name):
    # Dodjeljivanje spola na osnovu imena korištenjem gender-guesser Library.
    d = gender.Detector()
    return d.get_gender(first_name)

def update_gender_in_dataset(file_path, sheet_name):
    # Čitanje podataka iz Excel datoteke
    df = pd.read_excel(file_path, sheet_name=sheet_name)

    # Provjera postoji li 'FirstName' stupac
    if 'FirstName' not in df.columns:
        raise ValueError("Ne postoji stupac naziva 'FirstName'.")

    # Stvaranje riječnika za pohranu spola za svako ime
    gender_dict = {}

    print(f"Dohvat podataka o spolu za {len(df)} imena...")

    # Dohvati spol za svako ime
    for first_name in df['FirstName']:
        # Izdvoji ime
        processed_first_name = get_first_name(first_name)
        # Pogađanje spola na osnovu imena
        gender_dict[first_name] = get_gender(processed_first_name)
        print(f"Obrađeno: {first_name} -> {gender_dict[first_name]}")

    # Mapiranje podataka o spolu natrag u DataFrame
    df['Gender'] = df['FirstName'].map(gender_dict)

    # Datoteka u koju će se spremiti podaci
    output_file = 'C:\\Users\\Josipa\\Desktop\\rezervacije-hotel-gender-guesser3.xlsx'

    # Spremanje podataka u novi Excel dokument
    df.to_excel(output_file, index=False)

    print(f"Podaci spremljeni u: {output_file}")

    return df, output_file

file_path = 'C:\\Users\\Josipa\\Desktop\\rezervacije-hotel.xlsx'
sheet_name = '2023'

# Poziv funkcije
df, output_file = update_gender_in_dataset(file_path, sheet_name)
```

Kod 3. Gender-guesser detector

Gender-guesser je biblioteka koja se koristi za predviđanje spola osobe na osnovu imena [2]. Instalira se naredbom `'pip install gender-guesser'`. Rezultat može biti jedno od sljedećeg: *unknown* (ime nije pronađeno), *andy* (*androgynous*), *male*, *female*, *mostly\_male* i *mostly\_female*. Razlika između *'andy'* i *'unknown'* je da je utvrđeno da prvo ima veću vjerojatnost da bude muško ime nego žensko, dok drugo označava da ime nije pronađeno u bazi. Nakon pokretanja skripte prolazi se kroz podatke i korištenjem filter opcije u Excelu neodređeni podaci se ručno mijenjaju da na kraju prikazuju samo dvije opcije, M (*male*) ili F (*female*).

Za generiranje godine rođenja koristit će se skripta koja na osnovu sezonskih kretanja gostiju predviđa kojoj dobnoj skupini pripada osoba i na osnovu toga generira nasumični datum rođenja (Kod 4). Definirane su četiri dobne skupine i u ovisnosti o vremenu boravka dodijelit će im se datumi rođenja. Ako je vrijeme boravka u smještajnoj jedinici tijekom siječnja, veljače, studenog i prosinca, dodijelit će se datum rođenja u dobnoj skupini 50 – 85 godina. Za boravak tijekom ožujka, travnja, rujna i listopada, gost će pripasti dobnoj skupini 35 – 50 godina, a tijekom svibnja, lipnja, srpnja i kolovoza dodijelit će se dobna skupina 18 – 35 godina.

```
import pandas as pd
import random
from datetime import datetime, timedelta

# Funkcija koja generira random datum rođenja za zadani age range
def generate_random_dob(min_age, max_age):
    today = datetime.today()
    start_date = today - timedelta(days=max_age*365)
    end_date = today - timedelta(days=min_age*365)
    random_date = start_date + (end_date - start_date) * random.random()
    return random_date.strftime('%Y-%m-%d')

# Funkcija koja dodjeljuje datum rođenja ovisno o mjesecu u kojem se boravi
def assign_dob_based_on_month(month):
    if month in [1, 2, 11, 12]:
        # Elderly (50-85 godina)
        return generate_random_dob(51, 85)
    elif month in [3, 4, 9, 10]:
        # Adults (35-50 godina)
        return generate_random_dob(35, 50)
    elif month in [5, 6, 7, 8]:
        # Young-adults (18-35 godina)
        return generate_random_dob(18, 35)

#Funkcija koja dodjeljuje datum rođenja djeci, prolazi kroz podatke i ako je više
#od 2 ljudi pod istom rezervacijom i sa istim prezimenom zapisuje prvih 2 kao
#roditelje i ostale kao djecu
def assign_family_dob(guest_data):
    for res_num, group in guest_data.groupby('ReservationID'):
        # Filtriranje gostiju sa istim prezimenom
        last_name_groups = group.groupby('LastName')
        for _, family in last_name_groups:
            if len(family) > 2:
                # Prvih 2 roditelji, ostali djeca
                for i, row in family.iterrows():
                    if i > family.index[1]:
                        guest_data.at[i, 'DateOfBirth'] = generate_random_dob(0, 18)
    return guest_data
```

Kod 4. Generiranje datuma rođenja – 1



Kod 4 također sadrži funkciju koja računa datum rođenja za djecu na temelju više ljudi sa istim prezimenom unutar istog broja rezervacije, gdje se prvih dvoje zapisuju kao roditelje, a ostale kao djecu.

Kod 5 prikazuje funkciju koja računa mjesec boravka u smještajnoj jedinici, a računa se na osnovu razlike između datuma prijave i datuma odjave. Zatim, čita podatke iz Excel datoteke i stvara stupac za pohranu datuma rođenja, poziva prethodno definirane funkcije i sprema generirane podatke u novu Excel datoteku.

```
# Funkcija koja računa mjesec boravka na osnovu sredine između
#Check-in-a i Check-out-a.
def calculate_stay_month(checkin_date, checkout_date):
    checkin = pd.to_datetime(checkin_date)
    checkout = pd.to_datetime(checkout_date)
    midpoint = checkin + (checkout - checkin) / 2
    return midpoint.month

# Funkcija za obradu Excel datoteke i spremanje u novu datoteku
def process_guest_data_from_excel(file_path, sheet_name, new_file_path):
    # Čitanje podataka iz Excel datoteke
    guest_data = pd.read_excel(file_path, sheet_name=sheet_name)

    # Provjera postoji li 'DateOfBirth' stupav, ako ne stvori ga
    if 'DateOfBirth' not in guest_data.columns:
        guest_data['DateOfBirth'] = None

    # Računaje mjeseca boravka na osnovu Check-in-a i Check-out-a
    guest_data['stay_month'] = guest_data.apply(lambda row:
                                                calculate_stay_month(row['CheckinDate'],
                                                                      row['CheckoutDate']), axis=1)

    # Dodjeljivanje datuma rođenja na osnovu mjeseca boravka
    for i, row in guest_data.iterrows():
        if pd.isnull(row['DateOfBirth']):
            guest_data.at[i, 'DateOfBirth'] = assign_dob_based_on_month(row['stay_month'])

    # Provjera podataka za obitelji
    guest_data = assign_family_dob(guest_data)

    # Spremanje podataka u novi file
    guest_data.to_excel(new_file_path, sheet_name=sheet_name, index=False)

    print(f"Podaci sa generiranim datumima rođenja spremljeni su u: {new_file_path}")

file_path = 'C:\\Users\\Josipa\\Desktop\\rezervacije-hotel.xlsx'
new_file_path = 'C:\\Users\\Josipa\\Desktop\\rezervacije-hotel-2023DoB.xlsx'
sheet_name = '2023'

process_guest_data_from_excel(file_path, sheet_name, new_file_path)
```

*Kod 5. Generiranje datuma rođenja - 2*

Posljednja skripta prikazuje generiranje jedinstvenog ID-a gosta, a dobiveni podaci će se kasnije koristiti za popunjavanje tablice o gostima u bazi podataka.

```
import pandas as pd

file_path = 'C:\\Users\\Josipa\\Desktop\\rezervacije-hotel.xlsx'

sheet_names = ['2019', '2020', '2021', '2022', '2023']

all_guests = pd.DataFrame()

for sheet in sheet_names:
    df = pd.read_excel(file_path, sheet_name=sheet)
    df['SheetName'] = sheet
    all_guests = pd.concat([all_guests, df], ignore_index=True)

columns_to_keep = ['FirstName', 'LastName', 'Gender', 'DateOfBirth', 'Country']
all_guests_filtered = all_guests[columns_to_keep]

unique_guests = all_guests_filtered.drop_duplicates(subset=['FirstName', 'LastName'],
                                                    keep='first').reset_index(drop=True)

unique_guests['GuestID'] = range(6953, 6953 + len(unique_guests))

all_guests_with_id = pd.merge(all_guests, unique_guests[['FirstName', 'LastName', 'GuestID']],
                              on=['FirstName', 'LastName'], how='left')

output_file = 'C:\\Users\\Josipa\\Desktop\\rezervacije-hotel-GuestIDs.xlsx'
with pd.ExcelWriter(output_file, engine='xlsxwriter') as writer:
    for sheet in sheet_names:
        sheet_data = all_guests_with_id[all_guests_with_id['SheetName'] == sheet].drop(columns=['SheetName'])
        sheet_data.to_excel(writer, sheet_name=sheet, index=False)

