- **Functional Dependency**

Suppose R is a relation, X,Y are attributes over R, t1,t2 are tuples in the relation R:

	X	Y
t1	**a1**	**b1**
t2	**a2**	**b2**

X→Y if and only if {(t1.X=t2.X)═►(t1.Y=t2.Y)}

Example1: Suppose you have the following table:

A	B
a1	b1
a2	b1

A→B is true (B is Functionally Dependent (FD) on A) because each value in A achieves one atomic value in B.

The following table:

A	B
a1	b1
a1	b1

A→B is true (B is FD on A) because each value in A achieves one atomic value in B.

The following table:

A	B
a1	b1
a1	b2

A→B is False (B is Not FD on A) because each value in A achieves two different values in B.

The Following table:

A	B
a1	b1
a2	b2

A→B is True (B is FD on A) because each value in A achieves different value in B.

- **Partial Dependency**

A functional dependency A → B is a *full* functional dependency if removal of any attribute from A results in the dependency no longer existing. A functional dependency A→B is a **partially dependency** if there is some attribute that can be removed from A and yet the dependency still holds.

FD : AB → CDE where AB is Primary Key.

Then, { A → C; A → D; A → E; B → C; B → D; B → E}
all are Partial Dependencies.

Example1: Suppose you have the following table:

stud_id	course_name	m_grade	f_grade
111	Programming	44	79
112	DB	50	88
111	OOP	60	80

Both (stud_id, course_name) are PK, and m_grade and f_grade depend on this PK.

So m_grade and f_grade are partially depended on only course_name instead on all PK componenets.

- **Partial Dependency occurs when there is PK consists of many keys (composite key) and non-primary keys depended on part of this composite key.**

- **Transitive Dependency**

  A condition where A, B, and C are attributes of a relation such that if A→B and B→C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

- Non-attribute key determines non-attributes key.

  Example: the table below:

Book	Genre	Author	Author Nationality
Twenty Thousand Leagues Under the Sea	Science Fiction	Jules Verne	French
Journey to the Center of the Earth	Science Fiction	Jules Verne	French
Leaves of Grass	Poetry	Walt Whitman	American
Anna Karenina	Literary Fiction	Leo Tolstoy	Russian
A Confession	Religious Autobiography	Leo Tolstoy	Russian

The functional dependency {Book} → {Author Nationality} applies; that is, if we know the book, we know the author's nationality. Furthermore:

- {Book} → {Author}
- {Author} does not → {Book}
- {Author} → {Author Nationality}

Therefore {Book} → {Author Nationality} is a transitive dependency.

-First Normal Form (1NF)

Example-1: Suppose a company wants to store the names and contact details of its employees. It creates a table that looks like this:

emp_id	emp_name	emp_address	emp_pos
1022	Sameer Hakim	Basrah	Manager HR Head Assist.
1223	Ibrahim Kamil	Basrah	Sales Officer
1221	Ali Hussain	Basrah	Sales Manager HR Manager
1322	Hayat Fadhil	Basrah	Marketing Director HR Head Assist.

Three Employees (Sameer Hakim, Ali Hussain, and Hayat Fadhil) have two positions.The rule to make the relation in 1NF is (each tuple has single atomic value), so we should separate the tuples which have more than one value in the column.The result table is:

emp_id	emp_name	emp_address	emp_pos
1022	Sameer Hakim	Basrah	Manager
1022	Sameer Hakim	Basrah	HR Head Assist.
1223	Ibrahim Kamil	Basrah	Sales Officer
1221	Ali Hussain	Basrah	Sales Manager HR Manager
1221	Ali Hussain	Basrah	HR Manager
1322	Hayat Fadhil	Basrah	Marketing Director HR Head Assist.
1322	Hayat Fadhil	Basrah	HR Head Assist.

