Assignment 2   
DataBase Management – Theory   
Spring 2018

***laverty@rmu.edu***

Contents

[**Introduction 4**](#_Toc482355463)

[**Researching Your Answers 4**](#_Toc482355464)

[**A Little Advice Before you start 5**](#_Toc482355465)

[**Demonstrating Knowledge and Increased Penalties for Irrelevant Answers 6**](#_Toc482355466)

[**You Must Submit YOUR Answer in this Original Word Document to Blackboard 6**](#_Toc482355467)

[**You Must RENAME this Original Word Document to Include your LAST NAME 7**](#_Toc482355468)

[**NEVER submitted an Assignment as an Email Attachment 7**](#_Toc482355469)

[**ONLY Submit a FINAL Version of ALL Assignment 7**](#_Toc482355470)

[**Requests to Clear Previously Assignments for Re-Submission 7**](#_Toc482355471)

[**Submitting Late Assignments 7**](#_Toc482355472)

[**Academic Integrity and Plagiarism 8**](#_Toc482355473)

[**How to complete Content Questions 9**](#_Toc482355474)

[**1.0 Introduction to the System z Architecture 10**](#_Toc482355475)

[**1.1 Introduction to Relational Model Terminology 10**](#_Toc482355476)

[**1.2 Questions - Basic Relational Concepts 11**](#_Toc482355477)

[**1.3 Questions - Degree and Cardinality 12**](#_Toc482355478)

[**1.4 Questions - Relational Constraints 13**](#_Toc482355479)

[**1.5 Questions - Relational Views 13**](#_Toc482355480)

[**2.0 - Relational Algebra Concepts 15**](#_Toc482355481)

[**2.1 Questions - Projection and Selection 15**](#_Toc482355482)

[**2.2 Questions - Unions, Intersection and Difference 17**](#_Toc482355483)

[**2.3 Questions - Relational Inner Joins, Equi-Joins, Left Joins, Right Joins, and Self Joins 19**](#_Toc482355484)

[**3.0 Data Structures 22**](#_Toc482355485)

[**3.1 Introduction to Data Structures 22**](#_Toc482355486)

[**3.2 Stacks 22**](#_Toc482355487)

[**3.2.1 Advantages and Disadvantages of Stacks 22**](#_Toc482355488)

[**3.2.2 Uses of Stacks 23**](#_Toc482355489)

[**3.3 FIFO Queues 24**](#_Toc482355490)

[**3.3.1 Uses of FIFO Queues 25**](#_Toc482355491)

[**3.3.2 Priority Queues 25**](#_Toc482355492)

[**3.3.3 Uses of Priority Queues 25**](#_Toc482355493)

[**3.3.4 Advantages and Disadvantages of Queue Data Structures 25**](#_Toc482355494)

[**3.4 Ordered Linked List 26**](#_Toc482355495)

[**3.4.1 Advantages and Disadvantages of an Ordered Link List 26**](#_Toc482355496)

[**3.5 Two-Way Link List 27**](#_Toc482355497)

[**3.6 Circular List 27**](#_Toc482355498)

[**3.7 Inverted Files or Pointers 28**](#_Toc482355499)

[**3.8 Binary Tree 29**](#_Toc482355500)

[**3.8.1 Advantages and Disadvantages of Binary Tree 30**](#_Toc482355501)

[**3.9 Bn-Tree 30**](#_Toc482355502)

[**3.9.1 Uses of Bn Tree 30**](#_Toc482355503)

[**3.9.2 Advantages of Bn Tree 31**](#_Toc482355504)

[**3.9.3 Oracle b-tree index 31**](#_Toc482355505)

[**3.10 Hash Tables 32**](#_Toc482355506)

[**3.10.1 Hash Collision Strategies 33**](#_Toc482355507)

[**3.10.2 Uses of a Hash Table 33**](#_Toc482355508)

[**3.10.3 Advantages of a Hash Table 33**](#_Toc482355509)

[**3.10.4 Disadvantages of a Hash Table 33**](#_Toc482355510)

[**3.10.11 Factors affecting the performance of Hashing 34**](#_Toc482355511)

[**3.11 Indexed Sequential (VSAM) 34**](#_Toc482355512)

[**4.0 Questions – Data Structures 36**](#_Toc482355513)

**Enter your Name Here 🡺 Puja Ghosh**

# Introduction

## Researching Your Answers

Most requirements of this assignment will require you to research answers from your text book (**you must read the text book to get some answers**), from the Internet, from a video or any other reasonable source. Many Internet sources, video links, text book and Blackboard presentations are provided in this document to help you START your research.

Assignments are always a great place to read and reference your text book. May students assume that they should start by reading the text book. While this reading the text book is ALWAYS at great idea, the following assignment questions may help you focus on what is important in this course***. One strategy is to review an assignment question for important keywords, e.g., multitasking, Procedure Division, virtualization, etc. Then look for those key words in either 1) your text book "detailed" table of contents in the beginning of the book, or 2) in the index at the end of the book. Some text books have key word glossaries at the end of the chapter or end of the book.***

Every semester the Internet sources, e.g., a Google keyword search, or Internet video, e.g., a YouTube keyword search may be improved. Better students start with research sources provided in this assignment, and then search for improved or updated answers. ***While your objective may be to demonstrate your knowledge to EARN an excellent grade on this assignment, better students are always looking towards the future to impress internship and career recruiters for those interesting and high entry-level salaries.*** Employers don't pay you for a grade in any single class or assignment. Employers will pay you BIG MONEY for demonstrated knowledge or skills. Hopefully, this assignment will prepare you.

A single research source maybe very incomplete or the format or the content may not be appropriate for some required answers. Some students do not READ the text book or review the Blackboard presentations provided to you in Blackboard. Some answers are only available from Blackboard presentations or update documents. When assignment requirements may be only answered by viewing a YouTube video, you will be notified.

You may cut-and-paste answers whenever appropriate. You MUST synthesize your answers to include multiple sources. I would be impressed that you consult a Google image search and paste and appropriate image to supplement your answer.

While I permit cut-and-paste, I do expect you use your own words so that you:

1. Organize the answer

2. Demonstrate that you have read what you have cut-and-pasted

3. Use any means that clearly displays that you have gained knowledge.

## A Little Advice Before you start

There is NO requirement to read each reference link or view each video in detail. Some of the links will have overlapping content. Some links will provide more information than the question requirements, but employers consider these topics to be minimal knowledge of a RMU CIS graduate.

It is recommended to visit each reference link and overview the content. Then read each question and return to each reference link or video as needed. You may supplement your answers with content from your text book by using a question keyword and looking up in the glossary or index in the back of the book. PDF text books can be easily searched for keywords.

It is not required to read your text books before working on this assignment unless specified in a requirement. Text book contents are used to support quiz questions (which contain the answers), which are used on the tests. The reference links and videos are excellent resources.

