

Analiză soluții baze de date pentru aplicația E-SAFETY DRIVING APPLICATION (ESDA)

Abstract: *Implementarea unui sistem software implică de obicei și alegerea unui sistem de gestionare a bazelor de date. Această alegere necesită decizia de a folosi un sistem de gestionare a bazelor de date NoSQL sau un sistem de gestionare a bazelor de date SQL. Peterson (2022) identifică o serie parametri de diferențiere între cele două tipuri de sisteme de gestionare și îi rezumă la cinci diferențe majore. Sharma & Dave (2012) analizează comparativ cele două tipuri și identifică o serie de zece trăsături care disting sistemele de gestionare a bazelor de date NoSQL de sistemele de gestionare a bazelor de date SQL. Un proces asemănător este desfășurat și de cei de la Talend (2021) care identifică cinci diferențe practice în ceea ce privește diferențele dintre cele două tipuri de sisteme de gestionare a bazelor de date. SQL este abrevierea de la „Structured Query Language” și indică o serie de comenzi de programare folosite pentru stocarea și gestionarea datelor prin intermediul unui sistem relațional (RDMS). Reiese de aici că termenul NoSQL desemnează posibilitatea de stocare și gestionare a datelor prin alte mijloace decât cele de tip SQL.*

Sistemul RDMS, care stă la baza SQL, presupune stocarea datelor în obiecte numite tabele, definite în termeni de coloane și rânduri. Astfel, datele sunt stocate într-un format tabelar care indică relațiile dintre datele stocate. Mai mult, sistemul de tip RDMS permite ilustrarea explicită a relațiilor dintre tabele (folosind de exemplu „foreign key”). Un sistem de gestionare a bazelor de date SQL implică așadar metode de interogare a datelor stocate într-un format tabelar, format care indică relația dintre date. Pe de altă, parte, sistemele de gestionare NoSQL implică stocarea datelor și în alte formate decât cel tabular.

Pentru a înțelege diferența dintre cele două tipuri de sisteme de gestionare a datelor, cei de la MongoDB (2021) oferă un exemplu ilustrativ. În acest sens, aceștia propun următorul scenariu de stocare de date care poate fi abordat atât prin intermediul SQL cât și prin intermediul NoSQL. Scenariul presupune colectarea datelor despre numele, numărul de telefon, adresa și hobby-urile unor clienți. Putem observa că în timp ce la primele trei variabile există o singură posibilitate de răspuns, ultima variabilă permite un răspuns cu mai multe elemente. Astfel, reprezentarea acestor date într-un RDMS va presupune crearea a două tabele ce prezintă conexiuni între ele.

Users				
ID	first_name	last_name	cell	city
1	Leslie	Yepp	8125552344	Pawnee

Hobbies		
ID	user_id	hobby
10	1	scrapbooking
11	1	eating waffles
12	1	working

Reprezentare de tip RDMS

Dat fiind că variabila asociată hobby-urilor putea conține mai mult decât un singur răspuns per client, reprezentarea acestora într-o singură celulă de tabel ar fi fost imposibilă.. Astfel, a fost decisă crearea a două tabele. Pe baza acestei reprezentări, baza de date poate fi interogată prin comenzi de tip SQL. În acest sens, obținerea informațiilor despre clienți și hobby-urile acestora necesită combinarea tabelor din imaginea reprezentată în figura de mai sus. Pe de altă parte, reprezentarea într-un sistem de gestionare al bazelor de date NoSQL poate lua forma prezentată în figura de mai jos.

```
{
  "_id": 1,
  "first_name": "Leslie",
  "last_name": "Yepp",
  "cell": "8125552344",
  "city": "Pawnee",
  "hobbies": ["scrapbooking", "eating waffles", "working"]
}
```

Reprezentare de tip NoSQL

Așa cum se poate observa, în cazul NoSQL, atât datele despre clienți cât și datele despre hobby-urile acestora sunt stocate într-un singur obiect, oferind astfel posibilitatea de interogare mai rapidă decât în cazul sistemului RDMS, care presupune combinarea a două obiecte/tabele. Reies de aici primele diferențe dintre sistemele de gestionare a bazelor de date SQL și sistemele de gestionare a bazelor de date NoSQL. Este vorba în primul rând de reprezentarea datelor.

