

lien http iqdesk

<http://iqlab1732700872001.francecentral.cloudapp.azure.com/guacamole>

student/Pa\$\$w0rd

Connexion ssh native

iqlab1509792544002.francecentr

al.cloudapp.azure.com

student / Pa\$\$w0rd2022

ports :

st1:

49754 (iqsrv000)

62905(iqsrv001)

52582

st2 :

58850 - iqsrv003

51692

54311

st3:

53170 iqsrv006

60428

51253

st4:

50259 iqsrv009

57791

63884

st5:

60918 iqsrv012

55299

50297

st6:

58090 iqsrv015

53955

61450

st7:

54171 iqsrv018

60999

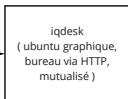
54161

st8:

50783 iqsrv021

60842

53581



login local : **st<n>/password**

connexion locale : iqsrv000, iqsrv001, iqsrv002, ...

Les VMs sont allumées le temps de la formation.

Nom stagiaire Prénom stagiaire Email stagiaire

AINOUZ Ahmed a.ainouz@cgi.com **st1**

DIOP Ahmadou Moustapha ahmadou.diop@cgi.com **st2**

KARKAZI Hamza hamza.karkazi@cgi.com **st3**

BEZZIZ Soufiane soufiane.bezziz@cgi.com **st4**

ERRAJI El Hassan elhassan.erraji@cgi.com **st5**

KARKACH Mohamed mohamed.karkach@cgi.com **st6**

LAZRAK Mohammed mohammed.lazrak@cgi.com **st7**

MOULTALA Fouzia fouzia.moultala@cgi.com **st8**

Fenetre Guacamole : Ctrl + Alt + Shift

RDP : Activities / Terminal

En RDP Direct :

iqlab1732700872001.francecentral.cloudapp.azure.com/guacamole

student/password

Préparation de la machine :
Copier vers le serveur (iqdesk /install/iq) :
rcp Linux--- student@iqsrv00n:~

ssh student@iqsrv00n
extraire l'installable :
sudo mkdir /install
chmod 777 /install
mv Linux--- /install
cd /install
tar xf Linux---

Préparation du dossier de destination :
sudo mkdir /opt/iq
chown student:student /opt/iq

Lancer l'install...

Prérequis :

`sudo yum install ksh csh`

A l'utilisation des outils :

source IQ.sh

- dbisql : outil graphique de requêtage
- dbisqlc : outil console de requêtage.

Préparation d'un dossier de BDD :

`sudo mkdir -p /var/iq/demo`

`sudo chown -R student:student /var/iq`

Vous devez être dans /var/iq/demo

lancez :

`/opt/iq/IQ-16_1/demo/mkiqdemo.sh -dba philippe -pwd password -absolute`

Quels serveurs tournent et leurs paramètres ?

stop_iq

##	Owner	PID	Started	CPU Time	Additional Information
----	-------	-----	---------	----------	------------------------

1:	student	32651	14:45 00:00:00	SVR:iqsr024_iqdemo	DB:iqdemo PORT:2638
----	---------	-------	----------------	--------------------	---------------------

/opt/iq/IQ-16_1/bin64/iqsr16 @iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o /opt/iq/IQ-16_1/logfiles/iqsr024_iqdemo.0001.srvlog -hn 5

Lancement du serveur :

`start_iq @iqdemo.cfg iqdemo.db`

`dbisqlc -c "uid=philippe;pwd=password;dbn=iqdemo"`

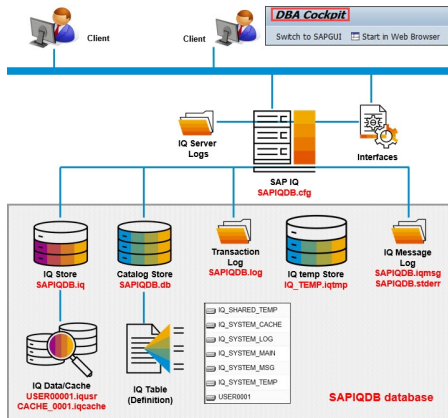
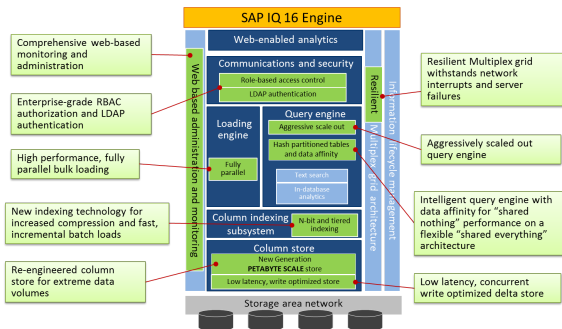
Vous pouvez aussi vous inspirer de cette commande :