print(f"Ažurirani podaci spremljeni u: {output_file}")
```

Kod 6. Generiranje ID-a za goste

Slika 3 prikazuje podatke nakon pripreme. Definiranjem naziva stupaca jasnije je prikazano što podaci označavaju.

1	RoomNum	ReservationID	GuestID	Name	LastName	FirstName	Gender	DateOfBirth	Country	CheckinDate	CheckoutDate	PricePerNight	TotalPrice	Agency	ReservationType	PaymentMethod
2	101	2018/10185	6953	JELENKO, ALOJZ	JELENKO	ALOJZ	M	09/09/1941	SLOVENIJA	30/12/2018	01/01/2019	34.40 €	68.80 €	SAJKO TURIZEM d.o.o.	HGB	card
3	101	2018/10185	6954	JELENKO, VALERIJA	JELENKO	VALERIJA	F	18/02/1972	SLOVENIJA	30/12/2018	01/01/2019	34.40 €	68.80 €	SAJKO TURIZEM d.o.o.	HGB	card
4	101	2018/10185	6955	JELENKO, PETRA	JELENKO	PETRA	F	24/10/2007	SLOVENIJA	30/12/2018	01/01/2019	34.40 €	68.80 €	SAJKO TURIZEM d.o.o.	HGB	card
5	101	2019/100	6956	SALINAS, IZAAK	SALINAS	IZAAK	M	24/05/1979	SIJEDINIJE AN	21/03/2019	23/03/2019	30.00 €	60.00 €	HAPPY TOURSDruzina	IBBG	card
6	101	2019/100	6957	MANSOUR, ANDREW	MANSOUR	ANDREW	M	11/12/1986	SIJEDINIJE AN	21/03/2019	23/03/2019	30.00 €	60.00 €	HAPPY TOURSDruzina	IBBG	card
7	101	2019/100	6958	DE LA ROSA, MATTHEW	DE LA ROSA	MATTHEW	M	14/04/1980	SIJEDINIJE AN	21/03/2019	23/03/2019	30.00 €	60.00 €	HAPPY TOURSDruzina	IBBG	card
8	101	2019/280	6959	CHENG, SHENG CHIH	CHENG	SHENG CHIH	M	20/12/1984	KINA	24/03/2019	25/03/2019	28.00 €	28.00 €	SUNSHINE INTERNATIC	BBG	card
9	101	2019/280	6960	OU, MEI HSUAN	OU	MEI HSUAN	M	25/06/1989	KINA	24/03/2019	25/03/2019	28.00 €	28.00 €	SUNSHINE INTERNATIC	BBG	card
10	101	2019/2729	6961	KANERVA, LEENA ANNE	KANERVA	LEENA ANNELI	F	12/12/1978	FINSKA	26/03/2019	27/03/2019	24.00 €	24.00 €	GRAND HOTEL PARK	HBB	card
11	101	2019/2729	6962	AKREN, TUJIA INGRID	AKREN	TUJIA INGRID	M	18/12/1977	FINSKA	26/03/2019	27/03/2019	24.00 €	24.00 €	GRAND HOTEL PARK	HBB	card
12	101	2019/228	6963	RAJENDRAN, KAMALINI	RAJENDRAN	KAMALINI	F	03/08/1981	VELIKA BRITAN	02/04/2019	09/04/2019	40.00 €	280.00 €	MEETING POINT CROA	HB	card
13	101	2019/228	6964	RAJENDRAN, KANAGARAJ	RAJENDRAN	KANAGARATNAM	M	08/05/1979	VELIKA BRITAN	02/04/2019	09/04/2019	40.00 €	280.00 €	MEETING POINT CROA	HB	card
14	101	2019/228	6965	RAJENDRAN, YALINI	RAJENDRAN	YALINI	F	15/06/2014	VELIKA BRITAN	02/04/2019	09/04/2019	32.00 €	224.00 €	MEETING POINT CROA	HB	card
15	101	2019/108	6966	JURIC, MARTA	JURIC	MARTA	F	18/11/1985	HRVATSKA	09/04/2019	12/04/2019	40.67 €	122.01 €	EŠKO d.o.o. za trgovinu	HB	card

Slika 3. Izgled podataka nakon pripreme

## 4. TRANSAKCIJSKE BAZE PODATAKA – OLTP

U modernim poslovnim sustavima transakcijske baze podataka igraju ključnu ulogu u upravljanju svakodnevnim operacijama. Dizajnirane su za upravljanje transakcijama u stvarnom vremenu koje zahtijevaju trenutnu obradu. Podržavaju velike količine transakcija i pružaju pouzdano upravljanje podacima. Osiguravaju da se sve transakcije obrađuju ispravno i dosljedno, te pomažu organizacijama da prate i upravljaju svojim operacijama u stvarnom vremenu, pružajući uvide potrebne za odgovore na dinamična poslovna okruženja.

Prilikom oblikovanja transakcijske baze podataka koristi se relacijski model gdje su podaci normalizirani do oblika treće normalne forme (*Third Normal Form* - 3NF) kako bi se smanjila redundantnost i osigurao integritet podataka. Svrha korištenja ove tehnike je izgradnja baze podataka koja je optimizirana za transakcijske aplikacije koje se izvode povrh baze podataka. Primjeri korištenja transakcijskih sustava mogu biti transakcije e-trgovine, bankovni sustavi, CRM sustavi za upravljanje odnosima s klijentima i sl. [3].

### 4.1. Ključna svojstva transakcijskih baza podataka

Transakcijske baze podataka koriste vrstu obrade podataka koja se naziva mrežna obrada transakcija (On Line Transaction Processing - OLTP). OLTP omogućuje istovremeno izvršavanje brojnih transakcija [4].

#### 1. Istodobni pristup

OLTP sustavi često imaju veliki broj korisnika koji istovremeno pristupaju istim podacima. Istodobnost upravlja procesom pisanja kada dva različita korisnika pokušaju poduzeti radnju u isto vrijeme. U tom slučaju jedan korisnik mora pričekati dok drugi korisnik ne završi obradu prije nego što može dovršiti svoju radnju. Primjer može biti sprječavanje da dva korisnika kupe posljednji artikl u online trgovini ili da dva putnika rezerviraju isto mjesto u avionu.

#### 2. Skalabilnost

Transakcijske baze podataka moraju biti sposobne za brzo skaliranje kako bi zadovoljile zahtjeve korisnika. Skalabilnost osigurava da sustavi mogu podnijeti skokove prometa, elastično se šireći kada je to potrebno. Primjer je mrežno bankarstvo gdje veći broj klijenata želi pristupiti svojim računima, često u isto, popularno vrijeme.

#### 3. Brzina

Većina digitalnih transakcija mora se izvršiti nevjerojatno brzo. Transakcijske baze podataka mogu rukovati višestrukim istodobnim zahtjevima za čitanje i pisanje jako brzo. Primjer je internetska prodaja gdje tisuće kupaca očekuju brzu i jednostavnu kupnju, naplatu i provjeru plaćanja, brzina obrade mora biti u korak.

#### 4. Sigurnost podataka

Transakcijske baze podataka sadrže velike količine osjetljivih podataka o korisnicima. Uspješan upad u jedan od ovih sustava može ugroziti korisničke podatke i informacije o računima. Održavanje sigurnosti podataka štiti podatke o korisničkim računima i ugled poslovnih sustava.

## 5. Mogućnost oporavka

U slučajevima pada sustava, transakcijske baze podataka su dizajnirane da se mogu obnoviti, štiteći integritet podataka i osiguravajući kontinuitet za korisnika.

### 4.2. ACID svojstva

Transakcijske baze podataka pridržavaju se ACID svojstva (*Atomicity, Consistency, Isolation, Durability*) [4].

Atomarnost (*Atomicity*) zahtijeva da svaki korak u transakciji bude dovršen u cijelosti prije nego što se preda u bazu podataka. Ako dođe do neuspjeha u bilo kojoj fazi transakcije, ostali koraci također neće uspjeti ili će se poništiti.

Konzistentnost (*Consistency*) osigurava da se dogodi jedna od dvije stvari: transakcija je ili uspješno zapisana u bazu podataka, premještajući je iz jednog važećeg stanja u drugo, ili je automatski vraćena. To omogućuje da baza podataka ostane u stalnom stanju unutarnje dosljednosti.

Izolacija (*Isolation*) služi da bi se istovremene transakcije odvojile jedne od drugih kako bi se spriječili konflikti. Na transakcije koje nisu u potpunosti obrađene druge transakcije ne mogu djelovati ili ih mijenjati.

Dugotrajnost (*Durability*) osigurava da se rezultati završenih transakcija trajno pohrane, čak i u slučaju pada sustava.

### 4.3. Tipovi transakcija u transakcijskim bazama podataka

Unutar transakcijskih baza podataka postoje različite vrste pohranjenih transakcija. Te vrste transakcija mogu biti pojedinačne i višestruke transakcije [5].

Pojedinačna transakcija unutar baze podataka odnosi se na jedinicu rada koja se sastoji od jedne ili više operacija baze podataka. Bez obzira na to koliko je operacija baze podataka uključeno, transakcija mora biti potpuni zapis transakcije prema definicijama sustava za upravljanje bazom podataka. Evidencija ovih transakcija može se pronaći u dnevniku transakcija. Primjer može biti poslana narudžba putem online trgovine ili novac podignut s bankomata [5].

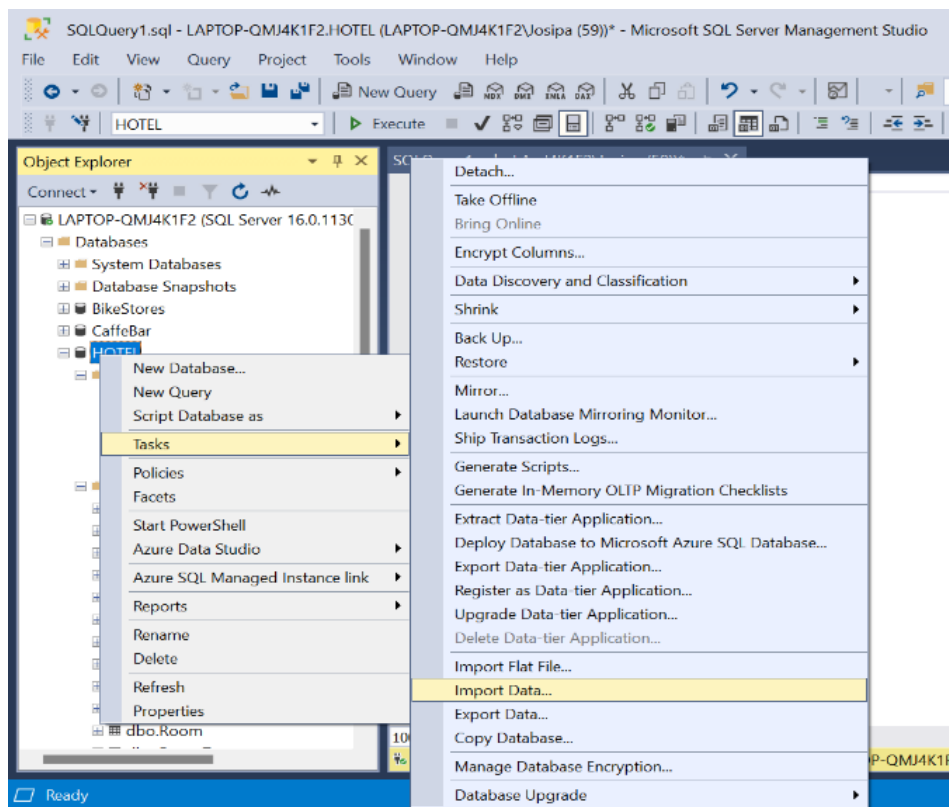
Višestruka transakcija naziva se i distribuirana transakcija, a ona uključuje višestruke, međuzavisne transakcije koje se protežu kroz niz različitih baza podataka i sustava. Unutar svake od ovih baza podataka ili sustava može se dogoditi jedna ili više operacija baze podataka. Primjer ovih transakcija uključuje transakciju s više dokumenata gdje se podaci o klijentu ažuriraju, a povezane fakture i naplata također moraju biti ažurirani, ili provjeru ovisnosti kroz IT projekt putem softvera za upravljanje projektima [5].

Transakcijske baze podataka su optimizirane za rad s velikim brojem istovremenih korisnika, osiguravaju dosljednost i točnost podataka bez obzira koliko transakcija istovremeno pristupa bazi.

#### 4.4. Izrada relacijske baze podataka i učitavanje podataka

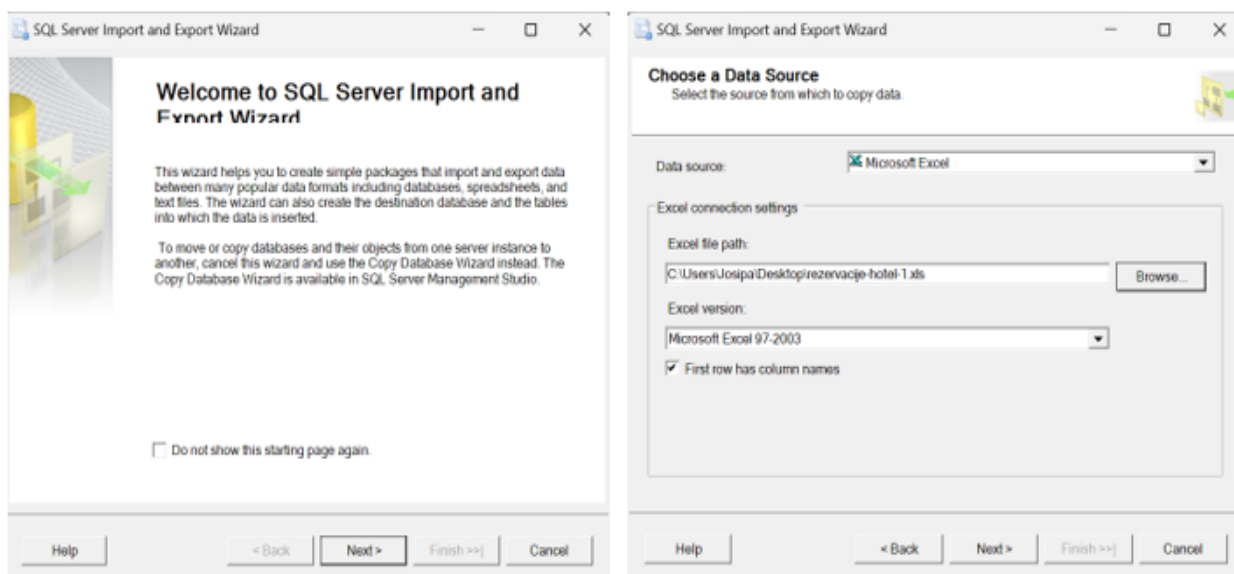
Za izradu relacijske baze podataka koristit će se *SQL Server Management Studio* (SSMS). To je softverska aplikacija koju je razvio Microsoft, a koristi se za konfiguriranje, upravljanje i administriranje svih komponenti unutar *Microsoft SQL Servera*. U SSMS-u stvara se nova baza podataka naziva HOTEL sa sljedećim tablicama: *Hotel*, *Room*, *RoomType*, *Reservation*, *Guest*, *ReservationType*, *Agency* i *Payment*. Nakon stvaranja tablica i prethodno objašnjenih stupaca koje će sadržavati, potrebno je definirati relacije između tablica korištenjem stranih ključeva. Nakon stvaranja baze podataka slijedi proces uvoza pripremljenih podataka.

Podaci će se u bazu učitati korištenjem *Import and Export Wizarda*. Desnim klikom miša na bazu otvara se prozor naredbi od kojih se izabire 'Tasks', pa 'Import Data' (Slika 4).



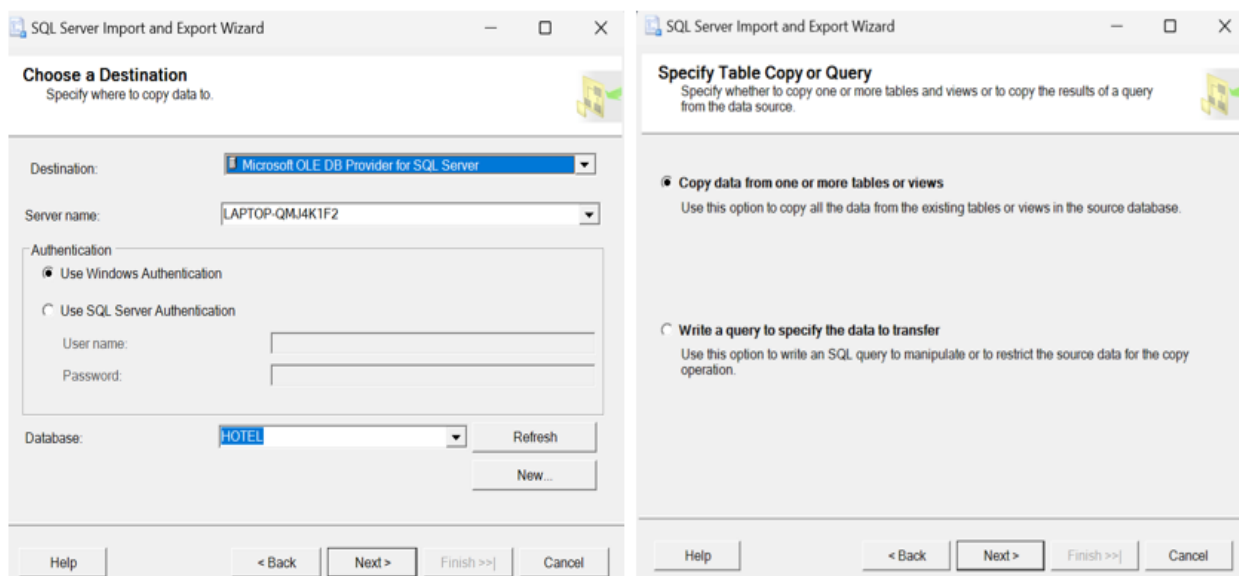
Slika 4. Uvoz podataka - 1

Prvi korak je odabir izvora podataka iz kojeg će se podaci kopirati (Slika 5). U ovom slučaju to je Excel datoteka. Navodi se put do datoteke, bira se verzija Excel-a koja se koristi i označava se da prvi red sadrži nazive stupaca. Ako prvi red ne sadrži nazive stupaca potrebno je onemogućiti ovu opciju. Ako datoteka sadrži nazive stupaca, a opcija je onemogućena, onda će se nazivi stupaca učitati kao prvi redak podataka. Ako ipak ne postoje nazivi stupaca, a navedena opcija je označena, onda će se prvi red podataka u datoteci tretirati kao nazivi stupaca.



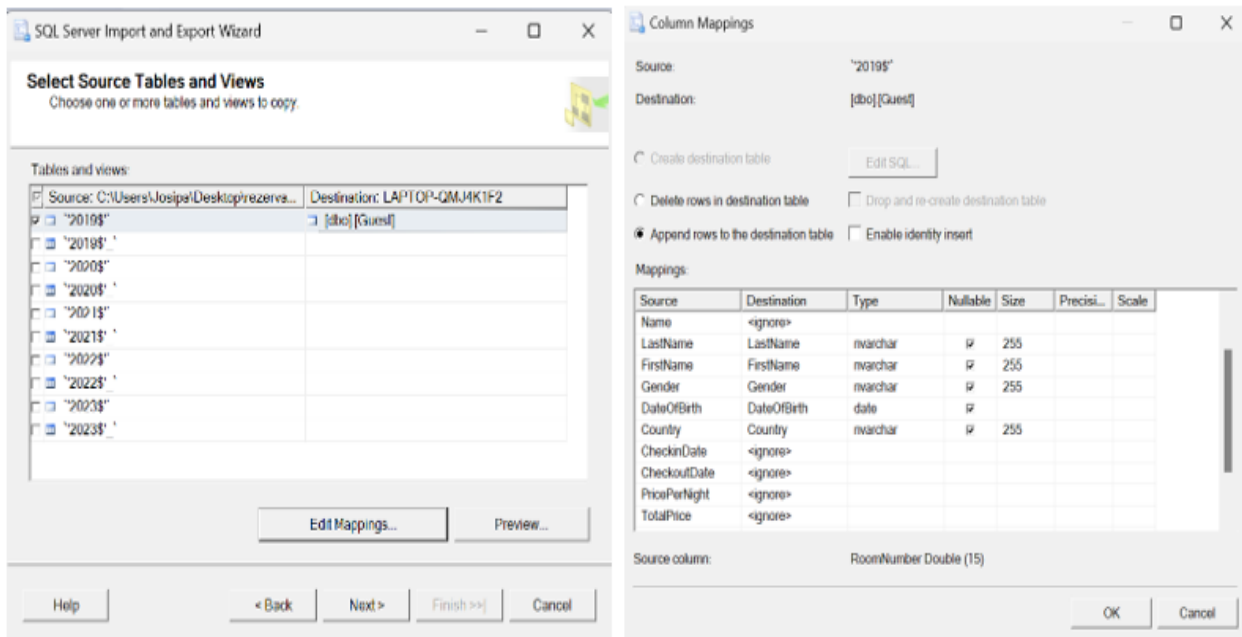
Slika 5. Uvoz podataka – 2

Za destinaciju u koju će se kopirati podaci odabire se prethodno stvorena baza podataka, a da bi to bilo moguće potrebo je imati instaliran *'Microsoft OLE DB Provider for SQL Server'*. Koristi se *'Windows Authentication'* i odabire baza podataka nazvana HOTEL. Idući korak nudi dvije opcije, kopiju podataka ili pisanje SQL upita za prijenos podatka (Slika 6). U ovom slučaju izabrat će se kopiranje podataka.



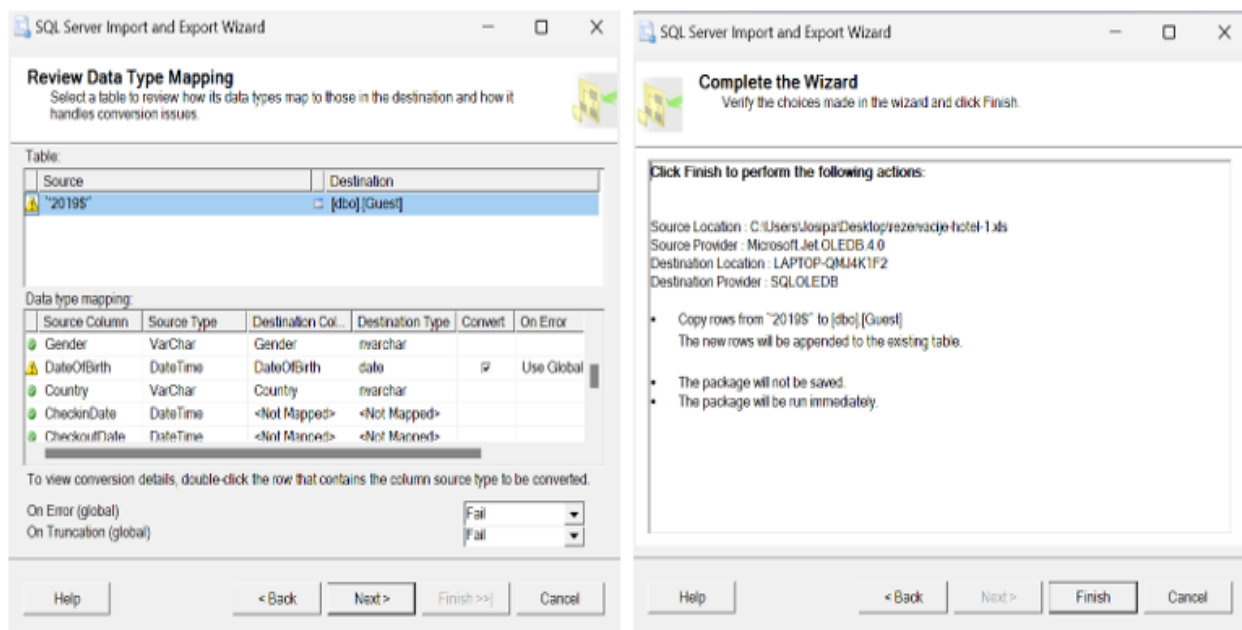
Slika 6. Uvoz podataka – 3

Sljedeći korak je odabir izvora podataka i tablica u koju će se kopirati podaci (Slika 7). *'Edit Mappings'* opcija nudi pregled kako se povezuju podaci, odnosno što će se gdje kopirati u bazu podataka.



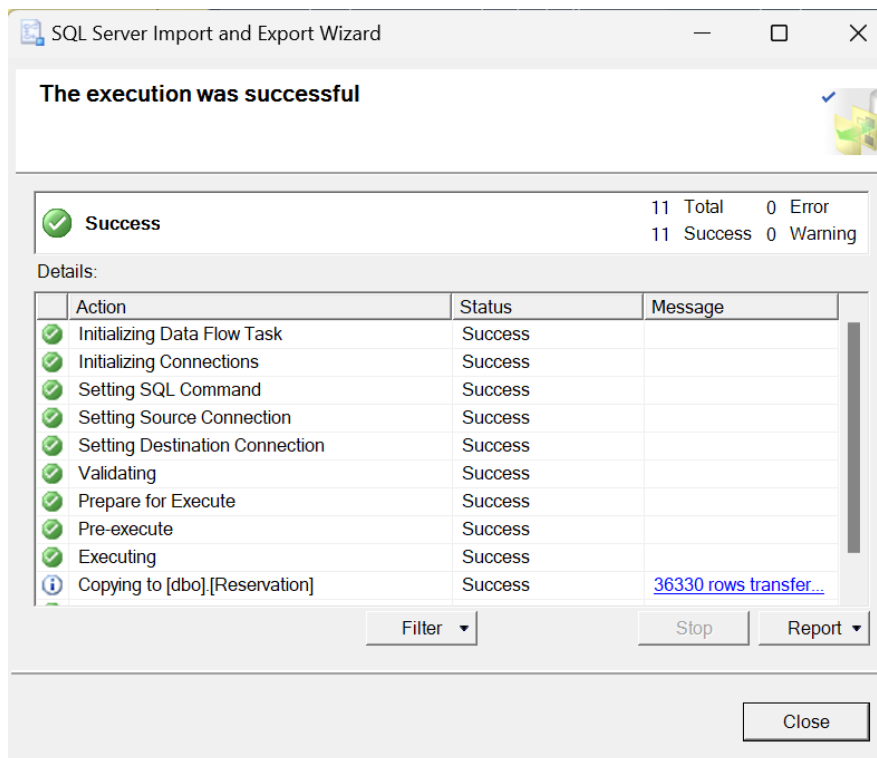
Slika 7. Uvoz podataka – 4

Zadnji korak je mapiranje tipova podataka iz izvorišta s odredištem (Slika 8). Ovdje se najčešće javljaju greške prilikom kopiranja podataka, pa je potrebno osigurati pravilno definirane tipove podataka i u izvornim podacima i u odredišnoj bazi prije samog kopiranja podataka. Odabirom 'Finish' opcije potvrđuju se prethodni koraci i pokreće se kopiranje podataka.



Slika 8. Uvoz podataka - 5

Slika 9 prikazuje uspješan proces kopiranja podataka.



*Slika 9. Završetak procesa uvoza podataka*

Navedeni proces je potrebno ponoviti za sve skupove podataka koje treba učitati u bazu.

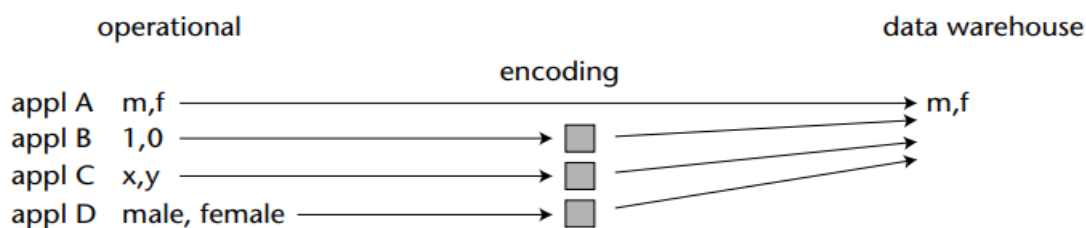


## 5. SKLADIŠTA PODATAKA – OLAP

Prema definiciji W. Inmona u knjizi koju je objavio 1992., skladište podataka je subjektivno orijentiran, integriran, nepromjenjiv i vremenski ovisan skup podataka za podršku procesima odlučivanja [6].

Subjektivno orijentiran znači da su podaci organizirani po poslovnim subjektima, odnosno temama. U transakcijskom sustavu možemo imati npr. sustav za plaće, sustav za prodaju, sustav za nabavu, dok bi u skladištu podataka ti podaci bili organizirani oko zaposlenika, potrošača, dobavljača. Ovakav pristup omogućuje prikupljanje podataka koji su grupirani prema specifičnim područjima interesa i koji su relevantni za donošenje poslovnih odluka, npr. plaće po zaposlenicima, prodaja po dobavljaču i sl.

Integriran podrazumijeva da su podaci u skladištu pohranjeni kao jedinstvena, konzistentna cjelina. Podaci u skladište podataka najčešće dolaze iz više različitih izvora, te je potrebno uskladiti nazive i strukture raznih varijabli, mjerne jedinice, ključeve i sl. Klasičan primjer jest usklađivanje formata prikaza datuma iz raznih izvora ili spol osobe koji je negdje prikazan kao M/Ž, F/M ili 0/1. Proces integracije podataka jedan je od najzahtjevnijih procesa pri skladištenju podataka, a rezultat su konzistentni i pravovaljani podaci koji prikazuju jedinstvenu sliku poslovnog procesa.



Slika 10. Proces integracije podataka [6]

Nepromjenjiv znači da se podaci ne mijenjaju u skladištu podataka. Novi podaci se periodički dodaju, ali stari se ne mijenjaju, za razliku od transakcijske baze podataka gdje su učestale i operacije izmjene i brisanja podataka. Ako u nekom slučaju i dođe do promjene podataka to se rješava tako da se staro stanje zadrži i označi kao nevažeće, a novo stanje se bilježi novim retkom. Ovakav način osigurava konzistentnost povijesnih informacija koje su bitne za poslovne analize.

Vremenski različit podrazumijeva da skladište ima i vremensku dimenziju koja omogućuje pregled podataka u vremenskom kontekstu. Svaki zapis u skladištu podataka nosi oznaku vremena koja odražava kada su podaci prikupljeni ili obrađeni, te omogućuje analizu povijesnih trendova i uvida kroz različite vremenske okvire. To ne znači da se podaci mijenjaju s vremenom (kao npr. stanje na bankovnom računu) već da se podaci mogu sagledati u vremenu, odnosno možemo vidjeti kakva je, npr., bila prodaja po godinama, mjesecima, kvartalima. Dakle, podaci u skladištu su slijed uzoraka stanja u vremenu iz transakcijske baze.

S druge strane, R. Kimball definira skladište podataka kao kopiju transakcijskih podataka koji su posebno strukturirani za upite i analize [7].

### 5.1. Ciljevi izgradnje skladišta podataka (prema R. Kimballu):

Za razvoj dobrog skladišta podataka ključno je odrediti potrebe i ciljeve organizacije. Prema tome, skladište podataka mora [7]:

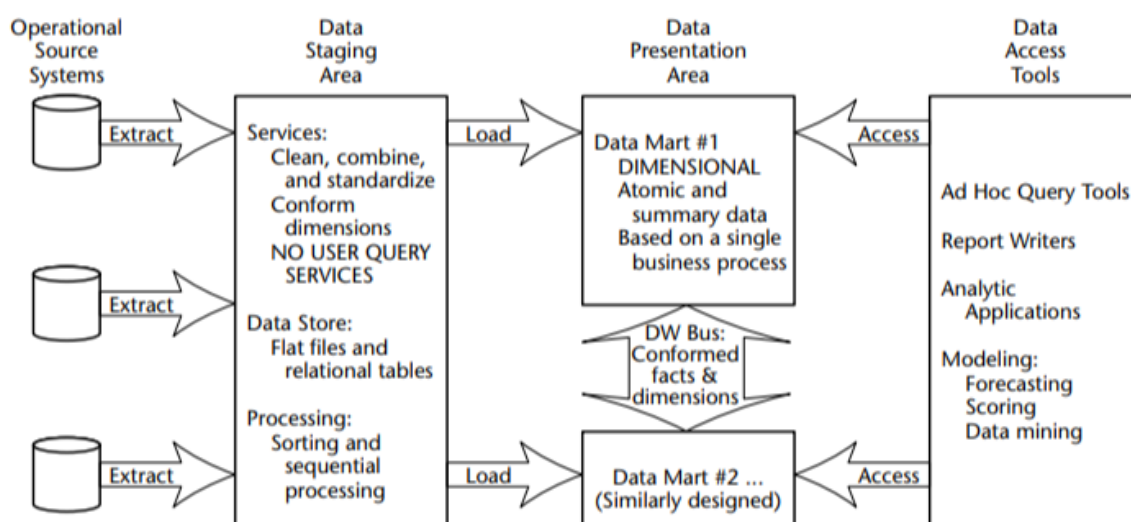
1. Učiniti informacije organizacije lako dostupnima  
Skladište podataka mora učiniti informacije organizacije lako dostupnima za poslovne korisnike. To uključuje dizajniranje za jasnoću i jednostavnost korištenja kako bi korisnici mogli intuitivno pristupiti podacima i razumjeti ih. Alati bi trebali omogućiti besprijekornu manipulaciju podacima, obično poznatu kao „*slicing and dicing*“, istovremeno osiguravajući brzo vrijeme odgovora na upite.
2. Dosljedno prezentirati informacije organizacije  
Podaci u skladištu moraju biti vjerodostojni. Moraju biti pažljivo sastavljeni iz raznih izvora organizacije, očišćeni, kvalitetni i pušteni na korištenje tek kada su za to prikladni. To znači da svi podaci moraju biti evidentirani i potpuni.
3. Biti prilagodljivo i otporno na promjene  
Promjene u skladištima podataka ne bi smjele utjecati na postojeće podatke. Ako se traže odgovori na nova pitanja ili se dodaju podaci u skladište, postojeći podaci bi trebali ostati nepromijenjeni. Ako se neki podaci modificiraju to se mora odraditi na ispravan način.
4. Sigurno zaštititi naše podatke  
Skladišta podataka moraju kontrolirati pristup povjerljivim informacijama organizacija, štiteći ih od potencijalne zlouporabe. Učinkoviti sigurnosni protokoli ključni su za održavanje povjerljivosti i integriteta pohranjenih informacija.
5. Služiti kao temelj za bolje donošenje odluka  
Skladište podataka treba služiti kao čvrst temelj za donošenje bitnih odluka na temelju informacija. Pružajući točne i relevantne podatke pomaže poslovnim organizacijama, usklađujući se s ulogom sustava za podršku odlučivanju.
6. Biti prihvaćeno od strane poslovnih zajednica  
Dizajn prilagođen korisniku i jednostavnost korištenja ključni su za poticanje usvajanja. Ako poslovni korisnici prihvate skladište podataka nakon obuke i nastave se baviti njime, to označava učinkovitu implementaciju.