În cazul sistemelor de gestionare a bazelor de date SQL, datele sunt reprezentate în formă tabulară. În cazul sistemelor de gestionare a bazelor de date NoSQL, datele pot fi reprezentate și în altă formă decât cea tabulară, în funcție de sistemul ales. În cazul exemplului portretizat mai sus, a fost ales un sistem de gestionare de tip document. Detalii despre acest tip precum și alte tipuri de sisteme NoSQL sunt prezentate în secțiunea următoare.

Alături de diferențe în ceea ce privește interogarea bazei de date, sistemele de gestionare

a bazelor de date NoSQL și sistemele de gestionare a bazelor de date SQL prezintă diferențe ce țin de limbajul de interogare al bazei de date și scalabilitatea sistemului de gestionare a datelor (Smallcombe, 2021). Sistemele de gestionare a bazelor de date SQL au o schemă de interogare fixă în timp ce sistemele de gestionare a bazelor de date NoSQL au o schemă de interogare dinamică. Acest aspect permite sistemelor de gestionare a bazelor de date NoSQL să ofere posibilitatea de a defini modelul de date prin raportare la fluxurile de lucru, în timp ce sistemele de tip SQL necesită eforturi de integrare a unui model de date predefinit în fluxurile de lucru.

De asemenea, sistemele de gestionare a bazelor de date SQL sunt optimizate pentru scalabilitate din punct de vedere vertical, în timp ce sistemele de gestionare a bazelor de date NoSQL sunt optimizate pentru scalabilitate din punct de vedere orizontal. În vederea definirii termenilor, cei de la CloudZero (2021) detaliază diferența dintre scalabilitatea orizontală și verticală. Scalabilitatea din punct de vedere vertical se referă la creșterea capacității de procesare a unui sistem în vederea gestionării afluxului de trafic prin intermediul adăugării unor noi noduri (servere) de procesare. Scalabilitatea din punct de vedere vertical presupune creșterea capacității de procesare a unui sistem în vederea gestionării afluxului de trafic prin intermediul îmbunătățirii performanței unui nod (server) de procesare.

Referitor la sistemele de gestionare a bazelor de date, scalabilitatea acestora devine importantă în cazul multiplicării interogărilor către baza de date. Astfel, scalarea unui sistem de gestionare a bazelor de date SQL este optimizată pentru creșterea capacității pe verticală prin adăugarea de capacitate CPU, RAM sau SSD mașinii virtuale, în timp ce scalarea unui sistem de gestionare a bazelor de date NoSQL este optimizată pentru creșterea capacității pe orizontală, prin adăugarea mai multor mașini virtuale (Smallcombe, 2021).

În ceea ce privește alegerea unui sistem de gestionare SQL sau NoSQL, Sheldon (2021) sugerează o serie de diferențe între cele două tipuri de sisteme pe baza cărora decizia poate fi luată. De exemplu, prima diferență se referă la structura datelor.

Autorul sugerează că alegerea unui sistem de gestionare a bazelor de date SQL este mai degrabă potrivită în cazul în care datele sunt structurate, în timp ce alegerea unui sistem de gestionare a bazelor de date NoSQL este mai degrabă potrivită în cazul în care datele sunt nestructurate. Cei de la NetApp (2022) oferă o serie de exemple de date nestructurate precum datele media, colecții de documente (spre exemplu, e-mail), date de tip IoT sau AI. Christine (2021) discută de asemenea diferențele dintre datele structurate și datele nestructurate și indică spre diferența dintre datele stocate în formă de tabel și alte forme de date.

