TERADATA

Introduction to Teradata

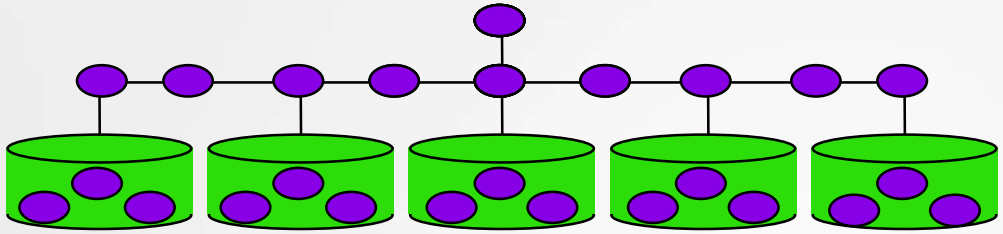
Big Classes .com

TERADATA

How Teradata Works

Big Classes .com

How Does Teradata Store Rows?



- Teradata uses hashing algorithm to randomly and evenly distribute data across all AMPs.
- The rows of *every* table are distributed among all AMPs - and ideally will be evenly distributed among all AMPs.
- Each AMP is responsible for a *subset* of the rows of each table.
- Evenly distributed tables result in evenly distributed workloads.
- The data is not placed in any particular order

The benefits of unordered data include:

- No maintenance needed to preserve order, and
- It is independent of any query being submitted.

The benefits of automatic data placement include:

- Distribution is the same regardless of data volume
- Distribution is based on row content, not data demographics

Primary Indexes

- The mechanism used to assign a row to an AMP
- A table must have a Primary Index
- The Primary Index cannot be changed

UPI

- If the index choice of column(s) is unique, we call this a *UPI* (Unique Primary Index).
- A UPI choice will result in even distribution of the rows of the table across all AMPs.

UPI's guarantee even data distribution and eliminate duplicate row checking.

NUPI

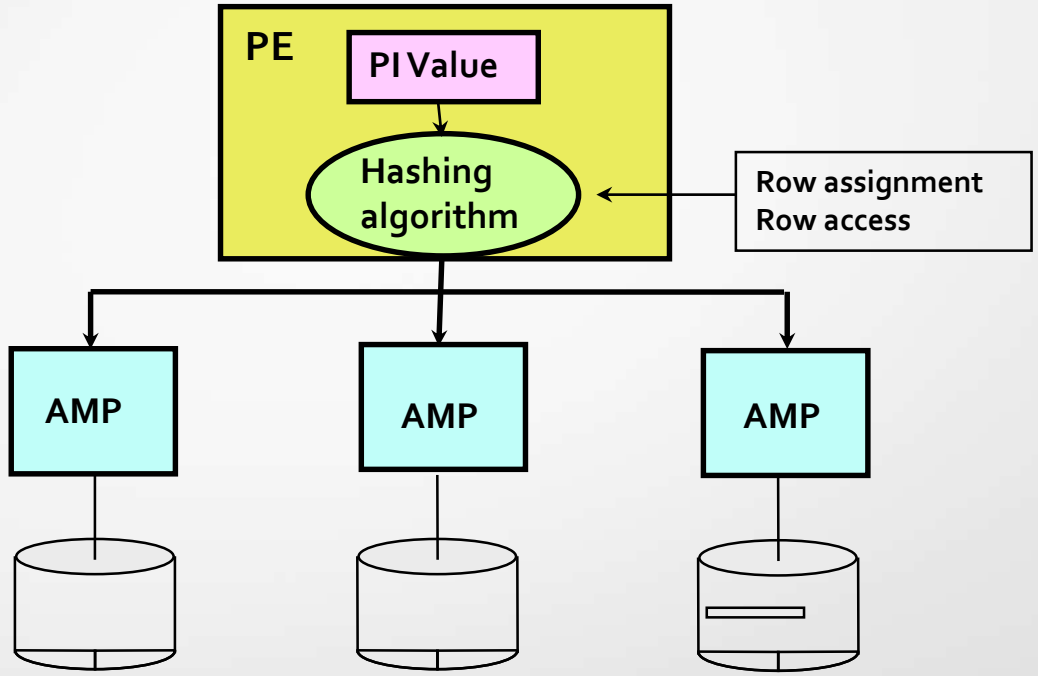
- If the index choice of column(s) isn't unique, we call this a *NUPI* (Non-Unique Primary Index).
- A NUPI choice will result in even distribution of the rows of the table proportional to the degree of uniqueness of the index.