Example-2: Suppose you have the following table:

cust_id	cust_name	item_id	id_price
10331	Ahmed	210 293	20$ 30$
10233	Ali	210 211 213	20$ 30$ 40$
10332	Ameer	201 200 203	40$ 20$ 20$

So the 1NF of the table is:

cust_id	cust_name	item_id	id_price
10331	Ahmed	210 293	20$ 30$
10331	Ahmed	293	30$
10233	Ali	210	20$
10233	Ali	211	30$
10233	Ali	213	40$
10332	Ameer	201	40$
10332	Ameer	200	20$
10332	Ameer	203	20$

Example-3: Convert the following table into 1NF:

stud_name	stud_birth	course_id	course_name	grade
ali	1995	IS102 IS1205	C++ Web Design	70 80
ahmed	1995	IS202 IS304 IS101	Programming DB C++	80 79 85

stud_name	stud_birth	course_id	course_name	grade
ali	1995	IS102	C++	70
ali	1995	IS1205	Web Design	80
ahmed	1995	IS202	Programming	80

ahmed	1995	IS304	DB	79
ahmed	1995	IS101	C++	85

-Second Normal Form (2NF)

2NF table is a relation that is in First Normal Form and every non-primary-key attribute is fully functionally dependent on the primary key (not partial dependency).

Exmaple-1: Suppose you have the following table:

teacher_id	**teach_mat**	**teacher_age**
111	c++	40
112	OOP	43
111	DB	40
123	DW	42
123	ADB	42

So the PK can be the composite key (teacher_id, teach_mat), teacher_age is non-primary key depends on PK (teacher_id,teach_mat). We can find that teacher_age depends on teacher_id alone which satisfying the partial dependency rule, so we should remove this dependency. So the first rule of converting table into 2NF should assure that the table be in 1NF and the table above in 1NF.

The second rule is removing partial dependency and to do so you should divide the table into sub-tables and ensure that the non-primary attributes depends on primary attributes.

Teacher Material table

teacher_id	**teach_mat**
111	c++
112	OOP
111	DB
123	DW
123	ADB

Teacher Age table

teacher_id	**teacher_age**
111	40
123	43
123	42

Example-2: Convert the following table into 2NF:

name	**age**	**pet**	**pet_name**
Jimmy	10	Dog	Rex
Lesy	11	Cat	Kitty
Jimmy	10	Cat	Fluff
Lesy	11	Dog	Max
Heather	11	Cat	Kimba

- The table in 1NF.
- Divide the table to many tables to ensure that the NP attribute depend on PK.

Names table

name	**age**
Jimmy	10
Lesy	11
Heather	11

Pet table

name	**pet**	**pet_name**
Jimmy	Dog	Rex
Lesy	Cat	Kitty
Jimmy	Cat	Fluff
Lesy	Dog	Max
Heather	Cat	Kimba

Example-3: convert the following table into 2NF:

stud_name	**stud_birth**	**course_id**	**course_name**	**grade**
ali	1995	IS102	C++	70
ali	1995	IS1205	Web Design	80
ahmed	1995	IS202	Programming	80
ahmed	1995	IS304	DB	79
ahmed	1995	IS101	C++	85

The table in 1NF so we should start dividing the table to ensure the PK dependency.

Students Names table

stud_name	**stud_birth**
ali	1995
ahmed	1995

Students Grade table

stud_name	**course_id**	**course_name**	**grade**
ali	IS102	C++	70
ali	IS1205	Web Design	80
ahmed	IS202	Programming	80
ahmed	IS304	DB	79
ahmed	IS101	C++	85

-Third Normal Form (3NF)

To ensure that the table in the 3NF:

- Table in 2NF.
- No transitive dependency found.

Example-1: suppose a company wants to store the complete address of each employee:

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
112	Ali	12010	Basrah	Ashar	Ashar
123	Sally	12011	Basrah	Basrah	Nadhran
132	Ahmed	13011	Thi-Qar	Shatra	Shatra
122	Hamid	12013	Basrah	Shat-Alarab	Tannuma

So the table in 2NF

But (emp_state, emp_city, emp_district) depend on non-primary key (emp_zip),so:

Employee Names table:

emp_id	emp_name	emp_zip
112	Ali	12010
123	Sally	12011
132	Ahmed	13011
122	Hamid	12013

And Employee Location table:

emp_zip	emp_state	emp_city	emp_district
12010	Basrah	Ashar	Ashar
12011	Basrah	Basrah	Nadhran
13011	Thi-Qar	Shatra	Shatra
12013	Basrah	Shat-Alarab	Tannuma

Example-2: Suppose you have the following table:

book_id	**genre_id**	**genre_type**	**book_price**
1	1	Gardening	10$
2	2	Sport	20$
3	1	Gardening	12$
4	3	Travel	15$
5	2	Sport	12$

So the table in 2NF but genre_type depends on genre_id (transitive dependency).

