Jared Martin
Prudhvi Sai Nikith
Tom K. Jose
October 3, 2021
ISMG 6080

Project Stage 2

- *Company Background*

Northwind Technology is a managed service provider specializing in providing technology infrastructure to small and medium businesses. The company offers a suite of off-the-shelf software for human resources, payroll, accounting, and messaging to customers. Northwind maintains the servers and supporting infrastructure for these applications and the customers can consume the applications without worrying about maintaining servers. The company also offers a service where customers can bring their own applications and Northwind will install, operate, and maintain the application. The company operates approximately 2500 servers in four data centers for around 100 customers.

The company has various engineering departments that support servers and applications for various purposes. These departments participate in projects to deploy new applications or add functionality to existing applications for customers. These departments are also responsible for ongoing maintenance of applications and servers.

- *Problem*

Northwind engineers need an accurate source for information about servers and applications deployed for customers. To fulfill customer requests, they need to know which servers to work on, which departments to assign work to, and when work can be performed. Currently, Northwind tracks information about servers in a spreadsheet. The spreadsheet was sufficient when Northwind had far fewer customers but is no longer useful. It is difficult to maintain, frequently has multiple versions in circulation with different information, and cannot be used to generate billing information for customers. The spreadsheet also suffers from data integrity problems due to misspellings and inconsistent data entry. Moving this information into a database will make the information more easily accessible to all engineer and make the information more trustworthy by enforcing consistency. A database will also enable the company to generate reports for the applications deployed for each customer, application versions for end-of-life tracking, and more.

- *Data Requirements*

Northwind deploys a variety of applications for customers. The company can manage one or more applications for each customer. There can be multiple versions of each application and the company needs to track the different versions. Northwind provides the application license to its customers and needs to track the quantity of licenses. Each application for each customer has a unique time window where it can be unavailable for maintenance. Applications are made by a vendor and the company needs to track sales and technical support contacts for each vendor.

Each application is supported by one of several engineering departments at Northwind. The department that supports an application is also responsible for supporting the servers that each application runs on. Each department has an engineer on-call for emergencies and the database needs to track the name and phone number of the on-call engineer.

Applications run on servers that have a unique name. The start and end of a server's lifecycle, operating system, number of CPUs, amount of memory, and storage capacity for each server needs to be tracked for billing and maintenance purposes. Billing is managed by a separate system that retrieves information about each server from this database. Each server resides in a data center. A data center is a physical location where many servers are located. Occasionally, engineers at Northwind need to contact a data center for work that requires physical interaction with a server. The Operating System of each server also needs to be tracked. Like the applications, Northwind provides a license for the Operating System that runs on each server and must track the license key, quantity of licenses, and the date that the vendor will stop providing support. Each Operating System is made by a vendor.

- *Four Sentence Template Data Requirements*

- Entities:
    - Database needs to store information about each application. For each application, the database needs to store its name, version, number of licenses, and license key.
    - Database needs to store information about each customer. For each customer, the database needs to store the company name, the industry the company operates in, contact information for the account representative, and contact information for an IT representative.
    - Database needs to store information about each data center. For each data center, the database needs to store its name, address, and a main phone number.
    - Database needs to store information about each engineering department. For each department, the database needs to store its name and contact information for the on-call engineer.
    - Database needs to store information about each Operating System. For each Operating System, the database needs to store its name, number of licenses, and the license key.
    - Database needs to store information about each server. For each server, the database needs to store its name, starting date, ending date, the number of CPUs, amount of memory, and its storage capacity.
    - Database needs to store information about each vendor. For each vendor, the database needs to store the name, contact information for an account representative, and contact information for technical support.
- Relationships:
    - Database needs to store the datacenter each server resides in.
    - Database needs to store the operating system each server runs.
    - Database needs to store the application each server runs.
    - Database needs to store the vendor for each operating system.
    - Database needs to store the vendor for each application.
    - Database needs to store the department that supports each application.

    - Database needs to store the applications deployed for each customer.
- Relationship Cardinalities:
    - Each server resides in a single datacenter. Each datacenter has many servers. For example, server IS40 resides in the datacenter us-west-2. The datacenter us-west-2 has the servers IS40, DS100, and more.
    - Each server runs a single operating system. Each operating system runs on many servers. For example, server IS40 runs operating system Windows Server 2019. The operating system Windows Server 2019 runs on the servers IS50, DS09, and more.
    - Each server runs a single application. Each application runs on many servers. For example, server IS40 runs the application named Carbon Black. The application Carbon Black runs on servers IS40, IS50, and more.
    - Each vendor makes multiple applications or operating systems. Each application or operating system is made by one vendor. For example, vendor Microsoft makes applications Exchange, Dynamics, and more. The application named Exchange is only made by Microsoft.
    - Each department supports multiple applications. Each application is supported by one department. For example, the application named Carbon Black is supported by the Security Engineering department. The Security Engineering department supports applications named Carbon Black, Symantec, and more.
    - Each customer can have many applications. Each application can be deployed for many customers. For example, the customer named Blue Yonder Airlines can have the applications named Exchange, Carbon Black, and more. The application named Carbon Black can be deployed for Blue Yonder Airlines, AdventureWorks, and more.
- Relationship Attributes:
    - Database needs to store the maintenance window for each application for each customer. For example, the application named Carbon Black that is deployed for the customer named Blue Yonder airlines can have a maintenance window of Mondays from 1 AM to 3 AM.

