**Tvrtka analizira podatke kako bi se identificirali ključni klijenti koji su generirali najveće troškove najma opreme i vozila tijekom odabrane vremenske periode. Analiza ponašanja visokopotrosačkih klijenata može pružiti dublji uvid u trendove na tržištu najma vozila i opreme te pomoći u donošenju informiranih poslovnih odluka.**

**Tvrtka analizira koji su to klijenti s najviše potrošenog novca na najam vozila i opreme. Time može bolje razumjeti potrebe tih klijenata i planirati kako im bolje pružiti usluge.**

1. **PRIKAŽI 3 KLIJENTA S NAJVIŠE POTROŠENOG NOVCA NA NAJAM VOZILA I OPREME**:

Da bismo pronašli klijente s najviše potrošenog novca na najam vozila i opreme, u upitu ćemo zbrojiti iznose transakcija za najam vozila i opreme za svakog klijenta te ih sortirati silazno prema ukupnom iznosu.

SELECT k.id AS id\_klijent,

CONCAT(k.ime, ' ', k.prezime) AS ime\_prezime\_klijenta,

SUM(t.iznos) AS ukupni\_iznos

FROM klijent k

INNER JOIN najam\_vozila nv ON k.id = nv.id\_klijent\_najam

INNER JOIN transakcija t ON nv.id\_transakcija\_najam = t.id

INNER JOIN oprema\_na\_najmu onr ON nv.id = onr.id\_najam\_vozila

GROUP BY k.id

ORDER BY ukupni\_iznos DESC

LIMIT 3;

Ovaj SQL upit omogućava tvrtki identifikaciju ključnih klijenata koji su generirali najveće troškove najma vozila i opreme tijekom odabranog vremenskog perioda. Evo detaljnog opisa koraka u izvršavanju upita:

1. **Selektiranje potrebnih podataka:**
   * Počinjemo s odabirom atributa koje želimo prikazati u rezultatima. U ovom slučaju, odabiremo identifikator klijenta (**k.id**), ime i prezime klijenta spojeno u jedan tekstualni atribut pomoću funkcije **CONCAT(k.ime, ' ', k.prezime)**, te ukupni iznos potrošenog novca na najam vozila i opreme za svakog klijenta, što je izračunato funkcijom **SUM(t.iznos)**.
2. **Povezivanje tablica i grupiranje podataka:**
   * Nakon odabira potrebnih atributa, koristimo INNER JOIN kako bismo povezali tablice **klijent**, **najam\_vozila**, **transakcija** i **oprema\_na\_najmu** na odgovarajuće ključeve. Ovo osigurava da svaki redak rezultata sadrži informacije iz svih navedenih tablica koje su relevantne za izračun ukupnog iznosa potrošenog novca.
   * Nakon toga, koristimo GROUP BY klauzulu kako bismo grupirali rezultate po identifikatoru klijenta (**k.id**). Ovo omogućava izračun ukupnog iznosa potrošenog novca za svakog klijenta zasebno.
3. **Sortiranje rezultata i ograničenje na prvih 3 klijenta:**
   * Nakon grupiranja podataka, sortiramo rezultate silazno prema ukupnom iznosu potrošenog novca (atribut **ukupni\_iznos**) koristeći ORDER BY klauzulu. To osigurava da prvo prikažemo klijente s najvećim potrošnjama na vrhu rezultata.
   * Konačno, koristimo LIMIT klauzulu kako bismo ograničili prikaz samo na prvih 3 klijenta. Ovo nam omogućava fokusiranje na najveće potrošače.