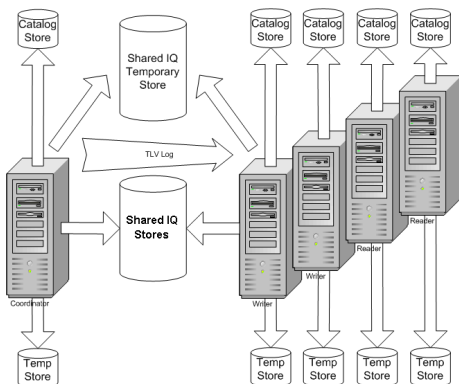
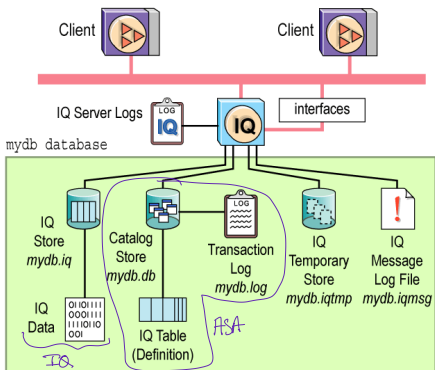
`uid=DBA;pwd=sql123;eng=bld_demo_9476;dbn=utility_db;links=tcPIP{host=iqsr024;port=9476}`

Pour se connecter à distance : (depuis iqdesk) - eng=champ SRV: de stop_iq

`dbisqlc -c`

`"uid=philippe;pwd=password;dbn=iqdemo;eng=iqsr024_iqdemo;links=tcPIP{host=iqsr024;port=2638}"`





stop iq affiche :
SVR:iqsr024_iqdemo DB:iqdemo PORT:2638 ← Port TCP

nom du ENGINE

nom de la base

nom de la BDD (DB:)

deviens :
dbisqlc -c
"uid=philippe;pwd=password;dbn=iqdemo;eng=iqsr024_iqdemo;links=tcip{host=iqsr024;port=2638}"

voire login/pass (non récupérable)

hostname

Création d'une base de données "From Scratch" :

Fichier de params :

[student@iqsr024 iqdb]\$ cat voidparams.cfg

```
-n iqsr  
-su DBA,sql123  
-c 48m  
-gd all  
-gm 10  
-gp 4096  
-x tcip{port=9813}
```

-iqmc 64

-iqtc 64

start iq @voidparams.cfg

dbisqlc -c

"uid=DBA;pwd=sql123;eng=iqsr;dbn=utility_db;links=tcip{host=iqsr024;port=9813}"

Création de la base de données (<https://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38159.1600/doc/html/rhi1334847427020.html>)

Serveur fantôme :

start iq -su dba,sql123 -n my_server

dbisqlc -c

"uid=DBA;pwd=sql123;eng=my_server;dbn=utility_db;links=tcip{host=iqsr024;port=2638}"

```
CREATE DATABASE 'iqdb.db'
  BLANK PADDING ON
  DBA USER 'philippe'
  DBA PASSWORD 'password'
  // Minimum_Password_Length
  CASE IGNORE
  IQ PATH 'iqdb.iq' // RELATIVE SYS_SPACE
  IQ SIZE 100 // For 15.0 release demo has 25/75 partition
  IQ RESERVE 200
  IQ PAGE SIZE 131072 // Block Size for 12.5 and up is 128k
  // JAVA ON
  // JCONNECT ON
  TEMPORARY PATH 'iqdb.iqtmp'
  TEMPORARY SIZE 25
  TEMPORARY RESERVE 200;

// CONNECT_ASIQ_STRING
;
```

Ensuite, lancer un serveur en prod :

stop_iq

start_iq @iqdb.cfg iqdb.db

avec iqdb.cfg :

Je ne vais pas forcer le port ni le login/mot de passe

-n iqsrv

#-su DBA,sql123

-c 48m

-gd all

-gm 10

-gp 4096

#-x tcpip{port=9813}

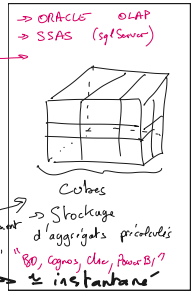
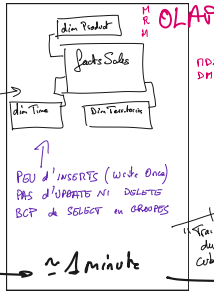
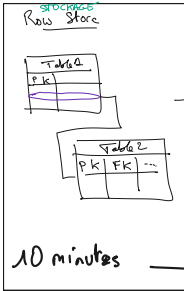
-iqmc 64

-iqtc 64

OLTP

Stockage Row Store OK (déconseillé)
Colonne Store Recommandé

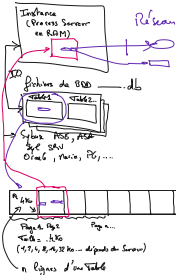
MULTIDIMENSIONNEL



Stockage Relationnel / Référentiel

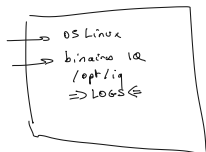
Data Warehouse /
Table de faits et dimensions
lignes jointures (pas recommandé)