Izgradnja uspješnog skladišta podataka zahtijeva razumijevanje i prilagođavanje jedinstvenim zahtjevima poslovnih korisnika i poslovnih procesa koje oni obrađuju.

## 5.2. Komponente skladišta podataka

Svaka komponenta skladišta podataka služi određenoj funkciji. Potrebno je naučiti strateški značaj svih komponenti i kako njima učinkovito upravljati. Jedna od najvećih prijetnji uspješnom skladištu podataka je miješanje uloga i funkcija komponenti [7].

Kao što je prikazano na slici 11, postoje četiri različite komponente skladišta podataka. To su sustavi operacijskih izvora, područje pripreme podataka, područje prezentacije podataka i alati za pristup podacima.



Slika 11. Komponente skladišta podataka [7]

### 1. Operational Source Systems

Prvi sloj predstavlja izvore podataka, odnosno sustave evidencije koji bilježe poslovne transakcije. Izvori podataka tipično budu relacijske baze podataka, tekstualne datoteke, Excel datoteke, web servisi i sl.

### 2. Data Staging Area

Područje postavljanja podataka ujedno je i područje pohrane podataka i skup procesa koji se nazivaju ETL (*Extract, Transform, Load*). Ovaj sloj nije dostupan krajnjim korisnicima, te ne može pružiti usluge upita i prezentacije podataka. To je za korisnika skriveno područje, gdje se podaci čiste, transformiraju i na kraju izvoze u jedan ili više spremnika podataka.

### 3. Data Presentation Area

Sloj prezentacije podataka je područje gdje su podaci organizirani, pohranjeni i dostupni za izravno postavljanje upita korisnicima, piscima izvještaja i drugim aplikacijama za analitiku.

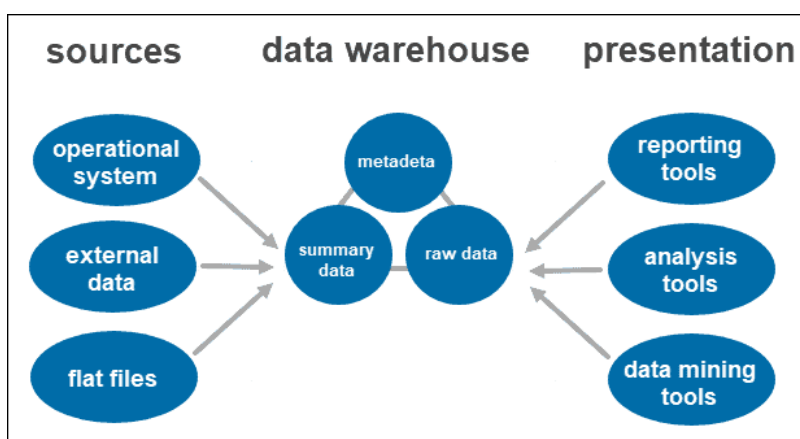
### 4. Data Access Tools

Alati za pristup podacima omogućuju korisnicima korištenje prezentacijskog područja za analitičko donošenje odluka. Postoje brojni alati za pregled, analizu i vizualizaciju podataka, a neki od njih su Power BI, Tableau, QlikView, Looker Studio, Sisense...

### 5.3. Arhitekture skladišta podataka

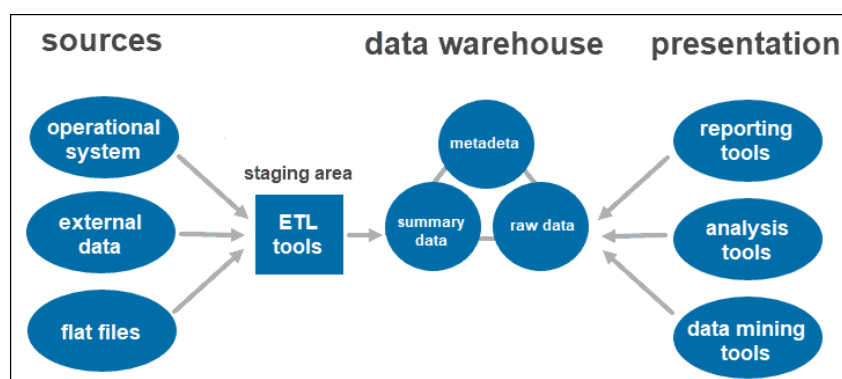
Postoje tri glavne vrste arhitekture skladišta podataka: jednoslojna, dvoslojna i troslojna arhitektura. Svaka od navedenih arhitektura ima specifične karakteristike, prednosti i nedostatke, a odabir određene arhitekture ovisit će o potrebama analiza, količini podataka te zahtjevima organizacija.

Jednoslojna (*Single-tier*) arhitektura (Slika 12) je najjednostavniji oblik, gdje se svi podaci, analitički alati i korisnički pristup nalaze na istom sloju. Ovakav pristup arhitekture skladišta podataka nije često prakticiran. Glavni cilj je uklanjanje redundancije minimiziranjem količine pohranjenih podataka. Glavni nedostatak ove arhitekture je što nema komponente koja razdvaja analitičku i transakcijsku obradu [8].



Slika 12. Jednoslojna arhitektura skladišta podataka [8]

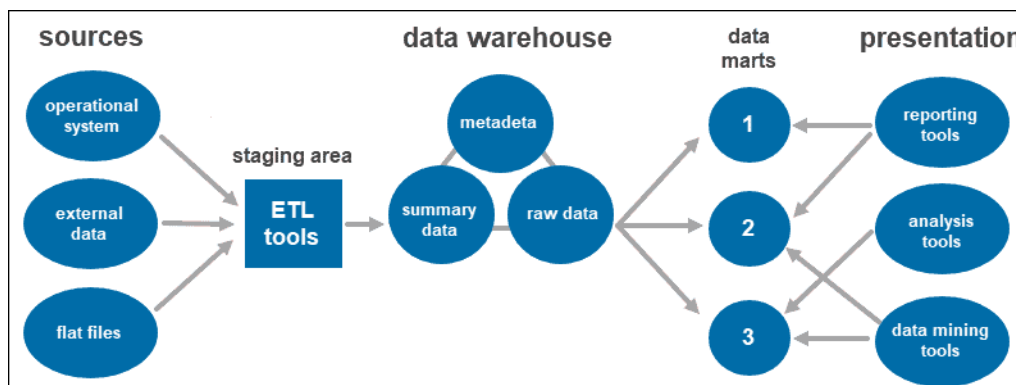
Dvoslojna (*Two-tier*) arhitektura (slika 13) sadrži sloj pripreme podataka (*Data Staging Area - DSA*) što omogućuje čišćenje i pripremu svih podataka koji ulaze u skladište podataka. Dodavanjem ovog sloja osigurano je da su podaci koji se učitavaju u skladište očišćeni i u odgovarajućem formatu, što omogućuje i bolje upravljanje velikim količinama podataka. U odnosu na jednoslojnu arhitekturu veća je složenost implementacije, te zbog procesa obrade i pripreme podataka može doći do kašnjenja u dostupnosti podataka.



Slika 13. Dvoslojna arhitektura skladišta podataka [8]

Troslojna (*Three-tier*) arhitektura (slika 14) je najčešći odabir prilikom izgradnje skladišta podataka. Prvi sloj je baza podataka gdje se spremaju očišćeni i transformirani podaci. Drugi sloj je aplikacijski sloj koji uređuje podatke kako bi bili prikladni za analize. Treći sloj je mjesto gdje korisnik pristupa podacima i komunicira s njima, koristi alate za kreiranje izvještaja, upita,

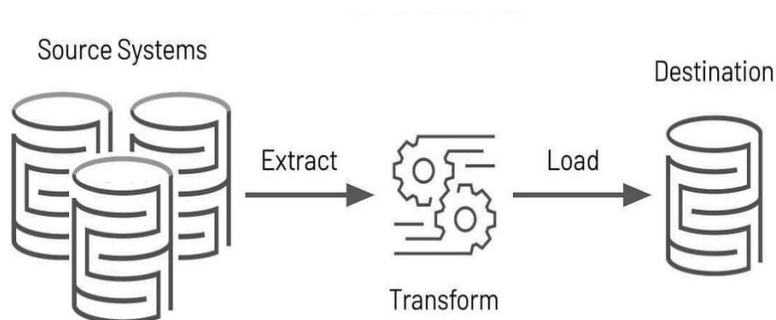
alate za analizu ili rudarenje podataka [8]. Ovakav model omogućuje bolju organizaciju, veću skalabilnost i optimizaciju analitičkih procesa, ali je također i najzahtjevniji model za implementaciju i održavanje.



Slika 14. Troslojna arhitektura skladišta podataka [8]

#### 5.4. ETL procesi

ETL (*Extract, Transform, Load*) označava procese prema kojima se podaci iz raznih izvora pripremaju za pohranu u skladište podataka (Slika 13).



Slika 15. Prikaz ETL procesa [13]

*Extract* – Ekstrakcija podataka je prvi proces, a predstavlja izdvajanje relevantnih podataka iz izvora podataka.

*Transform* – Transformacija je korak u kojem se izdvojeni podaci prebacuju u područje za pripremu gdje se čiste i transformiraju u skladu sa shemom odredišnog skladišta podataka.

*Load* – Učitavanje je dio procesa gdje se pripremljeni podaci učitavaju u činjeničnu tablicu i dimenzijske tablice skladišta podataka. Može se koristiti pojedinačno učitavanje niza zapisa ili masovno učitavanje (*bulk load*) korištenjem određenog alata namijenjenog za rad s odabranim sustavom za upravljanje bazom podataka.

ETL procesi omogućuju integraciju podataka iz različitih izvora, osiguravajući da skladište podataka ima cjelokupan pregled poslovanja organizacije. Transformacijom podataka prije nego što se učitaju u skladište, ETL procesi pomažu osigurati da su podaci točni, dosljedni i spremni za analizu.

## 5.5. Metodologije izgradnje skladišta podataka

Iako danas postoje brojne metodologije za izgradnju skladišta podataka, sve se temelje na dva pristupa. To su odozgo prema dolje (*top-down*) pristup i odozdo prema gore (*bottom-up*) pristup.

Pristup odozgo prema dolje, koji je uveo W. Inmon, oslanja se na normaliziranu *Entity Relationship* (ER) shemu s centraliziranim privremenim spremištem podataka (*Data Staging Area* - DSA). Ovaj pristup zahtijeva veće inicijalne troškove i duže vrijeme razvoja, jer se oslanja na detaljno planiranje arhitekture.

Drugi pristup, odozdo prema gore, temelji se na nezavisnim segmentima (*Data Marts* - DM) koji se integriraju u cjelinu skladišta podataka. Omogućuje brže rezultate jer se fokusira na inkrementalni razvoj nezavisnih dijelova, koji se mogu prilagođavati specifičnim potrebama krajnjih korisnika. Za ovakav pristup se obično koristi dimenzijska zvjezdasta shema, koja omogućuje lakšu analizu podataka.

U današnje vrijeme sve češće se koristi hibridni pristup, koji se naziva i *middle-out* pristup, a on kombinira prednosti dva navedena pristupa. Postoje i metodologije koje uzimaju u obzir poslovne ciljeve (*Goal-Driven Methodology*) ili korisničke potrebe (*User-Driven Methodology*), kako bi se što bolje prilagodile specifičnim zahtjevima poslovanja. Bez obzira na odabranu metodologiju, uspješna izgradnja skladišta podataka zahtijeva temeljito analiziranje i prikupljanje poslovnih i korisničkih zahtjeva, te usklađivanje s tehnološkim promjenama i potrebama tržišta.

## 5.6. Kvaliteta podataka u skladištima podataka

Analize podataka i izvještaji kreirani na osnovu tih podataka uvelike ovise o kvaliteti podataka pohranjenih u skladištu. Što su podaci u skladištu kvalitetniji i pouzdaniji, to su i rezultati analiza relevantniji. Kvaliteta podataka direktno utječe na prihvaćenost i vrijednost koju skladište podataka donosi korisnicima. Bez kvalitetnih podataka nijedno skladište ne može dodati vrijednost poslovanju.

Kvaliteta podataka mjeri koliko dobro skup podataka zadovoljava kriterije točnosti, potpunosti, valjanosti, dosljednosti, jedinstvenosti, pravovremenosti i prikladnosti za svrhu, a ključna je za sve inicijative upravljanja podacima unutar organizacije [9].

Točnost (*Accuracy*) podataka ocjenjuje koliko su vrijednosti podataka u skladu s referentnim izvorom.

Potpunost (*Completeness*) predstavlja količinu podataka koja je upotrebljiva ili potpuna. Ako postoji visok postotak vrijednosti koje nedostaju, to može dovesti do pogrešne analize ako podaci nisu reprezentativni za tipični uzorak podataka.

Valjanost (*Validity*) ocjenjuje koliko podaci odgovaraju zadanim pravilima ili formatima. Provjerava se zadovoljavaju li podaci pravila poslovanja, poput određenih tipova podataka, raspona vrijednosti ili obrazaca. Na primjer, broj telefona bi trebao imati standardni format koji olakšava njegovu identifikaciju i upotrebu.

Jedinstvenost (*Uniqueness*) se odnosi na količinu dupliciranih podataka u skupu podataka. Na primjer, u pregledu podataka o korisnicima očekuje se da svaki korisnik ima jedinstveni ID korisnika.

Pravovremenost (*Timeliness*) se odnosi na dostupnost podataka unutar očekivanog vremenskog perioda. U nekim situacijama, podaci moraju biti dostupni u stvarnom vremenu kako bi se mogli pravovremeno koristiti.

Dosljednost (*Consistency*) provjerava jesu li podaci iz više različitih skupova podataka usklađeni.

Prikladnost za svrhu (*Fitness for purpose*) osigurava da su podaci prilagođeni specifičnim poslovnim potrebama. Na primjer, podaci koji nisu dovoljno detaljni ili koji ne sadrže relevantne informacije možda neće biti prikladni za analize koje su potrebne organizaciji.

Loši podaci mogu imati značajne poslovne posljedice za organizacije. Ako se zanemari kvaliteta podataka, korisnici skladišta podataka imat će netočne i nepotpune podatke. To odmah znači da podaci nisu reprezentativni i da je analitika pogrešna. Nekvalitetni podaci u velikom broju slučajeva izgradnje skladišta podataka mogu izazvati neuspjeh, a samim tim i otpor krajnjih korisnika. Poboljšana kvaliteta podataka povećava povjerenje u donošenju bitnih poslovnih odluka, omogućuje bolju korisničku uslugu, povećava mogućnost dodavanja bolje vrijednosti uslugama, te smanjuje rizik od donošenja pogrešnih odluka. Osiguravanje točnosti, potpunosti, konzistentnosti i drugih bitnih dimenzija kvalitete podataka i dalje predstavlja izazov za mnogobrojne organizacije.

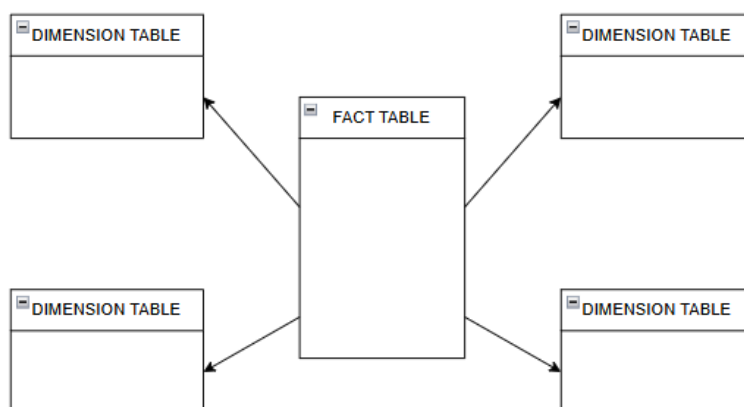
## 6. DIMENZIJSKO MODELIRANJE

Dimenzijsko modeliranje je tehnika za dizajn baza podataka koja se najčešće koristi u skladištima podataka i poslovnoj analitici. Cilj ove tehnike je pojednostaviti i organizirati podatke na način koji će omogućiti brzo i učinkovito pretraživanje, te analizu podataka i izvještavanje. Ralph Kimball u svojoj knjizi navodi da je dimenzijsko modeliranje novi naziv za staru tehniku koja čini baze podataka jednostavnima i razumljivima [7].

Dimenzijsko modeliranje omogućuje krajnjim korisnicima jednostavnost korištenja, efikasniji i smisleniji pristup informacijama koje su potrebne za kreiranje poslovnih izvještaja i donošenje bitnih poslovnih odluka u organizacijama. Ključni pojmovi koji se vežu uz dimenzijsko modeliranje su činjenična tablica, dimenzijska tablica, zvjezdasta shema, pahuljasta shema, a svi će biti objašnjeni u nastavku.

### 6.1. Zvjezdasta shema (*Star schema*)

U dimenzijskom modelu podatke prikazujemo kao činjenice ili kao dimenzije. Zvjezdasta shema je najčešći oblik dimenzijskog modeliranja. U središtu ovog modela je činjenična tablica, koja je okružena s više dimenzijskih tablica (Slika 16).



Slika 16. Star schema

### 6.2. Činjenična tablica (*Fact Table*)

Činjenična tablica je primarna tablica u dimenzijskom modelu. Red u tablici činjenica odgovara mjeri. Sve mjere u činjeničnoj tablici moraju biti iste razine granulacije, odnosno detaljnosti. Najkorisnije činjenice su numeričke i aditivne. Aditivnost je ključna jer se skladišta podataka nikada neće koristiti za dohvaćanje samo jednog retka činjenične tablice, nego za dohvaćanje stotina, tisuća ili čak milijuna redaka činjenica odjednom [7].

Granulacija predstavlja razinu detaljnosti podataka u skladištu. Kada su podaci prikazani s većim brojem detalja, razina granulacije je manja. U suprotnom, kada su podaci prikazani s manje detalja, razina granulacije je veća. Činjenice mogu biti aditivne, semiaditivne i neaditivne. Činjenične tablice prema razini granulacije dijelimo u tri kategorije: transakcijske, periodična snimka stanja i akumulirajuća snimka stanja.

Sve činjenične tablice imaju dva ili više stranih ključeva koji se povezuju s primarnim ključevima dimenzijskih tablica. Kada se svi ključevi u činjeničnoj tablici točno podudaraju sa



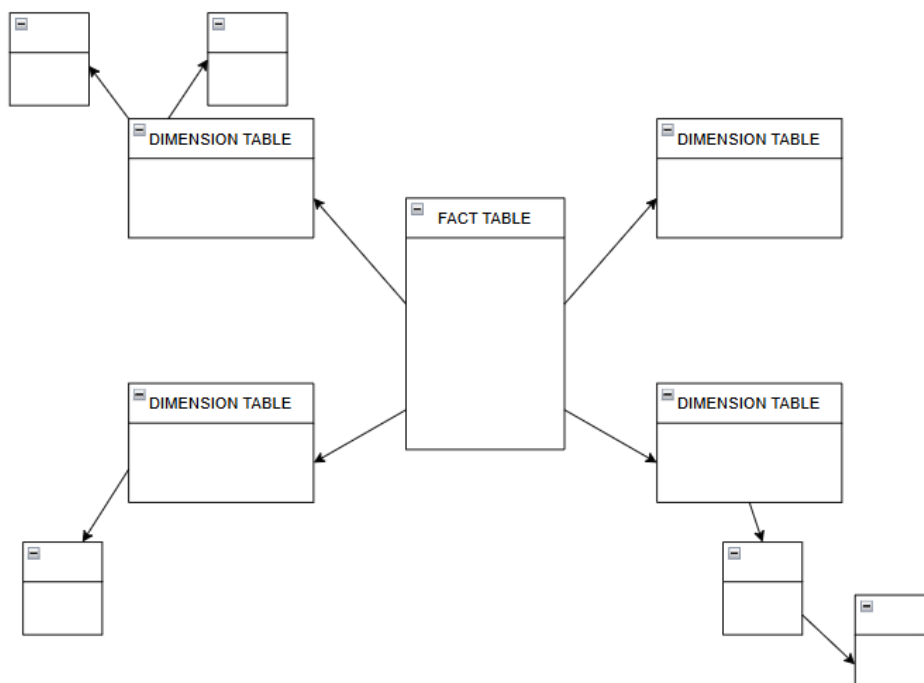
svojim primarnim ključevima u odgovarajućim dimenzijskim tablicama, kažemo da tablice zadovoljavaju referentni integritet. Sama činjenična tablica općenito ima vlastiti primarni ključ koji je sastavljen od podskupa stranih ključeva, a naziva se složeni ili spojeni ključ. U dimenzijskom modelu, svaka tablica koja označava odnos više-prema-više mora biti činjenična tablica. Sve ostale tablice su dimenzijske tablice [7].

### 6.3. Dimenzijske tablice

Dimenzijske tablice sadrže tekstualne opise poslovanja. U dobro dizajniranom dimenzijskom modelu imaju mnogo stupaca ili atributa. Atributi dimenzijskih tablica služe kao primarni izvor za ograničenje upita, grupiranje i oznake izvješća. Ključni su kako bi skladište podataka bilo upotrebljivo i razumljivo. Skladište podataka je onoliko dobro koliko su dobri atributi dimenzijskih tablica. Najbolji atributi su tekstualni i diskretni. Trebali bi se sastojati od stvarnih riječi, a ne od zagonetnih kratica. Dimenzijske tablice su obično visoko denormalizirane [7].

### 6.4. Pahuljasta shema (*Snowflake schema*)

Pahuljasti model je proširena verzija zvjezdastog modela gdje su dimenzijske tablice normalizirane (Slika 17). Na ovaj način upiti prema skladištu podataka postaju složeniji, jer je potrebno spajanje većeg broja tablica. Zbog toga ovakav način modeliranja može biti nepregledan za krajnje korisnike. Prednost je što ovakva struktura može optimizirati prostor za pohranu, jer se smanjuje redundancija podataka.



Slika 17. Snowflake schema

## 6.5. Četiri koraka izgradnje dimenzijskog modela:

Prema Kimballu, proces izgradnje dimenzijskog modela sastoji se od četiri koraka [7]:

1. Odabir poslovnog procesa  
Početni korak u dizajnu dimenzijskog modela je odluka koji poslovni proces analizirati i modelirati.  
Prvi izgrađeni dimenzijski model bi trebao biti onaj s najvećim utjecajem – trebao bi odgovoriti na najhitnija poslovna pitanja i biti lako dostupan za ekstrakciju podataka.  
Odabir poslovnog procesa temelji se na tome koji podaci su ključni za analizu i donošenje odluka.
2. Odabir granulacije činjenične tablice  
Nakon odabira poslovnog procesa potrebno je odrediti razinu detaljnosti. Činjenične tablice sadrže kvantitativne podatke koje će korisnici analizirati. Fokus je na ključnim metrikama koje su specifične za odabrani poslovni proces.  
Poželjno bi bilo razviti dimenzijski model za najsitnije informacije koje obuhvaća poslovni proces. Atomski podaci su najdetaljnije prikupljene informacije i takvi se podaci ne mogu dalje dijeliti.
3. Izbor dimenzija  
Pažljivo definirana granulacija određuje primarnu dimenzionalnost činjenične tablice. Dimenzije su deskriptivni atributi koji omogućuju analizu podataka iz različitih perspektiva.
4. Identifikacija činjenica  
Zadnji korak je određivanje koje činjenice će biti prikazane u činjeničnoj tablici. Činjenice prikazuju ono što mjerimo. Sve činjenice kandidati moraju biti na razini granulacije definirane u drugom koraku. Tipične činjenice su numerički aditivne brojke.

## 7. IZGRADNJA DIMENZIJSKOG MODELA

Skladište podataka prikazano na slici 18 koristit će se za praćenje rezervacija unutar smještajnih jedinica. Za izradu ovog dimenzijskog modela koristit će se zvjezdasta shema. Činjenična tablica `f_reservation` okružena je dimenzijskim tablicama `d_hotel`, `d_guest`, `d_room`, `d_date`, `d_reservationType`, `d_bookingOption` i `d_agency`. U nastavku će biti prikazano stvaranje skladišta podataka, dimenzijskih tablica, činjenične tablice te procedure za prijenos podataka iz transakcijske baze u skladište podataka.



Slika 18. Shema skladišta podataka

## 7.1. Stvaranje skladišta podataka i tablica

U SSMS-u pokreće se novi upit (*New Query*) i upisuje kod za stvaranje nove baze podataka (Kod 7):

