데이타솔루션 Microsoft

Azure Synapse Analytics(SQL DW)

2020년 09월

Overview

Fast
Unlimited Scale

MPP to quickly run complex queries on any scale

Flexible
Fit your Needs

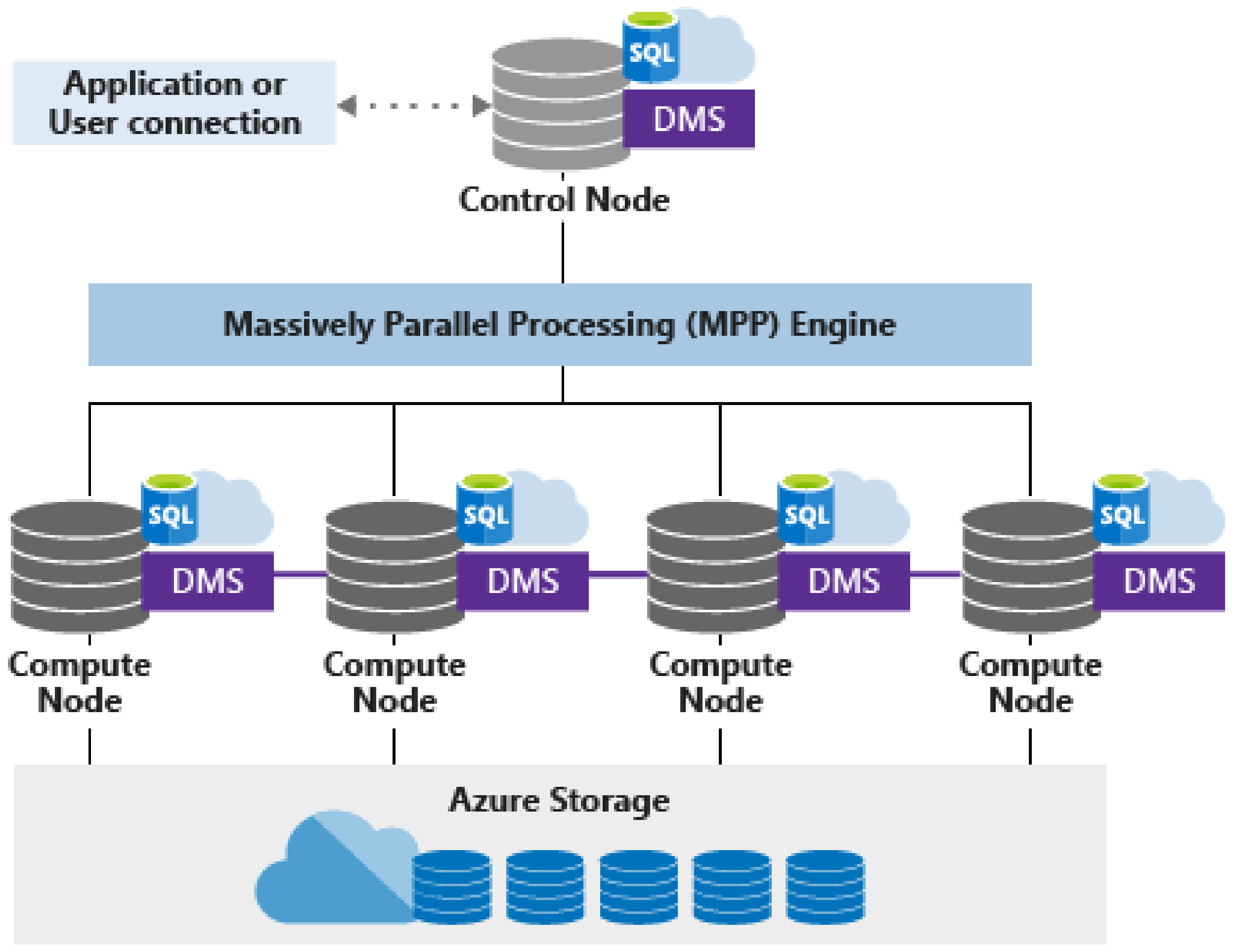
Optimize cost by scaling compute and storage independently

Trusted
Secure. Compliant. Reliable.

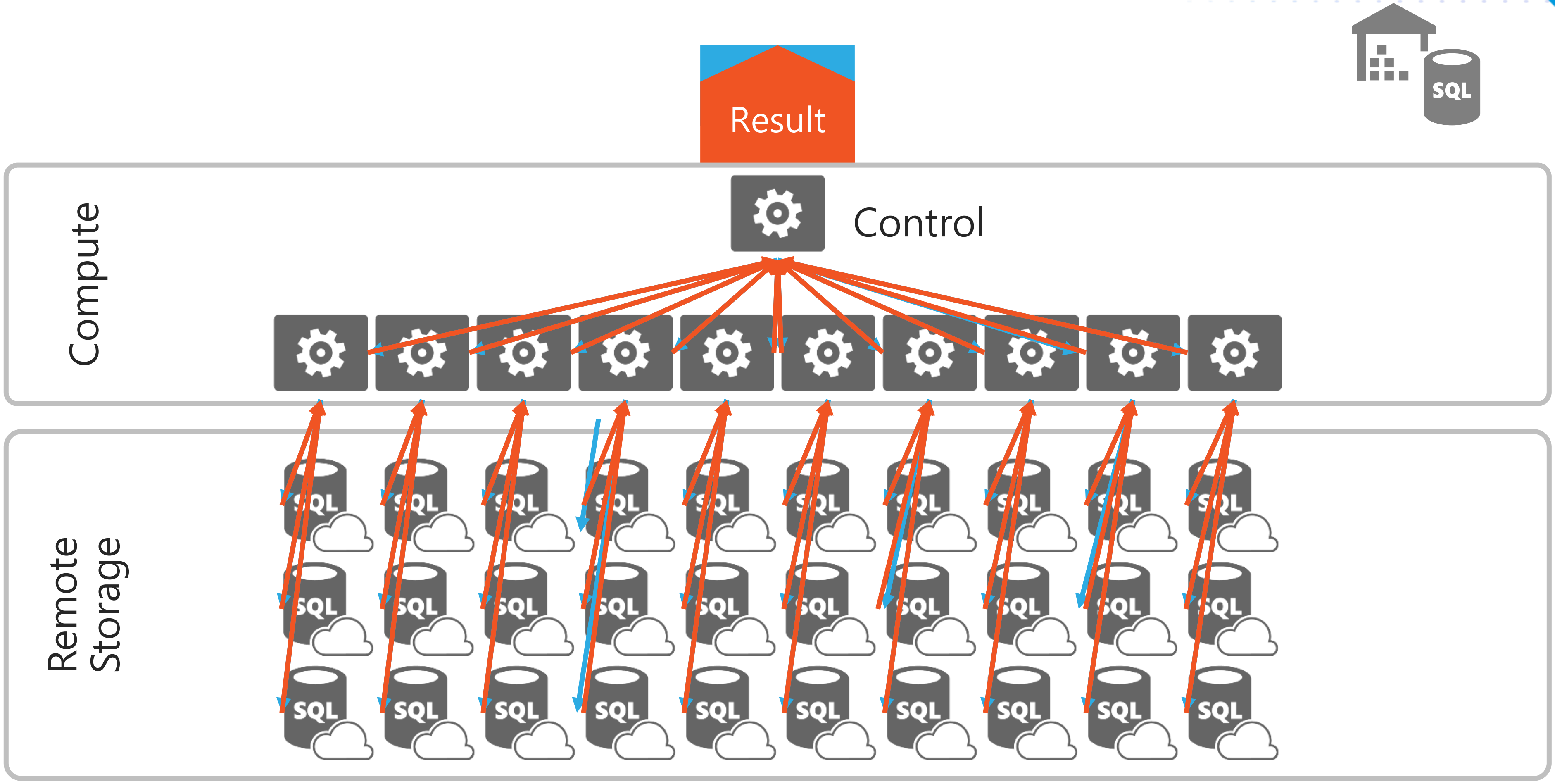
Built-in advanced security features including Encryption, Audit, Threat Detection, AAD and VNet

