Database Analytics and Performance on Data From Telecom Customer Service

Janka Šepetková, Rebeka Panáková, Michal Kvet
University of Žilina
Žilina, Slovakia
sepetkova@stud.uniza.sk , panakova2@stud.uniza.sk , michal.kvet@uniza.sk

***Abstract*—This work focuses on the analysis and performance optimization of a telecom company's database using indexing and related technologies. The dataset consists of records from 7,000 customers across various cities in California, capturing diverse customer behaviors, preferences, and service usage patterns. The analysis aims to identify key trends in customer churn, acquisition, and retention by examining demographic information, service details, and contract changes up to June 22, 2022. Additionally, the study explores how database indexing techniques improve query performance, enabling efficient data retrieval and enhancing decision-making. The findings provide insights into optimizing database operations while helping the company develop strategies to reduce churn and improve customer satisfaction.**

I. INTRODUCTION

The data used in this analysis was collected from 7000 customers located across various cities in California. These cities show the wide range of customers the telecom company serves, meaning the analysis looks at different customer needs, preferences, and how they use services. This variety of locations is important for understanding how different areas might affect **churn** (when customers leave the company), **acquisition** (when new customers join), and retention rates, and how the company can adjust its services and marketing to better meet the needs of each region. The data also contains detailed information about the customers' demographics, services, and current status.

The data was collected up to June 22, 2022. By looking at this period, we can find recent trends in customer acquisition, churn rates, and how factors like changes in services, prices, and contracts affect customer decisions. Understanding these trends will help the company find ways to improve customer retention and reduce churn.

II. IMPORTANCE OF DATA ANALYSIS AND PERFORMANCE EVALUATION STUDY

Data analysis is crucial for a variety of reasons, as it helps organizations, researchers, and individuals make informed decisions and gain valuable insights. Here's a breakdown of the importance of data analysis:

- Decision-making support – data analysis helps individuals and organizations make decisions based on real evidence rather than guesswork or intuition. This is especially valuable for businesses, as they can optimize strategies, allocate resources efficiently, and improve operations [1], [2].
- Patterns and trends management – by analyzing large datasets, patterns, trends, and correlations emerge that may not be immediately obvious. This allows organizations to anticipate future outcomes and adapt their strategies accordingly [3], [4].
- Operational efficiency – it enables businesses to streamline processes, reduce costs, and improve productivity [5].
- Predictive analysis – data analysis can predict future trends by examining historical data [5], [6], [7].
- Temporal evaluation – monitoring changes over time [8].
- Problem-solving – it helps identify the root causes of problems [9], [10].
- Customer insights – analyzing real customer data, habits, trends, etc [11].
- Cost reduction – it can reveal areas where resources are being wasted, such as redundant processes, overstocked inventory, or underutilized employees. This insight allows organizations to optimize their spending and reduce unnecessary costs [12].
- Competitive advantage – gaining a competitive edge by understanding their market, customer behavior, and industry trends. This allows them to make proactive adjustments to stay ahead of competitors [13], [14].
- Evidence-based research – in academic or scientific fields, data analysis is the backbone of evidence-based research. It allows researchers to draw valid conclusions, test hypotheses, and verify theories, leading to new discoveries and innovations.
- Data-driven analysis – emphasizing data analysis fosters a culture of decision-making driven by insights and facts rather than emotions or biases [15], [16].
- Regulatory compliance – companies are required to collect, store, and analyze data to comply with regulations. Proper data analysis ensures compliance with standards, such as those related to financial reporting, healthcare privacy, or environmental protection.

- Environmental aspect – it refers to the impact that collecting, storing, processing, and transmitting data has on the natural environment. As the digital world expands, so does its environmental footprint – energy consumption, carbon emmisions, e-waste, cooling systems, data transmisson, sustainability, etc. [17].

SQL performance tuning, is the process of optimizing SQL queries and database performance to improve efficiency and reduce execution time [5]. The goal of SQL tuning is to ensure that SQL queries execute as quickly and efficiently as possible, especially when working with large datasets or complex queries [6]. It can be done by various techniques, like indexes, query structure optimization, data layer optimization, normalization aiming to reduce full table scans and optimizing joins [7]. In this paper, performance is evaluated through the EXPLAIN PLAN, which the step-by-step details of the statement execution. By analyzing the query execution plan, inefficient parts of the query, such as full table scans or unnecessary joins can be easily identified to take steps to optimize them.

To evaluate the performance, we used the Oracle database system, which provides the largest portfolio of optimization options. On the other hand, the presented solutions are generally applicable to any database system.

III. DATA STRUCTURE AND KEY ATTRIBUTES

This dataset contains two tables (Zip_code_population (Fig. 1) and Customers (Fig. 2)) that hold distinct pieces of information.

