

MONGO DB

مونگو دیبی (Mongo DB) یکی از معروف ترین پایگاه داده های No SQL است که ساختار منعطفی دارد و بیشتر در پروژه هایی با حجم بالای داده استفاده می شود.

این پایگاه داده پلتفرمی متن باز و رایگان است و با مدل داده های مستند گرا (Document - Oriented) کار می کند و در ویندوز، مکینتاش و لینوکس قابل استفاده است. مقادیر داده ای ذخیره شده در مونگو دیبی، با دو کلید اولیه (Primary Key) و ثانویه (Secondary Key) مورد استفاده قرار می گیرند.

مونگو دیبی به علت مستند گرا بودن مدل ذخیره داده ها در مقایسه با دیتابیس های رابطه ای بسیار منعطف تر و مقیاس پذیرتر است و بسیاری از نیازمندی های کسب و کارها را برطرف می کند.

این پایگاه داده برای تقسیم داده ها و مدیریت بهتر سیستم از شاردینگ (Sharding) استفاده می کند. شاردینگ به معنی تکه تکه کردن است و در لود بالای شبکه انجام می شود. به گونه ای که دیتابیس به چند زیربخش تقسیم می شود تا روند پاسخ دهی به درخواست هایی که از سمت سرور می آید، راحت تر شود.

داده ها با دو کلید اولیه و ثانویه قابل دسترسی هستند و هر فیلدی قابلیت کلید شدن را دارد. این امر زمان دسترسی و پردازش داده را بسیار سریع می کند.

همانند سازی (Replication) یکی دیگر از خصوصیات مهم مونگو دیبی است. در این تکنیک از یک داده به عنوان داده اصلی کپی هایی تهیه شده و بخش های دیگری از سیستم پایگاه داده ذخیره می شود. در صورت از بین رفتن و یا مخدوش شدن این داده، داده های کپی شده به عنوان داده اصلی و جایگزین مورد استفاده قرار می گیرند.

مزیت های MONGO DB

این پایگاه داده سندگرا است بنابراین برخلاف پایگاه داده های رابطه ای، اسناد شبیه به جی سون را مدیریت می کند. این اجازه می دهد که مدل داده های برنامه بیشتر به شکل طبیعی شان نزدیک باشد به این خاطر که داده ها می توانند در سلسله مراتبی پیچیده و تودرتو قرار بگیرند اما قابلیت اجرای درخواست و شماره بندی شان را از دست ندهند.

توضیح ساده تر آنکه در دیتابیس هایی که ما معمولا با آن سروکار داشتیم، داده ها در جدولی ذخیره می شد که محدودیت ۲ بُعدی داشت یعنی فقط در سطر خاص و فیلد (ستون) خاص عملیات درج / ویرایش / حذف / انتخاب انجام می شد اما در مونگو اینگونه نیست! شما میتوانید در جدول user فیلد name را برابر با علیرضا بگذارید و فیلد address را همانند یک ورودی چند بعدی دیگر پر کنید...

همچنین همانطور که ابتدای پاسخ گفته شد، برای insert شما محدودیتی ندارید. یعنی نه لازم است ساختاری از قبل بچینید که بعد داخل آن ساختار insert کنید و نه ورودی شما محدود به عدد و رشته و داده باینری است. ورودی شما می تواند یک آرایه باشد!

به عبارت دیگر برای کار با مونگو نیاز نیست ستون های جدول را قبل از کار با بانک تعریف کنید بلکه در هر لحظه داده های خود را با هر شکلی می توانید در پایگاه داده وارد کنید ولی لازم است دقیقا بدانید چه کار قرار است انجام دهید زیرا اگر داده های یکسان در هر سند به اشکال کاملا متفاوتی ذخیره شوند و شما ندانید آن ها را چگونه ذخیره کرده اید، بازآوری اطلاعات برای پردازش و نمایش سخت می شود.

RETHINK DB

اولین پایگاه داده منبع باز و مقیاس پذیر json است که برای سایت های بدون انتظار استفاده میشود. این ساختار معماری پایگاه داده سنتی را با قرار دادن یک مدل دسترسی جدید هیجان انگیز وارونه می کند. به جای نظر سنجی برای تغییرات ، توسعه دهنده می تواند به RethinkDB بگوید که نتایج جستجوی به روز شده را به طور مداوم به برنامه ها بفرستد.

