Managing Streaming Database System for Education by Using Partial Replication Approach to Update the Database

Ammar Thaher Yaseen Al Abd Alazeez
Department of Computer Science, College of Computer Science and Mathematics, University of Mosul, Mosul – Iraq.

Abstract

Most classical SQL databases store information that does not change frequently. Models are client relationship management applications or site applications that update like clockwork/seconds. Up to this point, software engineers needed to compose sophisticated code to make databases work with constant changing or streaming information. Creating grate micro services to deal with stream information are normal and frequently delicate. A streaming database works with dynamic information that changes repeatedly. This changing information triggers activities in the database inquiries and applications rather than the reverse way around. The main characteristics of streaming databases are that continuous insert new data records, update the database status in real time, and answer online queries with consistency database state. This research follows a streaming database technique for managing stream educational information for lecturers in Computer Science Department supporting it with the lot occasions simultaneously. This research applies a partial replication approach that update the database periodically depending on the updating triggers. The replication accompanied by a change in the number of triggers indicates a significant education event. Experiments led the belief that the streaming database structures are extra appropriate for utilize educational information in universities. This is come from the fact that the traditional database system spent about 5 minutes for updating 10 rows in two different computers, while in suggested streaming database system we spent only about 4 minutes. This kind of online updating and its results can be used in education systems in various ways.

Keywords: Database, Streaming Database, Data Mining, Big Data.

Received: 2/8/2021 Accepted: 6/4/2022

نظام ادارة قواعد البيانات المتدفقة للتعليم باستخدام طريقة الاستنساخ الجزئي لتحديث قاعدة البيانات

عمار ظاهر ياسين ال عبدالعزيز

قسم علم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل، موصل - العراق.

المستخلص

معظم قواعد البيانات التقليدية SQL تخزن بيانات والتي غالباً لا تتغير كثيراً. الامثلة عليها تشمل تطبيقات ادارة معلومات الزبائن او تطبيقات صفحات الانترنيت والتي تحدث معلوماتها كل بضع ثوان او دقائق. الى وقت قريب، تمكن مبرمجي ومطوري قواعد البيانات من كتابة شفرات برمجية مكثفة لجعل قواعد البيانات تتعامل مع التغييرات المتتالية او البيانات المتدفقة. تطوير الخدمات المصغرة micro services المكثفة لمعالجة استمرار البيانات المتدفقة هو امر مكلف وغالبا ما يكون غير مستقر. قاعدة البيانات المتدفقة تعمل مع البيانات الفعالة والتي تتغير بشكل مستمر. هذه البيانات المتغيرة بشكل مستمر تعمل كقوادح لعمليات معينة في استعلامات قاعدة البيانات وتطبيقاتها بدلا من الطريقة التقليدية. الخصائص الرئيسة لقواعد البيانات المتدفقة هي الاستمرار في ادخال القيود الجديدة، تحديث حالة قاعدة البيانات في الوقت الحقيقي، والاجابة بشكل مباشر على الاستعلامات مع استقرار لحالة قاعدة البيانات. هذا البحث يتبع تقنيات قواعد البيانات المتدفقة في ادارة قواعد بيانات التعليم لاساتذة قسم علوم الحاسوب بتجهيزها بالمعلومات الانية. هذا البحث يطبق تقنية الاستنساخ الجزئي لتحديث قواعد البيانات المتدفقة بشكل دوري بالاعتماد على قيمة حد العتبة والتي تستخدم كقادح لعملية التحديث. الاستنساخ مصحوب بتغيير عدد من القوادح والتي تشير الى التغييرات التعليمية المهمة. التجارب قادت الى الاعتقاد بان قواعد البيانات المتدفقة هي فرص اضافية لاستغلال المعلومات التعليمية في الجامعات. هذا اتى من الحقيقة ان نظام قواعد البيانات التقليدية تستهلك حوالي ٥ دقائق لتحديث ١٠ قيود في حاسوبين مختلفين، بينما قواعد البيانات المتدفقة المقترحة تستهلك حوالي ٤ دقائق فقط. هذا النوع من التحديث الآني ونتائجه يمكن ان يستخدم في انظمة التعليم المعتمدة بمختلف الطرائق.