	SEPETKOVA.ZIP_CODE_POPULATION	
P *	ZIP_CODE	VARCHAR2 (10 CHAR)
	POPULATION	NUMBER (10)
	ZIP_CODE_POPULATION_PK (ZIP_CODE)	

Fig. 1. Table Zip_code_population

A. Table customers (Fig. 1)

1) *Demographic Data:* This includes basic customer information such as Customer ID, Gender, Age, Marital Status, Number of Dependents, and more. This helps provide insight into the customer profile.

2) *Service Usage Data:* Information on which services customers are subscribed to, including Phone Service, Internet Service, Monthly Charges, and Data Usage.

3) *Churn Information:* The dataset indicates if a customer churned, stayed, or joined the company by the end of each quarter. It also records the Churn Category and Churn Reason.

4) *Geographic Data:* Information about the customer's city, zip code, latitude, and longitude provides insight into customer distribution across California.

B. Table zip_code_population (Fig. 2)

1) *Geographic Data:* The estimated population of the zip code area. This value represents the total number of people living in that zip code and helps provide context for customer distribution and market size in each region.

	SEPETKOVA.CUSTOMERS	
P *	CUSTOMER_ID	VARCHAR2 (10 BYTE)
	GENDER	VARCHAR2 (10 BYTE)
	AGE	NUMBER (*,0)
	MARRIED	CHAR (1 BYTE)
	NUMBER_OF_DEPENDENTS	NUMBER (*,0)
	CITY	VARCHAR2 (50 BYTE)
	ZIP_CODE	VARCHAR2 (10 BYTE)
	LATITUDE	NUMBER (9,6)
	LONGITUDE	NUMBER (10,6)
	NUMBER_OF_REFERRALS	NUMBER (*,0)
	TENURE_IN_MONTHS	NUMBER (*,0)
	OFFER	VARCHAR2 (20 BYTE)
	PHONE_SERVICE	CHAR (1 BYTE)
	AVG_MONTHLY_LONG_DISTANCE_CHARGES	NUMBER (5,2)
	MULTIPLE_LINES	CHAR (1 BYTE)
	INTERNET_SERVICE	CHAR (1 BYTE)
	INTERNET_TYPE	VARCHAR2 (20 BYTE)
	AVG_MONTHLY_GB_DOWNLOAD	NUMBER (*,0)
	ONLINE_SECURITY	CHAR (1 BYTE)
	ONLINE_BACKUP	CHAR (1 BYTE)
	DEVICE_PROTECTION_PLAN	CHAR (1 BYTE)
	PREMIUM_TECH_SUPPORT	CHAR (1 BYTE)
	STREAMING_TV	CHAR (1 BYTE)
	STREAMING_MOVIES	CHAR (1 BYTE)
	STREAMING_MUSIC	CHAR (1 BYTE)
	UNLIMITED_DATA	CHAR (1 BYTE)
	CONTRACT	VARCHAR2 (20 BYTE)
	PAPERLESS_BILLING	CHAR (1 BYTE)
	PAYMENT_METHOD	VARCHAR2 (20 BYTE)
	MONTHLY_CHARGE	NUMBER (5,2)
	TOTAL_CHARGES	NUMBER (10,2)
	TOTAL_REFUNDS	NUMBER (10,2)
	TOTAL_EXTRA_DATA_CHARGES	NUMBER (10,2)
	TOTAL_LONG_DISTANCE_CHARGES	NUMBER (10,2)
	TOTAL_REVENUE	NUMBER (10,2)
	CUSTOMER_STATUS	VARCHAR2 (20 BYTE)
	CHURN_CATEGORY	VARCHAR2 (20 BYTE)
	CHURN_REASON	VARCHAR2 (50 BYTE)
	CUSTOMERS_PK (CUSTOMER_ID)	

Fig. 2. Customers table

IV. RESEARCH

When we created the CUSTOMERS table, we defined a primary key on the CUSTOMER_ID column, which automatically creates an index for fast lookups. However, if there are no additional unique constraints, Oracle won't create any other automatic indexes. Without these additional indexes, queries involving columns other than CUSTOMER_ID may perform slower, as Oracle must scan the table without optimized access paths.

A. Churn percentage across cities

To see relationships in the data, we aimed to visualize the churn percentage for company customers across different cities. This would help us identify any potential issues or areas for improvement. To achieve this, we used the SQL query in Fig. 3.

```
SELECT c.City, COUNT(*) AS Churned_Customers, z.Population,
    ROUND((COUNT(*) * 100.0 / z.Population), 2) AS Churn_Percentage,
    DENSE_RANK() OVER (ORDER BY (COUNT(*) * 100.0 / z.Population) DESC) AS Rank
FROM CUSTOMERS c
JOIN ZIP_CODE_POPULATION z
    ON c.Zip_Code = z.Zip_Code
WHERE c.Customer_Status = 'Churned'
GROUP BY City, Population
ORDER BY Rank;
```

Fig. 3. Select for churn percentage across cities

An example of the above query is shown in Fig. 4.

