
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

DAT306

Migrate from a relational database to Amazon DynamoDB

Robert McCauley

DynamoDB Solutions Architect
AWS

Jason Hunter

DynamoDB Solutions Architect
AWS

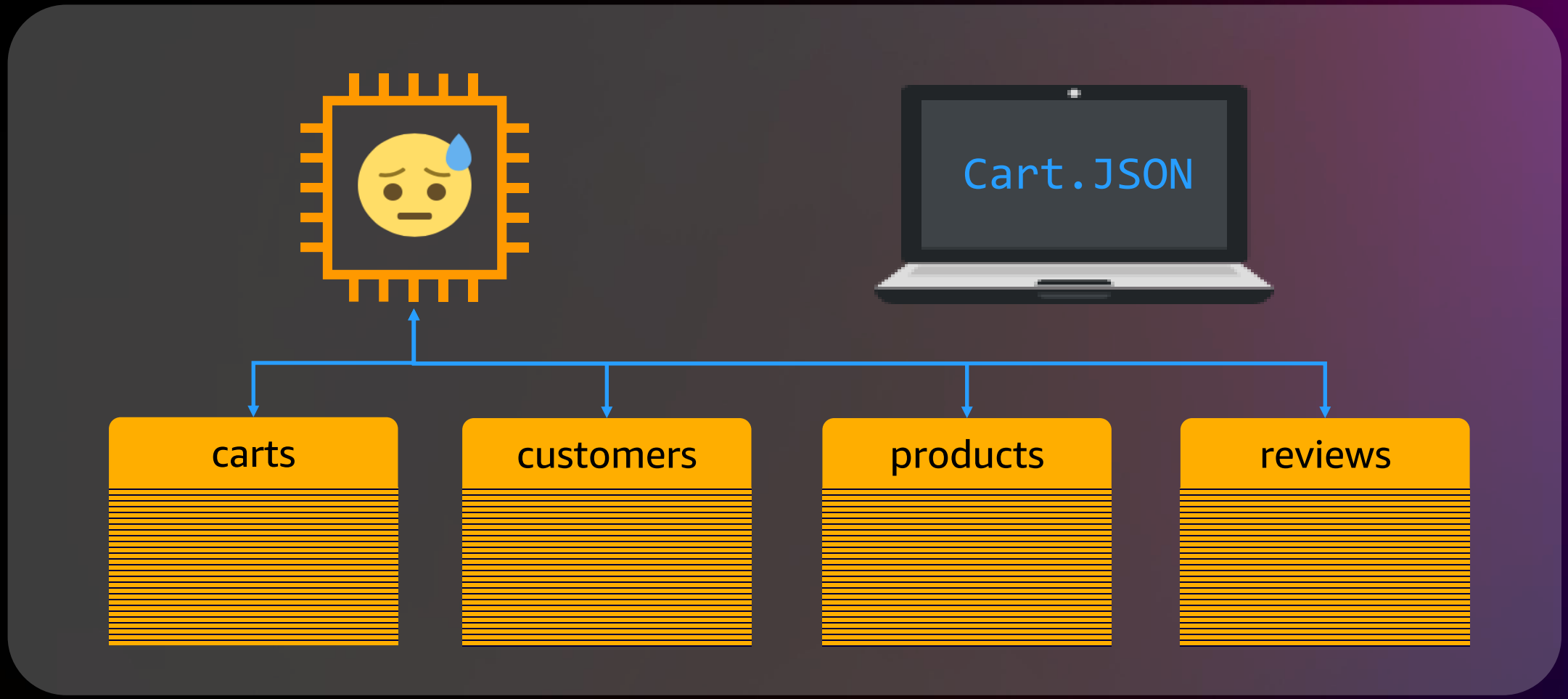
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Relational DB challenges

- Hard to scale
- Accepts unbounded work
- Monolith

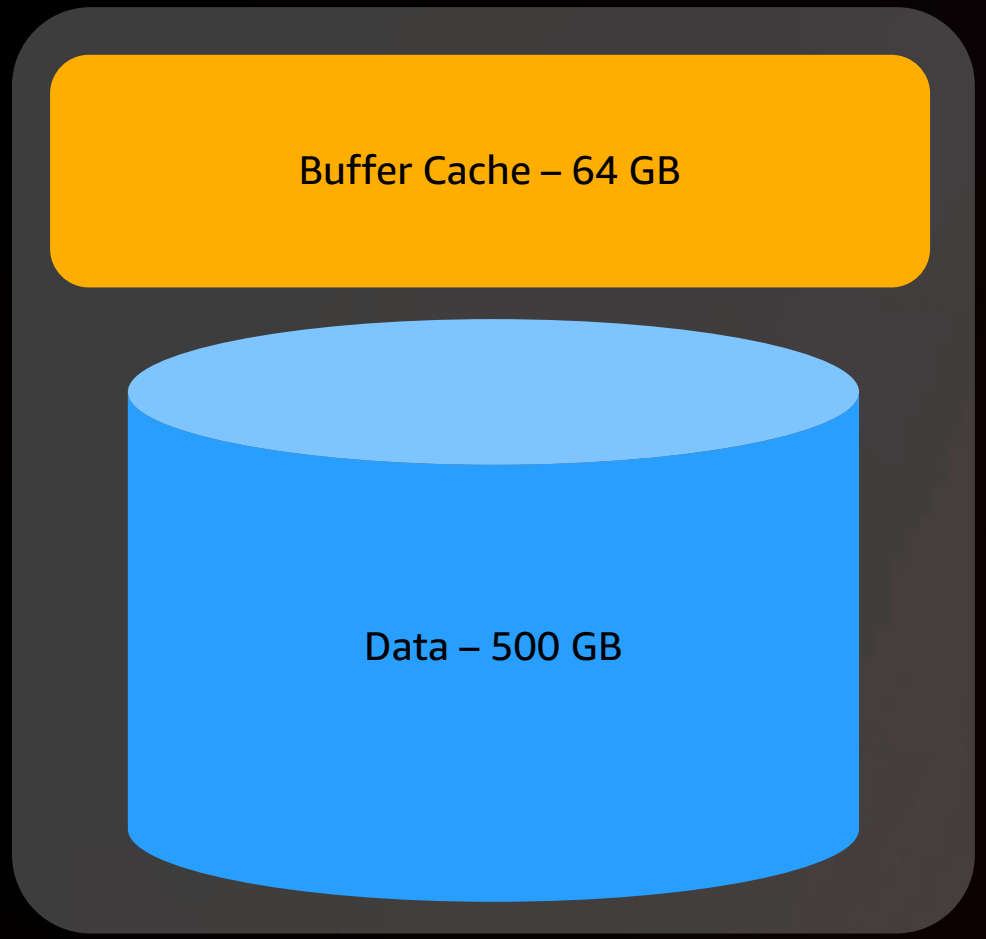
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

