UNIT-II

<u>RELATIONAL</u>

<u>DATABASE</u>

Structure of Relational Database:

Relational structure is a set of attributes and relation. In a relational structure of database, we use different terminology which are shown below:

Attributes

Primary key

Cust_id#	Cust_name	Cust_add	Cust_city	AccNo
C1	Shruti	Camp	Amravati	1260
C2	Viveka	Sainagar	Amravati	1751
C3	Kajol	Somalwada	Nagpur	2184
C4	Roocha	J M Road	Pune	3497
C5	Sayali	Uni Road	Solapur	4337

Relation

TUPLE

CADINALITY

------------------------Degree----------------------

Fig: Cust_Table

1)Relation: Relation is a table of data with primary key.

Example: Cust_table

2)Tuple: Tuple is a single record or any single row information.

Example : C1 Shruti Camp Amravati 1260

3)Attribute: Attribute is a title or headed by name of column. It is descriptive property possess by entity set.

Example: Cust_id ,Cust_name

4)Cardinality: Number of rows in the relation is called as cardinality.

<u>5)Degree:</u> Number of attributes in a relation is called as degree.

<u>6)Primary Key: It</u> is used to uniquely identify the value.

<u>7)Entity:</u> Entity is a basic unit used in modeling classes of concrete or abstract object.

<u>8)Domain:</u> Domain is a set of possible values generally it contain complete entity set with attributes and values.

Relational algebra

1] Relational algebra is a collection of operation to manipulate relation.

2] It is a procedural manipulation language. It describes the procedure for solving any type of problem .

3] It specifies the operation to be performed on existing relation to derive result relation

4] It defines the complete scheme for each of the result relation.

5] It specifies the operation and the order in which they are to be performed on tupple of relation.

6] It is just like a programming language.

Relational algebra is divided into following types of operations.

1] Set Oriented Operation:

The operation that are performed on particular number of set are called as set oriented operation. It is also called as ‘basic operations or traditional operators. Set oriented operation broadly divided into following four types:

(a) Union
(b) Intersection
(c) Difference
(d) Cartesian product

2]Relation Oriented Operation:

The set operation provide very limited data manipulation by relation oriented operation.

Relation oriented operation are broadly divided into following four types :

- **(a) Projection**
- **(b) Selection**
- **(c) Join**
- **(d) Division**

Set Oriented Operations

The operation performed on particular number of well-defined sets are called asset oriented operation or traditional operators.

Set oriented operation broadly divided into following four types:

(a)Union
(b)Intersection
(c)Difference
(d)Cartesian product

(a)Union:

The union of two sets A and B is the set that contains the elements belonging to set A or set B or both set A and set B. The union is denoted by ‘U’. It set A and B have any element in common then union will not duplicate those members.

Example:

(1) A={BCD}

B={AXY,BCD,PQR}

AUB={AXY,BCD,PQR}

(2)

P:

ID	Name
101	Shruti
102	Viveka
104	Kajol
106	Roocha
108	sayali

Q:

ID	Name
101	Shruti
103	Poonam
105	Nikita
110	Ashwini
118	Priyanka

R:	P U Q
ID	**Name**
101	Shruti
102	Viveka
103	Poonam
104	Kajol
105	Nikita
106	Roocha
108	Sayali
110	Ashwini
118	Priyanka

(b)Intersection:

Intersection selects common tuples from two relation. The intersection of two set A and B is a set composed of all common elements belonging to set A and set B.
The is denoted by ‘U*’.
If A and B are two sets, then A is included in B (A C B) if an only if each member of A is also a member of B.