```
CREATE DATABASE DW_HOTEL
GO

USE DW_HOTEL
GO
```

*Kod 7. Stvaranje skladišta podataka*

### 7.1.1. Dimenzijska tablica d\_hotel

Dimenzijska tablica d\_hotel koristit će se za pohranu informativnih podataka o hotelu, uključujući ime, adresu, poštanski broj, grad, državu, kontakt e-mail, broj zvjezdica, vrijeme prijave u hotel i vrijeme odjave iz hotela (Kod 8).

```
CREATE TABLE d_hotel (
    HotelKey INT PRIMARY KEY IDENTITY(1,1),
    HotelID INT UNIQUE,
    Name VARCHAR(255),
    Address VARCHAR(255),
    ZIPCode INT,
    City VARCHAR(100),
    Country VARCHAR(100),
    Email VARCHAR(255),
    Stars INT,
    CheckInTime TIME,
    CheckoutTime TIME
);
```

*Kod 8. Stvaranje dimenzijske tablice d\_hotel*

### 7.1.2. Dimenzijska tablica d\_guest

Dimenzijska tablica d\_guest sadrži podatke o gostima koji su imali rezervacije unutar smještajnih jedinica, a to su ime, prezime, država, spol i godine rođenja (Kod 9).

```
CREATE TABLE d_guest (
    GuestKey INT PRIMARY KEY IDENTITY(1,1),
    GuestID INT UNIQUE,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    Country VARCHAR(100),
    Gender VARCHAR(50),
    DateOfBirth DATE
);
```

*Kod 9. Stvaranje dimenzijske tablice d\_guest*

### 7.1.3. Dimenzijska tablica d\_room

Dimenzijska tablica d\_room sadrži informacije o sobama unutar smještajnih jedinica, a to su broj sobe, broj kata na kojem se soba nalazi, tip sobe, opis i sadržaje koje soba posjeduje (Kod 10).

```

CREATE TABLE d_room (
    RoomKey INT PRIMARY KEY IDENTITY(1,1),
    RoomNumber INT UNIQUE,
    FloorNumber INT,
    RoomTypeName VARCHAR(255),
    Description TEXT,
    Amenities TEXT
);

```

*Kod 10. Stvaranje dimenzijske tablice d\_room*

#### 7.1.4. Dimenzijska tablica d\_date

Dimenzijska tablica d\_date sadrži detaljne informacije o datumima, a koristit će se za vremenske analize rezervacija. Za svaki datum generiraju se i dodatni atributi, poput dana u mjesecu, naziv dana u tjednu, redni broj dana u tjednu, broj tjedna u godini, naziv mjeseca, broj mjeseca, kvartal, godina, indikatori vikenda i blagdana i godišnje doba, a potencijalno su zanimljivi za buduće analize podataka (Kod 11).

```

CREATE TABLE d_date (
    DateKey INT PRIMARY KEY,
    FullDate DATE,
    Day INT,
    DayOfWeek VARCHAR(50),
    DayOfWeekNumber INT,
    WeekOfYear INT,
    Month VARCHAR(50),
    MonthNumber INT,
    Quarter INT,
    Year INT,
    IsWeekend BIT,
    IsHoliday BIT,
    DayOfYear INT,
    Season VARCHAR(50)
);

```

*Kod 11. Stvaranje dimenzijske tablice d\_date*

#### 7.1.5. Dimenzijska tablica d\_reservationType

Dimenzijska tablica d\_reservationType pohranjuje sve tipove rezervacija unutar smještajnih jedinica. Neki od tipova rezervacija su noćenje s doručkom, puni pansion, polupansion, dnevni boravak, samo noćenje, itd. Dimenzijska tablica sadrži skraćene nazive, pune nazive i opise svih tipova rezervacija i što je u njih uključeno (Kod 12).

```

CREATE TABLE d_reservationType (
    ReservationTypeKey INT PRIMARY KEY IDENTITY(1,1),
    ReservationTypeID INT UNIQUE,
    ReservationTypeName VARCHAR(255),
    RTFullName VARCHAR(255),
    RTDescription VARCHAR(255)
);

```

*Kod 12. Stvaranje dimenzijske tablice d\_reservationType*

#### 7.1.6. Dimenzijska tablica d\_bookingOption

Budući da su u relacijskoj bazi podataka u tablici Agency pohranjene različite opcije izvora rezervacija, bilo da su to turističke agencije, razne ustanove i slično, u skladištu podataka dimenzijska tablica d\_bookingOption omogućit će filtriranje tih zapisa, te će se u dimenzijskoj tablici d\_agency nalaziti podaci samo o turističkim agencijama (Kod 13).

```
CREATE TABLE d_bookingOption (  
    BookingOptionKey INT PRIMARY KEY IDENTITY(1,1),  
    BookingOptionName VARCHAR(255)  
);
```

*Kod 13. Stvaranje dimenzijske tablice d\_bookingOption*

#### 7.1.7. Dimenzijska tablica d\_agency

Dimenzijska tablica d\_agency pohranjuje podatke o turističkim agencijama, uključujući naziv, adresu, državu i kontakt podatke, a omogućit će detaljnije analiziranje turističkih agencija i rezervacija napravljenih tim putem (Kod 14).

```
CREATE TABLE d_agency (  
    AgencyKey INT PRIMARY KEY IDENTITY(1,1),  
    AgencyID INT UNIQUE,  
    AgencyName VARCHAR(255),  
    AgencyAddress VARCHAR(255),  
    AgencyCountry VARCHAR(100),  
    AgencyEmail VARCHAR(255),  
    AgencyContact NVARCHAR(255)  
);
```

*Kod 14. Stvaranje dimenzijske tablice d\_agency*

#### 7.1.8. Činjenična tablica f\_reservation

Činjenična tablica f\_reservation sadrži ključne podatke o rezervacijama. Više stranih ključeva referenciraju dimenzijske tablice d\_hotel, d\_guest, d\_room, d\_date, d\_bookingOption, d\_agency i d\_reservationType. Tablica sadrži broj rezervacije koji može biti isti za više zapisa u bazi (ReservationNumber), cijenu noćenja (PricePerNight), broj noćenja (NumberOfNights), cijenu koja je umnožak cijene noćenja i broja noćenja (Price) i već sadrži PDV, turističku taksu (GuestTax), ukupnu cijenu sa dodanom turističkom taksom (TotalPrice), PDV koji se računa kao 13% od ukupne cijene (VAT), prihod koji se računa kada se od ukupne cijene oduzme PDV i turistička taksa (Income) te način plaćanja (PaymentType).

Price redak dodan je da bi se mogao izračunati PDV, budući da je on već uračunat u cijene noćenja. U TotalPrice je uključen i GuestTax, koji prema zakonu ne podliježe plaćanju poreza na dodanu vrijednost pa ne bi bilo točno računati PDV korištenjem ovog retka (Kod 15).

```

CREATE TABLE f_reservation (
ReservationKey INT PRIMARY KEY IDENTITY(1,1),
GuestKey INT,
HotelKey INT,
RoomKey INT,
CheckInDateKey INT,
CheckOutDateKey INT,
BookingOptionKey INT,
AgencyKey INT,
ReservationTypeKey INT,
ReservationNumber NVARCHAR(255),
PricePerNight DECIMAL(10,2),
NumberOfNights INT,
Price DECIMAL(10,2),
GuestTax DECIMAL(10,2),
TotalPrice DECIMAL(10,2),
VAT DECIMAL(10,2),
Income DECIMAL(10,2),
PaymentType VARCHAR(255)
FOREIGN KEY (GuestKey) REFERENCES d_guest(GuestKey),
FOREIGN KEY (HotelKey) REFERENCES d_hotel(HotelKey),
FOREIGN KEY (RoomKey) REFERENCES d_room(RoomKey),
FOREIGN KEY (CheckInDateKey) REFERENCES d_date(DateKey),
FOREIGN KEY (CheckOutDateKey) REFERENCES d_date(DateKey),
FOREIGN KEY (BookingOptionKey) REFERENCES
d_bookingOption(BookingOptionKey),
FOREIGN KEY (AgencyKey) REFERENCES d_agency(AgencyKey),
FOREIGN KEY (ReservationTypeKey) REFERENCES
d_reservationType(ReservationTypeKey)
);

```

Kod 15. Stvaranje činjenične tablice *f\_reservation*

Ovakva shema omogućit će agregaciju podataka po datumima, gostima, hotelima, tipovima rezervacija, agencijama te će korištenjem BI alata omogućiti kreiranje vizualizacija i izvještaja. U dimenzijskim tablicama kao primarni ključ koristi se surogatni ključ koji će se generirati prilikom učitavanja podataka u skladište.

Svi ključevi u skladištu podataka moraju biti sintetički surogatni ključevi. Ne smiju se koristiti izvorni ključevi [10]. Surogatni ključ se obično ne može interpretirati sam po sebi. Odnosno, ni na koji način nije pametni ključ. Također se naziva i umjetni ključ, cjelobrojni ključ i neprirodni ključ [7].

Ako se podaci u izvornoj bazi promijene, podaci u skladištu će ostati isti i stabilni. Surogatni ključevi su jedinstveni samo unutar skladišta podataka i nisu povezani sa poslovnim procesima i sustavima. Korištenjem surogatnih ključeva omogućena je bolja kontrola podataka, fleksibilnost u održavanju skladišta podataka te poboljšana učinkovitost upita. Omogućavaju i pregled povijesnih promjena, odnosno sporo mijenjajućih dimenzija (*Slowly Changing Dimensions* - SCD).

Jedinstveni ključ (*Unique key*) je identifikator iz izvornog sustava i povezan je sa poslovnim procesima, a ključan je tijekom ETL procesa. Kombinacijom surogatnih i jedinstvenih ključeva omogućena je bolja kontrola podataka, fleksibilnost u skladištu podataka i brže stvaranje analitičkih izvještaja.

## 7.2. ETL procesi

U nastavku su prikazani ETL procesi za učitavanje podataka iz izvorne baze u skladište podataka. Za svaku tablicu kreira se procedura koja će omogućiti navedeni proces. Pohranjene procedure (*Stored Procedures*) su unaprijed definirani skupovi SQL naredbi koji se pohranjuju u bazi podataka i mogu se ponovno koristiti. Omogućuju stvaranje SQL upita koji se pohranjuju i izvršavaju na poslužitelju. Također se mogu predmemorirati i ponovno koristiti. Glavna svrha pohranjenih procedura je sakriti izravne SQL upite iz koda i poboljšati performanse operacija baze podataka kao što su odabir, ažuriranje i brisanje podataka [11]. Postoje dvije vrste pohranjenih procedura: korisnički definirane pohranjene procedure i pohranjene procedure sustava. Korisnički definirane pohranjene procedure kombinacija su naredbi jezika za definiranje podataka (*Data Definition Language* - DDL) i jezika za manipulaciju podacima (*Data Manipulation Language* - DML), a procedure pohranjene u sustavu stvara i izvršava SQL Server za administrativne aktivnosti poslužitelja i preporuka je te procedure ne izmjenjivati.

Prva je procedura za učitavanje podataka o smještajnim jedinicama u dimenzijsku tablicu `d_hotel`. Podaci koji se učitavaju su `HotelID`, `Name`, `Address`, `ZIP Code`, `City`, `Country`, `Email`, `Stars`, `CheckInTime` i `CheckOutTime`. Korišteni `LEFT JOIN` omogućava da se unesu samo podaci koji već ne postoje u dimenzijskoj tablici (Kod 16).

