| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | | |
| --- | --- | --- | --- | --- | --- |
| final design | **Course:** | **Data Warehousing and Business Intelligence** | **Course Code:** | **DS3003** | |
| **Program:** | **BS (Data Science)** | **Semester:** | **Fall 2023** | |
| **Date:** | **Thu 14-Sep-2023** | **Total Marks:** | **10** | |
|  |  |  |  | |
| **Section:** | **BDS-5A** | **Max. Time:** | **10 min.** | |
| **Quiz:** | **1 (PDM)** |  |  | |
| \*\* | | | | |

**Q.** Consider the following normalized data structure:

SALES (saleId, storeId, saleDate, …)

SALES\_DETAIL (transactionId, saleId, itemId, itemQty, …)

Assume there are five million sales and thirty million sales details. Record length of both tables is same i.e., 100 bytes and each column of both tables (including PK/FK columns) is of same size i.e., 10 bytes.

Query: SELECT \* FROM sales S JOIN sales\_detail D ON S.saleId = D.saleId;

You are required to improve the performance of the above query using pre-join de-normalization technique. Show your de-normalized data structure schema and evaluate the increase in additional storage cost (in %age or MB) for the de-normalized data structure.

**Ans:**

**Normalizes Structure:**

**Sales: 5m \* 100 = 500m**

**Sales\_detail: 30m \* 100 = 3000m**

**Total cost: 3500m**

**Denormalized Structure:**

**30m \* 190 = 5700m**

**Increase in additional storage cost: ((5700 – 3500)/3500)\*100 = 63%**

**or**

**Increase in additional storage cost: 5700 – 3500= 2200m/1024\*1024= 2098 MB**