**difference between SQL and NoSQL**

SQL:

تعتمد على مخططات جداول محددة مسبقًا وتتطلب تعريف هيكلية البيانات بدقة قبل إدخال البيانات. تستخدم العلاقات بين الجداول لتمثيل البيانات.

NoSQL: لا تحتوي على مخططات جدول صارمة، وتتيح مرونة أكبر في تخزين أشكال متنوعة من البيانات، بما في ذلك الوثائق والأعمدة والرسوم البيانية.

القابلية للتوسع:

SQL: يمكن أن تكون صعبة التوسع بشكل أفقي (إضافة المزيد من الخوادم) بسبب تبني نموذج العلاقات.

NoSQL: تميل إلى أن تكون أكثر قابلية للتوسع بشكل أفقي، حيث يمكن إضافة خوادم جديدة بسهولة لمعالجة تحميل البيانات المتزايد.

الأداء:

SQL: توفر أداءً متسقًا مع تعيين مفاتيح رئيسية فعالة واستخدام مؤشرات متقدمة.

NoSQL: قد توفر أداءً ممتازًا للقراءة والكتابة السريعة للبيانات على نطاق واسع، خاصة عندما يكون التطبيق يتطلب تخزين واسترجاع كميات كبيرة من البيانات غير المهيكلة.

توافق البيئة:

SQL: تستخدم في التطبيقات التي تتطلب التعامل مع البيانات المهيكلة بشكل صارم والتي تحتاج إلى ضمانات النسبية الأدق.

NoSQL: يستخدم في التطبيقات التي تتطلب مرونة في هيكلية البيانات و/أو التوسع الأفقي الكبير و/أو القدرة على التعامل مع بيانات غير مهيكلة.

**2. difference between DROP, TRUNCATE, DELETE**

هناك ثلاثة أوامر شائعة في قواعد البيانات العلاقية مثل MySQL و PostgreSQL وغيرها، والتي تستخدم لحذف البيانات أو الجداول بشكل كامل، وهي:

**1. DROP:**

- الوصف: يتم استخدام الأمر DROP لحذف جدول بأكمله من قاعدة البيانات.

-التأثيرات: عند استخدام DROP TABLE، يتم حذف الجدول بما فيه من بيانات وبنية الجدول بشكل كامل ولا يمكن استرجاع البيانات المحذوفة.

-مثال:

DROP TABLE table\_name;

**2. TRUNCATE:**

- الوصف: يستخدم الأمر TRUNCATE لحذف جميع السجلات من الجدول.

- التأثيرات: عند استخدام TRUNCATE TABLE، يتم حذف جميع البيانات الموجودة في الجدول ولكن يتم الاحتفاظ ببنية الجدول.

- مثال:

TRUNCATE TABLE table\_name;

**3. DELETE:**

- الوصف: يستخدم الأمر DELETE لحذف سجلات محددة من الجدول بناءً على شرط معين.

-التأثيرات: عند استخدام DELETE، يتم حذف السجلات التي تتطابق مع الشرط المحدد فقط، ولكن بنية الجدول تظل كما هي.

- مثال:

DELETE FROM table\_name WHERE condition;

باختصار:

- DROP يحذف الجدول بأكمله.

- TRUNCATE يحذف جميع البيانات من الجدول ولكن يحافظ على بنية الجدول.

- DELETE يحذف سجلات محددة من الجدول بناءً على شرط معين دون تغيير بنية الجدول

. **A screenshot of a computer program

Description automatically generated**

**3.. what is the meaning of (CRUD)**

Create (إنشاء): يشير إلى عملية إنشاء بيانات جديدة في قاعدة البيانات. هذه العملية تتضمن إضافة سجل جديد إلى الجدول بالقيم المطلوبة.

Read (قراءة): يشير إلى عملية استعراض أو استرجاع البيانات من قاعدة البيانات. يمكن أن تكون عملية القراءة تتضمن استعراض كل البيانات المتاحة أو استرجاع بيانات محددة بناءً على معيار محدد.

Update (تحديث): يشير إلى عملية تحديث البيانات الموجودة في قاعدة البيانات. يتم ذلك عادةً عن طريق تغيير قيم الحقول لسجلات موجودة في الجدول.

Delete (حذف): يشير إلى عملية حذف البيانات من قاعدة البيانات. يمكن أن تشمل هذه العملية حذف سجلات فردية أو حتى جداول بأكملها.