The topics presented in the course assignments have been highly recommended by two or more employers who recruit RMU CIS students as minimum computer hardware, operating system, and application development knowledge. Employers expect that students should be able to present one or two sentences of the majority of keywords applicable to job requirements in a face-to-face interview.

On the other hand, each assignment is allocated 100 points out of a total of 1000 points. The number of questions or hands-on activity on each assignment varies. Assume that an average assignment has 50 requirement questions. This means that a requirement may be worth 2 points towards your final grade. The bottom line is that missing a few questions will have little effect of your assignment or final grade. Not completing an assignment will generally decrease your final grade by at least a letter grade. Do not waste time on the small problems.

## Demonstrating Knowledge and Increased Penalties for Irrelevant Answers

If you can DEMONSTRATE your knowledge of the topic for the requirement there will be no penalty for your answer. It is not the intention of this assignment to be "not picky". Parital credit will be awarded as appropriate.

If you cut-and-paste and pray that your instructor will not read your inappropriate and irrelevant answer, the question will be penalized by increasing the deduction points beyond the original requirement points. The instructor hates irrelevant cut-and-paste BS, or answers that appears that the student is guessing and hopes the instructor does not read the answer.

***The instructor reserves the right to increase the penalty for any submitted question or assignment that may be construed as "wasting the instructor's time".*** Therefore, a four-point requirement may be penalized as six-points (two additional point penalty for wasting the instructor's time). Sometimes blank answers will earn you’re a high assignment grade than BS answers. For example, a submitted 100-point assignment may be penalized as 125 minus points when your final grade is calculated for any assignment that should have never been submitted in the first place.

## You Must Submit YOUR Answer in this Original Word Document to Blackboard

***This Assignment Word Document will contain hidden markers that may be used to detect plagiarism and provide an audit trail of those who may have modified the Word document.***  Many students in my classes work very hard to complete and learn from their assignments. It is not fair to those students who have professionally demonstrated their knowledge to receive the same grade as those who have plagiarized their assignments

**You MUST answer ALL requirement in this Word document and ONLY THIS Word Document. You MAY NOT use or edit any other word processor, except any version of Microsoft Word.**

**Do not use GOOGLE DOCS or Open Office DOCX files at any time. If you use any other Word Processor you will be assigned ZERO credit.**

**If you do not have a copy of WORD**, you may use VMWARE VIEW (available from the RMU website) to access a virtual lab computer which contains any software needed for this course.

<http://www.rmu.edu/web/cms/departments-offices/administration-services/it/Pages/vmware-view.aspx>

NEVER STORE ANY DOCUMENTS ON THE DESKTOP OF VMWARE VIRTUAL COMPUTER. You will lose your document. It is preferable to store your documents on RMU Drive U: If necessary you can email the document to yourself.

## You Must RENAME this Original Word Document to Include your LAST NAME

**YOU MUST enter your name in the beginning of this document as provided and "Save As" this document using a new name that starts with your LAST NAME, assignment number and semester, e.g., Jones Assignment 1 Summer 2018.docx**

If you do not rename your document, your assignment will be penalized by 10%.

## NEVER submitted an Assignment as an Email Attachment

All assignments are to be submitted to the instructor by using the Assignment Link in the Blackboard system. Assignments submitted as an email attachment will NOT be graded. THE INSTRUCTOR NEVER ACCEPTS ANY ASSIGNMENT AS AN EMAIL ATTACHMENT FOR ANY REASON.

## ONLY Submit a FINAL Version of ALL Assignment

Never submit an incomplete assignment for grading. Only submit your final version of ALL assignment documents for grading. You can only submit an Assignment once.

## Requests to Clear Previously Assignments for Re-Submission

If you make an error submitting an assignment, you must contact the instructor to clear your previous assignment submission. If you made an error on any assignment, you may request that the previous assignment submission be cleared so that you may resubmit the assignment again. Please only submit a completed assignment.

## Submitting Late Assignments

While the assignments have a recommended due date, the instructor does not penalized your assignment grade if you are slightly late. Please do not send the instructor an email if you are going to submit your assignment late. The instructor is flexible and assumes you have a good excuse. After you are more than two-weeks, late the instructor does reserve to penalize the assignment or not accept the assignment if this late submission is unfair to other students enrolled in the course who had completed their assignments on time.

It has been the experience of the instructor that students who are excessively or consistently late, may ask a friend to provide them a copy of their assignment, which will violate the RMU Academic Integrity Policy. (Please carefully read the next section!) ***If a friend asks you for a copy of your assignment "to get an idea what the instructor wants", you are risking a zero assignment grade, an F final grade, or a RMU Academic warning or suspension.***

## Academic Integrity and Plagiarism

When an instructor has possession of an electronic document, it is easy to detect plagiarism. Microsoft Word provides a variety of FREE anti-plagiarizing tools. The content of your submitted Assignment WORD document will be COMPARED to each other student who has submitted this assignment in the current class or any previous class as time permits. ***The content of each student's assignment may NOT be copied from any other current or past student enrolled in this class. Each assignment is to be prepared by ONE student. Assignments are NOT a group-prepared assignment.***

Some students may attempt to SAVE AS another student's completed assignment and rename it using their name. Some students may attempt to Cut-and-Paste answers from one student's assignment document to another student's assignment document. But as time permits, the forensic tools used to compare ALL student's assignments with other assignment will often detect anomalies which will provide absolute proof of plagiarism. ***On-ground tests may be used to compare the student’s knowledge to performance on assignments. All acts of plagiarism and forensic data will be submitted the RMU Academic Integrity Board to determine university-wide penalties, such as grade penalties, warnings, suspension, and change of a previous course grade for previous course students. All current and previous students involved in the plagiarism may be affected RMU Academic Integrity Board.***

***If a friend asks you for a copy of your assignment "to get an idea what the instructor wants", you are risking a zero assignment grade, an F final grade, or a RMU Academic warning or suspension. You are responsible to protect your assignment Word Document.***

***You, however, may discuss assignment requirements, provide research assistance, assist other students to debug programs or other hands-on-requirements, tutor students, or provide other advice that may assist the students in acquiring knowledge. But the actual preparation of an individual assignment must have been completely prepared by the student who submitted the assignment. Sections of the assignments may be copied from the internet as per the individual assignment's directions. Please contact the instructor if you need assistance interpreting this RMU Academic Integrity Policy. (Ref.16-1.)***

Many believe that if you a "stupid" enough provide another student, whom may compete with you for a future internship or career, a copy of your assignment, then you deserve the same penalty as the other student. If you are a "real" friend, tutor your friend.

***The instructor reserves the right to require face-to-face hands-on demonstrations or face-to-face tests to provide additional evidence to be submitted to the RMU Academic Integrity Board.***

## How to complete Content Questions

Review questions are also be provided at the end of the tutorial. The following is an example of a review question format. Since type the answer in provided grey or colored box.

1. What is the purpose of a partitioned data set? Answer:

Type in the answer to the question into the grey or colored box.

