Indexing

المقدمة :-

هي طريقة تستخدم لتحسين أداء العمليات اللي بتم علي الdatabase مثل : (select - join - order by - where) و الوصول الى البيانات بشكل أسرع .

بيتكون من :-

1- search key :-

بيكون فيه القيم اللى بنستخدمها فى البحث و ممكن تكون مرتبه او لا

2- Data reference / pointer :-

دا مكان تخزين البيانات المرتبطة بال search key سواء كان (primary key - foreign key)

الخصائص :-

1- Access type :-

يعني الطرق المختلفة اللى بيتم بيها البحث مثل :- Where

Ex :-

```
Select * from `users` where id = 3;
```

هنا بيتم البحث عن المستخدم رقم 3 في جدول ال users

2- Access time :-

هو الوقت المستغرق للوصول لعنصر معين و قراءة بياناته و بيتكون من :

1- seek time :-

هو الوقت المستغرق للوصول للمسار المخزن عليه البيانات المطلوبة

2- Latency time :-

هو الوقت المنتظر حتي تصبح البيانات المطلوبه جاهزه للقراءة

3- Data transfer time :-

هو الزمن المستغرق لنقل البيانات الي المعالج او الذاكرة , كلما زادت سرعة النقل زادت كفاءة الجهاز في تحميل الملفات و تشغيل البرامج

3- Insertion time :-

الوقت المطلوب لإيجاد مكان مناسب لإضافة البيانات

4- Deletion time :-

الوقت المستغرق للعثور علي البيانات و حذفها

5- Space overhead :-

هي المساحة الإضافية المطلوبة لتخزين البيانات

انواع الindexing :-

1- Primary indexing :-

يتم ترتيب الصفوف فى الجدول بناءا علي الprimary key

2- Secondary indexing :-

تستخدم مع اي field لتحسين سرعة البحث

3- Clustered indexing :-

بتسرع من عملية ال search لكن بتكون بطيئه في عمليات ال insert , update

4- Non-clustered indexing

المميزات :-

1- تحسين الأداء عن طريق جعل ال search اسرع

2- تحسين عملية ال (order by) .
3- مع زيادة حجم البيانات تحافظ علي سرعتها .

- بمعنى في ال Indexing بيتم تنفيذها علي الDatabase تلقائى خصوصا على column ال ID لإنه بيكون Primary key لو بتعمل عمليات كتيرة علي Column معين غير الID و موجود فيه عناصر كتيرة ف عملية البحث هتاخد وقت كتير ليه ؟
  - ❖ عشان ال Database هتعمل scan لل table بالكامل (Full Table Scan) يعني هتبدأ ب اول row و تشوف الcolumns اللي فيه لغاية متوصل للcolumn اللي بتدور عليه و بعدين تقارن القيمة اللى موجودة فيه بالقيمة المطلوبة بس بياخد وقت كتير و عشان نحل المشكلة دي بنلجأ لل Indexing .
- في الIndexing بناخد الColumn اللي بنعمل عمليات عليه كتير و يتم انشاء فهرس منفصل لل Column دا لوحده بعيد ومنفصل عن ال table بيكون فيه القيم المخزنة و ال ID الخاص بكل row و عشان يوصل لقيمة معينة مش بيروح للtable بيروح لل الفهرس الاول بيدور فيه و بياخد مكان العنصر المطلوب .
  - ➢ لإنشاء Index ل cloumn معين في table :-
  - ➢ create index index_name on table(column);
  - ➢ لعرض جميه ال Indexs الخاصه ب table معين :-
  - ➢ show index from table_name;
  - ➢ لحذف ال Index :-
  - ➢ drop index index_name on table_name;
- ★ في بعض الDatabases عند جعل الColumn فريد (Unique) في دا بيعمل index ليه علطول لوحدة زى (MySQL - PostgreSql) .

Sql and NoSql DataBase

المقدمة :-

الاتنين انظمة لتخزين البيانات , لكن في اختلافات بينهم منها طريقة تنظيم البيانات و معالجتها و استخدامتهم .

-: SQL (structured query language)

هي قاعدة بيانات تستخدم Structure منظم يعتمد على الجداول و تخزين البيانات فيها و العلاقات بين هذه الجداول و تستخدم لغة (SQL) لتنفيذ ال query .

المميزات :-

1- هيكل منظم :- تعتمد علي جداول بينها علاقات محددة مسبقا بمخططات .
2- ضمان ان البيانات تظل صحيحه حتي في حالة حدوث خطأ في النظام (ACID) و هي (Atomicity - Consistency - Isolation - Durability)

- Atomicity -:

يعني بيتم تنفيذ ال Query بالكامل او لا يتم تنفيذها يعني مينفعش ال query تقف في النص و هي شغاله يعني مثلا : لو بننقل فلوس من حساب ل آخر مينفعش نخصم الفلوس من الحساب الاول و منزوداش للحساب التاني .

- Consistency -:

بعد تنفيذ اي query تظل الdatabase صحيحه بحيث ميكنش في اي بيانات غير سليمه يعني مثلا : مينفعش الحساب البنكي يكون فيه رصيد بالسالب .

- Isolation :-

يعني كل query بتتنفذ لوحدها بحيث لا تؤثر query علي التانيه الي ان تنتهي يعني مثلا : لو في عميلين بيعملوا تحويلين مختلفين في نفس الوقت ف كل تحويل بيحصل لوحده لعدم حدوث اخطاء في الحساب .

- Durability :-

يعني لما الquery تتنفذ بيتم حفظ التغيرات بشكل دائم حتي في حاله حدوث اعطال يعني مثلا : لو قمت بشراء منتجات و تم تسجيلها في ال database ف تفضل محفوظه حتي لو النظام تعطل .

أمثلة :-

(MySql - PostgreSql - Oracle Database - Microsoft Sql Server)

-: NoSQL(Not Only SQL)

هي قواعد بيانات مصممة للتعامل مع البيانات الكبيرة و غير المنظمة توفر مساحة و مرونة كبيرة في تخزين البيانات لا تعتمد على الجداول فقط .

المميزات :-

1- المرونة :- يسهل التعامل مع البيانات لإن مفيش Structure تم تحديدة .
2- يمكن توزيع البيانات على عدة خوادم .
3- الأداء العالى :- تم تصميمها لتسريع عملية القراءة و الكتابة مما يجعلها مناسبة للأنظمة الضخمة .

أمثلة :-

(MongoDB - Cassandra - Redis - Amazon SimpleDB)

الخلاصة :-

ال SQL بنستخدمها لو عندنا Structure محدد و ثابت لو بنعتمد علي Query معقدة و الRelation بين الجداول و بعض , بنستخدمها في (أنظمة الدفع - إدارة المخازن - CRM) .
بينما ال NoSQL بنستخدمها عند التعامل مع البيانات الضخمة و غير المنظمة و عند الحاجة لسرعة معالجة البيانات و بنستخدمها في (Social Media Platforms - Big Data) .