DATA PARTITIONING

AMR ELHELW

TECH VAULT

Scenario #1

order_id	order_date	customer_id	amount
1573	2022-05-28	16	648
1574	2023-01-26	10	921
1575	2023-09-25	3	810
1576	2022-01-25	18	1063
1577	2022-06-28	8	753
1578	2021-01-28	13	548
1579	2021-02-07	2	572
1580	2023-03-23	11	953
1581	2023-09-11	16	734
1582	2022-09-12	19	1070
1583	2023-04-07	12	385
1584	2023-07-31	16	930
1585	2023-09-22	9	716
1586	2023-07-17	15	766
1587	2022-12-20	8	1000
1588	2022-11-28	2	595
1589	2022-06-30	20	949
1590	2023-04-05	11	297
1591	2022-07-01	2	640
1592	2023-05-18	2	596
1593	2023-03-10	8	280
1594	2022-06-16	8	971
1595	2022-07-07	18	796
1596	2021-06-05	17	796

```sql
SELECT * FROM orders
WHERE order_date BETWEEN '2024-02-15' AND '2024-04-22'
```

- **Option 1: Full Table Scan**
  - Too slow
- **Option 2: Index on order_date**
  - Still need to access table to get other attributes

Amr Elhelw's TECH VAULT

Scenario #2

prod_id	name	category_id	quantity	notes
7	Prod 7	13	9	Q6MHA1AHfQH5bDK6nyWx7IEtSYxnk
8	Prod 8	13	27	u2mEigECohZbsqD8VwNboB6o5loxg
9	Prod 9	6	5	BTaVF7o19EBvfKxVxuDjVnK5IFjlVH
10	Prod 10	10	15	sSzy7BrPkF15tydp1dbmCPei7CynhH
11	Prod 11	7	5	riOvLKbqxCVliJs1cXNJwDondjuEf3
12	Prod 12	10	25	jQTDEQFCSU68qzvtWFF7D5W9kbwj
13	Prod 13	1	48	ZSqyyXulJE5lQGeld25Zq2R2kgBJih
14	Prod 14	2	24	cBcxwM4mfLclfNvTZgiqGT9aQXJhll
15	Prod 15	13	11	QvuVhElYs6M9l9hxFma9fyfzOTULlQ
16	Prod 16	3	44	4pazBYinEoJDj19h1Kb38GOfO8Z1AU

