

DAY-1

29/08/23

+

•

○

Data



- Collection of related info
- Types:
 - > Structured – SQL, Oracle, Excel
 - > Semi-structured – JSON
 - > Unstructured – social media
 - > 3D – facts (aggregation)
 - dimension (collection of attributes)

Managing Data

- Data should be stored in a permanent storage

File System

- Problems in reading and managing data
- Security
- Data redundancy
- Program data independence
- Inflexibility of data formats
- Access Issues

A – Atomicity

C – Consistency

I – Isolation

D – Durability

RDBMS

- Most popular model of DBMS
- More scientific
- Data is stored in tables
- Relations can be normalized
- Each row has a unique value
- Each column has values from the same domain

Data Model

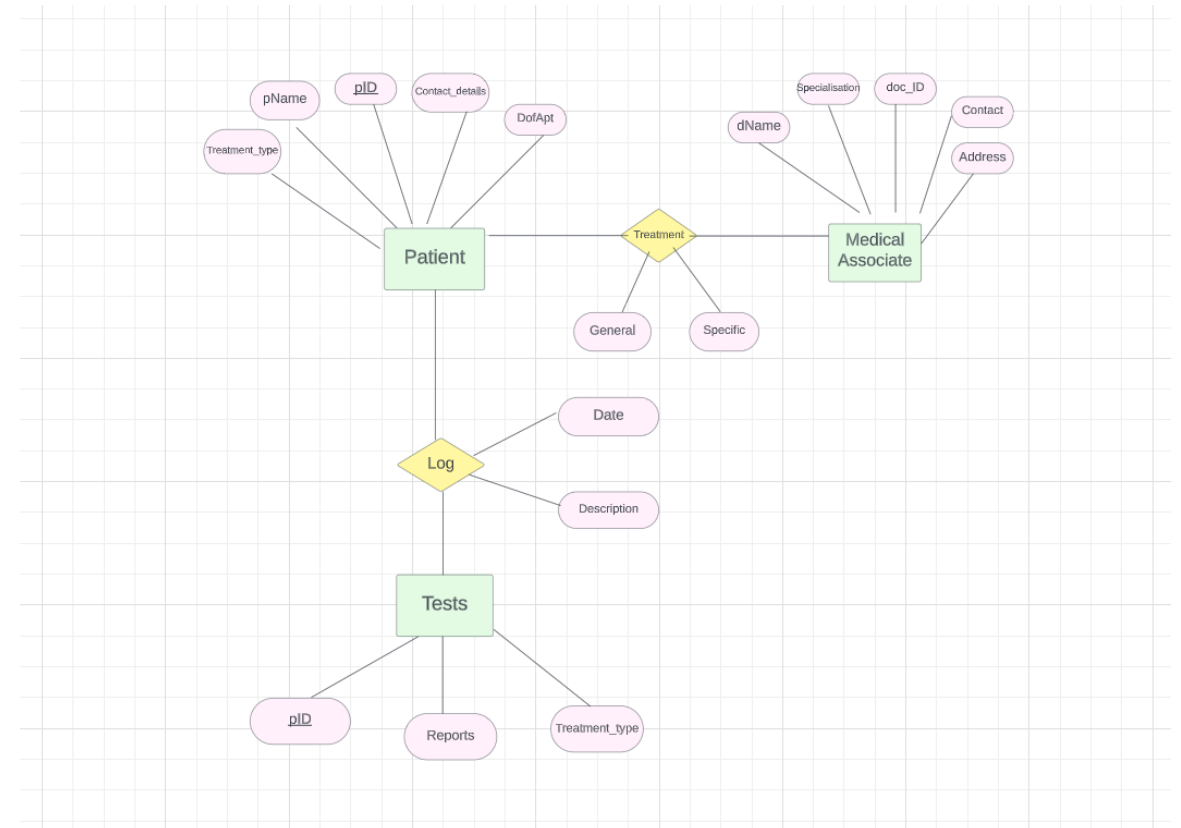
+



- Uses the concept of relationship and maintains the connectivity
- It consists of data constraints (rules) (default, check)
- Defines logical structure of how a database is modelled
- Defines how data is connected to each other