```
CREATE PROCEDURE sp_LoadHotels
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO d_hotel (HotelID, Name, Address, ZIPCode,
    City, Country, Email, Stars, CheckInTime, CheckoutTime)
    SELECT DISTINCT
        h.HotelID,
        h.Name,
        h.Address,
        h.[ZIP Code],
        h.City,
        h.Country,
        h.Email,
        h.Stars,
        h.CheckinTime,
        h.CheckoutTime
    FROM HOTEL.dbo.Hotel h
    LEFT JOIN d_hotel d ON h.HotelID = d.HotelID
    WHERE d.HotelID IS NULL;

    PRINT 'Hotels Loaded Successfully';
END;
```

*Kod 16. Procedura za učitavanje podataka u d\_hotel tablicu*



Sljedeća procedura iz izvorne baze učitava podatke o gostima i omogućuje da se ne unose duplicirani podaci. Nakon unosa prikazuje se poruka o uspješnom unosu podataka (Kod 17).

```
CREATE PROCEDURE sp_LoadGuests
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO d_guest (GuestID, FirstName, LastName, Country, Gender, DateOfBirth)
    SELECT DISTINCT
        g.GuestID,
        g.FirstName,
        g.LastName,
        g.Country,
        g.Gender,
        g.DateOfBirth
    FROM HOTEL.dbo.Guest g
    LEFT JOIN d_guest d ON g.GuestID = d.GuestID
    WHERE d.GuestID IS NULL;

    PRINT 'Guests Loaded Successfully';
END;
```

*Kod 17. Procedura za učitavanje podataka u d\_guest tablicu*

Pohranjena procedura koja slijedi učitava podatke o sobama smještajnih jedinica, kao što su broj sobe, broj kata na kojem se soba nalazi, naziv i sadržaje sobe. Važno je pravilno odraditi spajanje tablica, u ovom slučaju sa tablicom RoomType iz relacijske baze podataka, što će omogućiti korištenje podataka i iz navedene tablice. Nakon uspješnog učitavanja podataka dobije se potvrдна poruka (Kod 18).

```
CREATE PROCEDURE sp_LoadRooms
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO d_room (RoomNumber, FloorNumber, RoomTypeName, Description, Amenities)
    SELECT
        r.RoomNumber,
        r.FloorNumber,
        rt.Name AS RoomTypeName,
        rt.Description,
        rt.Amenities
    FROM HOTEL.dbo.Room r
    JOIN HOTEL.dbo.RoomType rt ON r.TypeID = rt.TypeID
    LEFT JOIN d_room d ON r.RoomNumber = d.RoomNumber
    WHERE d.RoomNumber IS NULL;

    PRINT 'Rooms Loaded Successfully';
END;
```

*Kod 18. Procedura za učitavanje podataka u d\_room tablicu*

Procedura `sp_LoadReservationTypes` (Kod 19) osigurava učitavanje podataka o tipovima rezervacija. Pravilno definirano spajanje tablica omogućuje unos jedinstvenih zapisa koji već ne postoje u skladištu podataka.

```
CREATE PROCEDURE sp_LoadReservationTypes
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO d_reservationType (ReservationTypeID, ReservationTypeName, RTFullName, RTDescription)
    SELECT DISTINCT
        r.ReservationTypeID,
        r.ReservationTypeName,
        r.RTFullName,
        r.RTDescription
    FROM HOTEL.dbo.ReservationType r
    LEFT JOIN d_reservationType d ON r.ReservationTypeID = d.ReservationTypeID
    WHERE d.ReservationTypeID IS NULL;

    PRINT 'Reservation Types Loaded Successfully';
END;
```

*Kod 19. Procedura za učitavanje podataka u `d_reservationType` tablicu*

Budući da se u tablici `Agency` u izvorišnoj bazi podataka spremaju svi načini rezervacije, uključujući i turističke agencije i individualne rezervacije, korištenjem dimenzijske tablice `d_BookingOption` u skladištu podataka omogućit će se filtriranje podataka prema navedenim vrijednostima (Kod 20). Tablica `d_bookingOption` imat će vrijednosti naziva *'Travel Agency'* za rezervacije napravljene putem turističkih agencija, *'Individuals'* za individualne rezervacije, *'Hotel'* za rezervacije napravljene unutar organizacije i *'Other'* za sve ostale načine rezerviranja pohranjene u relacijskoj bazi podataka.

```
INSERT INTO d_bookingOption (BookingOptionName) VALUES ('Travel Agency');
INSERT INTO d_bookingOption (BookingOptionName) VALUES ('Individuals');
INSERT INTO d_bookingOption (BookingOptionName) VALUES ('Hotel');
INSERT INTO d_bookingOption (BookingOptionName) VALUES ('Other');
```

*Kod 20. Dodavanje vrijednosti u `d_bookingOption` tablicu*

Pohranjena procedura `sp_LoadAgencies` na taj će način spremati samo vrijednosti pravih turističkih agencija, uključujući ID agencije, naziv, adresu, državu i kontakt podatke (Kod 21).

```
CREATE PROCEDURE sp_LoadAgencies
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO d_agency (AgencyID, AgencyName, AgencyAddress,
    AgencyCountry, AgencyEmail, AgencyContact)
    SELECT DISTINCT
        a.AgencyID,
        a.AgencyName,
        a.AgencyAddress,
        a.AgencyCountry,
        a.AgencyEmail,
        a.AgencyContact
    FROM HOTEL.dbo.Agency a
    LEFT JOIN d_agency d ON a.AgencyName = d.AgencyName
    WHERE d.AgencyName IS NULL
    AND a.AgencyAddress IS NOT NULL
    AND a.AgencyCountry IS NOT NULL
    AND a.AgencyEmail IS NOT NULL
    AND a.AgencyContact IS NOT NULL
    AND a.AgencyName NOT IN ('INDIVIDUALCI', 'HOTEL LERO');

    PRINT 'Travel Agencies loaded successfully (without duplicates).';
END;
```

*Kod 21. Procedura za učitavanje podataka u d\_agency tablicu*

Procedura za popunjavanje dimenzijske tablice `d_date` generira podatke za tablicu datuma, uključujući informacije o danima, mjesecima, kvartalima, vikendima i praznicima. Početni dio definira raspon datuma koji će se unijeti u tablicu, a korištenjem navedene WHILE petlje osigurava se unos podataka za svaki dan u definiranom rasponu datuma. Budući da će se analizirati podaci u razdoblju od 2019. do 2023. godine, za početni i krajnji datum datumske tablice definirati će se godina prije i godina nakon navedenog vremenskog perioda (Kod 22).

```
CREATE PROCEDURE sp_Populate_d_date
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @start_date DATE = '2018-01-01';
    DECLARE @end_date DATE = '2024-12-31';

    -- Delete existing records to prevent duplicates
    DELETE FROM d_date WHERE FullDate BETWEEN @start_date AND @end_date;

    WHILE @start_date <= @end_date
    BEGIN
        INSERT INTO d_date (
            DateKey,
            FullDate,
            Day,
            DayOfWeek,
            DayOfWeekNumber,
            WeekOfYear,
            Month,
            MonthNumber,
            Quarter,
            Year,
            IsWeekend,
            IsHoliday,
            DayOfYear,
            Season
        )
```

*Kod 22. Procedura za popunjavanje d\_date tablice - 1*

Dio koji slijedi definira postavljanje ključa d\_date tablice u formatu 'YYYYMMDD', unos cijelog datuma, dana u tjednu, brojčanog dana u tjednu, mjeseca, broja mjeseca, kvartala, godine i indikatora vikenda (Kod 23).

```
SELECT
  CONVERT(INT, FORMAT(@start_date, 'yyyyMMdd')) AS DateKey, --YYYYMMDD format
  @start_date AS FullDate,
  DAY(@start_date) AS Day,
  DATENAME(WEEKDAY, @start_date) AS DayOfWeek,
  DATEPART(WEEKDAY, @start_date) AS DayOfWeekNumber,
  DATEPART(WEEK, @start_date) AS WeekOfYear,
  DATENAME(MONTH, @start_date) AS Month,
  MONTH(@start_date) AS MonthNumber,
  DATEPART(QUARTER, @start_date) AS Quarter,
  YEAR(@start_date) AS Year,
  CASE
    WHEN DATENAME(WEEKDAY, @start_date) IN ('Saturday', 'Sunday') THEN 1 -- Saturday & Sunday
    ELSE 0
  END AS IsWeekend,
```

*Kod 23. Procedura za popunjavanje d\_date tablice – 2*

Idući dio procedure odnosi se na unošenje logike za računanje blagdana i neradnih dana u Republici Hrvatskoj (Kod 24). Za definirani vremenski period unose se datumi fiksnih i pomičnih blagdana. Datumi će biti uneseni pojedinačno za sve godine, budući da je tijekom navedenog vremenskog perioda došlo do promjene datuma za pojedine blagdane. Dan državnosti se do 2020. godine obilježavao 25. lipnja, a nakon promjene 30. svibnja. Dan neovisnosti bio je 8. listopada, a od 2020. taj datum je postao Dan hrvatskog sabora i spomendan, te više nije neradni dan. Također se i Dan sjećanja na žrtve Domovinskog rata i Dan sjećanja na žrtvu Vukovara i Škabrnje, koji se obilježava 18. studenoga, od 2020. godine obilježava kao blagdan. U slučaju da nije bilo promjena definirali bi se samo datumi fiksnih blagdana koji bi se odnosili na sve godine, a ručno bi se unosili samo pomični blagdani.

```
CASE
  WHEN @start_date IN (
    -- Croatian holidays - fixed
    '2018-01-01', '2018-01-06', '2018-05-01', '2018-06-22', '2018-06-25', '2018-08-05',
    '2018-08-15', '2018-10-08', '2018-11-01', '2018-12-25', '2018-12-26',
    '2019-01-01', '2019-01-06', '2019-05-01', '2019-06-22', '2019-06-25', '2019-08-05',
    '2019-08-15', '2019-10-08', '2019-11-01', '2019-12-25', '2019-12-26',
    '2020-01-01', '2020-01-06', '2020-05-01', '2020-05-30', '2020-06-22', '2020-08-05',
    '2020-08-15', '2020-11-01', '2020-11-18', '2020-12-25', '2020-12-26',
    '2021-01-01', '2021-01-06', '2021-05-01', '2021-05-30', '2021-06-22', '2021-08-05',
    '2021-08-15', '2021-11-01', '2021-11-18', '2021-12-25', '2021-12-26',
    '2022-01-01', '2022-01-06', '2022-05-01', '2022-05-30', '2022-06-22', '2022-08-05',
    '2022-08-15', '2022-11-01', '2022-11-18', '2022-12-25', '2022-12-26',
    '2023-01-01', '2023-01-06', '2023-05-01', '2023-05-30', '2023-06-22', '2023-08-05',
    '2023-08-15', '2023-11-01', '2023-11-18', '2023-12-25', '2023-12-26',
    '2024-01-01', '2024-01-06', '2024-05-01', '2024-05-30', '2024-06-22', '2024-08-05',
    '2024-08-15', '2024-11-01', '2024-11-18', '2024-12-25', '2024-12-26'
  ) OR @start_date IN (
    -- Croatian holidays - movable (Easter, Easter Monday, Corpus Christi)
    '2018-04-01', '2019-04-21', '2020-04-12', '2021-04-04', '2022-04-17',
    '2023-04-09', '2024-03-31'
  ) OR @start_date IN (
    '2018-04-02', '2019-04-22', '2020-04-13', '2021-04-05', '2022-04-18',
    '2023-04-10', '2024-04-01'
  ) OR @start_date IN (
    '2018-05-31', '2019-06-20', '2020-06-11', '2021-06-03', '2022-06-16',
    '2023-06-08', '2024-05-30'
  ) THEN 1 ELSE 0
END AS IsHoliday,
```

*Kod 24. Procedura za popunjavanje d\_date tablice – 3*

Posljednji dio procedure koristi se za definiranje dana u godini i godišnjih doba na osnovu kalendarskih datuma godišnjih doba (Kod 25).

```

DATEPART(DAYOFYEAR, @start_date) AS DayOfYear,
CASE
    WHEN (MONTH(@start_date) = 3 AND DAY(@start_date) >= 20) OR (MONTH(@start_date)
    IN (4, 5)) OR (MONTH(@start_date) = 6 AND DAY(@start_date) <= 20) THEN 'Spring'
    WHEN (MONTH(@start_date) = 6 AND DAY(@start_date) >= 21) OR (MONTH(@start_date)
    IN (7, 8)) OR (MONTH(@start_date) = 9 AND DAY(@start_date) <= 22) THEN 'Summer'
    WHEN (MONTH(@start_date) = 9 AND DAY(@start_date) >= 23) OR (MONTH(@start_date)
    IN (10, 11)) OR (MONTH(@start_date) = 12 AND DAY(@start_date) <= 20) THEN 'Autumn'
    ELSE 'winter' -- Dec 21 - Mar 19
END AS Season;

SET @start_date = DATEADD(DAY, 1, @start_date);
END;
END;

```

*Kod 25. Procedura za popunjavanje d\_date tablice – 4*

Nakon svih dimenzijskih procedura, stvara se procedura za popunjavanje činjenične tablice. Činjenična tablica povezuje goste, sobe, smještajne jedinice, datume dolaska i odlaska, tipove rezervacija, cijene noćenja i prihode u jednu centraliziranu tablicu. Prvi dio procedure definira dodjeljivanje cijene noćenja u zapisima gdje su ti podaci uneseni samo za jednu osobu unutar rezervacije, a odnose se na sve osobe. Na taj način cijena se ravnomjerno raspoređi na sve osobe, u onim zapisima gdje je to potrebno (Kod 26).

```

CREATE PROCEDURE dbo.sp_LoadReservations
AS
BEGIN
    SET NOCOUNT ON;

    -- PricePerNight distribution for guests with 0.00 price
    WITH PriceDistribution AS (
        SELECT
            s.ReservationNumber,
            s.RoomNumber,
            s.GuestID,
            s.CheckInDate,
            s.CheckOutDate,
            s.AgencyID,
            s.ReservationTypeID,
            p.PaymentMethod AS PaymentType,

            -- Assign price per night: If price is 0.00, evenly distribute
            CASE
                WHEN s.PricePerNight = 0.00 THEN
                    (SELECT MAX(s2.PricePerNight)
                     FROM HOTEL.dbo.Reservation s2
                     WHERE s2.ReservationNumber = s.ReservationNumber
                     AND s2.RoomNumber = s.RoomNumber)
                    / NULLIF(
                        (SELECT COUNT(DISTINCT g.GuestID)
                         FROM HOTEL.dbo.Reservation g
                         WHERE g.ReservationNumber = s.ReservationNumber
                         AND g.RoomNumber = s.RoomNumber), 1)
                ELSE s.PricePerNight
            END AS PricePerNight

        FROM HOTEL.dbo.Reservation s
        LEFT JOIN HOTEL.dbo.Payment p ON s.ReservationNumber = p.ReservationNumber
    ),

```

*Kod 26. Procedura za učitavanje podataka u f\_reservation tablicu-1*

Sljedeći isječak koda prikazuje način računanja cijene, turističke takse, PDV-a, te prihoda. Price označava cijenu po noćenju pomnoženu sa brojem noćenja i ta informacija već sadrži PDV, koji će se kasnije oduzeti da bi se dobio prihod. Računanje turističke takse definirano je posebnom funkcijom, koja će biti prikazana u nastavku. TotalPrice označava ukupnu cijenu naplaćenu gostu, a sadrži i turističku taksu i PDV. Income označava cijenu zarade smještajne jedinice po noćenju bez turističke takse i PDV-a (Kod 27).

```
--Main Reservation Data
ReservationData AS (
    SELECT DISTINCT
        g.GuestKey,
        h.HotelKey,
        r.RoomKey,
        dci.DateKey AS CheckInDateKey,
        dco.DateKey AS CheckOutDateKey,
        bo.BookingOptionKey,
        a.AgencyKey,
        rt.ReservationTypeKey,
        s.ReservationNumber,
        s.PricePerNight,
        DATEDIFF(DAY, s.CheckInDate, s.CheckOutDate) AS NumberOfNights,

        -- Calculate Price
        (s.PricePerNight * DATEDIFF(DAY, s.CheckInDate, s.CheckOutDate)) AS Price,

        -- Call Function to Calculate Guest Tax
        dbo.CalculateGuestTax(g.GuestKey, dci.DateKey) AS GuestTax,

        -- Calculate Total Price = Price + GuestTax
        (s.PricePerNight * DATEDIFF(DAY, s.CheckInDate, s.CheckOutDate))
        + dbo.CalculateGuestTax(g.GuestKey, dci.DateKey) AS TotalPrice,

        -- VAT Calculation: (Price * 13) / 113
        ((s.PricePerNight * DATEDIFF(DAY, s.CheckInDate, s.CheckOutDate)) * 13) / 113 AS VAT,

        -- Income Calculation: Total Price - VAT - GuestTax
        (
            (s.PricePerNight * DATEDIFF(DAY, s.CheckInDate, s.CheckOutDate))
            + dbo.CalculateGuestTax(g.GuestKey, dci.DateKey)
        )
        - (((s.PricePerNight * DATEDIFF(DAY, s.CheckInDate, s.CheckOutDate)) * 13) / 113) -- VAT Calculation
        - dbo.CalculateGuestTax(g.GuestKey, dci.DateKey) AS Income,

        s.PaymentType
    )
```

*Kod 27. Procedura za učitavanje podataka u f\_reservation-2*

Definiranje JOIN izraza ključno je za povezivanje svih potrebnih tablica za ekstrakciju podataka. Također, d\_BookingOption sortira podatke na osnovu naziva agencije iz izvorišne baze podataka (Kod 28).

```

FROM PriceDistribution s
JOIN d_guest g ON s.GuestID = g.GuestID
JOIN d_room r ON s.RoomNumber = r.RoomNumber
JOIN HOTEL.dbo.Room sroom ON s.RoomNumber = sroom.RoomNumber
JOIN d_hotel h ON sroom.HotelID = h.HotelID
JOIN d_date dci ON s.CheckInDate = dci.FullDate
JOIN d_date dco ON s.CheckOutDate = dco.FullDate
LEFT JOIN d_agency a ON s.AgencyID = a.AgencyID
LEFT JOIN HOTEL.dbo.Agency sa ON s.AgencyID = sa.AgencyID
JOIN d_reservationType rt ON s.ReservationTypeID = rt.ReservationTypeID
LEFT JOIN d_bookingOption bo
    ON bo.BookingOptionName =
    CASE
        WHEN sa.AgencyName = 'INDIVIDUALCI' THEN 'Individuals'
        WHEN sa.AgencyName = 'HOTEL LERO' THEN 'Hotel'
        WHEN a.AgencyKey IS NOT NULL THEN 'Travel Agency'
        ELSE 'Other'
    END
)

```

*Kod 28. Procedura za učitavanje podataka u f\_reservation tablicu-3*

Nakon svih potrebnih transformacija, pripremljeni podaci se učitavaju u činjeničnu tablicu (Kod 29). Ovakav način omogućuje čisto i strukturirano umetanje podataka.

```

--Insert Data into Fact Table
INSERT INTO f_reservation (
    GuestKey, HotelKey, RoomKey, CheckInDateKey, CheckOutDateKey,
    BookingOptionKey, AgencyKey, ReservationTypeKey, ReservationNumber,
    PricePerNight, NumberOfNights, Price, GuestTax, TotalPrice, VAT, Income, PaymentType
)
SELECT DISTINCT
    GuestKey, HotelKey, RoomKey, CheckInDateKey, CheckOutDateKey,
    BookingOptionKey, AgencyKey, ReservationTypeKey, ReservationNumber,
    PricePerNight, NumberOfNights, Price, GuestTax, TotalPrice, VAT, Income, PaymentType
FROM ReservationData;

PRINT 'Reservations Loaded Successfully!';
END;

```

*Kod 29. Procedura za učitavanje podataka u f\_reservation tablicu-4*

Za izvršavanje svih procedura potrebno je pokrenuti naredbu EXEC i ime procedure, pazeći da se uvijek prvo pokrenu procedure za dimenzijske tablice, a nakon njih za činjeničnu tablicu (Kod 30).