Alegerea unui sistem de gestionare a bazelor de date trebuie să considere și limbajul de interogare preferat. De exemplu, pentru un sistem de gestionare a bazelor de date SQL, limbajul folosit este rezumat strict la elementele ANSI/ISO. Pe de altă parte, pentru un sistem de gestionare a bazelor de date NoSQL, limbajul folosit poate varia în funcție de implementare și operații, oferind astfel șansa de a crea interogări prin limbaje de tip JavaScript.

Așa cum a fost menționat mai sus, un sistem de gestionare a bazelor de date SQL are o schemă de reprezentare pre-definită, în timp ce un sistem de gestionare a bazelor de date NoSQL are o schemă de reprezentare dinamică. Acest aspect oferă o mai mare flexibilitate sistemelor de

gestionare a bazelor de date NoSQL care pot gestiona date cu structură diferită în cadrul aceleiași baze de date. Astfel, criteriul flexibilității reprezintă un alt aspect ce trebuie considerat în decizia de a adopta un sistem de gestionare a bazelor de dat SQL sau NoSQL.

Un alt criteriu se referă la integritatea datelor. Un sistem de gestionare a bazelor de date SQL aderă principiilor ACID, care asigură o integritate sporită datelor gestionate. Pe de altă parte, un sistem de gestionare a bazelor de date NoSQL aderă la principiile BASE, care ridică probleme în materie de integritate a datelor. Astfel, în scenariile prelucrării datelor care necesită un grad sporit de integritate (spre exemplu, tranzacții financiare) este preferat un sistem de gestionare a bazelor de date SQL.

Problema reprezintă un alt criteriu care să ajute în decizia de a folosit un sistem de gestionare a bazelor de date SQL sau un sistem de gestionare a bazelor de date NoSQL. Astfel, un sistem de gestionare a bazelor de date NoSQL este optimizat pentru sistemele informatice care presupun seturi de date distribuite în mai multe centre, în timp ce un sistem de gestionare a bazelor de date SQL este optimizat pentru sisteme informatice care au datele centralizate într- un singur centru.

Un alt criteriu în vederea căruia se decide utilizarea unui sistem de gestionare a bazelor de date SQL sau a unui sistem de gestionare a bazelor de date NoSQL este complexitatea interogărilor. În exemplul prezentat în Figura 2.1 și Figura 2.2, poate fi observat avantajul unui sistem de gestionare a bazelor de date NoSQL în realizarea de interogări simple. Totuși, în cazul interogărilor complexe, un sistem de gestionare a bazelor de date SQL este superior în materie de timp necesar pentru realizarea acestora decât un sistem de gestionare a bazelor de date NoSQL.

În final, ultimul criteriu se referă la maturitatea sistemelor. Sistemele de gestionare a bazelor de date SQL au o maturitate superioară comparativ cu sistemele de gestionare a bazelor de date NoSQL. Acest aspect se traduce prin utilizarea de tehnologii bine cunoscute și comunități de suport dezvoltate pentru sistemele de gestionare a bazelor de date SQL.

Pornind de la aceste criterii introduse de Sheldon (2021), acesta oferă o grilă de decizie în vederea alegerii unui sistem de gestionare a bazelor de date SQL sau NoSQL (a se vedea Tabel 2-1). În baza acestei grile, a fost decisă alegerea unui sistem NoSQL pentru dezvoltarea platformei Optimize.

Criterii de alegere între sistemele SQL și NoSQL

SQL	NoSQL
Datele sunt structurate	Datele nu sunt structurate
Sistemul informatic operează cu un model de date predefinit precum cel al tranzacțiilor financiare sau contabile	Sistemul informatic necesită flexibilitate în alegerea modelului de date

