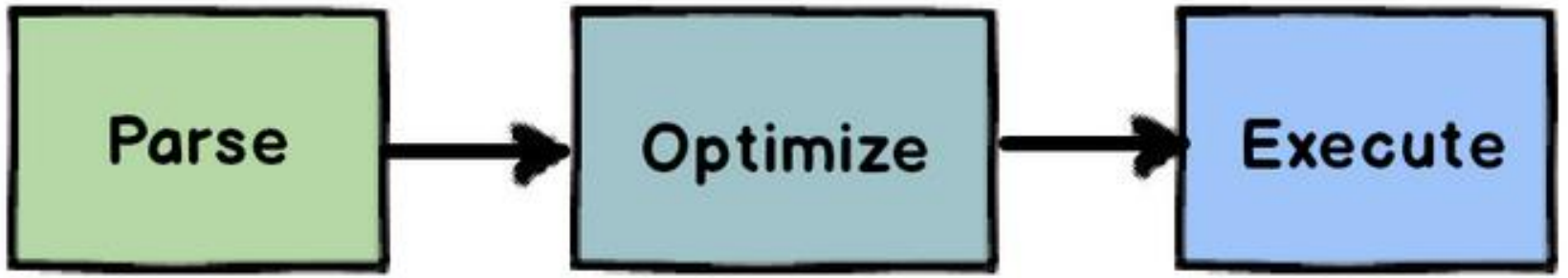
Tema 8:
OPTIMITZACIÓ DE CONSULTES

1er Desenvolupament d'aplicacions multiplataforma

Què pot afectar al rendiment d'una consulta?

- El tamany de les taules consultades (Núm de files)
- El nombre de taules combinades
- El nivell de selectivitat d'una consulta (WHERE...)
- Quantitat de files a ordenar (ORDER BY – GROUP BY
- Existència d'índex o no
- Qüestions físiques.
  - Rendiment de l'equip
  - Rendiment de la connexió

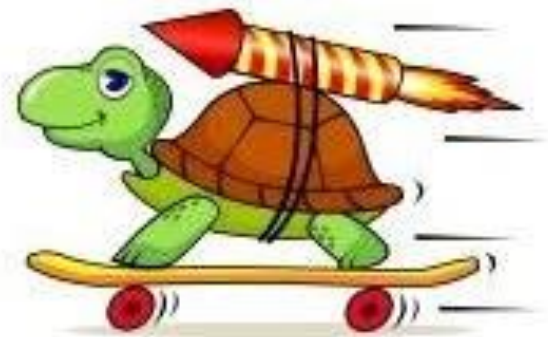
PASOS QUE ES PRODUEIXEN EN UNA SQL



PARSE: Analitzador de sintaxi.

OPTIMIZE: Optimitzar operacions i generar un pla d'execució. (com ha d'executar la consulta (Hi ha subselects, vistes? Indexos?)

EXECUTE: Retornar el resultat segons el pla d'execució.



Base de datos no indexada

Base de datos indexada

INDEX

Permeten optimitzar les consultes ordenant la columna que genera el índex de forma alfabètica.

- Els inserts, updates i deletes siguin algo més lents. (60% són selects)
- La primary key es considera com a INDEX.
- Per cada índex que es crea, es genera una nova taula ordenada per l'índex escollit. No s'ha d'abusar dels índex. Ocupen espai de memòria.
- Exemple:

  Sabem que sempre demanarem per nom, caldria crear un índex en el camp nom

```
create table client(
    codi char(2) primary key,
    nom varchar(20)
)engine=innodb;
```

AVANTATGES UTILITZA INDEX

- Disminuir el temps d'execució en consultes amb ordenació (ORDER BY) o agrupament (GROUP BY).
- Si una consulta utilitza una condició simple on la columna és la condició que està indexada, les files seran recuperades directament a partir del índex, sense consultar la taula.
- Disminució dràstica del temps d'execució de consultes de taules de gran tamany. Sinó, NO.

Per què no utilitzem índex a tot?

La creació d'índex també té efectes negatius.

- Els inserts, updates i deletes que es realitzin en alguna taula que tingui índexs, augmentaran el temps d'execució.
- Els índex, generen noves taules i aquestes s'han d'emmagatzemar en algun lloc.

  Per tant, ocupen espai al disc.

¿Quan utilizar-los?