	CITY	CHURNED_CUSTOMERS	POPULATION	CHURN_PERCENTAGE	RANK
1	Dunnigan	2	19	10.53	1
2	Los Angeles	2	21	9.52	2
3	Glencoe	2	21	9.52	2
4	Malibu	1	11	9.09	3
5	Ludlow	2	23	8.7	4
6	Twain	4	73	5.48	5
7	Weimar	1	31	3.23	6

Fig. 4. Output for select for churn percentage across cities

After reviewing the execution plan, we noticed that the cost of the operation was quite high – 75 (Fig. 5). The most significant cost are related to the full table scanning.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1869	75
WINDOW		SORT	1869	75
HASH		GROUP BY	1869	75
HASH JOIN			1869	71
Access Predicates				
C.ZIP_CODE=Z.ZIP_CODE				
TABLE ACCESS	ZIP_CODE_POPULATION	FULL	1671	3
TABLE ACCESS	CUSTOMERS	FULL	1869	68
Filter Predicates				
C.CUSTOMER_STATUS='Churned'				

Fig. 5. Plan for select for churn percentage across cities

The high cost could have been caused by the demanding JOIN operation. To optimize this, we used an explicit ON clause instead of USING to ensure the most efficient use of indexes. The USING clause might be less flexible in certain cases because it is applied to columns with the same names in both tables. By using an explicit ON clause, we gain more control over the type of JOIN join being performed, which can result in better performance. Specifically, the ON clause allows us to be more precise in how the join conditions are applied, enabling the database to make better use of indexes and reduce unnecessary full table scans.

Despite using the ON clause and explicitly creating an index on the Zip_Code and Customer_Status columns in the CUSTOMERS table, we were unable to reduce the cost of the query (Fig. 6). Chart for the Churn percentage is depicted by Fig. 7.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1869	75
WINDOW		SORT	1869	75
HASH		GROUP BY	1869	75
HASH JOIN			1869	71
Access Predicates				
C.ZIP_CODE=Z.ZIP_CODE				
NESTED LOOPS			1869	71
NESTED LOOPS				
STATISTICS COLLECTOR				
TABLE ACCESS	ZIP_CODE_POPULATION	FULL	1671	3
INDEX	IDX_ZIP_CODE	RANGE SCAN		
Access Predicates				
C.ZIP_CODE=Z.ZIP_CODE				
TABLE ACCESS	CUSTOMERS	BY INDEX R...	1	68
Filter Predicates				
C.CUSTOMER_STATUS='Churned'				
TABLE ACCESS	CUSTOMERS	FULL	1869	68
Filter Predicates				
C.CUSTOMER_STATUS='Churned'				

Fig. 6. Plan for select for churn percentage across cities with index

This graph shows the churn percentage for each city, helping identify areas with higher customer loss. Cities with high churn may indicate service issues, pricing concerns, or strong competition, requiring targeted improvements. By analyzing these trends, the company can optimize retention strategies and enhance customer satisfaction in specific locations

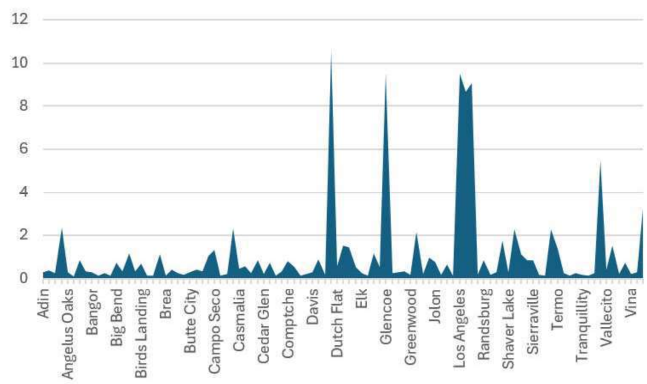

Fig. 7. Graf Churn percentage

This graph shows the churn percentage for each city, helping identify areas with higher customer loss. Cities with high churn may indicate service issues, pricing concerns, or strong competition, requiring targeted improvements. By analyzing these trends, the company can optimize retention strategies and enhance customer satisfaction in specific locations

B. Elderly which leave the company

For our target group, we focused on seniors, because they are often influenced by social recommendations, which can significantly impact their decisions to stay with or leave a service provider. By analyzing this specific demographic, we can develop more targeted strategies.

To detect older men who are retired and have churned, we need to perform filtering based on age, gender, and customer status. This query showed that the only option is to perform a full table scan (Fig. 8).

