# Database Systems Concepts and Design

CSC201S2/G2

# Chapter 4: Normalisation

# Introduction

- Normalization is the process of organizing data in a database.
- Includes creating tables and establishing relationships between those tables
- Redundancy and inconsistent dependency.

# Unnormalized form(UNF)

- A table that contains one or more repeating groups.
- An attribute or group of attributes within a table that occurs with **multiple values for a single occurrence of the nominated key attributes of that table**.
- A UNF model will suffer problems like **data redundancy** thus it lacks the efficiency of database normalisation.

# Example: Repeating Groups

Repeating groups

| CustNo | Cname | PropNo | PAddr | RntSt | RntFnsh | Rent | OwnerNo | OName |
|---|---|---|---|---|---|---|---|---|
| CR76 | John Kay | PG4 | 6 Lawrence St, Elmont | 7/1/10 | 8/31/06 | 700 | CO40 | Tina Murphy |
| | | PG16 | 5 Nova Dr, East Meadow | 9/1/06 | 9/1/08 | 900 | CO93 | Tony Shaw |
| CR56 | Aline Stewart | PG4 | 6 Lawrence St, Elmont | 9/1/02 | 6/10/04 | 700 | CO40 | Tina Murphy |
| | | PG36 | 2 Manor Rd Scarsdale | 8/1/04 | 12/1/05 | 750 | CO93 | Tony Shaw |
| | | PG16 | 5 Nova Dr, East Meadow | 8/1/06 | 9/1/10 | 900 | CO93 | Tony Shaw |

# Redundant Information

- Data redundancy occurs when the same piece of data is stored in two or more separate places.
- Aim of relational database design is to group attribute into relations to minimize data redundancy and thereby reduce the file storage space required by the implemented base relations.

# Example: Data Redundancy

| Staff_no | Sname | Position | Salary | Branch_no | Baddress |
|---|---|---|---|---|---|
| S1 | John | Manager | 10000 | B005 | 22 deccan, Pune. |
| S2 | Ann | Assistant | 5000 | B003 | 10 bandra, Mumbai. |
| S3 | Suhas | Supervisor | 7000 | B007 | 32 Main st, Nasik. |
| S4 | Julie | Assistant | 5000 | B007 | 32 Main st, Nasik. |
| S5 | Mary | Assistant | 5000 | B005 | 22 deccan, Pune. |

- In **Staffbranch** relation there is redundant data. Branch address is repeated for every member of staff located at that branch.

# Update Anomalies

- Relations that have redundant data may have problems called **update anomalies**.
- Type of update anomalies are:
  - Insertion
  - Deletion
  - Modification

# Example: Update Anomalies

**EMP_PROJ** ( *Emp*#, Proj#, Ename, Pname, No_hours)

**Modification Anomaly**

- Changing the project name of project number P1 from "Billing" to "Customer-Accounting" may cause this update to be made for all 100 employees working on project P1

# Example: Insert Anomalies

**EMP_PROJ** ( *Emp*#, Proj#, Ename, Pname, No_hours)

- Cannot insert a project unless an employee is assigned to .
- Inversely- Cannot insert an employee unless he/she is assigned to a project.

# Example: Delete Anomaly

**EMP_PROJ** ( *Emp*#, Proj#, Ename, Pname, No_hours)

When a project is deleted, it will result in deleting all the employees who work on that project. Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

# Example: Update Anomalies

- Insert a new staff into the StaffBranch relation;
- Delete a tuple that represents the last member of staff located at a branch B007;
- Change the address of branch B003.

| staffNo | sName | position | salary | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 | 22 Deer Rd, London |
| SG37 | Ann Beech | Assistant | 12000 | B003 | 163 Main St,Glasgow |
| SG14 | David Ford | Supervisor | 18000 | B003 | 163 Main St,Glasgow |
| SA9 | Mary Howe | Assistant | 9000 | B007 | 16 Argyll St,Aberdeen |
| SG5 | Susan Brand | Manager | 24000 | B003 | 163 Main St,Glasgow |
| SL41 | Julie Lee | Assistant | 9000 | B005 | 22 Deer Rd, London |

# Normalisation

- Database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
- There are three main reasons to normalize a database.
    - minimize duplicate data,
    - minimize or avoid update anomalies
    - simplify queries.

# Normalisation

A process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Steps to normalize the database:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Code Normal Form (BCNF)

# Types of Dependencies

Dependencies in DBMS is a relation between two or more attributes:

- Functional Dependency
- Fully-Functional Dependency
- Partial Dependency
- Transitive Dependency

# Functional Dependencies

If the information stored in a table can uniquely determine another information in the same table, then it is called **Functional Dependency**.

If A and B are attributes of a relation R, B is **functionally dependent** on A (A → B), if **each value of A in R is associated with exactly one value of B in R**.