**It is recommended that you use Table of Contents at the beginning of the tutorial to review and navigate to the concept presented in the review question. Students will find that using the document FIND tool or searching GOOGLE may also be valuable for researching the review question answer.**

# 1.0 Introduction to the Relational Model

**Read the Introduction to Database Management PDF presentation In Week 1**

**Read "Relational Model” - Chapter 4 - Database Systems: A Practical Approach to Design, Implementation, and Management.**

**Video - Database Lesson #1 of 8 - Introduction to Databases -** [**https://www.youtube.com/watch?v=4Z9KEBexzcM**](https://www.youtube.com/watch?v=4Z9KEBexzcM)

**Video - The Fundamental Concepts of Database system PART 1 -** [**http://www.youtube.com/watch?v=fR\_xnOv2r6E&feature=relmfu**](http://www.youtube.com/watch?v=fR_xnOv2r6E&feature=relmfu)

**Video - The Fundamental Concepts of Database system PART 2 -** [**http://www.youtube.com/watch?v=3ywXh3NyUO0**](http://www.youtube.com/watch?v=3ywXh3NyUO0)

**Video - The Fundamental Concepts of Database system PART 3 - http://www.youtube.com/watch?v=ZWDGIHhvkNU&feature=relmfu**

## 1.1 Introduction to Relational Model Terminology

RDBMS Concepts - http://www.studytonight.com/dbms/rdbms-concept

Relation Data Model - <http://www.tutorialspoint.com/dbms/relational_data_model.htm>

Relational database - <https://en.wikipedia.org/?title=Relational_database>

Flat file - <http://searchsqlserver.techtarget.com/definition/flat-file>

Overview of Database - http://www.studytonight.com/dbms/overview-of-dbms

The relational model represents the conceptual design, storing, retrieving and processing (relational calculus). The relation model is a mathematical theoretical description. A relational database. The relational database management system (RDBMS) like DB2, Oracle, MySQl, SQL Server) implements the relational model. The concept of relational model is theory, and a RDBMS is a real-world application of that theory.

Edgar F. Codd of IBM was responsible for developing the relational model in 1970. In 1974, Codd and other IBM researchers of the System R project created the first relational database managements system and SQL which was called System R and later named DB2. System R and DB2 was designed to be a more flexible and powerful alternative to IBM's IMS hierarchical model DBMS. The concepts of the relational mode and System R and the relational model (except for the details of their error codes) was made available to all researchers. In 1982, Oracle released their first RDBMS, which was designed for other platforms than IBM, which was based on the research of Codd and IBM. However, when it comes to large scale RDBMSs requirements Oracle is executed on an IBM mainframe using DB2.

The following concepts will apply relational model terminology to RDMS SQL and applied concepts. In a previous assignment, you were required to execute a SQL script you were required to execute the following SQL script which contained a SQL CREATE TABLE which created the storage structure to store that created an employee table named EMP and Department table name named DEPT.

|  |  |
| --- | --- |
| **Employee Table** | **Department Table** |
| **CREATE TABLE EMP (**  **EMPNO CHAR(6) NOT NULL ,**  **FIRSTNME VARCHAR(12) NOT NULL ,**  **MIDINIT CHAR(1) NOT NULL ,**  **LASTNAME VARCHAR(15) NOT NULL ,**  **WORKDEPT CHAR(3) ,**  **PHONENO CHAR(4) ,**  **HIREDATE DATE ,**  **JOB CHAR(8) ,**  **EDLEVEL SMALLINT NOT NULL ,**  **SEX CHAR(1) ,**  **BIRTHDATE DATE ,**  **SALARY DECIMAL(9,2) ,**  **BONUS DECIMAL(9,2) ,**  **COMM DECIMAL(9,2) ) ;.** | **CREATE TABLE DEPT (**  **DEPTNO CHAR(3) NOT NULL ,**  **DEPTNAME VARCHAR(29) NOT NULL ,**  **MGRNO CHAR(6) ,**  **ADMRDEPT CHAR(3) NOT NULL ,**  **LOCATION CHAR(16) ) ;** |

## 1.2 Questions - Basic Relational Concepts

1. Using relational model terminology the concept of the table named EMP would be defined as a(n) \_\_\_\_\_\_\_\_\_\_\_\_.

Answer =>

Attributes

.Summer 2018

2. Using relational model terminology the concept of the EMP column names, e.g., EMPNO, FIRSTNME, etc. would be defined as \_\_\_\_\_\_\_\_\_\_\_\_. Answer =>

domain names

.Summer 2018

3. Using relational model terminology the concept that defines the set of values that may be stored in the EMP column names, e.g., EMPNO, FIRSTNME, etc. would be defined as \_\_\_\_\_\_\_\_\_\_\_\_. Answer =>

.domain definitionSummer 2018

4. Using relational model terminology the concept that describes that one or more rows are used store data for one or more employees would be \_\_\_\_\_\_\_\_\_\_\_\_. Answer =>

tuples

.Summer 2018

## 1.3 Questions - Degree and Cardinality

5. Using relational model terminology and given previous CREATE TABLE statement, what would be the **degree** of the relation as applied to the EMP table? Answer =>

the degree would be two because there two entities are participating in a relationship. That is why binary.

.Summer 2018

6. Using relational model terminology and given previous CREATE TABLE statement, why would the degree of the EMP table would be describe as **n-ary**? Answer =>

.Summer 2018

7. Based on concepts of the relational model, describe the concept of the **cardinality** as applied to the EMP table? Answer =>

cardinality applied to EMP table would be one,as each employee will work for one dept.

.Summer 2018

8. Based on the concept of the **cardinality,** provide a practical explanation that would explain why cardinality may increase from 10 to 15? Answer =>

if a participation of a relationship of an entity increases the cardinality increases too.

.Summer 2018

9. One can describe the structure of data stored using three different points of view: 1) the relational model terminology, 2) relational database management system terminology, and 3) the traditional flat file terminology. Complete the following table

|  |  |  |
| --- | --- | --- |
| **Relational Model Terminology** | **RDBMS terminology** | **Flat file terminology** |
| Relation | Table | File or dataset |
| Tuple |  |  |
| Attribute |  |  |

What’s referential integrity? - http://www.programmerinterview.com/index.php/database-sql/what-is-referential-integrity/

What is the advantages in using foreign keys and disadvantages in not using? - <http://stackoverflow.com/questions/18830786/what-is-the-advantages-in-using-foreign-keys-and-disadvantages-in-not-using>

Foreign key - <https://en.wikipedia.org/?title=Foreign_key>

SQL FOREIGN KEY Constraint - <http://www.w3schools.com/sql/sql_foreignkey.asp>

Primary and Foreign Key Constraints - <https://msdn.microsoft.com/en-us/library/ms179610.aspx>

In SQL, what are the differences between primary, foreign, and unique keys?

http://www.programmerinterview.com/index.php/database-sql/differences-between-primary-and-foreign-keys/

10. Using relational model terminology what would be effect of creating two tables with the table name of EMP? Answer =>

.Summer 2018

## 1.4 Questions - Relational Constraints

11. Given the previous CREATE TABLE EMP statement and the relational model, what would one do to ensure that a given employee is stored in only one row? (Hint: This is an example of a relational key.) Answer =>

if any tuple unique ness constraint is applied in each row ,then the unique key should be different in each . this will prevent any repetation of record.