Example:

(1) A={BCD}

B={ AXY, BCD, PQR }

AUB={BCD}

(2)

P:	
ID	Name
101	Shruti
102	Viveka
104	Kajol
106	Roocha
108	Sayali

Q	
ID	Name
101	Shruti
103	Poonam
105	Nikita
110	Ashwini
118	Priyanka

R:	P U Q
ID	Name
101	Shruti

(c) Difference:

In difference operation removes common tuples from the first relation. The difference of two sets A and B is a set that contain all element that are members of A but not B. It is denoted by "-"

Example:

(1) A={BCD}

B={AXY, BCD, PQR}

A-B=Ø (NULL SET)

B-A={AXY, PQR}

(2)

P:	
ID	Name
101	Shruti
102	Viveka
103	Poonam
104	Kajol
105	Nikita

Q:	
ID	Name
101	Shruti
102	Viveka
106	Roocha

P-Q	
ID	Name
103	Poonam
104	Kajol
105	Nikita

Q-P	
ID	Name
106	Roocha

(d) Cartesian product:

The Cartesian product of two relation is a concatenation of the tuples belonging to two relations. A new relation scheme will be created consisting of all possible combinations of the tuples. The Cartesian product of two sets A and B is a set containing of all ordered pair (a,b) for which **a Є A And b Є B . It is denoted by "X".**

Example:

- J={BCD,ABC}
  K={Hierrarchical,Relational}

  J X K = {(BCD,Hierarchical),(BCD,Relational),
  (ABC, Hierarchical), (ABC,Relational)}

  K X J ={ (Hierarchical,BCD),(Relational,ABC),
  (Hierarchical,BCD), (Relational,ABC)}

  J X K and K X J are entirely different in above example.

P:	
ID	Name
101	Shruti
102	Viveka
103	Poonam
104	Kajol

Q
S
J1
J2

R:		
ID	Name	S
101	Shruti	J1
101	Shruti	J2
102	Viveka	J1
102	Viveka	J2
103	Poonam	J1
103	Poonam	J2
104	Kajol	J1
104	Kajol	J2

RELATIONAL ORIENTED OPERATION:

The set oriented operation provide very limited data manipulation facility which can be overcome by relation oriented operation, Relation oriented operation are broadly divided into following four types:

- **Projection**
- **Selection**
- **Join**
- **Division**

Projection:

- **Projection of a relation is defined as a projection of all tuples over some set of attributes.**

- **It is also called as "Vertical Subset " of a relation.**

- **It is use to either reduce the number of attributes in the resultant relation or to reorder attributes.**

- **It is also called as a attributes restriction.**

- **It is denoted by "π**

Name	Age	Proffesion
Shruti	18	Analyst
Kajol	20	Receptionist
Nikita	24	Analyst
Poonam	21	Programmer
Priyanka	22	Programmer

Fig: Employee Database

Examples:

- **Π Name , Age (Employee)**
  = Select Name, Age From Employee
- **Π Profession, Age (Employee)**
  = Select Profession , Age From Employee
- **Π Age (Employee)**
  = Select Age From Employee
- **Π Name , Profession (Employee)**
  = Select Name, Profession From Employee

(b) Selection:-

1) Selection is nothing but to select particular Tuple of relation.

2) It can produce "Horizontal Subset " for any given relation.

3) The action is defined over the complete set of attributes name but only the subset of the Tuples are included in the result.

4) To have Tuple included in the result relation , specified selection conditions or predicates must be satisfied by it.

5) It is represented by symbol 'σ'.

6) It is also known as "Restriction Operation".

Name	Age	Proffesion
Shruti	18	Analyst
Kajol	20	Receptionist
Poonam	24	Analyst
Viveka	21	Programmer
Sayali	22	Programmer

Fig:- Employee Database

Examples:-

- σ age > 50 (Employee)
  Select age From Employee where age>50.
- σ name , age (Employee)
  Select age , name From Employee.
- σ age (Employee)
  Select age From Employee.
- σ name , age ,profession (Employee)
  Select name , age and profession From Employee.

(C) Join:-

1) Join operator allow to combine two relations to form a single new relations.

2) The tuples from the operand relations that participate in the operation and contribute to the result or related.

3) It allows the processing of relationships existing between the operand relations.

4) It is basically the Cartesian product of the relation followed by a selection operation.

5) It can performed in various type like equi join, semi join, inner join ,outer join.

6) It is represented by symbol '∞'.