At large scale, JOINs are taxing



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unpredictable performance



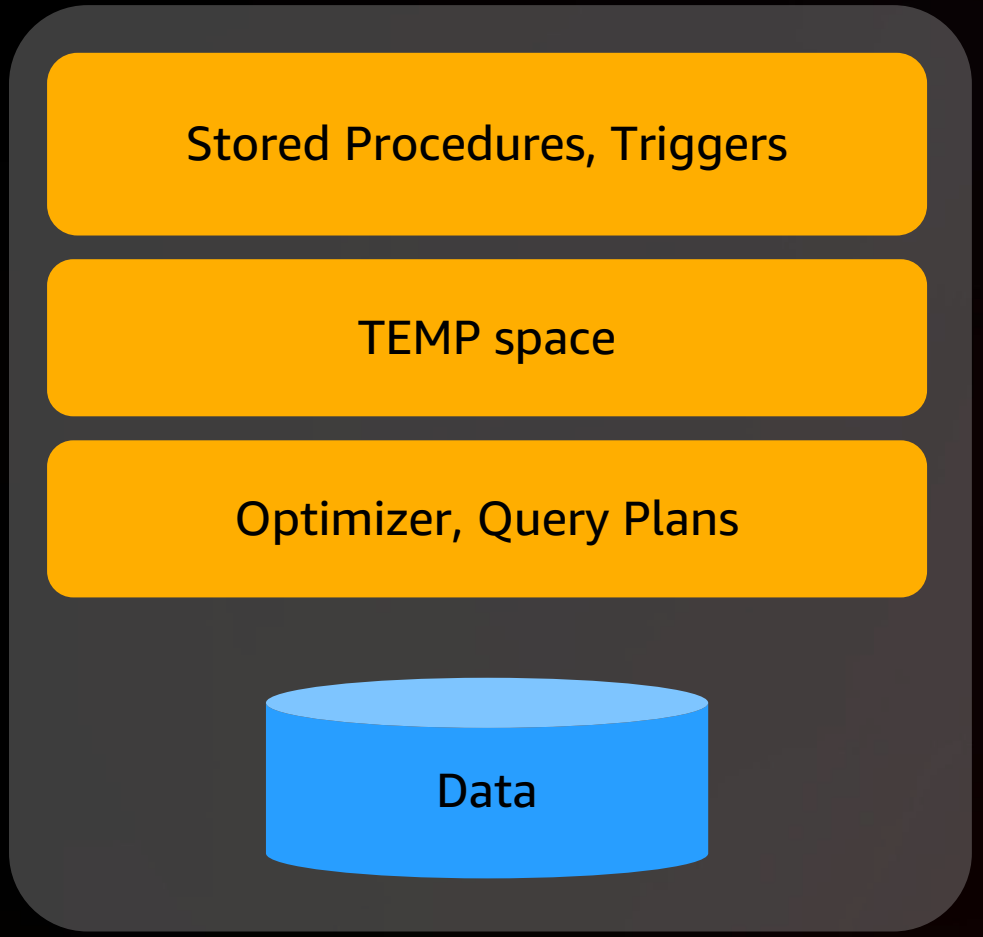

```
SELECT *
FROM Table
WHERE Column15 = "xyz"
```

Year 1: **here you go!**

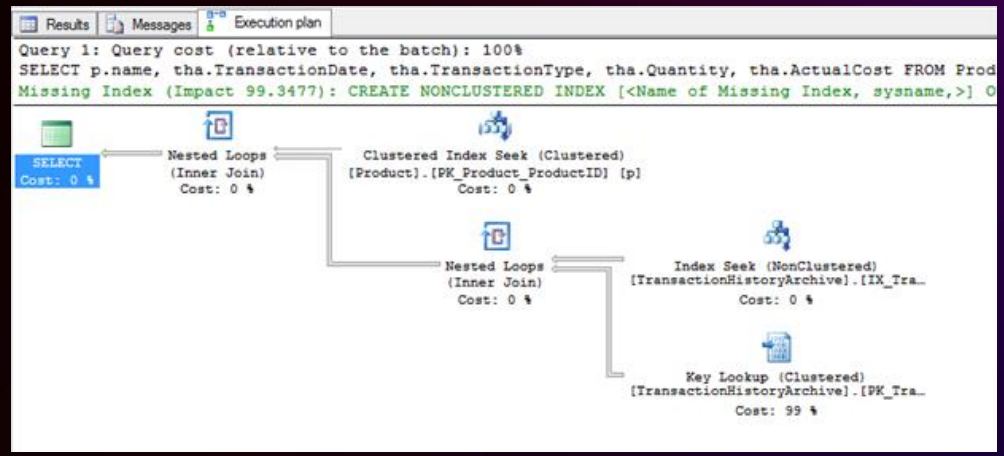
Year 2: **please wait.....**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Application code in database