.Summer 2018

12. The relationship between the DEPT and the EMP is very important to consider in the relational model and RDBMSs. A given DEPT row or tuple may have a relationship to many EMP rows who work in a given department. This relationship between the DEPT and EMP table is created by the common column named WORKDEPT and DEPTNO (Notice the column names do not match but must have the same only the data type.) Which relational model concepts would you implements would you implemented to ensure that no employee can be stored in an EMP row, unless they have a valid WORKDEPT column which matches a DEPTNO in the DEPT table? (Hint: This is an example of two types of relational keys.) Be specific! Answer =>

emp table WORKDEPT would be foreing key of each employee which will connect to the primary of DEPTNO.

.Summer 2018

## 1.5 Questions - Relational Views

What is a Relational Database View? <http://www.essentialsql.com/what-is-a-relational-database-view/>

View (SQL) - <https://en.wikipedia.org/wiki/View_%28SQL%29>

What are benefits of using view in database? <http://stackoverflow.com/questions/7450423/what-are-benefits-of-using-view-in-database>

Given the following CREATE VIEW statement answer the following questions.

**CREATE VIEW EMPLOYEE**

**(EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT,**

**PHONENO, HIREDATE, JOB, EDLEVEL, SEX,**

**BIRTHDATE, SALARY, BONUS, COMM)**

**AS**

**SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT,**

**PHONENO, HIREDATE, JOB, EDLEVEL, SEX,**

**BIRTHDATE, SALARY, BONUS, COMM**

**FROM EMP**

**WHERE EMPNO <= '000340';**

13. In the relational model what is the function or advantages of implementing a relational view? Answer =>

views are used for security purposes, as view displays only selected data, which is not permanent.

14. Using the previous SQL CREATE VIEW statement, applying n the relational model, what is the name of the view? Answer =>

From the previous SQL CREATE VIEW statement, the name of the view is EMPLOYEE.

.Summer 2018

15. Using the previous SQL CREATE VIEW statement, applying n the relational model, what is the name of the base relation? Answer =>

here base relation is EMP

.Summer 2018

# 2.0 - Relational Algebra Concepts

**Read “Relational Algebra and Relational Calculus” - Chapter 5 - Database Systems: A Practical Approach to Design, Implementation, and Management.**

## 2.1 Questions - Projection and Selection

What is the difference between Select and Project Operations? http://stackoverflow.com/questions/5432573/what-is-the-difference-between-select-and-project-operations

**What are projection and selection? -** [**http://stackoverflow.com/questions/1031076/what-are-projection-and-selection**](http://stackoverflow.com/questions/1031076/what-are-projection-and-selection)

**Join, Projection and Selection - http://help.sap.com/saphelp\_erp60\_sp/helpdata/en/cf/21ec6a446011d189700000e8322d00/content.htm**

Relational Algebra - <http://infolab.stanford.edu/~ullman/fcdb/aut07/slides/ra.pdf>

Given the following data stored in the Project table.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PROJ (Project) Table** | | | | | | | |
| **PROJNO** | **PROJNAME** | **DEPTNO** | **RESPEMP** | **PRSTAFF** | **PRSTDATE** | **PRENDATE** | **MAJPROJ** |
| **CHAR(6)** | **VARCHAR(24)** | **CHAR(3)** | **CHAR(6)** | **DECIMAL(5,2)** | **DATE** | **DATE** | **CHAR(6)** |
| 'AD3100' | 'ADMIN SERVICES' | 'D01' | '000010' | +006.50 | '1982-01-01' | '1983-02-01' | ' ' |
| 'AD3110' | 'GENERAL ADMIN SYSTEMS' | 'D21' | '000070' | +006.00 | '1982-01-01' | '1983-02-01' | 'AD3100' |
| 'AD3111' | 'PAYROLL PROGRAMMING' | 'D21' | '000230' | +002.00 | '1982-01-01' | '1983-02-01' | 'AD3110' |
| 'AD3112' | 'PERSONNEL PROGRAMMING' | 'D21' | '000250' | +001.00 | '1982-01-01' | '1983-02-01' | AD3110' |
| 'AD3113' | 'ACCOUNT PROGRAMMING' | 'D21' | '000270' | +002.00, | '1982-01-01' | '1983-02-01' | 'AD3110' |
| 'IF1000' | 'QUERY SERVICES' | 'C01' | '000030' | +002.00 | '1982-01-01' | '1983-02-01' | NULL |
| 'IF2000' | 'USER EDUCATION' | 'C01' | '000030' | +001.00 | '1982-01-01' | '1983-02-01' | NULL |
| 'MA2100' | 'WELD LINE AUTOMATION' | 'D01' | '000010' | +012.00 | '1982-01-01' | '1983-02-01' | NULL |
| 'MA2110' | 'W L PROGRAMMING' | 'D11' | '000060' | +009.00 | '1982-01-01' | '1983-02-01' | 'MA2100' |
| 'MA2111' | 'W L PROGRAM DESIGN' | 'D11' | '000220' | +002.00 | '1982-01-01' | '1982-12-01' | 'MA2110 |
| 'MA2112 | 'W L ROBOT DESIGN' | 'D11 | '000150' | +003.00 | '1982-01-01' | '1982-12-01' | 'MA2110' |
| 'MA2113' | 'W L PROD CONT PROGS' | 'D11' | '000160' | +003.00, | '1982-02-15' | '1982-12-01' | 'MA2110' |
| 'OP1000' | 'OPERATION SUPPORT' | 'E01' | '000050' | +006.00 | '1982-01-01' | '1983-02-01' | NULL |
| 'OP1010' | 'OPERATION' | 'E11' | '000090' | +005.00 | '1982-01-01' | '1983-02-01' | 'OP1000' |
| 'OP2000' | 'GEN SYSTEMS SERVICES' | 'E01 | '000050' | +005.00 | '1982-01-01' | '1983-02-01' | NULL |
| 'OP2010' | 'SYSTEMS SUPPORT' | 'E21' | '000100' | +004.00 | '1982-01-01' | '1983-02-01' | 'OP2000' |
| 'OP2011' | 'SCP SYSTEMS SUPPORT' | 'E21' | '000320' | +001.00 | '1982-01-01' | '1983-02-01' | 'OP2010' |
| 'OP2012' | 'APPLICATIONS SUPPORT' | 'E21' | '000330' | +001.00 | '1982-01-01' | '1983-02-01' | 'OP2010' |
| 'OP2013' | 'DB/DC SUPPORT' | 'E21' | '000340' | +001.00 | '1982-01-01' | '1983-02-01' | 'OP2010' |
| 'PL2100' | 'WELD LINE PLANNING' | 'B01' | '000020' | +001.00 | '1982-01-01' | '1982-09-15' | 'MA2100' |