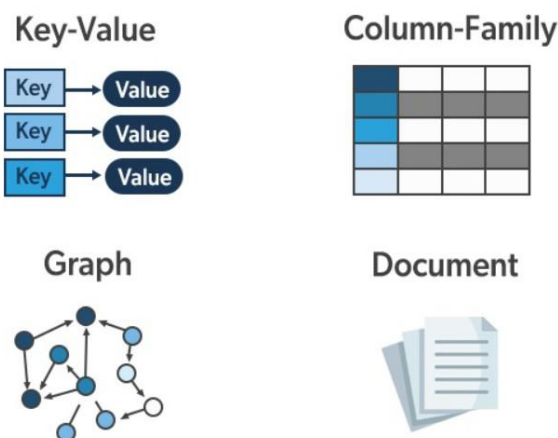
Sunt vizate dezvoltarea de interogări complexe	Este vizată dezvoltarea rapidă a sistemului și reducerea efortului de structurare a datelor
Este vizată creșterea capacității sistemului informatic pe verticală (prin adăugarea de resurse hardware mașinii virtuale)	Este vizată creșterea capacității sistemului informatic pe orizontală (prin adăugarea de mașini virtuale)

Tipuri de baze de date NoSQL

Spre deosebire de sistemele de gestionare a bazelor de date SQL, care prezintă o arhitectură tabulară, sistemele de gestionare a bazelor de date NoSQL pot prezenta mai multe arhitecturi. În acest sens, secțiunea de față introduce și descrie cele patru arhitecturi principale ale sistemelor gestionare a bazelor de date NoSQL, respectiv arhitectura de tip cheie-valoare, arhitectura de tip coloană, arhitectura de tip graf și arhitectura de tip document, reprezentate în figura de mai jos.

În rândurile ce urmează vor fi prezentate, pe rând, cele patru tipuri de sisteme de gestionare a bazelor de date NoSQL. Prezentarea va fi făcută prin intermediul unei metode comparative ce folosește ca punct de referință sistemele de gestionare a bazelor de date SQL. Finalul secțiunii introduce o analiză comparativă a celor patru tipuri de sisteme de gestionare a bazelor de date NoSQL.

NoSQL



Tipuri de sisteme de gestionare a bazelor de date NoSQL

Un sistem de gestionare a bazelor de date NoSQL de tip cheie-valoare presupune operarea cu o bază de date care stochează datele, așa cum numele sugerează, în forma de cheie-valoare (Williams, 2021a). Acest tip de bază de date presupune existența unui dicționar care

stochează o serie de obiecte identificabile prin intermediul unei „chei”. Fiecare obiect conține o valoare. Astfel, gestionarea datelor se realizează prin operarea cheilor care identifică poziția obiectului în dicționar.

Datele din bazele de date de tip cheie-valoare sunt stocate ca perechi de chei-valoare. Pentru exemplificare, este propusă ilustrația din figura de mai sus, în care se poate observa traducerea unei reprezentări de tip RDMS într-una de tip cheie-valoare.

Tabelul angajatilor

employee_id	first_name	last_name	address
1	John	Doe	New York
2	Benjamin	Button	Chicago
3	Mycroft	Holmes	London

Employees: Table

```
employee:$employee_id:$attribute_name = $value

employee:1:first_name = "John"
employee:1:last_name = "Doe"
employee:1:address = "New York"

employee:2:first_name = "Benjamin"
employee:2:last_name = "Button"
employee:2:address = "Chicago"

employee:3:first_name = "Mycroft"
employee:3:last_name = "Holmes"
employee:3:address = "London"
```

Reprezentare tip cheie-valoare a unui tabel RDMS

În figura de mai sus stocarea numelui „John” are loc prin asocierea acestuia unui obiect identificabil prin cheia „employee:1:first_name”. Mai departe, există trei operații realizabile prin intermediul unui sistem de gestionare a bazelor de date NoSQL asupra unei baze de date de tip cheie-valoare. Prima operație, de tip „put”, permite adăugarea unei perechi cheie-valoare. De exemplu, adăugarea unui nou student prin intermediul unui obiect nou care are asociată o cheie unică. A doua operație, de tip „get”, permite interogarea unei chei și a valorii asociate. În final, operația de tip „delete” permite ștergerea unei perechi de tip cheie-valoare.

