

```

4780 GOTO 5000
4790 :
4800 REM
4801 REM
4802 REM
4803 REM
4810 :
4820 PRINT
4825 W=V+1
4830 FOR X
4835 FOR I
4840 PRINT
4850 NEXT:
4860 PRINT
4870 FOR I
4880 IF MD
(I+1);:GOT
4890 PRINT
4900 NEXT
4910 PRINT
4920 FOR I
4925 PRINT
4930 IF MD
Q";:GOTO 4
4935 PRINT
4940 NEXT:PRINT
4950 PRINT"#####";
4960 FOR I=2 TO 24 STEP 2
4965 PRINT"|";
4970 IF MD*(I+W-1)="#####" THEN PRINT"#####";
M$(I)"|";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT"#####"

```

# Teradata Basic

RNDr. Ondřej Zýka



# Něco z historie

- Založena v roce 1979 v garáži v Kalifornii (Brentwood).
- Původně součástí NCR
- Teradata symbolizuje schopnost spravovat extrémní množství dat.
- Primárně určena pro datové sklady a BI aplikace.
- Založena na „shared nothing“ architektuře umožňující lineární škálovatelnost.
- V současnosti využívána v největších datových skladech
  - Svět
    - Wall-Mart
    - DHL
    - Bank of Amerika
    - Coca Cola, ...
  - V Česku
    - KB, ČSOB, ..
    - Česká pojišťovna, Česká pošta, ...
    - TO2, Vodafone, ...



# Komponenty

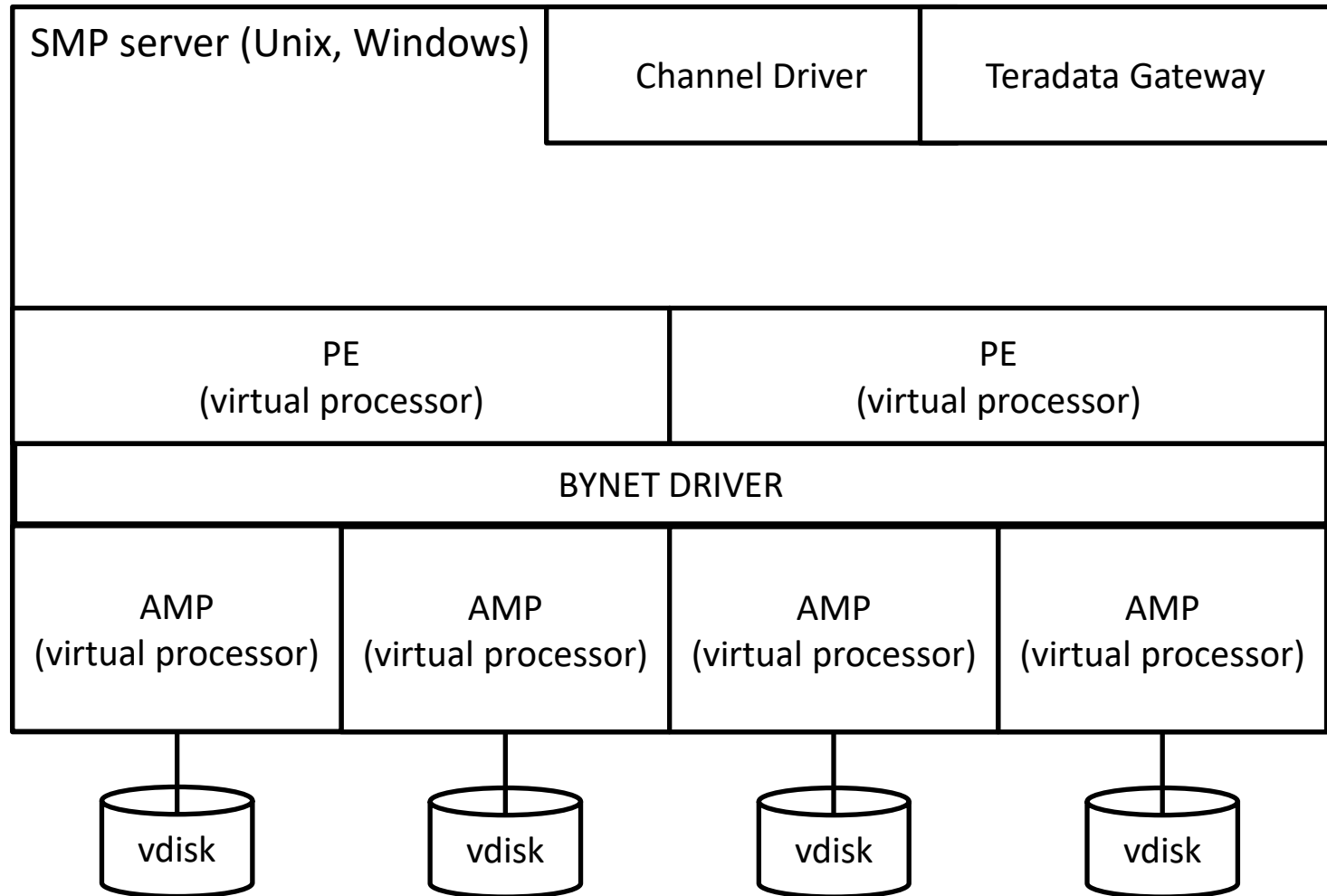
- AMP – Access modul processor
  - Virtuální procesor s vlastním diskem (jednotka Shared Nothing architektury)
  - Zodpovědný za získání požadovaných dat ze svých diskových úložišť
  - Realizuje **fyzické zpracování dat** (řazení, agregace, formát, konverze, ...).
  - Reaguje na instrukce zasláné od PE prostřednictvím BYNET.
- Parsing engine – PE
  - Virtuální procesor
  - Parsuje a optimalizuje dotazy, předává požadavky jednotlivým AMPům
- Node (SMP uzel)
  - Počítač s architekturou SMP, na kterém běží jednotlivé Virtuální procesory



