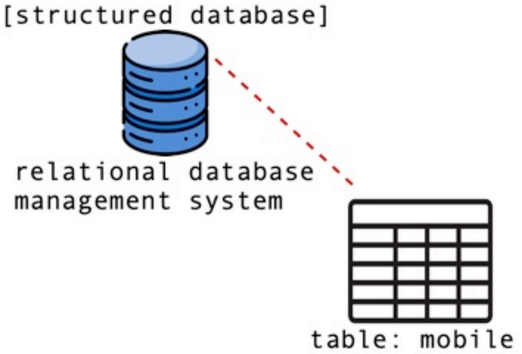
[structured database]



mobile_no	model_no	model_nm	manufacturer	network_type	operating_system	ram	processor	battery_capacity
1	A001	Iphone11	Apple	GSM	IOS	NULL	A12	5000
2	N001	Nokia1100	Nokia	GSM	NULL	NULL	NULL	1000
3	SM010	Galaxy23	Samsung	GSM/CDMA	Andriod	512	Snapdragon	6000

The RDBMS is an structured database management system, which means all the records in table has same and fixed-set of columns within them, irrespective of whether they have data or not.

For eg.. when we are storing mobiles information in the RDBMS Tables, each mobile has different attributes of data like few are smart has operating system and others are keytone based old mobiles.
Not all the mobiles has same attributes of data, but inorder to accomodate and store different types of mobiles in the RDBMS table, we need to define the table with all possible columns.

So that each record has all columns where if that mobile has the data for that column we store it otherwise will store it as NULL. Thus results in huge amount of memory wastage.

When we are building modern world applications like
1. ecommerce
2. social networking
3. feeds/news
etc
the face of data changes per each entity, so how to persist such data?

Even though RDBMS databases fails in storing/representing such data, people continued to explore different mechanisms in storing this data aspart of RDBMS database itself. one technic being used is columnar tables.

mobile

mobile_no	model_no	model_nm	manufacturer	network_type	battery_capacity
1	A001	Iphone11	Apple	GSM	5000
2	N001	Nokia1100	Nokia	GSM	1000
3	SM010	Galaxy23	Samsung	GSM/CDMA	6000

mobile_features [columnar table]

mobile_no(fk)	feature_nm	feature_value
1	operating system	ios
1	processor	A12
3	operating system	andriod
3	processor	snapdragon
3	ram	512gb

each record in this table is representing a column of the original mobile table.
In-Short: The columns of data is stored as records in this table.

select * from mobile where mobile_no = 1; returns 1 record
select * from mobile_features where mobile_no = 1; returns 2 records

wrap into object