Seamlessly compatible across Microsoft and 3rd party services

Massively Parallel Proccessing Architecture

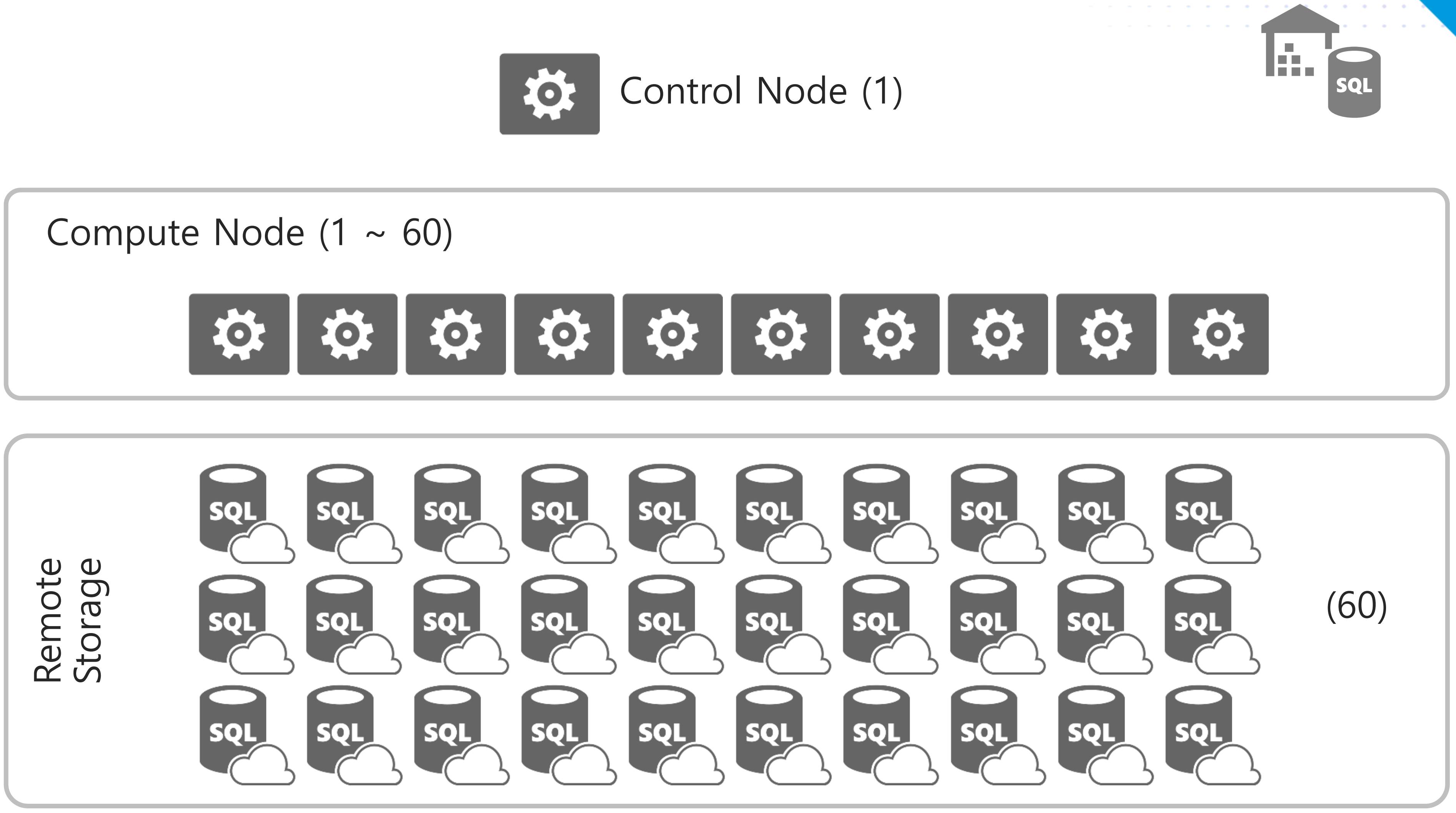


Distributed Queries



데이타솔루션 Microsoft

Scaling Compute & Storage Independently

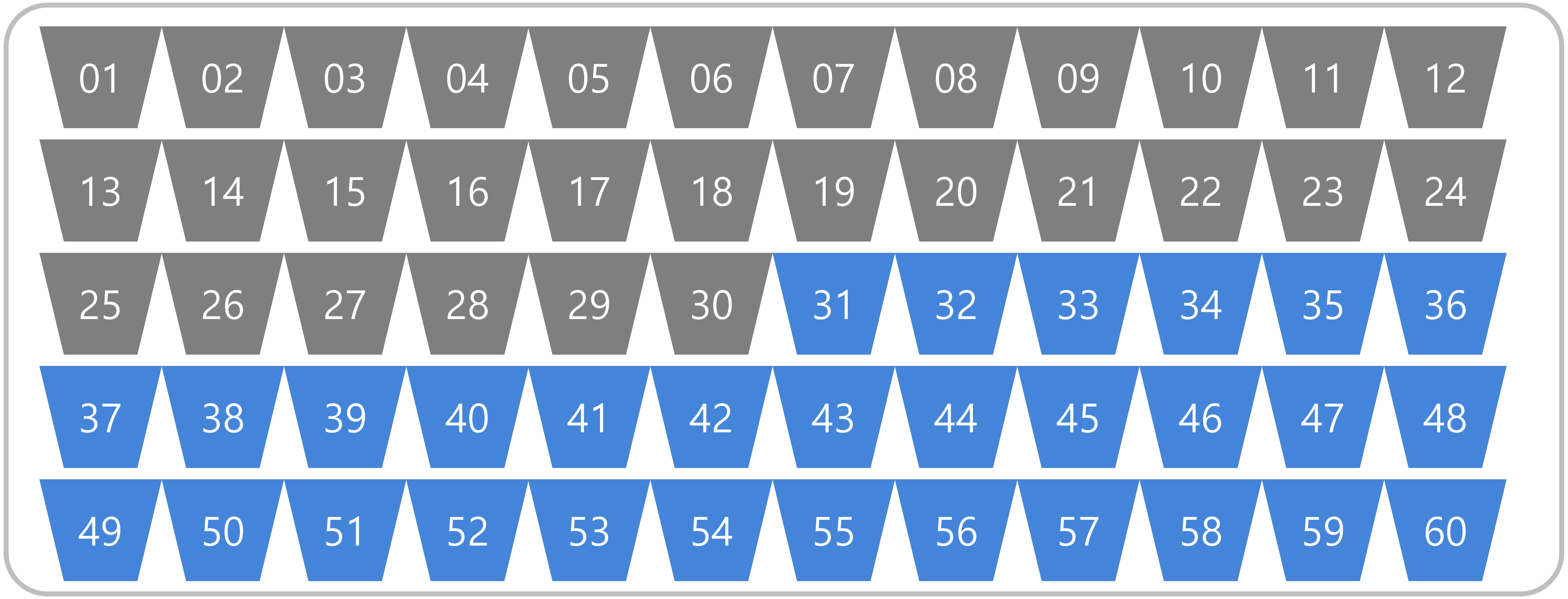


데이타솔루션 Microsoft

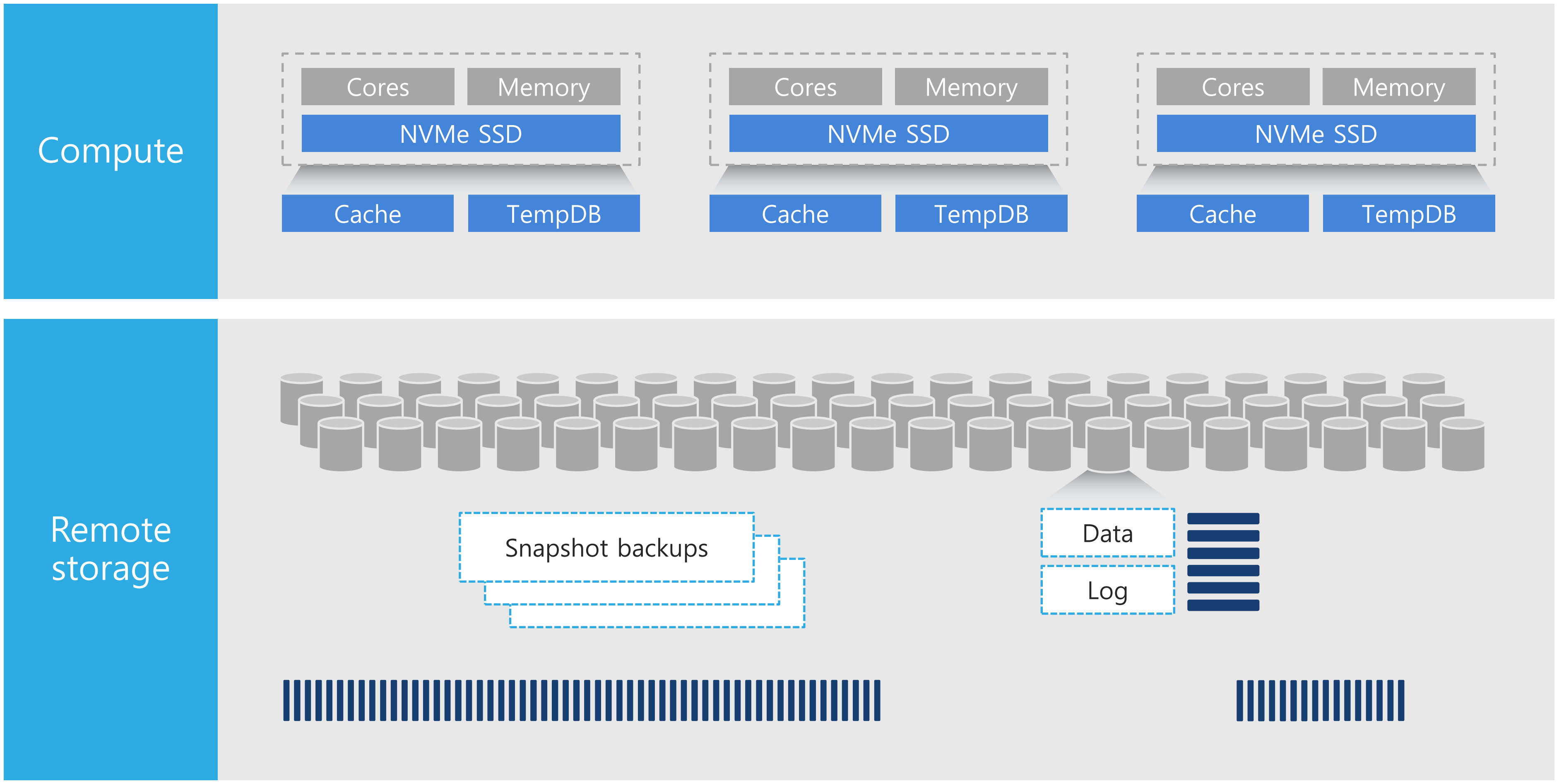
DataWarehouse Unit

DWu 1000c

Node = cDWu / 500



Compute optimized Gen2 Architecture



데이타솔루션 Microsoft

• Table Distribution

- Round-robin distributed
  - Distributes table rows evenly across all distributions at random.
- Hash distributed
  - Distributes table rows across the Compute nodes by using a deterministic hash function to assign each row to one distribution.
- Replicate
  - Full copy of table accessible on each Compute node.

