Multidimensional & Bitmap Indexes

Section 14.2 and 14.4

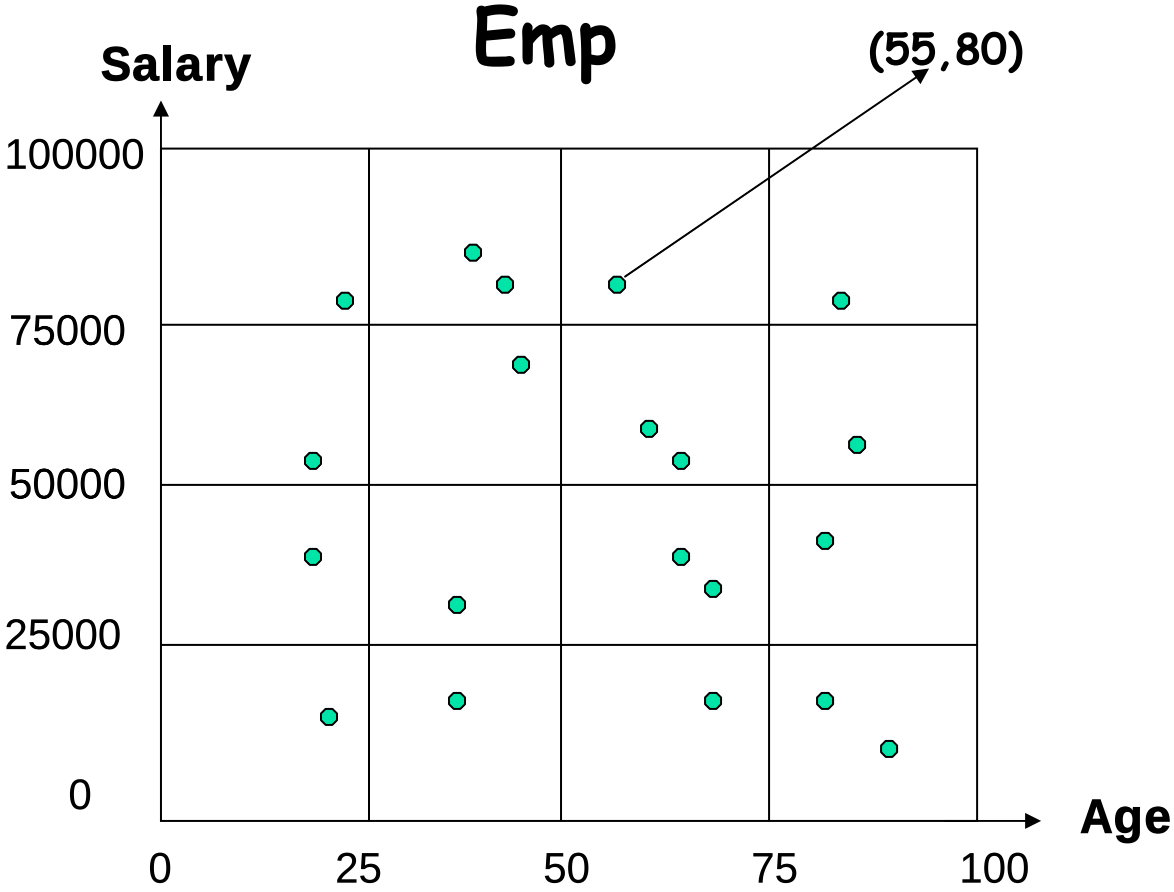
Lecture outline

- Multidimensional Data
- Grid Files
- Bitmap Indexes
- Summary
- Reading List

- Multidimensional Data

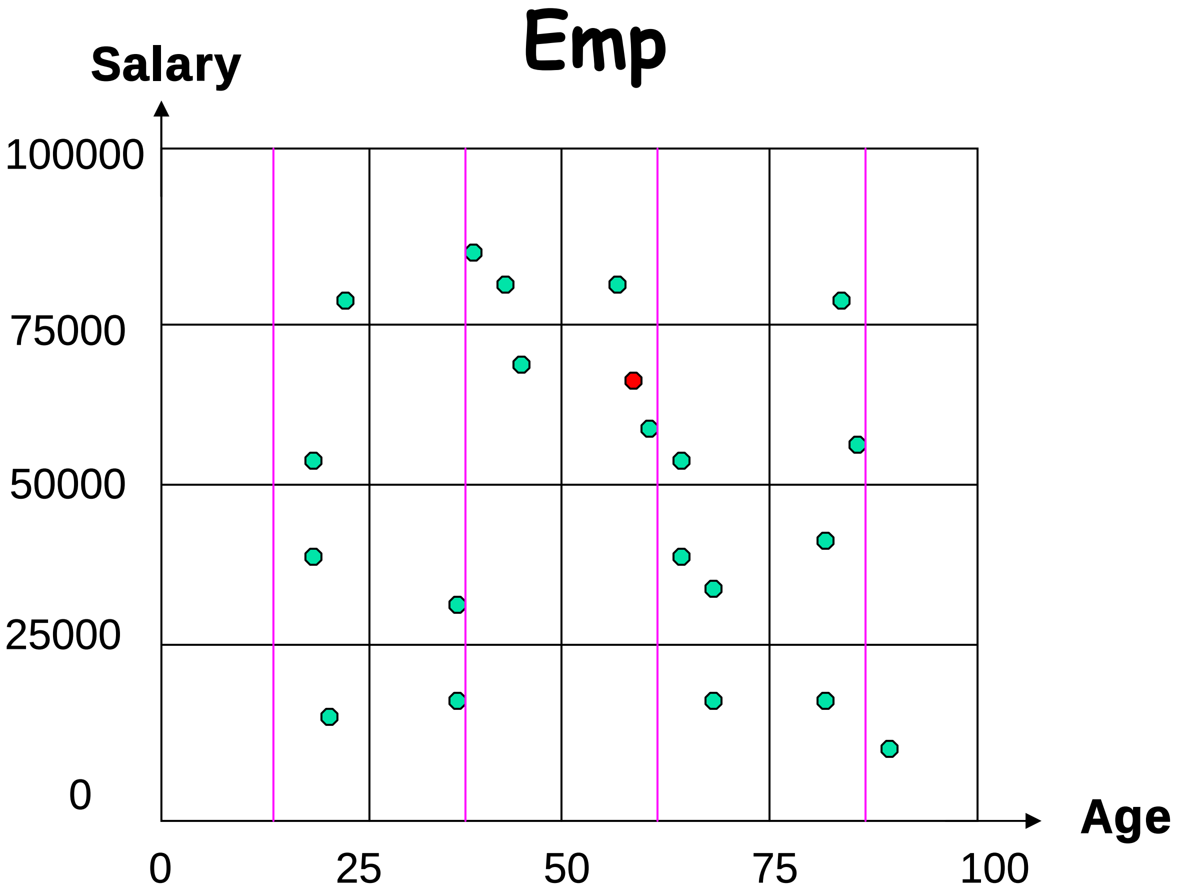
- Visualizing data as stored in *n* dimensional space
- Support queries that are not common in SQL
- Applications:
    - Geographical information system
    - Data warehouse
        - Data cubes
    - Multidimensional queries
- Examples:
    - Grid file
    - BANG file
    - BV-Trees

