|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Introduction to databases  Week 1 – 2 |
| |  |  |  | | --- | --- | --- | | brunkonjaa | 9/30/25 | ITD | |

**What is Normalisation?**

* The process of splitting up data into properly organised tables.
* Goal: make data **clear, consistent, and efficient**.
* Prevents anomalies (insert, update, delete mistakes).
* Saves space and avoids duplication.
* Based on Edgar Codd’s principle: *“The key, the whole key, and nothing but the key.”*

**The Normal Forms**

* **UNF (Unnormalized Form):**
  + All data in one messy table, often with repeats.
* **1NF (First Normal Form):**
  + Remove repeating groups, make values atomic, add primary key.
* **2NF (Second Normal Form):**
  + Remove partial dependencies (non-key fields can’t depend on part of a composite key).
* **3NF (Third Normal Form):**
  + Remove transitive dependencies (non-key fields can’t depend on another non-key).
* **4NF (Fourth Normal Form):**
  + Remove multi-valued dependencies (e.g., multiple independent lists in one table).

**Examples:**

1. **DVD Rental Store**

* **UNF:** Members + DVDs + Rentals in one big table
* **1NF:** Members table | DVDs table
* **2NF:** Members table | DVDs table | Rentals table (links Members ↔ DVDs with rental dates)
* **3NF:** Members table | DVDs table (Certificate → CertificateCode) | Rentals table | Certificates table (age ratings stored separately)
* **UNF**: messy, same member/DVD written over and over.
* **1NF**: separate lists stop duplicate names and titles.
* **2NF**: Rentals become their own thing → clean record of who rented what and when.
* **3NF**: Certificates pulled out → avoids repeating “U” or “15+” everywhere.

Result = each fact stored once, no confusion.

1. **Games Store (to 4NF)**

* **UNF:** Customers + Items + Suppliers + Newsletters in one big table
* **1NF:** Customers table | Items table
* **2NF:** Customers table | Items table | Sales table (junction for purchases)
* **3NF:** Customers table | Items table | Sales table | Suppliers table
* **4NF:** Customers table | Items table | Sales table | Suppliers table | Subscriptions table (newsletter split off)
* **UNF**: customers, products, suppliers, and even newsletters jumbled together.
* **1NF**: split into clear lists (Customers, Items).
* **2NF**: sales are their own thing (who bought what).
* **3NF**: suppliers split out, so their details aren’t repeated.
* **4NF**: newsletters separated, since they’re independent of purchases.

Each “theme” gets its own table, no duplication across them.

1. **Trainee / Module Example**

* **UNF:** Trainees + Modules + Results in one table
* **1NF:** Trainees table | Modules table (linked with composite key TraineeNo+ModuleNo)
* **2NF:** Trainees table | Modules table | Results table
* **3NF:** Trainees table | Modules table | Results table | Levels table (remove transitive dependency)
* **UNF**: trainees, modules, and results repeated together.
* **1NF**: one row per trainee–module link.
* **2NF**: separate results and modules properly.
* **3NF**: “Level” (like Beginner, Advanced) split into its own table.

Keeps trainee info, module info, results, and levels in separate clean buckets.

1. **Vet Clinic (Exercise 1)**

* **UNF:** Pet info + Visits + Procedures in one table
* **1NF:** Pet table | Pet\_Visit table
* **2NF:** Pet table | Pet\_Visit table | Procedure table (ProcedureName depends only on ProcedureNo)
* **3NF:** Pet table | Pet\_Visit table | Procedure table (already clean, no further splits)
* **UNF**: pet info, visit info, and procedure details all repeated.
* **1NF**: separate pets from visits.
* **2NF**: procedures pulled out, so “X-Ray” or “Check-up” isn’t written every time.
* **3NF**: already tidy, nothing else depends on non-keys.

Clear split: Pets, Procedures, and Visits.

1. **Orders (Exercise 2)**

* **UNF:** Orders + Customers + Products in one form with repeats
* **1NF:** Order table | Product table
* **2NF:** Customer table | Product table | Order table | OrderLine table (junction)
* **3NF:** Customer table | Product table | Order table | OrderLine table (already clean, no extra dependencies)
* **UNF**: messy form — one order with customer and many products repeated.
* **1NF**: split products from orders.
* **2NF**: pull customers and products into their own tables, link them with OrderLine.
* **3NF**: no hidden dependencies left, everything linked through IDs.

End result: Customers, Products, Orders, and OrderLines — no duplication.

**Practical Steps (to use every time):**

1. Write the raw messy data in **UNF**.
2. Convert to **1NF**: remove repeats, ensure atomic values, add primary key.
3. Convert to **2NF**: remove partial key dependencies.
4. Convert to **3NF**: remove non-key → non-key dependencies.
5. (Optional) check for **4NF / BCNF** if there are still multi-valued or candidate key anomalies.

* **Normalisation in one line:**
  + split a messy table into smaller linked ones until every fact depends **only on the key, the whole key, and nothing but the key**.