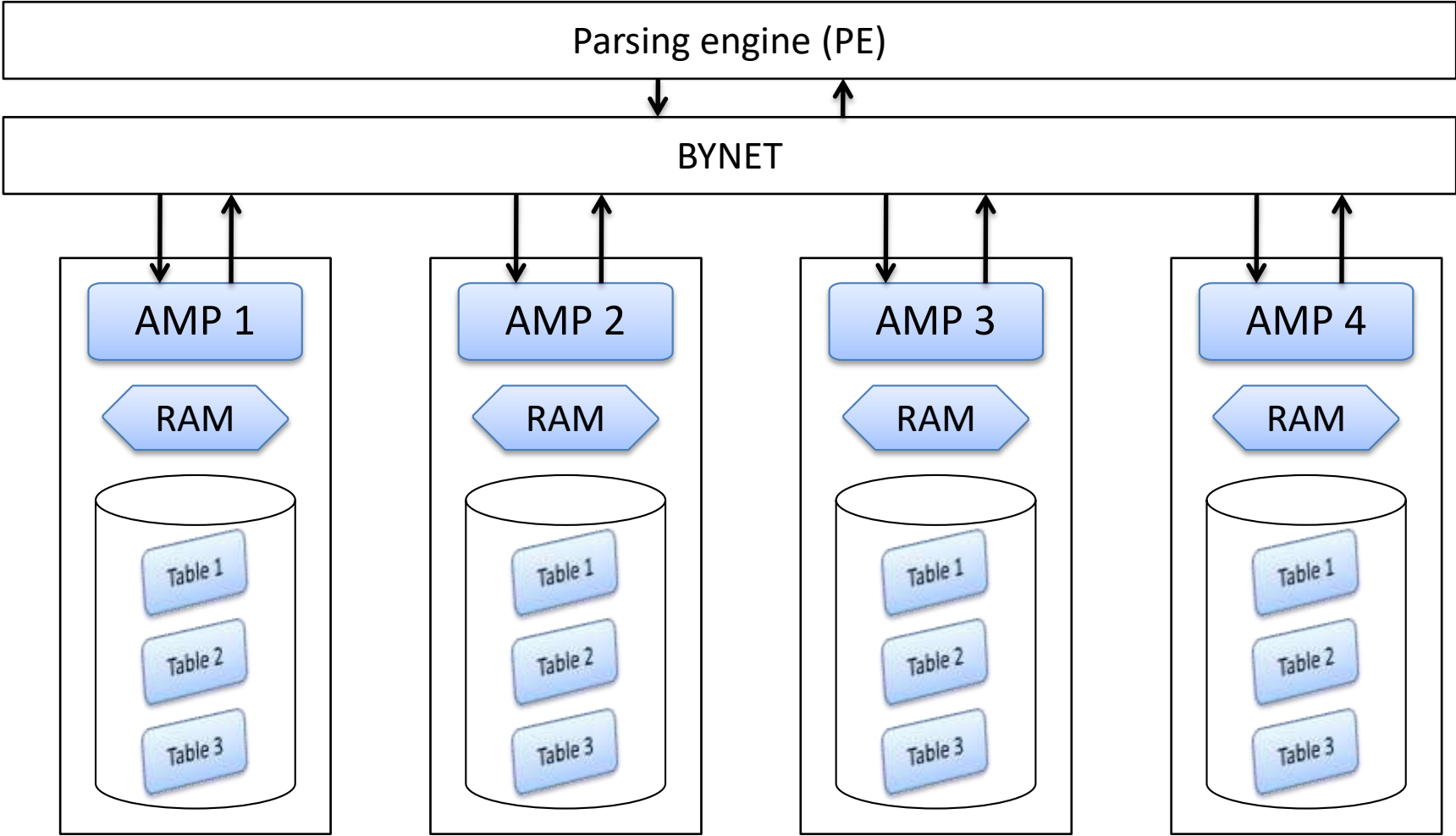
# Komponenty

- BYNET
  - Komunikační kanál (duplikovaný) pro předávání požadavků a výsledků
  - Zpráva je směrována k určitému AMPu, který provede výpočet, a odezva (výsledek) je směrován zpět k PE prostřednictvím BYNET
    - **Broadcast** – zpráva je zaslána všem uzlům v systému a následně i všem AMPům.
