

INTERNATIONAL ADVANCED LEVEL

INFORMATION TECHNOLOGY

UNIT 2

UNIT 3: Topic 12: Manipulating data



Academic Year: 2024-2025

✍ Prepared by: Irikefe Hope Oniovosa

1. DATA INTEGRITY

A. Importance & Governance

1. Why Data Integrity Matters

- **Accuracy:** Decisions based on incorrect data can lead to financial loss or safety risks.
- **Validity:** Ensures data meets business rules (e.g., dates, IDs).
- **Relevance:** Keeps datasets focused on the information you truly need.

2. Data Governance Frameworks

- **Policies & Standards:** Define who owns data, who may edit it, and how it must be maintained.
- **Roles:** Data Stewards (maintain data quality), Data Custodians (manage technical storage), Data Users (consume data responsibly).
- **Processes:** Regular audits, change management, and issue-resolution workflows.

B. Data Dictionary

1. Definition & Role

- A **data dictionary** is a centralized repository of metadata: it describes each data element in your system so everyone understands its meaning, format, and usage.

2. Typical Fields in a Dictionary Entry

Field Name	Data Type	Length	Description
StudentID	VARCHAR	8	Unique identifier, format YYYYNNNN
FirstName	VARCHAR	50	Student's given name
DateOfBirth	DATE	—	In ISO format, YYYY-MM-DD
Email	VARCHAR	100	Must follow standard email format

1. Why It Helps

- **Clarity:** New team members understand each field's purpose.
- **Consistency:** Prevents different developers using different formats.
- **Documentation:** Acts as living documentation for your database.

C. Construction & Interpretation Exercise

1. Table Schema Provided

Table: Courses

- CourseCode CHAR(6)
- Title VARCHAR(100)
- Credits INT
- StartDate DATE
- IsActive CHAR(1)

2. Hands-On Task

In pairs, build data dictionary entries for each field:

- **CourseCode:**
 - Type: CHAR
 - Length: 6
 - Description: Alphanumeric code, format AAA999 (three letters, three digits)
- **Title:** ...
- **Credits:** ...
- **StartDate:** ...
- **IsActive:** ...

3. Share & Discuss:

Compare entries and refine descriptions to be precise and unambiguous.

D. Validation Rules

1. Rule Types & Examples

Rule Type	Definition	Example for Student Registration
Presence	Field must not be empty	StudentID cannot be null
Range	Numeric/date within a specified range	Age between 16 and 100
Lookup	Value must exist in another table	CourseCode must match an existing course

Rule Type	Definition	Example for Student Registration
List	Value must be one of a predefined set	Gender $\in \{\text{"M"}, \text{"F"}, \text{"O"}\}$
Length	Character length within limits	Password length between 8 and 16 characters
Format	Pattern matching (e.g., regex)	Email must match $^{\wedge}[\backslash w.\%+-]+\text{@}[\backslash w.-]+\backslash.[A-Za-z]\{2,\}\$$
Check Digit	Algorithmic checksum (e.g., ISBN, credit card)	StudentID's last digit validates mod 11

2.0 Exercise: Designing Validation Rules

- **Scenario:** A student registration form with fields: StudentID, FirstName, LastName, Email, Age, MajorCode.
- **Task:** In small groups, draft validation rules for each field:
 - **StudentID:** Presence, format AAA999, check digit.
 - **Email:** Presence, format regex.
 - **Age:** Presence, range 16–100.
 - **MajorCode:** Presence, lookup against Majors table.
- **Presentation:** Each group shares their rule set; discuss potential edge cases and how to enforce these rules technically (in code or database constraints).

A. Data Redundancy & Its Problems

Unnormalized Table Example

- **Redundancy:** Alice's data appears twice.

OrderID	OrderDate	CustomerID	CustomerName	CustomerEmail	ProductID	ProductName	Quantity
1001	2025-04-01	C001	Alice Smith	alice@example.co m	P10	Widget A	2
1002	2025-04-02	C002	Bob Jones	bob@example.co m	P12	Widget B	1
1003	2025-04-02	C001	Alice Smith	alice@example.co m	P11	Widget C	5

-
- **Update Anomaly:** Changing Alice's email requires updating **both** rows.
- **Insertion/Deletion Anomalies:**
 - Cannot add a new customer without an order.
 - Deleting Alice's only order would remove her customer record entirely.