Entity-Relationship Model

- Based on real-world entities
- Best used for the conceptual design of a database



Functional Dependency



- A concept that specifies the relationship between two sets of attributes
- $X \rightarrow Y$ (X – determinant, Y – dependent) between primary key and non-key attributes
- Trivial Dependency
- Non – Trivial Dependency

Normalization

- Avoiding data redundancy, insertion anomaly, update anomaly, and delete anomaly
- To reduce ambiguity
- Different types:
 - > 1NF

1NF

- Eliminate the repeating groups in individual tables
- Does not contain composite or multi-valued attribute
- 1 NF when:
 - > only single valued attributes
 - > domain does not change

2NF

- Every non-key attribute are functionally dependent on composite key
- If it's not in 2NF -> update anomaly
- 2NF when:
 - > 1NF
 - > No partial dependency

3NF

- Prob in 2NF:
 - > Suffer from update anomaly
 - > Transitive dependency
- 3NF when:
 - > must be in 2NF
 - > Transitive functional dependency of non-prime attribute on any super key should be removed

Employee_ID	Name	Job_Code	Job	State_Code	Home_State
E001	Alice	J01	Chef	26	Michigan
		J02	Waiter	26	Michigan
E002	Bob	J02	Waiter	56	Wyoming
		J03	Bartender	56	Wyoming
E003	Robin	J04	Manager	56	Wyoming

1NF:

Employee_ID	Name	Job_Code	Job	State_Code	Home_State
E001	Alice	J01	Chef	26	Michigan
E001	Alice	J02	Waiter	26	Michigan
E002	Bob	J02	Waiter	56	Wyoming
E002	Bob	J03	Bartender	56	Wyoming
E003	Robin	J04	Manager	56	Wyoming

2NF:

Employee_ID	Job_Code
E001	J01
E001	J02
E002	J02
E002	J03
E003	J04

Employee_ID	Name	State_Code	Home_State
E001	Alice	26	Michigan
E002	Bob	56	Wyoming
E003	Robin	56	Wyoming

Job_Code	Job
J01	Chef
J02	Waiter
J03	Bartender
J04	Manager

3NF:

Employee_ID	Name	State_Code
E001	Alice	26
E002	Bob	56
E003	Robin	56

Employee_ID	Job_Code
E001	J01
E001	J02
E002	J02
E002	J03
E003	J04

Job_Code	Job
J01	Chef
J02	Waiter
J03	Bartender
J04	Manager

State_Code	Home_State
26	Michigan
56	Wyoming

INTRODUCTION TO DATA WAREHOUSING

+

•

○

The Business Problem

+

•

- Data can be inconsistent, duplicated, and contradictory
- Data is spread across many systems
- Fundamental questions can't be answered

What is a Data Warehouse

+



- Centralized repository for storing huge amounts of data records
- Optimized for querying (read the data)
- Incrementally loaded with new data regularly
- Basis for enterprise BI solutions

Data Warehouse Architectures

- Central Data Warehouse
- Departmental Data Marts
- Hub-and-Spoke

+



Data Warehousing Projects

Data Warehousing Project Roles



Components of a Data Warehousing Solution

- Data Sources
- ETL and Data Cleansing
- Data Warehouse
- Reporting & Analysis

E – Extract

T – Transform

L – Load



DAY 2

30/08/23

+

•

○

The Dimensional Model

+



- Transactions:
 - > OLTP – Online Transaction Process – Databases – CRUD (create, read, update, delete)
 - > OLAP – Online Analytical Process – Business Intelligence Implementation
 1. Data Warehouse – historical data (2D)
 2. Cube - Analysis Solution – multi dimensional data
- Facts and dimensions – facts (numbers)
 - dimensions (attributes) – currency, product, customer

The Dimensional Model

+



- Schemas: structure of the data
- 2 types : star & snowflake schema
- Snowflake – connecting dimensions with sub dimensions and sub - sub dimensions

Documenting Dimensional Models



Type of Measure

- Additive
- Semi – Additive
- Non – Additive

BIG DATA

+

•

○

Big Data

- High volume
- High velocity
- High variety

WHY BIG DATA?

