

# INFO20003 Tutorial – Week 6

(Tutorial: Storage and Indexing)

## Objectives:

This tutorial will cover:

- I. Storage and Indexing review – key concepts with examples – 30 mins
- II. Exercises – 30 mins

## Key Concepts:

**NOTE for students:** This is a brief summary of some of the concepts taught in lecture 10. The lectures contain detailed content related to these and many more concepts. These notes should be considered quick revision instead of a sole resource for the course material.

- Files, pages and records
- File organisations
  - Heap file organisation
  - Sorted file organisation
  - Index file organisation
- What is an index?
- Hash-based indexing
- B-tree indexes

## Exercises:

### 1. Choosing an index

You are asked to create an index on a suitable attribute. What are the important aspects you will analyse to make this decision? To get you started, the following might help you by providing scaffolding to the discussion:

- Primary vs. secondary index
- Clustered vs. unclustered index
- Hash vs. tree indexes

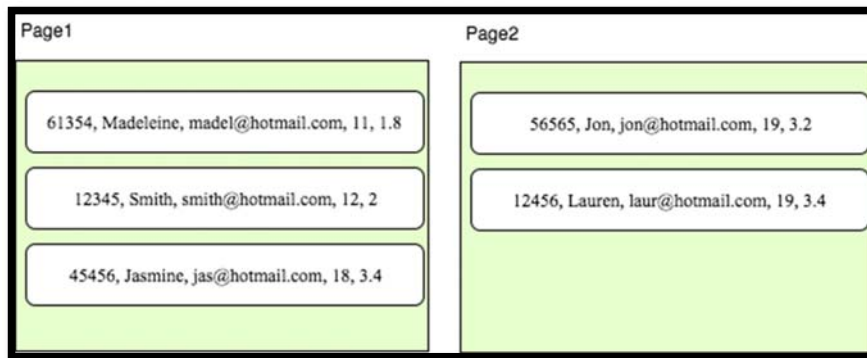
*Continued over page*

## 2. Data entries of an index:

Consider the following instance of the relation Student (SID, Name, Email, Age, GPA):

| SID   | Name      | Email  | Age | GPA |
|-------|-----------|--|-----|-----|
| 61354 | Madeleine | <a href="mailto:madel@hotmail.com">madel@hotmail.com</a> | 11  | 1.8 |
| 12345 | Smith     | <a href="mailto:smith@hotmail.com">smith@hotmail.com</a> | 12  | 2.0 |
| 45456 | Jasmine   | <a href="mailto:jas@hotmail.com">jas@hotmail.com</a>     | 18  | 3.4 |
| 56565 | Jon       | <a href="mailto:jon@hotmail.com">jon@hotmail.com</a>     | 19  | 3.2 |
| 12456 | Lauren    | <a href="mailto:laur@hotmail.com">laur@hotmail.com</a>   | 19  | 3.4 |

As you can see the tuples are sorted by age and we are assuming that the order of tuple is the same when stored on disk. The first record is on page 1 and each page can contain only 3 records. The arrangement of the records is shown below:



Show what the *data entries* of the index will look like for:

- An index on Age
- An index on GPA

*Continued over page*

**3. Consider the following relations:**

Employee (EmployeeID, EmployeeName, Salary, Age, DepartmentID) <sup>FK</sup>

Department (DepartmentID, DepartmentBudget, DepartmentFloor, ManagerID) <sup>FK</sup>

In the database, the salary of employees ranges from AUD10,000 to AUD100,000, age varies from 20-80 years and each department has 5 employees on average. In addition, there are 10 floors, and the budgets of the departments vary from AUD10,000 to AUD 1million.

Given the following two queries frequently used by the business, which index would you prefer to speed up the query? Why?

a. **SELECT** DepartmentID  
**FROM** Department  
**WHERE** DepartmentFloor = 10  
**AND** DepartmentBudget < 15000;

- A) Clustered Hash index on DepartmentFloor
- B) Unclustered Hash Index on DepartmentFloor
- C) Clustered B+ tree index on (DepartmentFloor, DepartmentBudget)
- D) Unclustered hash index on DepartmentBudget
- E) No need for an index

b. **SELECT** EmployeeName, Age, Salary  
**FROM** Employee;

- A) Clustered hash index on (EmployeeName, Salary)
- B) Unclustered hash Index on (EmployeeName, Age)
- C) Clustered B+ tree index on (EmployeeName, Age, Salary)
- D) Unclustered hash index on (EmployeeID, DepartmentID)
- E) No need for an index