B. Normalization Concepts

1. First Normal Form (1NF)

- **Rule:** Eliminate repeating groups; each field contains only atomic (single) values.
- **Apply to Example:** Ensure `ProductID`, `Quantity` are separate rows rather than comma-separated lists.

2. Second Normal Form (2NF)

- **Rule:** Meet 1NF **and** remove partial dependencies—i.e., every non-key attribute must depend on the **whole** primary key.
- **Example:** In an `OrderItems` table with composite primary key (`OrderID`, `ProductID`), move `Customer` fields out since they depend only on `OrderID`.

3. Third Normal Form (3NF)

- **Rule:** Meet 2NF **and** remove transitive dependencies—i.e., non-key attributes depend only on the primary key, not on other non-key attributes.

- **Example:** If `CustomerEmail` depends on `CustomerID`, it belongs in a separate `Customers` table, not in `Orders`.

C. Normalization Exercise

Unnormalized Dataset

```
OrderID | OrderDate | CustomerID | CustomerName | CustomerEmail | ProductID |
ProductName | Quantity
```

Step 1: 1NF

- Break out each product as its own row (already atomic).

Step 2: 2NF

- Create three tables:

Orders

OrderID (PK)

OrderDate

CustomerID (FK)

OrderItems

OrderID (PK, FK)

ProductID (PK, FK)

Quantity

Customers

CustomerID (PK)

CustomerName

CustomerEmail

Step 3: 3NF

- Products in their own table:

Products

ProductID (PK)

ProductName

- Final schema has no partial or transitive dependencies.

D. Logical Data Modeling

- **Tables & Keys:**
 - **Primary Key (PK):** Unique identifier (e.g., `OrderID`, `CustomerID`).
 - **Foreign Key (FK):** References a PK in another table (e.g., `CustomerID` in `Orders`).
- **Relationship Types:**
 - **One-to-Many:** One customer $\rightarrow$ many orders (`Customers` $\rightarrow$ `Orders`).
 - **Many-to-Many:** `Orders` $\leftrightarrow$ `Products` via `OrderItems` junction table.

ERD Excerpt:

```
Customers (CustomerID PK)
|
|-< Orders (OrderID PK, CustomerID FK)
|
|-< OrderItems (OrderID FK, ProductID FK) >- Products (ProductID PK)
```

E. Group Task: ERD for a Library System

Entities & Attributes:

- **Books** (`BookID` PK, Title, Author, ISBN)
- **Members** (`MemberID` PK, Name, Email)
- **Loans** (`LoanID` PK, `BookID` FK, `MemberID` FK, `LoanDate`, `ReturnDate`)

Relationships:

- One **Book** can appear in many **Loans** (over time).
- One **Member** can have many **Loans**.
- **Loans** is a junction table modeling the many-to-many time-based interaction.

Task:

In small groups, draw an ERD showing tables, PK/FK, and relationships. Label cardinalities (1:N).
Review and discuss each group's design.

3. BIG DATA

A. Characteristics (the “5 Vs”)

1. Volume

- **Definition:** Massive amounts of data (terabytes to exabytes).
- **Example:** Social media platforms generate hundreds of petabytes of posts, images, and videos annually.

2. Velocity

- **Definition:** The speed at which data is generated and must be processed.
- **Example:** Financial markets produce millions of trade records per second that require real-time analysis.

3. Variety

- **Definition:** Different formats and sources—structured, semi-structured, unstructured.
- **Example:** IoT sensors (numeric readings), server logs (text), camera footage (video), social media feeds (text/images).

4. Veracity

- **Definition:** Trustworthiness and quality of data.
- **Example:** User-generated content can include noise or false information; sensor data may be incomplete or faulty.

5. Value

- **Definition:** The actionable insights and business benefits derived from data.
- **Example:** Analyzing customer purchase history to recommend new products, boosting sales and loyalty.

B. Infrastructure

1. Collection Mechanisms

- **Sensors & IoT Devices:** Temperature, motion, biometric sensors.
- **Logs & Clickstreams:** Web server logs, application logs recording user interactions.