- Grid files



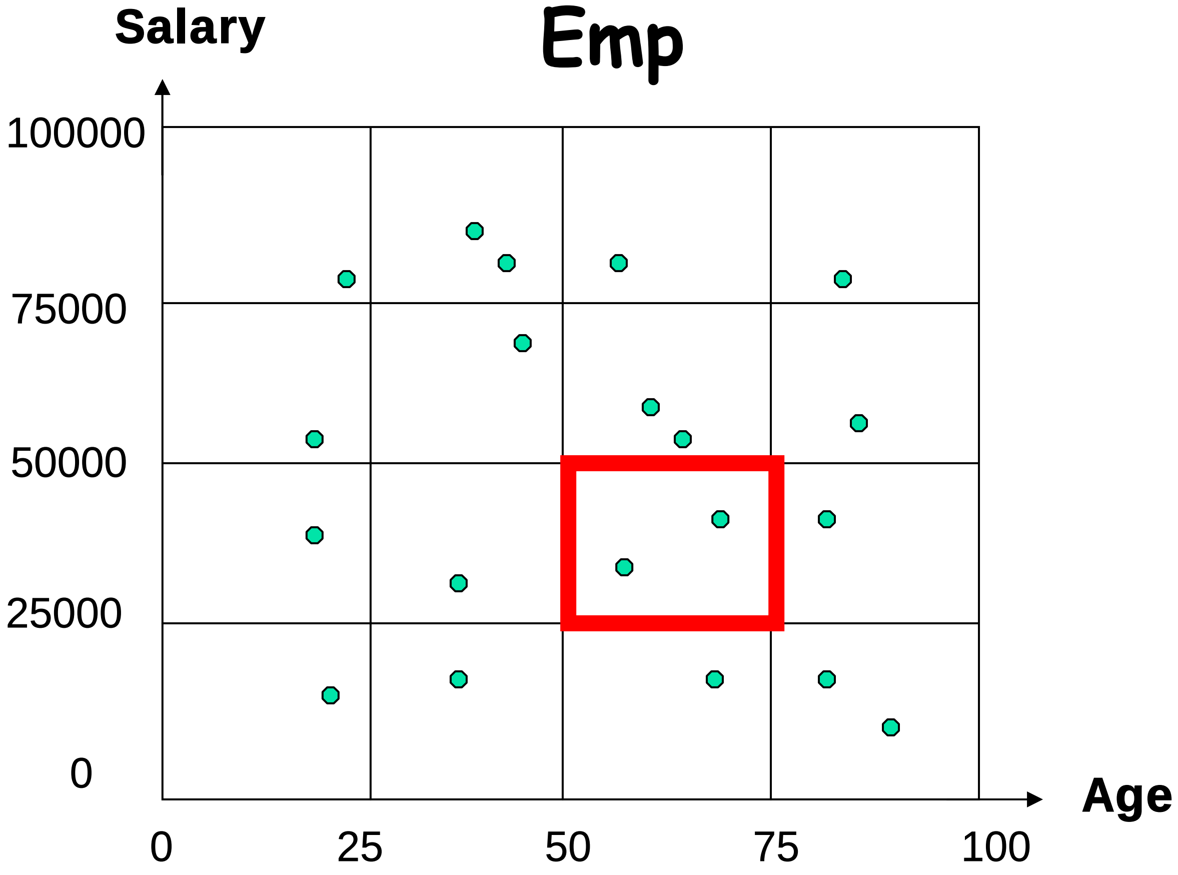
-- Insertion

```
INSERT INTO Emp
VALUES(60,70000);
```

