BIR NECHTA JADVALLARNI BIRLASHTIRISH:

SQLDA JOIN OPERATORLARI BILAN ISHLASH

Mahmudov Dilmurod
Farg‘ona davlat texnika universiteti
“Axborot texnologiyalari va telekommunikatsiya” fakulteti talabasi
e-mail: samsunggallaxiy35@gmail.com

Annotatsiya

Ushbu maqolada SQL tilida bir nechta jadvallarni birlashtirish usullari, JOIN operatorlarining ishlash prinsiplari hamda ularning ma’lumotlar bazasidagi ahamiyati yoritilgan. INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, CROSS JOIN va SELF JOIN operatorlarining vazifalari, qo‘llanilish sohalari va amaliy misollari tahlil qilingan. Shuningdek, jadvallar orasidagi bog‘lanishlar, Primary Key va Foreign Key tushunchalari ham ko‘rib chiqilgan. JOIN operatorlaridan foydalanishning afzalliklari va ayrim kamchiliklari ilmiy asosda bayon etilgan. SQL so‘rovlarini optimallashtirish, indekslardan foydalanish va zamonaviy ma’lumotlar bazasi tizimlarida JOIN operatorlarining ishlash mexanizmi haqida ham ma’lumot berilgan. Maqola ma’lumotlar bazasi bilan ishlovchi talabalar va dasturchilar uchun nazariy hamda amaliy ahamiyatga ega.

Kalit so‘zlar: SQL, JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, CROSS JOIN, SELF JOIN, ma’lumotlar bazasi, SELECT, relational database, MySQL, PostgreSQL.

Kirish

Hozirgi kunda axborot texnologiyalarining rivojlanishi natijasida katta hajmdagi ma’lumotlarni saqlash va boshqarish muhim masalaga aylandi. Korxonalar, banklar, universitetlar, internet do‘konlar va davlat tashkilotlarida ma’lumotlar bazasi tizimlaridan keng foydalaniladi. Zamonaviy axborot tizimlarida ma’lumotlar odatda bir nechta jadvallarda saqlanadi va ular o‘zaro bog‘langan bo‘ladi.

Masalan, universitet tizimida talabalar haqidagi ma’lumotlar bitta jadvalda, fanlar boshqa jadvalda, baholar esa alohida jadvalda saqlanishi mumkin. Ushbu ma’lumotlarni birlashtirib olish uchun SQL tilidagi JOIN operatorlaridan foydalaniladi.

JOIN operatorlari ma’lumotlar bazasining eng muhim qismlaridan biri hisoblanadi. Ular yordamida bir nechta jadvallar orasidagi bog‘lanishlar asosida kerakli ma’lumotlarni olish mumkin bo‘ladi. JOIN operatorlari dasturchilar va ma’lumotlar bazasi administratorlari uchun juda muhim vosita hisoblanadi.

Bugungi kunda MySQL, PostgreSQL, Oracle, SQL Server kabi mashhur ma’lumotlar bazasi tizimlarida JOIN operatorlari keng qo‘llaniladi. Katta hajmdagi ma’lumotlar bilan ishlashda JOIN operatorlaridan samarali foydalanish tizim tezligi va unumdorligiga bevosita ta’sir qiladi.

Ushbu maqolaning maqsadi — SQL tilidagi JOIN operatorlarini o‘rganish, ularning ishlash tamoyillarini tahlil qilish va amaliy qo‘llanilish sohalarini yoritishdan iborat.

ASOSIY QISM

Ma’lumotlar bazasi va jadvallar o‘rtasidagi bog‘lanish

Ma’lumotlar bazasida jadvallar o‘zaro bog‘langan holda ishlaydi. Ushbu bog‘lanishlar Primary Key va Foreign Key yordamida tashkil qilinadi.

Primary Key

Primary Key — jadvaldagi har bir qatorni noyob aniqlovchi ustun hisoblanadi. U takrorlanmaydi va NULL qiymat qabul qilmaydi.