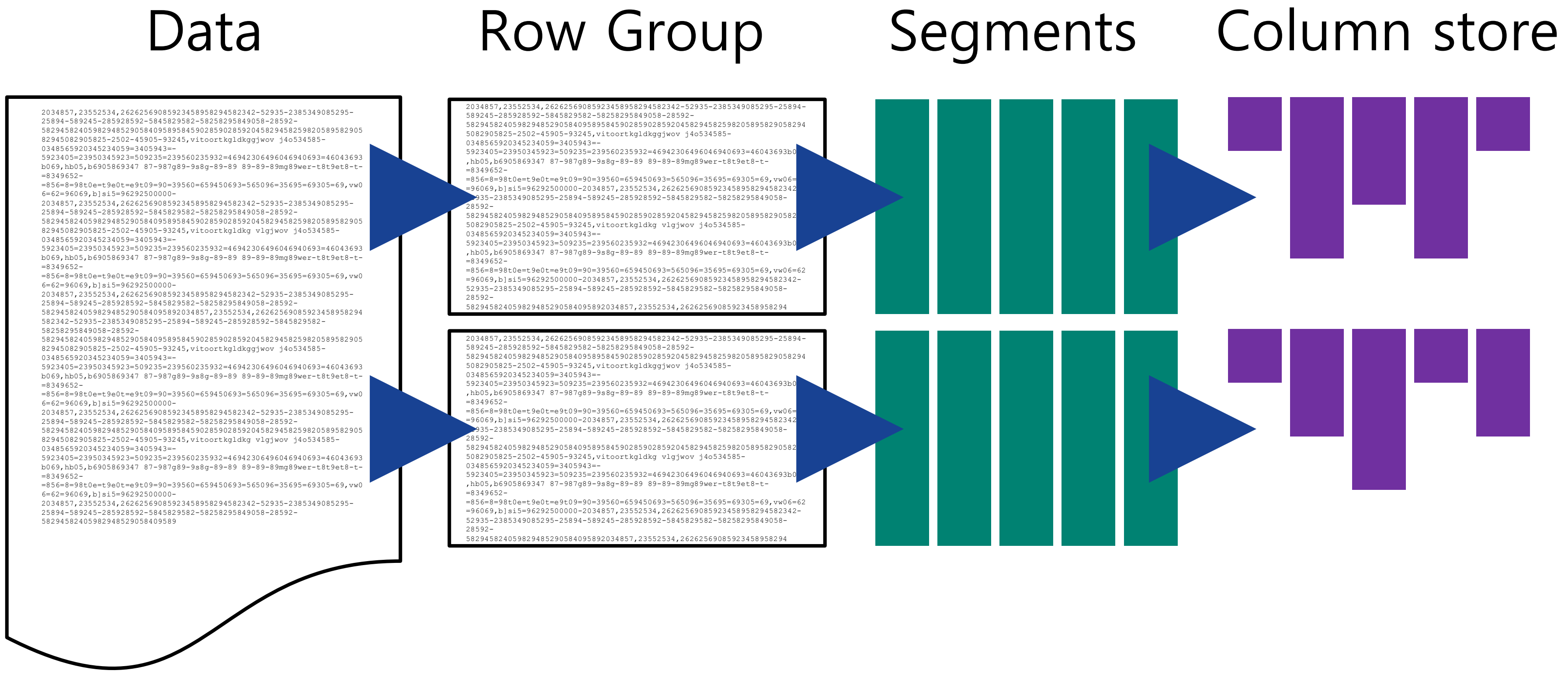
```
CREATE TABLE dbo.OrderTable
(
    OrderId  INT NOT NULL,
    Date     DATE NOT NULL,
    Name     VARCHAR(2),
    Country  VARCHAR(2)
)
WITH
(
  CLUSTERED COLUMNSTORE INDEX,
  DISTRIBUTION = HASH([OrderId]) |
          ROUND ROBIN |
          REPLICATE
);
```

Table Index

- Clustered Columnstore index
  - Highest level of data compression
  - Best overall query performance
  - Support for ordered Columnstore segments
- Clustered index
  - Performant for looking up a single to few rows
- Heap
  - Faster loading and landing temporary data
  - Best for small lookup tables
- Nonclustered indexes
  - Enable ordering of multiple columns in a table
  - Allows multiple nonclustered on a single table

```
-- Create table with index
CREATE TABLE orderTable
(
    OrderId  INT NOT NULL,
    Date     DATE NOT NULL,
    Name     VARCHAR(2),
    Country  VARCHAR(2)
)
WITH
(
     CLUSTERED COLUMNSTORE INDEX |
    HEAP |
    CLUSTERED INDEX (OrderId)
);


-- Add non-clustered index to table
CREATE INDEX NameIndex ON orderTable (Name);
```