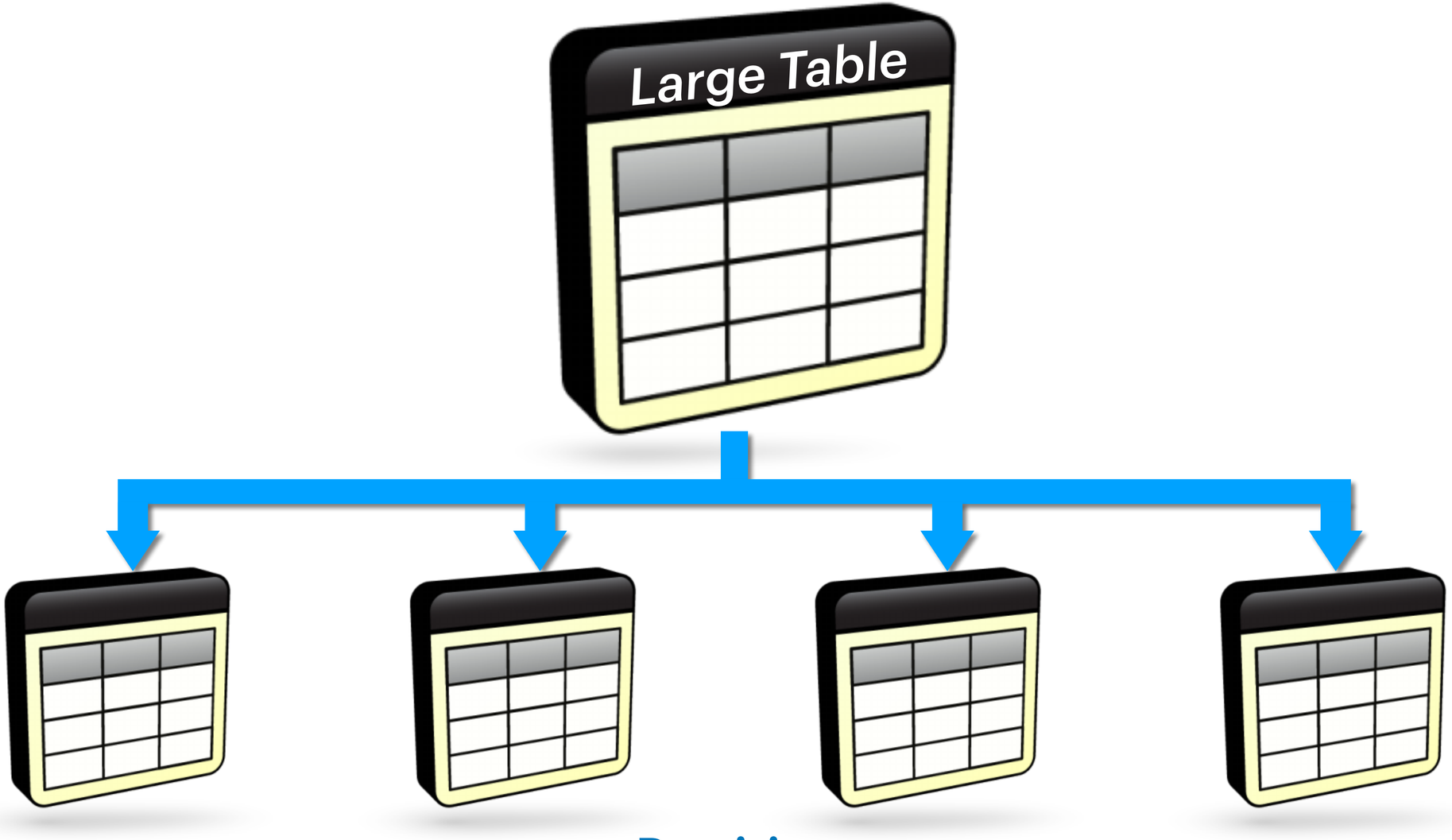
```sql
SELECT prod_id, name, quantity
FROM products;
```

```sql
SELECT category_id, sum(quantity)
FROM products
GROUP BY category_id;
```

- **Option 1: Full Table Scan**
  - Need to read full tuples (including the “notes” column) - too much I/O.

- **Option 2: Covering indexes for common queries**
  - Need many indexes - update cost too high
  - Columns with frequent updates not good as index keys (e.g. quantity)

