MYSQL ADATBÁZIS-KEZELÉS MESTERKURZUS

Adatformázás és Rendszerezés

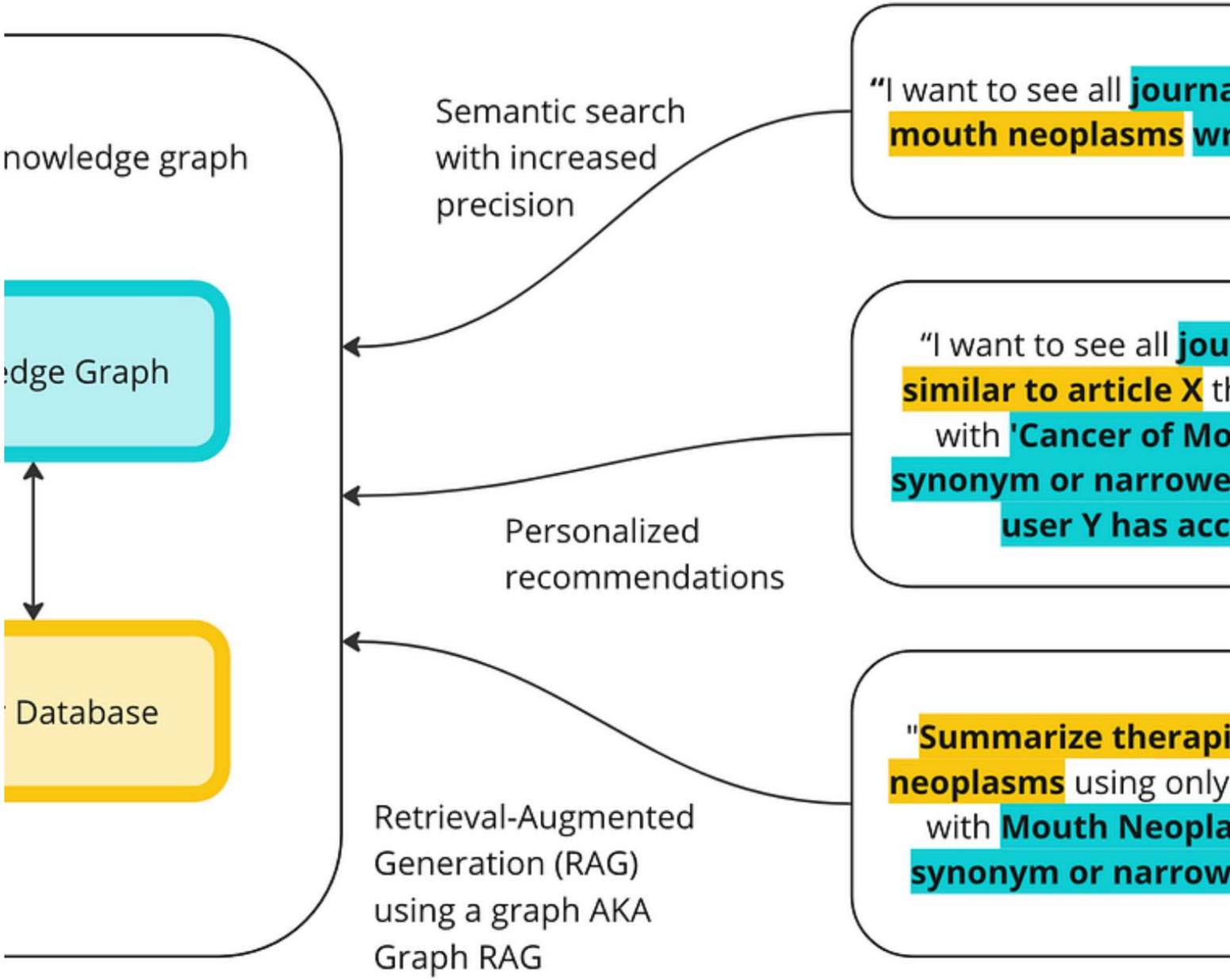
GROUP BY, ORDER BY, LIMIT, DISTINCT és a lekérdezések finomhangolása

AZ ADATOK URALMA

Nyers Adatból Használható Információ

A SELECT önmagában egy ömlesztett listát ad. Az igazi érték a strukturálásban, a csoportosításban és a pontos megjelenítésben rejlik.

- Releváns rekordok izolálása.
- Logikus sorrend felállítása.
- Eredmények aggregálása.



EGYEDI ÉRTÉKEK KINYERÉSE

Duplikációk Eltávolítása: DISTINCT

```
-- Ismétlődő műfajok listája (sok azonos):
SELECT mufaj FROM filmek;

-- Csak a létező kategóriák egyszer:
SELECT DISTINCT mufaj FROM filmek;
```

- Azonos értékű sorok összevonása.
- A SELECT után közvetlenül áll.
- Több oszlop esetén a kombinációjuknak kell egyedinek lennie.

SORRENDBE ÁLLÍTÁS

Az ORDER BY Szabályai

`ORDER BY oszlop`

Alapértelmezett sorrendbe rendezés a megadott mező alapján.

`ASC (Ascending)`

Kisebbtől a nagyobbig. A-tól Z-ig. Alapbeállítás, elhagyható.

`DESC (Descending)`

Nagyobbtól a kisebbig. Z-től A-ig. Legfrissebb dátum elöl.