REPORTING

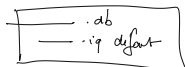


MOLAP → Multidimensional OLAP → Columnstore / NDX (déphase)
 ROLAP → Relationnel → Row Store (à jour)
 HOLAP → Hybrid → Combinaison des deux

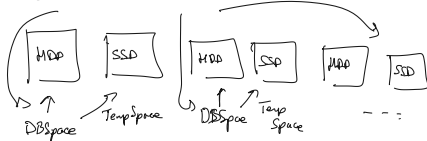
disks principal OS



DB Space



CREATE TABLE



En SAN → vous pouvez créer $n \times 2$
 DBSpace selon n Coeurs

Création de contenu :

[https://infocenter.sybase.com/help/index.jsp?](https://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38159.1600/doc/html/rhi1334847427020.html)

[topic=/com.sybase.infocenter.dc38159.1600/doc/html/rhi1334847427020.html](https://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38159.1600/doc/html/rhi1334847427020.html)

```
CREATE DBSPACE mydbspace2 USING FILE mydbfile2 '/var/iq/mydbfile2.iq' SIZE 10 RESERVE 20MB
```

```
create table test ( id int, nom varchar(50) ) IN mydbspace
```

```
insert into test values ( 3, 'fdsfsfjhdslkj')
```

Exemple renommer un fichier de dbspace :

```
alter dbspace mydbspace readonly
```

```
alter dbspace mydbspace offline
```

```
alter dbspace mydbspace rename 'mydbfile2.iq'
```

```
-- renommer le fichier sous bash ( mv )
```

```
alter dbspace mydbspace online
```

```
alter dbspace mydbspace readwrite
```

Définition d'une connexion à IQ Depuis Talend / JDBC :(jconn4.jar est disponible sous /opt/iq/IQ-16_1/java

Database Connection

New Database Connection on repository - Step 2/2

Define the connection parameters

DB Type: JDBC

JDBC URL: jdbc:sybase:Tds:iqsrv024:2638

Drivers:

- driver
- jconn4.jar
- jodbc4.jar

Driver Class: com.sybase.jdbc4.jdbc.SybDriver

User Id: philippe

Password: *****

Mapping file: Mapping Mysql

Test connection

Export as context

Revert Context

< Back Next > Cancel Finish

Database: Sybase (ASE and IQ) Apply

Property Type: Built-In

Use an existing connection

DB Version: Sybase 16

Server: iqsrv024 Port: 2638 Database: iqdb Sybase Schema: philippe

Username: philippe Password: *****

Schema: Built-In Edit schema

Table Name: test

Query Type: Built-In Guess Query Guess schema

Query: SELECT FROM test

Gestion de la sécurité :

1) Création du groupe :

```
CREATE USER sales_group
```

```
GRANT GROUP TO sales_group
```

2) création de l'utilisateur :

```
CREATE USER my_user IDENTIFIED BY 'pa55word'
```

```
GRANT CONNECT TO sophie
```

3) Création du groupe :

```
GRANT CONNECT TO test_group IDENTIFIED BY 'password'
```

```
GRANT GROUP TO test_group
```

4) distribuer les droits au groupe :

```
GRANT ALL ON philippe.test TO test_group
```

5) affecter le user au groupe :

```
GRANT MEMBERSHIP IN GROUP test_group TO sophie
```

6) Modification des droits :

```
revoke ALL on philippe.test FROM test_group
```

```
GRANT SELECT ON philippe.test TO test_group
```

7) Faire passer un USER en tant que GROUPE et faire de l'affectation :

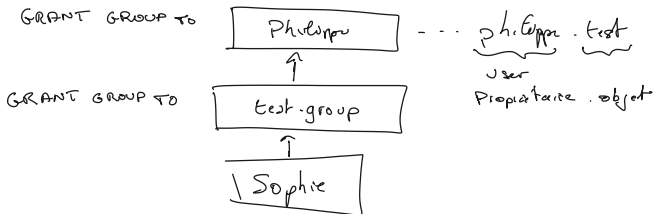
```
grant group to philippe
```

```
GRANT MEMBERSHIP IN GROUP philippe TO test_group
```

Le fait qu'un user fasse partie d'un autre user ou groupe, permet de ne pas avoir à préciser dans la recherche, l'autre user/groupe en tant que préfixe :

```
select * from test
```

```
select * from philippe.test ( si le user ne fait pas partie du groupe )
```



Chargement de fichier Bulk Insert

CREATE TABLE Contacts

```
(
  ID          integer      NOT NULL ,
  Surname     varchar(50)   NOT NULL ,
  GivenName   varchar(50)   NOT NULL ,
  Title       varchar(50)   NULL   ,
  Street      varchar(50)   NULL   ,
  City        varchar(50)   NULL   ,
  State       varchar(50)   NULL   ,
  Country     varchar(50)   NULL   ,
  PostalCode  varchar(50)   NULL   ,
  Phone       varchar(50)   NULL   ,
  Fax        varchar(50)   NULL   ,
  CustomerID  integer      NULL
);
```

drop table contacts;

```
load table Contacts (ID, Surname, GivenName, Title, Street, City, State, Country,
PostalCode, Phone, Fax, CustomerID) from
'/opt/iq/IQ-16_1/demo/demodata/contact.dat'
ROW DELIMITED BY '\n'
DELIMITED BY ','
QUOTES OFF
ESCAPES OFF;
```

select count(*) from Contacts;

insert into contacts select * from contacts;