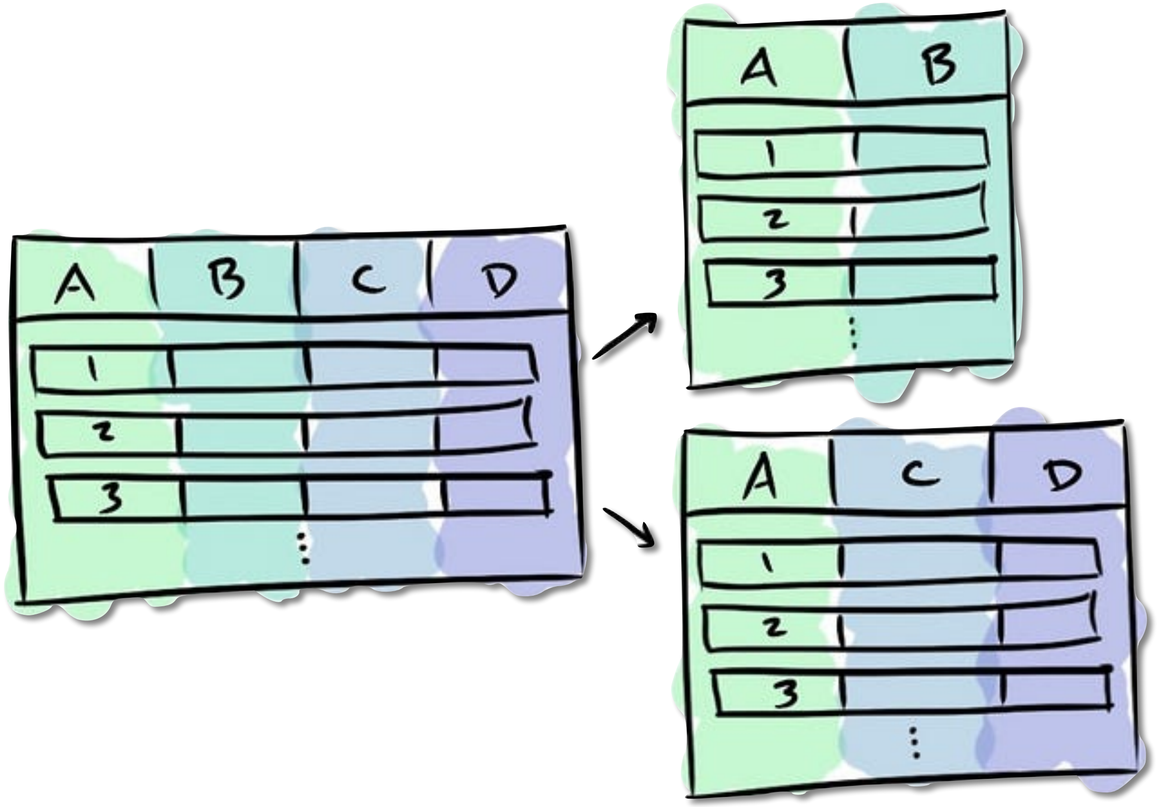
Partitioning



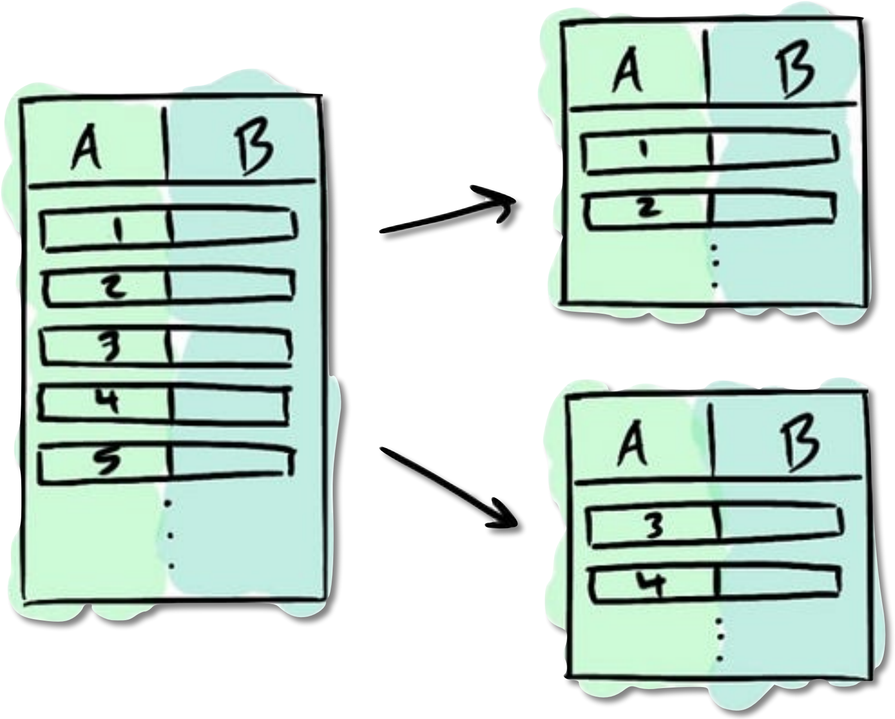
Benefits of Partitioning

- Improved Query Performance
- Manageability
- Increased Parallelism
- Efficient Use of Resources

