

Indexing in database

ال indexing هو ترتيب المعلومات بشكل معين في ال database بمعنى تاني بيعمل فهرسه للبيانات طيب هتقول ليه اعمل كذا هقولك لان دا ببسهل ويبسرع شغل واداء ال database لانها بدل ما تعمل ال read لل table كلها لا هيا هتعرف توصل للمعلومه بسرعه بسبب ال indexing وبكدا نفهم ان عندنا نوعين بحث هما

عشوائي —> Full table scan

مرتب —> Indexing

في كتابة الكود لما اقول ان الخانه دى مثلا unique بشكل تلقائي بيعمل عليها indexing وال fk وال pk كلها لازم يكون ليها index طيب ايه ال fk / pk

Fk stand for : foreign key

PK stand for : primary key

ودول عاملين زى شجره لان المفتاح الثانوى وهو ال fk دا بياخد قيمته من ال pk ويبقى بينهم علاقه بتربط الجداول ببعضها

Foreign key example					
Customers			Orders		
CustomerNo	FirstName	LastName	OrderNo	EmployeeNo	CustomerNo
1	Sally	Thompson	1	1	42
2	Sally	Henderson	2	4	1
3	Harry	Henderson	3	1	68
4	Sandra	Wellington	4	2	4
6	Arthur	Paulson	5	3	6
7	Derry	Beckham	6	2	112
8	Maude	Jones	7	2	8

Primary key Foreign key

في المثال دا ال order table فيها خانه للرقم الخاص بالعميل الرقم دا بيبقى خاص بالعميل صاحب ال order دا بس هنا بقى دا ببسهل عليا انى اعرف عدد او ايه ال orders اللي عاملها العميل وهنا ال الرقم بتاع العميل في ال customers table هو ال pk وفي ال orders table هو ال fk

وال indexing بينقسم ل 3 انواع وهما كالتالى

Primary / clustering / secondary

نبدأ بال primary ودا بينقسم لجزئين هما

Dense / sparse

ال dense هو نوع من انواع ال Indexing سيكون على ال unordered table وكمان لازم عناصر ال index اللي في ال table ال index تكون قد العناصر الموجوده في ال table اللي بيشار عليها ال index

UP	•	UP	Agra	1,604,300
USA	•	USA	Chicago	2,789,378
Nepal	•	Nepal	Kathmandu	1,456,634
UK	•	UK	Cambridge	1,360,364

نيجي بقي لل sparse ودا بيتم على ال ordered table هنا ال indexing بيكون مبني اكر نقدر نقول على ال address اللي database تبدا تدور منه على الداتا اللي هيا عايزها بمعنى ان عدد عناصر ال index table مش قد ال main table