```

EXEC dbo.sp_LoadHotels;

EXEC dbo.sp_LoadGuests;

EXEC dbo.sp_LoadRooms;

EXEC dbo.sp_LoadReservationTypes;

EXEC dbo.sp_LoadAgencies;

EXEC dbo.sp_Populate_d_date;

EXEC dbo.sp_LoadReservations;

```

*Kod 30. Naredbe za izvršavanje pohranjenih procedura*

Kod 31 prikazuje funkciju koja računa iznos turističke takse, na osnovu godina gosta tijekom boravka u smještajnoj jedinici. Prvo se računa koliko gost ima godina prilikom prijave u smještajnu jedinicu.

```
CREATE FUNCTION dbo.CalculateGuestTax
(
    @GuestKey INT,
    @CheckInDateKey INT
)
RETURNS DECIMAL(10,2)
AS
BEGIN
    DECLARE @DateOfBirth DATE;
    DECLARE @CheckInDate DATE;
    DECLARE @GuestAge INT;
    DECLARE @GuestTaxMultiplier DECIMAL(3,2);
    DECLARE @TaxRate DECIMAL(10,2);

    -- Get Date of Birth of the guest
    SELECT @DateOfBirth = g.DateOfBirth
    FROM d_guest g
    WHERE g.GuestKey = @GuestKey;

    -- Get Check-In Date from d_date table using DateKey
    SELECT @CheckInDate = d.FullDate
    FROM d_date d
    WHERE d.DateKey = @CheckInDateKey;

    -- Calculate Guest Age at the time of Check-In
    SET @GuestAge = DATEDIFF(YEAR, @DateOfBirth, @CheckInDate)
        - CASE WHEN FORMAT(@DateOfBirth, 'MMdd') > FORMAT(@CheckInDate, 'MMdd') THEN 1 ELSE 0 END;
```

*Kod 31. Funkcija za računanje turističke takse - 1*

Gosti ispod 12 godina ne plaćaju turističku taksu, gosti od 12 do 18 plaćaju 50% iznosa turističke takse, a odrasli plaćaju punu cijenu iznosa turističke takse. Iznosi taksi dodatno su definirani tijekom godina i mjesečnih razdoblja (Kod 32).



```

-- Determine Guest Tax Multiplier Based on Age
SET @GuestTaxMultiplier =
CASE
    WHEN @GuestAge < 12 THEN 0          -- Under 12, no tax
    WHEN @GuestAge BETWEEN 12 AND 17 THEN 0.5 -- 50% tax for 12-17
    ELSE 1                               -- Full tax for 18 and above
END;

-- Determine Tax Rate Based on Check-In Year and Season
SET @TaxRate =
CASE
    WHEN YEAR(@CheckInDate) = 2019 THEN 1.20
    WHEN YEAR(@CheckInDate) = 2020 THEN 1.40
    WHEN YEAR(@CheckInDate) = 2021 THEN 1.40
    WHEN YEAR(@CheckInDate) = 2022 AND (MONTH(@CheckInDate) BETWEEN 1 AND 3
    OR MONTH(@CheckInDate) BETWEEN 10 AND 12) THEN 1.40
    WHEN YEAR(@CheckInDate) = 2022 AND (MONTH(@CheckInDate) BETWEEN 4 AND 9) THEN 2.70
    WHEN YEAR(@CheckInDate) = 2023 AND (MONTH(@CheckInDate) BETWEEN 1 AND 3
    OR MONTH(@CheckInDate) BETWEEN 10 AND 12) THEN 1.90
    WHEN YEAR(@CheckInDate) = 2023 AND (MONTH(@CheckInDate) BETWEEN 4 AND 9) THEN 2.70
    ELSE 0
END;