    - **Point-to-point** – zpráva je zaslána specifickému uzlu v systému a jen jeho AMPům.
  - **Tolerance chyb** – dva paralelní BYNET (0,1) a každý z nich redundance.
  - **Load Balancing** – zátěž je rozkládána mezi BYNET0 a BYNET1.
- BYNET Driver
  - Komunikační kanál mezi virtuálními procesory (AMP, PE) v rámci jednoho Node

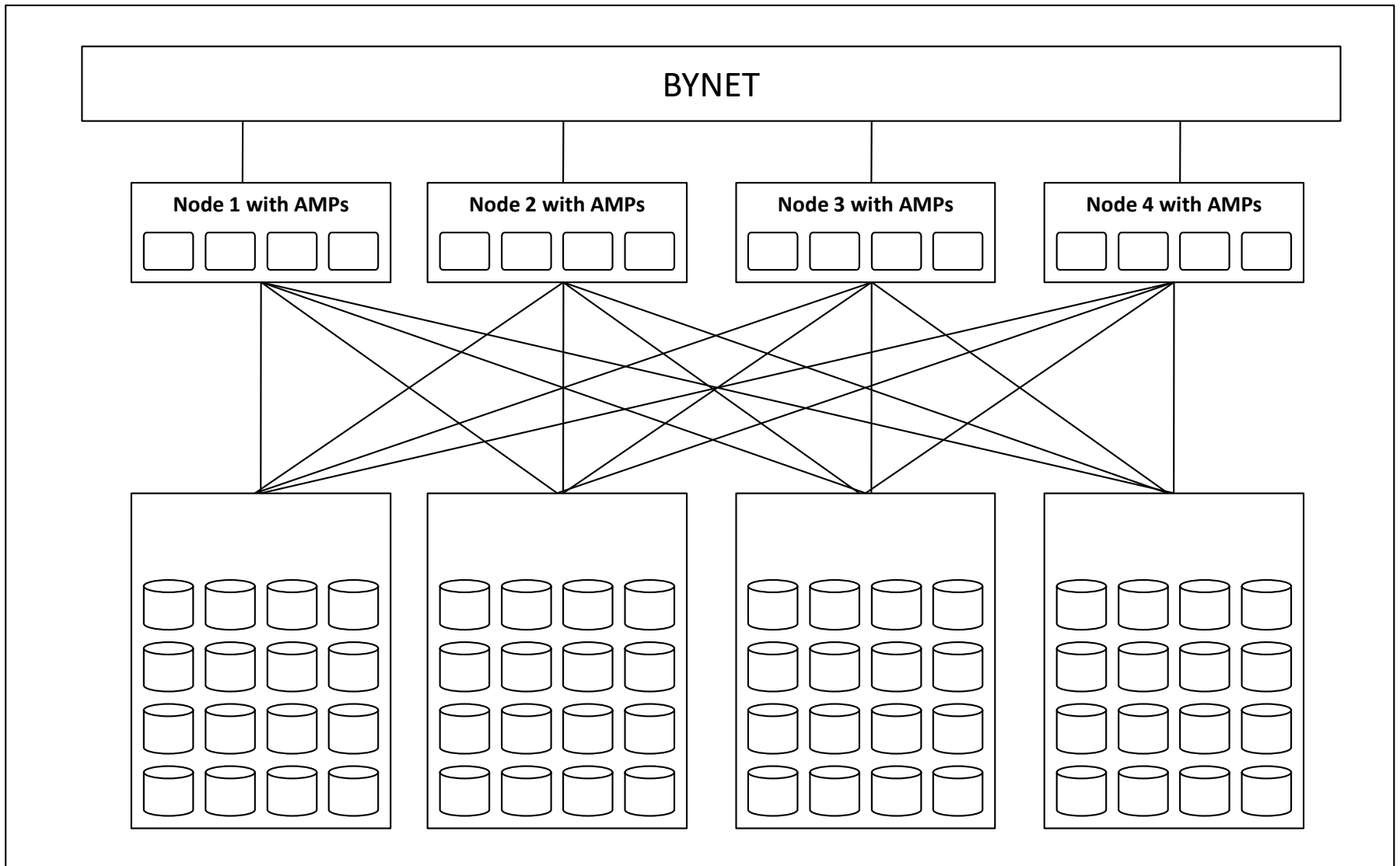
# Node (SMP uzel)