- Store Information

+



Hadoop Architecture

- Master slave architecture
- RAID (Randomized Access Independent Disk)
- RAID 0, RAID 1, RAID 5, RAID 10(1+0)

+



AZURE + FUNDAMENTALS



Cloud Computing

- What is Cloud Computing?

Delivery of computing services over the internet enabling faster innovation, flexible resources, and economies of scale

-Compute –Networking –Storage

- Cloud Models

->Private Cloud – Within an organization

->Public Cloud – Used publicly (azure)

->Hybrid Cloud – Combination of two

- CapEx (Capital expenditure) vs. OpEx (Operational expenditure)
- CapEx – Spending money on physical infrastructure
- OpEx – Pay-as-you-go services

Cloud Computing

- Consumption-based model:
 - > end users pay for resources they use
- Benefits of Cloud:
 - > High availability -> Scalability -> Predictability -> Governance > Elasticity
 - > Reliability -> Security -> Manageability
- Cloud Services
 - > Infrastructure as a Service (IaaS)
 - > Platform as a Service (PaaS)
 - > Software as a Service (SaaS)
- Shared Responsibility Model



Azure Architecture & Services

- Explore Sandbox
- Azure Architectural Components
- Azure resources:

VM's, Storage Accounts, Database, Resource Groups, Web Apps



Azure Architecture & Services

- Azure storage access tiers:
 - > Hot – accessed frequently
 - > Cool – infrequently accessed and stored for at least 30 days
 - > Archive - 180 days
- Identity, Access, & Security
 - > Azure Active Directory (AAD) – identity & access management services
- Azure Multi-Factor Authentication
- Azure role-based access control (RBAC)

MANAGEMENT & GOVERNANCE



Cost Management

- Factors affecting costs:

-> Resource Type -> Consumption -> Maintenance -> Geography -> Network Traffic

-> Subscription

+

•

DAY 3

31/08/23

+

•

○

What to use for Data

- Storage Account
- Data Lake Store
- Azure Databricks
- Azure CosmosDB
- Azure SQL Database
- Azure Synapse Analytics
- Azure Stream Analytics
- Azure Data Factory



TRANSACTION – SQL



What is Transact – SQL

- SQLDB – Microsoft
 - > T – SQL
 - Oracle
 - > PL – SQL
- SQL Statement Types:
 - > DDL – physical structure (CREATE, ALTER, DROP)
 - > DML – logical information (INSERT, UPDATE, DELETE, SELECT, TRUNCATE)
 - > DCL – permissions assignments (GRANT, REVOKE, DENY)

+



What is Transact – SQL

- Part Names
 - > Table Only – Single Part
 - > Schema.Table – Two Part Table
 - > Database.Schema.Table – Three Part Table
 - > Server.Database.Schema.Table – Four Part Table

SQL – Lab

- SQL Query Editor
 - > Firewall Rules
 - Server Level
 - allowing
 - Database Level
 - user specific

✓ Your deployment is complete



Deployment name : Microsoft.SQLDatabase.ne... Start time : 8/31/2023, 10:31:14 AM
Subscription : npunext-1680261916031 Correlation ID : a2b72110-9e6c-442a-9e20...
Resource group : Nabiha

> Deployment details

✓ Next steps

[Go to resource](#)

```
4 select * from tblnew
5 --alter table
6 ALTER Table tblnew
7 add age int
8
9 select * from [SalesLT].[Product]
```

Results Messages

🔍 Search to filter items...

id	name	age
----	------	-----

No results

SQL – Lab

- ALTER
- INSERT
- UPDATE
- DELETE
- TRUNCATE

```

34  --delete
35  delete from License
36  where custname = 'Jack'
37
38

```

Results Messages

Search to filter items...

custid	custname
1	Robin

```

38  --truncate
39  truncate table License
40

```

Results Messages

Search to filter items...

