Made by Shravan Kumar

Submited to CSS CORP

# DATABASE

## INTRODUCTION TO DATABASE

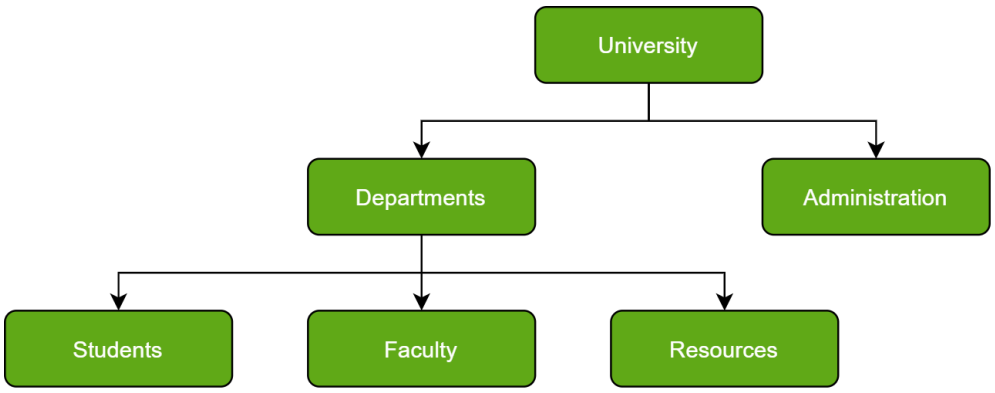
### 1) What is database?

Databases can store very large numbers of records efficiently, It is easy to add new data and edit or delete old data. Data can be searched easily, More than one person can access the same database at the same time - multi-access

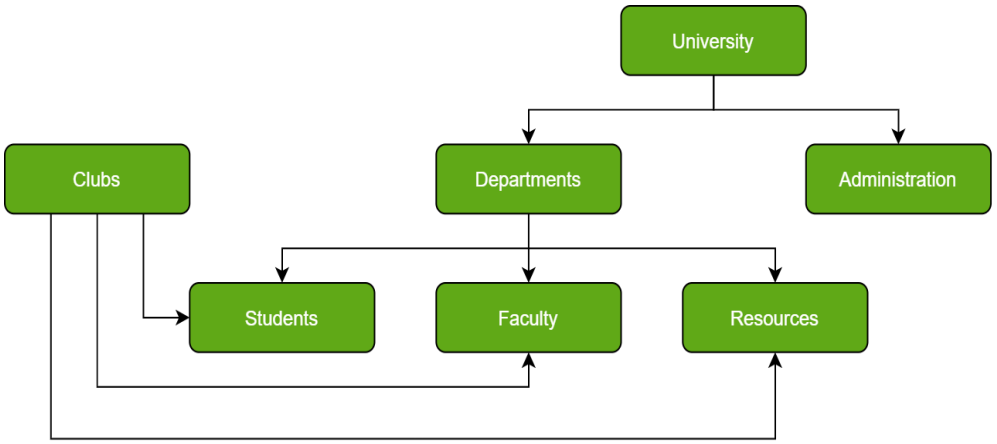
### 2) Modes of storing data ?

=>Hierarchical Data model
=>Network Data Model
=>Relational Data Model

Hierarchical database model resembles a tree structure, similar to a folder architecture in your computer system. The relationships between records are pre-defined in a one to one manner, between 'parent and child' nodes. They require the user to pass a hierarchy in order to access needed data. Due to limitations, such databases may be confined to specific uses.



Network database models also have a hierarchical structure. However, instead of using a single-parent tree hierarchy, this model supports many to many relationships, as child tables can have more than one parent.




A relational database management system (RDBMS) is a program that allows you to create, update, and administer a relational database. Most relational database management systems use the SQL language to access the database.

| Roll no. | Student Name | Marks Awarded |
|---|---|---|
| 1 | Raman Triphati | 86 |
| 2 | Rajan Govindan | 94 |
| 3 | Mahesh Nandalal | 94 |

Key = 94

| Marks Awarded | Student Name | Rank | Scholarship |
|---|---|---|---|
| 94 | Rajan Govindan | 17 | Yes |
| 94 | Mahesh Nandlal | 16 | Yes |

| Section | Student Name | Marks Awarded | Rank |
|---|---|---|---|
| A | Raman Tripathi | 86 | 43 |
| B | Rajan Govindan | 94 | 17 |
| C | Mahesh Nandlal | 94 | 16 |