# Architektura „Shared nothing“



# Fyzická architektura



# Diskové prostory

- Permanent space
  - Limit na uživatele
  - Data, indexy, fallback, ...
  - Rovnoměrně distribuován na jednotlivé AMPy
- Spool space
  - Globální limit na uživatele
  - Diskový prostor pro mezivýsledky
  - Všechn nepřidělený prostor
- Temp space
  - Global temporary table



# Uložení a přístup k datům

Primární klíč	Primární index
Logický koncept z LDM.	Fyzický mechanismus pro přístup k datům.
Teradata nevyžaduje jeho definici.	Každé tabulce musí být přiřazen.
Musí být unikátní.	Může být unikátní, ale nemusí být.
Jednoznačně identifikuje řádek.	Obecně nejednoznačná identifikace.
Neakceptuje NULL hodnoty.	Akceptuje NULL hodnoty.
Neimplikuje přístup k datům.	Implikuje nejefektivnější přístup k datům.

# Primární index a adresace

## Hashování hodnot primárního indexu

- **Řádky** všech tabulek jsou v Teradatě **distribučovány dle hash**.

## Proces adresace

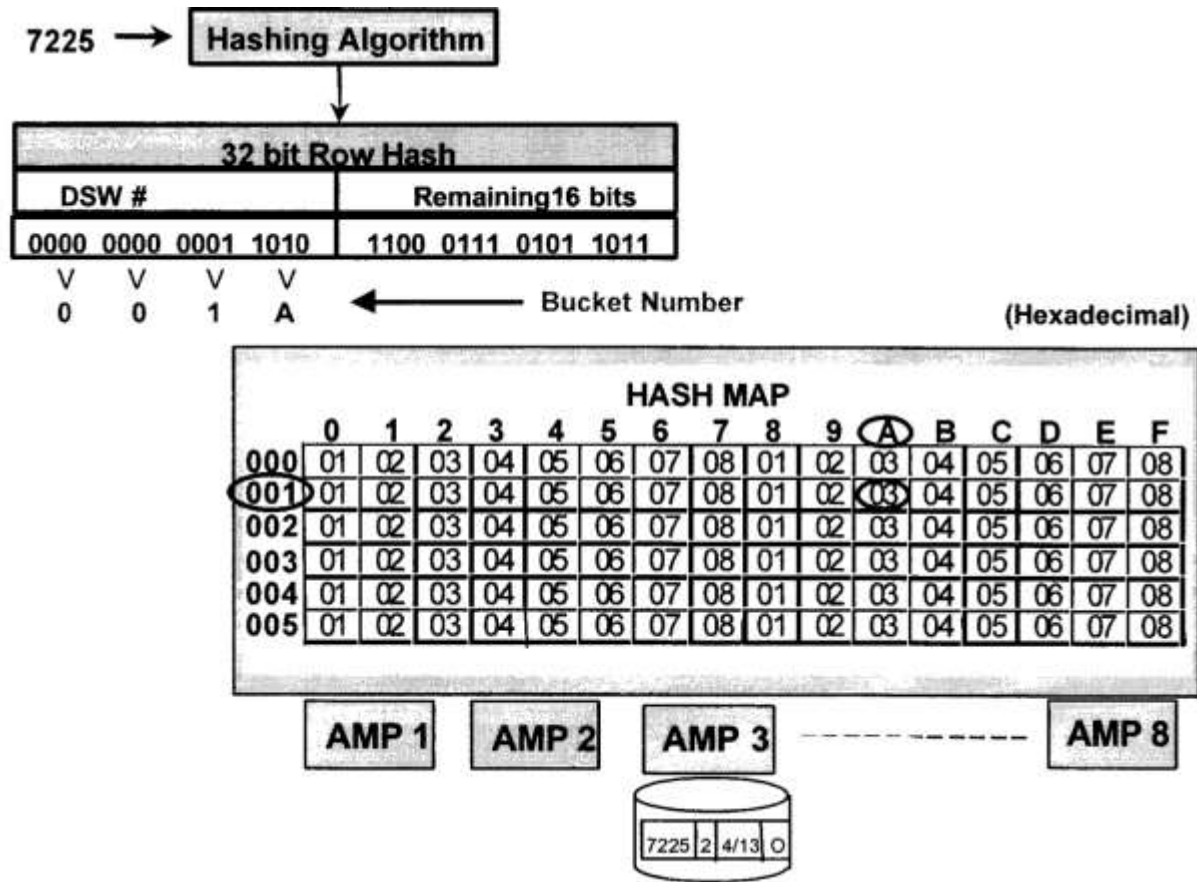
1. Z hodnoty primárního indexu je vypočten hash **32-bit Row Hash**.
2. Prvních **16 bitů** se nazývá **Destination Selection Word (DSW)**.
3. Řádek se svou Row Hash je poslán na komunikační vrstvu **BYNET**.
4. Na základě **DSW** je v **BYNET Hash Map** identifikován **cílový AMP**.
5. Celých **32 bitů Row Hash** je cílovým AMPem použito k **nalezení řádku**.

## Identifikace řádků

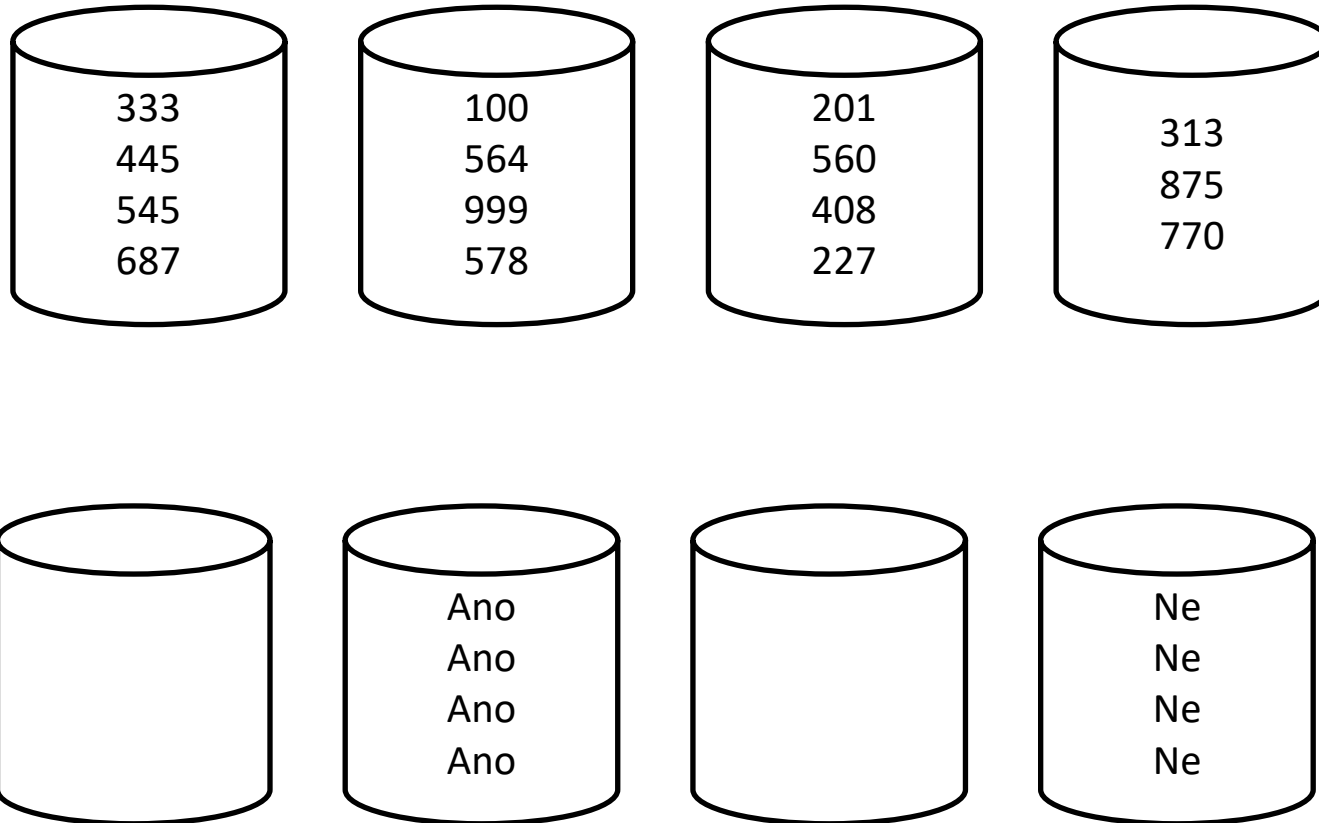
- Dva různé PI řádku mohou vést ke stejnému row hash – **Hash Synonym**.
- V případě NUPI mohou existovat v PI duplicitní řádky se stejným row hash.