custid	custname
No results	

```

4  select * from tblnew
5  --alter table
6  ALTER Table tblnew
7  add age int
8
9  select * from [SalesLT].[Product]

```

Results Messages

Search to filter items...

id	name	age
No results		

```

28
29  --update
30  Update License
31  Set custname = 'Jack'
32  where trainerid = 102

```

Results Messages

Search to filter items...

custid	custname	trainerid	country	agelimit
1	Robin	101	India	18
2	Jack	102	USA	18

SQL – Lab

- Foreign Key Constraints

```
-- \
32 orderid int primary key,
33  customerid int constraint fkcon references License(custid),
34  [status] varchar(20)
35  )
36
37  insert into OrderDetail values (101, 2, 'ordered')
38
```

Results Messages

🔍 Search to filter items...

orderid	customerid	status
101	2	ordered

```

1  create table Doc
2  (
3  docid int primary key,
4  docname varchar(20) not null,
5  spec varchar(60) constraint defspec default 'General Physician',
6  addr varchar(40)
7  )
8
9  Insert into Doc values
10 (101,'Peter','Oncologist','Lal Bharti'),
11 (102,'Girish','Cardiologist','New Airport'),
12 (103,'Ali','Dermatologist','Kohefiza'),
13 (104,'Alice','Gynaecologist','Whitefield'),
14 (105,'Hazel','General Physician','Link Road');
15
16 select * from Doc

```

docid	docname	spec	addr
101	Peter	Oncologist	Lal Bharti
102	Girish	Cardiologist	New Airport
103	Ali	Dermatologist	Kohefiza
104	Alice	Gynaecologist	Whitefield
105	Hazel	General Physician	Link Road

```

create table Pat
(
PID varchar(10) primary key,
PName varchar(20) not null,
PContact int,
DoA varchar(20) )

Insert into Pat values
('P01','Robin',9876578,'31/08/2023'),
('P02','Sarah',9875437,'25/08/2023'),
('P03','Alex',7865439,'19/08/2023');


```

```
select * from Pat
```

PID	PName	PContact	DoA
P01	Robin	9876578	31/08/2023
P02	Sarah	9875437	25/08/2023
P03	Alex	7865439	19/08/2023

```
32  Alter Table Pat
33  ADD Treatment_Type varchar(255);
```

Results Messages

 Search to filter items...

PID	PName	PContact	DoA	Treatment_Type
P01	Robin	9876578	31/08/2023	
P02	Sarah	9875437	25/08/2023	
P03	Alex	7865439	19/08/2023	

```
create table Test
(
PatID varchar(10) constraint fkpid REFERENCES Pat(PID),
Reports varchar(30) not null)
```

```
Insert into Test values
('P01','Complete Blood Report'),
('P02','Blood Sugar'),
('P03','T1,T2,TSH');
```

PatID	Reports
P01	Complete Blood Report
P02	Blood Sugar
P03	T1,T2,TSH

SQL – Lab

- Foreign Key Constraints

```
-- \
32 orderid int primary key,
33  customerid int constraint fkcon references License(custid),
34  [status] varchar(20)
35  )
36
37  insert into OrderDetail values (101, 2, 'ordered')
38
```

Results Messages

🔍 Search to filter items...

orderid	customerid	status
101	2	ordered

DAY 4

01/09/23

+

•

○

SQL – Lab

+

- Joins and Subqueries

-> Inner Join – the value of the connecting column must match, i.e. gives only the matching data

-> Outer Join – 1. Left Outer Join – all values from left table + matching values from right table + unmatched values will be denoted as NULL

2. Right Outer Join – all values from right table + matching values from left table + unmatched values will be denoted as NULL

3. Full Outer Join – all information will return + unmatched values will be denoted as NULL

-> Cross Join – Table 1 X Table 2 (5X10=50 records will return)

-> Self Join

- Syntax

Select

From Table1 JOIN Table2

ON <predicate>;

SQL – Lab

- Inner Join

```
SELECT Pat.PID, Test.Reports
FROM Pat
INNER JOIN Test ON Pat.PID = Test.PatID;
```