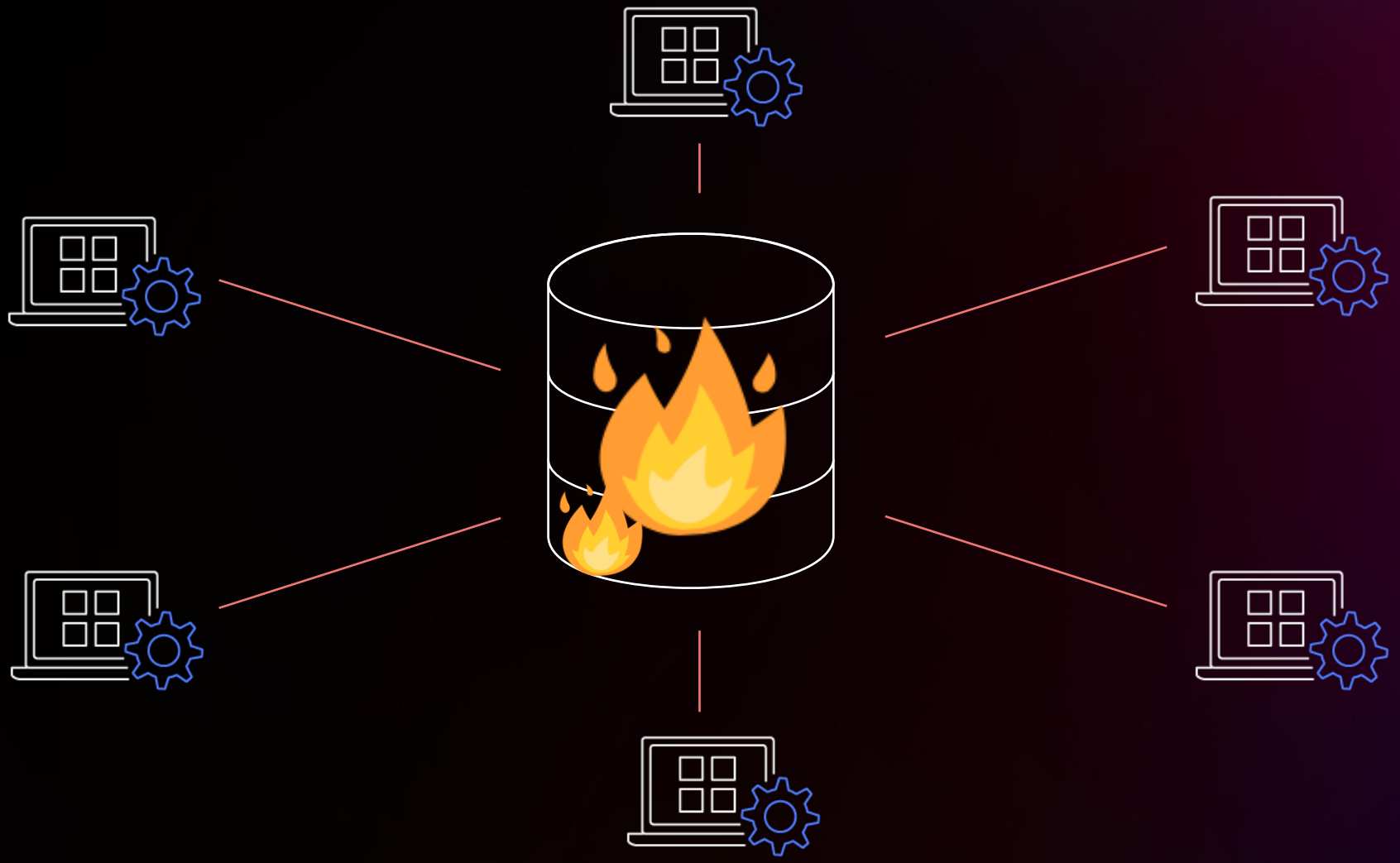



```
IF Balance > :x
THEN
    UPDATE.. WHERE..
ELSE
    IF Rating < :y
    THEN
      INSERT INTO..
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Monolith issue – Shared everything



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

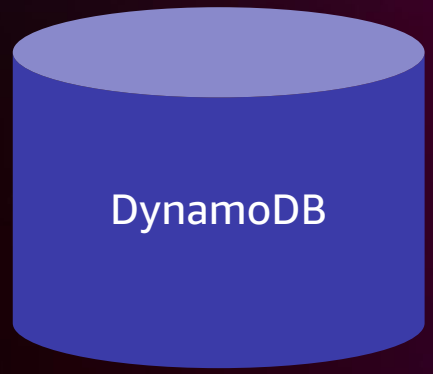
Should you migrate your relational application?

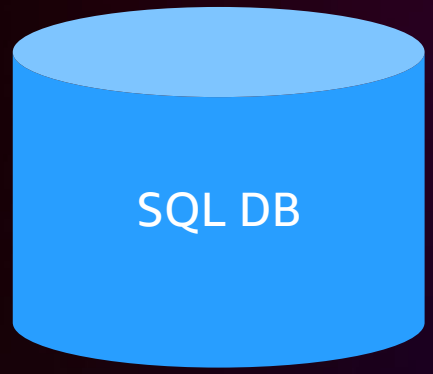
Best for Amazon DynamoDB

- Key-value lookups
- 1:many queries
- Serverless applications, Internet-scale applications
- JSON documents

Best for relational SQL

- ERP applications
- Data warehouses, data marts



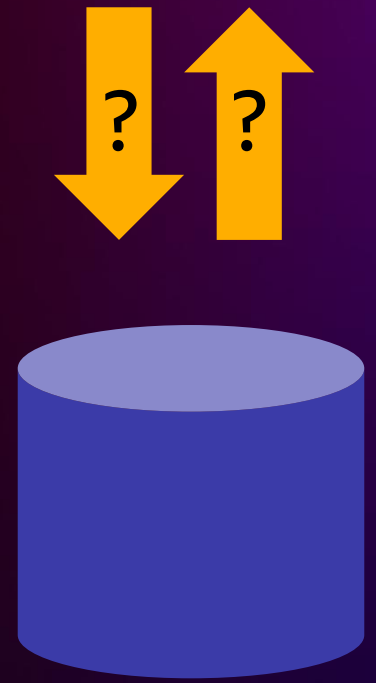


aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1: Document needed access patterns

Name	Input args	Single or set	Returned columns	Frequency per hour
Put order	orderID, custID	single	-	100
Update order	orderID	single	-	5000
Get customer orders	custID, Date range	set	orderID, orderLine, productID	200



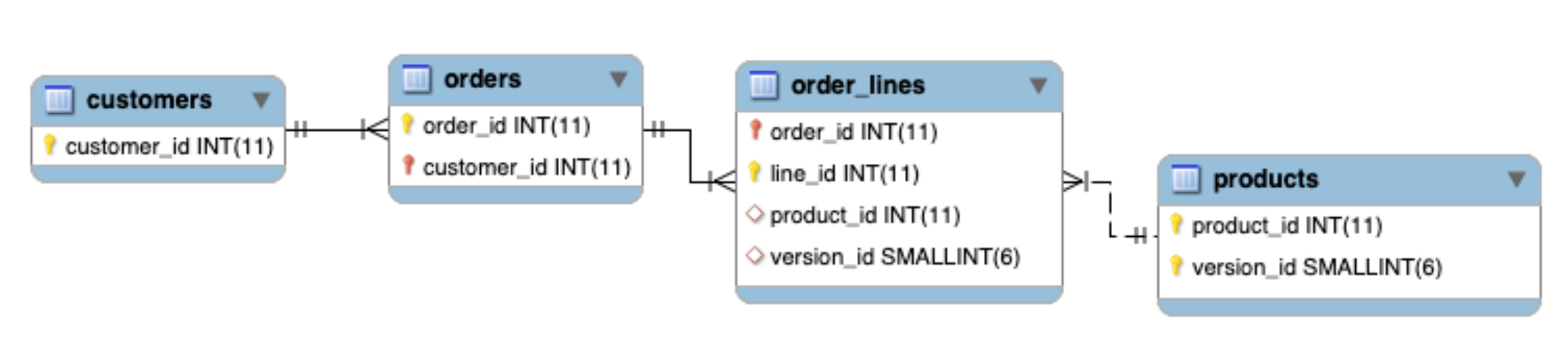
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Relational to Single Table: Data Shapes



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