1. The requirement to display **result data** or all projects that satisfies the condition that the DEPTNO equals 'D11', e.g., SELECT \* FROM PROJ WHERE DEPTNO ='D11', would be an example of which relational algebra concept? Answer =>

.selection operationSummer 2018

2. Copy and paste a sample of **result data** from the previous Project table that illustrates the requirement to display all projects that satisfies the condition that the DEPTNO equals 'D11, e.g., SELECT \* FROM PROJ WHERE DEPTNO ='D11'. Answer =>

.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 'MA2110' | 'W L PROGRAMMING' | 'D11' | '000060' | +009.00 | '1982-01-01' | '1983-02-01' | 'MA2100' |
| 'MA2111' | 'W L PROGRAM DESIGN' | 'D11' | '000220' | +002.00 | '1982-01-01' | '1982-12-01' | 'MA2110 |
| 'MA2112 | 'W L ROBOT DESIGN' | 'D11 | '000150' | +003.00 | '1982-01-01' | '1982-12-01' | 'MA2110' |
| 'MA2113' | 'W L PROD CONT PROGS' | 'D11' | '000160' | +003.00, | '1982-02-15' | '1982-12-01' | 'MA2110' |

Summer 2018

3. The requirement to display **result data** from only the columns named PROJNO, PROJNAME and DEPTNO for all projects, e.g., SELECT PROJNO, PROJNAME, DEPTNO FROM PROJ, , would be an example of which relational algebra concept? Answer =>

projection operation

.Summer 2018

4. Copy and paste a sample of ***result data*** from the previous Project table that illustrates the requirement to display only the columns named PROJNO, PROJNAME and DEPTNO for all projects, e.g., SELECT PROJNO, PROJNAME, DEPTNO FROM PROJ. Answer =>

.

|  |  |  |
| --- | --- | --- |
| **PROJNO** | **PROJNAME** | **DEPTNO** |
| **CHAR(6)** | **VARCHAR(24)** | **CHAR(3)** |
| 'AD3100' | 'ADMIN SERVICES' | 'D01' |
| 'AD3110' | 'GENERAL ADMIN SYSTEMS' | 'D21' |
| 'AD3111' | 'PAYROLL PROGRAMMING' | 'D21' |
| 'AD3112' | 'PERSONNEL PROGRAMMING' | 'D21' |
| 'AD3113' | 'ACCOUNT PROGRAMMING' | 'D21' |
| 'IF1000' | 'QUERY SERVICES' | 'C01' |
| 'IF2000' | 'USER EDUCATION' | 'C01' |
| 'MA2100' | 'WELD LINE AUTOMATION' | 'D01' |
| 'MA2110' | 'W L PROGRAMMING' | 'D11' |
| 'MA2111' | 'W L PROGRAM DESIGN' | 'D11' |
| 'MA2112 | 'W L ROBOT DESIGN' | 'D11 |
| 'MA2113' | 'W L PROD CONT PROGS' | 'D11' |
| 'OP1000' | 'OPERATION SUPPORT' | 'E01' |
| 'OP1010' | 'OPERATION' | 'E11' |
| 'OP2000' | 'GEN SYSTEMS SERVICES' | 'E01 |
| 'OP2010' | 'SYSTEMS SUPPORT' | 'E21' |
| 'OP2011' | 'SCP SYSTEMS SUPPORT' | 'E21' |
| 'OP2012' | 'APPLICATIONS SUPPORT' | 'E21' |
| 'OP2013' | 'DB/DC SUPPORT' | 'E21' |
| 'PL2100' | 'WELD LINE PLANNING' | 'B01' |

Summer 2018

5. The requirement to display only the columns named PROJNO, PROJNAME and DEPTNO for all projects that satisfies the condition that the DEPTNO equals 'D11', e.g., SELECT PROJNO, PROJNAME, DEPTNO FROM PROJ WHERE DEPTNO ='D11', would be an example of which relational algebra concepts? Answer =>

.first of all selection operation to display specified condition like D11, then projection operation to display PROJNO, PROJNAME, DEPTNOSummer 2018

6. Copy and paste a sample of **result data** from the previous Project table that illustrates the requirement to display only the columns named PROJNO, PROJNAME and DEPTNO for all projects that satisfies the condition that the DEPTNO equals 'D11', e.g., SELECT PROJNO, PROJNAME, DEPTNO FROM PROJ WHERE DEPTNO ='D11'. Answer =>

|  |  |  |
| --- | --- | --- |
| 'MA2110' | 'W L PROGRAMMING' | 'D11' |
| 'MA2111' | 'W L PROGRAM DESIGN' | 'D11' |
| 'MA2112 | 'W L ROBOT DESIGN' | 'D11 |
| 'MA2113' | 'W L PROD CONT PROGS' | 'D11' |

.Summer 2018

7. Which SQL keyword is used to implement the concept relational algebra selection? Answer =>

SELECT \* FROM WHERE

.Summer 2018

## 2.2 Questions - Unions, Intersection and Difference

Relational Algebra - http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/

Unions, intersections, and differences in dbms -

http://www.jooq.org/doc/2.5/manual/sql-building/sql-statements/select-statement/union-clause/

Relational Algebra Set Operations require two or more union compatible tables. Examples of Relational Algebra Set Operators are UNION, INTERSECTION and DIFFERENCE.

Given the follow two tables. The AFC North Teams table represents a selection of players for all player positions for AFC North teams, e.g., Steelers, Ravens and Bengals. The AFC North Team Quarterbacks and Punters table represents players who are quarterbacks or Punters for AFC North teams, e.g., Steelers, Ravens and Bengals.

|  |  |  |  |
| --- | --- | --- | --- |
| **AFC North Teams** | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 16 | Dalton, Andy | Bengals | QB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 19 | Flacco, Joe | Ravens | QB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |

|  |  |  |  |
| --- | --- | --- | --- |
| **AFC North Team QuarterBacks and Punters** | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam** | **PlayerPosition** |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 16 | Dalton, Andy | Bengals | QB |
| 19 | Flacco, Joe | Ravens | QB |
| 23 | Koch, Sam | Ravens | P |
| 24 | Berry, Jordan | Steelers | P |

8. Describe the purpose of a union operation. Answer =>

Combine the result of different relation. If we need to combine data from two different tables,

9. Copy and paste a sample of **result data** from the previous AFC North Teams and AFC North Team QuarterBacks and Punters that would be the result of the **union** between these two tables. Answer =>

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 16 | Dalton, Andy | Bengals | QB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 19 | Flacco, Joe | Ravens | QB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |
| 23 | Koch, Sam | Ravens | P |
| 24 | Berry, Jordan | Steelers | P |

10. What is a union-compatible relation or table? Answer =>

Union compatible relation means tables from which you want to make union should have same attribute.