Book types table:

book_id	**genre_id**	**book_price**
1	1	10$
2	2	20$
3	1	12$
4	3	15$
5	2	12$

Book Genre Types table:

genre_id	**genre_type**
1	Gardening
2	Sport
3	Travel

- All non-primary keys are depended on PK in each table.

Example-3: Suppose you have the following tables:

name	**age**
Jimmy	10
Lesy	11
Heather	11

name	**pet**	**pet_name**
Jimmy	Dog	Rex
Lesy	Cat	Kitty
Jimmy	Cat	Fluff
Lesy	Dog	Max
Heather	Cat	Kimba

To convert them into 3NF we should ensure all non-primary key attributes depend on PK attributes.

So:

Pet Names table:

pet_id	**pet**	**pet_name**	**name_id**
1	Dog	Rex	1
2	Cat	Kitty	2
3	Cat	Fluff	1
4	Dog	Max	2
5	Cat	Kimba	3

Kids Names table:

name_id	**name**	**age**
1	Jimmy	10
2	Lesy	11
3	Heather	11

Q_1:The table shown in the below figure lists sample dentist/patient appointment data. A patient is given an appointment at a specific time and date with a dentist located at a particular surgery. On each day of patient appointments, a dentist is allocated to a specific surgery for that day.

staffNo	dentistName	patNo	patName	appointment date	time	surgeryNo
S1011	Tony Smith	P100	Gillian White	12-Sep-04	10.00	S15
S1011	Tony Smith	P105	Jill Bell	12-Sep-04	12.00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sep-04	10.00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sep-04	14.00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Sep-04	16.30	S15
S1032	Robin Plevin	P110	John Walker	15-Sep-04	18.00	S13

a) The table shown in figure is susceptible to update anomalies. Provide examples of insertion, deletion, and update anomalies.

b) Identify the functional dependencies represented by the attributes shown in the table of figure. State any assumptions you make about the data and the attributes shown in this table.

c) Describe and illustrate the process of normalizing the table shown in figure to 3NF relations. Identify the primary, alternate, and foreign keys in your 3NF relations.

Q_2: The table shown in figure is susceptible to update anomalies:

NIN	contractNo	hours	eName	hNo	hLoc
1135	C1024	16	Smith J	H25	East Kilbride
1057	C1024	24	Hocine D	H25	East Kilbride
1068	C1025	28	White T	H4	Glasgow
1135	C1025	15	Smith J	H4	Glasgow

a) Provide examples of insertion, deletion, and update anomalies.

b) Identify the functional dependencies represented by the attributes shown in the table of the figure. State any assumptions you make about the data and the attributes shown in this table.

c) Describe and illustrate the process of normalizing the table shown in Figure 13.20 to 3NF. Identify primary, alternate and foreign keys in your relations.

Q_3: Examine the Patient Medication Form for the *Wellmeadows Hospital* case study shown in figure.

Wellmeadows Hospital
Patient Medication Form

Patient Number: P10034

Full Name: Robert MacDonald **Ward Number:** Ward 11

Bed Number: 84 **Ward Name:** Orthopaedic

Drug Number	Name	Description	Dosage	Method of Admin	Units per Day	Start Date	Finish Date
10223	Morphine	Pain Killer	10mg/ml	Oral	50	24/03/04	24/04/05
10334	Tetracyclene	Antibiotic	0.5mg/ml	IV	10	24/03/04	17/04/04
10223	Morphine	Pain Killer	10mg/ml	Oral	10	25/04/05	02/05/06

(a) Identify the functional dependencies represented by the attributes shown in the form in the figure.

State any assumptions you make about the data and the attributes shown in this form.

(b) Describe and illustrate the process of normalizing the attributes shown in the figure to produce a set of well-designed 3NF relations.

(c) Identify the primary, alternate, and foreign keys in your 3NF relations.

Q_4: convert the following tables into 3NF and give examples of (update, delete, and insert anomalies):

1-

item	**colors**	**price**	**tax**
T-Shirt	red, blue	12$	0.60
Trouser	red, yellow	12$	0.60
T-Shirt	red, blue	12$	0.60
Skirt	black, blue	25$	1.25

2-

name	**assignment1**	**assignment2**
Ali Ahmed	IS201	IS202
Sami Ali	IS201	IS203
Ahmed Jabbar	IS201	IS203

3-

course_id	**semester_id**	**students_num**	**course_name**
IT101	201301	25	Database
IT101	201302	25	Database
IT102	201301	30	Web Design
IT102	201302	35	Web Design
IT103	201401	20	Networking