Misol:

id	ism
1	Ali
2	Vali
3	Hasan

Bu yerda **id** ustuni Primary Key hisoblanadi.

Foreign Key

Foreign Key boshqa jadvaldagi Primary Key bilan bog‘lanishni ta’minlaydi.

student_id	baho
1	5
2	4
3	5

Bu jadvaldagi **student_id** ustuni Foreign Key bo‘lib xizmat qiladi.

Primary Key va Foreign Key yordamida jadvallar o‘rtasida bog‘lanish hosil qilinadi va JOIN operatorlari orqali ular birlashtiriladi.

JOIN operatori haqida tushuncha

JOIN operatori ikki yoki undan ortiq jadvallarni birlashtirish uchun ishlatiladi. JOIN yordamida bir nechta jadvallardagi bog‘langan ma’lumotlarni olish mumkin.

JOIN operatorlari quyidagi turlarga bo‘linadi:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- CROSS JOIN
- SELF JOIN

Har bir JOIN turi ma’lum vazifani bajaradi.

INNER JOIN

INNER JOIN eng ko‘p ishlatiladigan JOIN turi hisoblanadi. Ushbu operator faqat ikkala jadvalda mos keluvchi ma’lumotlarni chiqaradi.

Sintaksis

```
SELECT ustun_nomi
FROM jadval1
INNER JOIN jadval2
ON jadval1.id = jadval2.id;
```

Misol

```
SELECT ism, baho
FROM talabalar
INNER JOIN baholar
ON talabalar.id = baholar.student_id;
```

Natija

ism	baho
Ali	5

ism	baho
Vali	4
Hasan	5

INNER JOIN yordamida faqat mos ma’lumotlar olinadi.

INNER JOIN afzalliklari

- kerakli ma’lumotlarni aniq chiqaradi;
- ortiqcha NULL qiymatlar bo‘lmaydi;
- hisobotlar tayyorlashda qulay.

LEFT JOIN

LEFT JOIN chap tomondagi jadvaldagi barcha ma’lumotlarni chiqaradi. Agar o‘ng jadvalda mos qiymat bo‘lmasa, NULL qiymat qaytariladi.

Misol

```
SELECT ism, baho
FROM talabalar
LEFT JOIN baholar
ON talabalar.id = baholar.student_id;
```

Agar ayrim talabalar bahoga ega bo‘lmasa ham, ularning ismlari natijada chiqadi.

LEFT JOIN qo‘llanilishi

LEFT JOIN quyidagi hollarda foydali:

- barcha foydalanuvchilar ro‘yxatini chiqarish;
- ayrim ma’lumotlar yetishmaganda ham natijani saqlash;
- hisobotlarda barcha asosiy ma’lumotlarni chiqarish.

RIGHT JOIN

RIGHT JOIN o‘ng jadvaldagi barcha ma’lumotlarni chiqaradi.

Misol

SELECT ism, baho
FROM talabalar
RIGHT JOIN baholar
ON talabalar.id = baholar.student_id;

RIGHT JOIN amaliyotda LEFT JOIN ga nisbatan kamroq ishlatiladi. Chunki ko‘pchilik dasturchilar asosiy jadvalni chap tomonda joylashtiradi.

RIGHT JOIN afzalliklari

- o‘ng jadvaldagi barcha ma’lumotlarni saqlab qoladi;
- tahliliy hisobotlarda foydali.

FULL JOIN

FULL JOIN ikkala jadvaldagi barcha ma’lumotlarni chiqaradi.

Misol

SELECT ism, baho
FROM talabalar
FULL JOIN baholar
ON talabalar.id = baholar.student_id;

Agar mos ma’lumot bo‘lmasa, NULL qiymat chiqariladi.

FULL JOIN xususiyatlari

- ikkala jadvaldagi barcha qatorlarni saqlaydi;
- katta ma’lumotlarni tahlil qilishda foydali;
- ayrim ma’lumotlar yetishmasa ham natija chiqaradi.