**.4self join**

الـ Self Join هو نوع من عمليات الانضمام في SQL حيث يتم الانضمام بين جدول ونفسه. هذا يتيح لك مقارنة الصفوف داخل نفس الجدول. في الـ Self Join ، تستخدم نفس الاسم المستعار للجدول (أو أسماء مختلفة إذا تم الرجوع إلى الجدول أكثر من مرة) لتمثيل نفس الجدول في عملية الانضمام. هذا يتيح لك إنشاء علاقات منطقية بين الصفوف داخل نفس الجدول.

يُستخدم الـ Self Join عادةً عندما تحتاج إلى مقارنة السجلات داخل نفس الجدول ، مثل عندما تحتوي على بيانات تسلسلية أو عندما تحتاج إلى العثور على السجلات ذات الصلة بناءً على معايير معينة.

إليك مثالًا على Self Join:

لنفترض أن لديك جدول يُسمى "الموظفين" بأعمدة `رقم\_الموظف` و `رقم\_المدير`. يحتوي عمود `رقم\_المدير` على رقم المدير لكل موظف.

يمكنك استخدام Self Join للعثور على اسم مدير كل موظف:

SELECT e.employee\_id, e.employee\_name, m.employee\_name AS manager\_name

FROM employees e

JOIN employees m ON e.manager\_id = m.employee\_id;

في هذا المثال:

- نقوم بعملية الانضمام بين الجدول "الموظفين" ونفسه باستخدام الاسماء المستعارة `e` و `m` لتمثيل الجدول في حالتين مختلفتين.

- نقوم بعملية الانضمام استنادًا إلى عمود `رقم\_المدير` في جدول "الموظفين" (`e`) المطابق لعمود `رقم\_الموظف` في نفس الجدول (`m`) ، مما ينشئ العلاقة بين الموظفين ومدرائهم.

- نقوم باختيار الأعمدة `رقم\_الموظف` و `اسم\_الموظف` من جدول "الموظفين" (`e`) ، بالإضافة إلى عمود `اسم\_المدير` من جدول "الموظفين" (`m`) الذي يمثل اسم المدير

ستعيد هذه الاستعلام نتائج تظهر رقم واسم كل موظف ، بالإضافة إلى اسم مدير كل موظف.

**difference about join and union.5**

عملية الانضمام (Join):

يستخدم لدمج الصفوف من جداول مختلفة استنادًا إلى شروط محددة.

يتيح لك الحصول على بيانات من عدة جداول في نفس الاستعلام.

يستخدم شروط الانضمام مثل INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN لتحديد كيفية دمج الصفوف بين الجداول.

عملية الاتحاد (Union):

يستخدم لجمع الصفوف من نتائج استعلامات مختلفة في نفس الهيكل.

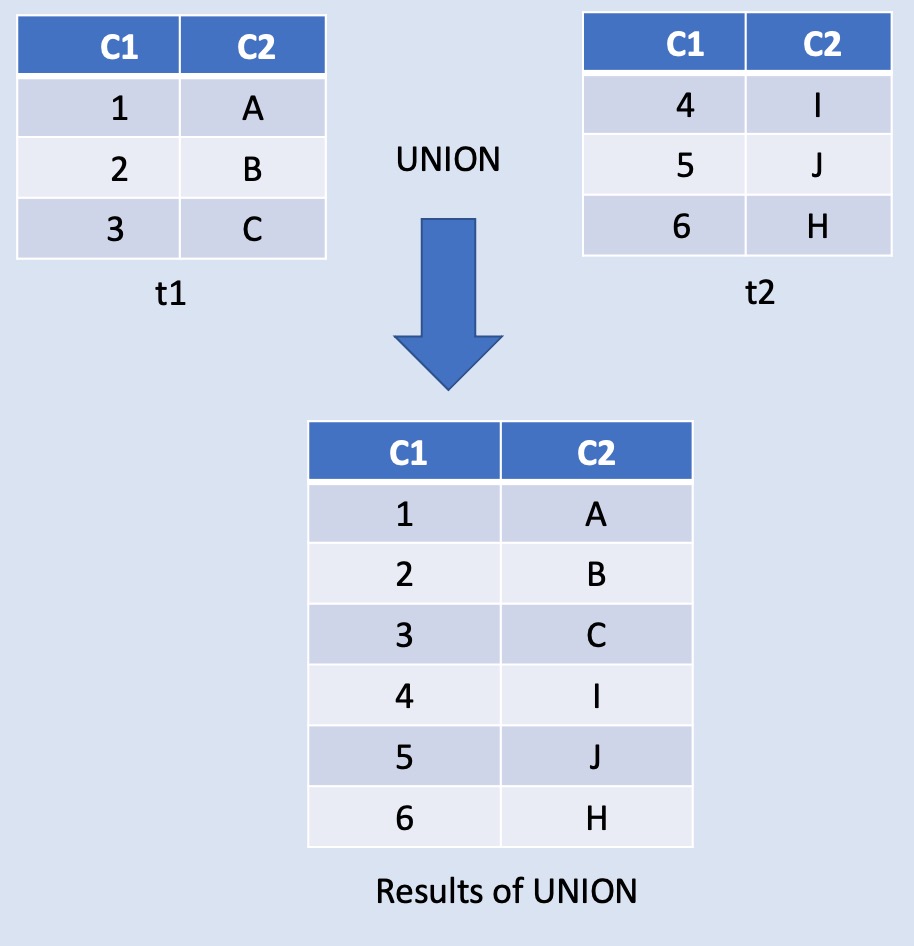
يجمع بين نتائج استعلامات مختلفة ويعرضها كنتيجة واحدة.