11. Assume you have access to mailing list which provides the email addresses for medical doctors. Assume you also have access to an email address to a member on Pennsylvania household which has a family income above $100,000. If you were a creating an email list to market to market to potential high level income individuals, why might the union of these two email lists be beneficial? Answer =>

12. Copy and paste a sample of **result data** from the previous AFC North Teams and AFC North Team QuarterBacks that would be the result of the **union** between these two tables. Answer =>

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 16 | Dalton, Andy | Bengals | QB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 19 | Flacco, Joe | Ravens | QB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |
| 23 | Koch, Sam | Ravens | P |
| 24 | Berry, Jordan | Steelers | P |

13. Describe the purpose of an intersection operation. Answer =>

Intersection operation returns result which are common record in two tables.

14. Copy and paste a sample of **result data** from the previous AFC North Teams and AFC North Team QuarterBacks and Punters that would be the result of the **intersection** between these two tables. Answer =>

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 16 | Dalton, Andy | Bengals | QB |
| 19 | Flacco, Joe | Ravens | QB |

15. Describe the purpose of a difference operation. Answer =>

Difference operation returns all rows from the select query 1 which is not presented in the query 2 table.

16. Copy and paste a sample of **result data** from the previous AFC North Teams and AFC North Team QuarterBacks and Punters that would be the result of the **difference** between the previous AFC North Teams and AFC North Team QuarterBacks and Punters. Answer =>

|  |  |  |  |
| --- | --- | --- | --- |
| **PlayerID** | **PlayerName** | **PlayerTeam** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |

17. Copy and paste a sample of **result data** from the previous AFC North Team Players and AFC North Team QuarterBacks and Punters that would be the result of the **difference** between the previous AFC North Team QuarterBacks and Punters and AFC North Teams. Answer =>

|  |  |  |  |
| --- | --- | --- | --- |
| **PlayerID** | **PlayerName** | **PlayerTeam** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |

18. Review the previous email list scenario. Assume you have access to mailing list which provides the email addresses for medical doctors. Assume you also have access to an email address to a member on Pennsylvania household which has a family income above $100,000. In simple language describe the resulting email that would result as the difference between High Income PA residents and Medical Doctors. Answer =>

## 2.3 Questions - Relational Inner Joins, Equi-Joins, Left Joins, Right Joins, and Self Joins

Given the follow two tables. The AFC North Teams Players table represents a selection of players for all player positions for AFC North teams, e.g., Steelers, Ravens and Bengals. The AFC North Teams Information table represents the information for the teams, rather than the players.

.

|  |  |  |  |
| --- | --- | --- | --- |
| **AFC North Team Players** | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam**  **FK** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 16 | Dalton, Andy | Bengals | QB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 19 | Flacco, Joe | Ravens | QB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |

|  |  |  |  |
| --- | --- | --- | --- |
| **AFC North Teams Information** | | | |
| **PlayerTeam**  **PK** | **City\_State** | **Primary Owner** | **Join\_NFL** |
| Steelers | Pittsburgh, PA | Rooney | 1933 |
| Ravens | Baltimore, MD | Bisciotti | 1966 |
| Browns | Cleveland, OH | Lerner | 1949 |

SQL - Using Joins - http://www.tutorialspoint.com/sql/sql-using-joins.htm

Join (SQL) - <https://en.wikipedia.org/wiki/Join_%28SQL%29>

SQL JOIN - <http://www.w3schools.com/sql/sql_join.asp>

SQL - SELF JOINS - http://www.tutorialspoint.com/sql/sql-self-joins.htm

Using Self-Joins - https://technet.microsoft.com/en-us/library/ms177490%28v=sql.105%29.aspx

Relational databases are usually normalized to eliminate duplication of information such as when objects have one-to-many relationships. Joining separate tables for Department and Employee effectively creates another table which combines the information from both tables. This is at some expense in terms of the time it takes to compute the join. While it is also possible to simply maintain a de-normalized table if speed is important, duplicate information may take extra space, and add the expense and complexity of maintaining data integrity if data which is duplicated later changes.

19. Copy and paste a sample of result data from the previous AFC North Team Players and AFC North Teams Information table that would be the result of an **Inner Join**. Answer =>

**Primary Owner**

|  |  |  |  |
| --- | --- | --- | --- |
| Ravens | Carter, DeAndre | Bisciotti | 1933 |
| Steelers | Roethlisberger, Ben | Rooney | 1966 |
| Ravens | Smith Sr., Steve | Bisciotti | 1933 |
| Steelers | Harrison, James | Rooney | 1966 |
| Steelers | Bryant, Martavis | Rooney | 1966 |
| Steelers | Bell, Le'Veon | Rooney | 1966 |
| Ravens | Flacco, Joe | Bisciotti | 1933 |
| Steelers | Allen, Cortez | Rooney | 1966 |

20. Copy and paste a sample of result data from the previous AFC North Team Players and AFC North Teams Information tables that would be the result of an **Equi-Join.** Answer =>

|  |  |  |
| --- | --- | --- |
| Carter, DeAndre | Rooney | 1933 |
| Roethlisberger, Ben | Bisciotti | 1966 |
| Smith Sr., Steve | Bisciotti | 1966 |
| Melvin, Rashaan | Rooney | 1933 |
| Harrison, James | Bisciotti | 1966 |
| Bryant, Martavis | Bisciotti | 1966 |
| Bell, Le'Veon | Rooney | 1933 |
| Flacco, Joe | Rooney | 1933 |

21. Not all versions of SQL provide for an Equi-Join operation because of its dangers in interpreting the results. Explain the dangers of an Equijoin. Answer =>

22. What is the function of an Outer Join? Answer =>

outer join displays data set which will give inner join data and also include some rows for which no corresponding data is found in other table.

Copy and paste a sample of result data from the previous AFC North Team Players and AFC North Teams 23. Information tables that would be the result of a **Left Outer Join** between theAFC North Team Players and AFC North Teams Information tables Answer =>

|  |  |  |  |
| --- | --- | --- | --- |
| **AFC North Team Players** | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam**  **FK** | **Primary Owner** | **Join\_NFL** | | |
| 11 | Carter, DeAndre | Ravens | Bisciotti | 1966 |
| 12 | Roethlisberger, Ben | Steelers | Rooney | 1933 | | |
| 13 | Smith Sr., Steve | Ravens | Bisciotti | 1966 |
| 14 | Melvin, Rashaan | Raven | NULL | NULL | |
| 15 | Harrison, James | Steelers | Rooney | 1933 | | |
| 16 | Dalton, Andy | Bengals | NULL | NULL | |
| 17 | Bryant, Martavis | Steelers | Rooney | 1933 | | |
| 18 | Bell, Le'Veon | Steelers | Rooney | 1933 | | |
| 19 | Flacco, Joe | Ravens | Bisciotti | 1966 |
| 20 | Allen, Cortez | Steelers | Rooney | 1933 | | |
| 21 | Burfict, Vontaze | Bengals | NULL | NULL | |
| 22 | Jones, Marvin | Bengals | NULL | NULL | |

