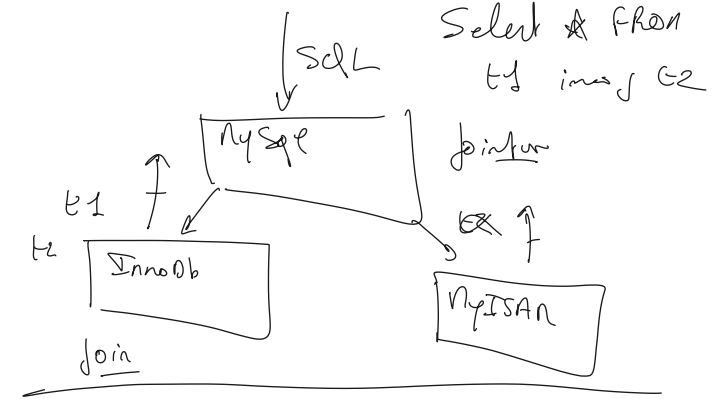
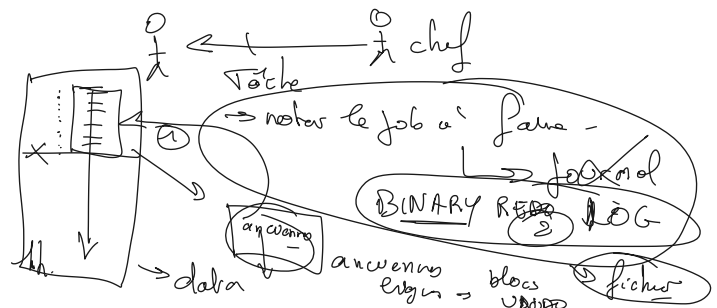


