# EE4202 Database Systems

## NORMALIZATION

# CONSTRAINT VIOLATIONS

- Whenever a database is modified, the relational database constraints must not be violated.
- 3 modifications exist as,
  - ❑ Insert – A new tuple is inserted into the relation
  - ❑ Update – An existing tuple is modified.
  - ❑ Delete – An existing tuple is removed.
- Schema based constraints which can be violated in each operation are,
  - ❑ Insert: Domain constraint, key constraint, null constraint, Entity integrity, Referential integrity.
  - ❑ Update: Domain constraint, key constraint, null constraint ,Entity integrity, Referential integrity.
  - ❑ Delete: Referential Integrity.
- SQL does not allow schema based constraints to violate in all insert, update and delete operations. But, data anomalies such as loss of data, having null values can occur due to referential integrity. SQL ensures that the database stays at least in 1NF.
- But, data constraints such as partial functional dependencies and transitive dependencies can be violated in all insert, update, delete operations causing data anomalies. There is no way to prevent data anomalies unless the database is normalized.

# Schema based constraint violations, Data Anomalies

```
UPDATE EMPLOYEE
SET Age = 'seven’
WHERE Empl_ID = 'EE001';
```

Domain Constraint Violation

```
INSERT INTO EMPLOYEE (Empl_ID, Dept_ID, Designation, Age)
VALUES (‘CE013', 'CEE', 'Lecturer’, ‘Eight’);
```

```
UPDATE EMPLOYEE
SET  Empl_ID = 'EE001’
WHERE Empl_ID = 'EE002';
```

Key Constraint Violation

```
DELETE FROM DEPARTMENT
WHERE Dept_ID = "CEE";
```

```
UPDATE DEPARTMENT
SET  Dept_ID = "EEE"
WHERE Dept_ID = "MME";
```

Referential Integrity Constraint Violation

# NORMALIZATION

Aims of Normalization are ;

1. To **reduce** insert, update, deletion anomalies; that is to reduce schema constraint violations during those operations.

   Ex: a) Reduce data anomalie during insert, update, delete operations in BCNF by not allowing multiple candidate keys, partial functional dependencies, transitive dependencies within a relation. b) Even if anomalies occur due to referential integrity; only few data is affected as data is divided into different tables.

2. To minimize redundancy (repetition of data) thus efficiently storing data – Ex: in 4NF; non-trivial multivalued dependencies are not allowed.

- Normalization is informally goodness of relational design. Higher the normal form; higher is the goodness.
- A normal form must meet certain conditions.
- During normalization, a given relation is broken down into multiple relations based on multiple factors such as data dependencies and schema-based constraints.

# NORMAL FORMS

Following Normal Forms will be studied in this context.

- First Normal Form → 1NF
- Second Normal Form → 2NF
- Third Normal Form → 3NF
- Boyce-Codd Normal Form → BCNF (An extension of third normal form)
- Fourth Normal Form → 4NF

# FIRST NORMAL FORM – 1NF

- **Rule:** Any relation must be at least in 1NF in a relational database.
- Conditions required for first normal form are,
  - Domain constraint – Every database element must be atomic and single valued within its domain.
  - That is multivalued attributes and composite attributes are forbidden in first normal form.
  - Key constraint - Primary key of the relation is identified out of multiple candidate keys.
- If there are multivalued attributes, they must be taken into a separate relation along with the primary key of the original relation. In the resulting relation containing the multivalued attribute, the combination of attributes become primary key.

Ex: The table VEHICLE (<u>Vehicle_ID</u>, {Color}, Model) is not in 1NF.