Example:-

EMPLOYEE

ID	NAME
100	Ashwini
101	Priyanka
102	Nikita

SALARY

ID	SALARY
100	10000
101	15000
102	20000

EMPLOYEE ∞SALARY

ID	NAME	SALARY
100	Ashwini	10000
101	Priyanka	15000
102	Nikita	20000

(d) Division:

1) Division operation is useful when query involve for all objects having all the specified properties.

2) It is represented by symbol '÷'.

Example:

P(p)	
A	**B**
a1	b1
a1	b2
a2	b1
a3	b1
a4	b2
a5	b1
a5	b2

Q(q):
B
b1
b2

Result:

R(r)
A
a1
a5

RELATIONAL CALCULAS:

- Relational calculus is a query system where in queries are expressed as variables and formulas on these variables.

- It means calculating with relations is based on predicate calculus, which is calculating with predicates.

- It is formal language use to symbolize logical arguments in mathematics.

- It can build preposition that specifies a certain property or characteristic of an object.

- Prepositions specifying a property consist of an expression that names an individual object and another expression called the predicate, that stands for the individual object possess.

- Predicate calculus is a formal language consisting of symbols..

- **Primitive symbols are variable, constant, predicates.**

- **Logical connectors are also used and as follows.**

a) **NOT for negation**

⇨ **~,! (Bar), NOT**

b) **OR for disjunction**

⇨ **V**

c) **AND for conjunction**

⇨ **^**

d) **Implication for**

⇨ **condition** $\rightarrow$

e) **Biconditional for both are identical**

⇨ $\rightleftarrows$

f) **Equivalence for equality**

⇨ $\leftrightarrow$

- **Predicate calculus are of two type :**

1.Monadic predicate or one place predicate.

2.Two-place predicate

- **Other formula are formed with the use of quantifiers**

a) Universal or “for all” → U

b) Existential or “for some”→ Ǝ

or their exist.

c) Equivalence → ≡

E-R Model

- The E-R model is based on perception of real words which consist of collection of objects.

- These objects are referred as entities in E-R model.

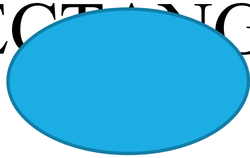
- An entity is an object that is distinguishable from others object by specific set of attributes.

- The data base structure employing the E-R model is usually shown pictorially using E-R diagram.

The following convention are used for drawing E-R diagram

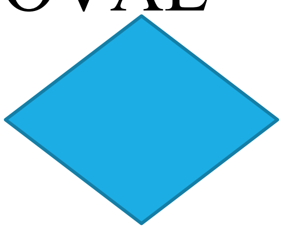
Entity set is shown by this diagram

RECTANGLE



The attribute of entity set is shown by this symbol.

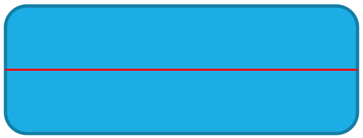
OVAL



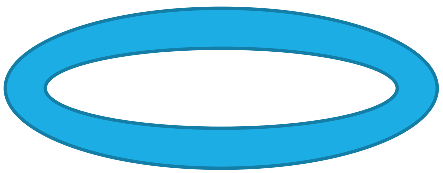
The relationship among the no. of entity set is shown by this symbol.

DIAMOND

LINE. The nature of relationship between entity, attribute and relationship diamond is shown by this symbol.



The attribute of entity set when using a DASHED ELLIPS.