SELECT k.id AS id\_klijent, -- Odaberi ID klijenta i preimenuj ga kao "id\_klijent"

CONCAT(k.ime, ' ', k.prezime) AS ime\_prezime\_klijenta, -- Spoji ime i prezime klijenta i preimenuj rezultat kao "ime\_prezime\_klijenta"

SUM(t.iznos) AS ukupni\_iznos -- Izračunaj ukupni iznos transakcija i nazovi ga "ukupni\_iznos"

FROM klijent k -- Tablica klijenata

INNER JOIN najam\_vozila nv ON k.id = nv.id\_klijent\_najam -- Pridruži tablicu najma vozila s klijentima

INNER JOIN transakcija t ON nv.id\_transakcija\_najam = t.id -- Pridruži tablicu transakcija s najmom vozila

INNER JOIN oprema\_na\_najmu onr ON nv.id = onr.id\_najam\_vozila -- Pridruži tablicu opreme na najmu s tablicom najma vozila

GROUP BY k.id -- Grupiraj rezultate po ID-u klijenta

ORDER BY ukupni\_iznos DESC -- Poredaj rezultate silazno po ukupnom iznosu

LIMIT 3; -- Prikazi prvA 3 retka rezultata

Ovaj upit će izračunati ukupni iznos potrošenog novca na najam vozila i opreme za 3 klijenta, grupirati rezultate po klijentu, sortirati ih silazno prema ukupnom iznosu te vratiti informacije o klijentu s najviše potrošenog novca na vrhu rezultata.

Evo korak-po-korak objašnjenja upita:

SELECT: Prvo definiramo što želimo izvući iz baze podataka. U ovom slučaju, želimo dobiti ID klijenta, ime i prezime klijenta te ukupni iznos potrošenog novca.

FROM: Zatim navodimo iz kojih tablica želimo dohvatiti podatke. Koristimo tablice klijent, najam\_vozila i transakcija.

JOIN: Povezujemo tablice koristeći JOIN kako bismo dobili sve potrebne informacije. Spajamo klijent tablicu s najam\_vozila tablicom prema ID-u klijenta te zatim najam\_vozila tablicu s transakcija tablicom prema ID-u transakcije.

GROUP BY: Grupiramo rezultate po ID-u klijenta. To znači da će se sve transakcije za istog klijenta zbrojiti zajedno.

SUM: Koristimo funkciju SUM kako bismo zbrojili iznose transakcija za svakog klijenta.

ORDER BY: Sortiramo rezultate prema ukupnom iznosu silazno (od najvećeg prema najmanjem). To nam omogućuje da prvi rezultat bude klijent s najviše potrošenog novca.

LIMIT: Na kraju ograničavamo broj rezultata koje želimo vratiti na 3, jer nas zanimaju samo klijenti s najviše potrošenog novca.

**Ako se određena vrsta opreme rijetko iznajmljuje, tvrtka može smanjiti zalihu te opreme kako bi oslobodila skladišni prostor i smanjila troškove održavanja zaliha. Također može donijeti odluke o tome treba li smanjiti količinu određene opreme ili je ukloniti iz ponude.**

**2.Prikaz 5 vrsta dodatne ppremu koja se najmanje puta iznajmila:**

Pogled koji će izračunati broj najmova za svaku vrstu opreme:

CREATE VIEW Broj\_Najmova\_Opreme AS

SELECT o.naziv AS Vrsta\_Opreme, COUNT(onr.id\_najam\_vozila) AS Broj\_Najmova

FROM oprema o

LEFT JOIN oprema\_na\_najmu onr ON o.id = onr.id\_oprema

GROUP BY o.naziv;

Zatim ćemo koristiti taj pogled u našem upitu kako bismo pronašli vrstu opreme koja se najmanje iznajmljivala:

SELECT Vrsta\_Opreme

FROM Broj\_Najmova\_Opreme

ORDER BY Broj\_Najmova ASC

LIMIT 5;