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			306	69
SORT		ORDER BY	306	69
SORT		GROUP BY NOSORT	306	69
TABLE ACCESS	CUSTOMERS	FULL	306	68
Filter Predicates				
AND				
CUSTOMER_STATUS='Churned'				
AGE>55				
GENDER='Male'				

Fig. 8. Plan for churned elderly

Since these columns have a low cardinality, we can significantly improve the query performance by creating a **bitmap index** on these columns (Fig. 9).

```
CREATE BITMAP INDEX idx_gender_status ON CUSTOMERS (Gender, Age, Customer_Status);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			306	6
SORT		ORDER BY	306	6
SORT		GROUP BY NOSORT	306	6
BITMAP CONVERSION		COUNT	306	5
BITMAP INDEX	IDX_GENDER_STATUS	FAST FULL SCAN		
Filter Predicates				
AND				
CUSTOMER_STATUS='Churned'				
AGE>55				
GENDER='Male'				

Fig. 9. Plan for churned elderly with bitmap index

We can see significant improvements when using a bitmap index. However, the optimizer used a fast full scan, meaning the entire table was scanned, but in a faster mode. To further optimize performance, we recommend reordering the columns in the index. To test this, we created a new index with a more logical column order (Fig. 9).

```
CREATE BITMAP INDEX ibx_gend_stat ON CUSTOMERS (Customer_Status, Gender, Age);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			306	3
SORT		ORDER BY	306	3
SORT		GROUP BY NOSORT	306	3
BITMAP CONVERSION		COUNT	306	2
BITMAP INDEX	IBX_GEND_STAT	RANGE SCAN		
Access Predicates				
AND				
CUSTOMER_STATUS='Churned'				
GENDER='Male'				
AGE>55				
Filter Predicates				
AND				
CUSTOMER_STATUS='Churned'				
GENDER='Male'				
AGE>55				

Fig. 10. Plan for churned elderly with optimized bitmap index

Now we can see that we have achieved an improvement, as the execution cost of the query is 50% lower (refer. Fig. 9 and Fig. 10).

C. *Rewarding loyal customers*

The company highly values its loyal customers. In an effort to recognize their commitment, has designed a reward system for those who meet specific criteria (Fig. 11).

Required criteria for top 10 most loyal customers are: Loyalty is defined by the number of people they have referred, their continued support by staying with us, and their high payment contributions.

1. **Referral Activity**: Customers who have referred others to the service are prioritized, demonstrating active support for the company.
2. **Be part of the top 3 fastest-growing cities**: Which means that in last quarter have the most of new Joined customers.
3. **Effectively Usage**: The rewards will focus on those who use more data and pay efficiently, meaning they consume a high amount of data while keeping their costs low.

```
WITH RankedCustomers AS (
        SELECT CUSTOMER_ID, CITY, NUMBER_OF_REFERRALS, MONTHLY_CHARGE / AVG_MONTHLY_GB_DOWNLOAD,
            SUM(NUMBER_OF_REFERRALS) OVER (PARTITION BY CITY
                ORDER BY NUMBER_OF_REFERRALS DESC, MONTHLY_CHARGE / AVG_MONTHLY_GB_DOWNLOAD  DESC) AS rn
         FROM customers
         WHERE customer_status = 'Stayed'),
    TopCustomers AS (
        SELECT * FROM RankedCustomers WHERE rn <= 10)

SELECT TopCustomers.CITY, COUNT(TopCustomers.CUSTOMER_ID) AS loyal_customers_count,
    COUNT(CASE WHEN c.customer_status = 'Joined' THEN 1 END) AS joined_customers_count,
    LISTAGG(TopCustomers.CUSTOMER_ID, ', ') WITHIN GROUP (ORDER BY TopCustomers.NUMBER_OF_REFERRALS DESC)
        AS top_10_customers_ids
    FROM TopCustomers
    join customers c on c.CITY = TopCustomers.CITY
    GROUP BY TopCustomers.CITY
    ORDER BY loyal_customers_count DESC, joined_customers_count DESC
    FETCH FIRST 3 ROWS ONLY;
```

Fig. 11. Select with analytical functions

Select used for this requirement consists of multiple Common Table Expression (CTE). In the first it is used ranking of customers based on their loyalty (referrals and charges). Sum over analytic function calculates the cumulative sum of NUMBER_OF_REFERRALS for customers within each CITY, ordered by their referral activity and average monthly data usage. Filter 'Stayed' include only those who have stayed with the company, ensuring we are looking at loyal customers (Fig. 11).

Another CTE filters out only the top 10 customers from each city, based on their cumulative number of referrals (ranked in the previous CTE).