**VEHICLE**

| <u>Vehicle_ID</u> | Color | Model |
|---|---|---|
| VH001 | Red<br>Green | Nissan |
| VH002 | Black<br>White<br>Grey | Suzuki |

X 1NF

Normalization →

**VEHICLE_COLOR**

| <u>Vehicle_ID</u> | <u>Color</u> |
|---|---|
| VH001 | Red |
| VH001 | Green |
| VH002 | Black |
| VH002 | White |
| VH002 | Grey |

✓ 1NF

**VEHICLE_MODEL**

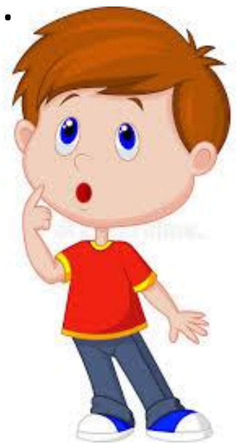
| <u>Vehicle_ID</u> | Model |
|---|---|
| VH001 | Nissan |
| VH002 | Suzuki |

✓ 1NF

# FIRST NORMAL FORM – 1NF

- If there is a composite attribute, they should be broken into atomic attributes in the same relation.

  Ex: The table STUDENT (Student_ID, name, Address (Street, city, country)) is not in 1NF.



**STUDENT**

| <u>Student_ID</u> | name | Address |
|---|---|---|
| EG_3310 | Vishwajith | 55/2<br>Galle<br>Sri Lanka |
| EG_3290 | Caroline | Baker Street<br>London<br>England |

X **1NF**

Normalization

**STUDENT**

| <u>Student_ID</u> | name | Street | City | Country |
|---|---|---|---|---|
| EG_3310 | Vishwajith | 55/2 | Galle | Sri Lanka |
| EG_3290 | Caroline | Baker Street | London | England |

✓ **1NF**

# SECOND NORMAL FORM – 2NF

- Conditions required for a relation to be in Second Normal Form (2NF) are,
  - ❑ Be in first normal form
  - ❑ Every non-prime attribute must be **only fully functionally dependent** on any prime attribute/atrributes. That is every non-prime attribute is not partially functional dependent on any key (composite key).

Note: If there is only one attribute as the candidate key and that attribute is primary key, then table is automatically in 2NF.

Ex: The relation STUDENT(Stu_ID, Course_ID, Stu_name, Course_name, Project) is in 1NF.

STUDENT COURSE

| Stu_ID | Course_ID | Stu_name | Course_name | Project |
|---|---|---|---|---|
| EG_001 | EE4202 | Gihan | Database | Airport |
| EG_001 | EE4302 | Gihan | Machines | NULL |
| EG_002 | EE3201 | Tony | Electronic prj | NULL |
| EG_003 | EE3201 | Akalanka | Electronic prj | Fire Alarm |

✓ 1NF X 2NF

Normalization

STUDENT

| Student_ID | Stu_name |
|---|---|
| EG_001 | Gihan |
| EG_002 | Tony |
| EG_003 | Akalanka |

✓ 1NF
✓ 2NF

COURSE

| Course_ID | Course_name |
|---|---|
| EE4202 | Database |
| EE4302 | Machines |
| EE3201 | Electronic_prj |

✓ 1NF
✓ 2NF

PROJECT

| Stu_ID | Course_ID | Project |
|---|---|---|
| EG_001 | EE4202 | Airport |
| EG_001 | EE4302 | NULL |
| EG_002 | EE3201 | NULL |
| EG_003 | EE3201 | Fire Alarm |

✓ 1NF
✓ 2NF

# THIRD NORMAL FORM – 3NF

- Conditions required for a relation to be in Third Normal Form (3NF) are,
  - ❑ Be in Second normal form.
  - ❑ Every non-prime attribute of the relation must be non-transitively dependent on every key.

Ex: The relation EMPLOYEE (Empl_ID, Dept_ID, Area, Designation, Dept_Head) is in 2NF.

**EMPLOYEE**

| Empl_ID | Dept_ID | Area | Designation | Dept_Head |
|---|---|---|---|---|
| EE001 | EIE | Software | Lecturer | Sajitha |
| EE002 | EIE | Software | Senior Lecturer | Sajitha |
| MM007 | MME | Automobile | Lecturer | Randitha |
| MM001 | MME | Manufacturing | Senior Lecturer | Randitha |
| CE001 | CEE | Hydraulic | Senior Lecturer | Layanthi |
| CE003 | CEE | Structural | Lecturer | Layanthi |
| CE007 | CEE | Structural | Senior Lecturer | Layanthi |
| EE005 | EIE | Electronics | Lecturer | Sajitha |