CROSS JOIN

CROSS JOIN jadvallardagi barcha qatorlarni o‘zaro ko‘paytiradi.

Misol

```
SELECT *
FROM talabalar
CROSS JOIN fanlar;
```

Agar:

- birinchi jadvalda 3 ta qator;
- ikkinchi jadvalda 4 ta qator

bo‘lsa, natijada 12 ta qator hosil bo‘ladi.

CROSS JOIN qo‘llanilishi

- barcha kombinatsiyalarni hosil qilish;
- test ma’lumotlari yaratish;
- matematik va statistik hisob-kitoblarda.

SELF JOIN

SELF JOIN jadvalning o‘zini o‘zi bilan bog‘lash uchun ishlatiladi.

Misol

```
SELECT A.ism, B.ism
FROM xodimlar A, xodimlar B
WHERE A.rahbar_id = B.id;
```

SELF JOIN odatda:

- tashkilot strukturasi;
- rahbar va xodimlar bog‘lanishi;
- daraxtsimon strukturalar

kabi holatlarda qo‘llaniladi.

JOIN operatorlarining ishlash mexanizmi

JOIN operatorlari ishlaganda ma’lumotlar bazasi boshqaruv tizimi jadvallar orasidagi bog‘lanishlarni tekshiradi va ON sharti asosida mos qatorlarni topadi.

Jarayon quyidagi bosqichlarda amalga oshadi:

1. Jadvallar tanlanadi.
2. ON sharti tekshiriladi.
3. Mos qatorlar aniqlanadi.
4. Natijaviy jadval shakllantiriladi.

Katta hajmdagi ma’lumotlarda ushbu jarayon ko‘p resurs talab qilishi mumkin.

JOIN operatorlarining afzalliklari

JOIN operatorlari quyidagi afzalliklarga ega:

1. Ma’lumotlarni birlashtirish

Bir nechta jadvallardan kerakli ma’lumotlarni olish imkonini beradi.

2. Takroriy ma’lumotlarni kamaytirish

Normalizatsiya orqali ma’lumotlar alohida jadvallarda saqlanadi.

3. Murakkab hisobotlar yaratish

Turli jadvallardagi ma’lumotlarni birlashtirib hisobot olish mumkin.

4. Qulay boshqaruv

Katta ma’lumotlar bazasini samarali boshqarishga yordam beradi.

5. Ma’lumotlar yaxlitligi

Primary Key va Foreign Key orqali ma’lumotlarning to‘g‘riligi saqlanadi.

JOIN operatorlarining kamchiliklari

Ba’zi hollarda JOIN operatorlari quyidagi muammolarni yuzaga keltirishi mumkin:

- katta hajmdagi ma’lumotlarda so‘rov sekin ishlashi;
- ortiqcha xotira sarfi;
- murakkab so‘rovlar yozish zarurati;
- noto‘g‘ri JOIN ishlatilsa ma’lumotlarning takrorlanishi;
- noto‘g‘ri indekslash sababli tizim sekinlashishi.

Shuning uchun JOIN operatorlarini optimallashtirish muhim hisoblanadi.

JOIN operatorlarini optimallashtirish

SQL so‘rovlarini tez ishlashi uchun quyidagi usullar tavsiya etiladi:

Indekslardan foydalanish

Primary Key va Foreign Key ustunlariga indeks qo‘yish JOIN tezligini oshiradi.

Kerakli ustunlarni tanlash

SELECT *

o‘rniga faqat kerakli ustunlarni chiqarish tavsiya qilinadi.

WHERE shartidan foydalanish

Keraksiz ma’lumotlarni filtrlash tizim yuklanishini kamaytiradi.

Normalizatsiya

Jadvallarni to‘g‘ri loyihalash JOIN samaradorligini oshiradi.

Query Optimization

Zamonaviy DBMS tizimlari JOIN so‘rovlarini avtomatik optimallashtiradi.