The main query retrieves the cities with the most loyal customers, counting those who have referred the most people and those who have the "Joined" status. The LISTAGG function is necessary to identify the specific customers, who contributed to the loyalty metrics in each city. By creating a comma-separated list of customer IDs for the top 10 loyal customers, it allows the company to see exactly those individuals, who were responsible for driving the highest number of referrals. This makes it possible to reward these top contributors with the promised rewards, ensuring that the company can recognize and incentivize the customers who have had the greatest impact on its growth.

	CITY	LOYAL...	JOINED...	TOP_10_CUSTOMERS_IDS
1	San Diego	285	7	3327-YBAKM, 3327-YBAKM, 3327-YBAKM, 3327-YBAKM,
2	Covina	165	11	4282-ACRXS, 4282-ACRXS, 4282-ACRXS, 4282-ACRXS,
3	Newport Beach	160	10	5791-KAJFD, 5791-KAJFD, 5791-KAJFD, 5791-KAJFD,

Fig. 12. Output of select loyalty of customers

D. *Fastest-growing cities based on new customers (Quarterly growth rate)*

This query is designed to identify the fastest-growing cities based on new customer acquisitions over the last quarter. It does so by counting customers who joined within the last 3 months and ranking cities by the number of new customers (Fig. 13).

```
WITH CityGrowth AS (
    SELECT c.CITY, p.POPULATION,
           COUNT(CASE WHEN c.TENURE_IN_MONTHS BETWEEN
                        (SELECT MAX(TENURE_IN_MONTHS) - 3 FROM CUSTOMERS)
                    AND (SELECT MAX(TENURE_IN_MONTHS) FROM CUSTOMERS)
                THEN 1 END) AS new_customers
    FROM CUSTOMERS c
    JOIN ZIP_CODE_POPULATION p ON c.ZIP_CODE = p.ZIP_CODE
    GROUP BY c.CITY, p.POPULATION
)
SELECT CITY, POPULATION, new_customers,
       RANK() OVER (ORDER BY new_customers DESC) AS growth_rank
FROM CityGrowth
FETCH FIRST 3 ROWS ONLY;
```

Fig. 13. Select Fastest-growing cities based on new customers

City Growth - This part calculates the number of new customers in each city by checking if their TENURE_IN_MONTHS (how long they've been a customer) falls within the last 3 months. It also joins the ZIP_CODE_POPULATION table to include population data for each city.

The inner SELECT MAX(TENURE_IN_MONTHS) query finds the most recent tenure (e.g., the most recently joined customer).

Then, MAX(TENURE_IN_MONTHS) - 3 calculates the tenure threshold for the last 3 months.

The COUNT(CASE WHEN ...) counts how many customers joined within this timeframe.

This query is help full in expansion planning, comparing new customer growth with city size - Do small or large cities grow faster? And it aids in resource allocation - More investment in fast-growing areas.

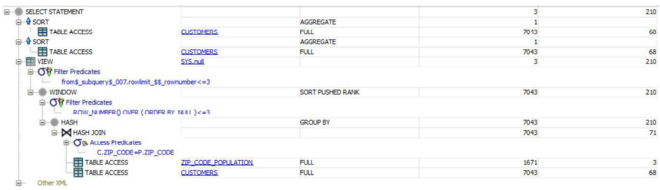

Fig. 14. Plan for Fastest-growing cities based on new customers without index

```
CREATE INDEX idx_tenure_city_zip ON CUSTOMERS (TENURE_IN_MONTHS, CITY, ZIP_CODE);
```

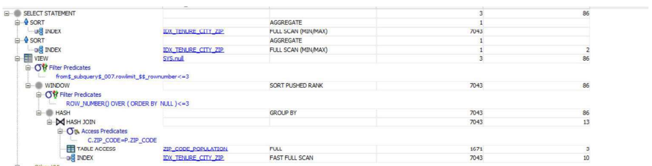

Fig. 15. Plan for Fastest-growing cities based on new customers with index

Second plan (with index) uses a FAST FULL INDEX SCAN on the IDX_TENURE_CITY_ZIP index, allowing for faster data access and reducing disk reads. Data is retrieved more efficiently using the index, reducing the number of rows that need to be read and processed. The query execution time in the second case is expected to be significantly faster compared to the first case.

E. Monthly spending trends for churned vs. active customers with city size

This SQL query analyzes customer spending behavior over time based on their customer status. It does so by:

Joining the CUSTOMERS and POPULATION tables on ZIP_CODE to include population data.

Calculating the trend of average spending (MONTHLY_CHARGE) for each customer status group, ordered by tenure in months.