Column Store taxonomy



Row Store VS Column Store

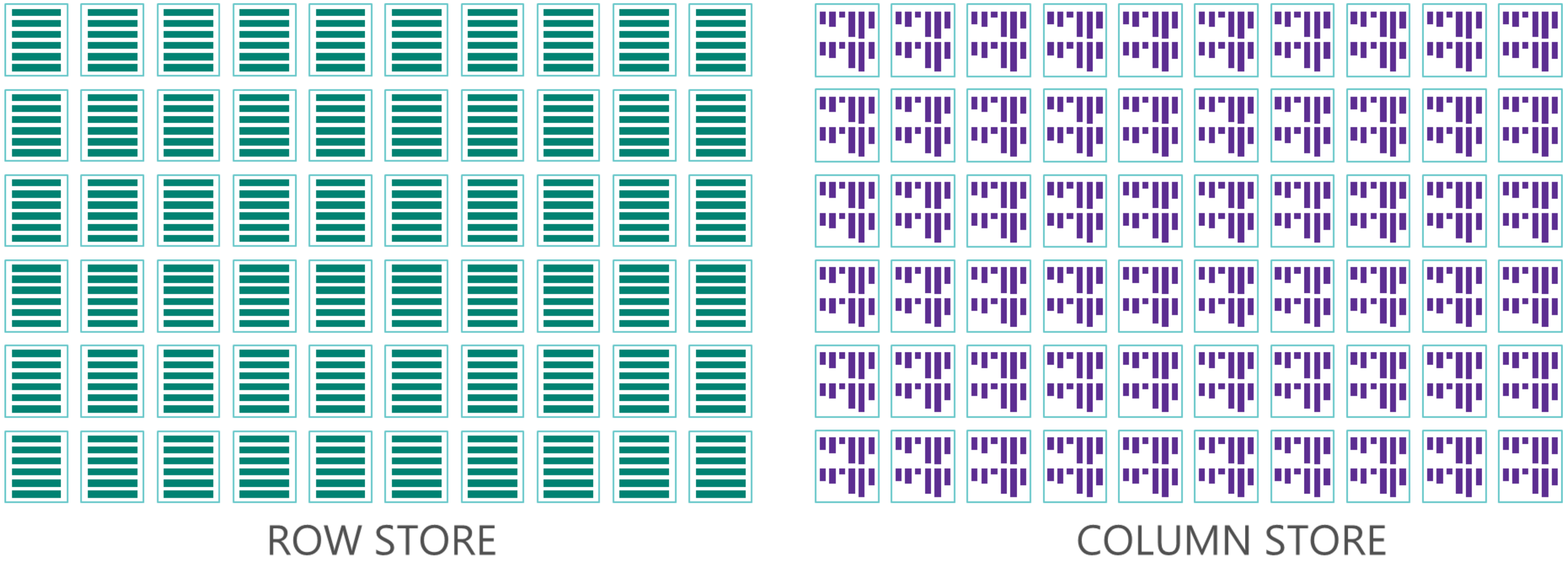


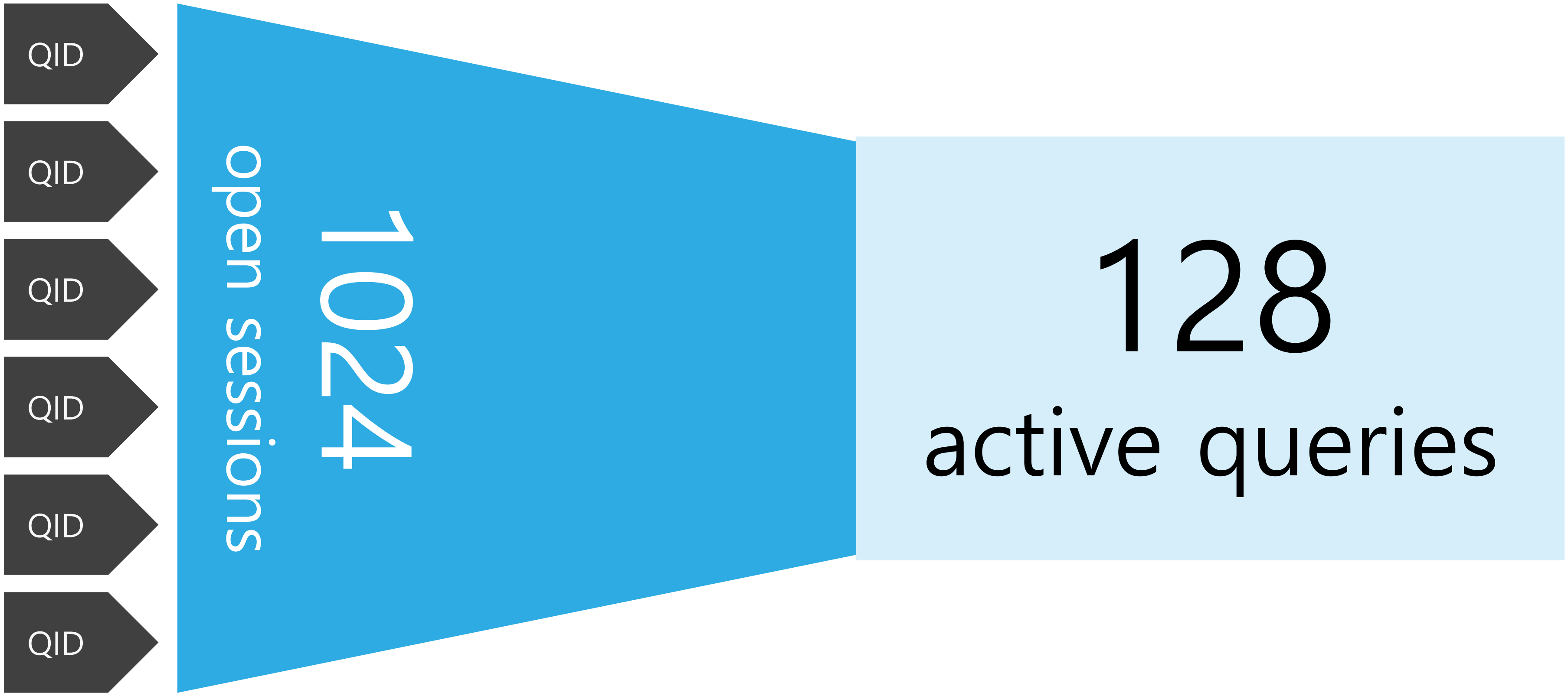
Table Partition

- Overview
  - Table partitions divide data into smaller groups
  - In most cases, partitions are created on a date column
  - Supported on all table types
  - RANGE RIGHT – Used for time partitions
  - RANGE LEFT – Used for number partitions

- Benefits
  - Improves efficiency and performance of loading and querying by limiting the scope to subset of data.
  - Offers significant query performance enhancements where filtering on the partition key can eliminate unnecessary scans and eliminate IO.

```
CREATE TABLE partitionedOrderTable
(
    OrderId  INT NOT NULL,
    Date     DATE NOT NULL,
    Name     VARCHAR(2),
    Country  VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]),
    PARTITION (
    [Date] RANGE RIGHT FOR VALUES (
    '2000-01-01', '2001-01-01', '2002-01-01',
    '2003-01-01', '2004-01-01', '2005-01-01'
    )
  )
);
```

Concurrency : Queries



Resource Class Type

- Static Resource Class

  Static resource classes allocate the same amount of memory independent of the current SLO.
  They are well suited for managing concurrency on a data set size that is fixed.

- Dynamic Resource Class

  Dynamic resource classes allocate a variable amount of memory depending on the current SLO.
  They are well suited for data sets that are growing or variable in size.
  Users start out with smallrc dynamic resource class by default.

Static resource classes:

```
staticrc10 | staticrc20 | staticrc30 |
staticrc40 | staticrc50 | staticrc60 |
staticrc70 | staticrc80
```

Dynamic resource classes:

```
smallrc | mediumrc | largerc | xlargerc
```

Concurrency Limits

Concurrency Slots

Service Level	Max Concurrent Queries	Max Concurrency Slots	Dynamic Resource Classes				Static Resource Classes							
			smallrc	mediumrc	largerc	xlargerc	staticrc10	staticrc20	staticrc30	staticrc40	staticrc50	staticrc60	staticrc70	staticrc80
DW100c	4	4	1	1	1	2	1	2	4	4	4	4	4	4
DW200c	8	8	1	1	1	5	1	2	4	8	8	8	8	8
DW300c	12	12	1	1	2	8	1	2	4	8	8	8	8	8
DW400c	16	16	1	1	3	11	1	2	4	8	16	16	16	16
DW500c	20	20	1	2	4	14	1	2	4	8	16	16	16	16
DW1000c	32	40	1	4	8	28	1	2	4	8	16	32	32	32
DW1500c	32	60	1	6	13	42	1	2	4	8	16	32	32	32
DW2000c	48	80	2	8	17	56	1	2	4	8	16	32	64	64
DW2500c	48	100	3	10	22	70	1	2	4	8	16	32	64	64
DW3000c	64	120	3	12	26	84	1	2	4	8	16	32	64	64
DW5000c	64	200	6	20	44	140	1	2	4	8	16	32	64	128
DW6000c	128	240	7	24	52	168	1	2	4	8	16	32	64	128
DW7500c	128	300	9	30	66	210	1	2	4	8	16	32	64	128
DW10000c	128	400	12	40	88	280	1	2	4	8	16	32	64	128
DW15000c	128	600	18	60	132	420	1	2	4	8	16	32	64	128
DW30000c	128	1200	36	120	264	840	1	2	4	8	16	32	64	128

Concurrency Limits

Memory

Service Level	Dynamic Resource Classes			
	smallrc	mediumrc	largerc	xlargerc
DW100c	25%	25%	25%	70%
DW200c	12.5%	12.5%	22%	70%
DW300c	8%	10%	22%	70%
DW400c	6.25%	10%	22%	70%
DW500c	20%	10%	22%	70%
DW1000c ~ DW30000c	3%	10%	22%	70%

Staticrc Memory

```
SELECT name, Effective_request_min_resource_grant_percent
  FROM sys.dm_workload_management_workload_groups_stats
  WHERE name like 'staticrc%'
```

데이타솔루션 Microsoft

Workload Classification

Overview

Allows you to map a query to an allocation of resources via pre-determined rules. Use this in combination with workload importance to effectively share resources across different workload types.