RethinkDB اولین پایگاه داده منبع باز و مقیاس پذیر است که به طور خاص برای انتقال داده ها به برنامه ها در زمان واقعی طراحی شده است. این به طور چشمگیری زمان و تلاش لازم برای ساخت برنامه های مقیاس پذیر در زمان واقعی را کاهش می دهد.

در واقع RethinkDB به کاهش چشمگیر زمان و تلاش برای ساختن برنامه های واقعی کمک میکند.

RethinkDB یک زبان جستجوی انعطاف پذیر ، عملیات بصری و نظارت بر API ها ارائه می دهد و راه اندازی و یادگیری آن آسان است.

چه موقعی انتخاب RethinkDB بهترین گزینه است؟

زمانی انتخاب خوبی است که برنامه شما از پاسخ های بیدرنگ بهره مند شده باشد.

مدل دسترسی به پایگاه داده با استفاده از نظرسنجی یا پرس و جو در وب به خوبی کار میکند زیرا مستقیماً به درخواست HTTP پاسخ میدهد ولی با این حال برنامه های مدرن نیاز به ارسال داده به طور مستقیم به مشتری در زمان واقعی دارند مانند موارد ذیل:

برنامه های مشارکتی وب و موبایل

برنامه های تجزیه و تحلیلی

بازی های چند نفره

دستگاه های متصل به هم

به عنوان مثال ، هنگامی که کاربر موقعیت یک دکمه را در یک برنامه طراحی مشترک تغییر می دهد ، سرور باید سایر کاربران را که به طور همزمان روی یک پروژه کار می کنند ، مطلع کند.

چند مثال از شرکت های مختلفی که از این دیتابیس استفاده کردند:

✱ Jive Software از RethinkDB برای تأمین انرژی وب و برنامه های تلفن همراه واکنشی استفاده می کند

✱ Narrative Clip و Pristine.io از RethinkDB برای تأمین زیرساخت ابر برای دستگاه های متصل استفاده می کنند

✱ Platzi و Workshape.io برای تأمین تجزیه و تحلیل بی درنگ از RethinkDB استفاده می کنند

✱ NodeCraft و GameServerKings از RethinkDB برای تأمین انرژی بازیهای چند نفره با مقیاس گسترده استفاده می کنند

پیاده سازی معماری فشار فوری در زمان واقعی نیاز به طراحی مجدد بیشتر اجزای پایگاه داده ، از جمله موتور اجرای درخواست ، سیستم توزیع شده ، زیر سیستم ذخیره سازی و موتور ذخیره سازی دارد. از آنجا که این معماری بر روی هر جز component پایگاه

داده تأثیر می گذارد ، RethinkDB از ابتدا در C ++ پیاده سازی شده است. RethinkDB طی ۵ سال توسط تیمی از کارشناسان پایگاه داده با کمک صدها نفر از مشارکت کنندگان از سراسر جهان ساخته شد.

COACH DB

دیتابیس Apache CouchDB توسط بنیاد نرم افزاری Apache توسعه داده شده و با زبان Erlang نوشته شده است. یک دیتابیس متن باز که از فرمت ها و پروتکل های مختلفی برای ذخیره، انتقال و پردازش داده های خود استفاده می کند.

از JSON برای ذخیره داده ها، از زبان JavaScript به عنوان زبان کوئری با روش پردازش MapReduce و از HTTP برای API استفاده می کند. داکيومنت ها، واحدهای اصلی این دیتابیس هستند و metadata را شامل می شوند. فیلدهای داکيومنت، به صورت منحصر به فرد نامگذاری شده اند و مقداری با نوع های مختلف را شامل می شوند، هیچ محدودیتی برای اندازه متن یا تعداد عناصر وجود ندارد.

ویژگی های دیتابیس :

Replication: این دیتابیس ساده ترین شکل replication را ارائه می دهد، هیچ دیتابیس دیگری در این مورد به این اندازه ساده نیست.

Document Storage: این یک دیتابیس NoSQL است که از document storage پیروی می کند و در آن هر فیلد به طور اختصاصی نامگذاری می شود و مقداری از انواع داده های مختلف مانند متن، اعداد، Boolean، لیست ها و ... در آن قرار می گیرند.

خصوصیات ACID: طراحی فایل CouchDB از تمام خصوصیات ACID پیروی می کند.

امنیت: همچنین امنیت در سطح دیتابیس را ارائه می کند و مجوزها به readers و admins تقسیم می شود، که خوانندگان می توانند عملیات خواندن و نوشتن در دیتابیس را انجام دهند.

