## Cheatography

## **Test Cheat Sheet**

by Abbay Kutte via cheatography.com/101747/cs/21182/

test1				test1 (cont)		Test 2
Stage	Stage	Stage	Stage	\\eudvmms-	\\eudvmms-	Stage source data in
source data	source data	source data	source data	qs501\dev-	qs501\dev-	QVD files and then load
in QVD files	in QVD files	in QVD files	in QVD files	\1000.Dat-	\1000.Dat-	from the QVD as this will
and then	and then	and then	and then	a QVD -	a QVD -	avoid strain on the source
load from	load from	load from	load from	Layer\1.Q-	Layer\1.Q-	systems and possibly
the QVD as	the QVD as	the QVD as	the QVD as	VD\1.Extr-	VD\1.Extr-	network bandwidth as
this will	this will	this will	this will	act\QV-	act\QV-	well as be a lot quicker,
avoid strain	avoid strain	avoid strain	avoid strain	_QVD_B-	_QVD_B-	safer and more
on the	on the	on the	on the	efore\	efore\	productive • Break out
source	source	source	source			different data source load
systems	systems	systems	systems			process into different
and	and	and	and			script sections and use a
possibly	possibly	possibly	possibly			Exit Script section, which
network	network	network	network			can be easily moved to
bandwidth	bandwidth	bandwidth	bandwidth			test each of your load
as well as	as well as	as well as	as well as			processes separately • If
be a lot	be a lot	be a lot	be a lot			possible, develop with a
quicker,	guicker,	quicker,	guicker,			meaningful subset of data
safer and	safer and	safer and	safer and			using Where clauses
more	more	more	more			and/or Exists clauses in
productive •	productive •	productive •	productive •			the load process to
Break out	Break out	Break out	Break out			ensure you maintain
different	different	different	different			relevant key matches •
data source	data source	data source	data source			Avoid trying to create
oad	load	load	load			overly large applications
orocess into	process into	process into	process into			covering multiple use
different	different	different	different			cases, it is far more
script	script	script	script			efficient to create several
sections	sections	sections	sections			smaller applications each
and use an	and use an	and use an	and use an			covering a discrete user
Exit Script	Exit Script	Exit Script	Exit Script			journey • Remove
section,	section,	section,	section,			synthetic keys and where
which can	which can	which can	which can			possible and circular
be easily	be easily	be easily	be easily			references • Remove (or
moved to	moved to	moved to	moved to			comment out a better
est each of	test each of	test each of	test each of			practice) all unused fields
our load	your load	your load	your load			from the load • Remove
orocesses	processes	processes	processes			or simplify time stamps
separately •	separately •	separately •	separately •			(for example you don't
f possible,	If possible,	If possible,	If possible,			need 1/100th of a second
develop	develop	develop	develop			so you could use the ceil
with a	with a	with a	with a			function to round up to th
meaningful	meaningful	meaningful	meaningful			nearest minute) or highly
subset of	subset of	subset of	subset of			unique system fields •
data using	data using	data using	data using			Use Limited Load in
Where	Where	Where	Where			debug mode to test your
clauses	clauses	clauses	clauses			
and/or	and/or	and/or	and/or			logic of the script before
			and/or Exists			running a full reload or use the First function to
Exists	Exists	Exists				
clauses in	clauses in	clauses in	clauses in			limit the load • Use
the load	the load	the load	the load			Autonumber to replace
process to	process to	process to	process to			text string based key

ensure you ensure you maintain maintain relevant key relevant key matches • matches • Avoid trying Avoid trying to create to create overly large overly large applications applications covering covering multiple use multiple use cases, it is cases, it is far more far more efficient to efficient to create create several several smaller smaller applications applications each each covering a covering a discrete discrete user journey user journey Remove Remove synthetic synthetic keys and keys and where where possible possible and circular and circular references • references • Remove (or Remove (or comment comment out a better out a better practice) all practice) all unused unused fields from fields from the load • the load • Remove or Remove or simplify simplify time stamps time stamps (for (for example example you don't you don't need need 1/100th of a 1/100th of a second so second so you could you could use the ceil use the ceil function to function to round up to round up to the nearest the nearest minute) or minute) or highly highly unique unique system system fields • Use fields • Use Limited Limited Load in Load in debug debug mode to test mode to test