Al doilea tip de sistem de gestionare a bazelor de date NoSQL este cel de tip coloană. Acest tip de bază de date, așa cum sugerează numele, presupune organizarea datelor pe coloană. În vederea ilustrării, este propusă Figura 3.3 care ilustrează diferența dintre bazele de date de tip rând (tipice sistemelor de gestionare a bazelor de date SQL) și bazele de date de tip coloană.

Baza de date locală

Baza de Date Locală (BDL) conține datele primite de la dispozitivul MobileEye, atunci când acesta generează alerte, împreună cu date adiționale. Datele adiționale, adică cele care nu fac parte obligatoriu din raportarea unui eveniment de către MobileEye, sunt împărțite în trei seturi de date, anume:

- Date fixe de identificare;
- Date volatile de raportare;
- Date volatile adiționale.

Datele fixe de identificare au rolul de a stabili în mod unic identitatea vehiculului) și identitatea șoferului, precum și o suită de amprente de timp specifice cursei vehiculului respectiv, cum ar fi:

- ID vehicul [ID_VEHICUL] (șir alfa-numeric ce conține indicatori despre numărul dispozitivului DiaMOTO integrat pe vehicul);
- ID șofer [ID_SOFER] (șir alfa-numeric ce conține identificatorul unic asociat conducătorului vehiculului);
- Dată activare [DATA_ACTIVARE] (șir de caractere de tip date/time specific MySQL) reprezintă data calendaristică în formatul MySQL
- Timp activare [TIMP_ACTIVARE] (șir de caractere de tip date/time specific MySQL) reprezintă amprenta de timp de tip HH:MM:SS ce înregistrează momentul în care dispozitivul DiaMOTO a fost activat.
- Timp pornire [TIMP_PORNIRE] (șir de caractere de tip date/time specific MySQL) reprezintă amprenta de timp de tip HH:MM:SS ce înregistrează momentul în care vehiculul începe să se deplaseze. Acest timp este necesar, deoarece poate exista o diferență de timp între momentul activării dispozitivului DiaMOTO și cel al începerii deplasării efective a vehiculului;
- Timp oprire intermediară [TIMP_OPRIRE_INTERMEDIARA] (șir de caractere de tip date/time specific MySQL) reprezintă amprenta de timp de tip HH:MM:SS ce înregistrează momentul în care vehiculul se oprește pentru mai mult de 5 minute (oprire în trafic, la semafor, în coloană etc.), prin folosirea unui numărător de timp. Dacă timpul de oprire este mai mic de 5 minute, atunci se va produce alterarea Bazei de Date Locale, eliminând înregistrarea opririi;
- Timp oprire destinație [TIMP_OPRIRE_DESTINATIE] (șir de caractere de tip date/time specific MySQL) reprezintă amprenta de timp de tip HH:MM:SS ce înregistrează momentul în care vehiculul se oprește la destinație.

Datele volatile de raportare sunt cele care, pe lângă datele fixe de identificare și cele volatile adiționale, însoțesc fiecare raportare a evenimentelor, sau raportările intermediare care au loc din timp în timp, în lipsa evenimentelor ADAS, și care contribuie la oferirea de informații suplimentare a premiselor și condițiilor în care se produce un eveniment ADAS, sau care vin să completeze anumite informații ce pot să explice comportamentul vehiculului/șoferului în trafic. Datele volatile de raportare, constau în:

- Poziție vehicul [POZITIE_VEHICUL] (șir de caractere alfa-numerice) reprezintă coordonatele geo-spațiale ale dispozitivului DiaMOTO;
- Viteză vehicul [VITEZA_VEHICUL] (șir de caractere alfa-numerice) reprezintă viteza vehiculului înregistrată în deplasare, atât în cazul raportărilor intermediare în lipsa evenimentelor ADAS, cât și pentru evenimentele ADAS;
- Timp eveniment [TIMP_EVENTIMENT] (șir de caractere de tip date/time specific MySQL) reprezintă amprenta de timp de tip HH:MM:SS ce înregistrează momentul în care se produce o avertizare a unui eveniment de către dispozitivul MobileEye.