		1	John	25
		2	Jack	24
		3	Amey	18
1		4	Ellena	29
4		5	Kate	31
6		6	Will	26

Index record Data block

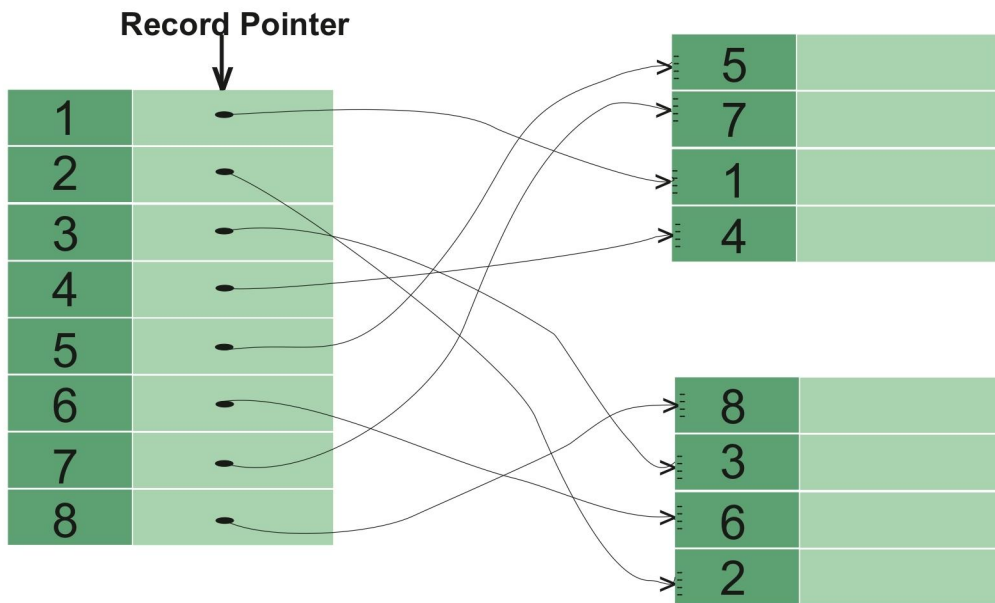
هنا في المثال دا ال Index بيشار على الرقم واحد طيب لو انا عايز الرقم 3 هجيبه ازاى هجيبه عن طريق ال index 1 بردو لان ال database هتبص على ال indexing هتلاقي ال 3 اصغر من ال 4 فا كدا اكيد ال 3 هلاقيها لو مشيت ورا العنوان اللي بيشار عليه ال index 1

Clustering

بيعمل ال indexing على اساس الترتيب الصحيح بمعنى انه الارقام في الجدول هتكون مترتبه بترتيبها الصحيح مش بشكل عشوائي بس في حاجه كمان الترتيب دا ال Index بتعملهم مش بيتكررو لازم يبقى unique كا index في ال indexing table وهو بيسهل البحث لانه مترتب ومعمول للبيانات فيه sorted منظم ودا مثال عليه



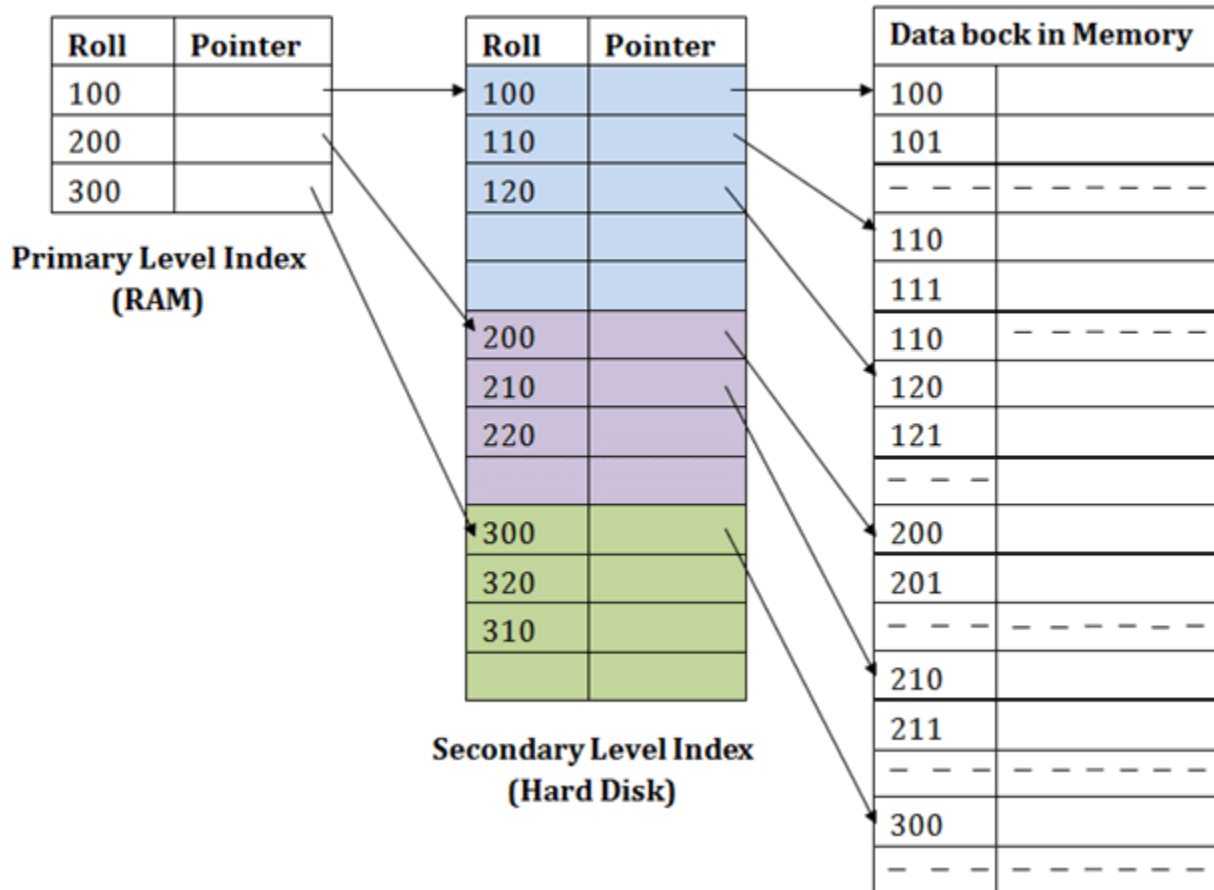
ودا مثال تانی علی شکل ال indexing لو مش clustered