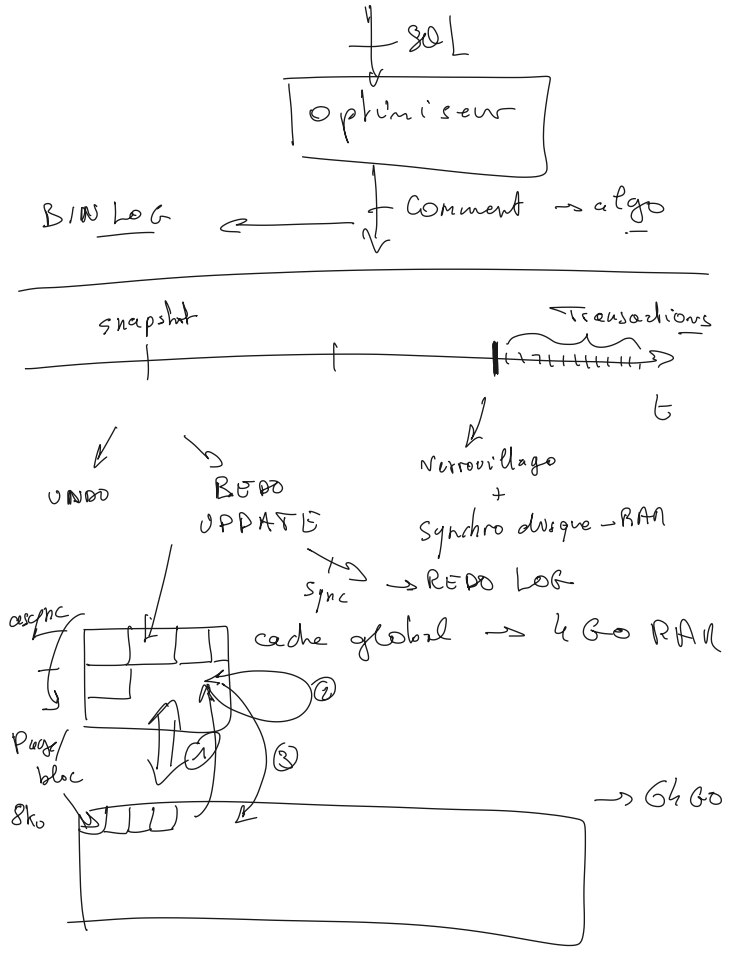
Atomicité

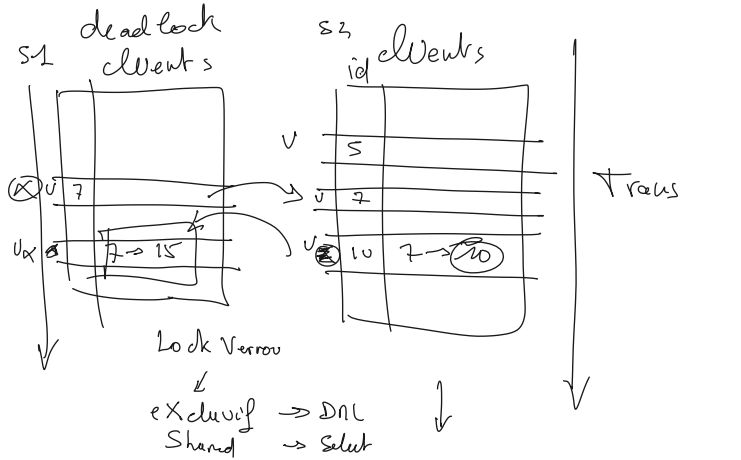
Cohérence

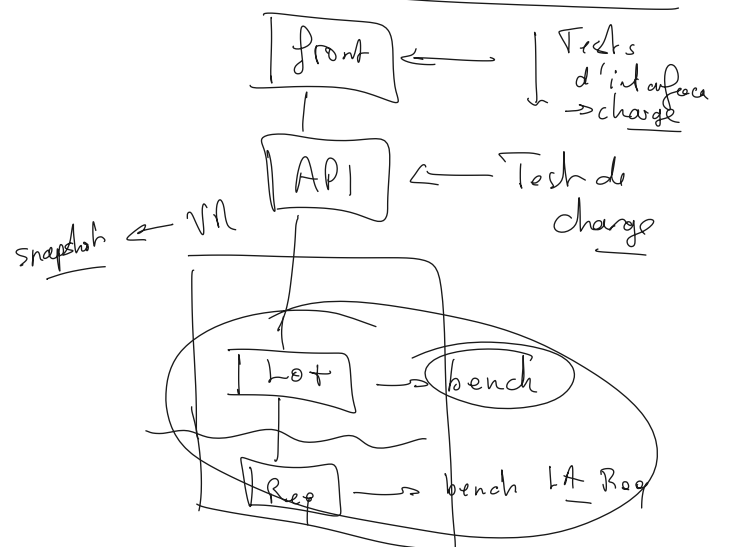
Isolation

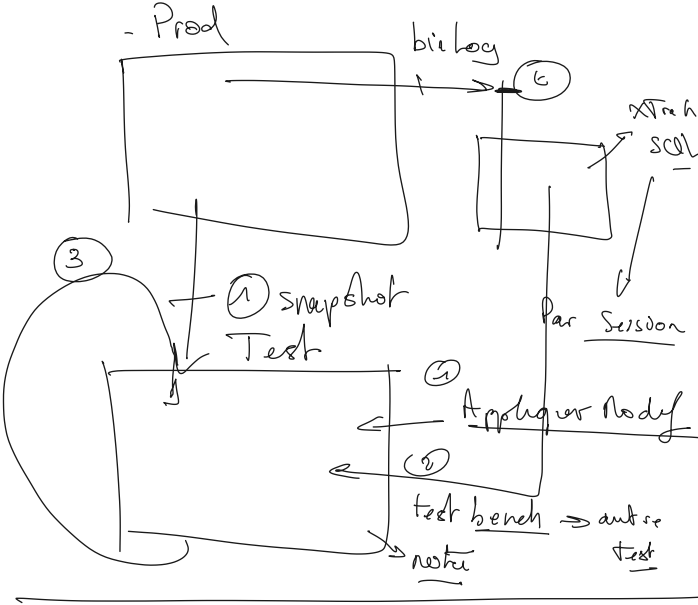
Durabilité



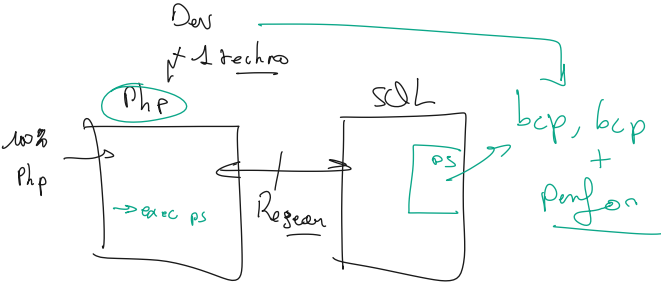

SQL
optimiseur
BIN LOG
Comment → algo
snapshot
Transactions
t
UNDO
REDO
UPDATE
Verrouillage
+
Synchro disque - RAM
sync
→ REDO LOG
async
cache global → 4 Go RAM
Page/
bloc
8ko
→ 64 Go


deadlock
clients
S1
S2
id
clients
V
5
U
7
Ux
7 → 15
U
7
U
10
7 → 10
Trans
Lock Verrou