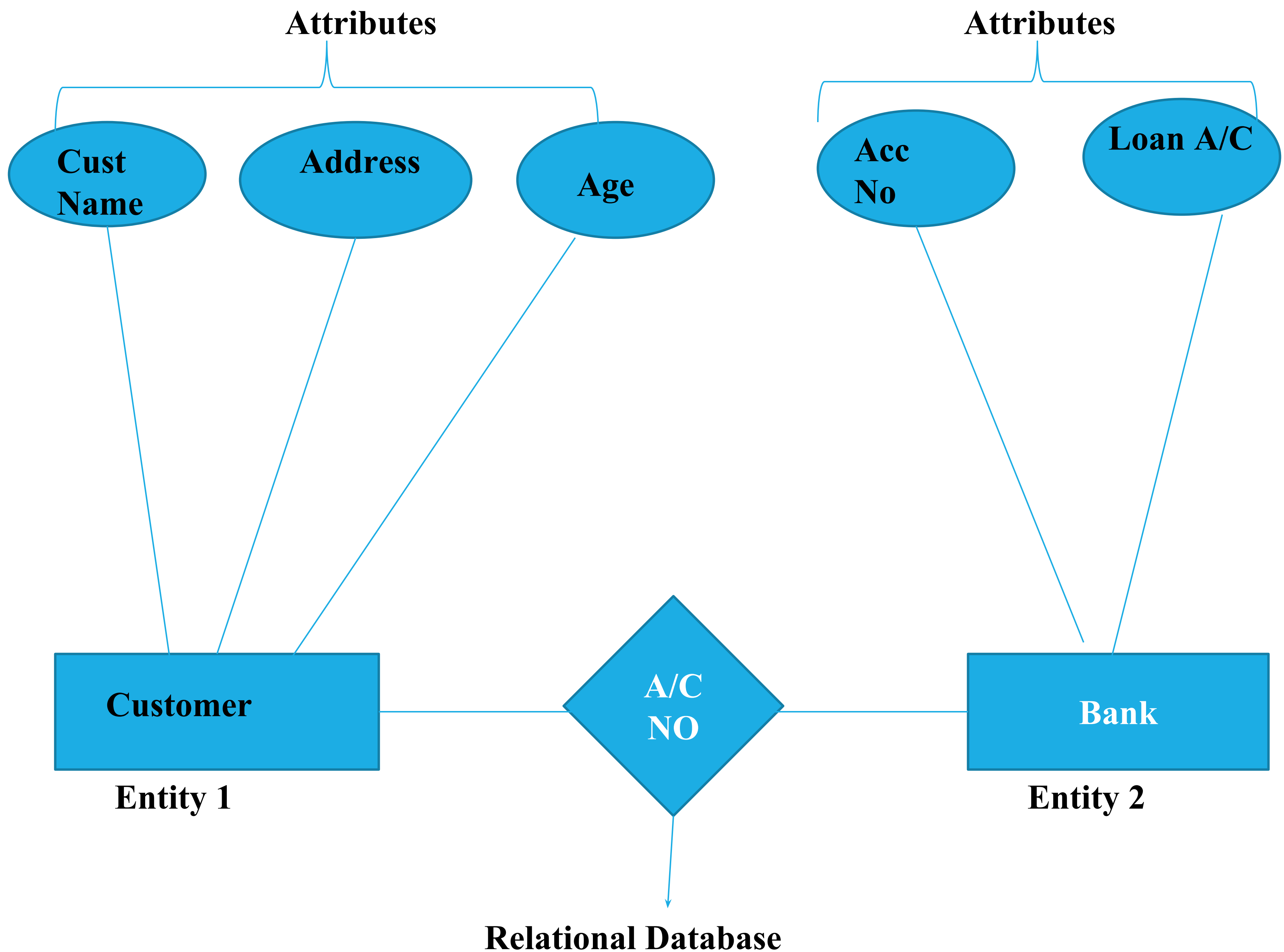


It represents multi-value attributes

DOUBLE ELLIPS.

Total participation of an a relationship set.

DOUBLE LINE.



Attributes
Attributes
Cust Name
Address
Age
Acc No
Loan A/C
Customer
A/C NO
Bank
Entity 1
Entity 2
Relational Database

TUPLE CALCULAS

a) Queries in tuple calculus are expressed by a tuple calculus expression.

b) A tuple calculus expression is of the form {X/f(x)}
Where, f is a f formula involving x & x represents a set of tuple variables. The expression characterizes a set of tuples of x such that formulas(x) is true.

c) A variable appearing in a formula is said to be free unless it is quantified by existential (for all) quantified

d) The variables quantified by or as said to be bound.

e) **In tuple calculus we defined quantified variable as t[A]**

Where,

T is a tuple variable of some relation.