Map/Reduce: دلیل اصلی محبوبیت CouchDB استفاده از سیستم پردازشی Map/Reduce است.

احراز هویت: CouchDB در حفظ احراز هویت از طریق سشن و کوکی مانند یک برنامه تحت وب، به ما کمک می کند.

HTTP API: همه موارد دارای یک URI (Unique Resource Identifier) منحصر به فرد هستند که از طریق HTTP نمایش داده می شوند. از متدهای HTTP مانند POST, GET, PUT و Delete برای چهار عملیات CRUD (Create, Read, Update, Delete) استفاده می کند.

معایب دیتابیس:

CouchDB فضای زیادی برای overhead اشغال می کند، که یکی از مهم ترین معایب این دیتابیس در مقایسه با سایر دیتابیسی ها به شما می رود.

CouchDB در مقایسه با بقیه دیتابیسی ها، overhead بیش تری ایجاد می کند.

ممکن است **Temporary view** ها در مجموعه داده‌های عظیم بسیار کند باشند.

ممکن است **Replication** در دیتابیس‌های بزرگ با شکست روبرو شود.

DYNAMO DB

Amazon DynamoDB یک سرویس پایگاه داده کاملاً مدیریت شده **NoSQL** است که عملکردی سریع و قابل پیش بینی را با مقیاس پذیری یکپارچه فراهم می کند.

این پایگاه داده به صورت ویژه برای برنامه‌هایی که به سرعت بالا و تأخیر بسیار پایین نیاز دارند توسعه داده شده است.

با توجه به توسعه و پشتیبانی این پایگاه داده توسط **Amazon**، **DynamoDB** به صورت کاملاً آنلاین و ابری تحت سرویس‌های وب **Amazon** در دسترس کاربران قرار می گیرد.

مدل‌های داده‌ای قابل انعطاف، عملکرد قابل اعتماد و سرعت بسیار بالا، این پایگاه داده را برای اکثر برنامه‌های کاربردی روز از جمله بازی‌ها، خدمات وب، اینترنت چیزها و غیره مناسب ساخته است.

با استفاده از **DynamoDB**، می توانید جداول پایگاه داده ایجاد کنید که می تواند هر مقدار داده را ذخیره و بازیابی کند و هر سطح از ترافیک درخواست را ارائه دهد.

شما می توانید **backup** ایجاد کرده و بازیابی نقطه به زمان را برای جداول **Amazon DynamoDB** خود فعال کنید. بازیابی به موقع به شما کمک می کند جداول شما در برابر عملیات نوشتن یا حذف تصادفی محافظت شود.

DynamoDB به شما امکان می دهد موارد منقضی شده را به طور خودکار از جداول حذف کنید تا به شما در کاهش استفاده از فضای ذخیره سازی و هزینه ذخیره اطلاعاتی که دیگر مربوط نیستند، کمک کند.

DynamoDB به طور خودکار داده ها و ترافیک جداول شما را بر روی تعداد کافی سرور پخش می کند تا بتواند نیازهای فضای ذخیره سازی و ذخیره سازی شما را برطرف کند، در حالی که عملکرد ثابت و سریع را حفظ می کند.

POSTGRES SQL

PostgreSQL یک سیستم مدیریت پایگاه داده می باشد که در فارسی به صورت پست گرس.کیو.ال تلفظ می شود و به راحتی محبوبیت خود را با دیتابیس **MySQL** به اشتراک می گذارد. این **DBMS** به عنوان یک سیستم مدیریت پایگاه داده رابطه ای شیئی به حساب می آید که در آن اشیاء تعریف شده توسط کاربر و رویکرد جدول برای ساخت ساختارهای داده پیچیده تر ترکیب می شوند.

طراحی **PostgreSQL** به صورت یکپارچه انجام می شود، اما طراحی **MySQL** به شکل ۲ لایه (لایه اول لایه ی **SQL** بوده و لایه دوم **storage engine**) می باشد.

یکی از بخش های مشترک در **PostgreSQL** و **MySQL**، سرعت می باشد. نحوه بهبود سرعت در این ۲ به دلیل الگوریتم های متفاوتی که دارند با هم فرق می کنند.

سهولت در استفاده از PostgreSQL، برای کاربران حرفه ای و سهولت استفاده از Mysql برای کاربران عادی نیز می باشد.

Postgresql با استفاده از ساختار ACID، از امنیت بیشتری نسبت به Mysql برخوردار است.