Datele volatile adiționale sunt cele care, pe lângă datele fixe de identificare și cele volatile de raportare însoțesc fiecare raportare intermediară temporizată, în lipsa evenimentelor ADAS, precum și raportările evenimentelor generate de dispozitivul MobilEye. Acestea au rolul de a completa tabloul informațional din jurul unui eveniment raportat, precum și creionarea premiselor și circumstanțelor referitoare la deplasarea vehiculului pe ruta prestabilită.

În general, datele volatile adiționale reprezintă Date Meteorologice Localizate (DML) extrase din aplicații specializate, cum ar fi cele meteo, fiind posibile avertizări în funcție de datele meteorologice primite. Aceste date volatile adiționale sunt:

- Temperatura [TEMPERATURA] (șir de caractere numerice) reprezintă temperatura mediului ambiant, în grade Celsius;
- Umiditatea [UMIDITEATE] (șir de caractere numerice) reprezintă umiditatea mediului ambiant, în procente;
- Viteza vântului [VITEZA_VANT] (șir de caractere numerice) reprezintă viteza vântului, în km/h;
- Ploaie [PLOAIE] (șir de caractere numerice de valoare 0 sau 1) reprezintă prezența sau absența fenomenului atmosferic;
- Ninsoare [NINSOARE] (șir de caractere numerice de valoare 0 sau 1) reprezintă prezența sau absența fenomenului atmosferic;
- Polei [POLEI] (șir de caractere numerice de valoare 0 sau 1) reprezintă prezența sau absența fenomenului atmosferic.

Baza de date generală

Baza de Date Generală conține datele primite de la toate dispozitivele montate pe vehiculele din flotă. BDG reprezintă o replică a BDL, cu deosebirea că pentru fiecare dispozitiv se regăsește o tabelă în Baza de Date Generală în care vor fi sincronizate datele primite, atunci când dispozitivele generează alerte, împreună cu date adiționale.

De asemenea, înregistrările din Baza de Date Generală vor conține și date transmise de către dispozitive în absența evenimentelor ADAS, acest lucru contribuind semnificativ la realizarea trasabilității rutei prestabilite, precum și la oferirea unor informații folositoare care pot descrie comportamentul vehiculului/șoferului în trafic.

Datele de lucru sunt caracteristice a două scenarii, anume în prezența unui/unor eveniment(e) ADAS și în absența unui/unor eveniment(e) ADAS. Primul scenariu are ca scop înregistrarea datelor alertelor generate de echipamentele MobileEye pentru evenimentele din

trafic, în timp ce pentru al doilea scenariu pachetul de date are aceeași compoziție, cu excepția codului evenimentului ADAS.

Aceste date au ca surse următoarele:

- Echipamentul MobilEye cu senzorii aferenți;
- Ceasul intern
- Furnizori de servicii meteo personalizate local.

Indiferent de scenariul în care ne aflăm, integritatea datelor raportate este unul dintre cele mai sensibile și, în același timp, importante probleme în stabilirea arhitecturii software pentru sistemul ESDA

Alertele generate de către echipamentele MobileEye reprezintă nucleul pe care se dezvoltă aplicația ESDA, urmărind atingerea obiectivului esențial al proiectului și anume reducerea semnificativă a posibilității producerii de accidente rutiere și evitarea producerii de vătămări corporale sau chiar deces a persoanelor implicate în aceste incidente.

Astfel, integritatea datelor raportate reprezintă o problemă de la sine înțeleasă, deoarece, pe lângă alertele generate de către echipamentele MobileEye la sesizarea unor evenimente ADAS în trafic, un alt lucru deosebit de important este să înțelegem împrejurările care au dus la generarea alertelor respective.

Atâta vreme cât comunicațiile de date se realizează normal, adică există o comunicație de date fiabilă între dispozitivele ntegrate în flota de vehicule și serverele ESDA sincronizarea datelor între Baza de Date Locală a unui dispozitiv și serverele ESDA se realizează cu ajutorul funcțiilor MySQL pentru scriere, inserare și modificare.