GYAKORLATI RENDEZÉS

Összetett Rendezési Stratégiák

```
-- Legdrágább termékek elöl:
SELECT cim, ar FROM konyvek ORDER BY ar DESC;

-- Kategória szerint, azon belül ár szerint csökkenő:
SELECT nev, ar FROM termekek ORDER BY kategoria ASC,
ar DESC;
```

A rendezéshez használt oszlopnak nem kötelező megjelennie a SELECT listában.

Többszintű rendezésnél balról jobbra érvényesül a prioritás.

EREDMÉNY KORLÁTOZÁSA

A Célzott Toplisták: LIMIT

```
LIMIT 5
```

Csak az első 5 találatot adja vissza.

Kritikus teljesítmény-optimalizálási eszköz nagy adatbázisoknál.
Gyakran az ORDER BY-al együtt használjuk toplistákhoz.

LAPOZÁS (PAGINATION)

LIMIT és OFFSET Használata

```
-- Oldalanként 10 termék, első oldal:
SELECT * FROM termekek LIMIT 10 OFFSET 0;

-- Második oldal (11-20. rekord):
SELECT * FROM termekek LIMIT 10 OFFSET 10;
```

- `LIMIT`: Visszaadott sorok maximális száma.
- `OFFSET`: Hány kezdő sort hagyjon ki a listából.
- Rövidített szintaxis MySQL-ben: `LIMIT offset, count` (pl. `LIMIT 10, 10`).

ADATOK ÖSSZEFOGÁSA

Hogyan Működik a GROUP BY?

1. Nyers Adatok
Tábla kiinduló rekordjai.

→

2. Csoportképzés
Mező szerinti partíciók.

→

3. Aggregáció
Függvényfutás csoportokon.

→

4. Végeredmény
Egy sor per kategória.

CSOPORTOSÍTOTT SZÁMÍTÁSOK

GROUP BY a Gyakorlatban

```
-- Hány film van műfajonként?
SELECT mufaj, COUNT(*) FROM filmek GROUP BY mufaj;

-- Mi az átlagár kategóriánként?
SELECT kategoria, AVG(ar) FROM termekek GROUP BY
kategoria;
```

A SELECT listában szereplő nem-aggregált mezőknek szerepelniük kell a GROUP BY-ban.

Összevonja az azonos értékeket, és a függvényeket ezeken a blokkokon futtatja.

A KÉT SZŰRŐ

A WHERE és a HAVING Különbsége

WHERE Szűrés

A csoportosítás (GROUP BY) ELŐTT fut le. Egyedi sorokat vizsgál.

```
WHERE ar > 1000
```

HAVING Szűrés

A csoportosítás UTÁN fut le. Magát az aggregált eredményt szűri.

```
HAVING COUNT(*) > 5
```

CSOPORTOK KISZŰRÉSE

HAVING Használata a Gyakorlatban

```
-- Csak azok a kategóriák, ahol az átlagár > 2000:
SELECT kategoria, AVG(ar)
FROM termekek
GROUP BY kategoria
HAVING AVG(ar) > 2000;
```

- A WHERE nem tud statisztikát (aggregátumot) vizsgálni.
- A HAVING kizárólag a GROUP BY-al együtt érvényes.

A HÁTTÉRBEN

Az SQL Logikai Végrehajtási Sorrendje

1. **FROM:** Tábla kiválasztása.
2. **WHERE:** Nyers sorok előzetes szűrése.
3. **GROUP BY:** Fennmaradó sorok kategóriákba rendezése.
4. **HAVING:** Csoportok utólagos szűrése.
5. **SELECT:** Megjelenítendő oszlopok kiválasztása.
6. **ORDER BY & LIMIT:** Formázás és darabszám-korlátozás.

PREZENTÁCIÓS FORMÁZÁS

Aliases (AS) Használata

```
-- Oszlopok átnevezése a kimeneten:
SELECT nev AS 'Termék Neve', ar AS 'Fogyasztói Ár'
FROM termekek;

-- Aggregátumok megnevezése:
SELECT kategoria, COUNT(*) AS db_szam
FROM termekek GROUP BY kategoria;
```

> Csak a megjelenítést változtatja meg, a tábla szerkezetét nem.

> Az AS kulcsszó elhagyható, de kiírása javasolt az olvashatóságért.

ÖSSZETETT KERESÉSEK

Beágyazott Lekérdezések (Subqueries) Alapjai

Mi az a Subquery?

Egy SELECT utasítás beágyazása egy másik lekérdezésbe (pl. a WHERE záradékba).

```
-- Termékek lekérése, melyek drágábbak az átlagnál:
SELECT nev, ar FROM termekek
WHERE ar > (SELECT AVG(ar) FROM termekek);
```

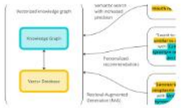
A LEGFONTOSABB SZABÁLY

A Lekérdezés Szintaktikája

A hibátlan SQL parancsok írásának titka a kötelező sorrend betartása.

```
SELECT [oszlopok]
FROM [tábla]
WHERE [feltételek]
GROUP BY [kategóriák]
HAVING [csoport-feltételek]
ORDER BY [rendezés]
LIMIT [mennyiség];
```

Image Sources



https://miro.medium.com/v2/resize:fit:1200/1*hrwv6zmmgogVNpQQlOIwIA.png

Source: medium.com