PID	Reports
P01	Complete Blood Report
P02	Blood Sugar
P03	T1,T2,TSH

- Left Outer Join

```
SELECT Pat.PID, Test.Reports
FROM Pat
LEFT JOIN Test ON Pat.PID = Test.PatID
```

PID	Reports
P01	Complete Blood Report
P02	Blood Sugar
P03	T1,T2,TSH
P04	<i>null</i>

- Right Outer Join

```
SELECT Pat.PID, Test.Reports
FROM Pat
RIGHT JOIN Test ON Pat.PID = Test.PatID
```

PID	Reports
P01	Complete Blood Report
P02	Blood Sugar
P03	T1,T2,TSH

- Full Outer Join

```
SELECT Pat.PID, Test.Reports
FROM Pat
FULL OUTER JOIN Test ON Pat.PID = Test.PatID
```

PID	Reports
P01	Complete Blood Report
P02	Blood Sugar
P03	T1,T2,TSH
P04	<i>null</i>



SQL – Lab

- Cross Join

```
SELECT Pat.PID, Pat.DoA, Doc.docname, Doc.spec
FROM Pat
CROSS JOIN Doc
```

Number of Records: 20

PID	DoA	docname	spec
P01	31/08/2023	Peter	Oncologist
P01	31/08/2023	Girish	Cardiologist
P01	31/08/2023	Ali	Dermatologist
P01	31/08/2023	Alice	Gynaecologist
P01	31/08/2023	Hazel	General Physician
P02	25/08/2023	Peter	Oncologist
P02	25/08/2023	Girish	Cardiologist
P02	25/08/2023	Ali	Dermatologist
P02	25/08/2023	Alice	Gynaecologist
P02	25/08/2023	Hazel	General Physician

- Self Join

```
SELECT Doc1.docname AS Doctor, Doc2.spec AS Specialization
FROM Doc AS Doc1 JOIN Doc AS Doc2 ON Doc1.docid=Doc2.docid;
```

Doctor	Specialization
Peter	Oncologist
Girish	Cardiologist
Ali	Dermatologist
Alice	Gynaecologist
Hazel	General Physician



SQL – Lab

- Scalar Subquery

```
Select PName, PID, Treatment_Type
from Pat
WHERE
PID=(SELECT PID from Pat where PID='P03');
```

PName	PID	Treatment_Type
Alex	P03	Kidney Transplant

- Multi-valued Subquery

```
Select PName
from Pat
WHERE PID IN (SELECT PID from Pat
where PID='P03' AND DoA='19/08/2023');
```

PName
Alex

- Correlated Subquery

```
Select PName,PID
from Pat AS p1
WHERE PID =
(SELECT MAX(PID)
FROM Pat AS p2
WHERE p2.PID=p1.PID)
ORDER BY DoA;
```

PName	PID
Adam	P04
Alex	P03
Sarah	P02
Robin	P01



SQL – Lab

- Returning Results from Views
-> First Line of Security



SQL – Lab



- Executing Stored Procedures
 - > Batch Statement – run more than one query in a statement
- Stored Procedure – A collection of T-SQL statements giving the capability to modify the data dynamically
 - Must have an execute permission (EXEC)
- Built-In Functions
 - Scalar Function – returns a single value as output
 - Aggregate Functions – used for summarizing the data (DistinctCount – non duplicate values, Sum, Max, Min, Count, Avg)
 - Window Functions – raking functions (Rowset, Rank, Dense_Rank)

SQL – Lab

- Indexes – row based & column based
- Four levels while working with indexes :
->Root Level ->Intermediate Level ->Leaf Level ->Data Level
- Rules
 1. Value \leq Required value & nearest to that
 2. Navigate to the next page/pointer
- Clustered Index – Stores the data in a sorted format and enhances the speed of searching (leaf and data page are one single page)
- Non-Clustered Index – leaf and the data level gets separated
- ColumnStore Index- concept of compression and enhances performance, only stores columns, 2 types
 1. Cluster – takes all columns, at least 104800 data
 2. Non-cluster – choice to select columns on which to apply indexes