Este important de realizat o rutină de comparare a datelor scrise în Baza de Date Generală cu cele din Baza de Date Locală, astfel evitându-se alterarea integrității datelor transmise prin Comunicațiile de date.

O măsură suplimentară de asigurare a siguranței integrității datelor este și aceea a comunicațiilor de date folosind protocolul TCP/IP, cunoscut fiind faptul că Transmission Control Protocol (TCP) este un protocol de comunicații ce asigură verificarea integrității încapsulării și a pachetelor de date în ansamblu cu ajutorul checksum, în cazul alterării integrității acestora solicitând retransmiterea lor.

De asemenea, faptul că modalitatea aleasă pentru stabilirea canalului de comunicații de date este cea de Virtual Private Network (VPN) presupune criptarea integrală a transmisiunii de date precum și alocarea de adrese Internet Protocol (IP) din domeniul claselor private, neutilizabile direct în Internet, asigură o probabilitate extrem de scăzută de alterare a datelor transmise de către dispozitive către serverele ESDA pe durata transmiterii acestora.

În cazul în care măsurile de asigurare a integrității datelor precizate mai sus sunt considerate insuficiente de către un beneficiar, este posibilă implementarea (opțională) a unui mecanism de tip CRC (Cyclic Redundancy Codes) prin implementarea unor algoritmi suplimentari de detecție a erorilor. De precizat totuși că orice strat suplimentar de verificare a integrității datelor care nu își dovedește imediata utilitate (cazul în care modalitățile de verificare a integrității sunt suficiente, fără verificare ciclică redundantă) poate conduce la îngreunarea

transmisiunii de date și astfel să scadă suplețea și robustețea sistemului ESDA

Astfel, ca procedură de lucru pentru sincronizarea evenimentelor cu baza de date ESDA trebuie parcurși cel puțin următorii pași:

1. Se culeg date de la echipamentul MobileEye, în cazul în care există un eveniment culege codul alocat evenimentului ADAS respectiv pentru care se generează alerta și completează câmpul [TIP_EVENT] din Baza de Date Locală (BDL) cu șirul de caractere alocat evenimentului ADAS respectiv (vezi capitolul 3), în caz contrar câmpul se completează cu șirul de caractere NONE;

2. În prezența evenimentelor ADAS datele primite de la echipamentul MobileEye completează câmpul [PREZENTA_EVENT] din Baza de Date Locală (BDL) cu valoarea 1, în caz contrar câmpul se va completa cu valoarea 0;

3. De fiecare dată când apare un eveniment ADAS raportat se realizează jurnalizarea sa în BDL, iar în lipsa evenimentelor ADAS se realizează o jurnalizare la fiecare minut

4. Odată cu raportarea evenimentelor ADAS, sau a jurnalizării de trafic la interval de 1 minut în absența evenimentelor ADAS, se jurnalizează în BDL și Datele fixe de identificare, Datele volatile de raportare și Datele volatile adiționale, conducând astfel la o trasabilitate mai bună a traficului și evenimentelor ADAS din trafic;

5. În urma fiecărei înregistrări în BDL se activează rutina software de verificare a conexiunii la Internet și a conexiunii cu serverele ESDA. În cazul în care conexiunea cu serverele ESDA este prezentă, se realizează sincronizarea datelor din BDL, înregistrare cu înregistrare, și se așteaptă confirmarea transmiterii cu succes de la serverele ESDA, moment în care înregistrarea pentru care s-a primit confirmarea se va șterge din BDL.

6. În cazul în care conexiunea cu serverele ESDA nu este prezentă, înregistrările din BDL se păstrează până la restabilirea conexiunii cu serverele ESDA, apoi se reia pasul procedural 5.