# NORMALIZATION

## INTRODUCTION TO NORMALIZATION

### 1) What is normalization ?

Normalization is the process of organizing the data in the database. Normalization divides the larger table into the smaller table and links them using relationship. Database designed based on Entity Relationship (ER) model may have some amount of inconsistency, ambiguity and redundancy. The normal form is used to reduce redundancy from the database table

### 2) What are the types anomaly?

| Student Details | | | Course Details | | | Result details | | |
|---|---|---|---|---|---|---|---|---|
| 1001 | Ram | 11/09/1986 | M4 | Basic Maths | 7 | 11/11/2004 | 89 | A |
| 1002 | Shyam | 12/08/1987 | M4 | Basic Maths | 7 | 11/11/2004 | 78 | B |
| 1001 | Ram | 23/06/1987 | H6 | | 4 | 11/11/2004 | 87 | A |
| 1003 | Sita | 16/07/1985 | C3 | Basic Chemistry | 11 | 11/11/2004 | 90 | A |
| 1004 | Gita | 24/09/1988 | B3 | | 8 | 11/11/2004 | 78 | B |
| 1002 | Shyam | 23/06/1988 | P3 | Basic Physics | 13 | 11/11/2004 | 67 | C |
| 1005 | Sunita | 14/09/1987 | P3 | Basic Physics | 13 | 11/11/2004 | 78 | B |
| 1003 | Sita | 23/10/1987 | B4 | | 5 | 11/11/2004 | 67 | C |
| 1005 | Sunita | 13/03/1990 | H6 | | 4 | 11/11/2004 | 56 | D |
| 1004 | Gita | 21/08/1987 | M4 | Basic Maths | 7 | 11/11/2004 | 78 | B |

Insert Anomaly: We cannot insert prospective course which does not have any registered student or we cannot insert student details that is yet to register for any course.

Update Anomaly: if we want to update the course M4's name we need to do this operation three times. Similarly we may have to update student 1003's name twice if it changes.

**Delete Anomaly:** if we want to delete a course M4 , in addition to M4 occurs details , other critical details of student also will be deleted. This kind of deletion is harmful to business. Moreover, M4 appears thrice in above table and needs to be deleted thrice.

**Duplicate Data:** Course M4's data is stored thrice and student 1002's data stored twice. This redundancy will increase as the number of course offerings increases.

## 3) Types of Dependency ?

**Functional dependent** The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

$$X \rightarrow Y$$

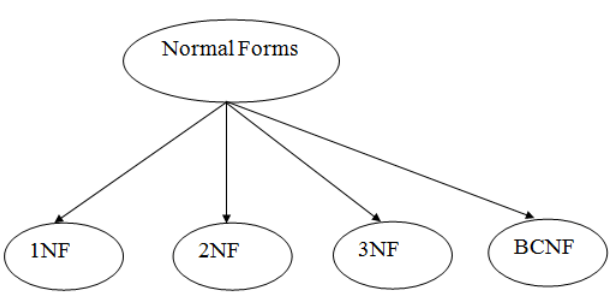
The left side of FD is known as a determinant, the right side of the production is known as a dependent.

**Full Functional Dependency** between any two attributes X and Y, when X is functionally dependent on Y and is not functionally dependent on any proper subset of Y.

**Transitive dependency** occurs when one non-prime attribute is dependent on another non-prime attribute

If P -> Q and Q -> R is true, then P-> R is a transitive dependency.

## 4) Types of Normal Forms?



| Normal Form | Description |
|---|---|
| 1NF | A relation is in 1NF if it contains an atomic value. |
| 2NF | A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key. |
| 3NF | A relation will be in 3NF if it is in 2NF and no transition dependency exists. |
| BCNF | A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency. |

## First Normal Form

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

**Example:** Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP_PHONE.

**EMPLOYEE table:**

| EMP_ID | EMP_NAME | EMP_PHONE | EMP_STATE |
|---|---|---|---|
| 14 | John | 7272826385, 9064738238 | UP |
| 20 | Harry | 8574783832 | Bihar |
| 12 | Sam | 7390372389, 8589830302 | Punjab |

The decomposition of the EMPLOYEE table into 1NF has been shown below:

| EMP_ID | EMP_NAME | EMP_PHONE | EMP_STATE |
|---|---|---|---|
| 14 | John | 7272826385 | UP |
| 14 | John | 9064738238 | UP |
| 20 | Harry | 8574783832 | Bihar |
| 12 | Sam | 7390372389 | Punjab |
| 12 | Sam | 8589830302 | Punjab |

## Second Normal Form

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

**Example:** Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

**TEACHER table**

| TEACHER_ID | SUBJECT | TEACHER_AGE |
|---|---|---|
| 25 | Chemistry | 30 |
| 25 | Biology | 30 |
| 47 | English | 35 |
| 83 | Math | 38 |
| 83 | Computer | 38 |

In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.