الكلمات المفتاحية: قواعد البيانات، قواعد البيانات المتدفقة، التنقيب عن البيانات، البيانات الضخمة.

Introduction

Database systems are a collection of related data and accepted for granted. Quite possibly the best programming tools at any point composed, they are known the innovation business. They have advanced over years with information warehousing, NoSQL (do not follow the restrictions of traditional relational database system), big data, and so on. However, every one of these changes, databases actually look and feel extensively like the databases of 50 years ago: static information to which you direct inquiries and anticipate that answers should return. Their latest development, is very diverse[1].

So in streaming event, the trigger is not you. The trigger is the information. The trigger is the appearance of an event. The information does not sit latently, as it does in a conventional relational database; it is consistently on the run[2].

Dynamic database techniques are the issue from the contrary side, permitting you to make triggers and emerged instances that react to changes made to a database table. These have been around for years; however the usefulness is restricted to the limits of the database itself: no floods of events are come to the rest of the world. Later database innovations - like MongoDB, Couch base and Rethink DB (examples of NoSQL database systems) - have consolidated the thought of tables that transmit stream events when records change. Applications would then be able to respond to these streams [3].

While the two methodologies - dynamic databases and streaming database events - may have all the assigns of being comparative, they are utilized in various way. Both expand on a similar three structure blocks: inquiries, tables, and event streams. Dynamic databases are better at questions over tables, yet they cannot inquiry stream events. Streaming database event can question both stream events and tables; however the table-

based inquiries are less good in class than their dynamic database partners.

Following fifty years, the database is transforming to the requirements of another expert. This transformation is not finished. In reality, it is likely just start. However, one thing appears to be clear: the normal response to the inquiry "What is database system?" is probably going to change, as the product world turns out to be less about programming that assists individuals with preferring tackle our job and more about the chains of programming that completely mechanize the universe[3].

Traditional Database vs. Streaming Database

With a relational database management system, also called a RDBMS, a manager loads information at a pre-order recurrence relying upon their prerequisites. With a streaming database, then again, information are gathered, prepared, and enhanced continuously - typically just after the actual information are made[4].

Streaming databases can be any database that is arranged to handle streaming information continuously. This can incorporate time-series databases, in-memory data grids, and many others [4]. Another significant use case for streaming data has to do with how it empowers you to set live alarms and notices for significant events across your business. By setting up constant alarms for those restrictive changes that matter, you can get that they have happened very quickly. You at this point do not need to stand by weeks or even a long time to discover that these progressions happened. Cautions are produced naturally so individuals who need this data likely take care of their responsibilities they will have it[5].

Lastly, streaming databases are a critical advantage when active applications that falls inside the microservices structure.

With micro services, you are planning an application as a progression of administrations that actually cooperate to frame whole system. This is as opposed to the classical utilization of old day's database systems. Figure 1 shows the primary contrast among traditional and streaming databases[4].

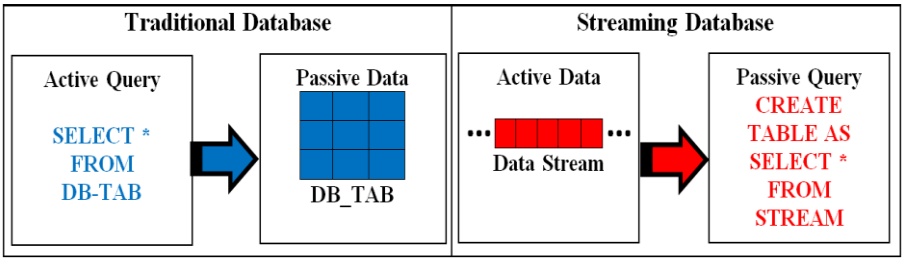

Figure 1: Differences between traditional and streaming databases

Related Works

Streaming Database (SDB) is a type of streaming information intended to convey constant knowledge that can improve business seriousness. Streaming data includes constant handling of information from large number of sources like sensors, financial exchanging, web-based business, web applications, social network (like Facebook, Twitter, etc.) and some more. By accumulating and processing these on-going streams information, ventures can utilize stream database systems to create knowledge to improve correctness, settle on better-educated choices, finish tasks, improve client assistance and act rapidly to make the most of business opportunity[6][7].