-- Calculate Final Guest Tax
RETURN @GuestTaxMultiplier * @TaxRate;
END;
GO

```

*Kod 32. Funkcija za računanje turističke takse*

Ove pohranjene procedure osiguravaju efikasan ETL proces i omogućuju automatizaciju unosa podataka u skladište, eliminaciju duplikata te pripremu podataka za analize korištenjem BI alata.

## 8. POWER BI

Alati poslovne inteligencije igraju ključnu ulogu u analizi podataka i donošenju odluka. Omogućuju prikupljanje, obradu i vizualizaciju podataka iz različitih izvora, čime pomažu u prepoznavanju trendova, unapređenju strategija i optimizaciji poslovnih procesa unutar organizacija. Među najpoznatijim BI alatima su Power BI, Tableau, Qlik Sense, Google Data Studio i Looker Studio.

Power BI predstavlja skup softverskih usluga, aplikacija i konektora koji zajedno omogućuju pretvaranje nepovezanih izvora podataka u koherentne, vizualno bogate i interaktivne uvide. Ovaj alat omogućuje jednostavno povezivanje s različitim izvorima podataka, stvaranje vizualizacija i analiza i dijeljenje s drugim korisnicima [12].

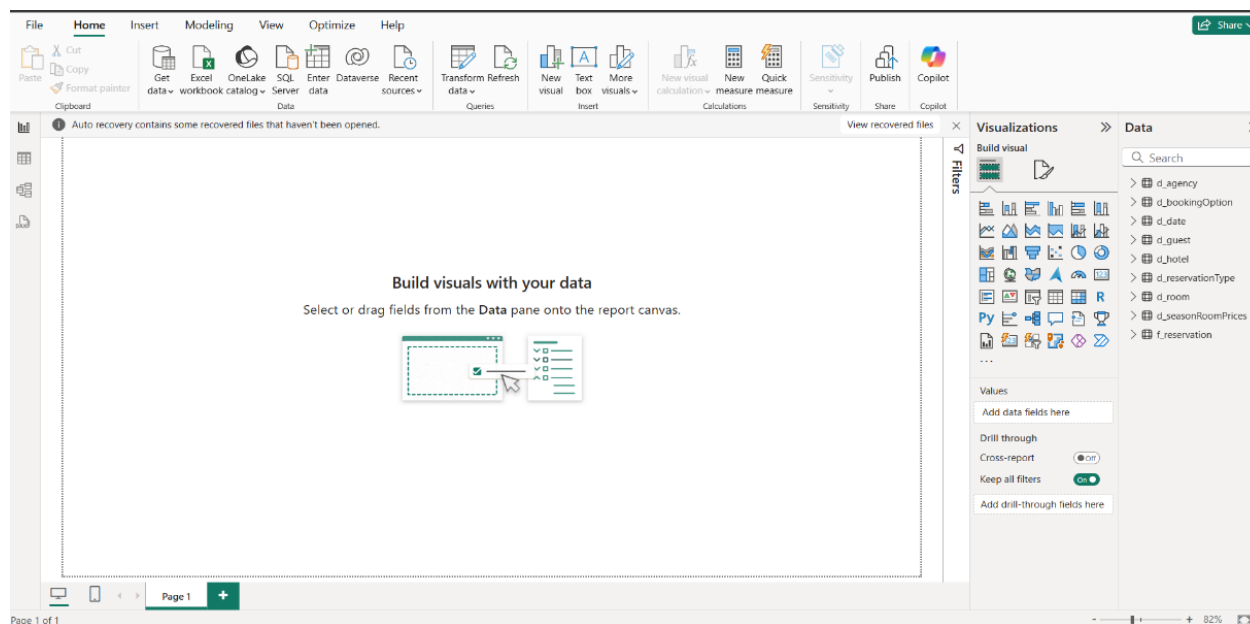
Tri osnovne komponente su Power BI Desktop, Power BI Service i Power BI Mobile. Power BI Desktop je Windows aplikacija koja služi za izradu izvještaja i analiza. Omogućuje povezivanje podataka iz različitih izvora, kao što su Excel, SQL Server, Google Analytics i brojni drugi. Korištenjem Power Query Editora omogućeno je čišćenje i transformacija podataka. DAX (*Data Analysis Expressions*) formule omogućuju izradu prilagođenih mjera i izračuna. Ugrađeni alati za vizualizaciju omogućuju stvaranje grafova, tabela, karata i brojnih drugih vizualnih prikaza. Mogu se dodavati filteri, sliceri i ostali interaktivni elementi kako bi se poboljšale analize podataka. Power BI Service je online usluga koja omogućuje objavu, dijeljenje i surađivanje na izvještajima unutar organizacije. Stvoreni izvještaji mogu se objaviti i dijeliti sa drugim korisnicima. Power BI Service može automatski dohvaćati nove podatke iz povezanih izvora. Omogućena je i integracija sa drugim alatima unutar Microsoft 365 ekosustava. Power BI Mobile je mobilna aplikacija za Windows, iOS i Android uređaje, a omogućuje pregled i analizu izvještaja u pokretu. Omogućen je interaktivni prikaz podataka direktno s mobilnih uređaja. Power BI Mobile može slati obavijesti kada se promijene ključni pokazatelji ili kad izvještaji dosegnu određene vrijednosti. Omogućen je i offline pristup podacima, a integracija s *Azure Active Directory* (AAD) pruža siguran pristup podacima.

Unutar organizacija, Power BI Desktop će koristiti analitičari za izradu izvještaja i analize podataka. Menadžeri i poslovni korisnici koristit će Power BI Service za pregled izvještaja i donošenje odluka, a mobilni korisnici, kao npr. prodajni timovi, koristit će Power BI Mobile za pregled ključnih pokazatelja u pokretu. Kombinacijom svih komponenti, Power BI omogućuje donošenje boljih odluka na temelju podataka i poboljšanje poslovnih procesa unutar organizacija.

### 8.1. Učitavanje podataka – povezivanje sa skladištem podataka

Power BI nudi mogućnosti povezivanja sa raznim izvorima podataka, kao što su Excel datoteke, baze podataka, tekstualne datoteke, web, CSV datoteke i brojne druge. U ovom slučaju koristit će se SQL Server i spojiti sa skladištem podataka. Power BI Desktop sučelje (Slika 19) sastoji se od trake s alatima (*Ribbon*), s desne strane nalazi se okno s podacima (*Data Pane*), okno za vizualizacije (*Visualizations Pane*) i okno za filtere (*Filters Pane*), a s lijeve strane nalaze se opcije izbora pogleda: *Report View*, *Table View*, *Model View* i *DAX query view*. U središtu je područje za izradu izvještaja (*Report Canvas*), glavni prostor u kojemu se dodaju i uređuju vizualizacije, te prilagođava izgled izvještaja. *Report View* predstavlja pogled prilikom stvaranja vizualizacija, *Table View* daje tablični prikaz podataka, *Model View*

omogućava vizualni prikaz tablica i definiranje veza među podacima, a DAX Query View služi za obradu i transformaciju podataka, omogućava čišćenje, filtriranje, spajanja podataka te dodavanje novih izračuna.



Slika 19. Izgled Power BI Desktop sučelja

U nastavku će biti prikazano stvaranje vizualizacija na osnovu učitanih podataka.

## 8.2. Izvještaj 1 – Pregled poslovanja smještajnih jedinica

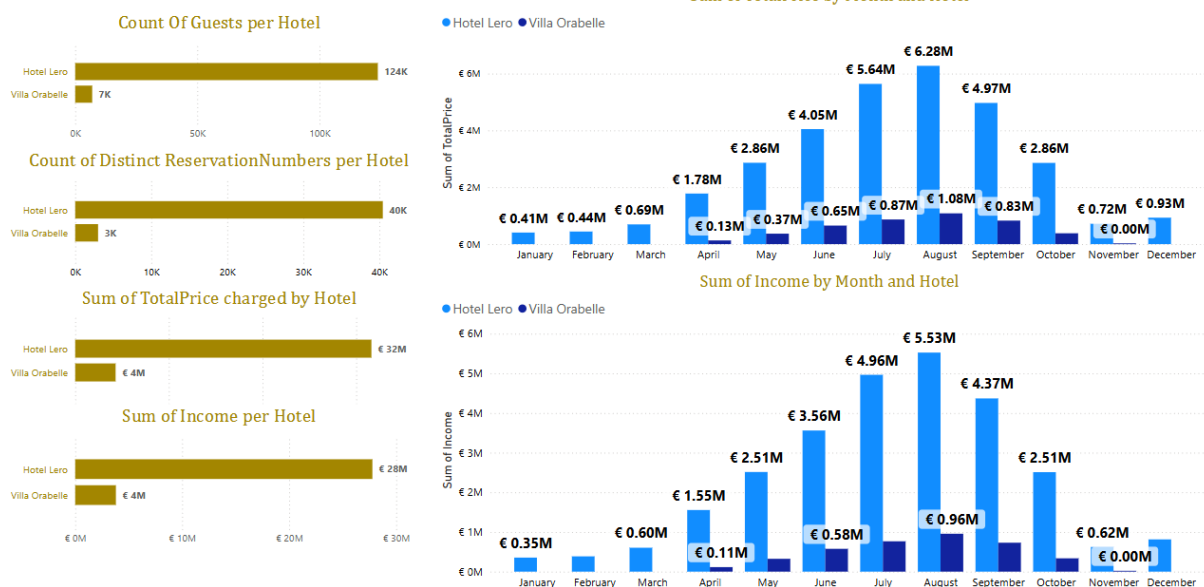
Podaci pohranjeni u skladištu podataka odnose se na dvije smještajne jedinice, Hotel Lero i Villu Orabelle. Hotel Lero posluje od 1972. godine, otvoren je kroz cijelu godinu i nudi 205 soba i 5 tipova soba. Villa Orabelle nalazi se u blizini hotela Lero, svoja vrata otvorila je 2019. godine, a nudi 19 soba i 9 tipova soba. Tijekom godine otvorena je u razdoblju od travnja do listopada.

U prvoj vizualizaciji (Slika 20) prikazana je usporedba poslovanja Hotela Lero i Ville Orabelle. Horizontalni stupčasti grafovi prikazuju broj gostiju po smještajnoj jedinici, broj rezervacija, ukupnu naplaćenu cijenu i ukupan prihod. Grupirani stupčasti grafovi prikazuju sumu ukupnih cijena i sumu prihoda po mjesecima i smještajnim jedinicama. *Year slicer* dodatno omogućuje pregled podataka po određenim godinama. Hotel Lero ostvaruje veće prihode, ima više gostiju i rezervacija, što je i očekivano budući da je hotel otvoren tijekom cijele godine i ima značajno veći broj soba u odnosu na Villu Orabelle. Najveći prihodi ostvaruju se tijekom ljetnih mjeseci, u srpnju i kolovozu, što odražava i sezonalnost poslovanja.

## GENERAL REPORT OF ACCOMMODATION UNITS

Select Year:

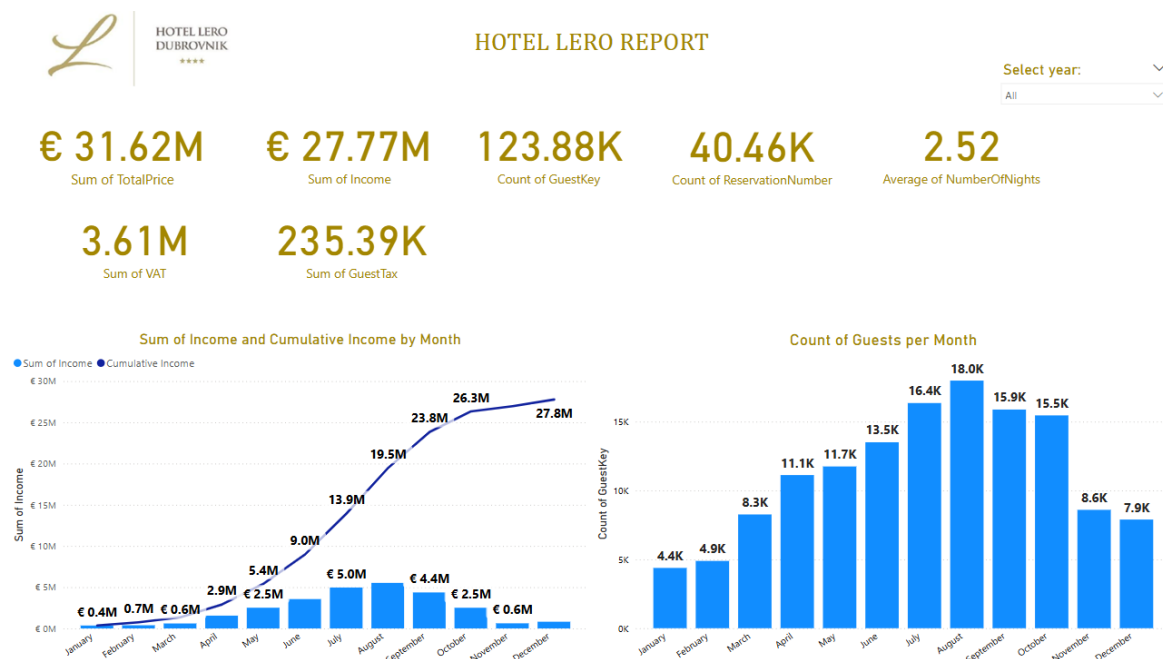
All



Slika 20. Pregled poslovanja smještajnih jedinica

### 8.3. Izvještaj 2 – Pregled poslovanja Hotela Lero

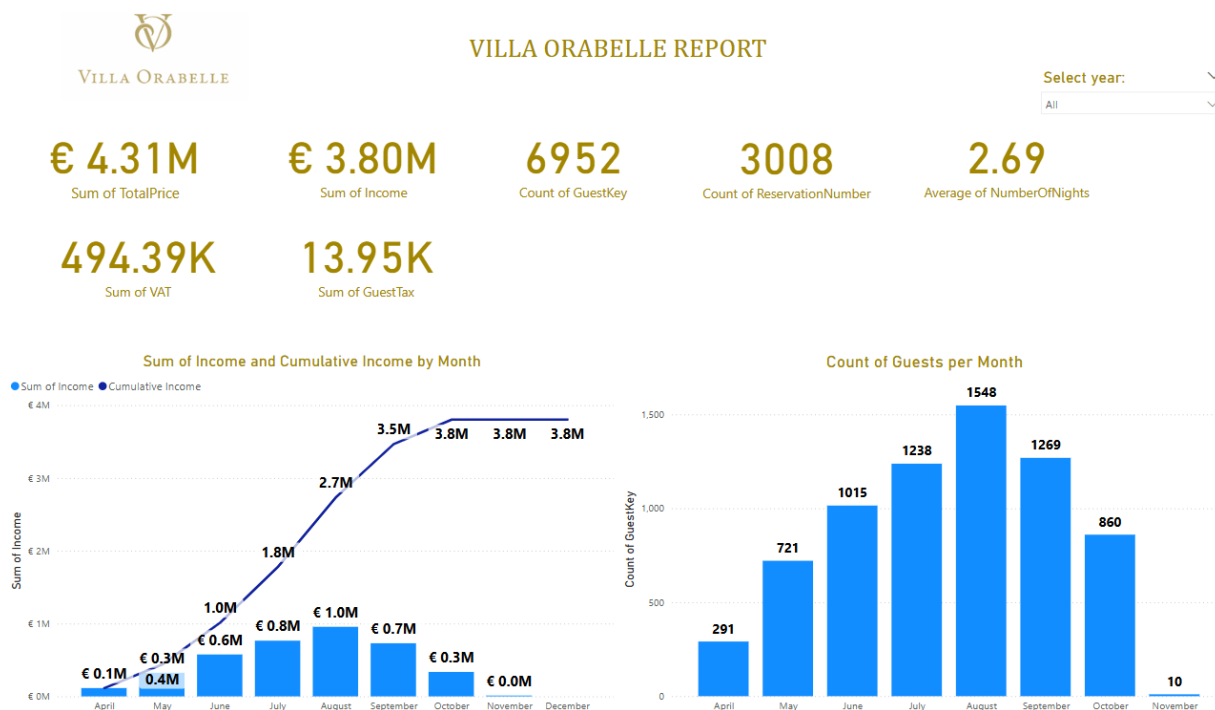
Drugi izvještaj daje uvid u poslovanje jedne smještajne jedinice, hotela Lero. Karticama su predstavljeni ključni financijski parametri koji pružaju detaljan uvid u poslovanje: suma ukupno naplaćenih rezervacija, suma prihoda, broj gostiju koji su boravili u hotelu, broj jedinstvenih rezervacija, prosječno trajanje boravka, sume PDV-a i turističke takse. Grafovima su također prikazane sume prihoda i kumulativni prihod po mjesecima, te kretanje broja gostiju po mjesecima. I u ovom slučaju može se koristiti *Year slicer* za dodatan pregled svih podataka po godinama.



Slika 21. Pregled poslovanja hotela Lero

## 8.4. Izvještaj 3 – Pregled poslovanja Ville Orabelle

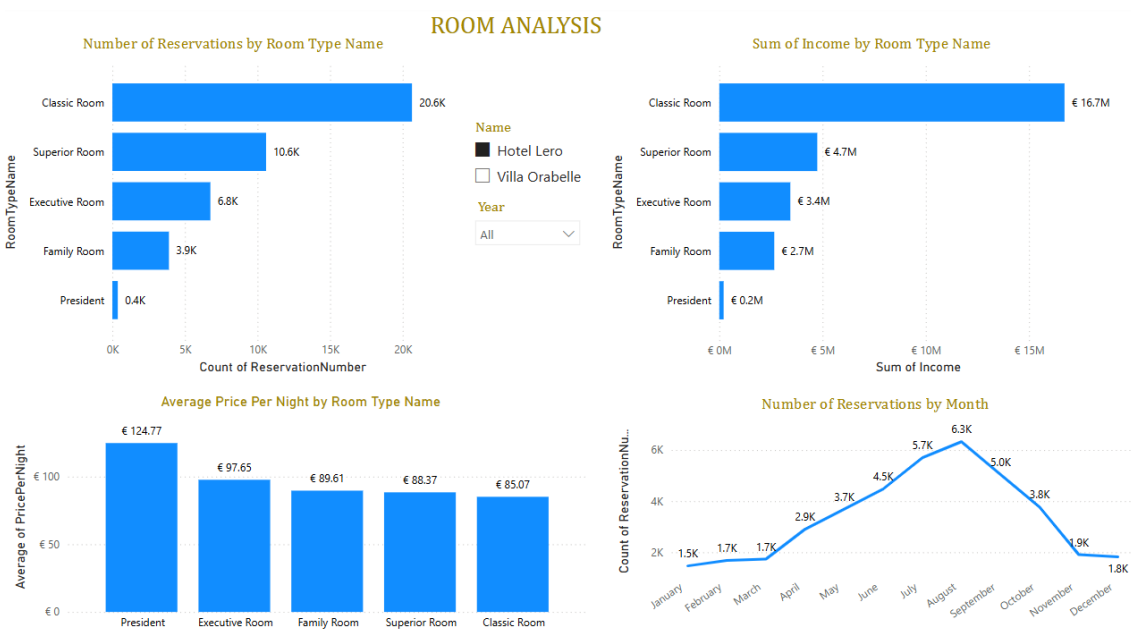
Ovaj izvještaj daje iste uvide kao i prethodni, samo za poslovanje smještajne jedinice Ville Orabelle.



Slika 22. Pregled poslovanja Ville Orabelle

## 8.5. Izvještaj 4 – Analiza soba hotela Lero

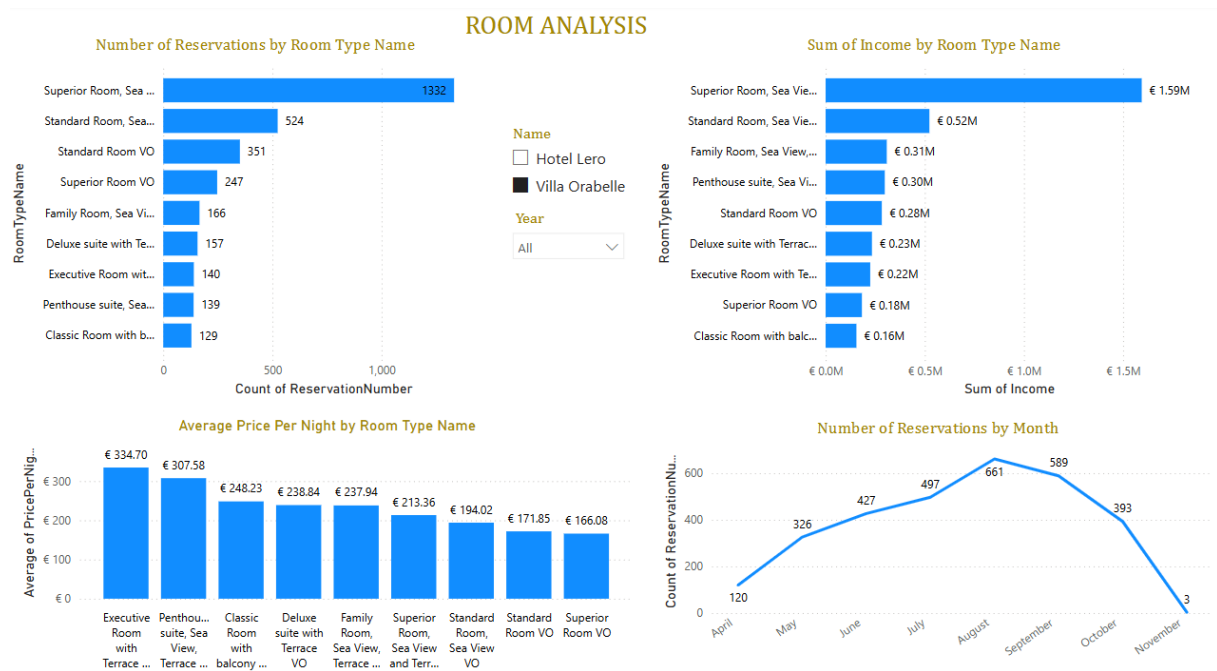
U ovom izvještaju prikazane su vrste soba hotela Lero. Horizontalni stupčasti grafovi omogućuju uvid u broj rezervacija prema tipu sobe i sumu prihoda, dok vertikalni stupčasti graf prikazuje prosječnu cijenu noćenja, a linijski graf omogućava pregled broja rezervacija kroz mjesec. Također je omogućeno filtriranje po godinama.



Slika 23. Analiza soba hotela Lero

## 8.6. Izvještaj 5 – Analiza soba Ville Orabelle

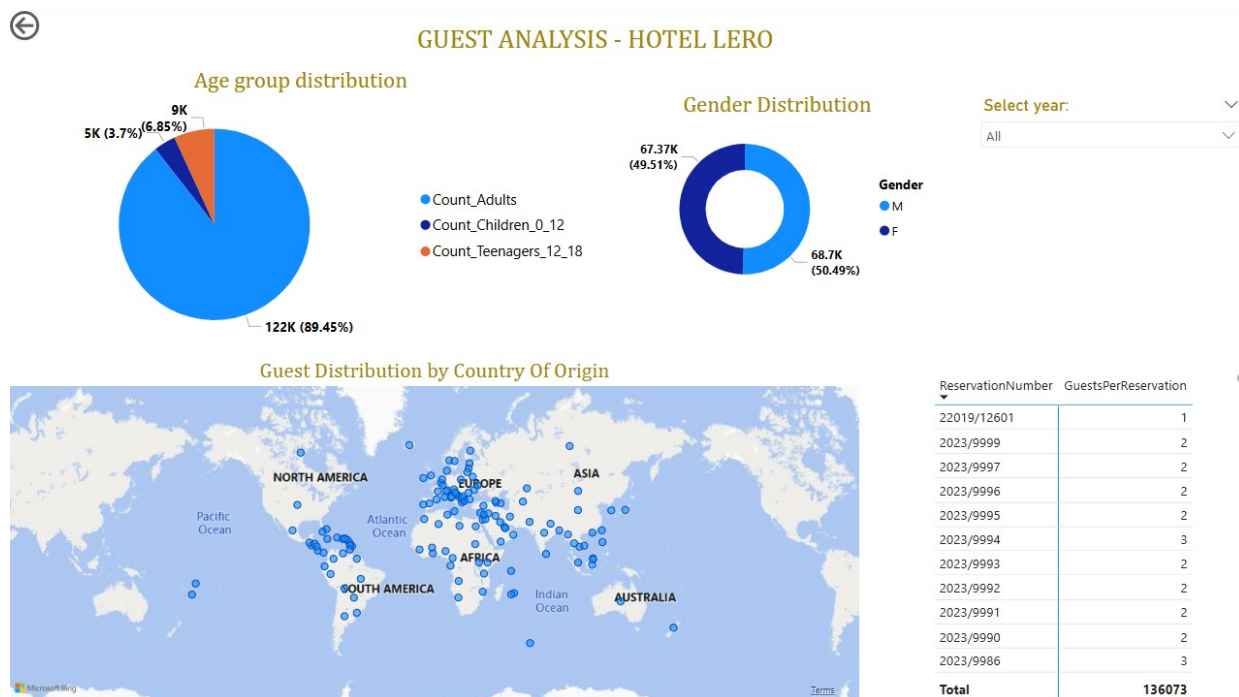
Ovaj izvještaj pruža uvide u analizu soba Ville Orabelle, a organizacija prikaza je kao u prethodnom izvještaju.



Slika 24. Analiza soba Ville Orabelle

## 8.7. Izvještaj 6 – Analiza gostiju hotela Lero

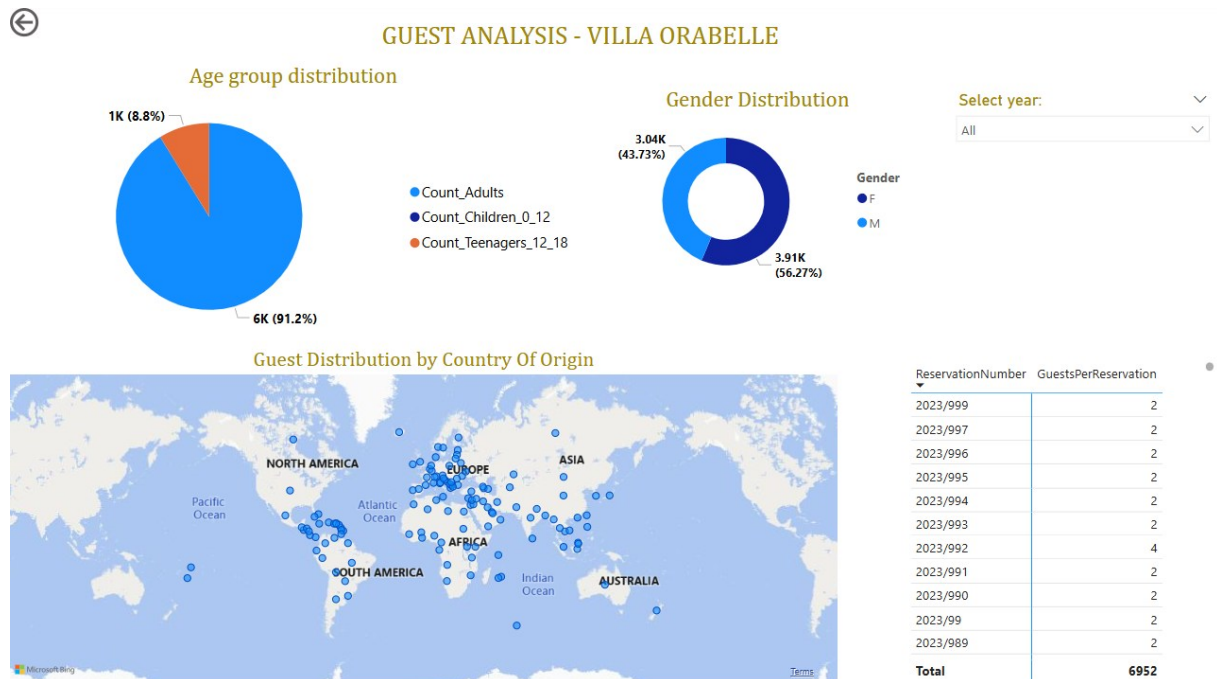
U ovom izvještaju prikazana je analiza gostiju koji su boravili u hotelu Lero. Prikazana je distribucija gostiju po dobnim skupinama, distribucija po spolu, ukupan broj gostiju po rezervaciji i distribucija gostiju prema državi iz koje dolaze.



Slika 25. Analiza gostiju hotela Lero

## 8.8. Izvještaj 7 – Analiza gostiju Ville Orabelle

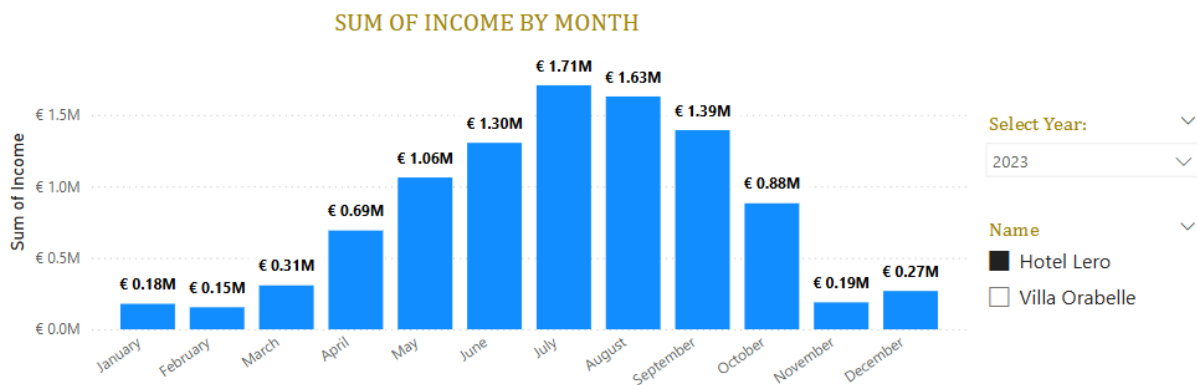
Isto kao i prethodni izvještaj, samo za goste Ville Orabelle.



Slika 26. Analiza gostiju Ville Orabelle

## 8.9. Izvještaj 8 – Analiza prihoda po mjesecima

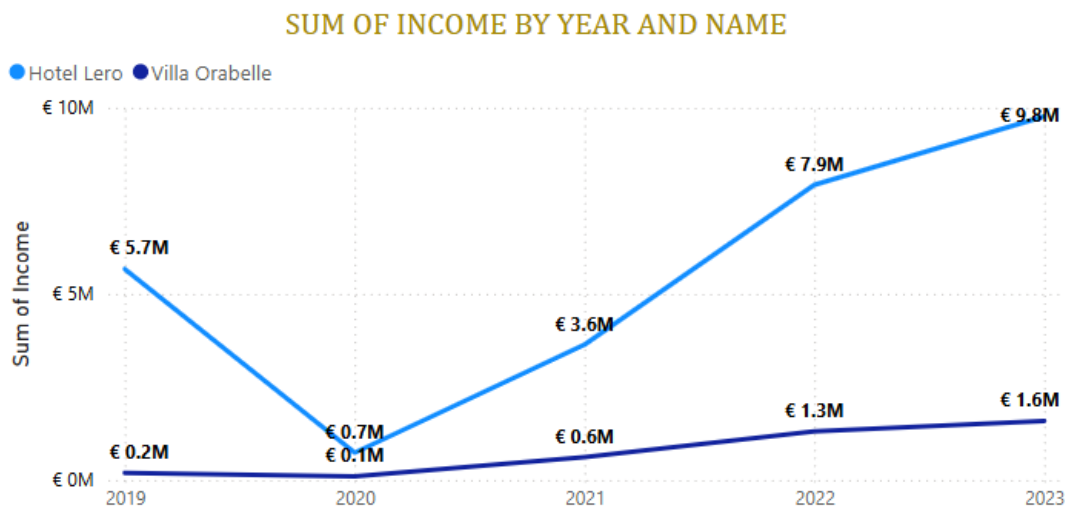
Ovaj izvještaj prikazuje sumu prihoda po mjesecima za hotel Lero u 2023. godini. Opcije filtriranja omogućuju odabir bilo koje godine ili cijelog raspona godina i odabir smještajne jedinice koju se želi analizirati. Hotel Lero je tijekom 2023. godine najviše prihoda ostvario tijekom srpnja i kolovoza.



Slika 27. Analiza prihoda po mjesecima

## 8.10. Izvještaj 9 – Analiza prihoda po godinama i smještajnim jedinicama

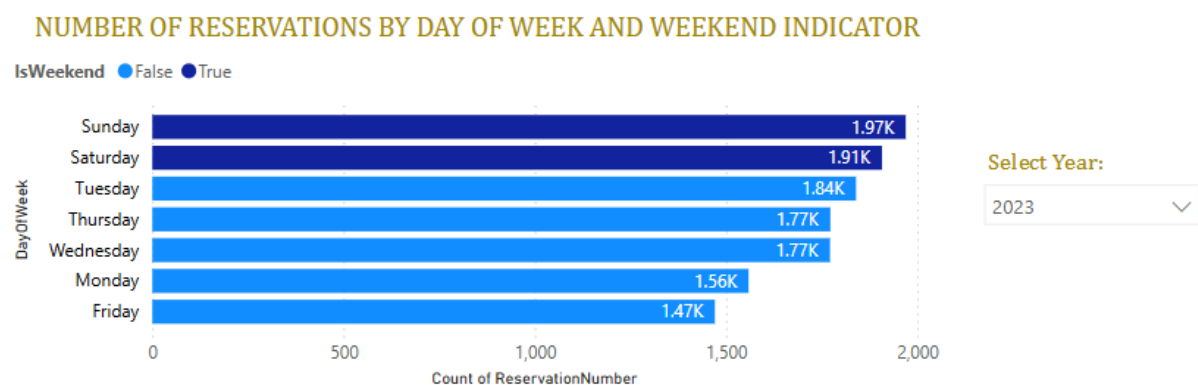
Linijski graf prikazuje rast i pad prihoda u vremenskom razdoblju od 2019. godine do 2023. godine za hotel Lero i Villu Orabelle. Suma prihoda prikazana je na y-osi u milijunima eura, a x-osa podijeljena je po godinama. Vidljiv je značajan pad prihoda tijekom 2020. godine koji je uzrokovala pojava koronavirusa, koji je obilježio globalnu krizu u turizmu. Nakon toga iz godine u godinu zabilježen je rast prihoda za obje smještajne jedinice.



Slika 28. Analiza prihoda po godinama i smještajnim jedinicama

## 8.11. Izvještaj 10 – Usporedba broja rezervacija tijekom vikenda i radnih dana

Horizontalni trakasti graf prikazuje broj rezervacija po danima u tjednu za 2023. godinu. Podaci su kategorizirani prema vikendu i radnim danima. Vidljivo je da je u 2023. godini najveći broj rezervacija ostvaren tijekom vikenda, dok petak ima najmanji broj rezervacija. Rezervacije tijekom tjedna su stabilne, ali ipak manje nego vikendom.



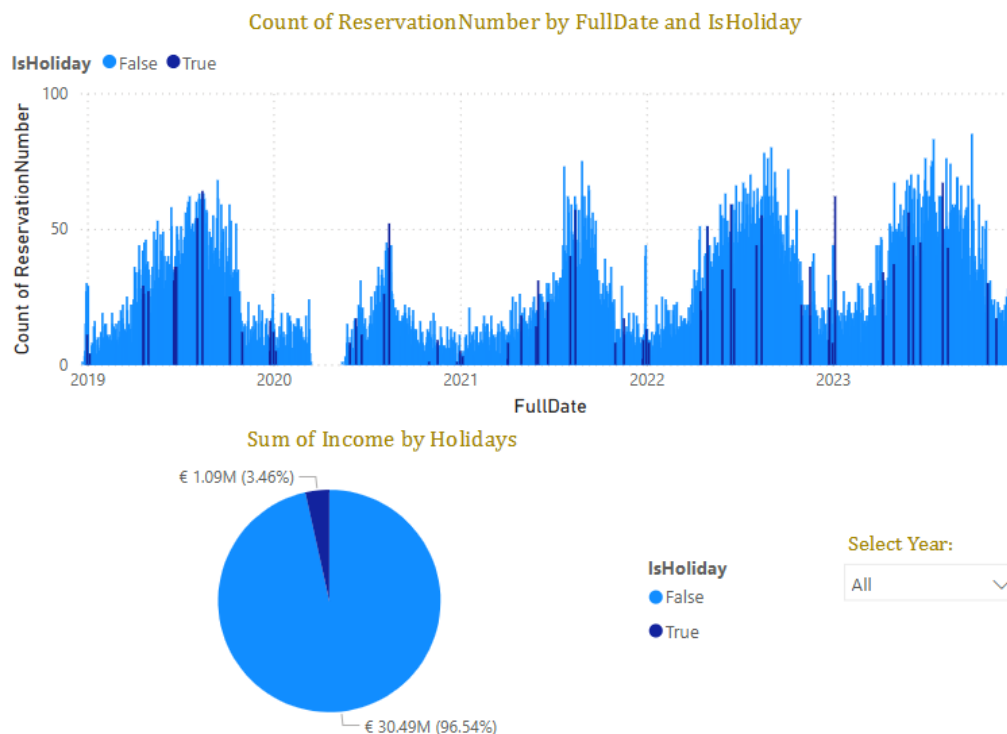
Slika 29. Broj rezervacija tijekom vikenda i radnih dana



## 8.12. Izvještaj 11 – Broj rezervacija i suma prihoda u odnosu na blagdane

Stupčasti graf prikazuje broj rezervacija kroz vrijeme u odnosu na blagdane. Tortni graf (*Pie Chart*) prikazuje sumu prihoda ostvarenih tijekom blagdana u usporedbi s redovnim danima. Iako blagdani generiraju značajan broj rezervacija, ukupni prihodi ostvareni u te dane su relativno niski, te je vidljivo da su redovni dani ključni za stabilnost jer čine većinu prihoda.

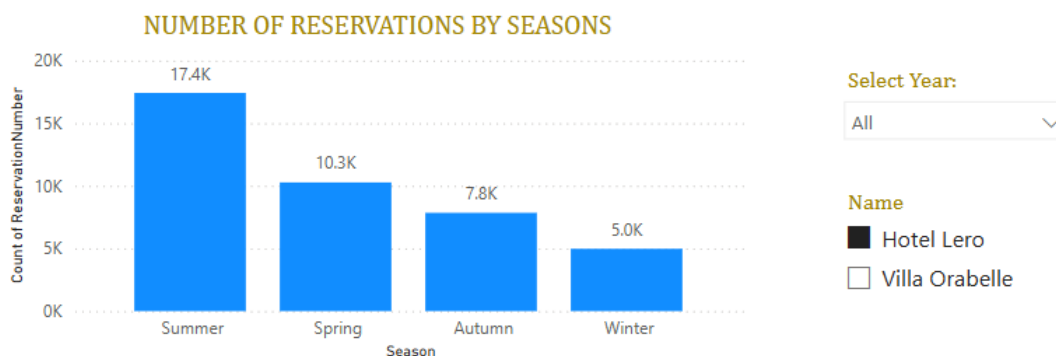
### NUMBER OF RESERVATIONS AND INCOME BY HOLIDAY INDICATOR



Slika 30. Broj rezervacija i prihodi tijekom blagdana

## 8.13. Izvještaj 12 – Broj rezervacija tijekom sezona u godini

Vizualizacija prikazuje broj rezervacija tijekom svih godišnjih doba. Vidljivo je da za hotel Lero ljetna sezona dominira po broju rezervacija, što daje zaključiti da je upravo ljetna sezona glavni turistički period. Proljeće i jesen također bilježe značajan broj rezervacija, dok je zimi taj broj najmanji.

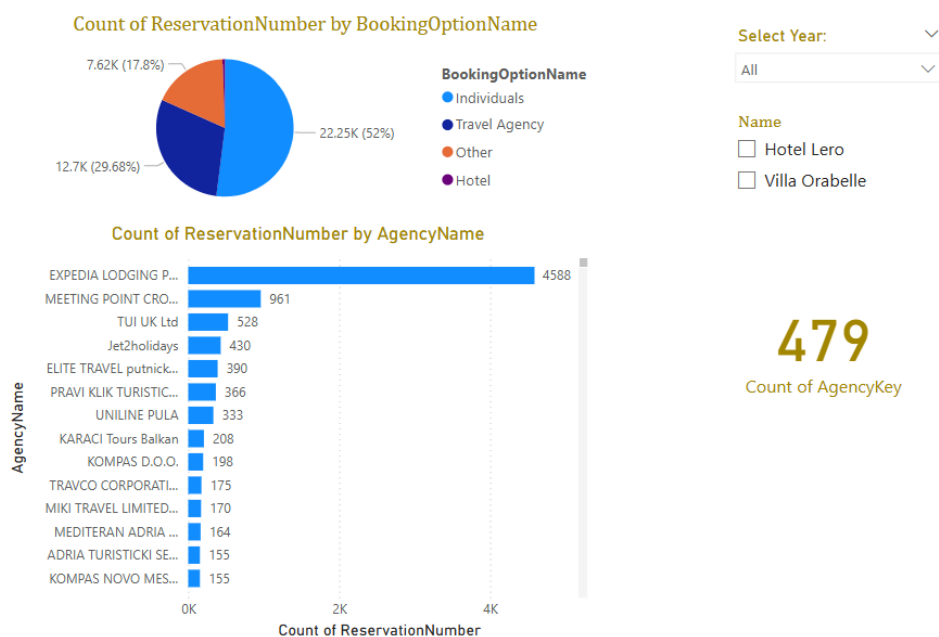


Slika 31. Broj rezervacija tijekom sezona u godini

## 8.14. Izvještaj 13 – Analiza opcija rezervacije i turističkih agencija

U ovom izvještaju prikazana je distribucija rezervacija prema načinu rezerviranja, odnosno jesu li to individualci, turističke agencije ili nešto treće. Prikazan je broj rezervacija prema agencijama te ukupan broj agencija, a dodatno filtriranje omogućuje detaljniji prikaz podataka po godinama i smještajnim jedinicama.

### AGENCY AND BOOKING OPTION ANALYSIS



Slika 32. Analiza agencija i opcija rezervacije

Vizualizacije stvorene korištenjem podataka prikazuju uvide koji su otkriveni u podacima. Power BI izvještaji mogu imati jednu stranicu s jednim vizualnim elementom ili više stranica punih vizualnih elemenata. Power BI Service omogućuje da se pojedinačna izvješća kombiniraju u nadzorne ploče (*Dashboard*).

Ove vizualizacije daju ključne uvide u poslovanja navedenih smještajnih jedinica i pomažu pri donošenju odluka i planiranju poslovnih strategija. Pomažu u razumijevanju kada je najveća potražnja za smještajem, daju uvide kako blagdani i godišnja doba utječu na rezervacije i prihode, koje agencije generiraju najviše putnika i prihoda, što može biti korisno za optimizaciju cijena i marketinških strategija.

## 8.15. DAX (*Data Analysis Expressions*) funkcije

DAX funkcije omogućuju izvođenje složenih izračuna i analiza podataka, stvaranjem izračunatih stupaca (*Calculated Columns*) i mjera (*Measures*).

Izračunati stupci su novi stupci koji se stvaraju u tablici korištenjem DAX upita. Koriste se kada je potreban novi podatak za filtriranje, sortiranje ili grupiranje i ostaju pohranjeni u modelu.

Mjere (*Measures*) su kalkulacije koje se dodaju samo kada su potrebne, na osnovu konteksta izvještaja. Računaju se dinamički i ne zauzimaju dodatan prostor kao izračunati stupci.

Primjer stvaranja mjere može biti stvaranje kumulativne sume prihoda. Funkcija (Kod 33) računa kumulativnu sumu prihoda i prikazuje trend kumulativne zarade. Mjera se dinamički učitava ovisno o kontekstu i filterima u izvještaju.