If a query request is not matched to a classifier, it is assigned to the default workload group (smallrc resource class).
The sys.workload_management_workload_classifiers and sys.workload_management_workload_classifier_details DMVs can be queried to view details about all active workload classifiers.

```
CREATE WORKLOAD CLASSIFIER classifier_name
WITH
(
   [WORKLOAD_GROUP = '<Resource Class>' ]
   [IMPORTANCE = {     LOW           |
                       BELOW_NORMAL  |
                       NORMAL        |
                       ABOVE_NORMAL  |
                       HIGH
                }
   ]
   [MEMBERNAME = ‘security_account’]
)
```

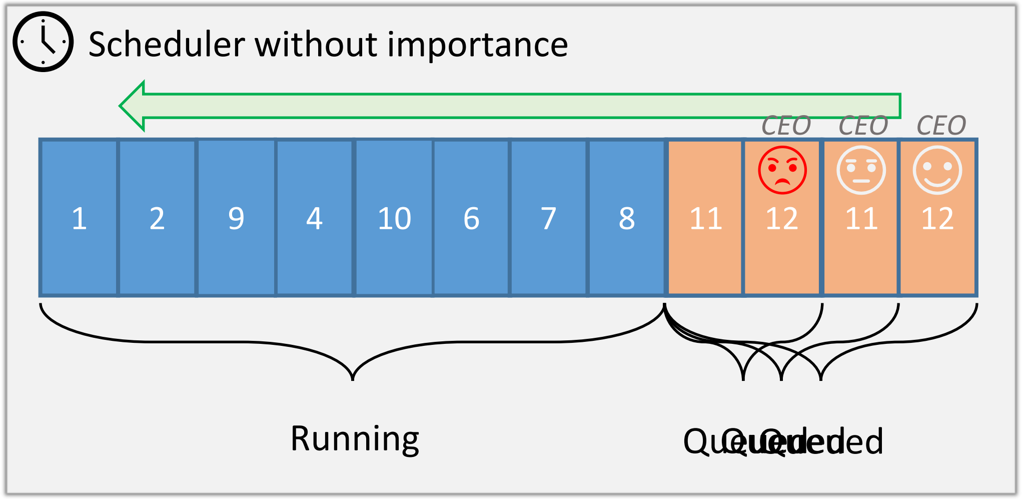
```
 WORKLOAD_GROUP: maps to an existing resource class
 IMPORTANCE: specifies relative importance of
             request
 MEMBERNAME: database user, role, AAD login or AAD
             group
```

WORKLOAD IMPORTANCE – No Importance

Overview

Workload importance allows you to prioritize the queries that get access to resources.

By default, queries are processed in a first-in, first-out order, and are released from the queue when resources required to execute the query are available.

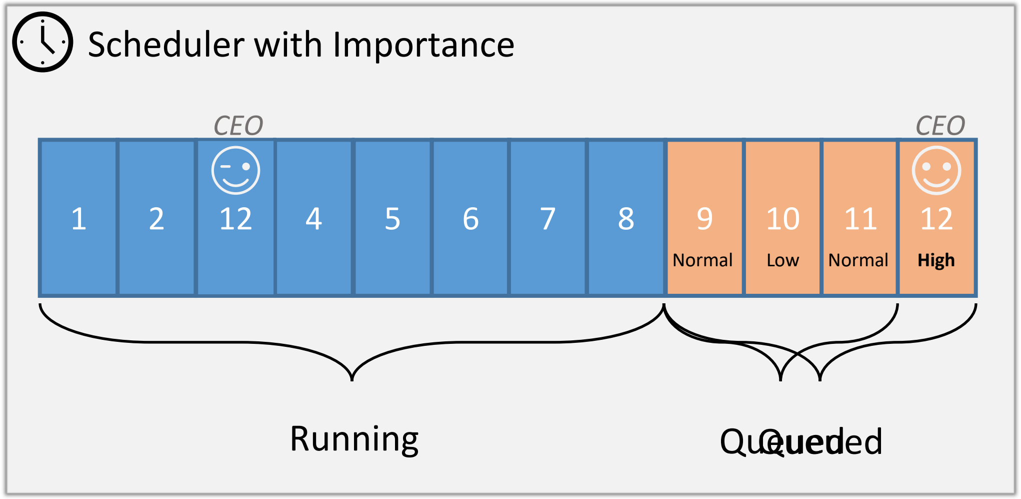


WORKLOAD IMPORTANCE – Importance

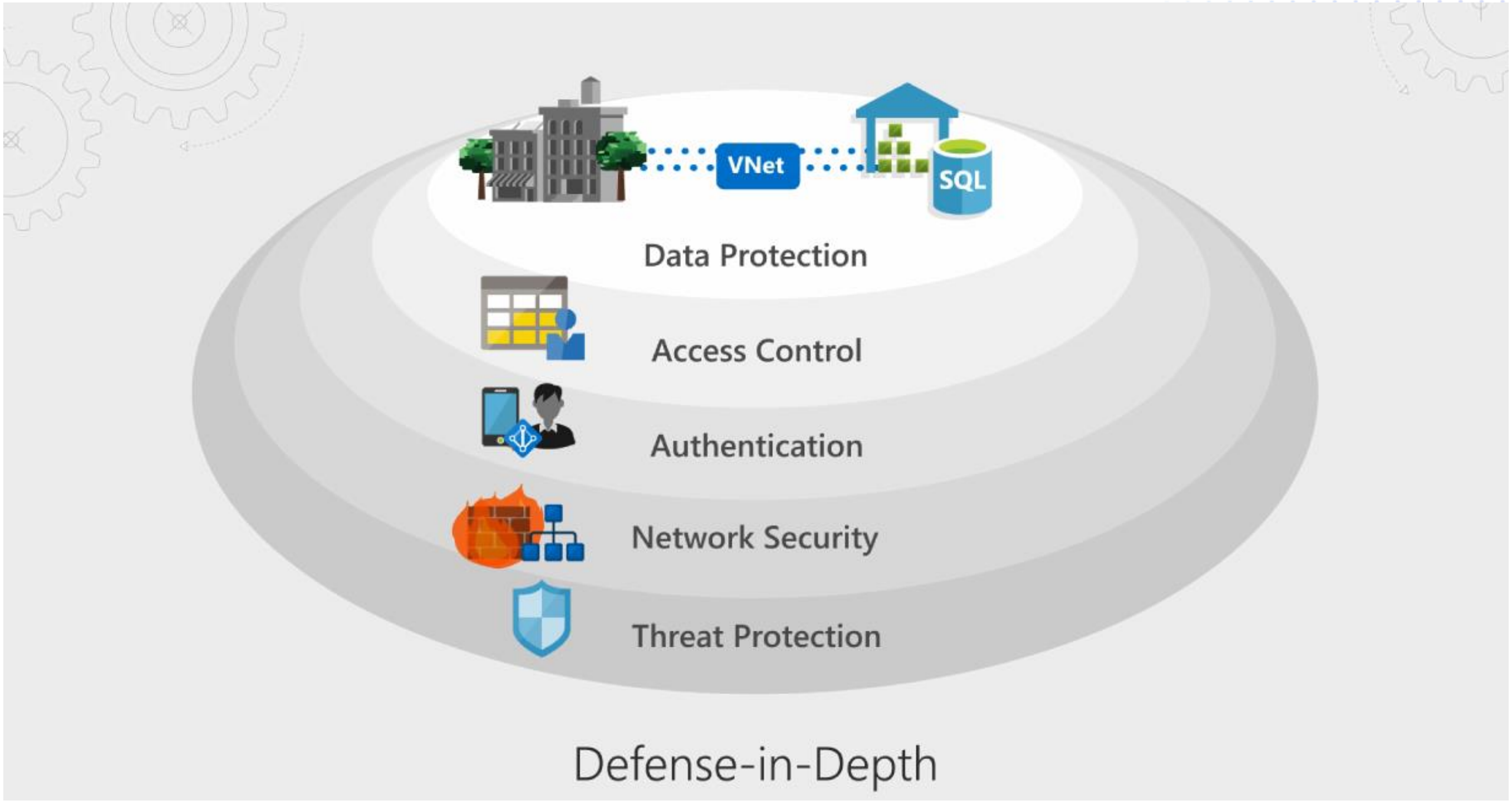
Overview

Workload importance allows you to prioritize the queries that get access to resources.