Dynamic streaming database requires a complex streaming engineering and Big Data arrangement like Hadoop and Apache Kafka. Kafka is a quick, multidisciplinary and strong distribute tool in informing framework that can uphold stream data preparing by working on information treatment. Kafka can

measure and execute in excess of 100,000 exchanges each second and is an ideal tool for empowering streaming database to help Big Data investigation and information lake activities[6].

However, utilizing Kafka for streaming database can make an assortment of difficulties also. Source frameworks might be unfavorably affected. A lot of custom advancement might be required. Also, scaling proficiently to help an enormous number of information sources might be troublesome [8]. That is the place where our suggested partial replication could work properly. In other words, we used the partial replication technique for updating our streaming educational database rather than using the complexity of Kafka streaming approach (see Section 9 for more information).

Data Streams and Internet of Things (IoT)

The Internet of Things (IoT) is all over the places around us, and the information is accumulating. The new small, embeddable PCs are empowering managers and specialists with the chance of utilizing the entirety of this information to control everything from industrial to individual homes. These named streaming databases which are devices intended to deal with both the continue approaching streaming data and unlimited inquiries from clients that need to settle on choices dependent on the data. Streaming databases are close to other new classes of program tools like time-series databases or log databases. All are intended to follow a progression of events and empower inquiries that can look and deliver measurable profiles time. The streaming databases can react to inquiries for information and furthermore measurements about the data, create reports from these questions, and populate the whole dashboards that track what is going on to permit the clients to settle on smart choices about the telemetry. Some streaming databases are intended to decrease the size of the data to save memory costs. They can by change a

value gathered each second with a mean calculated all the day[12].

Streaming Database

A data stream is approximately considered as persistent orders of requested information produced as identified events continuously. Instead of static information, the stream information is dynamic as new information continually arrive, and the obsolete information records are continually dropped off from the dataset [9]. In formal, a data stream *DS* is addressed as a *d*-dimensional limitless of records $S_1, S_2, \dots, S_n, \dots$, generated at a time point $t_1, t_2, \dots, t_n, \dots$, respectively, i.e.:

$$DS = < S_1, t_1 >, < S_2, t_2 >, \dots, < S_n, t_n >, \dots$$
$$= (s_{11}, s_{12}, \dots, s_{1d}, t_1), (s_{21}, s_{22}, \dots, s_{2d}, t_2), \dots, (s_{n1}, s_{n2}, \dots, s_{nd}, t_n), \dots$$

Instances of data streams incorporate streamed media data like online news, videos, Internet pages, phone records, and sensor data network [10].

The fundamental qualities of the data streams include [10]:

- New data arrive frequently at various paces.
- Existing data may be getting old and eliminated after they are processed.
- The size of data streams is huge and unbounded.
- The data cycle may not be known and non-fixed, for example its distribution may change after some time.
- Although there is a period of arriving data streams, there is no control which data record will process first.

The unique idea of data streams may identify the following[10]:

- The hidden examples behind static data continue as before while those behind stream data do change. This implies the examples in the static data can be found through one scan, while designs found with a more established form of the stream data should be refreshed considering the new data change.
- It is too expensive to even think about discarding existing examples mined from the stream data before and re-find the new examples, especially when the speed of data change is quick. In this manner, answers for stream data mining should be gradual as in the current examples are adjusted properly to impose the progressions to the data.

Streaming database is comprehensively characterized as a data store intended to gather, measure, as well as improve an approaching arrangement of data records (i.e., a data stream) continuously; commonly following the information made. This term does not belong to a specific class of database administration frameworks, but instead, applies to a few kinds of databases that handle streaming data progressively, remembering for network data, in-memory databases, NewSQL databases, NoSQL databases, and time-series databases. A streaming database gathers streaming data continuously for immediate processing or batch processing [11] (see Figure 2).