exclusif → DNL
Shared → Select

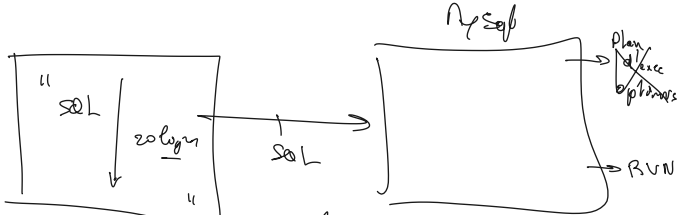

front
Tests
d'interface
→ charge
API
Tests de
charge
snapshot
VM
Lot
bench
Reg
bench LA Reg


- Prod
bin log
6
xTrach
SQL
3
1 snapshot
Test
Par Session
1
Appliquer Modif
2
test bench → autre
test
resto

Proc Stock


Dev
+ 1 techno
Php
SQL
100%
Php
exec ps
Regexp
ps
bcp, bcp
+
perfon

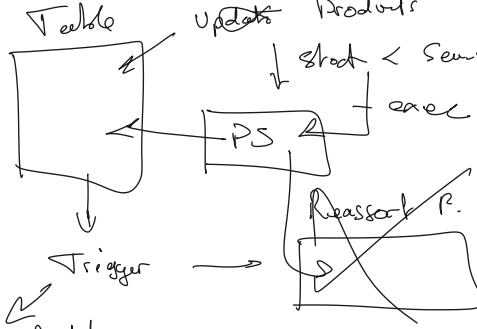
Vue



CREATE VIEW AS SELECT
→ Req " pré Compilé "

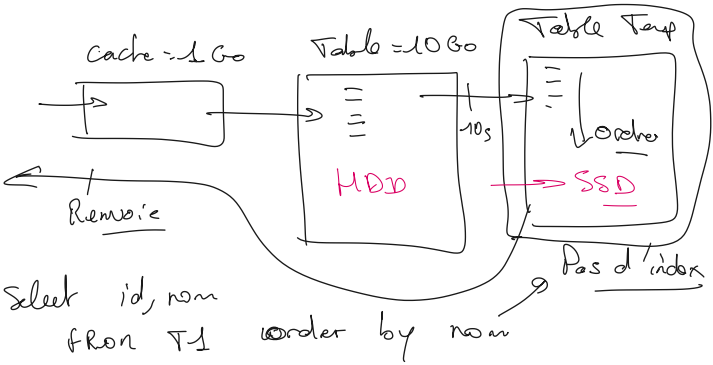
Compilé
↓
Comment → ou.. SANS Index.