```

1 Cumulative Income =
2 CALCULATE(
3     SUM(f_reservation[Income]),
4     FILTER(
5         ALLSELECTED(d_date),
6         d_date[MonthNumber] <= MAX(d_date[MonthNumber])
7     )
8 )

```

*Kod 33. Računanje kumulativne sume prihoda*

Drugi primjer je funkcija koja računa dob gosta u trenutku prijave u hotel, koristeći datum rođenja i datum prijave.

```

1 GuestAgeAtStay =
2 VAR GuestDOB =
3     LOOKUPVALUE(d_guest[DateOfBirth], d_guest[GuestKey], SELECTEDVALUE(f_reservation[GuestKey]))
4
5 VAR CheckInDate =
6     DATE(
7         LEFT(SELECTEDVALUE(f_reservation[CheckInDateKey]), 4),
8         MID(SELECTEDVALUE(f_reservation[CheckInDateKey]), 5, 2),
9         RIGHT(SELECTEDVALUE(f_reservation[CheckInDateKey]), 2)
10    )
11
12 RETURN
13     DATEDIFF(GuestDOB, CheckInDate, YEAR)
14

```

*Kod 34. Funkcija za izračun godina gosta*

Ova funkcija dalje omogućuje izračun broja gostiju podijeljen po navedenim dobnim skupinama (Kod 35, Kod 36, Kod 37) što je omogućilo analize dobnih skupina u izvještajima.

```

Count_Adults = CALCULATE(COUNT(f_reservation[GuestKey]),
| FILTER(f_reservation, [GuestAgeAtStay] >= 18))

```

*Kod 35. Računanje broja odraslih*

```

Count_Teenagers_12_18 = CALCULATE(COUNT(f_reservation[GuestKey]),
| FILTER(f_reservation, [GuestAgeAtStay] >= 12
| && [GuestAgeAtStay] < 18))

```

*Kod 36. Računanje broja mladih od 12 do 18 godina*

```

Count_Children_0_12 = CALCULATE(COUNT(f_reservation[GuestKey]), FILTER(f_reservation,
| [GuestAgeAtStay] < 12))

```

*Kod 37. Računanje broja djece do 12 godina*

DAX funkcije poboljšavaju analitičke mogućnosti modela podataka i omogućuju dublju analizu podataka.

## 9. ZAKLJUČAK

Korištenje PowerBI alata za stvaranje vizualizacija u poslovnoj analitici pokazalo se kao jako pristupačno rješenje. Ovaj diplomski rad pokazao je kako se uz jednostavno i dobro strukturirano skladište podataka mogu dobiti ključni poslovni uvidi koji će omogućiti donošenje informiranih odluka u sektoru turističkog smještaja. PowerBI se ističe svojim grafičkim sučeljem i intuitivnim alatima za analize, zbog čega je pristupačan i korisnicima koji nemaju tehničku pozadinu i iskustvo s programiranjem ili složenim bazama podataka. Ovaj alat pruža fleksibilnost prilikom stvaranja izvještaja i korisnicima omogućuje stvaranje dinamičnih vizualizacija u nekoliko klikova. Korištenjem predefiniраниh vizualizacija, filtera i interaktivnih izvještaja mogu se analizirati podaci i donositi bitne poslovne odluke. Intuitivne nadzorne ploče omogućuju menadžerima i zaposlenicima brz uvid u ključne pokazatelje poslovanja, bez potrebe za pisanjem složenih upita i razumijevanjem baza podataka. Ključna stvar koja omogućuje jednostavnost analiza u PowerBI-u je kvalitetno dizajnirano skladište podataka. Bez obzira na veličinu organizacije i transakcijske baze podataka, skladište podataka će omogućiti centraliziranu pohranu svih najvažnijih informacija. ETL procesi osigurati će da se podaci redovito ažuriraju i transformiraju u oblik koji omogućava brzo dohvaćanje i analize. PowerBI omogućuje integraciju s različitim izvorima podataka, uključujući Excel, SQL baze podataka, Google Analytics i brojne druge, čime se olakšava integracija podataka iz različitih izvora unutar jednog analitičkog sučelja. Ovaj rad je pokazao da moderna poslovna inteligencija, uz kvalitetno dizajnirano skladište podataka, omogućuje jednostavnu, efikasnu i vizualno bogatu analizu podataka. PowerBI, uz dobro definirane procese prikupljanja i obrade podataka, omogućuje organizacijama da poboljšaju poslovne procese, unaprijede donošenje odluka i na taj način steknu konkurentsku prednost.

## POPIS SLIKA

Slika 1. Prikaz izvornih podataka .....	5
Slika 2. ER dijagram relacijske baze podataka .....	6
Slika 3. Izgled podataka nakon pripreme .....	11
Slika 4. Uvoz podataka - 1 .....	14
Slika 5. Uvoz podataka – 2 .....	15
Slika 6. Uvoz podataka – 3 .....	15
Slika 7. Uvoz podataka – 4 .....	16
Slika 8. Uvoz podataka - 5 .....	16
Slika 9. Završetak procesa uvoza podataka .....	17
Slika 10. Proces integracije podataka [5] .....	18
Slika 11. Komponente skladišta podataka [6] .....	20
Slika 12. Jednoslojna arhitektura skladišta podataka [7] .....	21
Slika 13. Dvoslojna arhitektura skladišta podataka [7] .....	21
Slika 14. Troslojna arhitektura skladišta podataka [7] .....	22
Slika 15. Prikaz ETL procesa [12] .....	22
Slika 16. Star schema .....	25
Slika 17. Snowflake schema .....	26
Slika 18. Shema skladišta podataka .....	28
Slika 19. Izgled Power BI Desktop sučelja .....	44
Slika 20. Pregled poslovanja smještajnih jedinica .....	45
Slika 21. Pregled poslovanja hotela Lero .....	45
Slika 22. Pregled poslovanja Ville Orabelle .....	46
Slika 23. Analiza soba hotela Lero .....	46
Slika 24. Analiza soba Ville Orabelle .....	47
Slika 25. Analiza gostiju hotela Lero .....	47
Slika 26. Analiza gostiju Ville Orabelle .....	48
Slika 27. Analiza prihoda po mjesecima .....	48
Slika 28. Analiza prihoda po godinama i smještajnim jedinicama .....	49
Slika 29. Broj rezervacija tijekom vikenda i radnih dana .....	49
Slika 30. Broj rezervacija i prihodi tijekom blagdana .....	50
Slika 31. Broj rezervacija tijekom sezona u godini .....	50
Slika 32. Analiza agencija i opcija rezervacije .....	51

## POPIS KODA

Kod 1. Promjena valute HRK u EUR .....	6
Kod 2. Rješavanje nedostajućih vrijednosti broja rezervacije .....	7
Kod 3. Gender-guesser detector.....	8
Kod 4. Generiranje datuma rođenja – 1 .....	9
Kod 5. Generiranje datuma rođenja - 2.....	10
Kod 6. Generiranje ID-a za goste .....	11
Kod 7. Stvaranje skladišta podataka .....	29
Kod 8. Stvaranje dimenzijske tablice d_hotel.....	29
Kod 9. Stvaranje dimenzijske tablice d_guest .....	29
Kod 10. Stvaranje dimenzijske tablice d_room .....	30
Kod 11. Stvaranje dimenzijske tablice d_date .....	30
Kod 12. Stvaranje dimenzijske tablice d_reservationType.....	30
Kod 13. Stvaranje dimenzijske tablice d_bookingOption .....	31
Kod 14. Stvaranje dimenzijske tablice d_agency .....	31
Kod 15. Stvaranje činjenične tablice f_reservation .....	32
Kod 16. Procedura za učitavanje podataka u d_hotel tablicu .....	33
Kod 17. Procedura za učitavanje podataka u d_guest tablicu.....	34
Kod 18. Procedura za učitavanje podataka u d_room tablicu.....	34
Kod 19. Procedura za učitavanje podataka u d_reservationType tablicu .....	35
Kod 20. Dodavanje vrijednosti u d_bookingOption tablicu .....	35
Kod 21. Procedura za učitavanje podataka u d_agency tablicu.....	36
Kod 22. Procedura za popunjavanje d_date tablice - 1.....	36
Kod 23. Procedura za popunjavanje d_date tablice – 2 .....	37
Kod 24. Procedura za popunjavanje d_date tablice – 3 .....	37
Kod 25. Procedura za popunjavanje d_date tablice – 4 .....	38
Kod 26. Procedura za učitavanje podataka u f_reservation tablicu-1.....	38
Kod 27. Procedura za učitavanje podataka u f_reservation-2.....	39
Kod 28. Procedura za učitavanje podataka u f_reservation tablicu-3.....	40
Kod 29. Procedura za učitavanje podataka u f_reservation tablicu-4.....	40
Kod 30. Naredbe za izvršavanje pohranjenih procedura .....	40
Kod 31. Funkcija za računanje turističke takse - 1 .....	41
Kod 32. Funkcija za računanje turističke takse .....	42
Kod 33. Računanje kumulativne sume prihoda .....	52
Kod 34. Funkcija za izračun godina gosta .....	52
Kod 35. Računanje broja odraslih.....	52
Kod 36. Računanje broja mladih od 12 do 18 godina.....	52
Kod 37. Računanje broja djece do 12 godina .....	52

## LITERATURA

- [1] »What is a decision support system (DSS)?,« [Mrežno]. Dostupno: <https://www.techtarget.com/searchcio/definition/decision-support-system>. [Datum pristupa: 10. siječnja 2025].
- [2] »PyPI,« [Mrežno]. Dostupno: <https://pypi.org/project/gender-guesser/>. [Datum pristupa: 12. prosinca 2024].
- [3] »Data Warehouses and Transactional Databases: What's the Difference?,« [Mrežno]. Dostupno: <https://builtin.com/articles/data-warehouse-vs-transactional-database>. [Datum pristupa: 18. prosinca 2024].
- [4] »What Is a Transactional Database?,« [Mrežno]. Dostupno: <https://www.snowflake.com/guides/what-transactional-database/>. [Datum pristupa: 18. prosinca 2024].
- [5] »An Introduction to Transactional Databases,« [Mrežno]. Dostupno: <https://www.mongodb.com/resources/basics/databases/transactional-databases>. [Datum pristupa: 18. prosinca 2024].
- [6] W. H. Inmon, Building The Data Warehouse, Fourth Edition, J. Wiley, 2005.
- [7] R. Kimball i M. Ross, The Data Warehouse Toolkit, Second Edition, John Wiley & Sons, Inc., 2002..
- [8] »Data Warehouse Architecture Explained,« [Mrežno]. Dostupno: <https://phoenixnap.com/kb/data-warehouse-architecture-explained>. [Datum pristupa: 15. veljače 2025].
- [9] »What is data quality?,« IBM, [Mrežno]. Dostupno: <https://www.ibm.com/topics/data-quality#:~:text=IBM,governance%20initiatives%20within%20an%20organization..> [Datum pristupa: 12. siječnja 2025].
- [10] R. Kimball, L. Reeves, M. Ross i W. Thorntwaite, The Data Warehouse Lifecycle Toolkit - Expert Methods for Designing, Developing and Deploying Data Warehouses, J. Wiley.
- [11] »How To Create a Stored Procedure In SQL Server Management Studio (SSMS),« [Mrežno]. Dostupno: <https://www.csharp.com/article/how-to-create-a-stored-procedure-in-sql-server-management-studio/>. [Datum pristupa: 10. veljače 2025].
- [12] »What is Power BI?,« [Mrežno]. Dostupno: <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>. [Datum pristupa: 25. siječnja 2025].

[13] »Build an ETL Data Pipeline using Python,« [Mrežno]. Dostupno:  
<https://blog.det.life/build-an-etl-data-pipeline-using-python-139c6875b046>. [Datum  
pristupa: 28. prosinca 2024].



## IZJAVA

Izjavljujem pod punom moralnom odgovornošću da sam diplomski rad izradila samostalno, na temelju znanja stečenog u dosadašnjem obrazovanju, koristeći se navedenim izvorima podataka i uz stručno vodstvo mentora prof.dr.sc. Marija Miličevića, kojem se još jednom zahvaljujem na svoj pruženoj pomoći prilikom izrade ovog diplomskog rada.

Josipa Vranić

---