Why would you choose an Index that is different from the Primary Key?

- Join performance
- Known access paths

Data Storage based on Primary Index

- The value of the Primary Index for a specific row determines its AMP assignment.
- This is done using the hashing algorithm.



Accessing the row by its Primary Index value is:

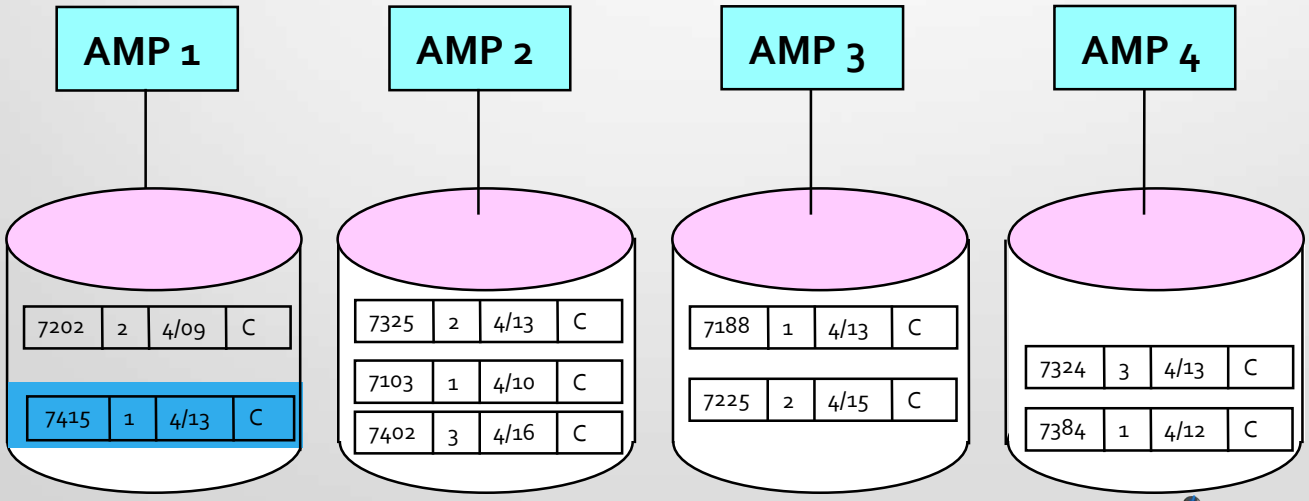
- Always a *one-AMP* operation
- The most efficient way to access a row

Row Distribution Using a UPI

Order

Order Number	Customer Number	Order Date	Order Status
PK			
UPI			
7325	2	4/13	O
7324	3	4/13	O
7415	1	4/13	C
7103	1	4/10	O
7225	2	4/15	C
7384	1	4/12	C
7402	3	4/16	C
7188	1	4/13	C
7202	2	4/09	C

The PK column(s) will often be used as a UPI.
PI values for Order_Number are known to be unique (it's a PK).
Teradata will distribute different index values evenly across AMPs.
Resulting row distribution among AMPs is uniform.