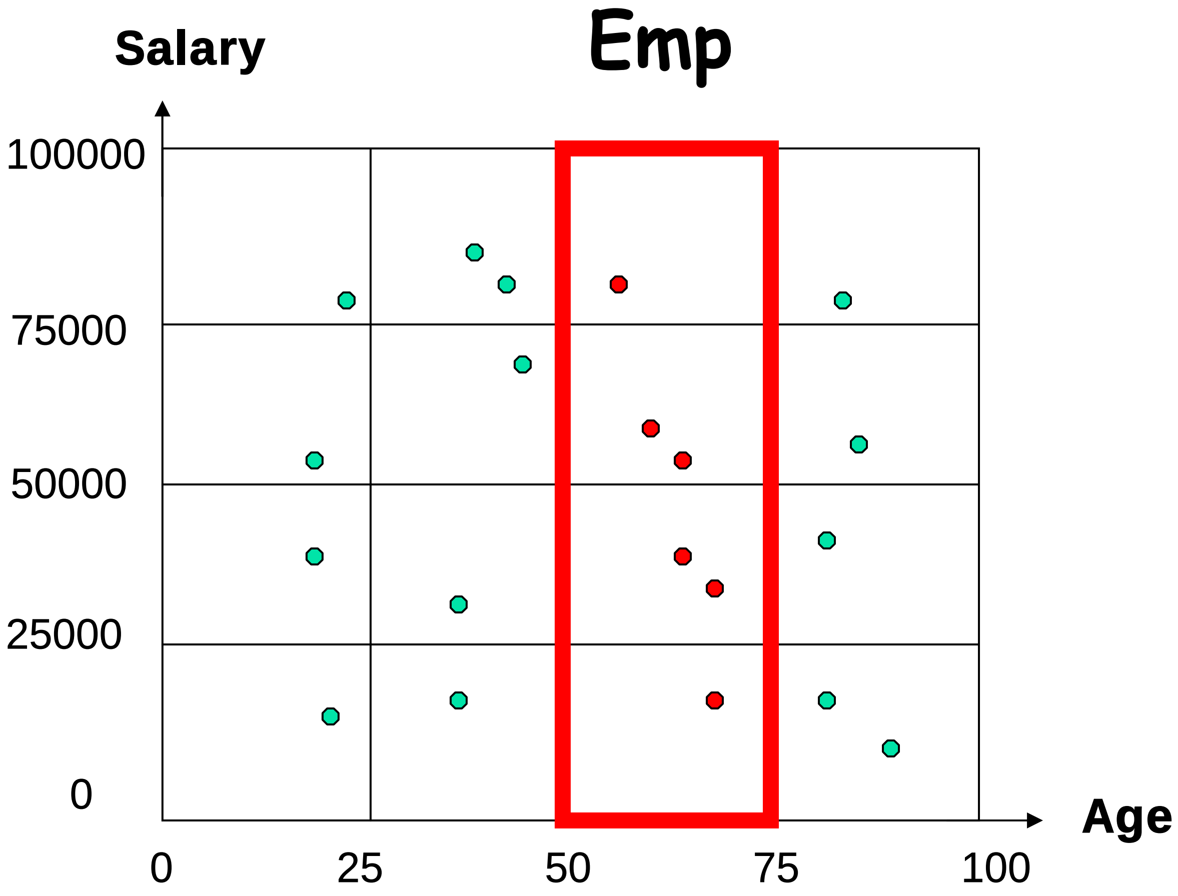


-- Exact Match Query

SELECT *
FROM Emp
Where age = 60
AND salary = 40000



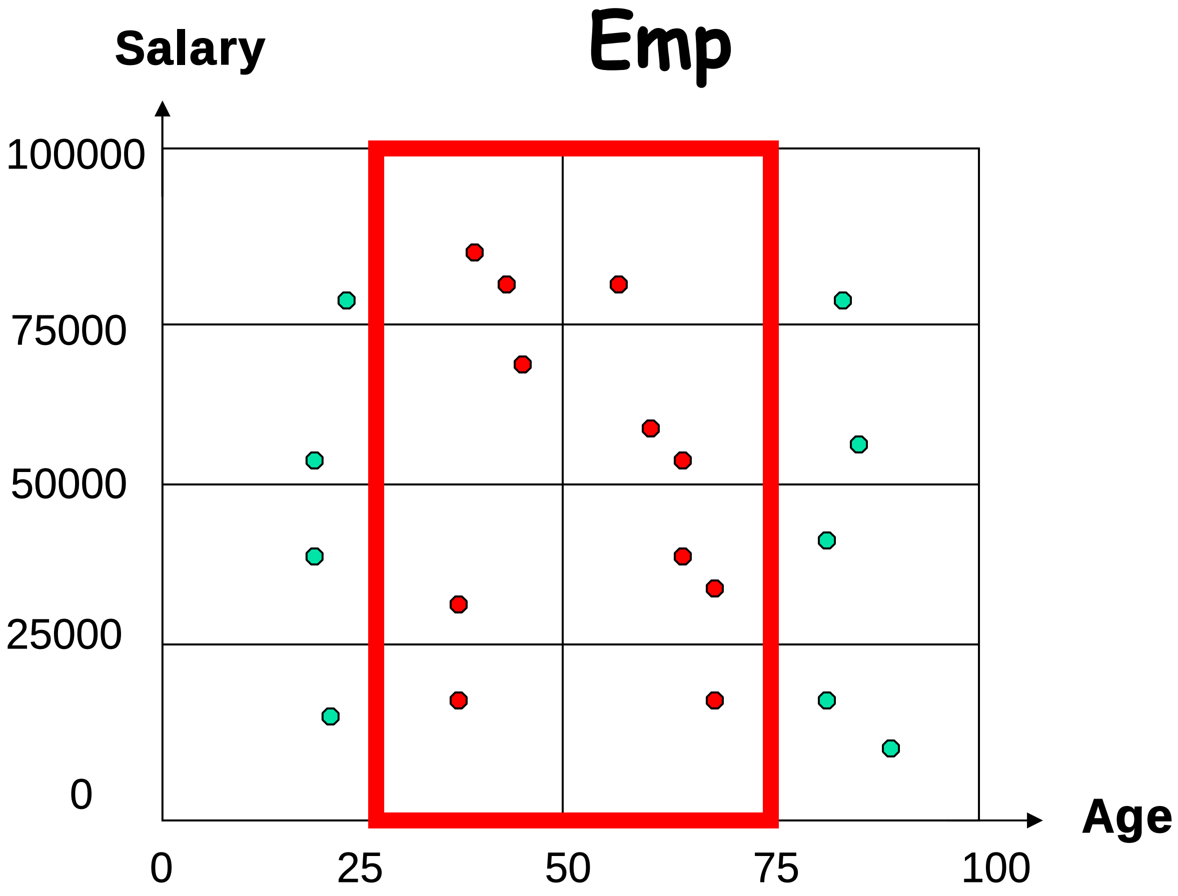
-- Partial Match Query



SELECT *
FROM Emp
Where age = 60

-- Range Query

SELECT *
FROM Emp
Where age < 60
AND age > 30



- Bitmap Indexes

- What is Bitmap Index
- Motivation for Bitmap Indexes
- Compressed Bitmaps
- Managing Bitmap Indexes
- Summary
- Reading List

-- What is Bitmap Index

- A bitmap index for attribute A of relation R is:
  - A collection of bit-vectors
  - The number of bit-vectors = the number of distinct values of *A* in R.
  - The length of each bit-vector = the cardinality of R.
  - The bit-vector for value v has 1 in position i, if the ith record has v in attribute *A*, and it has 0 there if not.
- Records are allocated permanent numbers
- There is a mapping between record numbers and record addresses.

-- Example

- Assume relation R with
  - 2 attributes A and B.
  - Attribute A is of type Integer and B is of type String.
  - 6 records, numbered 1 through 6 as follows:

	A,	B
1.	30,	foo
2.	30,	bar
3.	40,	baz
4.	50,	foo
5.	40,	bar
6.	30,	baz

  - A bit map for attribute B is:

value	Vector
foo	100100
bar	010010
baz	001001

-- Motivation for Bitmap Indexes

- Very efficient when used for partial match queries.
- The offer the advantage of buckets
  - Where we find tuples with specified in several attributes without first retrieving all the record that matched in each of the attributes.
- They can also help answer range queries

-- Compressed Bitmaps

- Assume:
  - The number of records in R are n
  - Attribute A has m distinct values in R
- The size of a bitmap index on attribute A is $m*n$.
- If m is large, then the number of 1's will be around $1/m$.
  - Opportunity to encode
- A common encoding approach is called run-length encoding.

--- Run-length encoding

- Represents runs