Using a window function (AVG() OVER (...)) to compute the average spending dynamically as tenure increases.

```
SELECT c.CUSTOMER_STATUS,
       c.TENURE_IN_MONTHS,
       p.POPULATION,
       AVG(c.MONTHLY CHARGE) OVER (PARTITION BY c.CUSTOMER STATUS ORDER BY c.TENURE IN MONTHS) AS avg spending trend
FROM CUSTOMERS c
JOIN ZIP_CODE_POPULATION p ON c.ZIP_CODE = p.ZIP_CODE;
```

Fig. 16. Select Monthly spending trends for churned vs. active customers with city size

This query is useful for discovering how does average spending change over time based on customer status. Example: Do longer-tenure customers spend more or less over time?

If spending drops for long-term customers, the company may need to introduce loyalty programs. If new customers spend more, it may indicate a pricing strategy shift. Linking POPULATION data with spending trends could reveal how city size/population affects customer loyalty and revenue.

Fig. 17. Plan Monthly spending trends for churned vs. active customers with city size without index

```
CREATE INDEX idx_status_tenure ON CUSTOMERS (CUSTOMER_STATUS, TENURE_IN_MONTHS, ZIP_CODE);
```

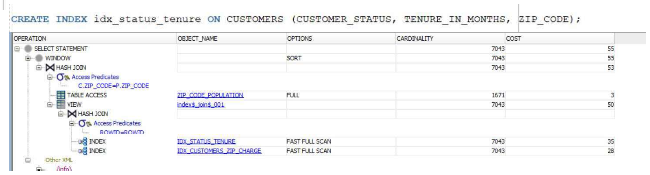

Fig. 18. Monthly spending trends for churned vs. active customers with city size with index

The indexed version is better because it lowers query cost, avoids full table scans, improves join efficiency, and speeds up filtering.

Number of CUSTOMER_STATUS

This chart displays the number of customers per CUSTOMER_STATUS. Categories are:

- Churned: Customers who left.
- Joined: New customers.
- Stayed: Customers who remained.

Most customers belong to the Stayed category, meaning customer retention is high. A smaller number of customers Churned. The Joined category is the smallest, indicating fewer new customers compared to the retained and lost ones.

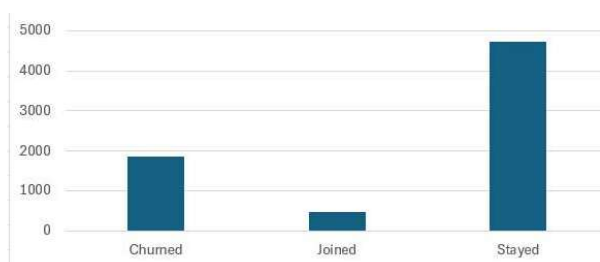

Fig. 19. Customer status

F. Customer Churn Risk Analysis by Zip Code

This SQL query is designed to analyze the churn risk of customers based on their monthly charges in comparison to the average charges within the same zip code. By identifying customers whose monthly charges are significantly lower than the average for their area, we can determine who may be at a higher risk of churning.

We are using hash join which are efficient when working with large tables.

```
SELECT /*+ USE_HASH(c p) */
    c.customer_id,
    c.zip_code,
    p.zip_code,
    c.tenure_in_months,
    c.monthly_charge,
    AVG(c.monthly_charge) OVER (PARTITION BY c.zip_code) AS avg_zip_code_charge,
    CASE
        WHEN c.monthly_charge < 0.8 * AVG(c.monthly_charge) OVER (PARTITION BY c.zip_code)
        THEN 'High Risk of Churn'
        ELSE 'Stable'
    END AS churn_risk
FROM customers c
JOIN zip_code_population p
ON c.zip_code = p.zip_code;
```

Fig. 20. Select Customer Churn Risk Analysis by Zip Code

This query is useful because it helps identify customers at risk of leaving due to significantly lower spending, allows comparison of customer spending patterns across different zip codes, companies can offer incentives or discounts to high-risk customers to reduce churn, helps businesses optimize pricing strategies based on regional spending behavior.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			7043	70
WINDOW		SORT	7043	70
TABLE ACCESS	CUSTOMERS	FULL	7043	68
Filter Predicates				
C.ZIP_CODE IS NOT NULL				

Fig. 21 Plan Customer Churn Risk Analysis by Zip Code without index

```
CREATE INDEX idx_customers_zip_hash ON customers (ORA_HASH(zip_code));

CREATE INDEX idx_customers_zip_charge ON customers(zip_code, monthly_charge);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
CREATE INDEX STATEMENT			7043	72
INDEX BUILD	IDX_CUSTOMERS_ZIP_HASH	NON UNIQUE		
SORT		CREATE INDEX	7043	
TABLE ACCESS	CUSTOMERS	FULL	7043	68
Other XML				
{info}				

Fig. 22 Plan Customer Churn Risk Analysis by Zip Code with index

Without index the query performs a FULL TABLE ACCESS on the CUSTOMERS table, meaning it scans the entire table to find the relevant data. It applies a filter predicate (C.ZIP_CODE IS NOT NULL), which can be inefficient if the table is large. The SORT operation suggests that SQL had to manually sort the results, increasing query cost. Overall query cost = 70, indicating a relatively high workload.

After creating an index a non-unique hash index (IDX_CUSTOMERS_ZIP_HASH) is created on the ZIP_CODE column. Future queries filtering by ZIP_CODE could use it instead of scanning the entire table. If the query utilizes the index, it will avoid FULL TABLE ACCESS, leading to faster lookups.

G. Customer Revenue and Retention Analysis by City

This query analyzes customer retention and revenue across different cities by evaluating total revenue, average monthly charges, data usage, and tenure distribution. It helps identify cities with high-value customers and retention trends.

