

Practical Task # 04

Optimisation through Partitioning

Task 4b

1. Consider the figure below and perform horizontal and vertical partitioning on it by choosing the appropriate partitioning technique. Also provide the justification why have you chosen that technique?

ID	User Name	First Name	Last Name	Email Address	Phone Number	Birth Date
1	MichellekDavis	Michellek	Davis	MichellekDavis@dayrep.com	9412233379	10/10/2086
2	dallasSGood	dallas	Good	DallasSGood@teleworm.us	8329843007	30/03/1972
3	TerranceDAviles	Terrance	Aviles	TerranceDaviles@armyspy.com	7045076659	20/04/1992
4	Undan1937	Susan	Roberts	SusanKRoberts@teleworm.us	4014126825	25/12/1996
5	Nempecovest1959	Lea	McGill	LeaJMcGill@jourrapide.com	2176567191	3/6/2000
6	Mandred	Regan	Coulter	ReganECoulter@jourrapide.com	6304832993	9/3/1998
7	Beglvaing	Richard	Endres	RichardJEndres@rhyrep.com	3096848793	12/8/1988
8	Biturnight	Gregory	Melton	GregoryMMelton@dayrep.com	6304236158	6/9/1977
9	Stiong	Bernice	Stevenson	BerniceGStevenson@rhyrep.com	7736461308	3/1/2001

2. Consider a scenario of FAST University. As one of the top deemed university, every year thousands of grads pass out. Now with enormous strength comes the unexpected scale at which the online system had to work.

Each batch has around 6000 -8000 computer science students and who are split among 20–30 branches. Each one will be assigned with a unique identifier as soon as they join the university. Now whatever the student wants, be it to view his mark sheets, achievements all he need to do is just type his unique identifier in the online portal and get to see the data.

But, but there is just one big problem over here. The student records keep growing exponentially every year with every new batch and after few years or so the student

records may go upto some lakhs and every time the user searches for his/her record, the database has to scan through all the records and provide the details requested which in turn increases the latency and page load time that frustrates the user. Also whenever the semester results were announced, the load use to be like crazy that the system goes for a toss for almost an hour or two because the DB can't handle that many connections and look ups.

How university will cool down the heat created by load, and tackles this issue. What kind of identifiers or keys will be used to access the DB efficiently? How records traversing will be reduced? To answer these question design a sample DW for this scenario.