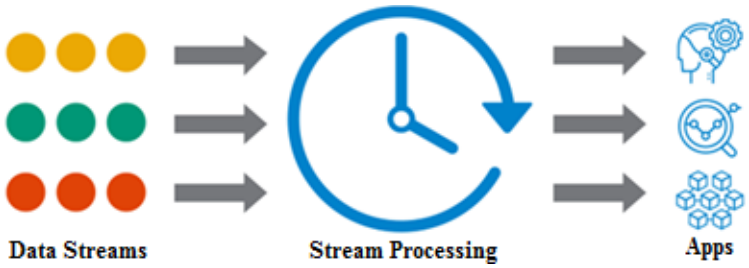

Figure 2: Streaming databases processing

Streaming database is opposed to conventional database administration frameworks (RDBMSs), in which a database manager would normally stack information through an ETL (extract, transform, and load) tool/process at ordinary periods like daily or week after week. A streaming database may be close to RDBMSs for present day use cases in bigger companies. As the volume of information proceeds to develop and the speed of information keeps on speeding up, a few advancements that once depended fundamentally on block-directed databases presently more intensely streaming database innovations (e.g., recommendation search-engines) [11].

The surge of streaming database use can be credited to the expanded utilization of Internet of Things (IoT) applications, automated works, and the utilization of constant investigation for on-going dynamic data. Business are moving quickly, and managers need to settle on choices dependent on updating information instead of utilizing past information put away in static databases. Main retailers, for example, Amazon and eBay investigate client transactions data streams to quantify good sell items and eliminate inactive items from online customers. Streaming databases can displace complex microservices, decreasing the time and cost to create applications. Item supervisors can expand their items an ideal opportunity to showcase by dissecting results and test data continuously. Use cases length businesses and specialized applications. For example, in the drug and biomedical areas, researchers utilize these databases to display results from large scope clinical examinations improving medication feasibility and submit times. By utilizing a streaming database, data clients can pose inquiries and get brings about continuous even as the fundamental data changes. There are various use cases for this innovation [1].

As expressed, probably the greatest benefit of streaming databases is that they take into consideration both the speed and

the adaptability required for projects where groups should have the option to settle on more educated decisions quicker and more productively than any other time in recent memory. This is useful in circumstances like fraud identification where consistently checks [4].

Streaming data is not the optimal for systems that are work with enormous data collections, however, that rely upon more profound degree of investigation. Streaming databases may likewise not be suitable with the sort of inheritance frameworks that typically just help more customary ways to deal with databases [4].

Technical and Business Use Cases of Streams Database

1. Technical Use Cases of Streams Database

Technologists are embracing streaming databases for an assortment of utilization cases. These incorporate [1][11]:

- Stream data advancement. One significant use case for streaming databases is saving data that can enhance streaming data. Since real-time data, particularly from IoT sources, is quite often moderate, getting that data together with reference data from a streaming database can give more setting to analysis.
- Real-time event catch and preparing. Numerous organizations need to become event driven, and streaming databases can help IT groups arrive while regularly giving a portion of similar advantages as conventional databases, for example, the capacity to associate with SQL-like programming languages.
- Microservices designs. Streaming databases can move data from reason constructed application to reason assembled application progressively, so they can fill in as the spine for sharing data and data in microservices models, which are getting more normal.

- Stream preparing. A large part of the data that individuals, applications, and machines make today is produced as a progression of on-going events. Streaming databases can execute continue questions to deal with these events as they happen instead of inactive clusters of simple data.

2. Business Use Cases of Streams Database

There are numerous reasons why business groups are urging their IT programmers to receive/process streaming databases. That is because business groups see streaming databases can empower them to [1][11][13]:

- Respond to events quicker than others.
- Enable continuous alarming for market changes.
- Support preventive continuous use cases.
- Analyse data continuously as it is created.
- Deploy continuous AI derivation batches data.

Replication Method of Streaming Database

Replication is a technique that control updating online and streaming databases. Generally, there are three types of replication. No replication, when there is no updating of database when there are changes happened to the database. Partial replication, when only the updating parts of database are modified on other parts. Full replication, when whole database update during a specific period [14]. Figure 3 explains the replication technique of database in two different sites (in Britain and Spain). The replication technique in this business system works in real-time approach for updating the database that distributed in different sites. The goal of the replication is to minimize transmission, improve performance, and support heavy multi-user access.