    - *Sample Data*

Servers and applications:

Server Name	**Operating System**	**Application**	**Version**	**License Key**	**Customer**	**MaintWindow**	**Department**	**Data Center**
AS01	Windows Server 2019	WorkDay	33	FDSF27578	Baldwin Museum	Monday 1-3	HR Eng	us-west-2
PS09	Windows Server 2012	PeopleSoft	9.2	AYK987	Baldwin Museum	Monday 1-3	Accounting Eng	us-west-2
AS19	Windows Server 2016	Salesforce	Fall 21	LKJ987-54KJH	Baldwin Museum	Monday 1-3	Sales Eng	us-east-1
AS21	Windows Server 2012	Jira	7	KJK9898j	Woodgrove Bank	Sunday 1-5	Project Eng	us-east-2

SEC90	Windows Server 2016	Carbon Black	8	LKJJL97KJH54	Blue Yonder Airlines	Wed 0-2	Security Eng	us-west-2
AS21	Windows Server 2016	Confluence	8.3.2	654KHKHTRD	Woodgrove Bank	Sunday 1-5	Project Eng	us-west-2
AS40	Windows Server 2012	WorkDay	32	L988413213	Blue Yonder Airlines	Wed 0-2	HR Eng	us-west-2
AS68	Windows Server 2016	Tableau	12	98KJHTBLEAU	Blue Yonder Airlines	Wed 0-2	Finance Eng	us-west-2
SEC80	Windows Server 2016	Carbon Black	8	KKHKHTRD	Woodgrove Bank	Sunday 1-5	Security Eng	us-west-2

Customers:

Name	**Account Contact**	**Tech Contact**
Baldwin Museum of Science	James @ 720-544-6548	Richard. 987-654-6541
Woodgrove Bank	Molly Shatner	Lumberg. bill@woodgrove.com
Blue Yonder Airlines	Russ May. 587-6325	Jason @ 987-651-8546

Data Centers:

Name	**Address**	**Phone**
us-west-2	123 Main Street, Denver, CO, 80265	658-985-2589
us-east-1	9748 Industrial Ave, Renton, VA, 54986	412-867-1813
us-east-2	9 Forest Way, Dayton, OH, 98762	892-573-9852
us-central-1	19 Prairie View, Chicago, IL, 80246	824-671-3955

Departments:

Name	OnCall	Phone	Phone
HR Eng	Jenny	303-867-5309	564-697-2365
Accounting Eng	Sean	720-544-1235	
Sales Eng	James	987-654-1234	548-445-6971
Project Eng	Heather	303-526-8551	
Security Eng	Jeff B.	658-975-3268	
Finance Eng	Ed	142-021-2022	

Vendors:

Name	**Rep**	**Tech Support**
Microsoft	Anna Hodges. ahodge@microsoft.com	1-800-MICROSOFT

Red Hat	Jim Aaron. jim@redhat.com	1-800-REDHAT

- *Entity Relationship Diagram*

Explanation of Entities and Relationships:

- Application: An application is a piece of software the company deploys and maintains for customers.
- Customer: A customer is a business that an application is deployed and maintained for.
- DataCenter: A datacenter is a physical location where servers reside.
- Department: A department is a group within the company that supports applications.
- OperatingSystem: An operating system runs on a server.
- Server: A server is a computer that resides in a datacenter and runs an operating system and application.
- Vendor: A vendor makes an application or operating system that runs on servers.
- Server-resides-in-DataCenter: This records the physical location of a server.
- Server-runs-OperatingSystem: This records the operating system that a server runs.
- Server-runs-Application: This records the application a server runs.
- OperatingSystem-made-by-Vendor: This records the vendor that makes an operating system.
- Application-made-by-Vendor: This records the vendor that makes an application.
- Application-supported-by-Department: This records the department that supports an application.
- Application-deployed-for-Customer: This records the applications that are deployed for each customer.

- *Relational Data Model*

- Tables:
  - Application: (AppName, Version, LicenseCount, LicenseKey)
  - Customer: (Customer#, Name, Industry, AcctRep, AcctRepPhone, TechRep, TechRepPhone)
  - DataCenter: (DCName, Address, MainPhone)
  - Department: (DeptName, OnCallName, OnCallPhone)
  - OperatingSystem: (OSName, LicenseCount, LicenseKey)
  - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage)
  - Vendor: (VendorName, AcctRepName, AcctRepPhone, SupportPhone, SupportEmail)
- Relationships
  - Server-resides-in-DataCenter: 1-n relationship. Add primary key of DataCenter table as a foreign key to Server table.
    - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage, *DCName*)
    - Foreign Key: DCName
  - Server-runs-OperatingSystem: 1-n relationship. Add primary key of

OperatingSystem table as a foreign key to Server table.

        - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage, *DCName*, *OSName*)
        - Foreign Key: OSName
    - Server-runs-Application: 1-n relationship. Add primary key of Application table as a foreign key to Server table.
        - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage, *DCName*, *OSName, AppName, AppVersion*)
        - Foreign Key: AppName, AppVersion
    - OperatingSystem-made-by-Vendor: 1-n relationship. Add primary key of Vendor table to OperatingSystem table.
        - OperatingSystem: (OSName, LicenseCount, LicenseKey, *VendorName*)
        - Foreign Key: VendorName
    - Application-made-by-Vendor: 1-n relationship. Add primary key of Vendor table to Application table.
        - Application: (AppName, Version, LicenseCount, LicenseKey, *VendorName*)
        - Foreign Key: VendorName
    - Application-supported-by-Department: 1-n relationship. Add primary key of Department table to Application table.
        - Application: (AppName, Version, LicenseCount, LicenseKey, *VendorName, DeptName*)
        - Foreign Key: DeptName
    - Application-deployed-for-Customer: m-n relationship. Create a new table with primary keys from Application and Customer tables with relationship attribute MaintWindow.
        - CustomerApplication: (AppName, Version, Customer#, MaintWindow)
        - Foreign Keys: AppName, Version, Customer#
- Final Conversion
    - Application: (AppName, Version, LicenseCount, LicenseKey, *VendorName, DeptName*)
        - Foreign Keys: VendorName, DeptName
    - Customer: (Customer#, Name, Industry, AcctRep, AcctRepPhone, TechRep, TechRepPhone)
    - CustomerApplication: (AppName, Version, Customer#, MaintWindow)
        - Foreign Keys, AppName, Version, Customer#
    - DataCenter: (DCName, Address, MainPhone)
    - Department: (DeptName, OnCallName, OnCallPhone)
    - OperatingSystem: (OSName, LicenseCount, LicenseKey, *VendorName*)
        - Foreign Keys: VendorName
    - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage, *OSName, AppName, AppVersion*)
        - Foreign Keys: OSName, AppName, AppVersion
    - Vendor: (VendorName, AcctRepName, AcctRepPhone, SupportPhone, SupportEmail)
- Functional Dependencies and Normal Forms:
    - Application: (AppName, Version, LicenseCount, LicenseKey, *VendorName,*

  DeptName)
    - FD: AppName + Version → LicenseCount, LicenseKey, VendorName, DeptName. In BCNF.
  - Customer: (Customer#, Name, Industry, AcctRep, AcctRepPhone, TechRep, TechRepPhone)
    - FD: Customer# → Name, Industry, AcctRep, AcctRepPhone, TechRep, TechRepPhone. In BCNF.
  - CustomerApplication: (AppName, Version, Customer#, MaintWindow)
    - FD: AppName + Version + Customer# → MaintWindow. In BCNF.
  - DataCenter: (DCName, Address, MainPhone)
    - FD: DCName → Address, MainPhone. In BCNF.
  - Department: (DeptName, OnCallName, OnCallPhone)
    - FD: DeptName → OnCallName, OnCallPhone
  - OperatingSystem: (OSName, LicenseCount, LicenseKey, *VendorName*)
    - FD: OSName → LicenseCount, LicenseKey, VendorName. In BCNF.
  - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage, *OSName, AppName, AppVersion*)
    - FD: ServerName → StartDate, EndDate, CPUs, Memory, Storage, OSName, AppName, AppVersion. In BCNF.
  - Vendor: (VendorName, AcctRepName, AcctRepPhone, SupportPhone, SupportEmail)
    - FD: VendorName → AcctRepName, AcctRepPhone, SupportPhone, SupportEmail. In BCNF.
- Final Result After Normalization
  - Application: (AppName, Version, LicenseCount, LicenseKey, *VendorName, DeptName*)
    - Foreign Keys: VendorName, DeptName
  - Customer: (Customer#, Name, Industry, AcctRep, AcctRepPhone, TechRep, TechRepPhone)
  - CustomerApplication: (AppName, Version, Customer#, MaintWindow)
    - Foreign Keys: AppName, Version, Customer#
  - DataCenter: (DCName, Address, MainPhone)
  - Department: (DeptName, OnCallName, OnCallPhone)
  - OperatingSystem: (OSName, LicenseCount, LicenseKey, *VendorName*)
    - Foreign Keys: VendorName
  - Server: (ServerName, StartDate, EndDate, CPUs, Memory, Storage, *OSName, AppName, AppVersion*)
    - Foreign Keys: OSName, AppName, AppVersion
  - Vendor: (VendorName, AcctRepName, AcctRepPhone, SupportPhone, SupportEmail)
- Check Foreign Keys
  - There are 8 tables with 9 foreign keys. The number of foreign keys is greater than the number of tables minus 1 (8 – 1 = 7) so this satisfies the minimum number of foreign keys.