مثلا عندنا ال category id دا مميز بيبقى خاص بال category بس ومينفعش يتكرر ولكنه مربوط بيه كدا products بمعنى انه بيشاول على قيم كثير فا هو كا قيمة فى حد ذاته Unique لكنه القيم اللى بيشاول عليها متعددة مش قيمه واحده بس

Secondary

هنا ال indexing هيكون على مرحلتين وهما التخزين فى ال ram ال index ودا اسمه ال primary level index والتخزين على ال drive ودا اسمه ال scndary level index وبيكون بالشكل دا



تقدر توصفها انا شبه ال multidimensional arrays لانها بتبقى address بيشاور على index فى table تانيه بتشاور على data

نيجى بقى لل بسؤال التانى معانا ايه هو ال character set و ايه ال collation

ال character set هي باختصار اخنا دلوقت عندنا لغات كثير صح زى الصينى و العربى و اللاتينى والانجليزى ال character set دى هي الطريقة بتاعت ال encode اللى بتستخدمها ال database فى تخزين اللغات المختلفه فا هيا encode techniques

اما بقى ال collation فا هيا الشروط اللى بيحصل بيها التشفير دا ونقد نستعمل التشفير او ال encode دا على اربع مستويات

Database / table / server / column

مثال على ال encode دلوقت لو عندنا شوية حروف وليكن a,b,c,A,B

و هنعرفهم بارقام بالشكل دا

$a = 1 / b = 2 / c = 3 / A = 4 / B = 5$

فا كدا ال a بقى الرمز بتاعها = 1 و ال B = 5 و هكذا و الشروط اللي بتتحكم فى التخزين والمقارنة بين الحروف بتاعة ال encode هاي ال collation و فى كام عبارته كدا نعرفهم من باب العلم بالشئ فى ال collation

Ci —> case insensitive

Cs —> case sensitive

Bin —> binary

ولازم نعرف ان كل character set بيكون ليها collation واحد او اكثر وبيكون ليها على الاقل واحد default

مثال على ال character set and collation

Utf8 —> is a character set

Utf8_general_ci —> is utf8 collation

عشان تعرف بيانات عن ال character sets بتستخدم sql statement

SHOW CHARACTER SET;

و دا لينك فى كل الاوامر اللي تقدر تتعامل بيها مع ال character set and collation

<https://www.geeksforgeeks.org/what-is-collation-and-character-set-in-mysql/>

Authentication vs authorization

نبدأ بال authentication ودى هيا الخطوه اللي الموقع بيعرف فيها هل العميل دا متسجل على الموقع دا ولا لا ومين العميل دا بمعنى هل العميل دا ليه بيانات متسجله على الموقع ولا دا اكونت مش موجود اما ال authorization هيا الصلاحيات اللي بيملكها ال user دا فى الموقع يعنى ايه بص يعنى مثلا انا لو عندى موقع الكترونى وعندى عمال شغالين فيه و users بيدخلو الموقع الحاجات اللي يقدر يعملها ال user فى الموقع مش زى العامل مثلا او زى صاحب الموقع العامل يقدر يضيف product جديد لكن ميقدرش يمسح product وال user يقدر يشتري بس ويتصفح لكن ميقدرش يعدل على ال products الموجوده