موارد کاربرد و ویژگی های PostgreSQL به صورت زیر می باشند:

نمایش داده های پیچیده و انتخاب گسترده ای از رابط های سفارشی.

تجزیه و تحلیل داده ها و انبارداری.

ساختن یک ابزار اتوماسیون بانک اطلاعاتی.

قابلیت های تحلیلی قوی.

سازگاری با ACID و موتور SQL قدرتمند.

تعریف نوع داده توسط خود کاربر.

کنترل همزمان چند نسخه (MVCC)

وراثت جدولی (Table inheritance)

MARIADB

به احتمال زیاد، نام پایگاه داده MySQL حداقل یک بار هم که شده به گوشتان برخورد کرده است. در سال ۲۰۰۹ طراحان و توسعه دهندگان اصلی MySQL یک دیتابیس جدید را بر مبنای همین MySQL ایجاد کردند و نام آن را MariaDB یا Maraia DataBase نامگذاری کردند. این پایگاه داده هم به شکل متن باز و رایگان منتشر شده است که می توان از آن استفاده کرد.

ویژگی های MariaDB ، پایگاه داده مطمئن

زمانی که شما از یک پایگاه داده استفاده می کنید، اطلاعات پروژه شما مثل رمز عبور یا اطلاعات مخاطب هایتان روی یک سری جدول ذخیره سازی می شود. زمانی که شما می خواهید از این داده ها استفاده کنید، با استفاده از این جدول ها، داده بازیابی می شود. حال اگر بتوانید از موتورهای ذخیره سازی استفاده کنید، قطعاً بازیابی اطلاعات شما بهتر و سریع تر اتفاق خواهد افتاد.

نکته ای که باید بگوییم این است که امروزه در MariaDB موتورهای ذخیره سازی یا Storage Engine، توسعه خوبی داشته اند که هر کدام از این ها می توانند امکانات جدیدی را به ما بدهند. از جمله این موتورهای افزوده شده می توان به XtraDB و Aria اشاره کرد.

امنیت MariaDB امروزه از MySQL بیشتر است. با این که MariaDB در واقع زیر مجموعه ای از همین MySQL است، اما به نظر می رسد در توسعه آن اصول امنیت پایگاه داده بیشتر رعات شده است. پس در استفاده از MariaDB نگران امنیت خود نباشید.

استفاده از امکانات می تواند باعث سهولت و لذت بخش شدن کار شود و MariaDB قطعاً توانسته با امکاناتی فراتر از MySQL، مورد توجه طرفداران خودش در جهان شود. امکاناتی همچون Multi Source Replication یا Dynamic column که شاید در آینده در MySQL و دیگر پایگاه های داده، اضافه شوند اما به هر حال MariaDB امروزه پیشگام در زمینه استفاده از این ویژگی ها است که می تواند نظر هر فردی را به خودش جلب کند.

پایگاه داده MariaDB کاربران مهمی را مانند **Wikipedia**، **Fedora**، **Open SUSE**، **Mozilla** و ... به سمت خود کشانده و با توجه به پشتیبانی خود شاید خیلی زود به چهره اول پایگاه داده در لینوکس و حتی ویندوز تبدیل شود.

بطور اختصار میتوان MariaDB را نسخه بهینه یافته **MySQL** نامید چون در عملکرد بسیار بهینه تر میباشد و همچنین امنیت بهتری دارد و در پاسخگویی به کوئری ها سریعتر عمل میکند و مهم تر همه اینکه منابع کمتری نیز مصرف میکند پس برای وب سایت های پر بازدید و هاستینگ هایی که تعداد زیادی وب سایت میزبانی میکنند بسیار مطلوب خواهد بود.

MSSQL

MSSQL یا **Microsoft SQL Server** یک دیتابیس سرور مدل رابطه ای تجاری (غیر رایگان) که توسط مایکروسافت ارائه شده و توسعه داده شده می شود که از دیتابیس سرور های محبوب بر روی سیستم عامل ویندوز می باشد و برای استفاده از این نرم افزار باید لایسنس آن از مایکروسافت تهیه شود، این دیتابیس سرور یک محصول می باشد که می تواند روی یک لپ تاپ تا یک شبکه بزرگ با سرور های قدرتمند استفاده شود.

MSSQL برای نرم افزار های تحت وب که روی پلت فرم ویندوز یا با **NET** و **ASP** نوشته شده اند استفاده می شود.