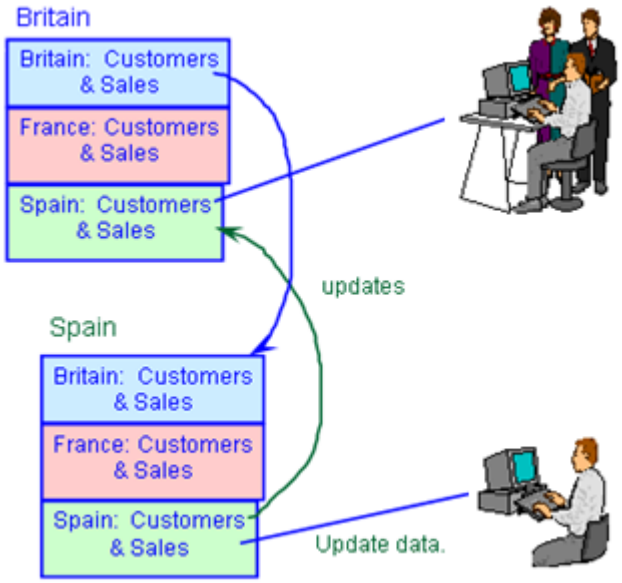

Figure 3: Replication processing in database system

Research Methodology

The research methodology of this paper divided into two parts: research problem and research importance.

Research problem

This research paper aims to address the problem of updating streaming educational database system periodically. From this main problem, a number of questions are splitted:

1. Are the update information saved correctly?
2. Is there a guarantee that the same information will be given at the same time and without repetition?
3. Are there accurate statistics of satisfying queries?

Research Importance

The importance of the research lies in addressing the questions in the study problem, by designing a database system for the lecturers in order to achieve the following:

1. Ensure that data is saved and retrieved directly, quickly and accurately when needed.
2. Ensure that the information are not repeated for one query.

MSDBE: Streaming Database System for Education

Generally, there are four main steps of database systems: Analysis step, Design step, Implementation step, and Evaluation step. The following will explain the steps of suggested system:

1. Analysis Step: The analysis step consists of two stages; collect information about lecturers in Computer Science Department and re-arrange the information and save it in a database to be ready for the next step.
2. Design Step: This step will use one of the popular design tool named data flow diagram (DFD) to design the suggested system. Figure 4 shows the main steps of the DFD. The first step includes reading the request as sequence of queries from the user as streams. After that, select the correct table from the database. The third step, update the database. Finally, display the output on the screen.

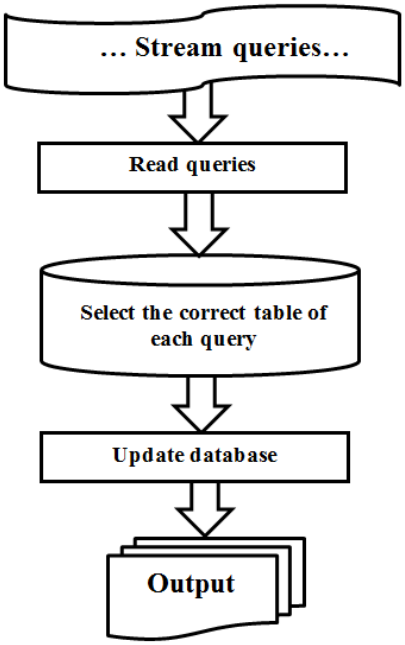

Figure 4: DFD of MSDBE

3. Implementation Step: This step includes two sub-steps. Firstly, create eight educational connected tables; LECTUERER (L_Id, L_Name, L_Degree, L_Addr, L_Specilist, L_Email, L_Phone), COOLEAGE(C_Id, C_Name, C_Addr), DEP(D_Id, D_Name, D_Addr), COURSE(CR_Id, CR_Name, Term), STUDENT(S_Id, S_Name, S_Stage), RESEARCH(R_Id, R_Name, R_Aouther), LIBRARY(L_Id, L_Name, L_Cat), LABROUTERY(LB_Id, LB_Name, L_No_Cmp).

   Secondly, prototype suggested replication method of educational streaming database. The idea of streaming database in this research depends on the partial replication technique. More precisely, after creates a database system for the Computer Science Department, first, identifies the threshold (*Thr*, e.g., 1 ml sec., 1 sec., 1 min., 1 hour, or update

trigger) that necessary to determine the updating period. Then, takes the current state of database and saves it as dump in the secondary storage. After that uploads the newest version of database snapshot and continues receiving new data stream. Figure 5 illustrates the pseudo code of the replication method.