JOIN operatorlarining qo‘llanilish sohalari

Bank tizimlari

Mijozlar, hisob raqamlari va tranzaksiyalarni bog‘lashda.

Universitet tizimlari

Talabalar, fanlar va baholarni birlashtirishda.

Internet do‘konlar

Mahsulot, buyurtma va foydalanuvchi ma’lumotlarini bog‘lashda.

Tibbiyot tizimlari

Bemorlar va davolash jarayonlarini boshqarishda.

Korxona boshqaruvi

Xodimlar va bo‘limlar ma’lumotlarini birlashtirishda.

Elektron hukumat tizimlari

Fuqarolar va davlat xizmatlari ma’lumotlarini bog‘lashda.

Zamonaviy ma’lumotlar bazalarida JOIN operatorlari

Hozirgi kunda MySQL, PostgreSQL, Oracle, SQL Server kabi tizimlarda JOIN operatorlari keng qo‘llaniladi. Zamonaviy tizimlar JOIN so‘rovlarini optimallashtirish uchun maxsus algoritmlardan foydalanadi.

Masalan:

- Hash Join;
- Merge Join;
- Nested Loop Join.

Ushbu algoritmlar katta hajmdagi ma’lumotlar bilan ishlashda yuqori samaradorlikni ta’minlaydi.

Bulut texnologiyalari va Big Data tizimlarida ham JOIN operatorlarining o‘rni juda katta hisoblanadi.

SQL JOIN operatorlarining kelajakdagi rivojlanishi

Kelajakda sun’iy intellekt va katta ma’lumotlar texnologiyalari rivojlanishi bilan SQL so‘rovlarini avtomatik optimallashtirish tizimlari yanada takomillashadi.

Bulutli ma’lumotlar bazalari, parallel hisoblash texnologiyalari va taqsimlangan tizimlarda JOIN operatorlaridan foydalanish yanada ortib boradi.

NoSQL tizimlari rivojlanayotgan bo‘lsa-da, relational database tizimlarida JOIN operatorlari asosiy vositalardan biri bo‘lib qolmoqda.

Xulosa

SQL tilidagi JOIN operatorlari ma’lumotlar bazasi bilan ishlashning muhim qismi hisoblanadi. Ushbu operatorlar yordamida bir nechta jadvallardagi ma’lumotlarni birlashtirish va kerakli natijalarni olish mumkin.

INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, CROSS JOIN va SELF JOIN operatorlari turli vaziyatlarda qo‘llanilib, ma’lumotlarni samarali boshqarish imkonini beradi.

JOIN operatorlarini to‘g‘ri qo‘llash ma’lumotlar bazasi unumdorligi va dastur ishlash tezligiga ijobiy ta’sir ko‘rsatadi. SQL so‘rovlarini optimallashtirish esa katta hajmdagi ma’lumotlar bilan ishlashda juda muhim hisoblanadi.

Kelajakda katta hajmdagi ma’lumotlar bilan ishlash jarayonlarida SQL va JOIN operatorlarining ahamiyati yanada ortib borishi kutilmoqda.

Foydalanilgan adabiyotlar

1. Elmasri, R., & Navathe, S. *Fundamentals of Database Systems*. Pearson, 2016.
2. Silberschatz, A., Korth, H., & Sudarshan, S. *Database System Concepts*. McGraw-Hill, 2019.
3. Coronel, C., & Morris, S. *Database Systems: Design, Implementation, and Management*. Cengage Learning, 2018.
4. Date, C. J. *An Introduction to Database Systems*. Pearson, 2003.
5. MySQL Documentation. *SQL JOIN Syntax and Usage*. Oracle Corporation, 2022.
6. PostgreSQL Global Development Group. *PostgreSQL Documentation*. 2023.

7. Microsoft Corporation. *SQL Server Documentation*. 2022.
8. Garcia-Molina, H., Ullman, J., & Widom, J. *Database Systems: The Complete Book*. Pearson, 2008.