ORACEL

۹۷ شرکت از ۱۰۰ تا کمپانی موجود در لیست برترین کمپانی های مجله **Fortune**، از دیتابیس اوراکل استفاده می کنند! همین موضوع عظمت این کمپانی و سیستم پایگاه داده آن را نشان می دهد. دیتابیس ها انواع مختلفی دارند و اوراکل همه آن ها را به مشتریانش (بسته به نیازی که دارند) ارائه می کند.

نکته: برای مدیریت پایگاه داده، یک سیستم مدیریتی مخصوص لازم داریم. به این سیستم **Database Management System** می گویند؛ اما نه انقدر طولانی! نام اختصاری آن **DBMS** است.

ویژگی های دیتابیس:

این دیتابیس کراس-پلتفرم است؛ یعنی روی پلتفرم های مختلف قابل اجرا است و می تواند روی سخت افزارها و سیستم عامل های مختلف مثل ویندوز، یونیکس و توزیعات مختلف لینوکس مورد استفاده قرار گیرد.

پایگاه داده اوراکل به پشتوانه ساختار شبکه ای، امکان برقراری ارتباط بین اپلیکیشن های مختلف از پلتفرمی دیگر را فراهم می کند. کمی پیچیده است! برای مثال اپلیکیشنی که روی ویندوز اجرا شده است، می تواند به پایگاه داده اوراکلی که روی لینوکس قرار دارد متصل شود.

این دیتابیس با **ACID** سازگار است. به این ترتیب استفاده از اطلاعات به صورت یکپارچه و مطمئن ممکن می شود.

ساختار منطقی داده ها: با داشتن این ویژگی، می توانید بدون دانستن مکان فیزیکی داده با دیتابیس ارتباط برقرار کنید.

پارتیشن بندی: این ویژگی کمک می کند جدول های بزرگ - در پایگاه داده رابطه ای - را به بخش های کوچک تر تقسیم کنید و هر قسمت را در بخشی از هارد ذخیره کنید.

کش مموری: این قابلیت باعث می شود با بزرگتر شدن پایگاه داده، سرعت عمل آن دچار مشکل نشود.

دیکشنری داده ها: با استفاده از این ویژگی که نوعی جدول داخلی است، اداره دیتابیس راحت تر از همیشه می شود.

بکاپ و ریکاوری: داشتن این قابلیت برای دیتابیس ها از واجبات است. پایگاه داده اوراکل هم با برخورداری از این ویژگی، از تمام اطلاعات و داده ها انواع بک آپ ها را می گیرد تا در صورت بروز مشکل بتواند بعداً آن ها را ریکاوری کند. این کار توسط اپلیکیشنی قدرتمند با نام RMAN انجام می شود که مخفف Recovery Manager است.

دسته بندی: این قابلیت اوراکل توسط اپلیکیشنی با نام RAC اجرا می شود. وظیفه این اپلیکیشن اطمینان از درستی عملکرد کل سیستم است. حتی در زمانی که یک یا چند سرور با مشکل مواجه شده باشند.

بررسی چند مفهوم در دیتابیس ها:

ORM عملاً یک لایه مترجم بین زبان برنامه نویسی و پایگاه داده رابطه ای است که این دو را به هم تبدیل می کند.

به عنوان مثال برای دریافت اطلاعات مربوط به یک کاربر خاص باید از دستور SQL زیر استفاده می کردیم.

```
SELECT * FROM users where name = 'Programchi';
```

شیوه جدید ORM باعث می شود تا برنامه نویس دیگر با چنین کدهای پیچیده ای دست و پنجه نرم نکند و تنها با اجرای یک دستور اطلاعات مربوط به کاربران را به طور مثال دریافت کند.

نمونه کدهای ORM در زبان PHP

```
$roka = User
```

```
::where( 'name', 'programmer' )
```

```
->with( array( 'Bills', 'Ads', 'Item' ) )
```

```
->first();
```

همانطور که مشاهده شد با شیوه جدید ORM تنها با چند خط می توانید شاید بیش از ۵۰ خط یک Query را بدون دانش SQL ایجاد کرد و از خوبی های ORM می توان پشتیبانی از ارتباط دیتابیس پیچیده را نیز نام برد مهم نیست چه نوع رابطه هایی وجود داشته باشد حتماً برایش راهی وجود خواهد داشت.

ACID

در علم کامپیوتر، تراکنش ها باید خاصیت زیر را داشته باشند

هر تراکنش باید مطلق باشد (همه یا هیچ چیز) باشد (A) Atomicity.