Trigger = SOURIS



LOG, Audit ---

Ou



Slow Query LOG :

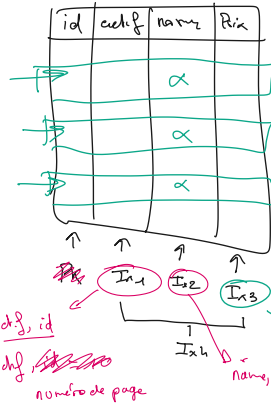
```
set GLOBAL slow_query_log=ON;
set global  log_queries_not_using_indexes=ON;
set global slow_query_log_file='slow.log';
```

Connaître l'état d'une variable (pour notre session tout au moins) :

```
show variables like 'slow_query_log_file';
```

Résultats :

```
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
 Time                 Id Command    Argument
 # Time: 2024-05-27T13:51:11.126884Z
 # User@Host: root[root] @  [172.18.0.3]  Id:    10
 # Query_time: 80.612741  Lock_time: 0.000004 Rows_sent: 9643414
Rows_examined: 9671503
 use geolife;
 SET timestamp=1716817790;
 select * from Activity a inner join TrackPoint t on a.id=t.activity_id;
```

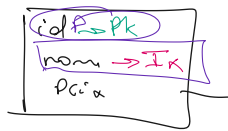


- Select id where actif = true

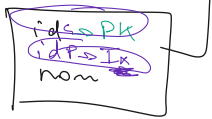
Select id
where name = "dig"

Select name
where prix = 10

Produit

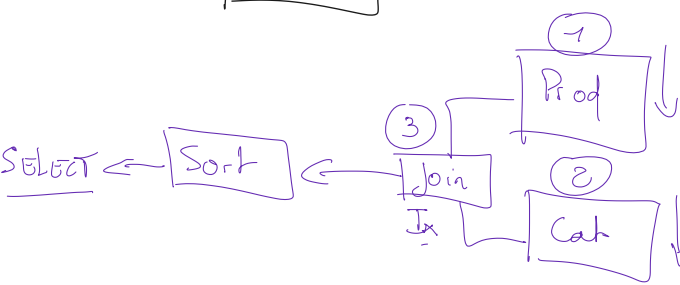


Cat



R

Select
nom p
FROM P
INNER JOIN C
Order by
nom p



Ix (nom)

Valeur d'index =
"CONCAT(col1, col2, col3, ...)"

- Select * From Produits where nom='A' order by Prix.
- Select * From Produits order by Nom

Table Villes — Recherche par Ville

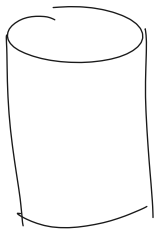
36 000 → Ix ? → OUI

Table Population — 65 000 000 — Ix

Colonne (cat)
- enfant
- ado
- adulte

Select nom_ FROM ... where cat = 'ado'

