1. Create a standard table, called Table 1, and an EAV representation, called Table 2, of this sparse matrix. Create these tables by “hand”. You do not need to write code. You may want to consult the lecture slides. Show the actual tables:

a)Table 1 will have as headers: A, B, C, & N.

b)Table 2 will have as its headers: R, C, & M. The “C” in Table 2 has a different meaning than the “C” in Table 1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 1** | |  |  |  | **EAV-Table 2** | | |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **A** | **B** | **C** | **N** |  | **R** | **C** | **M** |  | **N : Number of Houses** |
| 2 | 1 | 3 | 23 |  | 1 | A | 2 |  | **A: Area in 1000’s of square feet** |
| 1 | 3 | 3 | 29 |  | 2 | B | 1 |  | **B: Number of Bathrooms** |
| 2 | 1 | 1 | 17 |  | 3 | C | 3 |  | **C: Cost in $100,000** |
|  |  |  |  |  | 4 | N | 23 |  |  |
|  |  |  |  |  | 5 | A | 1 |  |  |
|  |  |  |  |  | 6 | B | 3 |  |  |
|  |  |  |  |  | 7 | C | 3 |  |  |
|  |  |  |  |  | 8 | N | 29 |  |  |
|  |  |  |  |  | 9 | A | 2 |  |  |
|  |  |  |  |  | 10 | B | 1 |  |  |
|  |  |  |  |  | 11 | C | 1 |  |  |
|  |  |  |  |  | 12 | N | 17 |  |  |

1. Change the schema of the data in item 1 above by changing the EAV table, called Table 2 in the following way: New values will represent Cost per Area (CPA). You can calculate CPA from the existing information. Modify this table by “hand”. You do not need to write code.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 3** | |  |  |  | **EAV-Table 4** | | |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **A** | **B** | **C** | **N** |  | **R** | **C** | **M** |  | **N : Cost Per Area (CPA)** |
| 2 | 1 | 3 | 1.5 |  | 1 | A | 2 |  | **A: Area in 1000’s of square feet** |
| 1 | 3 | 3 | 3 |  | 2 | B | 1 |  | **B: Number of Bathrooms** |
| 2 | 1 | 1 | 0.5 |  | 3 | C | 3 |  | **C: Cost in $100,000** |
|  |  |  |  |  | 4 | N | 1.5 |  |  |
|  |  |  |  |  | 5 | A | 1 |  |  |
|  |  |  |  |  | 6 | B | 3 |  |  |
|  |  |  |  |  | 7 | C | 3 |  |  |
|  |  |  |  |  | 8 | N | 3 |  |  |
|  |  |  |  |  | 9 | A | 2 |  |  |
|  |  |  |  |  | 10 | B | 1 |  |  |
|  |  |  |  |  | 11 | C | 1 |  |  |
|  |  |  |  |  | 12 | N | 0.5 |  |  |

*--Exercise 5*

*-- Write SQL to multiply a matrix in the sparse matrix format by 7. Use Matrix1 as your example matrix.*

*-- The result of the multiplication is in the sparse matrix format.*

*-- Do not use insert statements or create any new tables or views.*

*-- Answer for Exercise 5:*

*--Select without the multiplication.*

select rowid, columnid, value from matrix1 order by rowid, columnid;

*--Multiply sparse matrix value by 7*

update matrix1 set value = value \* 7;

commit;

*--Select after the multiplication*

select rowid, columnid, value from matrix1 order by rowid, columnid;

*--Exercise 6*

*-- Write SQL to transpose a matrix in the sparse matrix format. Use Matrix2 as your example matrix.*

*-- The result of the transposition will be in the sparse matrix format.*

*-- Do not use insert statements or create any new tables or views.*

*-- Answer for Exercise 6:*

*--Selecting M \* N matrix*

select rowid, columnid, value from matrix2 order by rowid, columnid;

*--Transposing the matrix to N \* M by switching rowid with columnid and columnid with rowid*

update matrix2 set rowid = columnid, columnid = rowid where rowid != columnid;

commit;

*--selecting the transposed matrix.*

select rowid, columnid, value from matrix2 order by rowid, columnid;

*--Exercise 7*

*-- Add two Matrices of same dimensions.*

*-- Write SQL to create a sparse matrix representation that is the addition of the matrices represented by Matrix1 and Matrix3*

*-- The result is in the sparse matrix format.*

*-- Do not use insert statements or create any new tables or views.*

*-- Answer for Exercise 7:*

SELECT Matrix1.RowID1 AS RowID1, Matrix1.ColumnID AS ColumnID, (Matrix1.Value + Matrix3.Value) AS Value

FROM Matrix1, Matrix3 WHERE (Matrix1.RowID1 = Matrix3.RowID1) and (Matrix1.columnid = Matrix3.columnid);

--I am not doing further validation in the SQL to ensure that Matrix1 and Matrix3 are of the same dimensions as the matrices defined in the exercise are of the same dimensions. (4 \* 3)