  - A run is a sequence of i 0's followed by a 1, by some suitable binary encoding of the integer i.

- A run of i 0's followed by a 1 is encoded by:
  - First computing how many bits are needed to represent i, Say j
  - Then represent the run by j-1 1's and a single 0 followed by j bits which represent i in binary.
  - The encoding for i = 1 is 01. j = 1
  - The encoding for i = 0 is 00. j = 1

- We concatenate the codes for each run together, and the sequence of bits is the encoding of the entire bit-vector

--- Run-length Encoding: Example

- Let us decode the sequence 11101101001011
- Staring at the beginning (left most bit):
  - First run: The first 0 is at position 4, so
    - j = 4,
    - i = 13
  - Second run: 001011
    - j = 1
    - i = 0
  - Last run: 1011
    - J = 1
    - I = 3
- Our entire run length is thus 13,0,3, hence our bit-vector is:

0000000000000110001

-- Managing Bitmap Indexes ...

- Finding bit vectors
  - Create secondary key with the attribute value as a search key
    - Btree
    - Hash
- Finding records
  - Create secondary key with the record number as a search key.

… -- Managing Bitmap Indexes

- Handling modification
  - Assume record numbers are not changed
    - Deletion
      - Tombstone replaces deleted record
      - Corresponding bit is set to 0
    - Insertion
      - Record assigned the next record number.
      - A bit of value 0 or 1 is appended to each bit vector
      - If new record contains a new value of the attribute, add one bit-vector.
    - Modification
      - Change the bit corresponding to the old value of the modified record to 0
      - Change the bit corresponding to the new value of the modified record to 1
      - If the new value is a new value of A, then insert a new bit-vector.

- Summary

- Multidimensional Data
- Grid Files
  - Insertion
  - Different kinds of queries
- Bitmap Index
  - Compression
  - Management

- Reading List

- Section 14.2 and 14.4 of GUW

END