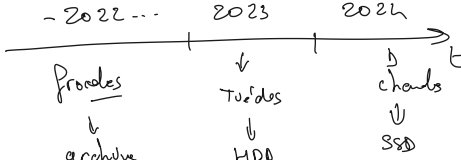
Partitionnement

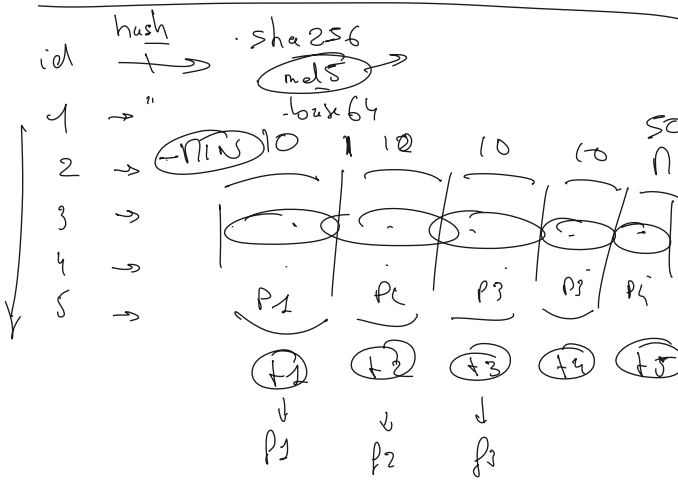


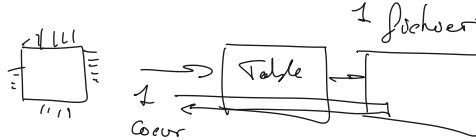
Ventiler → sur quel critère ?

- Pays ?
- année → catégorie
- on sait pas (hash)

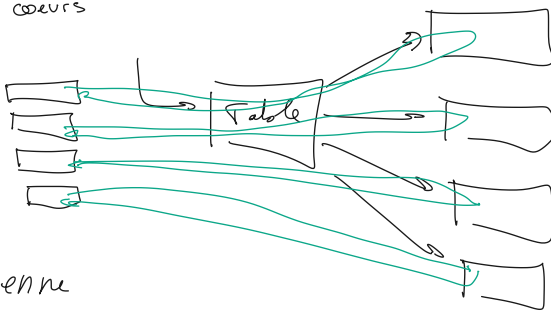
Volume → Trop conséquent







8 coeurs



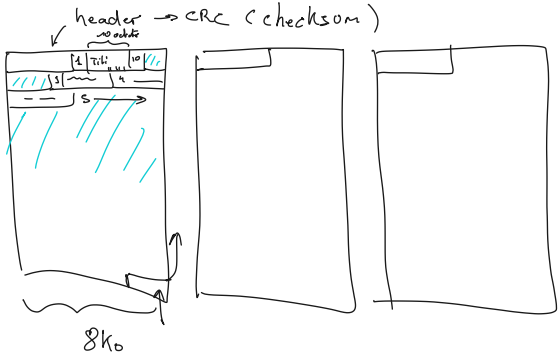
Moyenne

50 transactions

16 coeurs

CHAR (255)
VARCHAR (255)

'BONJOUR'



INSERT INTO Employes (1, 'titi', 10)

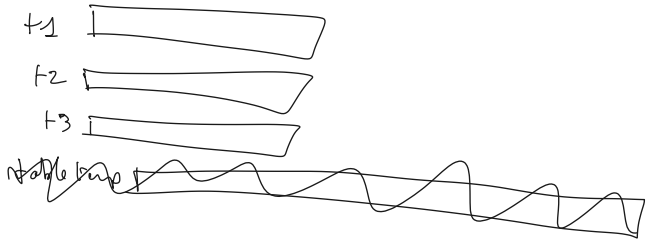
↓

CHAR(10)

CHAR → consomme + de place en général

Tables f1
f2
f3

imadb

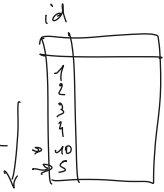


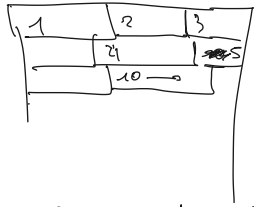
InnoDB (& MSSQL)

La PK est dite "clustered"