Concluzii

Alegerea tipului de baza de date se va face punctual pentru BDL și BDG în parte ținând cont de:

Structură datelor:

SQL: Bazele de date SQL utilizează un model relațional și folosesc tabele cu rânduri și coloane pentru a organiza datele într-o structură strictă. Relațiile între tabele sunt definite prin chei primare și chei externe.

NoSQL: Bazele de date NoSQL utilizează diferite modele de stocare a datelor, cum ar fi documente, grafuri, perechi cheie-valoare sau coloane. Aceasta permite flexibilitate în structura datelor și elimină constrângerile schemei stricte.

Scalabilitate:

SQL: Bazele de date SQL au în general o scalabilitate verticală, adică pot fi extinse prin adăugarea de resurse hardware mai puternice pentru a gestiona volumul crescut de date și trafic. Scalarea orizontală este posibilă, dar necesită sincronizare și gestionarea atentă a replicării

datelor.

NoSQL: Bazele de date NoSQL sunt concepute pentru a permite o scalabilitate orizontală ușoară. Acestea pot fi distribuite pe mai multe servere și pot gestiona încărcături mari de date și trafic prin adăugarea sau eliminarea de noduri din cluster.

Flexibilitatea schemei:

SQL: Bazele de date SQL utilizează scheme stricte, ceea ce înseamnă că structura și tipurile de date ale tabelor trebuie definite în prealabil. Orice deviație de la schema stabilită necesită modificări ale schemei și pot apărea dificultăți în gestionarea datelor nestructurate sau în schimbarea cerințelor de date.

NoSQL: Bazele de date NoSQL sunt flexibile în ceea ce privește schema datelor. Nu este nevoie de o schemă predefinită și pot fi adăugate, modificate sau eliminate câmpuri fără restricții. Acest lucru facilitează gestionarea datelor semi-structurate sau nestructurate și permite adaptabilitatea la schimbările cerințelor de date.

Performanță și viteza de acces:

SQL: Bazele de date SQL sunt optimizate pentru interogări complexe și tranzacții. Acestea oferă funcționalități avansate pentru optimizarea interogărilor și asigură consistența datelor. Cu toate acestea, performanța poate fi afectată în cazul unor interogări complexe sau cu volume mari de date.

NoSQL: Bazele de date NoSQL sunt concepute pentru a oferi o performanță ridicată și o viteza de acces sporită. Acestea sunt optimizate pentru operațiuni simple de citire/scriere și permit distribuirea eficientă a datelor pe mai multe noduri, ceea ce duce la timpi de răspuns mai rapizi.

Consistență și durabilitate:

SQL: Bazele de date SQL asigură de obicei consistența datelor prin utilizarea de tranzacții ACID (Atomicitate, Consistență, Izolare și Durabilitate). Aceasta garantează că datele sunt întotdeauna într-o stare coerentă și sigură.

NoSQL: Bazele de date NoSQL pot oferi diferite niveluri de consistență și durabilitate, în funcție de modelul specific utilizat. Anumite sisteme NoSQL, precum cele bazate pe perechi cheie-valoare, pot sacrifica consistența strictă pentru a obține o performanță mai mare și o scalabilitate mai bună.

Cercetarea s-a derulat în cadrul proiectului "E-SAFETY DRIVING APPLICATION (ESDA)", Cod SMIS 142406, finanțat prin Programul Operațional Competitivitate, Axa prioritară 2 - Tehnologia Informației și Comunicațiilor (TIC) pentru o economie digitală competitivă, Prioritatea de investiții 2b - Dezvoltarea produselor și serviciilor TIC, a comerțului electronic și a cererii de TIC, Obiectiv Specific OS 2.2 - Creșterea contribuției sectorului TIC pentru competitivitatea economică, Acțiunea 2.2.1 - Sprijinirea creșterii valorii adăugate generate de sectorul TIC și a inovării în domeniu prin dezvoltarea de clustere, Apelul de proiecte nr. 3

„Conținutul acestui material nu reprezintă în mod obligatoriu poziția oficială a Uniunii Europene sau a Guvernului României”