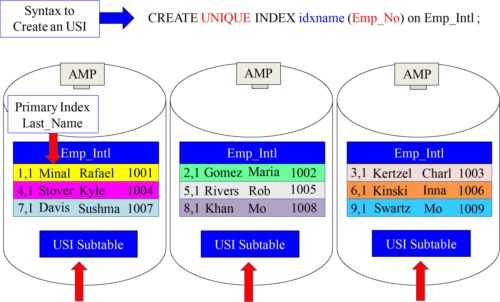
Chapter 5 - Secondary Indexes

“People can have the Model T in any color – so long as it’s black.”

- Henry Ford

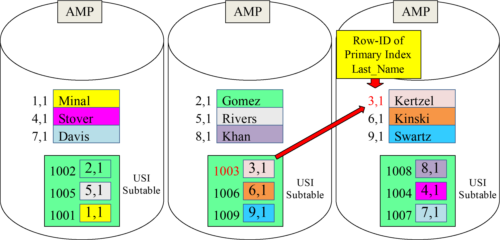
Creatinga Unique Secondary Index (USI)



A Subtable is created on each AMP anytime a secondary index is created

We created a Unique Secondary Index (USI) on the column Emp\_No named idxname. Remember, that a secondary index requires a **separate physical structure** (the subtable ), but a Primary Index , Partitioned Primary Index , Multi-Level Partitioned Primary index do NOT require a **separate physical structure**.

Whatis in a Unique Secondary Index (USI) Subtable?



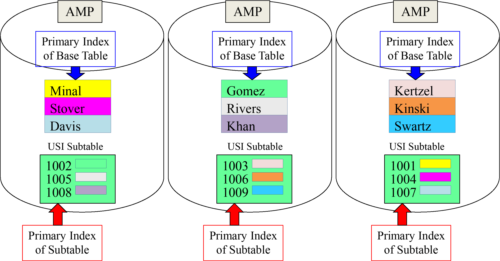
The USI Subtable contains two columns:

1. Emp\_No (The USI column)

2. Row-ID of the real Primary Index of the base table

Inside the secondary index subtable is the column Emp\_No (USI Column). Then, there is the corresponding Row-ID of the real Primary Index of the table. The USI subtable’s Primary Index is Emp\_No, and it is hashed and has a Row-ID like a normal table.

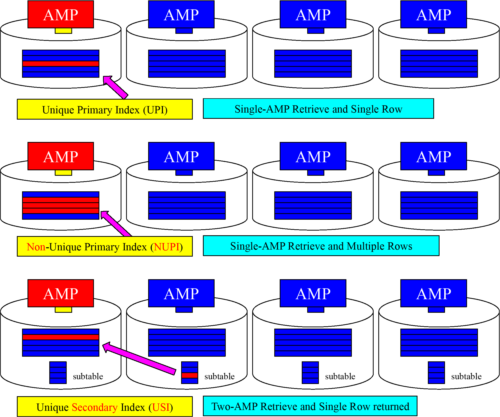
AUnique Secondary Index (USI) Subtable is hashed



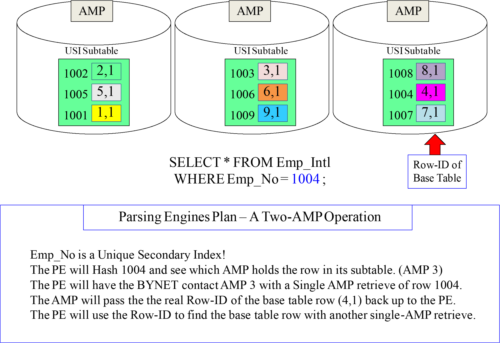
The USI Subtable spreads the rows evenly among the AMPs. The Primary Index of the Subtable is Emp\_No (The USI column).

Subtable rows are hashed by their Primary Index (Emp\_No) and distributed evenly. They contain the Row-ID of the base row, thus providing a pointer to the AMP and row.