⇒ Preuve est physique

→ la PK est stockée dans la ligne





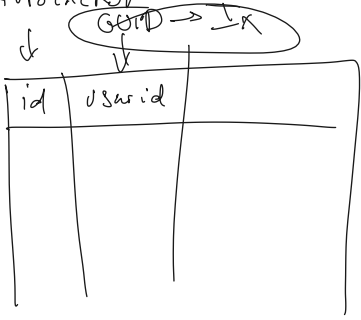
Auto increment

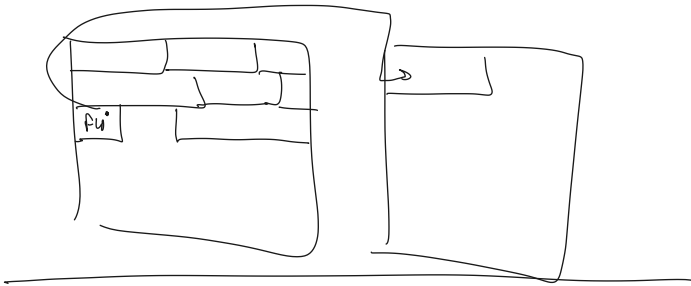
Souvent → PK est auto incrément

http://———/admin?user=

GUID UUID → PK InnoDB

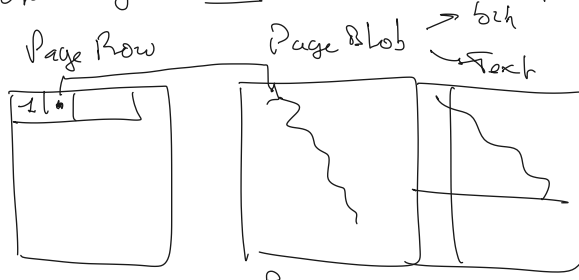
Auto incr

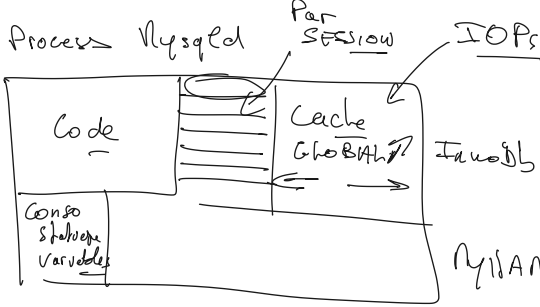


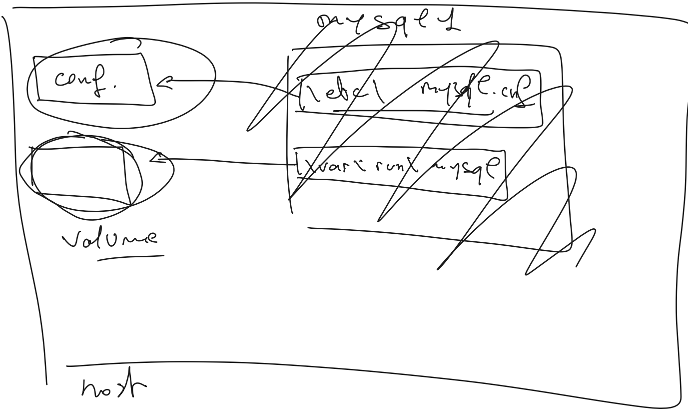


TEXT, BLOB

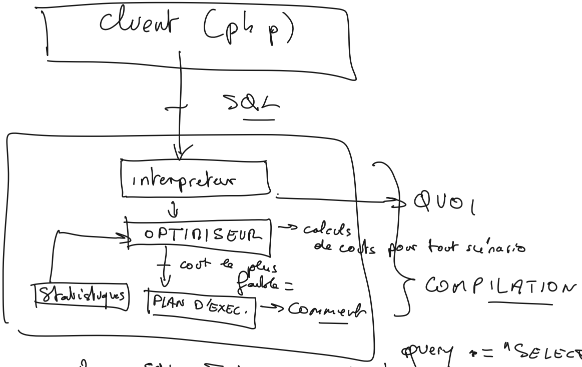
Tout type dont dans 1 page







OPTIMISEUR



Query Cache:

Req SQL Texte → hash

hash → PLAN

h → "

h → "

query = "SELECT * FROM Table where id = ?" ;