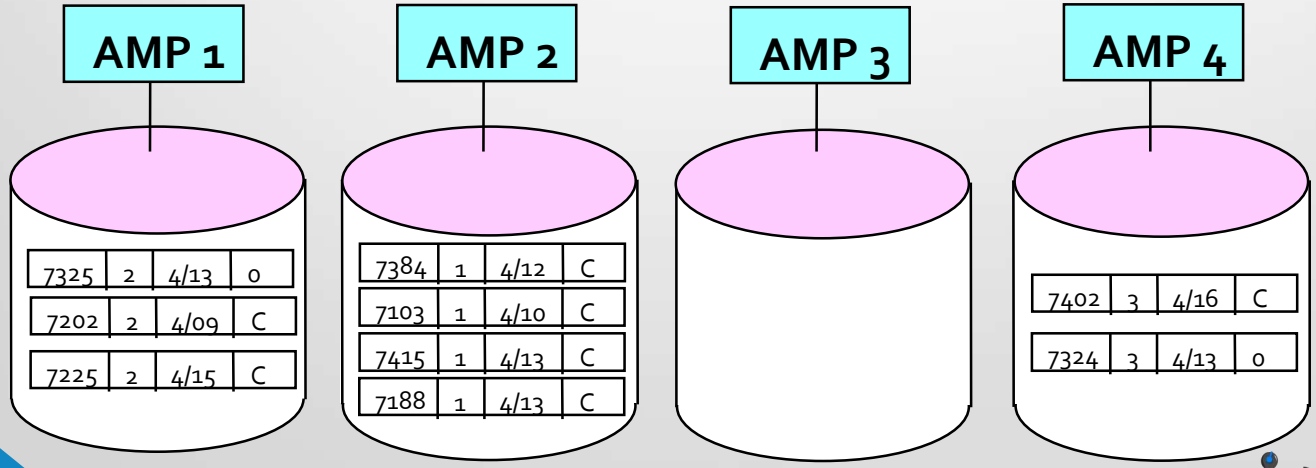


Row Distribution Using a NUPI

Order

Order Number	Customer Number	Order Date	Order Status
PK			
	NUPI		
7325	2	4/13	O
7324	3	4/13	O
7415	1	4/13	C
7103	1	4/10	O
7225	2	4/15	C
7384	1	4/12	C
7402	3	4/16	C
7188	1	4/13	C
7202	2	4/09	C