By specifying the priority in which queries should be executed, you can ensure that high-business value work is executed first on a busy data warehouse. These high-business value queries will remain on the queue until the required system resources are available.



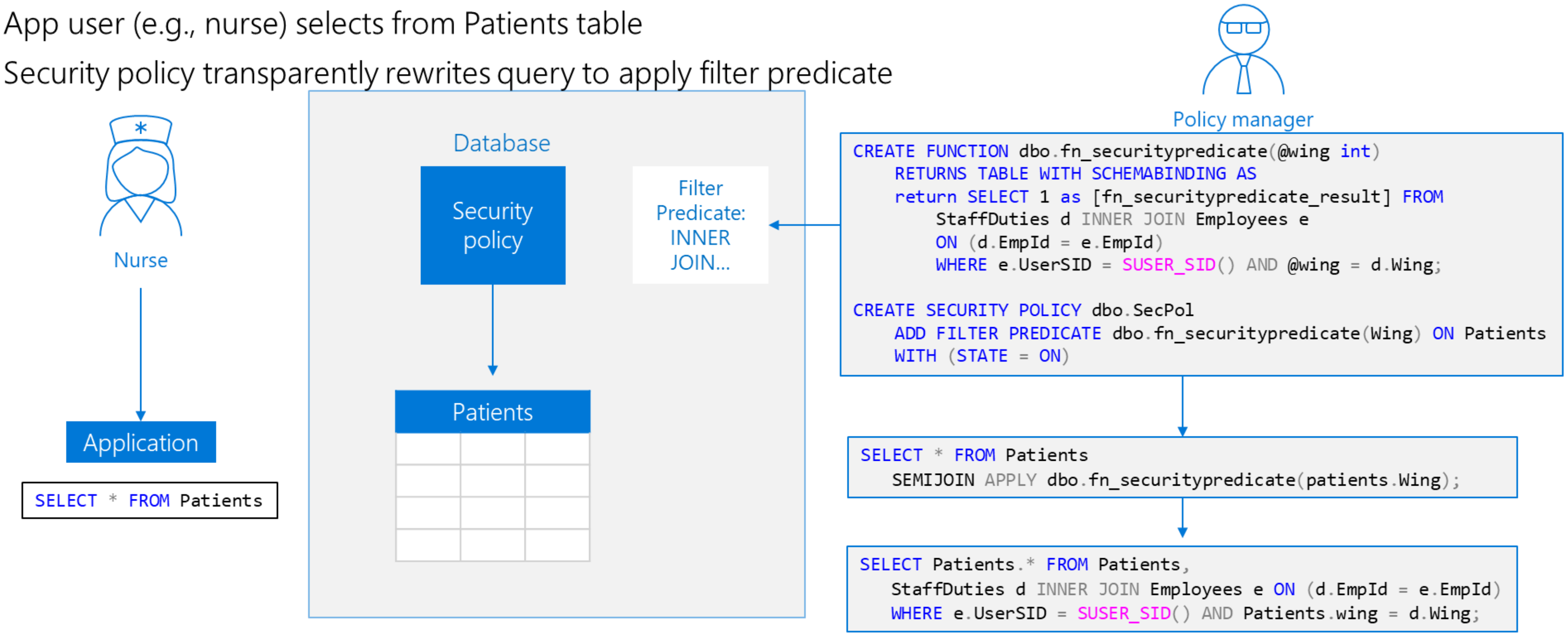
Enterprise grade security



Row Level Security

Three Steps:

1. Policy manager creates filter predicate and security policy in T-SQL, binding the predicate to the Patients table
2. App user (e.g., nurse) selects from Patients table
3. Security policy transparently rewrites query to apply filter predicate



```sql
SELECT * FROM Patients
```

```sql
CREATE FUNCTION dbo.fn_securitypredicate(@wing int)
    RETURNS TABLE WITH SCHEMABINDING AS
    return SELECT 1 as [fn_securitypredicate_result] FROM
        StaffDuties d INNER JOIN Employees e
        ON (d.EmpId = e.EmpId)
        WHERE e.UserSID = SUSER_SID() AND @wing = d.Wing;

CREATE SECURITY POLICY dbo.SecPol
    ADD FILTER PREDICATE dbo.fn_securitypredicate(Wing) ON Patients
    WITH (STATE = ON)
```

```sql
SELECT * FROM Patients
   SEMIJOIN APPLY dbo.fn_securitypredicate(patients.Wing);
```

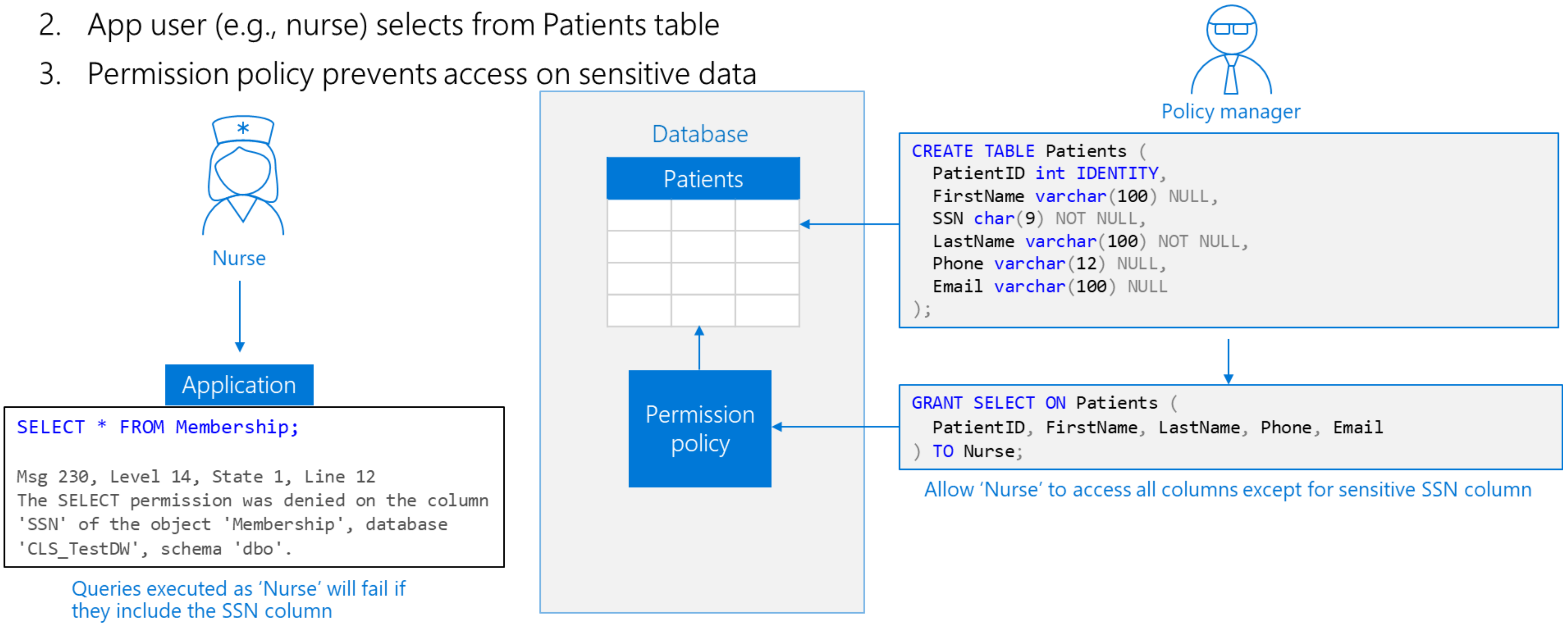
```sql
SELECT Patients.* FROM Patients,
   StaffDuties d INNER JOIN Employees e ON (d.EmpId = e.EmpId)
   WHERE e.UserSID = SUSER_SID() AND Patients.wing = d.Wing;
```