Suggested Partial Replication Method

Method Steps:

1. Identify the value of *Thr* (in this research we choose *Thr*=1min)
2. Take a dump of the database.
   $SQL > MYDUMPER --host=123.123.123.123 --port=3306 --user=*** --password=**** -B log --trx-consistency-only --triggers --routines -o /SQL/new_db/SA.log
3. Stop all replication on the secondary storage.
   $SQL > STOP SLAVE for channels;
   {Get the current values from SQL.}
   $SQL > SHOW GLOBAL VARIABLES;
4. Load the dump of the database.
   $ myloader -d /sqldata/new_db/ -s new_db
5. Clear all existing variables values so that we can overwrite it to include the new dump file.
   $SQL > RESET VARIABLES;
   {Set new values, including values from the SQL dump file.}
   $SQL > SET GLOBAL;
6. Start replication for the new DB as follows:
   $SQL> MODIFY MAS TO MAS_H, MAS_USER='REP_USER',
   MAS_PAS='REP_PAS',MAS_AUTO_POS=1 for CHAN 'na_db';
   $SQL> START SLA of chan 'na_db';

{Start replication for the DB that paused.}

$SQL> START SLAVE for channel;

7. Repeat the collection steps 2-6 on the old primary (now secondary) every *Thr*.

Figure 5: Proposed replication method

4. Evaluation Step: To evaluate the suggested MSDBE method, it conducted a sample of 10 queries composed of 5 reads and 5 updates (for example, *SELECT * FROM LECTURER WHER L_Name = "Ali"* – and – *UPDATE COURSE SET CR_NAME = "DATABASE2"*) using three scenarios. The first scenario was to test the system with 5 sequential reads and 5 sequential updates (R→U). The second scenario was to test the system with the 5 sequential updates and 5 sequential reads (U→R). The third scenario was to test the system with the 5 reads and 5 updates (R↔U) by take one from each group repeatedly. After tested the system with the three scenarios for twenty times, we discovered that the suggested method (S) is better than traditional one (T). The following figure shows the performance test of the system in each scenario.

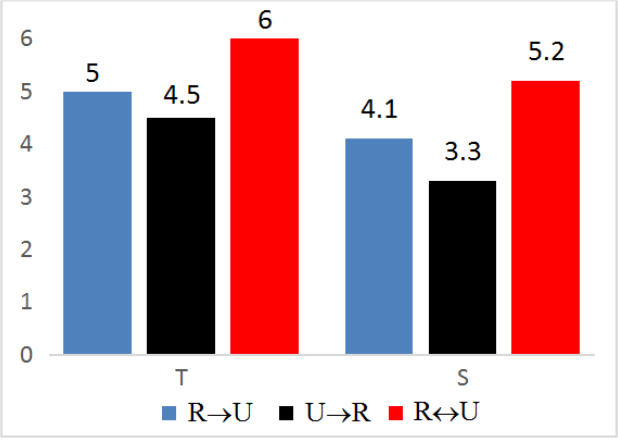

Figure 6: The performing test of three scenarios

Discussion

We created a traditional relational database for educational purpose. Means, we analyzed, designed, and implemented database system for the Computer Science Department. We used eight tables in this database. All tables are related and connected with each other's via keys. For managing the database we utilized Oracle as DBMS. We copied the system on two servers (computers) and connected them through computer network. We tested the traditional relational database system many times of the10 queries and take the average of execution time that utilized to finish all queries to check the efficiency of the system for repeating updates in each version/site of the system. After that, we added the technique that suggested for partial replication to the system and repeat the same test. As can see in Figure 6, the proposed method (S) in all scenarios is better than traditional way (T). We explored the easy and confident confirms of the updates in each version of the database system with high efficiency of suggested method.

Conclusion

The expanding utilization of streaming data events is deactivating data warehousing and using traditional SQL databases. The act of utilizing different microservices to join databases is being change by the utilization of streaming databases and basic SQL queries. These new programs permit designers to measure continually changing data and trigger events progressively with practically immediate idleness. Use cases range from monetary alarms to digital protection observing and on-going representations. Industrial utilizations incorporate checking IoT prototypes and AI methods. Product supervisors can abbreviate application advancement times by eliminating complex micro services with a couple of lines of SQL code. In

this research we showed that the replication procedure could help applying streaming database in educational databases. That is because the proposed partial replication method advanced traditional SQL method by at least one minute for each implementing queries.