Partition Types



Vertical Partitioning



Horizontal Partitioning

Vertical Partitioning

prod_id	name	category_id	quantity	notes
7	Prod 7	13	9	Q6MHA1AHfQH5bDK6nyWx7lEtSYxnk
8	Prod 8	13	27	u2mEigECohZbsqD8VwNboB6o5loxg
9	Prod 9	6	5	BTaVF7o19EBvfKxVxuDjVnK5IFjlVH
10	Prod 10	10	15	sSzy7BrPkF15tydp1dbmCPei7CynhH
11	Prod 11	7	5	riOvLKbqxCVliJs1cXNJwDondjuEf3
12	Prod 12	10	25	jQTDEQFCSU68qzvtWFF7D5W9kbwj
13	Prod 13	1	48	ZSqyyXulJE5lQGeld25Zq2R2kgBJih
14	Prod 14	2	24	cBcxwM4mfLclfNvTZgiqGT9aQXJhll
15	Prod 15	13	11	QvuVhElYs6M9l9hxFma9fyfzOTULlQ
16	Prod 16	3	44	4pazBYinEoJDj19h1Kb38GOfO8Z1AU

prod_id	name	category_id	quantity
7	Prod 7	13	9
8	Prod 8	13	27
9	Prod 9	6	5
10	Prod 10	10	15
11	Prod 11	7	5
12	Prod 12	10	25
13	Prod 13	1	48
14	Prod 14	2	24
15	Prod 15	13	11
16	Prod 16	3	44

prod_id	notes
7	Q6MHA1AHfQH5bDK6nyWx7lEtSYxnk
8	u2mEigECohZbsqD8VwNboB6o5loxg
9	BTaVF7o19EBvfKxVxuDjVnK5IFjlVH
10	sSzy7BrPkF15tydp1dbmCPei7CynhH
11	riOvLKbqxCVliJs1cXNJwDondjuEf3
12	jQTDEQFCSU68qzvtWFF7D5W9kbwj
13	ZSqyyXulJE5lQGeld25Zq2R2kgBJih
14	cBcxwM4mfLclfNvTZgiqGT9aQXJhll
15	QvuVhElYs6M9l9hxFma9fyfzOTULlQ
16	4pazBYinEoJDj19h1Kb38GOfO8Z1AU

Vertical Partitioning
DBMS Support

- No special support in most DBMSs
- Need to manually create and handle partitions

```sql
CREATE TABLE products (
    prod_id INT PRIMARY KEY,
    name VARCHAR(20) NOT NULL,
    category_id INT NOT NULL,
    quantity INT NOT NULL,
    notes TEXT
);
```

```sql
CREATE TABLE products_basic (
    prod_id INT PRIMARY KEY,
    name VARCHAR(20) NOT NULL,
    category_id INT NOT NULL,
    quantity INT NOT NULL
);

CREATE TABLE products_notes (
    prod_id INT PRIMARY KEY,
    notes TEXT
);
```

Vertical Partitioning

DBMS Support

Products_basic

prod_id	name	category_id	quantity
7	Prod 7	13	9
8	Prod 8	13	27
9	Prod 9	6	5
10	Prod 10	10	15
11	Prod 11	7	5
12	Prod 12	10	25
13	Prod 13	1	48
14	Prod 14	2	24
15	Prod 15	13	11
16	Prod 16	3	44

Products_notes

prod_id	notes
7	Q6MHA1AHfQH5bDK6nyWx7IEtSYxnI
8	u2mEigECohZbsqD8VwNboB6o5loxg
9	BTaVF7o19EBvfKxVxuDjVnK5IFjlVH
10	sSzy7BrPkF15tydp1dbmCPei7CynhH
11	riOvLKbqxCVliJs1cXNJwDondjuEf3
12	jQTDEQFCSU68qzvtWFF7D5W9kbwj
13	ZSqyyXulJE5lQGeId25Zq2R2kgBJih
14	cBcxwM4mfLclfNvTZgiqGT9aQXJhll
15	QvuVhElYs6M9l9hxFma9fyfzOTULlQ
16	4pazBYinEoJDj19h1Kb38GOfO8Z1AU