# Functional Dependencies

If the information stored in a table can uniquely determine another information in the same table, then it is called **Functional Dependency**.

If A and B are attributes of a relation R, B is **functionally dependent** on A (A → B), if **each value of A in R is associated with exactly one value of B in R**.

# Example: Functional Dependencies

**STUDENT**

| STUD_NO | STUD_NAME | STUD_PHONE | STUD_STATE | STUD_COUNTRY | STUD_AGE |
|---|---|---|---|---|---|
| 1 | RAM | 9716271721 | Haryana | India | 20 |
| 2 | RAM | 9898291281 | Punjab | India | 19 |
| 3 | SUJIT | 7898291981 | Rajsthan | India | 18 |
| 4 | SURESH | | Punjab | India | 21 |

STUD_NO -> STUD_NAME and

STUD_NO -> STUD_PHONE hold

*A **STUD_NO** uniquely identifies a **STUD_NAME** and **STUD_PHONE**

STUD_NAME->STUD_STATE does not hold

***Two students can have same name (Like RAM in the below table) and hence same state**

# Full-functional Dependencies

Full functional dependency indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, **but not on any proper subset of** A.

# Example

| supplier_id | item_id | price |
|---|---|---|
| 1 | 1 | 540 |
| 2 | 1 | 545 |
| 1 | 2 | 200 |
| 2 | 2 | 201 |
| 1 | 1 | 540 |
| 2 | 2 | 201 |
| 3 | 1 | 542 |

{supplier_id, item_id} -> price

*supplier_id  nor item_id can uniquely determine the price

*Both supplier_id and item_id together can do so

***Price full-functional depend on** supplier_id and item_id

# Partial Dependencies

A functional dependency P→Q is partially dependent if there is some attributes that can be removed from P and the dependency still holds.

# Example

| name | roll_no | course |
|---|---|---|
| Ravi | 2 | DBMS |
| Tim | 3 | OS |
| John | 5 | Java |

{name} -> course

{roll_no} -> course

{name, roll_no} -> course

*Both the attributes name and roll_no alone are able to uniquely identify a course

*The relationship is partially dependent

# Transitive Dependencies

When an **indirect relationship causes functional dependency** it is called **Transitive Dependency**.

Attribute B depends on attribute A (A -> B) and C depends on B (B -> C), indirectly establishing a dependency between A and C (A -> C).

# Example

| roll_no | name | city | zip-code |
|---|---|---|---|
| 1 | abc | pune | 411044 |
| 2 | jkl | mumbai | 400001 |
| 3 | uvw | pune | 411044 |
| 4 | xyz | delhi | 110001 |

{roll_no} -> city

{city} -> zip-code

{roll_no} -> zip_code

*roll-no = 1 has city=pune and city=pune will have zip-code=411044. So wherever roll-no is 1 , zip-code will be 411044

*The relationship is transitive dependency

# Exercise

| A | B | C |
|---|---|---|
| 7 | 1 | 8 |
| 7 | 2 | 5 |
| 7 | 3 | 5 |
| 5 | 8 | 8 |

Which functional dependencies holds relation

1. AB →C && C→ B
2. BC → A && B→C
3. BC →A && A→ C
4. AC→ B && B→C

| v | w | x | y | z |
|---|---|---|---|---|
| 7 | 8 | c | 9 | 4 |
| 8 | 7 | c | 9 | 4 |
| 7 | 8 | c | 2 | 4 |
| 7 | 8 | c | 2 | 2 |

Which functional dependencies holds relation

R(v, w, x, y, z)

1. v → wx
2. yz → x
3. x→ yz

# Exercise

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_2$ | $b_1$ | $c_3$ |

List all functional dependencies satisfied by the relation

R (A, B, C)

| α | β | γ |
|---|---|---|
| 6 | 1 | 7 |
| 4 | 3 | 7 |
| 4 | 8 | 1 |
| 6 | 7 | 1 |

Which functional dependencies are holds the relation

1. $\propto \rightarrow \beta$ && $\beta\gamma \rightarrow \propto$
2. $\gamma \rightarrow \beta$ && $\propto \rightarrow \beta$
3. $\beta \rightarrow \gamma$ && $\propto \beta \rightarrow \gamma$
4. $\propto \rightarrow \gamma$ && $\beta\gamma \rightarrow \propto$


Q
&
A

# First Normal Form (INF)

- All data values are atomic.
- INF is a relation in which the intersection of each row and column contains **one and only one value**.

***Approach for removing repeating groups:***

- Entering appropriate data in the empty columns of rows containing the repeating data.
- Placing the repeating data, along with a copy of the original key attribute(s), in a separate relation.
- A primary key is identified for the new relation.