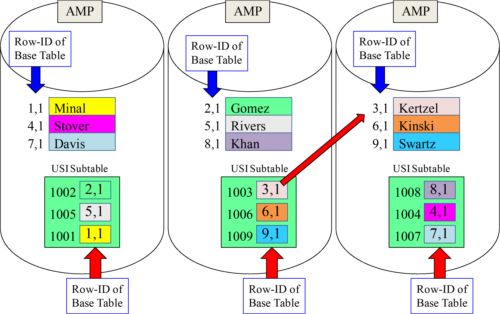
APrimary Index Query vs. a Unique Secondary Index Query



Howthe Parsing Engine uses the USI Subtable

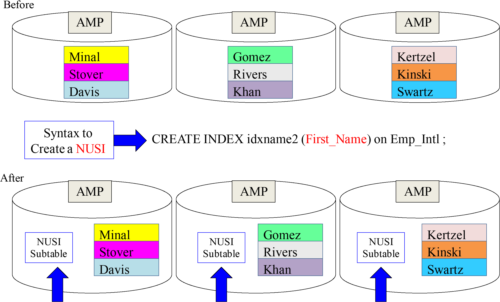


AUSI is a Two-AMP Operation



The first AMP is assigned to read the subtable and the second the base table. Two binary searches are performed in total, and one row is returned.

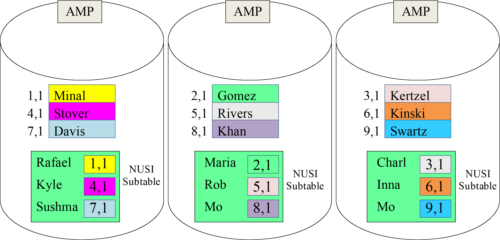
Creatinga Non-Unique Secondary Index (NUSI)



A Subtable is created on each AMP

We just created a Non-Unique Secondary Index on the column First\_Name.

Whatis in a Unique Secondary Index (USI) Subtable?



The NUSI Subtable only contains two columns:

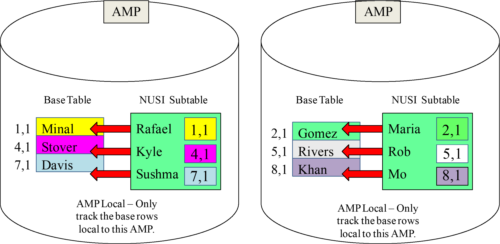
1. First\_Name (The NUSI column)

2. Row-ID of the real Primary Index of the base table

Inside the secondary index subtable is First\_Name column (NUSI Column). Then, there is the corresponding Row-ID of the real Primary Index of the table. The NUSI rows get their own Row-ID, but they are not hashed to different AMPs and stay AMP local.

.

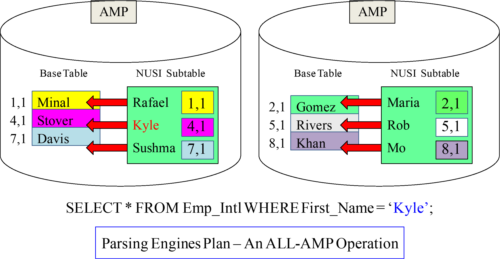
Non-Unique Secondary Index (NUSI) Subtable is AMP Local



The NUSI Subtable is AMP local because all values in the subtable are of those in the base table (on the same AMP).

Subtable rows match those of the base rows on the same AMP which is why they call it AMP Local. A NUSI query always searches all AMPs, but the intent is not to do a Full Table Scan. If there are 50 AMPs, then a minimum of 50 binary searches are done.

Howthe Parsing Engine uses the NUSI Subtable



First\_Name is a Non-Unique Secondary Index!

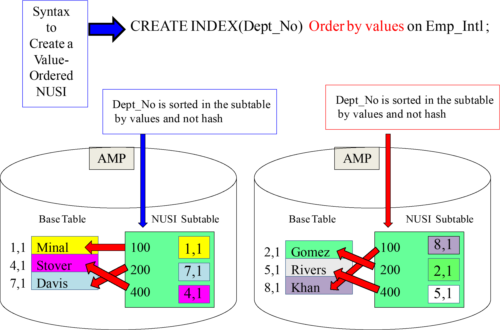
The Parsing Engine will order each AMP to check if they have a ‘Kyle’ in their NUSI Subtable.

Each AMP will simultaneously perform a binary search on their NUSI subtable.

If an AMP has a ‘Kyle’ the PE will order them to retrieve the base row (AMP Local).