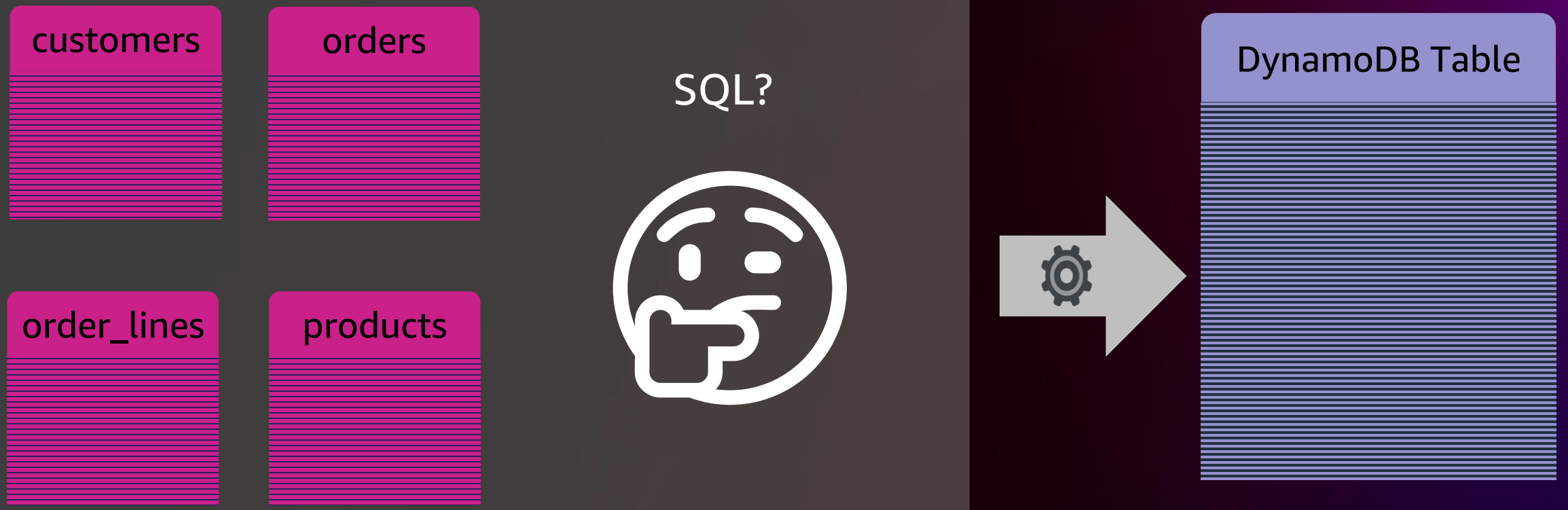
Understanding table data



Item Size	medium	small	small	large
Item Count	hundreds	millions	billions	dozen
Velocity	cool	warm	hot	cold
Type	Dimension	Fact Header	Fact	Dimension

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deciding how to combine many entities into one set with SQL



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DENORMALIZED with JOIN

Get-Item (cust-2, ord-2-line-3)

Orders

A single, complete record is returned

cust-2	ord-2-line-3	2022-12-30	pr-300

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

STACKED with UNION

Query (cust-2)

616-838-3000	cust-2

Returns an Item Collection

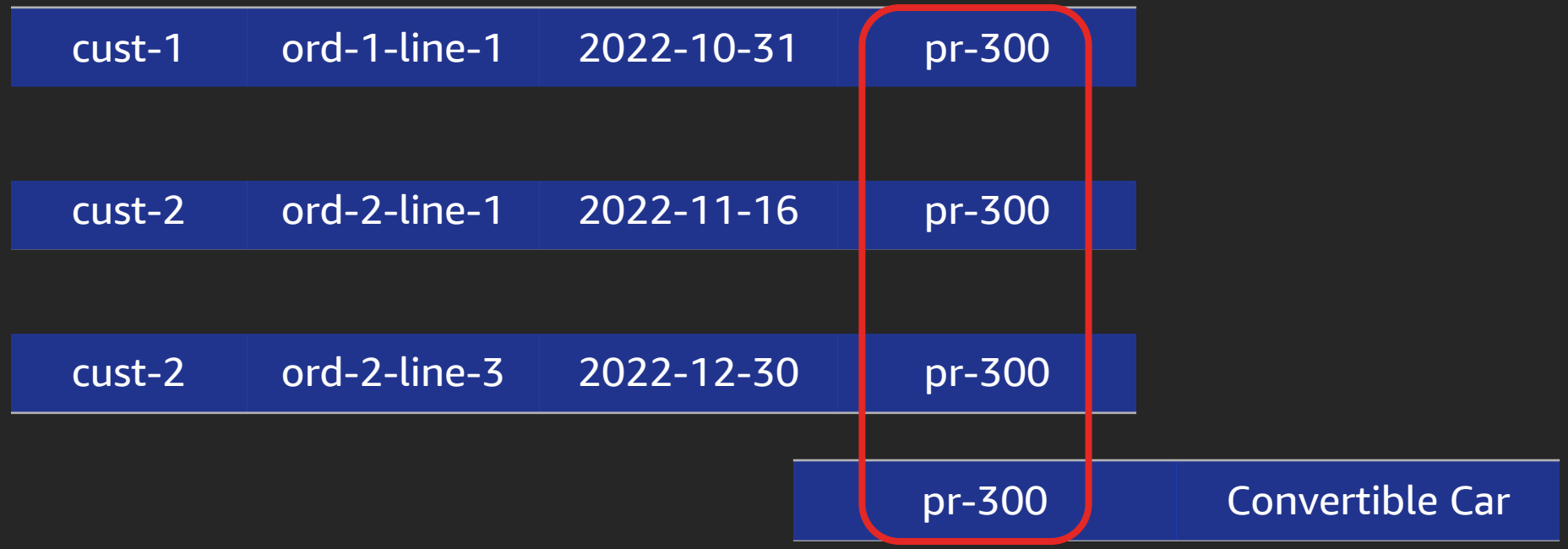
cust-2	ord-2-line-1	2022-11-16	pr-300
cust-2	ord-2-line-2	2022-12-26	pr-400
cust-2	ord-2-line-3	2022-12-30	pr-300

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

STACKED with UNION

Query GSI (pr-300)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Entity combinations: INNER JOIN vs. UNION ALL

Entity combining strategy:	Data shape	Typical DynamoDB usage pattern
JOIN (Denormalized)	• Wide dataset with duplicated column values	**Get-Item** • Single read • Lowest latency