Partial row

Partial row

Horizontal Partitioning

order_id	order_date	customer_id	amount
1573	2022-05-28	16	648
1574	2023-01-26	10	921
1575	2023-09-25	3	810
1576	2022-01-25	18	1063
1577	2022-06-28	8	753
1578	2021-01-28	13	548
1579	2021-02-07	2	572
1580	2023-03-23	11	953
1581	2023-09-11	16	734
1582	2022-09-12	19	1070
1583	2023-04-07	12	385
1584	2023-07-31	16	930
1585	2023-09-22	9	716
1586	2023-07-17	15	766
1587	2022-12-20	8	1000
1588	2022-11-28	2	595
1589	2022-06-30	20	949
1590	2023-04-05	11	297
1591	2022-07-01	2	640
1592	2023-05-18	2	596
1593	2023-03-10	8	280
1594	2022-06-16	8	971
1595	2022-07-07	18	796
1596	2021-06-05	17	796

order_id	order_date	customer_id	amount
177	2021-01-11	9	581
181	2021-03-15	14	921
191	2021-0		
207	2021-0		
210	2021-0		
260	2021-0		
273	2021-0		
289	2021-0		

order_id	order_date	customer_id	amount
150	2021-06-24	19	1302
159	2021-05-10	7	192
162	2021-0		
175	2021-0		
183	2021-0		
187	2021-0		
198	2021-0		
212	2021-0		

order_id	order_date	customer_id	amount
229	2021-08-04	14	1093
231	2021-08-15	7	167
251	2021-09-12	19	145
269	2021-08-26	19	458
317	2021-09-10	11	413
339	2021-08-23	19	303
340	2021-07-17	5	706
351	2021-08-17	14	144

Range Partitioning

Orders

order_id	order_date	customer_id	amount
1573	2022-05-28	16	648
1574	2023-01-26	10	921
1575	2023-09-25	3	810
1576	2022-01-25	18	1063
1577	2022-06-28	8	753
1578	2021-01-28	13	548
1579	2021-02-07	2	572
1580	2023-03-23	11	953
1581	2023-09-11	16	734
1582	2022-09-12	19	1070
1583	2023-04-07	12	385
1584	2023-07-31	16	930
1585	2023-09-22	9	716
1586	2023-07-17	15	766
1587	2022-12-20	8	1000
1588	2022-11-28	2	595
1589	2022-06-30	20	949
1590	2023-04-05	11	297
1591	2022-07-01	2	640
1592	2023-05-18	2	596
1593	2023-03-10	8	280
1594	2022-06-16	8	971
1595	2022-07-07	18	796

From 2021-01-01 to 2021-03-31

From 2021-04-01 to 2021-06-30

From 2021-07-01 to 2021-09-30

...

Orders_0

order_id	order_date	customer_id	amount
177	2021-01-11	9	581
181	2021-03-15	14	931
191	2021-03-11	13	445
207	2021-01-06	20	553
210	2021-01-13	17	471
260	2021-02-20	6	590
273	2021-03-20	5	583
289	2021-02-20	14	144

Orders_1

order_id	order_date	customer_id	amount
150	2021-06-24	19	1302
159	2021-05-10	7	192
162	2021-04-10	6	767
175	2021-06-21	8	764
183	2021-05-11	12	1079
187	2021-05-20	13	444