Access Control

Column Level Security

Three Steps:

1. Policy manager creates permission policy in T-SQL, binding the policy to the Patients table on a specific group
2. App user (e.g., nurse) selects from Patients table
3. Permission policy prevents access on sensitive data



```
SELECT * FROM Membership;

Msg 230, Level 14, State 1, Line 12
The SELECT permission was denied on the column
'SSN' of the object 'Membership', database
'CLS_TestDW', schema 'dbo'.
```

Queries executed as 'Nurse' will fail if they include the SSN column

```
CREATE TABLE Patients (
  PatientID int IDENTITY,
  FirstName varchar(100) NULL,
  SSN char(9) NOT NULL,
  LastName varchar(100) NOT NULL,
  Phone varchar(12) NULL,
  Email varchar(100) NULL
);
```

```
GRANT SELECT ON Patients (
  PatientID, FirstName, LastName, Phone, Email
) TO Nurse;
```

Allow 'Nurse' to access all columns except for sensitive SSN column

데이타솔루션 Microsoft

Dynamic Data Masking

Three steps

1. Security officer defines dynamic data masking policy in T-SQL over sensitive data in the Employee table. The security officer uses the built-in masking functions (default, email, random)

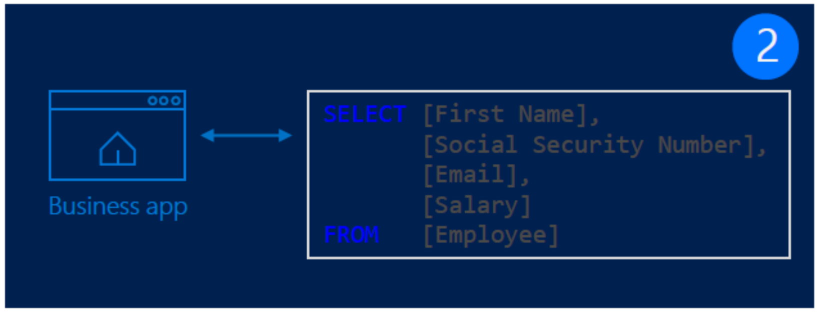
2. The app-user selects from the Employee table

3. The dynamic data masking policy obfuscates the sensitive data in the query results for non-privileged users

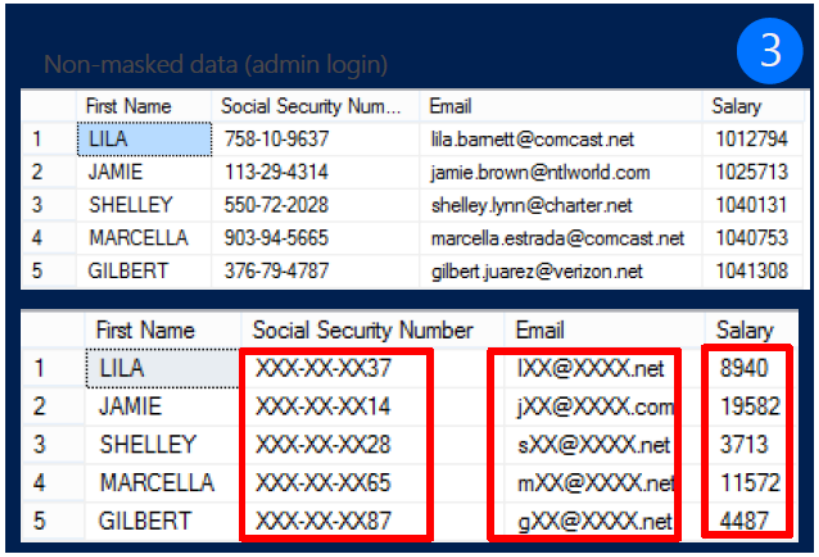
Security officer

1

```
ALTER TABLE [Employee]
ALTER COLUMN [SocialSecurityNumber]
ADD MASKED WITH (FUNCTION = 'DEFAULT()')

ALTER TABLE [Employee]
ALTER COLUMN [Email]
ADD MASKED WITH (FUNCTION = 'EMAIL()')

ALTER TABLE [Employee]
ALTER COLUMN [Salary]
ADD MASKED WITH (FUNCTION = 'RANDOM(1,20000)')

GRANT UNMASK to admin1
```



```
SELECT [First Name],
       [Social Security Number],
       [Email],
       [Salary]
FROM   [Employee]
```



Non-masked data (admin login)

	First Name	Social Security Num...	Email	Salary
1	LILA	758-10-9637	lila.barnett@comcast.net	1012794
2	JAMIE	113-29-4314	jamie.brown@ntlworld.com	1025713
3	SHELLEY	550-72-2028	shelley.lynn@charter.net	1040131
4	MARCELLA	903-94-5665	marcella.estrada@comcast.net	1040753
5	GILBERT	376-79-4787	gilbert.juarez@verizon.net	1041308

	First Name	Social Security Number	Email	Salary
1	LILA	XXX-XX-XX37	lXX@XXXX.net	8940
2	JAMIE	XXX-XX-XX14	jXX@XXXX.com	19582
3	SHELLEY	XXX-XX-XX28	sXX@XXXX.net	3713
4	MARCELLA	XXX-XX-XX65	mXX@XXXX.net	11572
5	GILBERT	XXX-XX-XX87	gXX@XXXX.net	4487

데이타솔루션 Microsoft

데이타솔루션

데이터로
새로운 세상을 그리다.

Thank you