Normalization

it is the process of organising data in a database efficiently by eliminating redundancy and inconsistent dependency.

s_id	s_name	age	couse_id	cname	instructor
1	Nilesh	23	101	Math101	A
1	Nilesh	23	102	Physics102	B
2	subrany	23	101	Math101	A
2	subrany	23	102	Physics102	B
2	subrany	23	103	chemistry	C

there are mainly three types of problem we faced.

Insertion Anomalies

Update Anomalies

Delete Anomalies

1) Insertion Anomalies

Insertion anomalies occur when we encounter difficulties inserting new data into a table. we get difficulties when we want to add only course information in the table.

2) **Update Anomalies**

An update Anomalies arise when data is redundant and inconsitent.

- we need to update value in multiple place.

3) **Delete Anomalies** A delete anomalies occur when we delete data from table leads to un intentional loss of data.

types of Normalization

1 nf
2nf
3nf
bcnf

1) first normal form (1NF):

- each cell contain only atomic value.
- there are no duplicate tuple.

st_id	st_name	courses
1	Amrendra	math103, Phy102, Bio103
2	Himanshy	Math, Physics
3	Mohit	math, chemistry

st_id	st_name	courses
1	Amrendra	math103,
2	Amrendra	Phy102
3	Amrendra	, Bio103
4	Himanshy	Math
5	Himanshy	, Physics
6	Mohit	math,
7.	Mohit	chemistry

2) second normal form (2 NF)

- it should be first normal form.
- All non-prime attributes is fully dependent on prime attributes.
- eliminate partial dependency.

st_id	s_name	c_id	c_name	Instructor
1	Amrendra	101	math 103,	A
1	Amrendra	101	Phy 103,	B
1	Amrendra	101	che 103,	C
2	Himanshy	102	Phy 102	B
3	Mohit	103	Che 103	C

Prime Attr → { st_id, c_id }

non prim Attr → { s_name, c_name, Instructor }

partially dependency.

3) third normal form :

- if should be in 2NF.
- all non prim attributes should be directly depends on prime attributes.
- eciminate transitive dependency.

st_id	s_name	c_id	c_name	Instructor
1	Amrendra	101	math101,	A
1	Amrendra	101	Phy 101,	B
1	Amrendra	101	che 101,	C
2	Himanshy	102	Phy102	B
3	Mohit	103	che 103	C

Prime Att → { st_id, c_id }

non prim Att → { s_name, c_name, Instructor }

4) BCNF (Boyce . codd normal form):

it should be in 3NF.

it should be super key.

$X \rightarrow Y$

s_id	c_id	c_name
1	101	Math101
1	102	Phy102
2	101	Math101
2	102	Phy102
2	103	chem103
3	101	Math101
3	107	chem108

prime attr [s.id, c-id]

non prime attr [course. name).

{ s_id, c_id } → course_name ✓

{ c_name } → { c_id }

{ c_id } → { coure_name } ✗

↓

not a superkey.

break till 9:24

* CRUD

C → Create

R → Read

U → update

D → delete

```
create table students (
    s_id int primary key auto_increment,
    f_name varchar(25) not null,
    l_name varchar(25)
    age tiny int,
    gender varchar(25),
    contact_no varchar(15),
    email_id varchar(30),
    gradution year date
);
```

```
insert into {table_name}
values (                    );
```

```
insert into students
values ( 1, 'mohit', 'gupta', 22, 'male', '94720', 'abc@'
                                '01|06|2023' );
```

default

you have to fill all values and also in same order.

```
insert into {table_name}
( c_name, t_name, age )
values (                    );
```

Read

```
select   {columns}
from     {table_name}
where    {conditions}
```


Operators

a = b	equal to
a <> b	does not equal
a > b	
a < b	
a ≥ b	
a ≤ b	