1. **Kreiranje pogleda za izračun broja najmova za svaku vrstu opreme:**
   * Prvo kreiramo pogled nazvan "Broj\_Najmova\_Opreme" koji će izračunati broj najmova za svaku vrstu opreme.
   * U ovom pogledu koristimo SELECT naredbu kako bismo odabrali naziv vrste opreme (**o.naziv**) i broj najmova za tu vrstu opreme, što računamo funkcijom COUNT.
   * Povezujemo tablicu "oprema" s tablicom "oprema\_na\_najmu" pomoću LEFT JOIN kako bismo osigurali da uključimo sve vrste opreme, čak i one koje se nisu iznajmljivale.
   * Konačno, grupiramo rezultate po nazivu vrste opreme pomoću GROUP BY klauzule.
2. **Korištenje pogleda za pronalaženje vrste opreme koja se najmanje puta iznajmljivala:**
   * Nakon što smo kreirali pogled za izračun broja najmova za svaku vrstu opreme, koristimo ga u glavnom upitu kako bismo pronašli vrstu opreme koja se najmanje puta iznajmljivala.
   * Koristimo SELECT naredbu za odabir naziva vrste opreme iz pogleda "Broj\_Najmova\_Opreme".
   * Sortiramo rezultate silazno po broju najmova (**Broj\_Najmova**) koristeći ORDER BY klauzulu s ASC opcijom kako bismo dobili vrstu opreme koja se najmanje iznajmljivala na vrhu rezultata.
   * Na kraju, koristimo LIMIT klauzulu kako bismo ograničili prikaz na prvih 5 vrsta opreme, fokusirajući se samo na one koje su se najmanje puta iznajmljivale.

CREATE VIEW Broj\_Najmova\_Opreme AS -- Kreiranje pogleda koji će izračunati broj najmova za svaku vrstu opreme

SELECT o.naziv AS Vrsta\_Opreme, COUNT(onr.id\_najam\_vozila) AS Broj\_Najmova -- Odabir naziva opreme i brojanje najmova za svaku vrstu opreme

FROM oprema o

LEFT JOIN oprema\_na\_najmu onr ON o.id = onr.id\_oprema -- Spajanje tablice oprema s tablicom oprema\_na\_najmu koristeći LEFT JOIN kako bismo uključili sve vrste opreme čak i ako nisu iznajmljene

GROUP BY o.naziv; -- Grupiranje rezultata po nazivu opreme

SELECT Vrsta\_Opreme -- Korištenje pogleda u glavnom upitu kako bismo pronašli vrstu opreme koja se najmanje iznajmljivala

FROM Broj\_Najmova\_Opreme

ORDER BY Broj\_Najmova ASC -- Sortiranje rezultata uzlazno prema broju najmova, tako da prva vrsta opreme bude ona koja se najmanje iznajmljivala

LIMIT 1;

**Tvrtka provjerava koja su vozila prešla više kilometara nego što je očekivano tijekom najma. To joj može pomoći da održava vozni park u dobrom stanju i izbjegava neočekivane troškove popravaka.**

**3.PROVJERA NAJMA VOZILA KOJI CU PREMAŠILI ODREĐENI BROJ KILOMETARA:**

SELECT nv.id, v.registracijska\_tablica, nv.datum\_pocetka, nv.datum\_zavrsetka,

nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza AS Prijeđeni\_Kilometri

FROM najam\_vozila nv

INNER JOIN vozilo v ON nv.id\_vozilo = v.id

WHERE nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza > 100;

Ovaj SQL upit omogućava tvrtki provjeru vozila koja su prešla više kilometara nego što je očekivano tijekom najma. Evo detaljnog opisa koraka u izvršavanju upita:

1. **Selektiranje potrebnih podataka:**
   * **Upit koristi SELECT klauzulu kako bi odabrao atribute koji će biti prikazani u rezultatima.**
   * **Odabrani atributi uključuju identifikator najma vozila (nv.id), registracijsku tablicu vozila (v.registracijska\_tablica), datume početka i završetka najma (nv.datum\_pocetka i nv.datum\_zavrsetka), te količinu prijeđenih kilometara (nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza) izračunatu kao razlika između završne i početne kilometraže.**
2. **Povezivanje tablica:**
   * **Koristi se INNER JOIN kako bi se povezale tablice "najam\_vozila" (nv) i "vozilo" (v) na odgovarajuće ključeve.**
   * **Povezivanje se odvija prema atributima id\_vozilo u tablici "najam\_vozila" i id u tablici "vozilo", što omogućava spajanje informacija o najmu vozila s informacijama o samom vozilu.**
3. **Postavljanje uvjeta filtriranja:**
   * **U WHERE dijelu upita dodan je uvjet nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza > 100, što provjerava je li količina prijeđenih kilometara veća od 100.**
   * **Ovaj uvjet osigurava da se prikažu samo najmovi vozila koji su premašili određeni broj kilometara.**

Ovaj upit omogućava tvrtki da identificira vozila koja su prešla više kilometara od očekivanog tijekom najma, što može pomoći u održavanju vozila u dobrom stanju i izbjegavanju neočekivanih troškova popravaka.

-- Selektiramo stupce koje sadrže ID najma vozila, registracijsku tablicu vozila, datume početka i završetka najma,

-- te računamo razliku između završne i početne kilometraže kako bismo dobili broj prijeđenih kilometara.

-- Ograničavamo rezultate na slučajeve gdje je razlika između završne i početne kilometraže veća od 100.

SELECT nv.id, v.registracijska\_tablica, nv.datum\_pocetka, nv.datum\_zavrsetka,

nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza AS Prijeđeni\_Kilometri

FROM najam\_vozila nv

INNER JOIN vozilo v ON nv.id\_vozilo = v.id -- Spajamo tablicu najma vozila s tablicom vozila na osnovu ID-a vozila

WHERE nv.zavrsna\_kilometraza - nv.pocetna\_kilometraza > 100; -- Filtriramo rezultate gdje je razlika između završne i početne kilometraže veća od 100

Ovaj upit provjerava najmove vozila koji su premašili 100 kilometara, prikazujući registracijsku tablicu vozila, datume najma i povratka, te ukupan broj prijeđenih kilometara.

**Tvrtka želi redovito analizirati svoje poslovanje iz prethodnih godina kako bi dobila uvid u trendove i promjene u potražnji za najmom vozila. Žele znati kako im je išlo poslovanje u godinama prije ove da vide kako su napredovali.**

**Tvrtka želi analizirati svoje poslovne troškove kako bi bolje razumjela gdje se trošio novac tokom prvih 6 mjeseci poslovanja u 2024.godini i kako bi identificirala područja u kojima može uštedjeti ili optimizirati troškove.**

**4. PRIKAŽI SVE POSLOVNE TROŠKOVE ZA REZERVACIJE KOJE SU SE DOGODILE U PRVIH 6 MJESECI U 2024.GOD:**

SELECT pt.\*

FROM poslovni\_trosak pt

JOIN transakcija t ON pt.id\_transakcija\_poslovnog\_troska = t.id

WHERE (MONTH(t.datum) BETWEEN 1 AND 6) AND YEAR(t.datum) = 2024;

1. **Selektiranje potrebnih podataka:**
   * Upit koristi SELECT klauzulu kako bi odabrao sve atribute iz tablice "poslovni\_trosak" (pt).
2. **Povezivanje tablica:**
   * Koristi se INNER JOIN kako bi se povezale tablice "poslovni\_trosak" (pt) i "transakcija" (t) na odgovarajuće ključeve.
   * Povezivanje se odvija prema atributu **id\_transakcija\_poslovnog\_troska** u tablici "poslovni\_trosak" i atributu **id** u tablici "transakcija", što omogućava spajanje informacija o poslovnim troškovima s informacijama o transakcijama.
3. **Postavljanje uvjeta filtriranja:**
   * U WHERE dijelu upita dodan je uvjet koji provjerava datume transakcija kako bi se prikazali samo oni poslovni troškovi koji su se dogodili u prvih 6 mjeseci 2024. godine.
   * Korištena je funkcija MONTH(t.datum) kako bi se dohvatio mjesec iz datuma transakcije, a zatim je provjereno je li taj mjesec između 1 i 6 (uključivo).
   * Dodatno, korištena je funkcija YEAR(t.datum) kako bi se provjerilo je li godina datuma transakcije jednaka 2024.