2. Storage Solutions

- **Hadoop Distributed File System (HDFS):** Handles large volumes of data across clusters.

- **NoSQL Databases:** Document stores (e.g., MongoDB), wide-column stores (e.g., Cassandra) for flexible schemas.

3. Transmission & Streaming

- **Platforms:** Apache Kafka, AWS Kinesis for ingesting and distributing high-velocity data streams.
- **Pipelines:** Real-time ETL (Extract-Transform-Load) moving data from source to analytics layer.

C. Impact Analysis

- **Data Access Latency:** Slow queries on massive datasets can hinder timely insights.
- **Processing Time:** Batch vs. real-time processing trade-offs (e.g., Spark Streaming vs. traditional MapReduce).
- **Network Bandwidth:** Moving petabytes between data centers requires robust network infrastructure.
- **Security & Privacy:** Encrypting data at rest/in transit; managing access controls and compliance (e.g., GDPR).

D. Core Data Concepts

1. Data Mining

- **Purpose:** Discover patterns and relationships (e.g., market basket analysis revealing that customers who buy bread also buy butter).

2. Data Warehousing

- **Purpose:** Centralized, structured repositories combining data from multiple sources for reporting and analysis (e.g., enterprise data warehouse).

3. Data Analytics

- **Descriptive Analytics:** Reporting on “what happened” (e.g., daily sales dashboards).
- **Predictive Analytics:** Forecasting “what might happen” (e.g., predicting equipment failure).

- **Prescriptive Analytics:** Advising “what should be done” (e.g., recommending optimal inventory levels).

E. Use Cases

1. Healthcare: Predictive Diagnostics

- Using patient history and sensor data to forecast disease onset (e.g., predicting heart-attack risk).

2. Smart Cities: Traffic Optimization

- Integrating real-time vehicle, weather, and event data to adjust traffic signals and reduce congestion.

3. Transportation: Route Planning

- Leveraging GPS, traffic, and weather data to calculate fastest or fuel-efficient routes for logistics fleets.

4. Fraud Detection

- Analyzing transaction patterns in real time to flag anomalies (e.g., unusual credit-card spending indicative of fraud).

F. Wrap-Up & Discussion

• Q&A:

- “Which of the 5 Vs do you think poses the biggest technical challenge?”
- “How does a data warehouse differ from a Hadoop cluster?”
- “What type of analytics would you use to recommend products to customers?”

ASSESSMENT

Formative

- **Quiz on Integrity & Normalization:**

Short multiple-choice or true/false questions covering data governance roles, data dictionary purposes, and definitions of 1NF/2NF/3NF.

- **Quick Build:**

Given a scenario (e.g., a membership form), write **three** validation rules (e.g., presence, format, range).

- **Peer Review:**

Swap examples of unnormalized vs. normalized tables and critique each other's designs and normalization steps.

Summative

- **Database Project:**

Design a small relational schema (minimum three tables), produce a data dictionary for each field, specify validation rules, and show tables in 3NF.

- **Big Data Case Study Report:**

Select a real-world Big Data application (e.g., predictive healthcare), and describe:

- The **5 Vs** of its data
- Its underpinning **infrastructure** (collection, storage, streaming)
- The kinds of **analytics** performed (descriptive, predictive, prescriptive)

ADDITIONAL RESOURCES

Videos

1. **Data Governance Fundamentals**

▶ [Data Governance Fundamentals – Natalie Greenwood YouTube](#)

2. **Database Normalization, Explained**

▶ [Data Normalization, Explained YouTube](#)

3. **Introduction to Big Data**

▶ [Big Data In 5 Minutes | What Is Big Data? YouTube](#)

Websites

- **Dataedo (Data Dictionary Tools):**

🌐 <https://dataedo.com>

- **W3Schools SQL Normalization Tutorial:**

🌐 <https://www.w3schools.com/sql/sql-normalization.asp>

- **AWS Big Data Services Documentation:**

🌐 <https://aws.amazon.com/big-data/>

Reading Material

- Articles on best practices in data validation (e.g., validating form inputs and database constraints)
- Textbook chapters on relational database design and normalization
- Whitepapers on Big Data architectures and analytics methodologies (e.g., from O'Reilly or IEEE)