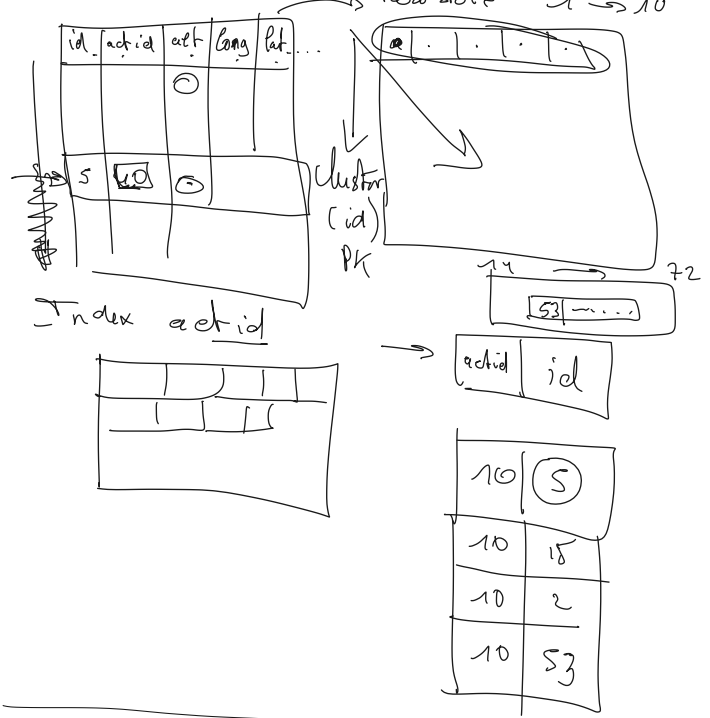


Table Persons

Nom	Prenom

60%

0,1%

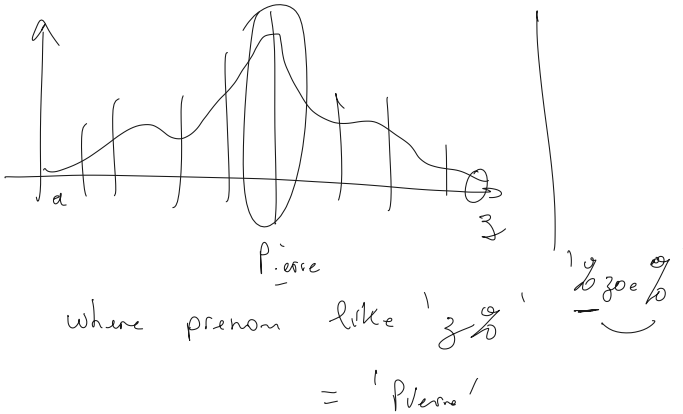


Table 7.1 Visual Explain Diagram Information

System Name	Color	Text on Visual Diagram	Tooltip Related Information
SYSTEM	Blue	Single row: system constant	Very low cost
CONST	Blue	Single row: constant	Very low cost
EQ_REF	Green	Unique Key Lookup	Low cost -- The optimizer is able to find an index that it can use to retrieve the required records. It is fast because the index search directly leads to the page with all the row data
REF	Green	Non-Unique Key Lookup	Low-medium -- Low if the number of matching rows is small; higher as the number of rows increases
FULLTEXT	Yellow	Fulltext Index Search	Specialized FULLTEXT search. Low -- for this specialized search requirement
REF_OR_NULL	Green	Key Lookup + Fetch NULL Values	Low-medium -- if the number of matching rows is small; higher as the number of rows increases
INDEX_MERGE	Green	Index Merge	Medium -- look for a better index selection in the query to improve performance
UNIQUE_SUBQUERY	Orange	Unique Key Lookup into table of subquery	Low -- Used for efficient Subquery processing
INDEX_SUBQUERY	Orange	Non-Unique Key Lookup into table of subquery	Low -- Used for efficient Subquery processing
RANGE	Orange	Index Range Scan	Medium -- partial index scan
INDEX	Red	Full Index Scan	High -- especially for large indexes
ALL	Red	Full Table Scan	Very High -- very costly for large tables, but less of an impact for small ones. No usable indexes were found for the table, which forces the optimizer to search every row. This could also mean that the search range is so broad that the index would be useless.
UNKNOWN	Black	unknown	Note: This is the default, in case a match cannot be determined

```
mysql> explain select a.transportation_mode from Activity a inner join User u on a.user_id=u.id where u.id='010';
+----+-------------+-------+------------+-------+---------------+---------+---------+------+------+----------+-------------+
| id | select_type | table | partitions | type  | possible_keys | key     | key_len | ref  | rows | filtered | Extra       |
+----+-------------+-------+------------+-------+---------------+---------+---------+------+------+----------+-------------+
|  1 | SIMPLE      | u     | NULL       | const | PRIMARY       | PRIMARY | 1022    | const |  1 |   100.00 | Using index |
|  1 | SIMPLE      | a     | NULL       | ref   | user_id       | user_id | 1022    | const |  488 |   100.00 | NULL        |