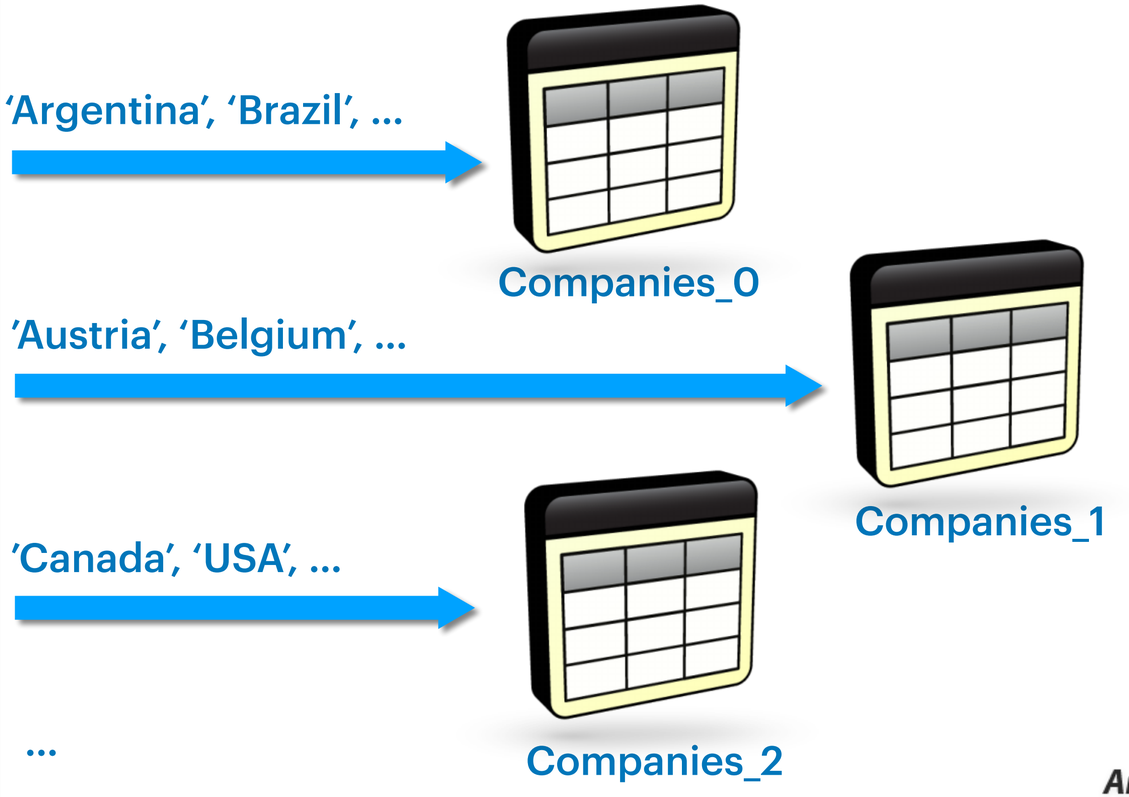
Orders_2

order_id	order_date	customer_id	amount
229	2021-08-04	14	1093
231	2021-08-15	7	167
251	2021-09-12	19	145
269	2021-08-26	19	458
317	2021-09-10	11	413
339	2021-08-23	19	303
340	2021-07-17	5	706
251	2021-08-17	14	144