باید محدودیت های رخ داده شده در سیستم را دنبال کند و سیستم را از یک حالت پایدار به حالت پایدار دیگر ببرد (C) Consistency

باید مستقل باشد و برای سایر تراکنش ها قابل مشاهده باشد (I) Isolation

پس از تکمیل شدن به طور مسلم باید یک وضعیت جدید در سیستم ایجاد کند ، یعنی باید پس از انجام حالت سیستم طبق قوانین پایگاه داده تغییر کند(Durability)

به طور کلی خواص اسیدی برای انجام کارهای همزمان (concurrency) بسیار مناسب است. اگر عملیاتی که در سیستم انجام میشوند همزمان رخ میدهند با خواص اسیدی مطمئن میشوید که هیچ یک از تراکنشها در یکدیگر اختلال پیش نمی آورد و همچنین از بابت صحت عملیاتها خیالتان آسوده خواهد بود.

transaction (ترنزاکشن) عملیاتی می باشد که به شما اجازه می دهد تا از صحت انجام یک فرآیند مطمئن و از درج ناقص اطلاعات جلوگیری کنید.

تصور کنید شما قصد انجام یک سری عملیات حساس روی دیتابیس (حذف- افزودن- ویرایش) دارید و باید مطمئن باشید که فرآیند تغییرات در SQL یا دیتابیس شما به درستی انجام شده است. برای اینکار شما میتوانید از قابلیت transaction استفاده کنید.

برای مثال ممکن است برای طراحی سایت بخواهیم تصویر کاربر را هنگام ثبت نام از وی دریافت نماییم و با شماره‌ی شناسه‌ی کاربری روی دیسک ذخیره کنیم. برای پیشگیری از ذخیره شدن اطلاعات بدون تصویر می توانیم ذخیره سازی اطلاعات کاربر را درون یک ترنزاکشن انجام دهیم، سپس تصویر را ذخیره کنیم و به پروفایل نسبت دهیم. اگر ذخیره سازی تصویر با مشکل مواجه شد، کافی است ترنزاکشن را به عقب برگردانیم تا تمامی اطلاعات ذخیره شده از بین بروند و خطای لازم را به کاربر نشان دهیم.

Database transactions شامل سه بخش می باشد:

۱- Creating a transaction - اجازه دادن به دیتابیس برای استفاده از ترنزاکشن

۲- Rolling back a transaction - زمانی که عملیات با موفقیت انجام نشده باشد. تمام تغییرات اعمال شده باید برگشت داده شود.

۳- Committing a transaction: به این معنی می باشد که عملیات با موفقیت انجام و تایید شده اند.

مشکل $n+1$ زمانی در برنامه نویسی زمانی رخ می دهد که یک نرم افزار با دیتابیس ارتباط دارد ، این بدان معنا است که برای یک درخواست دفعات مختلفی به دیتابیس کوئری ارسال می شود.

قطعا باید خیلی مراقب بود تا برای یک درخواست فقط یک کوئری به دیتابیس ارسال مگر در مواردی که دیگر امکان پذیر نیست.

قطعا عدم آشنایی با مشکل $n+1$ می تواند مشکلاتی در نرم افزار شما ایجاد کند.

برای دریافت اطلاعات از دیتابیس دو روش داریم Lazy loading و Eager loading

توی روش اول میایم به دیتا رو میگیریم و بعدا اگه لازم شد میایم دیتا های مرتبط با اون رو هم میگیریم.

توی روش دوم میایم به دیتا رو به همراه دیتا هایی که با اون دیتا relation داره رو میگیریم.

مشکلی که در حقیقت مشکل نیست. بلکه وقتی تعداد یوزرای سایت زیاد میشه خودش رو نشون میده. معنی $n+1$ رو تو این حالت بهتون توضیح میدم که شما بجای یه کوئری به دیتابیس برای CRUD مورد نیازت چند تا ریکوئست میزنی.

فرض کنید که شما تو یه وبلاگ میخوای پست رو به همراه تمام تگ هاش و دسته بندی هاش به کاربر نمایش بدی، چکار میکنی؟ **Lazy loading** یا **Eager loading**. چیزی که مشخصه اینکه شما اگه بخوای **Lazy loading** بکنی باید ۳ تا ریکوئست بزنی تا پست ها، دسته بندی ها و تگ های اون پست رو از دیتابیس بگیری. و این عملیات IO خیلی وحشتناک خودش رو تو یه چیزی مثل فیس بوک خودش رو بخوبی نمایش میده.

SaladAD