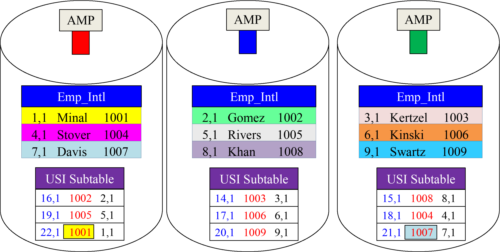
If there are 50 AMPs then all 50 AMPs perform a binary search simultaneously and if they find a “Kyle” they perform another binary search on their local base table.

Creatinga Value-Ordered NUSI



The Secondary Index Subtable will be sorted by the Values and not in Hash Order.

Quiz– Answer the Tough USI Questions



1) How are both tables sorted? \_\_\_\_\_\_\_\_

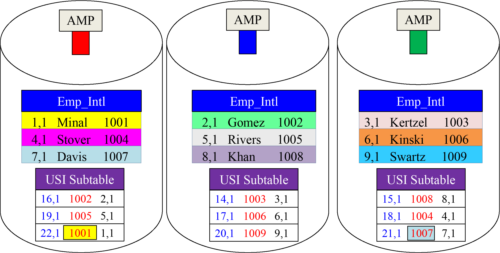
2) What was the Row-ID when Minal was hashed? \_\_\_\_\_\_\_

3) Looking in the subtable what is the Row-ID of the base for employee 1006? \_\_\_\_\_\_\_\_

4) When 1006 was placed in the subtable, which bucket in the hash map was chosen? \_\_\_\_\_\_\_\_\_\_\_\_\_\_

5) How many times is the Hash Map consulted on a query using a USI in the WHERE Clause? \_\_\_\_\_\_\_

Answerto Quiz – Answer the Tough USI Questions



1) How are both tables sorted? Row-ID

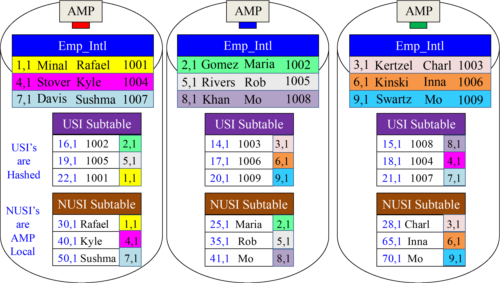
2) What was the Row-ID when Minal was hashed? 1,1

3) Looking in the subtable what is the Row-ID of the base for employee 1006? 6,1

4) When 1006 was placed in the subtable, which bucket in the hash map was chosen? Bucket 17

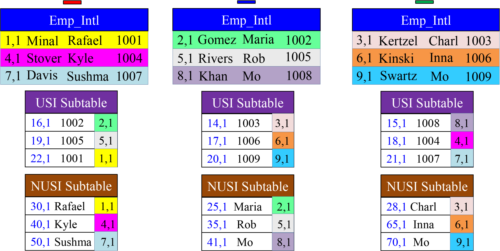
5) How many times is the Hash Map consulted on a query using a USI in the WHERE Clause? 2

APicture with a Base Table, USI, and NUSI Subtable



The Base Table has a Primary Index of Last\_Name. The USI was created on EmpNo and the NUSI on First\_Name. The USI rows are hashed to different AMPs, but the NUSI rows are AMP local. Both subtables contain the same Base Table Row-IDs.

Quiz– Tough Questions on the USI and NUSI Subtables



1) Why did the USI subtable row for Stover end up on AMP 3? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

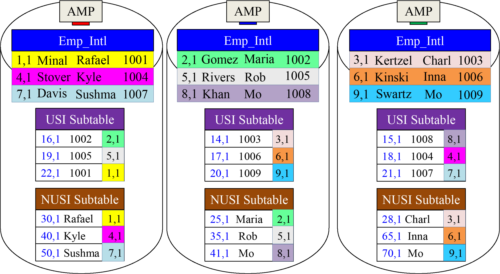
2) Why did NUSI subtable row for Stover end up on AMP 1? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) What is the Row Hash of Kyle? \_\_\_\_\_\_\_\_\_\_

4) If a NUSI Subtable is AMP Local, then what is the purpose of hashing the first name?\_\_\_\_\_\_\_\_\_\_\_\_

5) How many AMPs above would be in a USI query, and how many AMPS above in a NUSI query? \_\_\_ \_\_\_

Answer– Tough Questions on the USI and NUSI Subtables



1) Why did the USI subtable row for Stover end up on AMP 3? EmpNo 1004 was hashed and sent there.