UNION (Stacked)	• Different Entities share the table • Storage efficient	**Query** • Item Collection returned • May need multiple reads

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Relational tables stacked vertically with UNION ALL

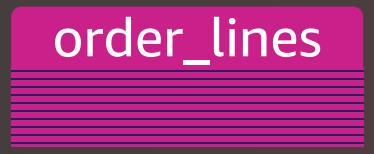
SELECT cust_id, cust_name, NULL as order_id, NULL as line_id, NULL as prd_id FROM customers



UNION ALL SELECT … FROM orders



UNION ALL SELECT … FROM order_lines



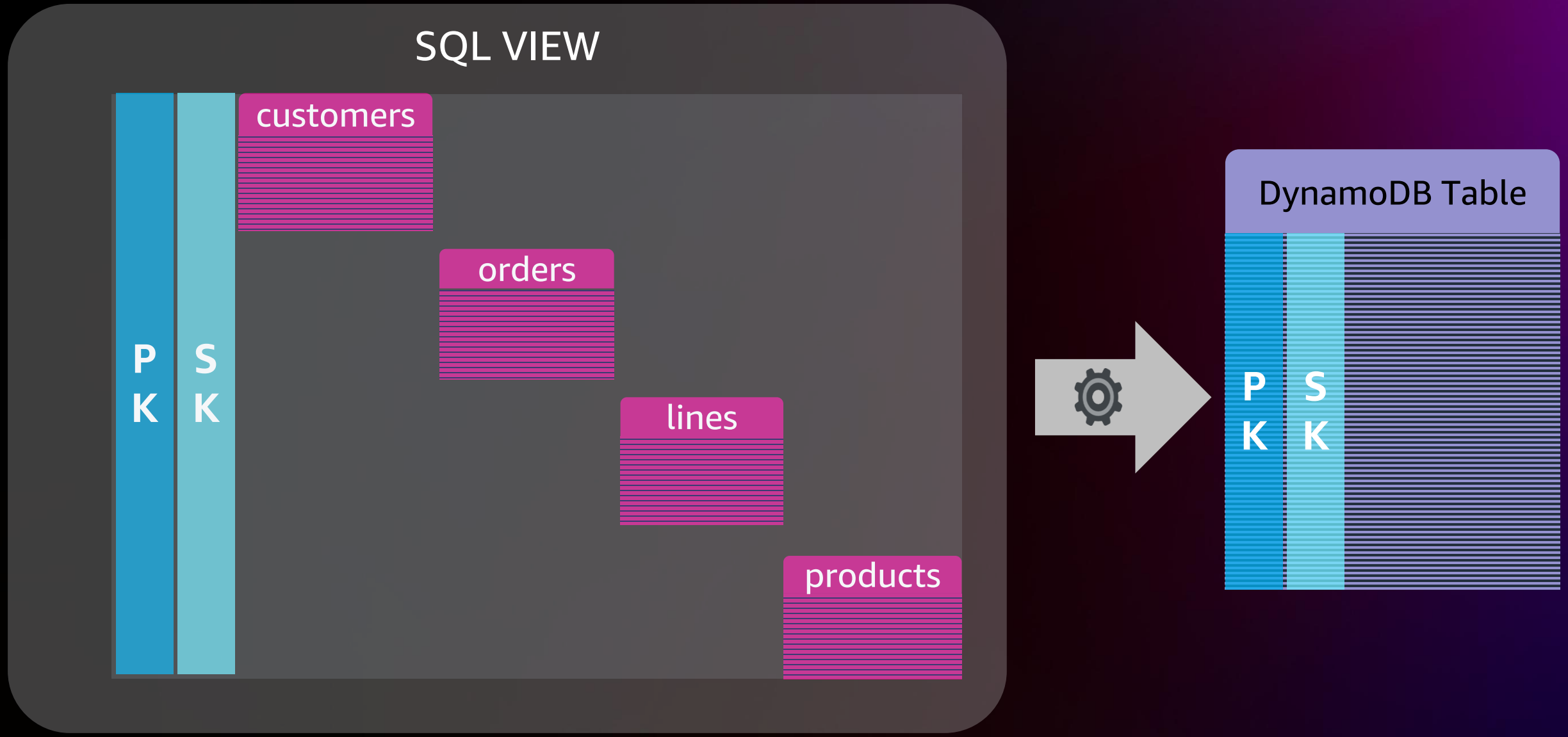
UNION ALL SELECT … FROM products



aws

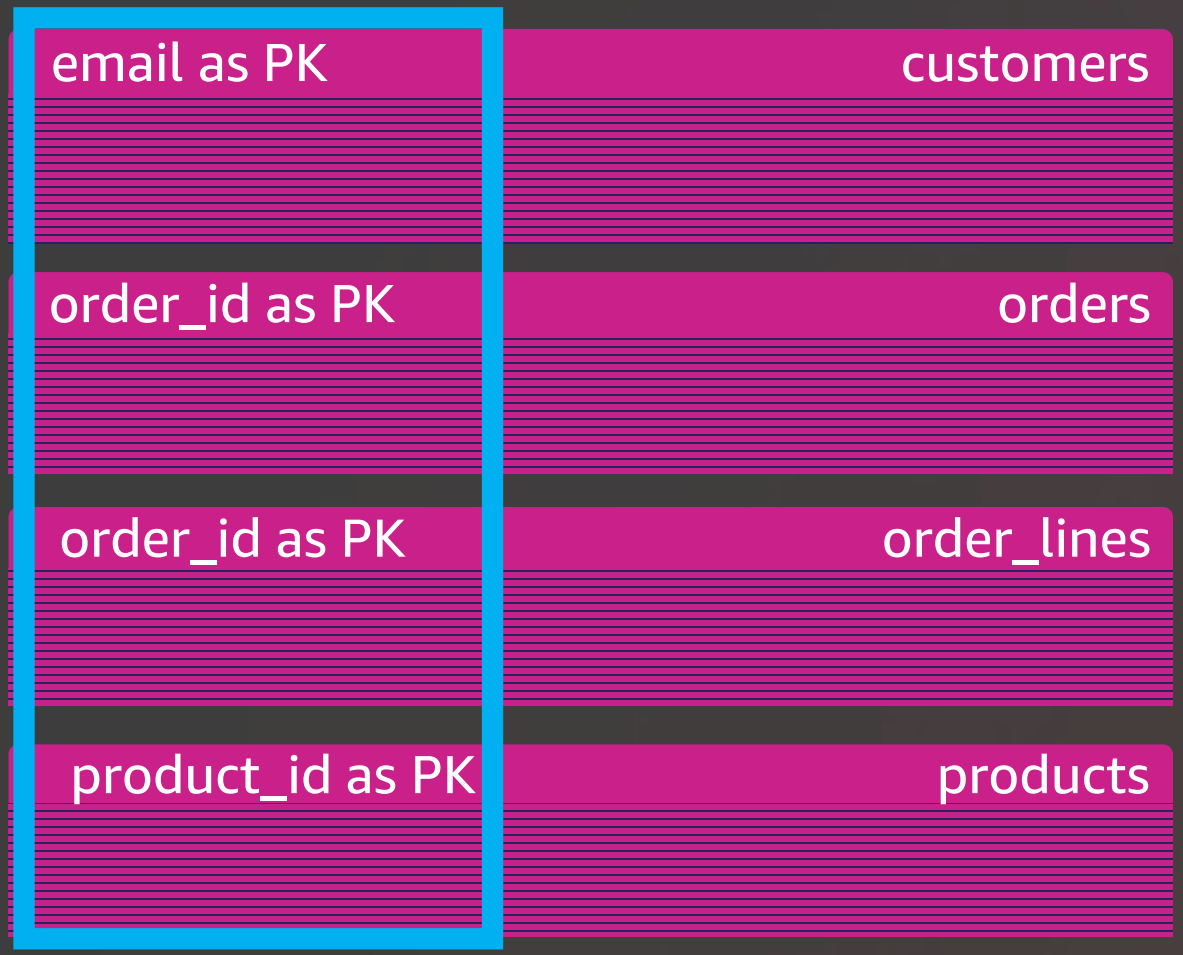
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Add col1 AS PK, col2 AS SK; Unique primary key



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Choose partition key

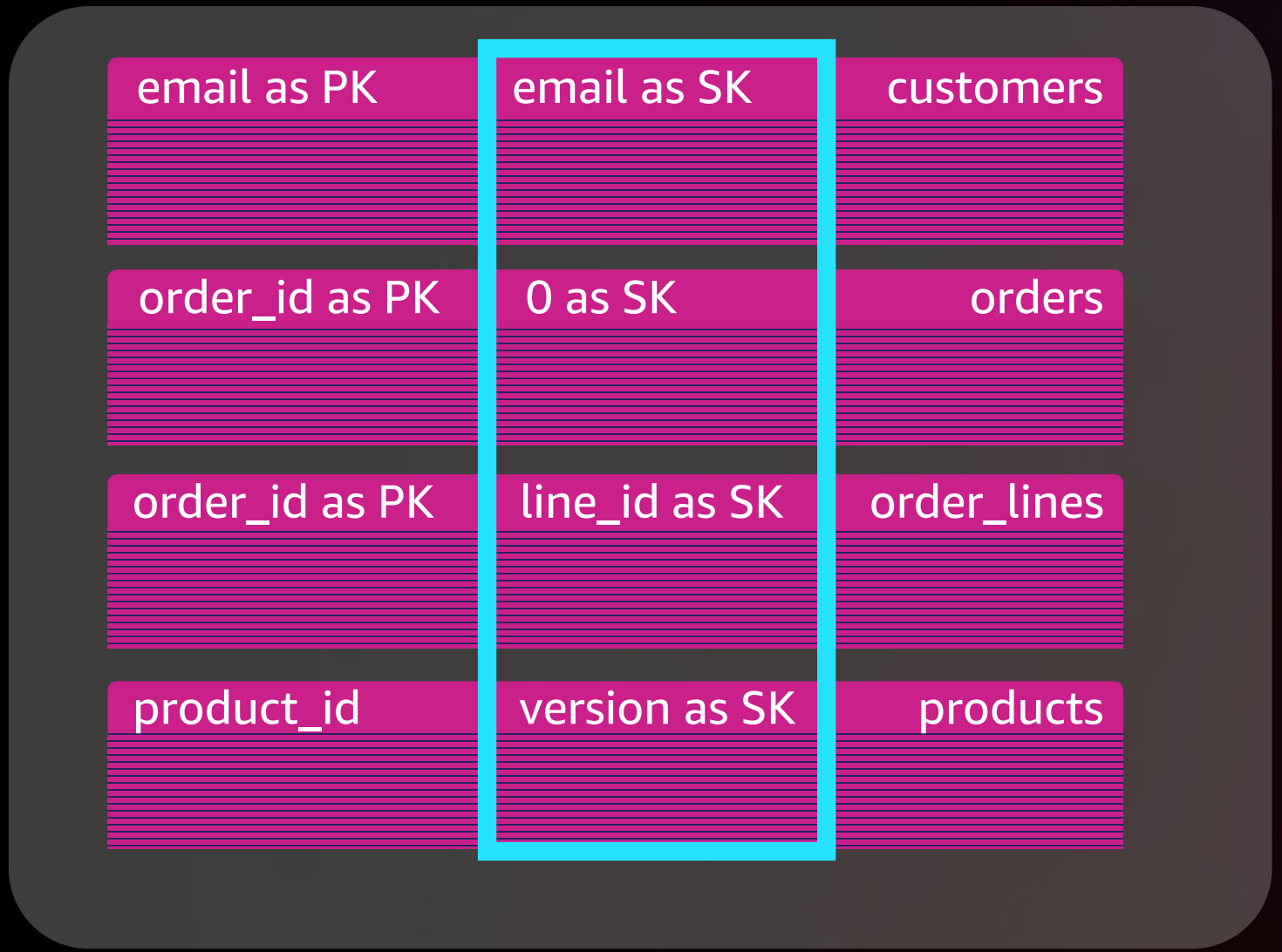


Possible PK choices:

- Highly cardinal ID
- Table’s primary key
- Parent ID
- Collection ID
- UUID/GUID

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Choose optional sort key



Possible SK choices:

- Same as PK
- Static 0
- Line Item ID
- ISO or epoch date
- Date.now()
- UUID/GUID

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Decorate key and item for readability