ensure you maintain relevant key matches • Avoid trying to create overly large applications covering multiple use cases, it is far more efficient to create several smaller applications each covering a discrete user journey Remove synthetic keys and where possible and circular references • Remove (or comment out a better practice) all unused fields from the load • Remove or simplify time stamps (for example you don't need 1/100th of a second so you could use the ceil function to round up to the nearest minute) or highly unique system fields • Use Limited Load in debug mode to test

ensure you maintain relevant kev matches • Avoid trying to create overly large applications covering multiple use cases, it is far more efficient to create several smaller applications each covering a discrete user journey Remove synthetic keys and where possible and circular references • Remove (or comment out a better practice) all unused fields from the load • Remove or simplify time stamps (for example you don't need 1/100th of a second so you could use the ceil function to round up to the nearest minute) or highly unique system fields • Use Limited Load in debug mode to test

fields with more efficient integers • Remove, join or concatenate unnecessary snow flaked tables • Avoid using nested if statements – alternatives are mapping tables in the load script and pick (match functions and Set Analysis with flag fields in the User Interface • Consider the use of incremental loads for large data sets that need to be regularly updated, this will reduce the load on the source system and speed up the overall load process

your logic of	your logic of	your logic of	your logic of
the script	the script	the script	the script
before	before	before	before
running a	running a	running a	running a
full reload	full reload	full reload	full reload
or use the	or use the	or use the	or use the
First	First	First	First
function to	function to	function to	function to
limit the	limit the	limit the	limit the
load • Use	load • Use	load • Use	load • Use
Autonumber	Autonumber	Autonumber	Autonumber
to replace	to replace	to replace	to replace
text string	text string	text string	text string
based key	based key	based key	based key
fields with	fields with	fields with	fields with
more	more	more	more
efficient	efficient	efficient	efficient
integers •	integers •	integers •	integers •
Remove,	Remove,	Remove,	Remove,
join or	join or	join or	join or
concatenate	concatenate	concatenate	concatenate
unnece-	unnece-	unnece-	unnece-
ssary snow	ssary snow	ssary snow	ssary snow
flaked	flaked	flaked	flaked
tables •	tables •	tables •	tables •
Avoid using	Avoid using	Avoid using	Avoid using
nested if	nested if	nested if	nested if
statements	statements	statements	statements
– altern-	– altern-	– altern-	– altern-
atives are	atives are	atives are	atives are
mapping	mapping	mapping	mapping
tables in the	tables in the	tables in the	tables in the
load script	load script	load script	load script
and pick	and pick	and pick	and pick
(match	(match	(match	(match
functions	functions	functions	functions
and Set	and Set	and Set	and Set
Analysis	Analysis	Analysis	Analysis
with flag	with flag	with flag	with flag
fields in the	fields in the	fields in the	fields in the
User	User	User	User
Interface •	Interface •	Interface •	Interface •
Consider	Consider	Consider	Consider
the use of	the use of	the use of	the use of
incremental	incremental	incremental	incremental
loads for	loads for	loads for	loads for
large data	large data	large data	large data
sets that	sets that	sets that	sets that
need to be	need to be	need to be	need to be
regularly	regularly	regularly	regularly
updated,	updated,	updated,	updated,
this will	this will	this will	this will
reduce the	reduce the	reduce the	reduce the
load on the	load on the	load on the	load on the
source	source	source	source
system and	system and	system and	system and
speed up	speed up	speed up	speed up
the overall	the overall	the overall	the overall

load	load	load	load
process	process	process	process

## By Abbay Kutte

cheatography.com/abbaykutte/ Not published yet. Last updated 20th November, 2019. Page 1 of 100. Sponsored by **ApolloPad.com** Everyone has a novel in them. Finish Yours! https://apollopad.com