Export de données :

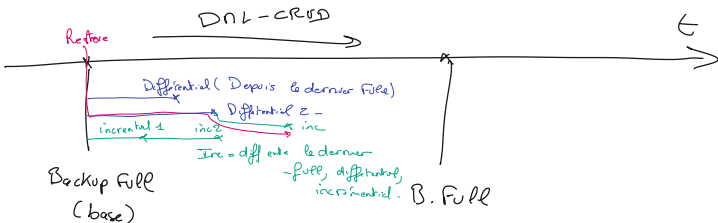
```
SELECT *
FROM sysobjects
WHERE name IN ('SYSCOLUMNS','SYSOPTIONS');
OUTPUT TO c:\objects.txt;
```

Imports / Exports :

<https://infocenter.sybase.com/help/index.jsp?>

[topic=/com.sybase.infocenter.dc00170.1510/html/iqapgv1/X200132.htm](https://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc00170.1510/html/iqapgv1/X200132.htm)

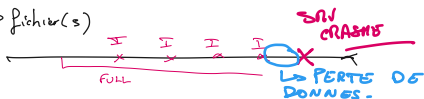
Backup / Restore



Backup Full
(base)

→ GROS

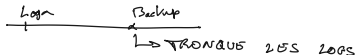
→ fichier(s)



Eviter la perte de données:

- Data

- Logs (Transactions) → en ajout constant



BACKUP LOGS -
ou gérer le fichier

Recommande de séparer Log et Data

si disques locaux.

(possible de forcer le startup sans
fichier Log)

Faire un backup :

BACKUP DATABASE FULL TO 'var/iq/iqdb.bak'

FULL

or

INCREMENTAL

Or

INCREMENTAL SINCE FULL

Effectuer une Restauration :

1) préparer le stockage

```
mkdir iqdb.recovered
```

```
cd iqdb.recovered
```

2) Lancer une bdd utility_db (server fantôme):

```
start_iq -su dba,sql123 -n my_server ( accessible uniquement aux dbas )
```

3) Lancer la restauration :

```
dbisqlc -c "uid=dba;pwd=sql123;eng=my_server;dbn=utility_db;links=tcipip{host=iqsrv024;port=2638}"
```

```
restore database 'iqdb2' from 'iqdb.bak'
```

-> cela n'a pas récupéré le fichier cfg qu'il faudra avoir copié préalablement.

4) arrêt du serveur temporaire

```
stop_iq
```

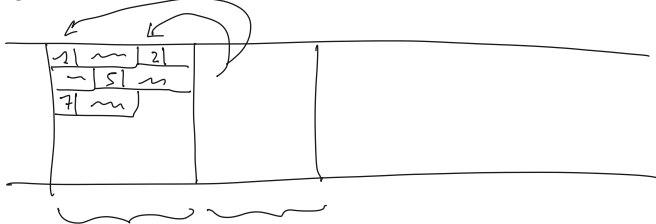
5) relancer le serveur monté

```
start_iq @iqdb.cfg iqdb2.db
```

```
dbisqlc -c "uid=philippe;pwd=password;eng=iqsrv;dbn=iqdb2;links=tcipip{host=iqsrv024;port=2638}"
```

Optimisation

Index cluster



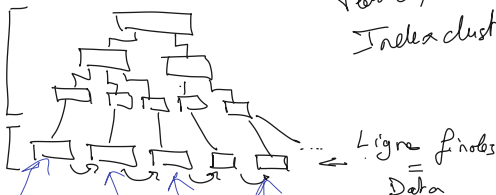
Page 7 (data)

Index

Table,
Index cluster

Index
(branches)

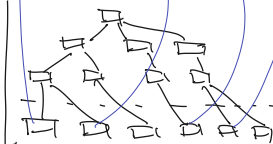
feuilles



Index Secondaire
(Pas une PK)

Index

feuilles



Pointeurs vers la
PK si PK il ya
N° de Page si pas de PK

Cardinalité :

Nb de Valeurs distinctes

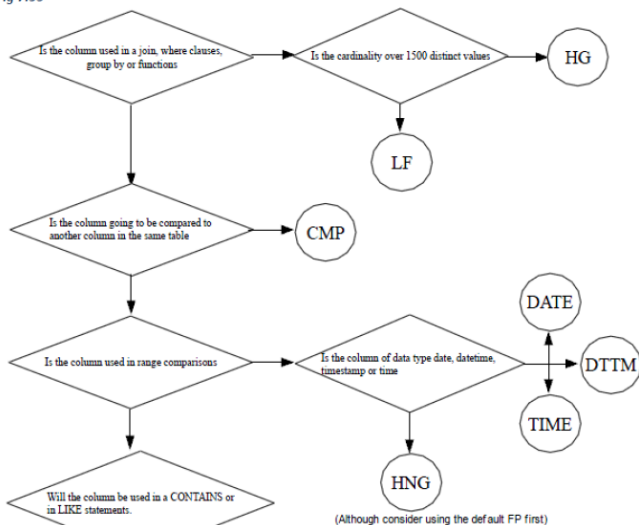
Nb de Valeurs totales

$0 < \text{Cardinalite} \leq 1$

↳ bcp de valeurs distinctes (uniques)