```
SELECT
    "pr-" + product_id     as PK,

    "ver-" + version_id    as SK,

    "product"              as Type,
    …
    …
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Duplicate data for queries, future use

```
SELECT
    "pr-" + product_id      as PK,
    "pr-" + product_id      as Product,

    event_date              as eventdate,
    YEAR(event_date)        as eventyear,
    QUARTER(event_date)     as eventqtr
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Date formats

- ISO 8601 date
  - 2022-05-17
  - 2022-05-17 06:45:11
  - 2022-05-17 06:45:11 d7fp4
- TTL
  - 1681876287
  - unix_timestamp(date_add(now(),interval 365 day))

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Concatenated formats

- Concatenated columns for index sort key

  status#date

  CONCAT(status, "#", date)

  state#city#office

  CONCAT(state, "#", city, "#", office)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

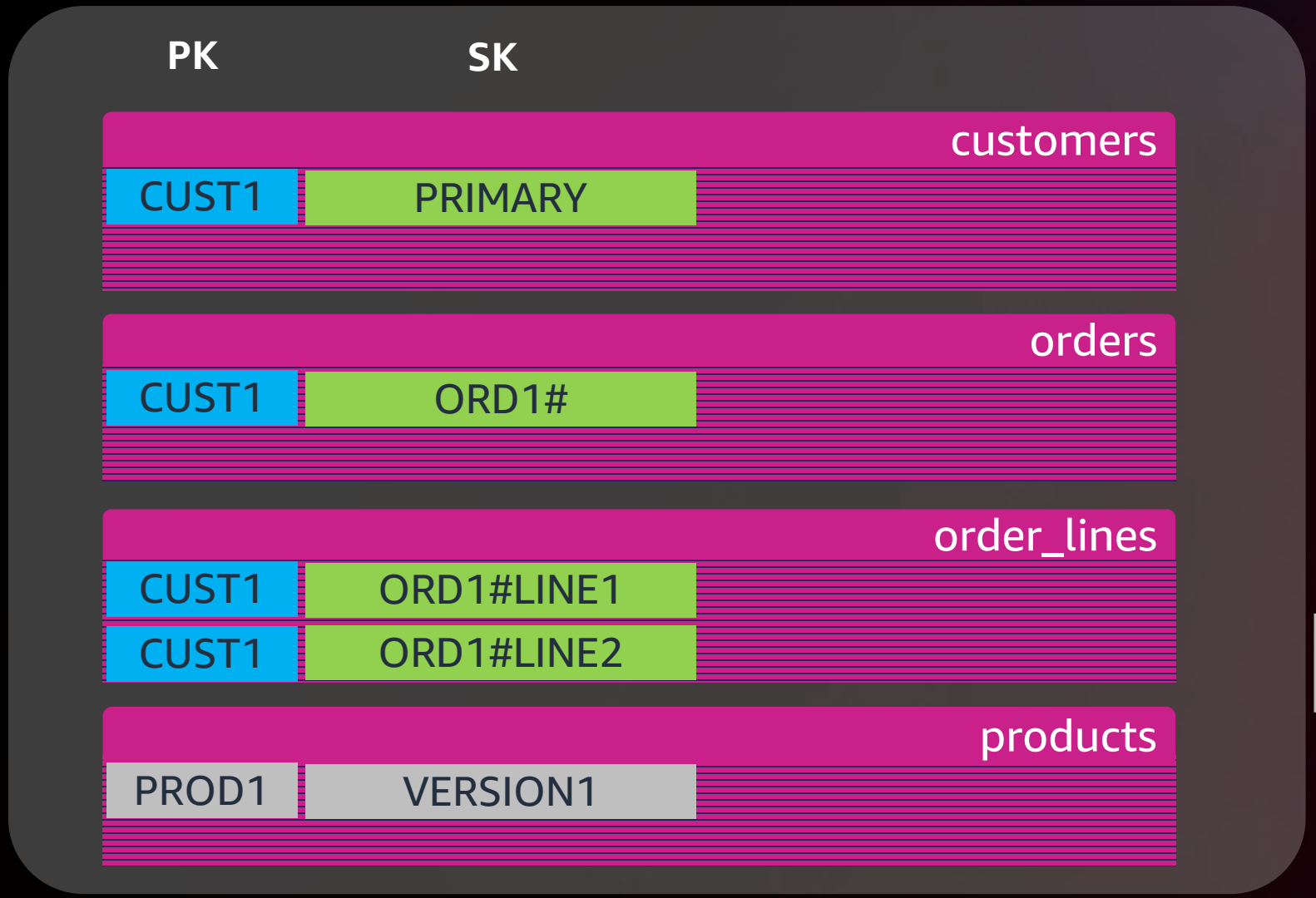
Sparse columns for sparse GSIs

