MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

DEPARTMENT

OF

COMPUTER ENGINEERING

CNG 351
Data Management and File Structures
Assignment 5

(5% of actual grade)

DUE DATE: 8 January, Sunday, 2022, 23:55 (Cyprus Time)

PURPOSE

This assignment aims to help you revise the last part of the course which is file structures and indexing. You will mainly have questions that will help you revise the following topics: Disk Storage, Basic File Structures and Hashing, and Indexing Structures for Files.

IMPORTANT RULES

1. Please make sure that your solutions are clearly explained.
2. Please make sure that your report is readable.
3. When you create an index, please clearly show all the steps.
4. Create a PDF file for your solution, and upload only one PDF file to ODTUClass, one team member is enough to upload the solution.
5. Your PDF file can include scans/images of handwritten solutions.
6. Assignments will be completed by a team of two people that was formed for the previous assignment. if there is a problem with your partner, please let us know.
7. Please submit a report that includes a cover page with the team details including their names/surnames and also student IDs.

GRADING

This assignment has four questions and the overall grading will be as follows:

1. Question 1 (Hash Index): 20 points;
2. Question 2 (Expandable Hash Index): 25 points;
3. Question 3 (B+ Tree): 25 points;
4. Question 4 (Clustering and Secondary Index): 25 points;
5. Report: 5 points. A good report means type written, complete (every section fulfilled), with clear explanations in English (where relevant), and submitted via ODTUCLASS as one combined PDF document. The first page of the report must be a title page which should clearly state team details and assignment number. Each use case should also include Assumptions clearly written. The footer of all subsequent pages should be numbered in the format x of y (eg 2 of 6), etc.

Question 1 (20 pts)

Consider the following relation that includes information about several subscription details in our system. This table keeps track of the subscription_id (which is unique), subscription_type (type of the subscription), monthly_price (the monthly amount paid for this subscription), and payment_type (type of the payment).

subscription_id	**subscription_type**	**monthly_price**	**payment_type**	**payment_date**
501	high	450	credit-card	20/01/2018
502	medium	350	paypal	11/12/2022
503	low	150	bank-transfer	30/8/2018
504	high	450	credit-card	3/1/2020
505	low	450	credit-card	15/10/2022
506	medium	350	paypal	30/7/2022
507	low	450	credit-card	3/6/2017

a) Imagine that you are using a Hash file at the back end to store the data given in this relation. Assume that the hash function h of this file takes subscription_id mod 5. For example, Let's look at the first tuple in the relation. h(501) mod 5 = 1. Assume that all hash values fit in one block, and only one record can fit in one block. Organise the data in this given relation in such a Hash file and show the resulting structure. Please note that for collision, the system uses a chained overflow approach. Please clearly show how your data is inserted and all the details of your calculations. You need to clearly show all your decisions.

b) Imagine that you are using another Hash file at the back end to store the data given in this relation. This time, assume that the hash function h takes both the subcription_id and also year of payment_date and calculates mod 5. For example, let's look at the first tuple in the relation. h (501+2018) mod 5 = 4. Assume that all hash values fit in one block, and only one record can fit in one block. Organise the data in this given relation in such Hash file and show the resulting structure. Please note that for collision, the system uses chained overflow approach. Please clearly show how your data is inserted and all the details of your calculations. You need to clearly show all your decisions.

c) Now, look at the file organization resulting from the above two questions. (1) Which hash function would you prefer to use and (2) why? Please clearly explain why you think one is better than the other. It is not enough just to say that one is better than the other, you need to justify your answer.

Question 2 (25 pts)

Imagine that your table in Question (1) now has more data. This time imagine that you are using an Extensible hashing to insert data. Load the records from subscription_id: 60, 64, 65, 67, 68, 72, 74, 76, 80, 81, 8, 86 and 116. Show the

structure of the directory at each step, and the global and local depths. Use the hash function h(K)= K mod 29. Imagine that each bucket is one disk block and holds 3 records. Please make sure that the depth details of all the buckets are given and also your step-by-step insertion is clearly shown. **Please use leftmost bits to put data into blocks!**
Example:
Assume the value is 001111.
You need to check from the leftmost bits → $\underline{0}01110$.

Question 3 (25 pts)

Construct a B+-tree for the following set of key values: (122, 169, 207, 201, 158, 184, 167, 175, 153, 159, 125, 176, 160 and 123)

- Assume that the tree is initially empty. Construct B+-trees for the cases where the number of pointers that will fit in one node is 3. When you construct the tree it is not enough to show the final tree structure. You need to show the step by step creation and also explanation/justification of each step.
- For the B+-tree of (a), show the steps involved in the following queries:
  - Find records with a search-key value of 175.
  - Find records with a search-key value between 159 and 167, inclusive.
- Show the form of the tree after deleting key values 169, 175, 122 and 160.

Question 4 (25 pts)

Consider the relation given below which stores details of faculty members in a university and answer the following questions accordingly. Assume each block can contain 4 records and the original data file is sorted by the primary key user_ID.

user_ID	name	surname	username	group_ID	dob
204	Nil	Heinritz	nheinritz0	100	6/24/97
10	Salim	Sloan	ssolan5	100	6/24/97
147	Rozen	Baum	mrozenbaum1	101	3/6/90
938	David	Irevie	irevie3	102	5/22/98
951	Hobard	Seabright	hseabright1	103	5/11/98
716	Shelly	Greendale	sgreendale2	100	7/24/78
414	Etienne	Mutlow	emutlow3	100	6/24/79
358	Jon	Standall	jstandall7	101	3/6/80
972	Man	Cathel	mcathel8	102	5/15/81
881	Kaila	Lembrick	klembrick0	103	5/22/82
986	Xena	Bosomworth	xbosomworth1	102	6/22/83
971	Hena	Bosomworth	henamworth1	100	6/10/80

1. Show the structure of the database when the given table is indexed using a clustering index with group_ID.

2. Show the structure of the database when the given table is indexed using a secondary key username.