Indication peu utile

Fig 7.35



```
alter dbspace IQ_SYSTEM_TEMP ADD 50 MB
```

```
alter dbspace IQ_SYSTEM_MAIN ADD FILE mainf1 'mainf1.iq' SIZE 1 GB
```

```
sp_iqdbspace
```

```
sp_iqdbspaceinfo
```

```
sp_iqdbsize
```

```
SET OPTION MINIMIZE_STORAGE='on'
```

```
SET OPTION Index_Advisor=ON
```

```
SET OPTION Index_Advisor_Max_Rows=1000
```

```
sp_iqindexadvice
```


Exercice concernant les Index :

Créer une base de données démo / ou charger des tables de la base de démos (avec des relations)
Dupliquez les données d'une table "centrale" (produit - catégories) : dupliquez les produits (attention à la PK qui ne devra plus être unique)

insert into matable select * from matable (duplique le contenu) pour avoir des dizaines de millions d'enregs

consultez les index

Demandez les hints sur les index par rapport à des requêtes dont jointure.

Créez un index HG ou LF en fonction de la cardinalité de la colonne de jointure, sur la colonne de jointure

Exécutez une jointure avant et après création de cet index

Pour faire un bench sur une requête :

```
trainer@iqdesk:~$ time dbisql -c
```

```
"uid=philippe;pwd=password;eng=iqsrv;dbn=iqdb;links=tcpiip{host=iqsrv024;port=2638}" "select * from contacts" >/dev/null
```

real 0m2.107s

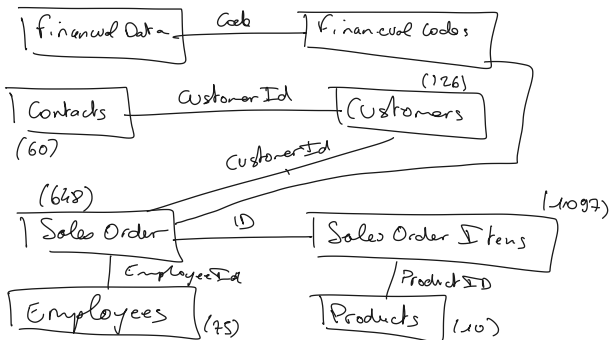
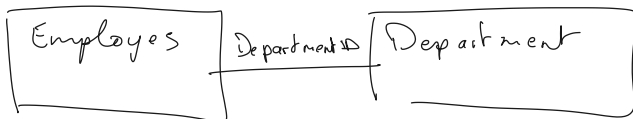
user 0m1.885s

sys 0m0.493s

Dupliquer une table sans PK :

```
create table testtest as (select * from contacts)
```

```
describe testtest
```



```
Multiplex : create iqn.2021-05.iqdesk:iq.trainer
create iqn.2021-05.iqdesk:iqsrv024.initiator01
/iscsi/iqn.20...4.initiator01> set auth userid=username
Parameter userid is now 'username'.
/iscsi/iqn.20...4.initiator01> set auth password=password
Parameter password is now 'password'.
```

https://www.server-world.info/en/note?os=CentOS_7&p=iscsi&f=2

Cargement des tables volumineuses de dbt3 :

Il y a les tables :

- rw-rw-r-- 1 student student 24M Oct 12 2016 customer.tbl
- rw-rw-r-- 1 student student 725M Oct 12 2016 lineitem.tbl
- rw-rw-r-- 1 student student 2.2K Oct 12 2016 nation.tbl
- rw-rw-r-- 1 student student 164M Oct 12 2016 orders.tbl
- rw-rw-r-- 1 student student 24M Oct 12 2016 part.tbl
- rw-rw-r-- 1 student student 114M Oct 12 2016 partsupp.tbl
- rw-rw-r-- 1 student student 389 Oct 12 2016 region.tbl
- rw-rw-r-- 1 student student 1.4M Oct 12 2016 supplier.tbl

récupération depuis un serveur :

scp trainer@iqdesk:/mnt/sdd/region.tbl .