- En les BBDD on sol predominar consultes SELECT amb WHERE davant INSERTS, UPDATES o DELETES.
- Utilitzar índex a columnes amb molta selectivitat, és a dir columnes on el seu valor es repeteixi varis cops.
- No indexar columnes que són modificades freqüentment. UPDATES
- Indexar Claus foranes en casos que el camp sigui utilitzat en WHERE.

En general: una bona configuració dels Índex en una base de dades, sense abusar, és essencial per oferir un bon servei.

OPTIMITZACIÓ CONSULTES

- Importància: (accés seqüencial)

  0,0001 seg x 20.000 usuaris = 2 seg.

  0,0001 seg x 100.000 usuaris = 10 seg. → 5 cops més.

Imaginem un servidor que sol·licita a SGBD 30 consultes per segon…

Com veure el que està passant: EXPLAIN

Mostra el pla d'execució de la consulta (WorkBench no es reflexa bé)

Exemple Necessitat Índex:

Veiem que la majoria de consultes són per ciutat en la clàusula WHERE.

Primary key

CODI	Nom	.....	Ciutat
COD 1	Maria		Barcelona
COD 2	Marcos		Girona
COD 3	Agustín		Madrid
COD 4	Begoña		Barcelona
COD 5	David		Barcelona
COD 6	Santi		Tarragona
COD 7	Ignasi		Reus
...	...		...
COD 1498	Oriol		Tarragona
COD 1499	Pablo		Reus
COD 1500	Angel		Viladecans

Index

Primary key

Ciutat	CODI	Nom	.....
Barcelona	COD 1	Maria	
Barcelona	COD 4	Begoña	
Barcelona	COD 5	David	
Girona	COD 2	Marcos	
Madrid	COD 3	Agustín	
Reus	COD 7	Ignasi	
Reus	COD 1499	Pablo	
...	...	...	
Tarragona	COD 6	Santi	
Tarragona	COD 1498	Oriol	
Viladecans	COD 1500	Angel	

Amb o sense Index

Tenint en compte que hi han 150.000 reg. i busquem el registres de la ciutat de “Barcelona”. Hi han 25. Sabent que cada cerca triga 0.001seg. Quan temps triga en retornar la informació:

- Sense INDEX: 150.000 reg. (accés seq.) X 0.0001 seg/reg = 15 seg.
- Amb Index: 25 reg (accés directe) x 0.0001 seg/reg = 0,0025 seg.

Trigaria 6000 cops més.

Tipus d'index

Per escollir un índex, es important analitzar en les sentencies SELECT quines són les columnes que més utilitzem en la part del WHERE.

Tipus d'índex en MySql

- **Primary Key:** Únic i no permet valors NULL *(només pot haver 1 en tota la taula)*
- **Unique:** valors únics, i permet valors NULL
- **Index:** Permet valors duplicats i valors NULL
- **Fulltext:** Utilitzat sobretot per cerques CHAR, VARCHAR (conté cada paraula del char o varchar)

Index: Sintaxi

La sintaxi per crear índex, és la següent:

CREATE INDEX <indexName> ON <tableName> (columnName);

CREATE INDEX IndexClient on client (nom);

CREATE UNIQUE INDEX<indexName> ON <tableName> (columnName);

Instruccions addicionals

- Per eliminar un índex d'una taula:

  DROP PRIMARY KEY

  DROP INDEX <indexName> on <tableName>

- Per modificar un índex:

  ALTER TABLE <tableName> **ADD** <indexType> [indexName> (column)

- Per mostrar els index:

  SHOW KEYS from <tableName>

REGLES DELS INDEX

- Quan una columna indexada està dins d'una expressió o funció SQL, l'índex NO serà utilitzat.

*SELECT * FROM pedidos WHERE TO_CHAR(fecha_ped, 'DD/MM/YYYY') = '15/02/1991' ;*

*SELECT * FROM pedidos WHERE fecha_ped = TO_DATE('15/02/1991', 'DD/MM/YYYY');*

SELECT DNI FROM Cliente WHERE a-1=b;

SELECT DNI FROM Cliente WHERE a=b+1;

REGLES DELS INDEX

- No s'utilitzarà l'índex, si la columna indexada es comparada per l'operador (!= o NOT)

  *SELECT * FROM clientes WHERE nro_cli != 1000*

  *SELECT * FROM clientes WHERE nro_cli < 1000 OR nro_cli > 1000*

Si l'índex és utilitzat únicament quan el primer caràcter de l'esquerra es diferent de %.

SELECT * FROM clientes WHERE nombre LIKE '%Suarez'

SELECT * FROM clientes WHERE nombre LIKE 'Suarez%'