24. Copy and paste a sample of **result data** from the previous AFC North Team Players and AFC North Teams Information tables that would be the result of a **Right Outer Join** between theAFC North Team Players and AFC North Teams Information tables Answer =>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AFC North Teams Information** | | | | |
| **PlayerTeam**  **PK** | **Primary Owner** | **Join\_NFL** | **PlayerID** | **PlayerName** | | |
| Steelers | Rooney | 1933 | 12 | Roethlisberger, Ben | | | |
| Ravens | Bisciotti | 1966 | 11 | Carter, DeAndre | | | |
| Browns | Lerner | 1949 | NULL | NULL | |
| Ravens | Bisciotti | 1966 | 13 | Smith Sr., Steve | | | |
| Steelers | Rooney | 1933 | 15 | Harrison, James | | | |
| Steelers | Rooney | 1933 | 17 | Bryant, Martavis | | | |
| Steelers | Rooney | 1933 | 18 | Bell, Le'Veon | | | |
| Ravens | Bisciotti | 1966 | 19 | Flacco, Joe | | | |
| Steelers | Rooney | 1933 | 20 | Allen, Cortez | | | |

25. Copy and paste a sample of **result data** from the previous AFC North Team Players and AFC North Teams Information tables that would be the result of a **Full Outer Join** between theAFC North Team Players and AFC North Teams Information tables. Answer =>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | Carter, DeAndre | Ravens | Bisciotti | 1966 |
| 12 | Roethlisberger, Ben | Steelers | Rooney | 1933 |
| 13 | Smith Sr., Steve | Ravens | Bisciotti | 1966 |
| 14 | Melvin, Rashaan | Raven | NULL | NULL |
| 15 | Harrison, James | Steelers | Rooney | 1933 |
| 16 | Dalton, Andy | Bengals | NULL | NULL |
| 17 | Bryant, Martavis | Steelers | Rooney | 1933 |
| 18 | Bell, Le'Veon | Steelers | Rooney | 1933 |
| 19 | Flacco, Joe | Ravens | Bisciotti | 1966 |
| 20 | Allen, Cortez | Steelers | Rooney | 1933 |
| 21 | Burfict, Vontaze | Bengals | NULL | NULL |
| 22 | Jones, Marvin | Bengals | NULL | NULL |
| NULL | NULL | Browns | Lerner | 1949 |

Review the following table

|  |  |  |  |
| --- | --- | --- | --- |
| **AFC North Team Players** | | | |
| **PlayerID** | **PlayerName** | **PlayerTeam**  **FK** | **PlayerPosition** |
| 11 | Carter, DeAndre | Ravens | WR |
| 12 | Roethlisberger, Ben | Steelers | QB |
| 13 | Smith Sr., Steve | Ravens | WR |
| 14 | Melvin, Rashaan | Raven | CB |
| 15 | Harrison, James | Steelers | LB |
| 16 | Dalton, Andy | Bengals | QB |
| 17 | Bryant, Martavis | Steelers | WR |
| 18 | Bell, Le'Veon | Steelers | RB |
| 19 | Flacco, Joe | Ravens | QB |
| 20 | Allen, Cortez | Steelers | CB |
| 21 | Burfict, Vontaze | Bengals | LB |
| 22 | Jones, Marvin | Bengals | WR |

26. How would one apply a self-join to find the pairings to each player who are members of the same Player Team? Answer =>

|  |  |
| --- | --- |
| Carter, DeAndre | Ravens |
| Smith Sr., Steve | Ravens | |
| Flacco, Joe | Ravens | |
| Roethlisberger, Ben | Steelers | |
| Harrison, James | Steelers | |
| Bryant, Martavis | Steelers | |
| Bell, Le'Veon | Steelers | |
| Allen, Cortez | Steelers | |
| Dalton, Andy | Bengals | |
| Burfict, Vontaze | Bengals | |
| Jones, Marvin | Bengals | |
| Melvin, Rashaan | Raven | |

# 3.0 Data Structures

## 3.1 Introduction to Data Structures

A data structure is a term used to describe the algorithms used store, order and retrieve data. A data structure is a particular way of organizing data in a computer so that it can be used efficiently. Many data structures are enhancements of the relative file access method RRDS VSAM datasets) or internal arrays or tables. Other data structures are enhancements of the indexed sequential file access method.

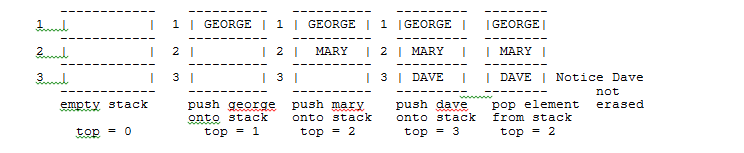
Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, databases use B-tree indexes for small percentages of data retrieval and compilers and databases use dynamic hash tables as look up tables.

Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design. Storing and retrieving can be carried out on data stored in both main memory and in secondary memory.

## 3.2 Stacks

**Stacks and Queues -** [**https://www.cs.cmu.edu/~adamchik/15-121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.html**](https://www.cs.cmu.edu/~adamchik/15-121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.html)

**STACKS** are a data structure that exhibits LIFO behavior. Elements can be added to the stack (pushed onto the stack). However, only the element that has been added the most recently can removed (pushed) from the stack. Given an array or relative file with 3 locations:



### 3.2.1 Advantages and Disadvantages of Stacks

|  |  |
| --- | --- |
| **Advantages of a Stack** | **Disadvantages of a Stack** |
| very efficient in terms of storage space and execution time | A stack is not order by any natural key. There is only an implied arrival order that the element at the top was the one most recently arrived even though we did not store the time of insertion |
| data is ordered by a LIFO arrival order | Requires a sequential search to search for a specific record.    Average sequential search = max. no of records/2 + 1, e.g. if there was 1000 records, it would take on the average 501 sequential accesses to find a given record. |

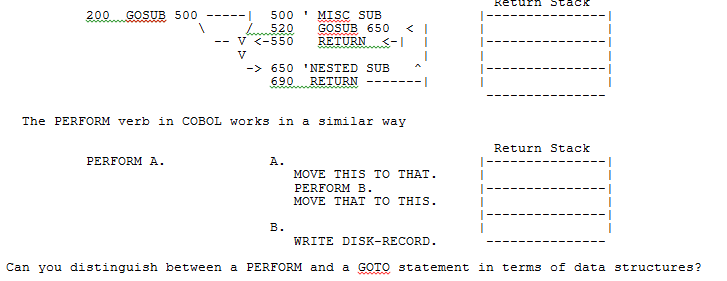
### 3.2.2 Uses of Stacks

1) A stack manages **memory space allocation** for operating systems, variables and programs. For example, every new program loaded into the PCs memory is pushed on the memory stack, Exiting a program does not erase a program, but merely resets the entry point for the current program to a new "top of stack".