```
SELECT
  CASE
       WHEN product = 'car' AND price > 70000
       THEN price
       ELSE NULL END
       AS Luxury

  CASE
       WHEN day IN ['SATURDAY','SUNDAY']
       THEN "Weekend" ELSE NULL END
       AS WeekendOnly
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

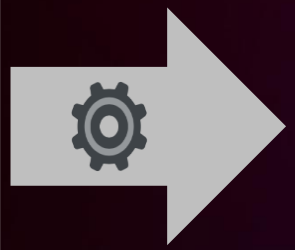
Align PK attributes for Item Collections

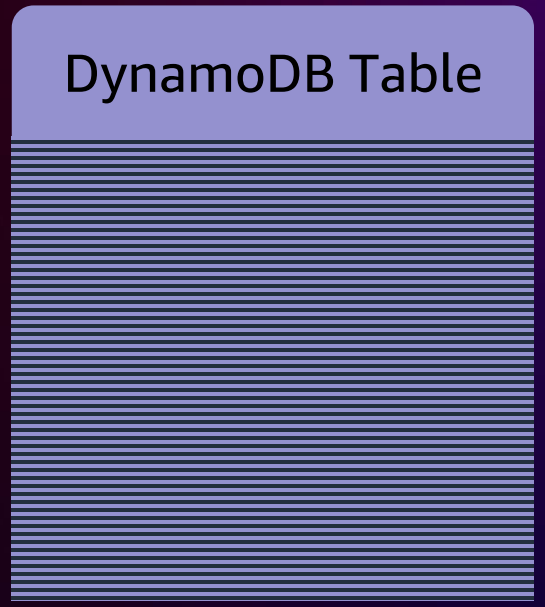


DynamoDB Query

PK = CUST1

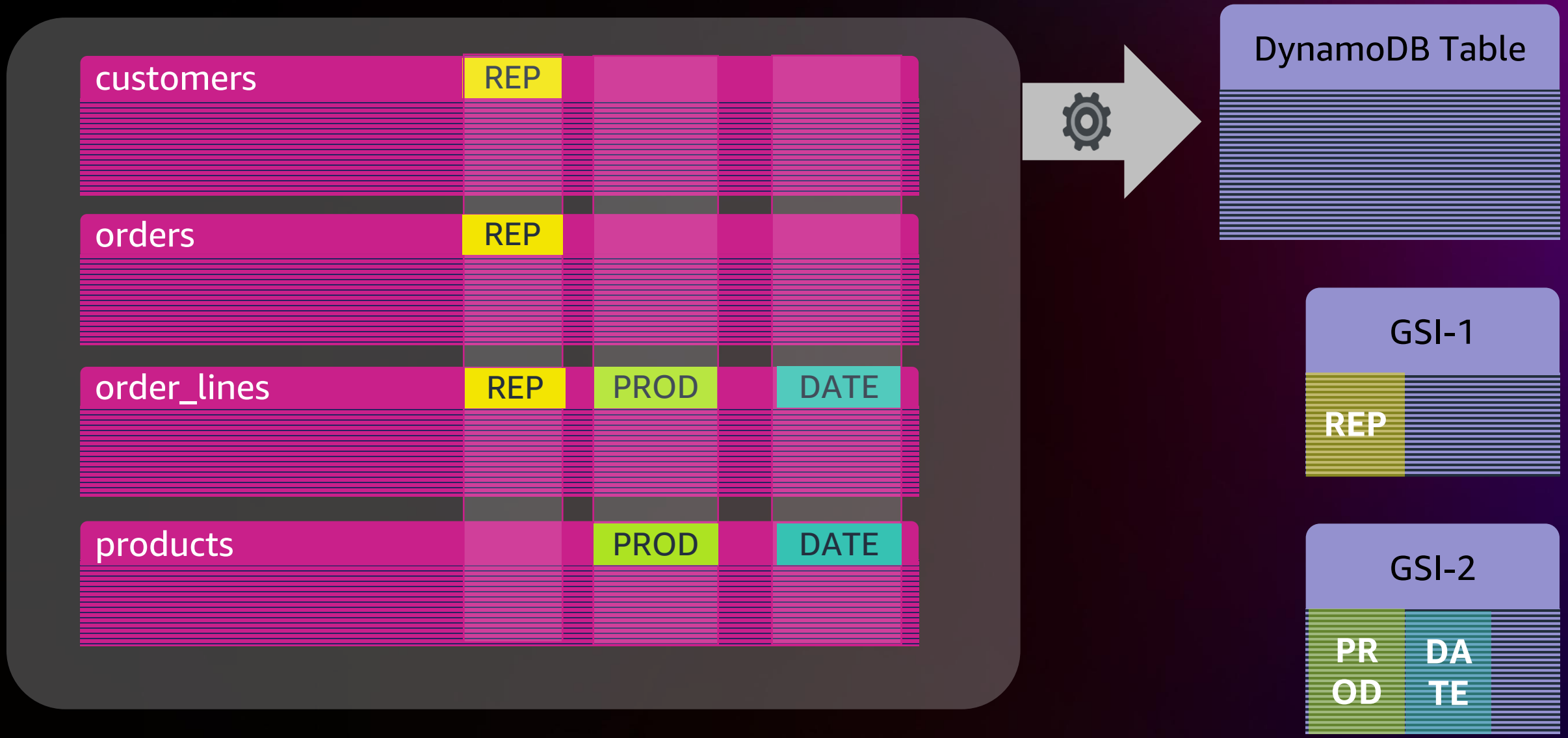
SK = BEGINS_WITH(ORD1#)





aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Align other attributes for Global Secondary Indexes