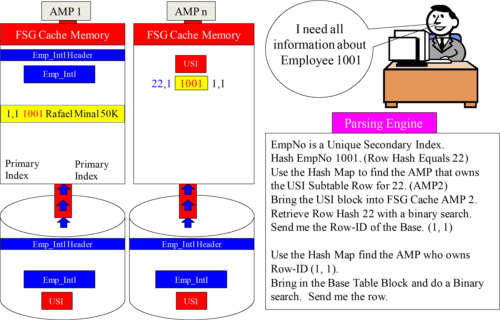
2) Why did NUSI subtable row for Stover end up on AMP 1? NUSI rows are always AMP local.

3) What is the Row Hash of Kyle? 40

4) If a NUSI Subtable is AMP Local, then what is the purpose of hashing the first name? Binary Search Sort

5) How many AMPs above would be in a USI query, and how many AMPS above in a NUSI query? 2 (3 all)

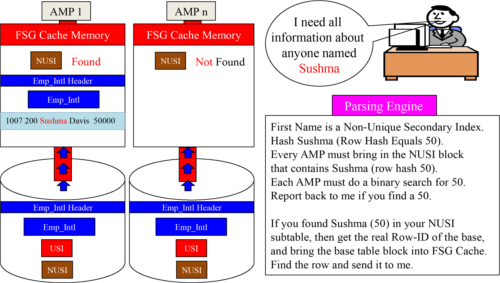
AQuery Using an USI Only Moves Two Blocks



Only one USI block and only one base table block are used on a USI search.

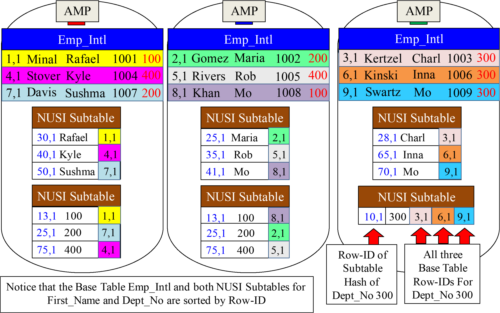
.

AQuery Using A NUSI Always Uses All AMPs



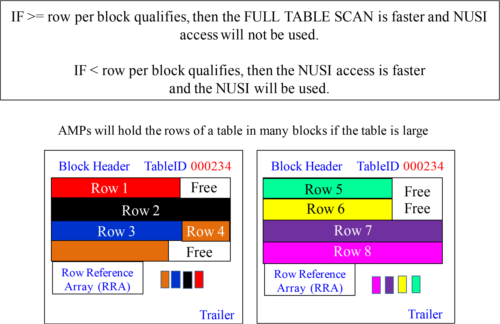
All AMPs move their NUSI block into FSG cache and do a binary search. For any AMP who finds they have a Sushma, they then move their base block into FSG Cache and they retrieve the row. A NUSI query is much slower than an USI query.

TwoNon-Unique Secondary Indexes (NUSI) on a Table



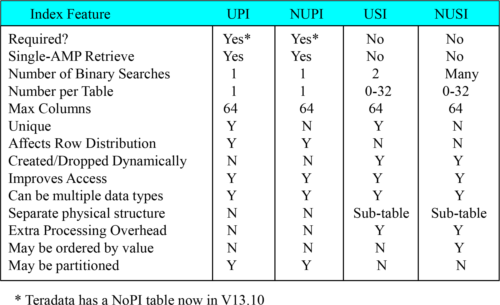
Notice that both NUSI subtables have the same Base Table Row-IDs (all 9 Row-IDs).

Howthe PE Decides on the NUSI or the Full Table Scan



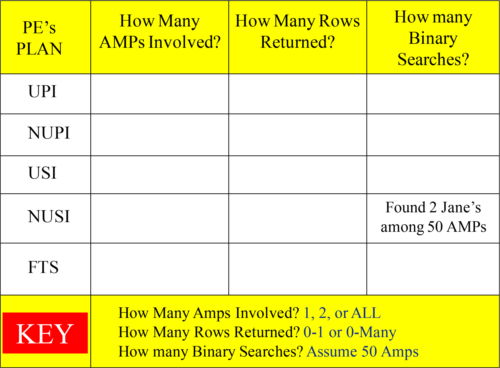
NUSI usage depends on the rows per block that qualify in the SQL WHERE clause**. A Full Table Scan done for values that represent a large percent of table.** A NUSI is used for values that represent a tiny percent of the table. When statistics are collected on a NUSI column, the PE has a much better idea of the sizing.