List Partitioning

Companies

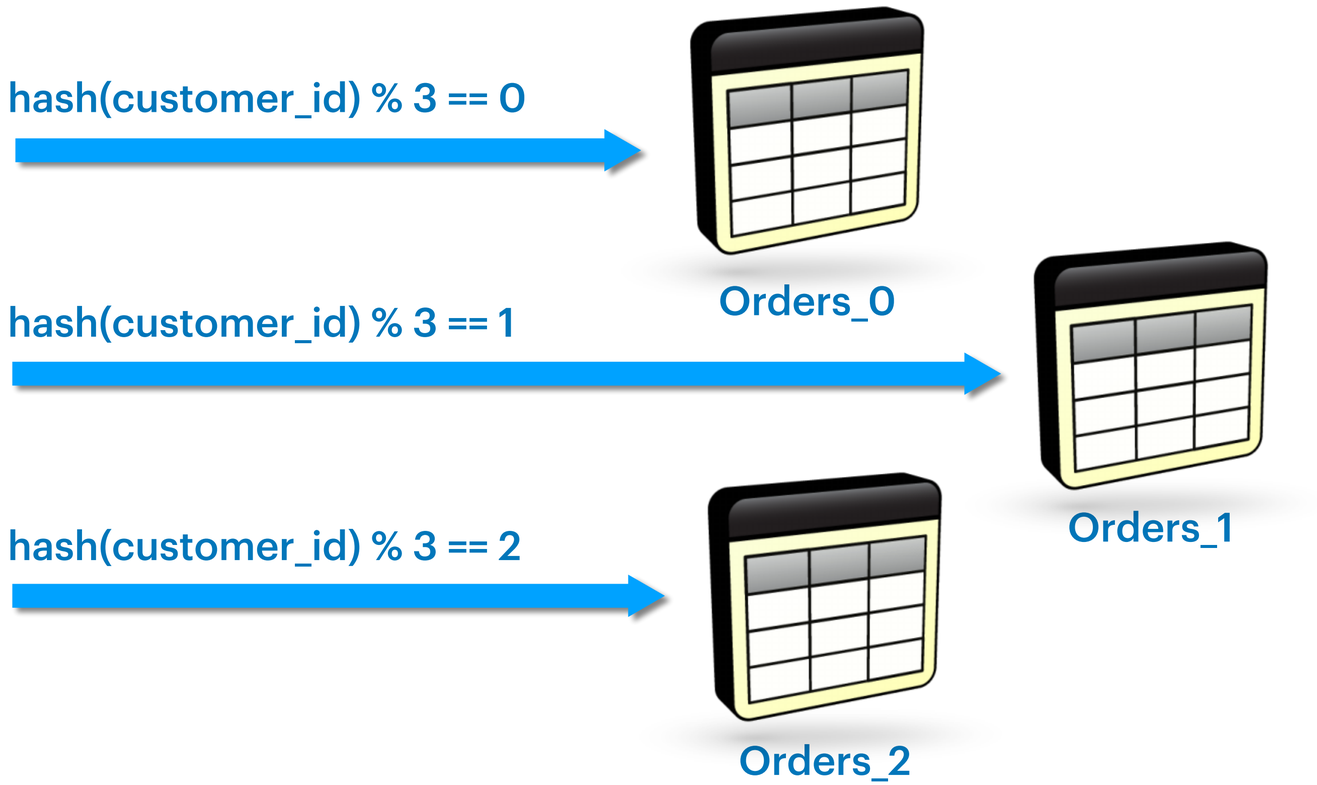
	company_name	country	region
1	Cactus Comidas para llevar	Argentina	[NULL]
2	Océano Atlántico Ltda.	Argentina	[NULL]
3	Rancho grande	Argentina	[NULL]
4	Ernst Handel	Austria	[NULL]
5	Piccolo und mehr	Austria	[NULL]
6	Maison Dewey	Belgium	[NULL]
7	Suprêmes délices	Belgium	[NULL]
8	Comércio Mineiro	Brazil	SP
9	Familia Arquibaldo	Brazil	SP
10	Gourmet Lanchonetes	Brazil	SP
11	Hanari Carnes	Brazil	RJ
12	Que Delícia	Brazil	RJ
13	Queen Cozinha	Brazil	SP
14	Ricardo Adocicados	Brazil	RJ
15	Tradição Hipermercados	Brazil	SP
16	Wellington Importadora	Brazil	SP
17	Bottom-Dollar Markets	Canada	BC
18	Laughing Bacchus Wine Cellars	Canada	BC



Hash Partitioning

Orders

order_id	order_date	customer_id	amount
1573	2022-05-28	16	648
1574	2023-01-26	10	921
1575	2023-09-25	3	810
1576	2022-01-25	18	1063
1577	2022-06-28	8	753
1578	2021-01-28	13	548
1579	2021-02-07	2	572
1580	2023-03-23	11	953
1581	2023-09-11	16	734
1582	2022-09-12	19	1070
1583	2023-04-07	12	385
1584	2023-07-31	16	930
1585	2023-09-22	9	716
1586	2023-07-17	15	766
1587	2022-12-20	8	1000
1588	2022-11-28	2	595
1589	2022-06-30	20	949
1590	2023-04-05	11	297
1591	2022-07-01	2	640
1592	2023-05-18	2	596
1593	2023-03-10	8	280
1594	2022-06-16	8	971
1595	2022-07-07	18	796
1596	2021-06-05	17	796



Horizontal Partitioning

Uses

- **Range**
  - Data that naturally falls into ranges, e.g. dates, numerical ranges
- **List**
  - Data has a finite set of distinct values, e.g., categories, regions
- **Hash**
  - Evenly distributes data when there’s no natural range or list

Partitioning vs. Sharding