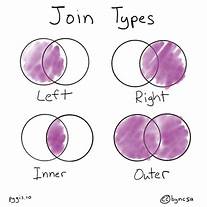
يجب أن تحتوي استعلامات الاتحاد على نفس عدد الأعمدة ونفس الأنواع من البيانات.

لتبسيط الفرق:

عملية الانضمام تستخدم لدمج البيانات من جداول مختلفة باستخدام شروط محددة.

عملية الاتحاد تستخدم لجمع البيانات من نتائج استعلامات مختلفة في هيكل واحد.

A screenshot of a computer

Description automatically generated

**6. Indexing and Indexing types in database**

Indexing في قواعد البيانات هو عملية إنشاء هيكل بيانات فرعي يساعد في تسريع استعلامات قاعدة البيانات عن طريق تحسين سرعة استرجاع البيانات. يقوم الفهرس بتنظيم البيانات في ترتيب معين (مثل الأبجدية أو الترتيب الرقمي) مما يجعل عمليات البحث أسرع وأكثر كفاءة.

أنواع الفهارس في قواعد البيانات تشمل

1. الفهرس التقليدي (Traditional Index): هو نوع فهرس شائع يستخدم لتسريع عمليات البحث في الأعمدة المحددة. يتم إنشاء الفهرس على قيمة العمود المحددة مما يساعد في تسريع البحث عن القيم المحددة.

2. الفهرس الفرعي (Composite Index): يستخدم لتسريع عمليات البحث التي تتضمن أكثر من عمود. يتم إنشاء الفهرس الفرعي على مجموعة من الأعمدة المختلفة مما يساعد في تحسين الأداء عند البحث باستخدام مجموعة معينة من الشروط.

3. الفهرس الفرعي المترتب (Clustered Index): يتم ترتيب البيانات في الفهرس الفرعي المترتب بنفس ترتيب البيانات في الجدول نفسه. هذا يعني أن البيانات مرتبة في الفهرس بنفس الترتيب الذي تم فيه تخزينها في الجدول، مما يساعد في تسريع البحث عن البيانات.

4. الفهرس الغير مترتب (Non-clustered Index): في هذا النوع من الفهارس، يتم تنظيم البيانات في الفهرس بطريقة مختلفة عن ترتيبها في الجدول نفسه. يستخدم هذا النوع من الفهارس لتسريع عمليات البحث والاستعلام بدون تأثير على ترتيب البيانات في الجدول.

5. الفهرس النصي (Full-text Index): يتم استخدامه لتسريع عمليات البحث في النصوص الطويلة مثل المقالات أو الوثائق. يتيح لك البحث بكلمات مفتاحية وعبارات داخل النصوص بشكل أسرع وأكثر دقة.

تُعتبر الفهارس أداة هامة في تحسين أداء قواعد البيانات وتقليل وقت الاستعلامات، ولكن يجب استخدامها بعناية وفقًا لاحتياجات التطبيق وحجم البيانات المخزنة.