DBT3 :

Création de la table et insertion :

```
CREATE TABLE REGION ( R_REGIONKEY INTEGER NOT NULL,  
                        R_NAME      CHAR(25) NOT NULL,  
                        R_COMMENT   VARCHAR(152));
```

```
load table region ( R_REGIONKEY , R_NAME,R_COMMENT) FROM  
'/home/student/region.tbl' delimited by '|' ESCAPES OFF;  
select * from region;
```



```

ALTER TABLE REGION
ADD PRIMARY KEY (R_REGIONKEY);

-- For table NATION
ALTER TABLE NATION
ADD PRIMARY KEY (N_NATIONKEY);

ALTER TABLE NATION
ADD FOREIGN KEY NATION_FK1 (N_REGIONKEY) references REGION(R_REGIONKEY);

-- For table PART
ALTER TABLE PART
ADD PRIMARY KEY (P_PARTKEY);

-- For table SUPPLIER
ALTER TABLE SUPPLIER
ADD PRIMARY KEY (S_SUPPKEY);

ALTER TABLE SUPPLIER
ADD FOREIGN KEY SUPPLIER_FK1 (S_NATIONKEY) references NATION(N_NATIONKEY);

-- For table PARTSUPP
ALTER TABLE PARTSUPP
ADD PRIMARY KEY (PS_PARTKEY,PS_SUPPKEY);

-- For table CUSTOMER
ALTER TABLE CUSTOMER
ADD PRIMARY KEY (C_CUSTKEY);

ALTER TABLE CUSTOMER
ADD FOREIGN KEY CUSTOMER_FK1 (C_NATIONKEY) references NATION(N_NATIONKEY);

-- For table LINEITEM
ALTER TABLE LINEITEM
ADD PRIMARY KEY (L_ORDERKEY,L_LINENUMBER);

-- For table ORDERS
ALTER TABLE ORDERS
ADD PRIMARY KEY (O_ORDERKEY);

-- For table PARTSUPP
ALTER TABLE PARTSUPP
ADD FOREIGN KEY PARTSUPP_FK1 (PS_SUPPKEY) references SUPPLIER(S_SUPPKEY);

ALTER TABLE PARTSUPP
ADD FOREIGN KEY PARTSUPP_FK2 (PS_PARTKEY) references PART(P_PARTKEY);

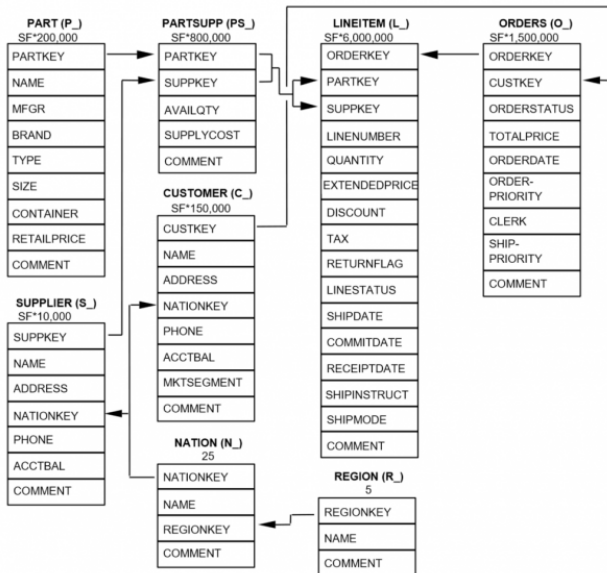
-- For table ORDERS
ALTER TABLE ORDERS
ADD FOREIGN KEY ORDERS_FK1 (O_CUSTKEY) references CUSTOMER(C_CUSTKEY);

-- For table LINEITEM
ALTER TABLE LINEITEM
ADD FOREIGN KEY LINEITEM_FK1 (L_ORDERKEY) references ORDERS(O_ORDERKEY);

ALTER TABLE LINEITEM
ADD FOREIGN KEY LINEITEM_FK2 (L_PARTKEY,L_SUPPKEY) references
    PARTSUPP(PS_PARTKEY, PS_SUPPKEY);

```

The TPC-H Schema



Testez le plan d'exécution :

SET TEMPORARY OPTION QUERY_PLAN_TEXT_ACCESS = ON;

SELECT HTML_PLAN ('SELECT count(*) FROM salesorderitems');

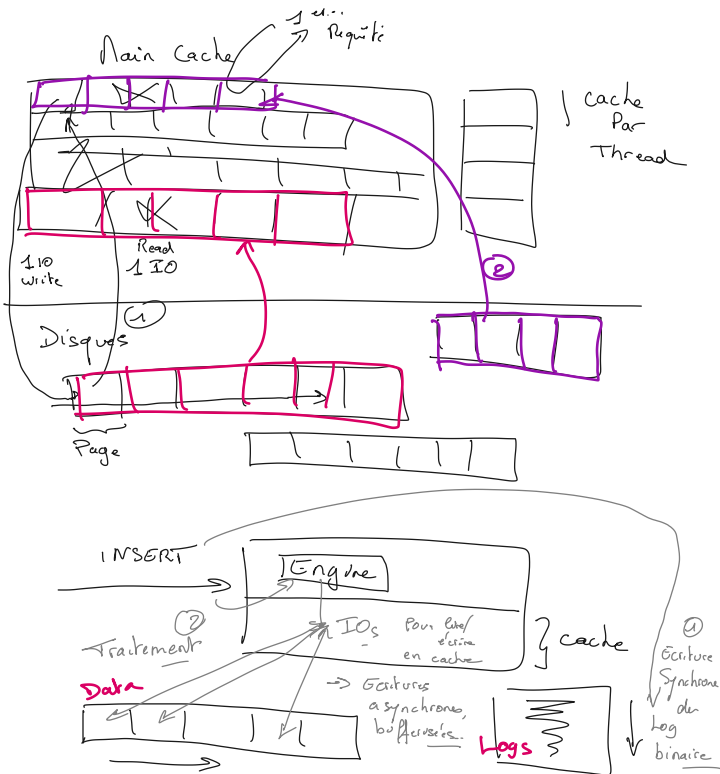
OUTPUT TO 'home/trainer/hplan.html' HEXADECIMAL ASIS QUOTE '';

