* Explain Functional Dependencies.

There are Five type of Functional Dependencies.

1) Trivial Dependency
2) Non- Trivial Dependency
3) Transitive Dependency
4) Full - Functionally Dependency
5) Partial Dependency.

1 Trivial Dependency:

In Trivial Dependency, A Function dependent is always a subset of the determinant.

Ex. X -> Y is Trivial Function Dependency then Y is subset of X.

Ex. {e_id, e_name} -> e_id

then e_id is proper subset of e_id and e_name.

2 Non- Trivial Dependency:

In Non- Trivial Dependency, the Function dependent is strictly not subset of the determinant.

Ex. {e_id, e_name} → e_age

then e_age is not sub set of e_id and e_name.

3. Transitive Dependency:

In transitive Dependency, Function Dependency is causes the indirect

Ex.

e_id → e_name

e_name → e_DoB

∴ e_id → e_DoB

Using e_id we can get e_DoB.

4 Full-Functionally Dependency:

In Full-Functionally Dependency, Function A is dependent on aB but not any proper subset of A.

Ex.

~~e_id → e_name~~

{e_DoB, e_MN, } → e_id

e-Name

e-name → e-address

Using e_addresses we can not get full deatial of Employee, We get only e_name.

5 Partial Dependency:

In Partial Dependency, Function A is ~~depent~~ dependent on B but with ~~not~~ proper subset of A.

EX

{S_enrolln, S_branch} → e.S_SIP

Using s_enrolln we can get S_SIP without using S_brand Attribute.

* Explain Armstrong's Axioms with its operations.

~~Ans~~

Armstrong's Axioms are set of rules when we applied Functional Dependency.

It can be apply to a set of Functional Dependency to derive other Functional Dependency.

There are seven Armstrong's Axioms Operations.

1 Reflexivity:

In reflexivity, if Y is proper sub set of X, then X is determine Y.

$\therefore X \supseteq Y \Rightarrow X \to Y \to X \supseteq Y$

2 Augmentation:

In Augmentation, if X determine Y, then XZ determine YZ for any value of Z

$\therefore X \to Y$

$\therefore XZ \to YZ$

3 Transitivity:

In Transitivity, if X determine Y and Y determine Z then X is also determine Z.

$\therefore X \to Y, \; Y \to Z$

$\therefore X \to Z$

4 Union:

In Union, if x determine Y and x determine Z then x is also determine YZ

$\therefore x \to Y, \quad x \to Z$
$\therefore x \to YZ$

5 Decomposition:

In Decomposition, if x determine YZ then, x is also determine Y and Z respectively.

$\therefore x \to YZ$
$\therefore x \to Y, \quad x \to Z$

6 Composition:

In Composition, if x determine Y and a determine b then xa determine by.

$\therefore x \to Y, \quad a \to b$
$\therefore xa \to Yb$

7 Pseudo Transtivity:

In Pseudo Transtivity, if x determine Y and xa determine b then

ya determine b.

$\therefore x \to y, \quad xa \to b$
$\therefore ya \to b$

* Explain Normalization with its type.

Normalization is the process of organizing data in database.

This includes creating tables and establishing relationships between the table.

Normalization divides the larger table into smaller and links using relationship.

There are six types of Normalization

1) 1st Normal Form
2) 2nd Normal Form
3) 3rd Normal Form
4) Boyce-Codd Normal form
5) 5th Normal form
6) 6th Normal form

1 1st Normalform:

Condition: Each tuple of the table should contain a single value.

In this form attributes contain only single value not contain Multivalues.

EX.

S_id	S_MN
1	(25, 26)
2	27
3	28
4	29

In 1st Normalform:

S_id	S_MN
1	25
1	26
2	27
3	28
4	29

In First table S_MN attribute 1st id contain two value so, we convert the table into 1st Normal form.

2 2nd Normal Form:

Condition: It should be follow 1st Normal Form and It is based on full functional dependency.

In this form relations with a primary key composed of two or more attributes.

Ex.