# Primární indexy - Hash mapa

- Hash mapa pro každé DSW asociuje AMP.
- K hash mapě v rámci adresace přistupuje BYNET.
- Na základě hash mapy je pro daný řádek „osloven“ příslušný AMP.



# Distribuce řádků tabulky



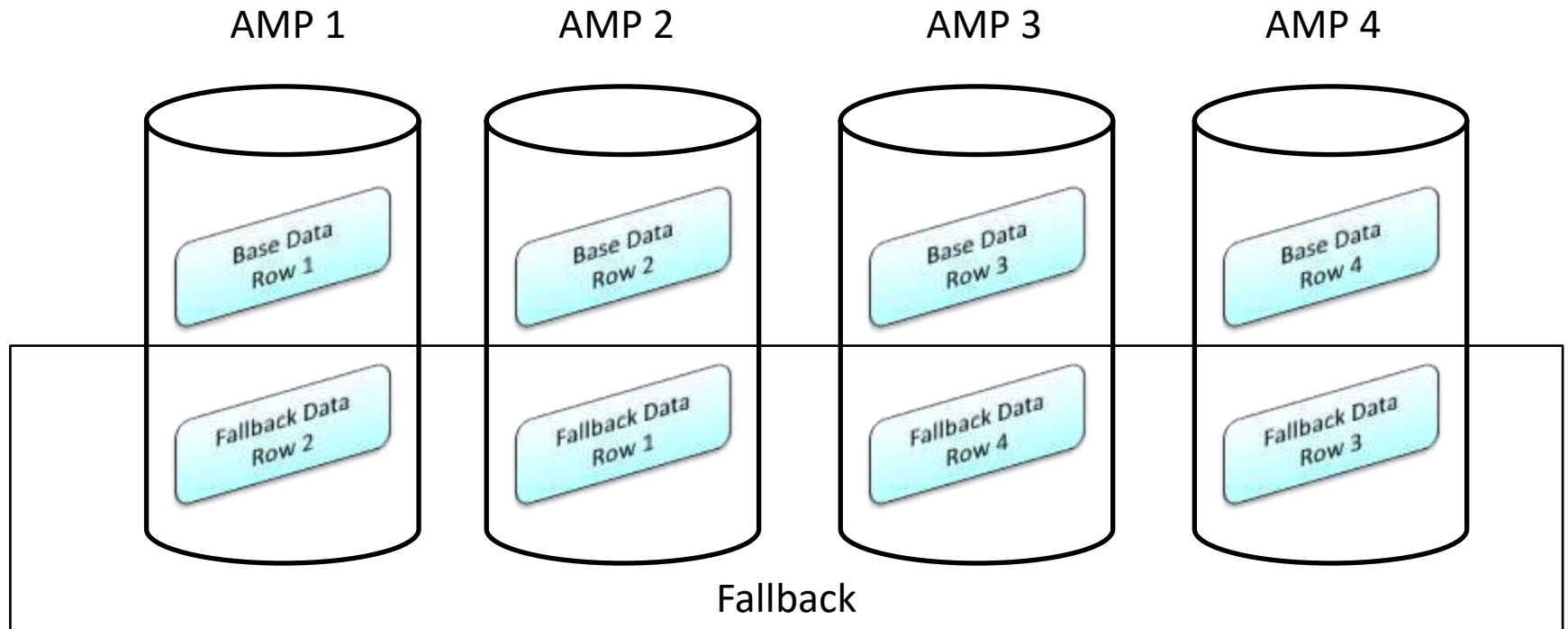
# Podpora výkonu databáze

- Indexy
  - Sekundární indexy (unikátní a neunikátní)
  - Covered Index
- Statistiky
- Partitioning
  - Partition Primary Index
  - Data jsou uložena na AMPu nejdříve podle partition, pak podle Hash value
- Join algoritmy
  - Redistribuce a duplikace
  - Merge join
  - Exclusion join
  - Hash join
  - Nested join
  - Product join
- Pravidla pro vytváření logického a fyzického modelu

# Podpora dostupnost

- Transient Journal
  - Řídí transakce a rollback. Na jednotlivých AMPech. Po ukončení transakce smazán.
- Fallback
  - Zabezpečení proti výpadku jednoho AMPu, sekundární záznam (kopie) je uložen na jiném AMPu
- Permanent Journal
  - Podpora obnovy databáze k určitému okamžiku
  - PJ je periodicky odléván na externí médium; zálohovány jsou pouze inkrementy.
- Down AMP Journal
  - Podpora zotavení AMPu při pádu

# Fallback



# Nástroje a služby

- Nástroje správy odpovídají „stáří“ databáze.
- Systém primárně zaměřen na výkon.
- Nemá vlastní nástroje na prezentaci a analýzu dat – spolupracuje se všemi velkými hráči na trhu
  - Microstrategy,
  - Cognos,
  - Oracle BI,
  - SAP BusinessObject,
  - Microsoft Reporting services.





# Industriální logické modely

- Teradata Communications Logical Data Model
- Teradata Financial Services Logical Data Model
- Teradata Healthcare Logical Data Model
- Teradata Insurance Logical Data Model
- Teradata Manufacturing Logical Data Model
- Teradata Media Logical Data Model