To convert the given table into 2NF, we decompose it into two tables

**TEACHER_DETAIL table:**

| TEACHER_ID | TEACHER_AGE |
|---|---|
| 25 | 30 |
| 47 | 35 |
| 83 | 38 |

**TEACHER_SUBJECT table:**

| TEACHER_ID | SUBJECT |
|---|---|
| 25 | Chemistry |
| 25 | Biology |
| 47 | English |
| 83 | Math |
| 83 | Computer |

## Third Normal Form

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

1. X is a super key.
2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.

**Example:**

**EMPLOYEE_DETAIL table:**

| EMP_ID | EMP_NAME | EMP_ZIP | EMP_STATE | EMP_CITY |
|---|---|---|---|---|
| 222 | Harry | 201010 | UP | Noida |
| 333 | Stephan | 02228 | US | Boston |
| 444 | Lan | 60007 | US | Chicago |
| 555 | Katharine | 06389 | UK | Norwich |
| 666 | John | 462007 | MP | Bhopal |

**Super key in the table above:**

{EMP_ID}, {EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}....so on
**Candidate key:** {EMP_ID}

**Non-prime attributes:** In the given table, all attributes except EMP_ID are non-prime.

**EMPLOYEE table:**

| EMP_ID | EMP_NAME | EMP_ZIP |
|---|---|---|
| 222 | Harry | 201010 |
| 333 | Stephan | 02228 |
| 444 | Lan | 60007 |
| 555 | Katharine | 06389 |
| 666 | John | 462007 |

**EMPLOYEE_ZIP table:**

| EMP_ZIP | EMP_STATE | EMP_CITY |
|---|---|---|
| 201010 | UP | Noida |
| 02228 | US | Boston |
| 60007 | US | Chicago |
| 06389 | UK | Norwich |
| 462007 | MP | Bhopal |

## Boyce Codd normal form (BCNF)

- BCNF is the advance version of 3NF. It is stricter than 3NF.
- A table is in BCNF if every functional dependency X → Y, X is the super key of the table.
- For BCNF, the table should be in 3NF, and for every FD, LHS is super key.

**Example:**

**EMPLOYEE table:**

| EMP_ID | EMP_COUNTRY | EMP_DEPT | DEPT_TYPE | EMP_DEPT_NO |
|---|---|---|---|---|
| 264 | India | Designing | D394 | 283 |
| 264 | India | Testing | D394 | 300 |
| 364 | UK | Stores | D283 | 232 |
| 364 | UK | Developing | D283 | 549 |

**In the above table Functional dependencies are as follows:**

1. EMP_ID → EMP_COUNTRY
2. EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

**Candidate key: {EMP-ID, EMP-DEPT}**

The table is not in BCNF because neither EMP_DEPT nor EMP_ID alone are keys.

To convert the given table into BCNF, we decompose it into three tables:

**EMP_COUNTRY table:**

| EMP_ID | EMP_COUNTRY |
|---|---|
| 264 | India |
| 264 | India |

**EMP_DEPT table:**

| EMP_DEPT | DEPT_TYPE | EMP_DEPT_NO |
|---|---|---|
| Designing | D394 | 283 |
| Testing | D394 | 300 |
| Stores | D283 | 232 |
| Developing | D283 | 549 |

**EMP_DEPT_MAPPING table:**

| EMP_ID | EMP_DEPT |
|---|---|
| D394 | 283 |
| D394 | 300 |
| D283 | 232 |
| D283 | 549 |

**Functional dependencies:**

1. EMP_ID → EMP_COUNTRY
2. EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

**Candidate keys:**

**For the first table:** EMP_ID
**For the second table:** EMP_DEPT
**For the third table:** {EMP_ID, EMP_DEPT}

Now, this is in BCNF because left side part of both the functional dependencies is a key.