*T_id	T_sub	T_age
1	A	25
1	B	25
2	C	26
3	D	27

In 2nd Normal form:

T_id	T_sub
1	A
1	B
2	C
3	D

T_id	T_age
1	25
1	25
2	26
3	27

In above Table-1 T_id is containt Primary key and it is relation with two attribute.

In last table, in 2nd normal form I_id has a relation with only One attribute.

3 3rd Normal form:

Condition: It should be follow 1st and 2nd normal form and it does not have transtive dependancy.

In this form one attribute relate with only one attribute.

Ex

e_id	e_name	e_zip	e_city	e_state
1	A	111	D	H
2	B	222	F	J
3	C	333	G	I

In this table {e_id, e_name} -> e_zip
e_zip -> {e_city, e_state}

then
{e_id, e_name} -> {e_city}

So, this is not condition in 3rd Normal form.

In Normal form:

e_id	e_name	e_zip
1	A	111
2	B	222
3	C	333

e-zip	e-city	e-state
111	D	H
222	F	J
333	G	I

In above table, both table are not containt transtive dependancy.

4 Boyce- codd Normalform:

Condition: It should be follow 1st, 2nd and 3rd normalform and It has trivial functional dependancy.

BCNF is a Advance version of 3rd Normalform.

Ex.

e_id	e_contry	e_dep	dep_type
1	A	C	1
2	B	D	2

In this table every attributes are not contant determinant.

In Boyce-Codd Normalform:

c_id	c_courlw
1	A
2	B

c_dep	dep_type
C	1
D	2

c_id	c_dep
1	C
2	D

In this table: Every Element have determinat.

5 4th Normalform:

Condition: It should be follow 1st, 2nd, 3rd and BCNF Normalform and not contain more than one multivalued dependency.

In this form, there are non-trivial multivalued dependency.

Ex.

S id	Course	Hobby
1	A	Sing
1	B	Dance
2	C	study
3	D	Dance

In this table s-id have more than One multivalue dependency

In 4th Normalform,

S_id	Course
1	A
1	B
2	C
3	D

S_id	Hobby
1	Sing
1	Dance
2	Study
3	Dance

In both table, S_id have only one multivalue dependancy.

6 5th Normalform:

Condition: It should be follow 1st, 2nd, 3rd, 4th and & BCNF normalform and it cannot be non loss decomposed.

In this normalform, relation decomposed into two relation must have loss-less property.

Ex

e_id	e_name	e_Course
1	A	D
2	B	E
3	C	F

In this table every have more than two relation.

In 5th normalform:

e-id	e-name
1	A
2	B
3	C

e-id	e-course
1	D
2	E
3	F

e-name	e-course
A	D
B	E
C	F

In this table, Every Attribute has only one relation.

* Explain Relational Decomposition with its type.

When a relation in the relation model is not in normal form then the relational decomposition is required.

Relational Decomposition, breaks the table into multiple tables.

There are two type of Relational Decomposition.

1) Lossless Decomposition
2) Dependency Preserving

1 Lossless Decomposition :

In Lossless Decomposition, It will give ~~gura~~ guarantees that the join of relations will result is the same relation as it was decomposed.

In this Decompostion, information is not lost from the relation that is ~~so~~ decomposed.

In lossless Decomposition, when we use natural join then decomposition give the original relation.

Ex. Student ⋈ Employee

S_id	S_name	e_id	e_name
1	A	3	C
2	B	4	D

lossless Decomposition

Student

S-id	S-name
1	A
2	B

Employee

e-id	e-name
3	C
4	D

In this both table, we use natural Join and result is same that Original result.

2 Dependency Preserving:

Dependency Preserving is a very important constraint in database

In this Dependency, One decomposed table must satisfy every table dependency

If a relation R is decomposed into two relation R_1 and R_2 then, at least R_1 or R_2 containt all the dependency.

Ex

e_id	e_name
1	A
2	B
3	C

e_name	e_age	e_age	e_dep
A	21	21	D
B	22	22	E
C	23	23	F

Dependency Preserving

e_id	e_name	e_age
1	A	21
2	B	22
3	C	23

(E1)

e_age	e_dep
21	D
22	E
23	F

(E_2)