firefox ~/hplan.html

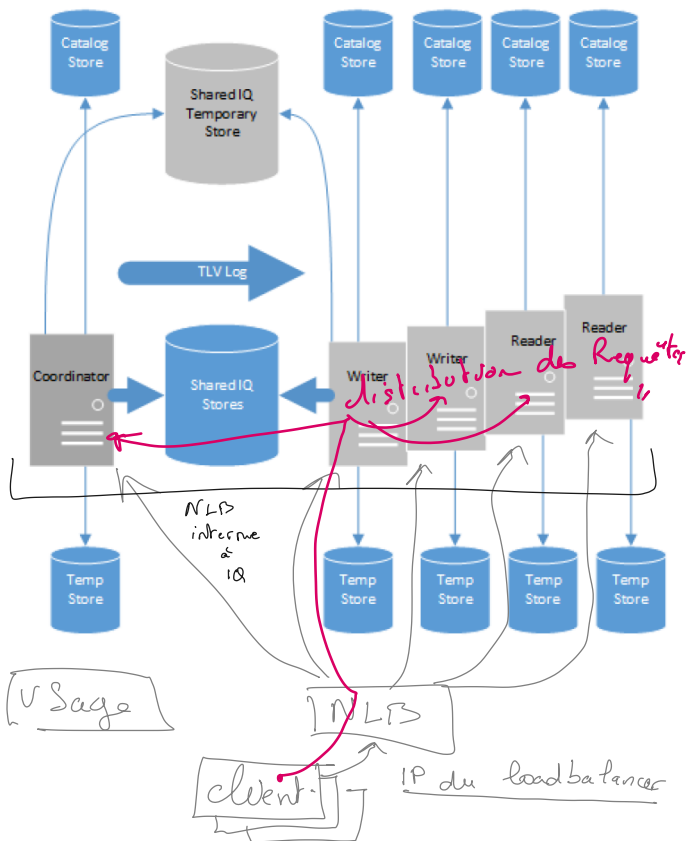
QUERY_PLAN_AFTER_RUN OFF (par défaut) : plan d'exécution estimé

QUERY_PLAN_AFTER_RUN ON : plan d'exécution réel : sauvegarde un .html côté serveur

QUERY_PLAN_AS_HTML_DIRECTORY : pour envoyer les html sur un autre répertoire



Nutplex



Par client = 1 cnx sur 1 des membres

Implémentation de Multiplex (mpx)

1 Serveur à la base = Simplex (spx)

① Création d'une BDD Simplex

→ attention aux chemins relatifs/absolus

(CREATE DATABASE

IQ STORE ... ← chemins.

CATALOG STORE - - -)

② Migration en mpx via la déclaration d'un nœud supplémentaire.

③ Reboot de ce coordinateur

④ Backup et restore de la BDD catalog du coordinateur sur le 2nd nœud

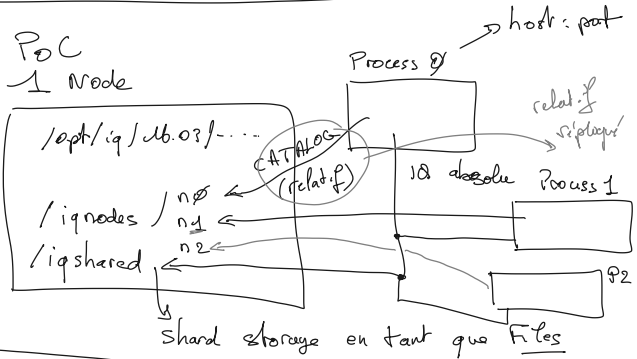
⑤ Démarrage du 2nd nœud
qui passe UP et accède aux IQ Shared, le coord. le verra en ligne et commencera la réplique des changements sur celui-ci.

⑥ (conf. d'un NLB vers ces nœuds)

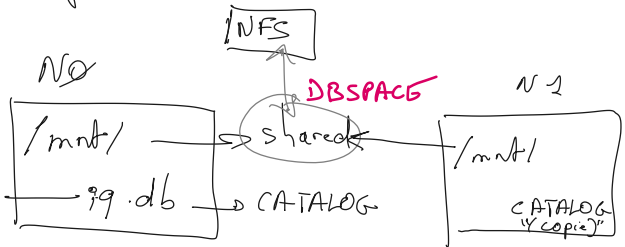
⑦ on répète ④ et ⑤ pour chaque nœud suppl.