- Teradata Retail Logical Data Model
- Teradata Transportation and Logistics Logical Data Model
- Teradata Travel and Hospitality Industry Logical Data Model
- Teradata Utilities Logical Data Model



# Aplikace podle obchodních požadavků

- Kumulovaná zkušenost ze stovek projektů datových skladů a projektů BI.
- Obsah jednotlivých modelů:
  - Více úrovní pohledu
    - Konceptuální pohled,
    - Funkční oblasti (Kontrakt, Účet, Kanál, Událost, Kampaň, Party, Produkt, ...),
    - Detailní logický model.
  - Podrobný popis jednotlivých entit včetně atributů a relací, jejich vy
  - Textový popis i modely.



# Aplikace podle obchodních požadavků

- Business Intelligence
- Data Mart Consolidation
- Master Data Management
- Tax and Revenue Management
  
- Customer Relationship Management
- Data Mining and Analytics
- Enterprise Risk Management
- SAP Integration
  
- Data Governance
- Data Warehouse Migration
- Financial Management



# Co si zapamatovat

- Jakou architekturu používá systém Teradata
  - Vysvětlete
- Popište hlavní komponenty systému Teradata
- Jaký je rozdíl mezi primárním klíčem a primárním indexem

```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT" ";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"XXXXXXXXXXXX";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"XXXXXXXXXXXX";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"XXXXXXXXXXXX";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)="  " THEN PRINT"
";:GOTO 4940
4935 PRINT " ";
4940 NEXT:PRINT" "
4950 PRINT"XXXXXXXXXXXX";
4960 FOR I=2 TO 24 STEP 2
4965 PRINT"|";
4970 IF MD$(I+W-1)="  " THEN PRINT"
"
M$(I)" ";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT" "

```



**Diskuse**

- Otázky
- Poznámky
- Komentáře
- Připomínky