✓ 1NF ✓ 2NF X 3NF

Normalization

| Dept_ID | Dept_Head |
|---|---|
| EIE | Sajitha |
| MME | Randitha |
| CEE | Layanthi |

✓ 1NF
✓ 2NF
✓ 3NF

| Empl_ID | Dept_ID | Designation |
|---|---|---|
| EE001 | EIE | Lecturer |
| EE002 | EIE | Senior Lecturer |
| MM007 | MME | Lecturer |
| MM001 | MME | Senior Lecturer |
| CE001 | CEE | Senior Lecturer |
| CE003 | CEE | Lecturer |
| CE007 | CEE | Lecturer |
| EE005 | EIE | Lecturer |

| Area | Dept_ID |
|---|---|
| Software | EIE |
| Automobile | MME |
| Manufacturing | MME |
| Hydraulic | CEE |
| Structural | CEE |
| Electronics | EIE |

✓ 1NF ✓ 2NF ✓ 3NF

# BOYCE CODD NORMAL FORM – BCNF

- Conditions required for a relation to be in Boyce-Codd Normal Form (BCNF) are,
  - ❑ Be in third normal form.
  - ❑ For all the nontrivial functional dependencies X ⟶ Y, X must be a superkey and Y must be a non-prime attribute (Multiple candidate keys are not allowed within a relation).

If **X → Y** and **Y is not a subset of X**, then it is called Non-trivial functional dependency.

Ex: The relation EMPLOYEE (Department, Empl_Bday, Empl_name, city, UPF_no,Expert_Area) is in 3NF.

**EMPLOYEE**

Normalization

| Department | Empl_Bday | Empl_name | city | UPF_no | Expert_Area |
|---|---|---|---|---|---|
| Electrical | 28/03/1995 | Wimal | Negombo | R07651 | Electronics |
| Electrical | 13/08/1990 | Wimal | Kandy | R05623 | Electronics |
| Electrical | 13/08/1990 | Piyal | Ampara | R03265 | Power |
| Civil | 23/03/1993 | Sunil | Kandy | R01410 | Environment |

UPF_no is a candidate key.

✓ **1NF** ✓ **2NF** ✓ **3NF** **X BCNF**

| Department | Empl_Bday | Empl_name | city |
|---|---|---|---|
| Electrical | 28/03/1995 | Wimal | Negombo |
| Electrical | 13/08/1990 | Wimal | Kandy |
| Electrical | 13/08/1990 | Piyal | Ampara |
| Civil | 23/03/1993 | Sunil | Kandy |

| Expert_Area | UPF_no |
|---|---|
| Electronics | R07651 |
| Electronics | R05623 |
| Power | R03265 |
| Environment | R01410 |

**HEAD**

✓ **1NF**

✓ **2NF**

✓ **3NF**

✓ **BCNF**

| Expert_Area | Department |
|---|---|
| Electronics | Electrical |
| Power | Electrical |
| Environment | Civil |

# FOURTH NORMAL FORM – 4NF

- Conditions required for a relation to be in Fourth Normal Form (4NF) are,
  - ❑ Be in Boyce Codd Normal Form.
  - ❑ It can consist of only Trivial multi-valued dependencies.

Rule: An all key relation is always in BCNF.

Ex: The relation EMPLOYEE (Emp_name, Proj_name, Dep_name) is in BCNF.

**EMPLOYEE**

X **non-trivial Multivalued Dependency**

| Emp_name | Proj_name | Dep_name |
|---|---|---|
| Silva | P1 | Sunil |
| Silva | P2 | Kamal |
| Silva | P1 | Kamal |
| Silva | P2 | Sunil |

✓ **1NF** ✓ **2NF** ✓ **3NF** ✓ **BCNF**

Normalization

✓ **Trivial Multivalued Dependency**

| Emp_name | Proj_name |
|---|---|
| Silva | P1 |
| Silva | P2 |

| Emp_name | Dep_name |
|---|---|
| Silva | Sunil |
| Silva | Kamal |

✓ **1NF**
✓ **2NF**
✓ **3NF**
✓ **BCNF**
✓ **4NF**