The future works will focus on three folds: Before the improvement of streaming databases, software engineers expected to discover approaches to join various databases and cycle evolving data. The standard practice was to create sets of micro services to deal with the high-volume of complex directions needed to handle streaming data with traditional SQL databases. These databases eliminate the need to figure micro services shortening the improvement time and decreasing costs. Also, constant business knowledge devices access streaming databases to produce figures and logical designs to give on-going answering to basic business and specialized employments. Observing streaming data can make monetary financial investors aware of market changes and variances as they occur. Finally, using the triggered replication technique in distributed real-time environment.

References

[1] Open Data Science ODSC (2021), “Intro to Streaming Databases,” *Open Data Science*, [Online]. Available: https://medium.com/@ODSC/intro-to-streaming-databases-c238818babdc. [Accessed: 20-Apr-2021].

[2] A. Al Abd Alazeez, S. Jassim, and H. Du (2017), “EINCKM: An Enhanced Prototype-based Method for Clustering Evolving Data Streams in Big Data,” *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, no. Icpram, pp. 173–183.

[3] B. Stopford (2020), “The Rise of the Event Streaming Database,” *TheNewStack*, [Online]. Available: https://thenewstack.io/the-rise-of-the-event-streaming-database/. [Accessed: 15-Apr-2021].

[4] DEVINSIDER (2021), “The purpose and benefits of data stream processing,” *DELLTecnologyWorld*, [Online]. Available: https://www.dev-insider.de/what-is-a-streaming-database-a-1002982/.

[5] A. T. Y. T. A. A. Alazeez (2020), “HPPD: A Hybrid Parallel Framework of Partition-based and Density-based Clustering Algorithms in Data Streams Ammar,” *Raf. J. Comp. Math’s.*, vol. 1, no. 1, pp. 67–82.

[6] QlikQ, “DATABASE STREAMING,” *QlikQ*, 2021. [Online]. Available: https://www.qlik.com/us/data-streaming/database-streaming. [Accessed: 13-Mar-2021].

[7] A. T. Y. T. A. A. Alazeez (2021), “DED: Drift Principle in Educational Evolved Data,” *Tikrit J. Pure Sci.*, vol. 26, no. 2, pp. 118–125.

[8] A. T. Y. A. A. A. Al Abd Alazeez (2021), “AEPRD: An Enhanced Algorithm for Predicting Results of Orthodontic Operations,” *J. Educ. Sci.*, vol. 30, no. 1, pp. 173–190.

[9] A. T. Y. Al Abd Alazeez, S. Jassim, and H. Du (2018), “SLDPC : Towards Second Order Learning for Detecting

Persistent Clusters in Data Streams," *2018 10th Comput. Sci. Electron. Eng.*, vol. 978-1–5386, pp. 248–253.

[10] A. Al Abd Alazeez, S. Jassim, and H. Du (2018), "TPICDS: A Two-phase Parallel Approach for Incremental Clustering of Data Streams," *24th Int. Eur. Conf. Parallel Distrib. Comput.*.

[11] Hazelcast (2021), "What Is a Streaming Database?," *Hazelcast*,. [Online], Available: https://hazelcast.com/glossary/streaming-database/. [Accessed: 12-Apr-2021].

[12] P. Wayner (2021), "Vision for future of computing," *Venture Beat*, [Online]. Available: https://venturebeat.com/2021/04/04/what-is-a-streaming-database/.

[13] A. Al Abd Alazeez, S. Jassim, and H. Du (2017), "EDDS: An Enhanced Density-Based Method for Clustering Data Streams," *2017 46th Int. Conf. Parallel Process. Work.*, pp. 103–112.

[14] C. Riccomini (2017), "Streaming databases in realtime with MySQL, Debezium, and Kafka," *WePay Chase Company*, [Online], Available: https://wecode.wepay.com/posts/streaming-databases-in-realtime-with-mysql-debezium-kafka. [Accessed: 01-Mar-2021].