```
WITH CustomerUsage AS (
    SELECT
        c.CUSTOMER_ID,
        c.CITY,
        c.ZIP_CODE,
        c.MONTHLY_CHARGE,
        c.TENURE_IN_MONTHS,
        c.TOTAL_CHARGES,
        c.AVG_MONTHLY_GB_DOWNLOAD,
        p.POPULATION,
        COUNT(*) OVER (PARTITION BY c.ZIP_CODE) AS customers_in_zip,
        RANK() OVER (PARTITION BY c.CITY ORDER BY c.TOTAL_CHARGES DESC) AS revenue_rank
    FROM CUSTOMERS c
    JOIN ZIP_CODE_POPULATION p ON c.ZIP_CODE = p.ZIP_CODE
    WHERE c.CUSTOMER_STATUS = 'Stayed'
)
SELECT
    cu.CITY,
    COUNT(DISTINCT cu.CUSTOMER_ID) AS customer_count,
    SUM(cu.TOTAL_CHARGES) AS total_revenue,
    AVG(cu.MONTHLY_CHARGE) AS avg_monthly_charge,
    MAX(cu.AVG_MONTHLY_GB_DOWNLOAD) AS max_data_usage,
    COUNT(CASE WHEN cu.TENURE_IN_MONTHS < 6 THEN 1 END) AS new_customers,
    (COUNT(CASE WHEN cu.TENURE_IN_MONTHS > 24 THEN 1 END) * 100.0 / COUNT(*)) AS long_term_customer_percentage
FROM CustomerUsage cu
GROUP BY cu.CITY
ORDER BY total_revenue DESC;
```

Fig. 23. Select Customer Revenue and Retention Analysis by City

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1105	72
SORT		ORDER BY	1105	72
HASH		GROUP BY	1105	72
VIEW	SYS.VW_DAG_0		4720	70
HASH		GROUP BY	4720	70
TABLE ACCESS	CUSTOMERS	FULL	4720	68
Filter Predicates				
AND				
C.CUSTOMER_STATUS='Stayed'				
C.ZIP_CODE IS NOT NULL				
Other XML				

Fig. 24. Customer Revenue and Retention Analysis by City without index

```
CREATE INDEX idx_customers_zip ON CUSTOMERS(ZIP_CODE);
CREATE INDEX idx_customers_city ON CUSTOMERS(CITY);
CREATE INDEX idx_customers_status_tenure ON CUSTOMERS(CUSTOMER_STATUS, TENURE_IN_MONTHS);
CREATE INDEX idx_zip_population ON ZIP_CODE_POPULATION(ZIP_CODE);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
CREATE INDEX STATEMENT			7043	72
INDEX BUILD	IDX_CUSTOMERS_ZIP	NON UNIQUE		
SORT		CREATE INDEX	7043	
INDEX	IDX_CUSTOMERS_ZIP	FAST FULL SCAN		
Other XML				
{info}				

Fig. 25. Customer Revenue and Retention Analysis by City with index

The second plan is better because indexes eliminate the need for full table scans, reducing I/O operations, sorting and filtering are optimized, improving response time and lower query cost means faster execution and better performance.

V. CONCLUSION

This study highlights the crucial role of database indexing in optimizing query performance for telecom customer analysis. By implementing and evaluating various indexing strategies, we demonstrated their impact on improving data retrieval efficiency. Despite initial challenges in reducing query costs, targeted indexing on key columns we significantly enhanced execution times, minimizing full table scans and improving resource allocation.

Beyond technical improvements, the findings offer strategic insights for the telecom industry. Understanding churn patterns at a city level allows company to implement targeted retention strategies.

ACKNOWLEDGMENT

We are grateful for the opportunity to be part of the *Database Analytics and Performance* course, where we explored the optimization of query. Analyzing and improving database queries was a fascinating experience, allowing us to gain valuable insights that will be highly useful in our professional careers. Since database query optimization is essential in frequent data processing, the knowledge gained from this study will help us develop more efficient and scalable solutions in the future. We appreciate the chance to apply theoretical concepts to practical challenges and deepen our understanding of database performance tuning.