A is an attribute of that relation

f) **The tuple calculus formulas are built from atoms, an atom can have either of forms given below.**

i) **s Є r, where s is tuple variable & r is a relation**

ii) **S[x] ɵ u[y], where s and u are tuple variable**

X is attribute on which s is defined.

Y is attribute on which u is defend.

ɵ is one of the comparison operators {=, ≠, <, <=,>,>=}

g) The formulas are built up from atom using following rules:

1) **An atom is a formula**
2) **If f and g are formulas, then (f),f V g, f ^ g, f→g are also formulas**
3) **If f(x) is a formula where x is free, then x (f(x)) and x (f(x)) are also formulas However now x s bound**

e.g. get empno of employees working on project "COMP 353".

Tuple calculus Example:

{ t[empno] | Ǝ u (u Є Assigned_to) ^
u[project no]='COMP353' ^
t[empno]= u[empno]
}

Where, t[empno] – set of tuple of empno
u[empno] – some of tuple of empno.

|-such that
Ǝ-their exist
Є -belongs to
t-variable used for set of tuple
u-variable used for some of tuple

The result of the above query is as follows:

1) Set of tuple t such that their exist a tuple u in assigned to with values "COMP353" for project no and same value of the employee no attribute in both u & t.

2) In this formulation, use specify the set of tuples t [empno] such that the predicate in true for each element of that set.

3)The predicate specifies that their exist some tupple u in the relation assigned to such that it has a value “COMP353” for project no attribute and also a value for empno attributes of the result tuple t is same as that for tuple u.

4) Free variables appear to the left of |(bar) symbol the variable t is free tuple variable in above formula and whatever the attribute and corresponding values.

5)The formula restricts t to the relation scheme [empno].

Domain Calculus

A] A domain calculus expression

{X|F(X)}

Where, F is a formula on X and X represents a set of domain variables. The expression characterizes X such that f(X) is true.

B] Domain calculus formula is also from/built from atoms.

C] R(R) is a relation and X {X1,X2,--------,Xn} in domain calculus, defined on subset of the relation attribute.

1]X Є R

2]XΘY or XΘC , where Θ is one of the comparison operators {=,#,<,<=,>,>=} X and Y are domain compatible variables and C is a domain compatible constant.

D]The formulas are built from atoms using the following rules:

1] An atom is a formula.

2] If f and g are formula then Γf,(f),fVg , fΛg , f→g are also formulas.

3] If f(x) are formula , the Ǝ X(f(x)) are also formula where X is free , then X(f(X)) are also formulas

Example:
Get empno of employees working on project “comp353”

Domain Calculus Example:

{e|*]p(<e,p> Є assigned to ^p=’COMP333}

The result of above query is as follows:

- **The method of converting this query into domain calculate expression is by conjunction during the existence of p, a project # . This project # is such that current value of domain variable e (domain of e being domain of emp #) and p (domain of p being domain of project #) are in tuple of relation assigned to and value of p is “COMP353”.**

- **In above formulation, we are specifying the set of domain value for the domain variable e such that the predicate is true. The predicate specific that there exit a value of domain variable p such that its current value along with the value of the domain variable e is in (the same tuple of) the relation assigned to the specific value of p is the value "COMP353".**

- **If we are inserting in particular known value of p, quantifiers can be dropped and the query can be simplified as**

{e|<e,p> Є assigned_to ^ p = ' COMP353 '}

NORMALIZATION

WHAT IS NORMALIZATION?

Data structuring is refined through A process called Normalization.

NEED OF NORMALIZATION:

Any relation satisfies a certain set of condition then it is said to be in a particular normal forms. The normal forms are used to minimize various types of anomalies & inconsistencies during database design.

NORMAL FORMS

The various normal forms are as follows:

1. **1NF**
2. **2NF**
3. **3NF**
4. **BCNF**

1st Normal Forms (1NF):

IT suggests that a relation R is in 1st NF if and only if all underlying Domain contains automatic values.