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration strategies and tools

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) easily and securely migrates and / or replicates your relational databases to AWS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS DMS limitations

- Transfers data from one table to one table, does not combine tables for NoSQL table format
- DMS can be used after you prepare your data with a SQL VIEW to look like your new DynamoDB table
- AWS DMS can run single threaded or multi threaded
- Each thread writes to DynamoDB at 200 WCU (12 MB per minute)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migration strategies

- USE DMS, AWS Glue or your favorite Extract, Transform, Load tool
- Write and run custom scripts to gain experience with bulk loads
- Archive legacy data and migrate only dimension data, starting fresh
- Write to both databases for some time to verify DynamoDB

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Avoiding throttles during data load

- Fixed size database servers slow down during heavy load
- DynamoDB sheds traffic for two reasons:
  - Throughput beyond provisioned limits
  - Hot key limits
    - > 1000 WCU/sec for any one PK

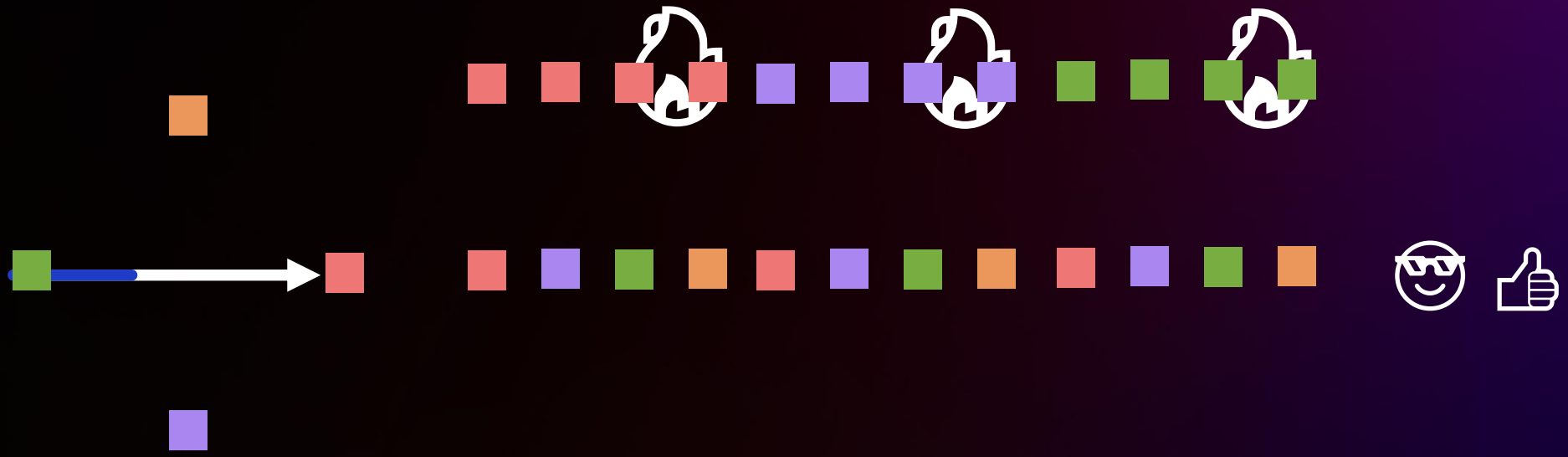


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data loading strategy

- When loading, avoid writing to any one Partition Key faster than 1,000/second, regardless of table capacity
- Use a Round Robin shuffle to fan out heavy writes



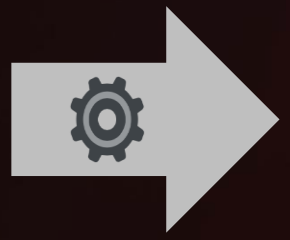


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Staging data to Amazon S3 then loading to DynamoDB



S3



Custom Code

- Lambda
- Container
- EC2



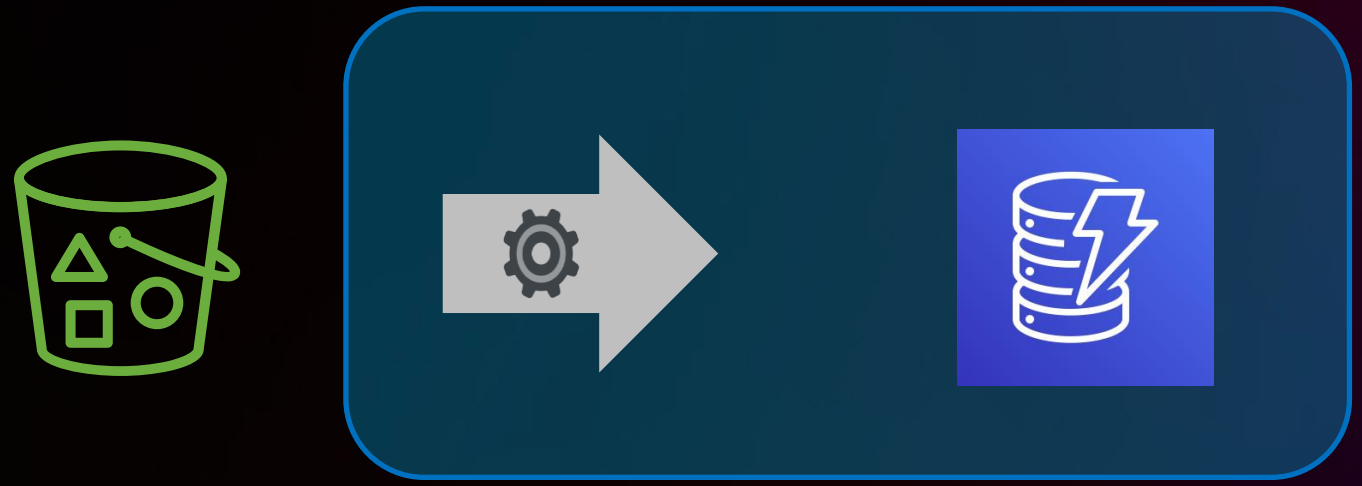
- Capacity Changes
- Risk of over provisioning
- Risk of under provisioning
- Throttle detection
- Error retries
- Pay for WCUs consumed

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New: DynamoDB import from S3 feature

- Fully managed within DynamoDB service
- No code, no Lambda, no ETL required
- No WCUs consumed or provisioned
- Imports CSV, DynamoDB JSON, Amazon Ion data



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Rob McCauley
@robmccauley

Jason Hunter
@hunterhacker



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.