Database join

ايه هو ال join هشرحها الاول بمثال بسيط دلوقت انا لو عندي orders table و users table وعايز اجيب البيانات المشتركة بين الجدولين والبيانات دي هيا ان ال user_id عمل ال order عمل ال order اللى ال id بتاعه كذا يعنى مع كل order او row فى ال orders table فى id بتاع ال user اللى عمل ال order انا بقى لو عايز اعرض البيانات المشتركة بين ال tables دي هستعمل ال join وهيا ببساطه بتقول لل database تجيب البيانات المشتركة بين الجدولين طيب الربط دا بيتم ازاي ؟ بس فى ال database فى حاجتين لازم تعرفهم وهما ال pk / fk طيب ايه هما دول

Pk —> primary key

Fk —> foreign key

الpk دا الضهر اللى بيسند الجدول منغيره الجدول بيقى ملغبط باختصار هو حاجه مبتكررش قيمتها فى الجدول اما ال fk فا دا بياخد قيمته من ال pk بتاع الجدول اللى انت ربطه معاه

طيب ايه انواع ال join فى 3 انواع لل join وهما

Inner / natural / outer

نبدأ ب natural

هنا البيانات اللى بتجيبها بين ال table هيا ال column اللى شبيها فى ال data type / column name وهنا النتيجة بيكون فيها نسخه واحده بس لل columns المتكرره يعنى لو عندي اثنين columns بنفس الاسم هيجبلى واحده منهم بس فى النتيجة لان الاثنين واحد

Syntax

```
SELECT *  
FROM Student NATURAL JOIN Marks;
```

```
SELECT *  
FROM student S INNER JOIN Marks M ON S.Roll_No = M.Roll_No;
```

النوع الثاني وهو ال inner join

هنا انت بتحدد العمود المشترك بين الجدولين وتحدده بعد عبارة on او where وهنا ال table بتاعت النتيجة هترجع كل الاعمده اللي انت بتحددها حتى لو شبه بعض يعنى لو فى ايتين column الاتتين user_id هتجيبهم فى ال نتيجة طالما حدثهم مش هتجيب نسخه واحده منهم زى ال natural وهنا الصفوف اللي بترجع او ال rows بيكون فيها الداتا اللي بين الجدولين بشكل فعى يعنى لو فى مثلا user فى ال table الاولى ملوش بيانات فى ال table الثانيه مش هتظهرله بيانات فى النتيجة

Syntax

```
select *  
from table1 INNER JOIN table2  
on table1.column_name = table2.column_name;
```

```
select *  
from table1 JOIN table2  
on table1.column_name = table2.column_name;
```

النوع الثالث هو ال outer join

ودا هنا بيرجع البيانات اللي الجدولين كلها سوا ليها matching او لا يعنى هيجيب البيانات المرتبطه بين الجدولين ولو مفيش بيانات مرتبطه هيجيب باقى البيانات برده

Syntax

```
select *  
from table1 LEFT OUTER JOIN table2  
on table1.column_name = table2.column_name;
```

```
select *  
from table1 RIGHT OUTER JOIN table2  
on table1.column_name = table2.column_name;
```

```
select *  
from table1 FULL OUTER JOIN table2  
on table1.column_name = table2.column_name;
```

Case and when

هي الحالة لما يتحقق نفذ شرط مغين على اساسها وتقدر تقول انها شبه match او switch في php

Syntax

```
SELECT FacultyID, Name, Department,  
CASE Gender  
  WHEN 'M' THEN 'Male'  
  WHEN 'F' THEN 'Female'  
END  
FROM Faculty
```