3 Architectures possibles

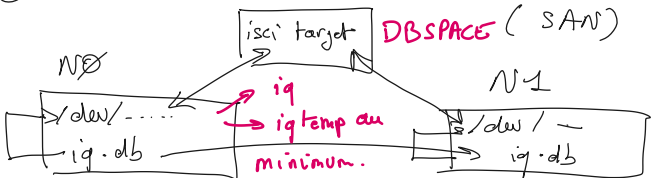
① PoC 1 Node



② No Binode via NFS (possible, pas forcément recommandé) - NAS



③ Multinode via iscsi raw devices



POC :

```
start_iq -su dba,sql123 -n n0
```

```
CREATE DATABASE './iq.db'  
IQ PATH '/var/iq/shared/iq.iq'  
IQ SIZE 100          // For 15.0 release demo has 25/75 partition  
IQ RESERVE 200  
IQ PAGE SIZE 131072 // Block Size for 12.5 and up is 128kMESSAGE PATH './iq.iqmsg'  
TEMPORARY PATH '/var/iq/shared/iq.iqtemp'  
TEMPORARY SIZE 25  
TEMPORARY RESERVE 200;DBA USER 'philippe'  
DBA PASSWORD 'password'  
CASE IGNORE  
BLANK PADDING ON
```

```
stop_iq
```

```
cat n0.cfg  
-n n0  
-c 48m  
-gd all  
-gm 10  
-gp 4096  
-x tcpip{port=3000}  
-iqmc 64  
-iqtc 64
```

```
start_iq @n0.cfg iq.db
```

(connexion sur le nouveau simplex :)

```
dbisql -c "uid=philippe;pwd=password;eng=n0;dbn=iq;links=tcpip{host=iqsr024;port=3000}"
```

Basculement en multiplex en ajoutant un node dans le cluster (2 opérations en 1) :

```
-- CREATE MULTIPLEX SERVER n1 DATABASE './iq.db' HOST 'localhost' PORT 3001 ROLE WRITER STATUS INCLUDED  
-- database path must be absolute
```

```
CREATE MULTIPLEX SERVER n1 DATABASE '/var/iq/n1/iq.db' HOST 'localhost' PORT 3001 ROLE WRITER STATUS INCLUDED  
-- ok
```

le redémarre le coordinateur :

```
start_iq @n0.cfg iq.db
```

<https://www.server-world.info/en>

Préparation du n1 :

```
cat n1.cfg  
-n n1  
-c 48m  
-gd all  
-gm 10  
-gp 4096  
-x tcpip{port=3001}  
-iqmc 64  
-iqtc 64
```

dbbackup -y -d -c "uid=philippe;pwd=password;links=tcpip{port=3000};eng=n0" .

Supprimer et régénérer le bin log de cette restauration :

```
rm -rf mpctest.log
```

9. Run dblog to reset the transaction log:

```
dblog -r -t iq.log iq.db
```

Augmenter les DBSpaces : (a exécuter SUR LE COORDINATEUR)

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD FILE mainfile 'var/iq/shared/mainfile.iq'
```

de qui implique de refaire un backup/restore du coordinateur vers les workers

Création en SAN :

```
CREATE DATABASE 'var/iq/n0/iq.db'  
IQ PATH '/dev/sdb1'  
IQ SIZE 100  
IQ RESERVE 200  
IQ PAGE SIZE 131072  
MESSAGE PATH 'var/iq/n0/iq.iqmsg'  
TEMPORARY PATH '/dev/sdb2'  
TEMPORARY SIZE 25  
TEMPORARY RESERVE 200  
DBA USER 'philippe'  
DBA PASSWORD 'password'  
CASE IGNORE  
BLANK PADDING ON
```

```
mkdir /var/lib/iscsi_disks
dd if=/dev/zero of=/var/lib/iscsi_disks/disk02.img count=0 bs=1 seek=5G
dd if=/dev/zero of=/var/lib/iscsi_disks/disk01.img count=0 bs=1 seek=10G
```

```
cat /etc/tgt/targets.conf
# This is a sample config file for tgt-admin.
#
# The "#" symbol disables the processing of a line.

# Set the driver. If not specified, defaults to "iscsi".
default-driver iscsi

# Set iSNS parameters, if needed
#iSNSServerIP 192.168.111.222
#iSNSServerPort 3205
#iSNSAccessControl On
#iSNS On

# Continue if tgtadm exits with non-zero code (equivalent of
# --ignore-errors command line option)
#ignore-errors yes
```

```
<target iqn.2015-12.mpx000:target00>
  # provided device as a iSCSI target

  backing-store /var/lib/iscsi_disks/disk01.img
  backing-store /var/lib/iscsi_disks/disk02.img
  # iSCSI Initiator's IP address you allow to connect

  initiator-address 10.0.0.32
  # authentication info ( set anyone you like for "username", "password" )

  incominguser username password
</target>
```

```
systemctl restart tgt
```