PrimaryIndex vs. Secondary Index

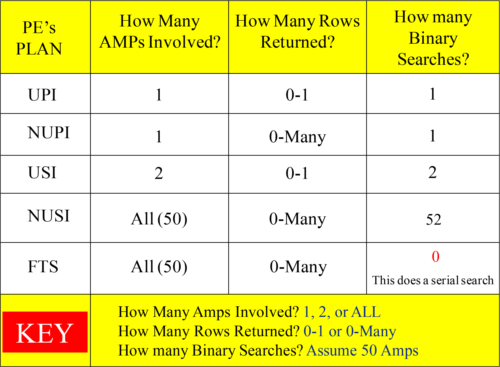


Primary indexes are required , but secondary indexes are optional . All tables must have a method of distributing rows among AMPs and that is done by the Primary Index. A table can only have **one primary index** , but it can have up to **32 secondary indexes**. Both primary and secondary indexes can contain up to 64 columns combined . The secondary index does not affect the distribution of rows. Rows are only distributed according to the Primary Index values. Primary indexes are hashed and stored so there is no extra overhead. Secondary indexes have a **subtable** that has extra overhead to **maintain** and **store.**

TheBigger Quiz



TheBigger Quiz Answers



MultipleChoice DBA

A large table called Big\_Table exists. Two completely different departments query Big\_Table regularly. The Sales Department always queries on a Unique column. The Marketing Department always queries on a Non-Unique column.

You are the DBA and it is up to you to choose the best indexing for Big\_Table. Choose the multiple choice answer that will work best so both departments are happy.

A) UPI and NUPI

B) USI and NUSI

C) UPI and NUSI

D) NUPI and USI

You have to pick either A, B, C or D. What do you think will work as the best indexing for Big\_Table? Remember that one department always queries on a Unique column, and the other department always queries on a Non-Unique column. Good Luck!

MultipleChoice DBA

A large table called Big\_Table exists. Two completely different departments query Big\_Table regularly. The Sales Department always queries on a Unique column. The Marketing Department always queries on a Non-Unique column.

You are the DBA, and it is up to you to choose the best indexing for Big\_Table. Choose the multiple choice answer that will work best so both departments are happy with the query times.

A) UPI and NUPI

B) USI and NUSI

C) UPI and NUSI

D) NUPI and USI

D is the best answer because the NUPI will provide the department who always queries on a Non-Unique column with a NUPI Single-AMP retrieve. The department who always queries on a Unique column can use the USI, which is a 2-AMP retrieve.

Whatare the Big Four Tactical Queries?

There are four Tactical Queries. A Tactical Query can be depended on to run in sub second time. Can you name the four Tactical Queries from fastest to slowest?

1)

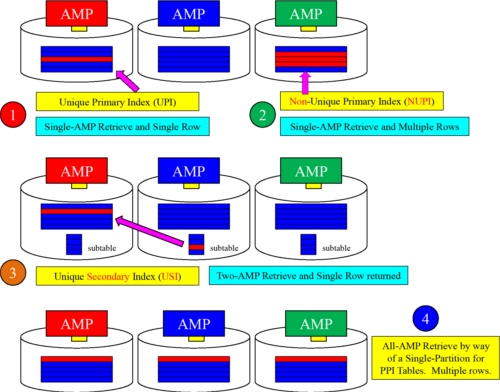
2)

3)

4)

Tactical queries will run very fast and they can be depended upon to run in sub second time. Can you list the four fastest queries in order from fastest to slowest (all must run in under sub second time)?

TheFour Tactical Queries



Tactical queries will run very fast, and they can be depended upon to run in sub second time. The fastest is, of course, example one with an UPI in the Where clause. Then, it is the NUPI in the Where clause. Then, it is the USI in the WHERE clause. Finally, it is an All AMP retrieve by way of a Single Partition.