Ovaj upit omogućava tvrtki da pregleda sve svoje poslovne troškove za rezervacije koje su se dogodile u prvih 6 mjeseci 2024. godine, što im pruža uvid u potrošnju i omogućava identificiranje područja u kojima mogu uštedjeti ili optimizirati troškove.

Top of Form

SELECT pt.\* -- Odaberi sve stupce iz tablice poslovnih troškova

FROM poslovni\_trosak pt -- Pridruži tablicu poslovnih troškova

JOIN transakcija t ON pt.id\_transakcija\_poslovnog\_troska = t.id -- Pridruži tablicu transakcija

WHERE (MONTH(t.datum) BETWEEN 1 AND 6) AND YEAR(t.datum) = 2024; -- Filtriraj transakcije koje su se dogodile u prvih šest mjeseci 2024.

Ovaj upit ima sljedeće korake:

1. **SELECT pt.\*:** Ovaj dio označava da želimo izlistati sve stupce (**\***) iz tablice **poslovni\_trosak**.
2. **FROM poslovni\_trosak pt:** Ovdje navodimo glavnu tablicu iz koje želimo dohvatiti podatke, a to je **poslovni\_trosak**. Kao alias za ovu tablicu koristimo **pt**.
3. **JOIN transakcija t ON pt.id\_transakcija\_poslovnog\_troska = t.id:** Ovo je ključni dio upita gdje spajamo tablicu **poslovni\_trosak** s tablicom **transakcija** na temelju njihovih ključeva. Spajamo ih tako da se vrijednost **id\_transakcija\_poslovnog\_troska** iz tablice **poslovni\_trosak** podudara s vrijednošću **id** u tablici **transakcija**.
4. **WHERE (MONTH(t.datum) BETWEEN 1 AND 6) AND YEAR(t.datum) = 2024;:** Ovo ograničenje filtrira rezultate prema datumu transakcije. Konkretno, želimo samo prvih 6 mjeseci u 2024. godini. Koristimo funkcije **MONTH(t.datum)**  **I YEAR(t.datum)** koje predstavljaju stupac datuma u tablici transakcija

**Tvrtka identificira države u kojima su smještene pravne osobe koje generiraju najveći promet u najmu vozila. Analizom najma vozila po državi sjedišta pravne osobe, tvrtka stječe uvid u geografsku distribuciju svojih najvažnijih klijenata. Razumijevanje ovih podataka može pomoći tvrtki u prilagodbi marketinških strategija i pružanju bolje usluge kako bi zadovoljila potrebe klijenata u različitim regijama."**

**Tvrtka proučava u kojim državama se nalaze tvrtke koje najviše koriste najam automobila. To joj može pomoći da shvati gdje najviše posluje i kako prilagoditi svoje usluge.**

**5.PRONALAŽENJE NAJMA VOZILA PO DRŽAVI SJEDIŠTA PRAVNE OSOBE:**

SELECT p.drzava\_sjediste AS Drzava, COUNT(nv.id) AS Broj\_Najmova

FROM pravna\_osoba p

INNER JOIN klijent k ON p.id = k.id\_pravna\_osoba

INNER JOIN najam\_vozila nv ON k.id = nv.id\_klijent\_najam

GROUP BY p.drzava\_sjediste

ORDER BY Broj\_Najmova DESC;

Ovaj SQL upit omogućava tvrtki identifikaciju država u kojima su smještene pravne osobe koje generiraju najveći promet u najmu vozila. Evo detaljnog opisa koraka u izvršavanju upita:

1. **Selektiranje potrebnih podataka:**
   * **Upit koristi SELECT klauzulu kako bi odabrao dva atributa:**
     + **p.drzava\_sjediste kao naziv države sjedišta pravne osobe.**
     + **COUNT(nv.id) kako bi se izračunao broj najmova vozila za svaku državu.**
2. **Povezivanje tablica:**
   * **Koristi se INNER JOIN kako bi se povezale tri tablice: "pravna\_osoba" (p), "klijent" (k) i "najam\_vozila" (nv).**
   * **Prvo se povezuju tablice "pravna\_osoba" (p) i "klijent" (k) prema atributu id\_pravna\_osoba.**
   * **Zatim se povezuju tablice "klijent" (k) i "najam\_vozila" (nv) prema atributu id\_klijent\_najam.**
3. **Grupiranje podataka:**
   * **Rezultati se grupiraju po atributu p.drzava\_sjediste kako bi se izračunao broj najmova vozila za svaku državu.**
4. **Sortiranje rezultata:**
   * **Rezultati se sortiraju silazno prema broju najmova vozila (Broj\_Najmova) koristeći ORDER BY klauzulu s DESC opcijom, što znači da će država s najvećim brojem najmova biti prva na popisu.**

**Ovaj upit omogućava pronalaženje broja najmova vozila za svaku državu sjedišta pravne osobe, što može pružiti uvid u aktivnosti najma vozila različitih tvrtki u različitim državama.**

SELECT p.drzava\_sjediste AS Drzava, COUNT(nv.id) AS Broj\_Najmova -- Odaberi drzavu sjedista pravnih osoba i broj najmova vozila, preimenuj stupce

FROM pravna\_osoba p -- Tablica pravnih osoba

INNER JOIN klijent k ON p.id = k.id\_pravna\_osoba -- Pridruzivanje tablice klijenata s tablicom pravnih osoba

INNER JOIN najam\_vozila nv ON k.id = nv.id\_klijent\_najam -- Pridruzivanje tablice najma vozila s tablicom klijenata

GROUP BY p.drzava\_sjediste -- Grupiranje rezultata po drzavi sjedista pravnih osoba

ORDER BY Broj\_Najmova DESC; -- Poredaj rezultate silazno po broju najmova vozila

Ovaj upit broji broj najmova vozila po državi sjedišta pravnih osoba koje su klijenti.

**Tvrtka treba detaljan uvid u troškove najma vozila po zaposleniku, što je korisno za identificiranje ključnih zaposlenika koji generiraju najveće troškove u ovoj kategoriji. Analizom troškova najma vozila po zaposleniku, tvrtka stječe uvid u individualne potrebe zaposlenika za mobilnošću i transportom te u efikasnost korištenja vozila unutar organizacije. Ove informacije su od ključnog značaja za optimizaciju poslovnih procesa, upravljanje resursima i prilagodbu politike najma vozila kako bi se postigla maksimalna efikasnost i smanjili nepotrebni troškovi.**

**Tvrtka istražuje koliko je novca svaki zaposlenik potrošio na najam automobila. To joj može pomoći da vidi tko ima najveće troškove u ovoj kategoriji i kako ih možda optimizirati.**

**6.PRIKAZ TROŠKOVA NAJMA VOZILA PO ZAPOSLENIKU:**

Prvo ćemo kreirati pogled koji će izračunati ukupne troškove najma vozila po zaposleniku:

CREATE VIEW Ukupni\_Troskovi\_Najma\_Zaposlenika AS

SELECT z.id AS id\_zaposlenik, z.ime, z.prezime, SUM(t.iznos) AS ukupni\_troskovi

FROM zaposlenik z

INNER JOIN najam\_vozila nv ON z.id = nv.id\_zaposlenik\_najam

INNER JOIN transakcija t ON nv.id\_transakcija\_najam = t.id

GROUP BY z.id, z.ime, z.prezime;

Zatim, možemo koristiti taj pogled u našem upitu:

SELECT ime, prezime, ukupni\_troskovi

FROM Ukupni\_Troskovi\_Najma\_Zaposlenika

ORDER BY ukupni\_troskovi DESC;

Ovaj SQL upit omogućava tvrtki prikaz troškova najma vozila po zaposleniku, što je korisno za identificiranje ključnih zaposlenika koji generiraju najveće troškove u toj kategoriji. Evo detaljnog opisa koraka u izvršavanju upita:

1. **Kreiranje pogleda za izračun ukupnih troškova najma vozila po zaposleniku:**
   * **Prvo se kreira pogled nazvan "Ukupni\_Troskovi\_Najma\_Zaposlenika" pomoću CREATE VIEW naredbe.**
   * **U ovom pogledu koriste se tri tablice: "zaposlenik" (z), "najam\_vozila" (nv) i "transakcija" (t).**
   * **Koristi se INNER JOIN kako bi se povezale tablice "zaposlenik" (z) i "najam\_vozila" (nv) prema atributu id\_zaposlenik\_najam.**
   * **Zatim se povezuju tablice "najam\_vozila" (nv) i "transakcija" (t) prema atributu id\_transakcija\_najam.**
   * **Rezultati se grupiraju po atributima z.id (identifikator zaposlenika), z.ime (ime zaposlenika) i z.prezime (prezime zaposlenika) kako bi se izračunali ukupni troškovi najma vozila za svakog zaposlenika.**
   * **SUM(t.iznos) se koristi za izračun ukupnih troškova najma vozila za svakog zaposlenika.**
2. **Korištenje pogleda u glavnom upitu:**
   * **Nakon što je pogled kreiran, možemo koristiti rezultate tog pogleda u glavnom upitu.**
   * **SELECT klauzula se koristi za odabir atributa koji će biti prikazani u rezultatima glavnog upita, u ovom slučaju to su ime, prezime i ukupni\_troskovi (ukupni troškovi najma vozila).**
   * **Rezultati se sortiraju silazno prema ukupnim troškovima (ukupni\_troskovi) koristeći ORDER BY klauzulu s DESC opcijom, što znači da će zaposlenici s najvišim ukupnim troškovima biti prvi na popisu.**

**Ovaj upit omogućava prikaz ukupnih troškova najma vozila po zaposleniku, što može biti korisno za praćenje troškova najma vozila za svakog zaposlenika u tvrtki.**

**Top of Form**

CREATE VIEW Ukupni\_Troskovi\_Najma\_Zaposlenika AS

-- Stvori pogled koji će prikazivati ukupne troškove najma vozila po zaposlenicima

SELECT z.id AS id\_zaposlenik, z.ime, z.prezime, SUM(t.iznos) AS ukupni\_troskovi

-- Odaberi ID, ime i prezime zaposlenika, te ukupni iznos troškova

FROM zaposlenik z -- Tablica zaposlenika

INNER JOIN najam\_vozila nv ON z.id = nv.id\_zaposlenik\_najam -- Pridruži tablicu najma vozila s tablicom zaposlenika

INNER JOIN transakcija t ON nv.id\_transakcija\_najam = t.id -- Pridruži tablicu transakcija s tablicom najma vozila

GROUP BY z.id, z.ime, z.prezime; -- Grupiraj rezultate po ID-u, imenu i prezimenu zaposlenika

-- Zatim, možemo koristiti taj pogled u našem upitu:

SELECT ime, prezime, ukupni\_troskovi

-- Odaberi ime, prezime i ukupne troškove zaposlenika iz pogleda

FROM Ukupni\_Troskovi\_Najma\_Zaposlenika -- Koristi stvoreni pogled

ORDER BY ukupni\_troskovi DESC; -- Poredaj rezultate silazno po ukupnim troškovima

Ovaj upit će prikazati ukupne troškove najma vozila po zaposleniku koristeći prethodno kreirani pogled.