Customer_Number may be the referred access column for ORDER table, thus a good index candidate. Values for Customer_Number are non-unique and therefore a NUPI. Rows with the same PI value distribute to the same AMP causing row distribution to be less uniform or skewed.



Secondary Indexes

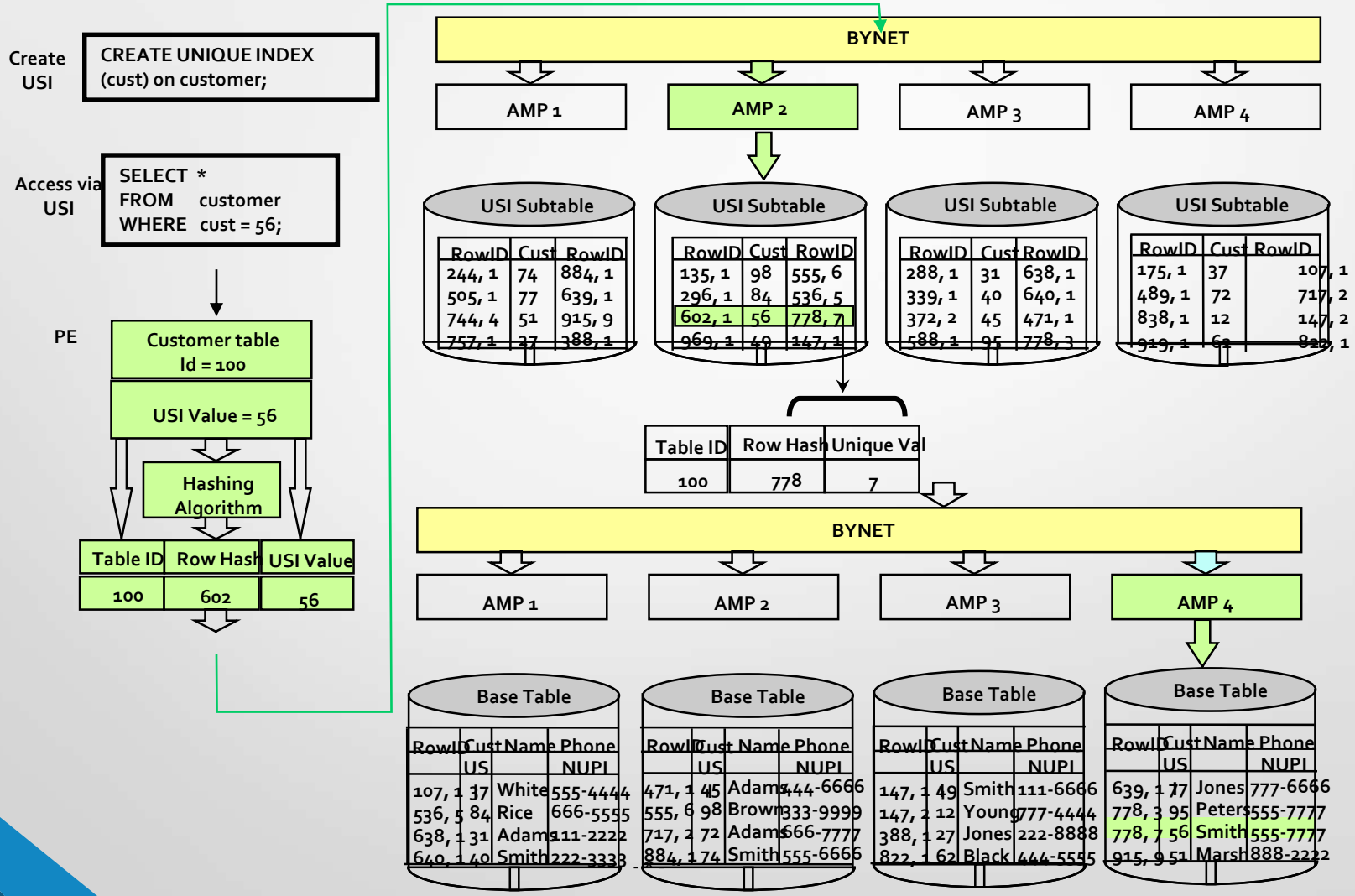
Three general ways to access a table:

- **Primary index access** (***one*-AMP access)**
- **Secondary index access** (***two*-or *all*-AMP access)**
- **Full Table Scan** (***all*-AMP access)**

A secondary index is an alternate path to the rows of a table.
A table can have from 0 to 32 secondary indexes.
Secondary indexes:

- Do not affect table distribution.
- Add overhead, both in terms of disk space and maintenance.
- May be added or dropped dynamically as needed.
- Are chosen to improve table performance.

Unique Secondary Index (USI) Access



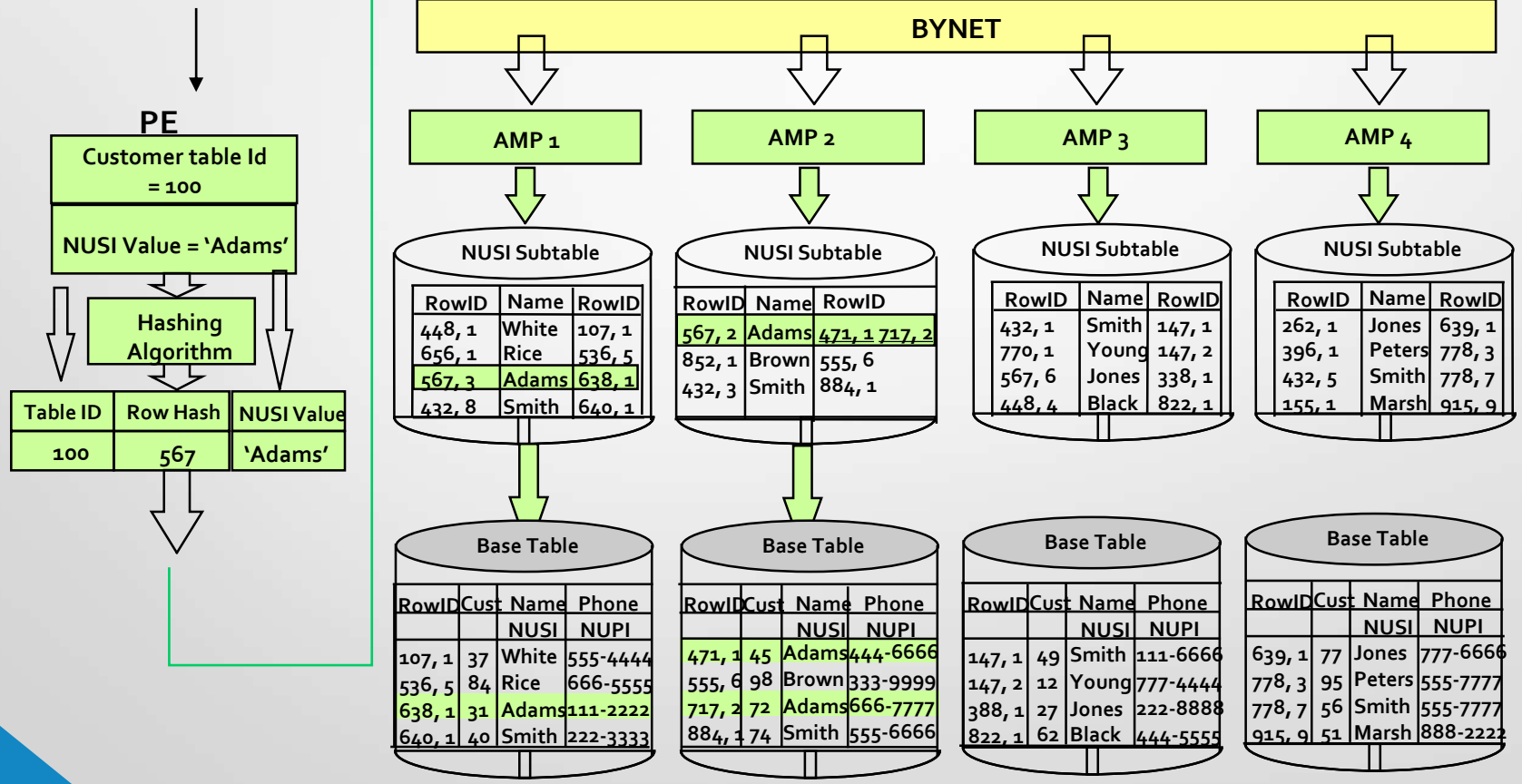
Non-Unique Secondary Index (NUSI) Access

Create NUSI

```
CREATE INDEX (name) on
customer;
```

Access via NUSI

```
SELECT *
FROM   customer
WHERE  name = 'Adams';
```



Comparison of Primary and Secondary Indexes

Index Feature	Primary	Secondary
Required?	Yes	No
Number per Table	1	0-32
Max Number of Columns	16	16
Unique or Non-Unique?	Both	Both
Affects Row Distribution	Yes	No
Created/Dropped Dynamically	No	Yes
Improves Access	Yes	Yes
Multiple Data Types	Yes	Yes
Separate Physical Structure	None	Sub-table
Extra Processing Overhead	No	Yes

Primary Keys and Primary Indexes

Indexes are conceptually different from *keys:*

A *PK* is a relational modeling convention which uniquely identified each row.

A *PI* is a Teradata convention which determines how the rows are stored and accessed.

Primary Key	Primary Index
Logical concept of data modeling	Physical mechanism for access and storage
Teradata doesn't need to recognize	Each table must have exactly one
No limit on column numbers	16-column limit
Documented in data model (Optional in CREATE TABLE)	Defined in CREATE TABLE statement
Must be unique	May be unique or non-unique
Uniquely identifies each row	Used to place and locate each row on an AMP
Values should not change	Values may be changed (Del+ Ins)
May not be NULL—requires a value	May be NULL
Does not imply an access path	Defines most efficient access path
Chosen for logical correctness	Chosen for physical performance

A significant percentage of tables may use the same columns for both the PK and PI.

A well-designed database will use a PI that is different from the PK for some tables.

Big Classes .com

Learn Teradata Online

Contact:
USA: +1 732 325 1626
India: +91 800 811 4040
Mail: info@bigclasses.com

 /bigclasses  /bigclasses  /bigclasses

http://bigclasses.com/teradata-online-training.html

Big Classes .com

Thank you

Watch Teradata DEMO Video On YouTube

www.youtube.com/user/bigclassescom