Example:

consider an un normalized relation schema sales persons which contains following attributes:

Empid	Emp name	Store branch	Dept.	Item No.	Item Desp.	Sales Price.
001	Anil	Khamla	hardware	TR100	Router	800
				SA 10	Saw	1500
002	Ram	Bharat Nagar	Auto Parts	MC164	Snow Tire	3000
				AC1462	Alternato	100
				BB1000	Battery	2000

Fig. Salesperson Relation

The normalized relations after 1NF has been applied is as follows:

Empid	Empname	Store branch	Dept
001	Anil	Khamla	Hardware
002	Ram	Bharat Nagar	Auto Parts

Fig. Sales Person Relation

Empid	Item no.	Item Desp.	Sales Prise
001	TR 100	Router	800
	SA 10	Saw	1500
	.	.	.
	.	.	.
	.	.	.

Fig. Salesperson Item relation

2ND NORMAL FORM(2NF):

A relation R is in 2nd normal form if & only if it is 1NF & every non key attribute depends on a key attribute or concatenated key.

Example: 1. Salesperson relation:

Empid	Emp_name	Store branch	Dept.
001	Anil	Khamla	H/W
.	.	.	.
.	.	.	.

2.Salesperson Item relations:

Empid	Item no.	Sales price
001	TR 100	800
	SA 10	1500

3.Item relation:

Item no.	Item description
TR 100	Router
SA 10	Saw

3RD NORMAL FORM(3NF):

A relation R is in 3rd normal form if & only if it is in 2NF & every non key attribute is non transitively dependent on the primary key.

Example:

Empid	Empname	Store Branch

1.Salesperson

(3)

Store Branch	Department

4.Store Branch

(1)

Empid	Item no.	Sales Price

2.Salesperson_item

(2)

Item no.	Item description

3.Item

4. BCNF(BOYCEE/CODD NORMAL FORM) :

A relation R is in BOYCEE / CODE NORMAL FORM (BCNF) if & only if every determinant is a candidate key. For a functional dependent relation where B is functionally dependent on A.

i.e. $A \longrightarrow B$.

ANOMALIES

WHAT IS ANOMALIES?

This repetition of information is undesirable and should be avoided. These relational schema used above leads to the problem of update anomalies.

Update anomalies can be classified as:-

1. **Insertion anomalies**
2. **Deletion anomalies**
3. **Modification anomalies**

FUNCTIONAL DEPENDENCIES

For a given relation R, attribute Y of R functionally dependent on attribute X of R if and only if each x-value in R is associated with one y-value of R. This dependency of Y on X is known as functional dependency. It is represented symbolically as:

X $\rightarrow$ Y IT IS READ AS Y IS FUNCTIONALLY DEPENDANT OF X.

EXAMPLE:

Emp_id	Emp_name	Dept.	Salary
001	Ram	Comp.	10,000
002	Shyam	accounts	8000

Emp. Emp_id $\rightarrow$ Emp. Emp_name

Functional dependency is used for basic 2 purposes which are as follows:
(A) to specify constraints on the set of relations.

(B) to check whether the relations are valid under the condition of functional dependency.

Assignment Questions based on Unit –2

Q.1.	Explain Structure of relational databases.	4
Q.2.	Explain Relation, Domain, Attribute with proper example.	3
Q.3.	What is key? Explain following keys. 1. primary 2. Foreign 3. Candidate 4. Super	4
Q.4.	Explain Relational Algebra.	5
Q.5.	Explain Set oriented operation in detail with proper example.	6
Q.6.	Explain Relation oriented operation in detail with example.	6
Q.7.	Explain Entity-Relationship model? With example.	4
Q.8.	Explain ER diagram ? explain with proper diagram & example.	4
Q.9.	What is Relational Calculus? Explain.	4
Q.10.	Explain Tuple Calculus with proper example.	4
Q. 11.	Explain Domain Calculus with proper example.	4
Q.12.	What is Relationship? Explain types of relationship.	6
Q.13.	What is Normalization? Explain with 1NF,2NF, 3NF normal forms.	6
Q.14.	Differentiate between 3NF & BCNF. (Any Four)	4
Q.15.	Explain Functional Dependency with proper example.	4
Q.16.	Explain Multi value Dependency with proper example.	4
Q.17.	What is anomaly? Explain types of anomalies.	5