This paper was supported by the **VEGA 1/0192/24** project - *Developing and applying advanced techniques for efficient processing of large-scale data in the intelligent transport systems environment.*

REFERENCES

[1] R. Cenková and W. Steingartner, "Luxury in the Time of COVID-19," *2022 IEEE 16th International Scientific Conference on Informatics (Informatics)*, Poprad, Slovakia, 2022, pp. 49-54, doi: 10.1109/Informatics57926.2022.10083451.

[2] D. Qi, B. Chang and W. Li, "Big Data Analysis and Intelligent Decision Support System for Environmental Water Quality: Application of Artificial Intelligence in Water Environmental Protection," *2024 3rd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*, Bristol, United Kingdom, 2024, pp. 169-174, doi: 10.1109/AIARS63200.2024.00037.

[3] S. S. Das, N. Deka, N. Sinha, S. Dhar, D. Bhattacharjee and S. Gupta, "Environmental monitoring using sensor data fusion," *2012 International Conference on Radar, Communication and Computing (ICRCC)*, Tiruvannamalai, India, 2012, pp. 83-86, doi: 10.1109/ICRCC.2012.6450552.

[4] J. Dostál *et al.*, "Innovative Concept of STEAM Education at Primary Schools in the Czech Republic - Support for Implementation in School Practice," *2022 IEEE 16th International Scientific Conference on Informatics (Informatics)*, Poprad, Slovakia, 2022, pp. 60-66, doi: 10.1109/Informatics57926.2022.10083467.

[5] R. Greenwald, R. Stackowiak, and J. Stern, *"Oracle Essentials: Oracle Database 12c"*, O'Reilly Media, 2013.

[6] D. Kuhn, and T. Kyte, *"Oracle Database Transactions and Locking Revealed: Building High Performance Through Concurrency"*, Apress, 2020.

[7] D. Kuhn, and T. Kyte, *"Expert Oracle Database Architecture: Techniques and Solutions for High Performance and Productivity."* Apress, 2021.

[8] M. Kvet, *"Developing Robust Date and Time Oriented Applications in Oracle Cloud: A comprehensive guide to efficient Date and time management in Oracle Cloud"*, Packt Publishing, 2023, ISBN: 978-1804611869

[9] J. Lee, J. Kang, S. Son and H. -M. Oh, "Numerical Weather Data-Driven Sensor Data Generation for PV Digital Twins: A Hybrid Model Approach," in *IEEE Access*, vol. 13, pp. 5009-5022, 2025, doi: 10.1109/ACCESS.2025.3525659.

[10] S. Morris, *"Resilient Oracle PL/SQL"*, O´Reolly, 2023.

[11] E. Mozzafari and A. Seffah, "From Visualization to Visual Mining: Application to Environmental Data," *First International Conference on Advances in Computer-Human Interaction*, Sainte Luce, Martinique, France, 2008, pp. 143-148, doi: 10.1109/ACHI.2008.29.

[12] A. Nuijten, A. Barel, *"Modern Oracle Database Programming: Level Up Your Skill Set to Oracle's Latest and Most Powerful Features in SQL, PL/SQL, and JSON"*, Apress, 2023

[13] B. Rosenzweig and E. Rakhimov, *"Oracle PL/SQL by Example"*, Oracle Press, 2023.

[14] M. Tayab, W. Zhou, M. Zhao and S. Li, "Big data and public services for environmental monitoring system," *2016 11th International Conference on Computer Science & Education (ICCSE)*, Nagoya, Japan, 2016, pp. 139-143, doi: 10.1109/ICCSE.2016.7581569.

[15] H. Xue, "Dynamic Integration and Analysis of Marine Environmental Monitoring Data Based on Support Vector Machine," *2023 Asia-Europe Conference on Electronics, Data Processing and Informatics (ACEDPI)*, Prague, Czech Republic, 2023, pp. 54-57, doi: 10.1109/ACEDPI58926.2023.00017.

L. Yuan, "Research on Practical Value and Teaching Strategies of Computer-Aided Translation Teaching Based on POA Concept," *2022 IEEE 5th Eurasian Conference on Educational Innovation (ECEI)*, Taipei, Taiwan, 2022, pp. 252-256, doi: 10.1109/ECEI53102.2022.9829524.

[16] Y. Zhao, M. Guan, Z. Tao and L. Guo, "Research on Intelligent Optimization Decision Support Platform of Solid Waste Environmental Risk Event Disposal Technology Under the Background of Big Data," *2022 International Conference on Cloud Computing, Big Data and Internet of Things (3CBIT)*, Wuhan, China, 2022, pp. 88-92, doi: 10.1109/3CBIT57391.2022.00026.

[17] Erasmus+ project EverGreen dealing with the complex data analytics: https://evergreen.uniza.sk/