+----+-------------+-------+------------+-------+---------------+---------+---------+------+------+----------+-------------+
2 rows in set, 1 warning (0.00 sec)

mysql> explain select /*+ NO_JOIN_INDEX(u) */ a.transportation_mode from Activity a inner join User u on a.user_id=u.id where u.id='010';
+----+-------------+-------+------------+-------+---------------+---------+---------+------+------+----------+-------------+
| id | select_type | table | partitions | type  | possible_keys | key     | key_len | ref  | rows | filtered | Extra       |
+----+-------------+-------+------------+-------+---------------+---------+---------+------+------+----------+-------------+
|  1 | SIMPLE      | u     | NULL       | index | NULL          | PRIMARY | 1022    | NULL |  182 |     0.55 | Using where; Using index |
|  1 | SIMPLE      | a     | NULL       | ref   | user_id       | user_id | 1022    | const |  488 |   100.00 | NULL        |
+----+-------------+-------+------------+-------+---------------+---------+---------+------+------+----------+-------------+
2 rows in set, 1 warning (0.00 sec)

mysql> explain select /*+ NO_INDEX(u) */ a.transportation_mode from Activity a inner join User u on a.user_id=u.id where u.id='010';
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-------------+
| id | select_type | table | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra       |
+----+-------------+-------+------------+------+---------------+------+---------+------+------+----------+-------------+
|  1 | SIMPLE      | u     | NULL       | ALL  | NULL          | NULL | NULL    | NULL |  182 |     0.55 | Using where |
|  1 | SIMPLE      | a     | NULL       | ref  | user_id       | user_id | 1022 | const |  488 |   100.00 | NULL        |
+----+-------------+-------+------------+------+---------------+---------+---------+------+------+----------+-------------+
2 rows in set, 1 warning (0.00 sec)
```

SI

BI - DataWarehouse

Transact OLTP

ETL

Schéma "OLAP"

Sales

Dimensions

CRM ERP etc....

OnLine Transact Processing

Reporting

BOPR

→ Requêtes Rex

OnLine Analytical Processing

PK, FK

→ SELECT

id	Nom	Prix	Qt

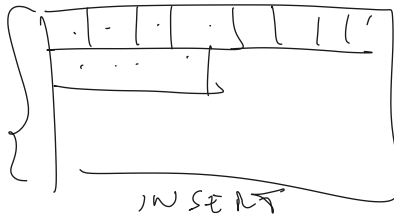
Test

"Mesures"

Select année, SUM(Prix) from ____ group by année

Structure id, nom, prix, qté

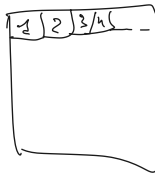
Row Store



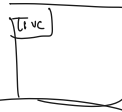
1 fichier = n Page

Colonne store

colonne, fichier id



nom



Select

SUM (Prix x Qte)

FROM —