What does it mean to "shell to the Windows command line". When one types CMD and he the RUN Prompt the command shell is loaded in separate address space than Windows. It you type CMD again a second command line is loaded on top of the pervious command shell. This similar of putting a hat on top of your head, then putting a second hat on top of the previous hat, etc.,

What does the EXIT at command line accomplish? The EXIT release the address space from command shell and returns to the previous program, e.g., Windows, Word, etc. It is analogous to removing the top hat on your head and return to the previous layer.

2) A stack manages the **backtracking process** for GOSUBS, Program and Methods CALLs (invocation) and the COBOL Program. The example uses the old-fashion Basic GOSUBs because it uses program line numbers. When a GOSUB is executed the programs pushes the line to return to. When the RETURN statement is executed (or the end of a COBOL paragraph or JAVA method, it pops the line to return to and transfer the program flow back to that point. It is kind of similar to Little Red Riding Hood dropping bread crumbs to find her way home.



Return Stack

3) Stacks are responsible for **executing math operations** in computers and calculators. For example 1) push the first operand (number) on the equation stack, push the second operand (number) on the equation stack, when the operator is encountered pop the top two elements, execute the math operation and push the answer on the equation stack .

There are three types of mathematical notations: INFIX, PREFIX, and POSTFIX. While humans are accustomed to understanding INFIX notation, a program language must convert the INFIX notation to POSTFIX notation to execute the mathematical expression.

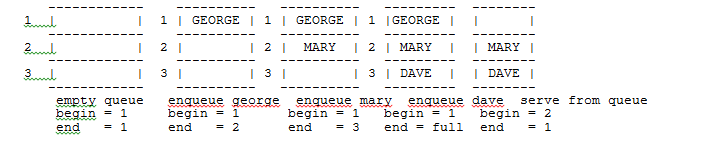
INFIX 3 + 4 operand, operator, operand

PREFIX + 3 4 operator, operand, operand

POSTFIX 3 4 + operand, operand, operator

## 3.3 FIFO Queues

**FIFO QUEUES** is a data structure that exhibits first‑in/first‑out behavior. A new item is added to the end of the queue (enqueue) and item is removed from the beginning of the queue (served). Notice again there is no natural key order. The advantages and disadvantages are similar to a stack, except this data structure exhibits a FIFO arrival order.



### 3.3.1 Uses of FIFO Queues

1) Printer queues

2) Waiting list for a closed class

3) Computer jobs waiting for a resource availability, e.g., tape drive or disk.

### 3.3.2 Priority Queues

**PRIORITY QUEUES** - Queues ignore time of arrival of the insertion of new data. A priority coefficient is calculated by some type of formula. The formula may be based upon a user's priority, the size of the job, urgency of the job, etc. The item is inserted into the queue, not at the end, but at place as close to beginning as its priority will permit it as compared to the other items in the queue at that time. Therefore, a priority queue is a highest-priority-in/first out structure. The priority coefficient must be stored in the queue.

### 3.3.3 Uses of Priority Queues

1) Allocating computer execution time in a time sharing operating system as well as providing an alternative for FIFO queues, e.g., schedule disk I/O requests by the shortest seek time). For example, CICS calculates a priority coefficient based upon the user, the nature of the transaction and the terminal from which the transaction will be executed.

2) Print queues

### 3.3.4 Advantages and Disadvantages of Queue Data Structures

**Advantages**

1) The data in the queue is ordered by priority.

**Disadvantages**

1) No lexographical order. Lexographical order can be any type of sequencing or order, e.g., numeric, strings, etc.

2) Extra storage required for the priority coefficient

3) Extra execution overhead required to calculate and compare the priority coefficient.

4) Insertion requires a sequential (some priority queues are in reality binary trees to be discussed later. this will improve the insertion time.)

## 3.4 Ordered Linked List

Linked Lists - https://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html

Video - Data Structures: Introduction to Linked Lists - https://www.youtube.com/watch?v=pBrz9HmjFOs

An Ordered Linked Listis a data structure that maintains order by a symbolic or real pointer. Records are inserted into the list or deleted from the list in the way that will preserve order without any additional sorting.

FIRST= 0 FIRST= 1 FIRST=1 FIRST= 3 FIRST =3 ------------ ----------- ---------- ---------- ----------

1 | |0 | 1 |GEORGE |0| 1 |GEORGE|2| 1 |GEORGE|2| 1 | |F|<-next

------------ ----------- ---------- ---------- ---------- free

2 | |0 | 2 | |0| 2 | MARY |0| 2 | MARY |0| 2 |MARY |0|

------------ ----------- ---------- ---------- ----------

3 | |0 | 3 | |0| 3 | |0| 3 | DAVE |1| 3 |DAVE |2|

------------ ----------- ---------- --------- ----------

empty list insert insert insert delete

george mary dave george

### 3.4.1 Advantages and Disadvantages of an Ordered Link List

**Advantages of an Ordered Link List**

1) May be order lexically, chronologically or it may be ordered by frequency of use. Order is dynamic, which means the order is preserved after insertion and deletion without any additional sorting.

**Disadvantages of an Ordered Link List**

1) Additional storage is required to store the pointer to the next ordered record.

2) Insertion, deletion and inquiry may involve a lengthy sequential search and the list becomes larger (n/2 +1).

**Uses of an Ordered Linked List.**

1) Maintaining key order in file structures and in databases.

2) Connecting fragmented segments of a file stored on a disk, i.e., MSDOS

## 3.5 Two-Way Link List

**A Two-Way Link List** may be implemented by adding an additional pointer that will point to the previous ordered record. Therefore, a one way list can only be transversed from the first element to the last; but, not in reverse order.

FREE = FULL

FIRST= 3 LAST = 2

---------------

1 |GEORGE |2 | 3|

---------------

2 | MARY |0 | 1|

---------------

3 | DAVE |1 | 0|

---------------

| ^

V |

## 3.6 Circular List

A **Circular List** can be implemented by changing the value of the end of the list pointer from 0 (end of list) to the first element in the list.

FREE = FULL

FIRST= 3 LAST = 2

---------------

1 |GEORGE |2 | 3|

---------------

2 | MARY |3 | 1| Notice no null or zero pointers

---------------

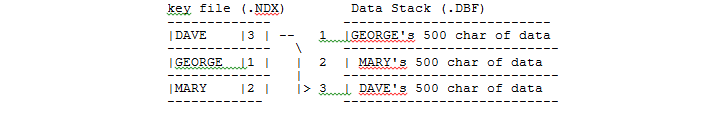
3 | DAVE |1 | 2|

---------------

## 3.7 Inverted Files or Pointers

**An Inverted File** or pointer maintains two separate files. The first file is a data stack that inserts new data by arrival order. The second file is a key file that only stores the key field and pointer (a cross-file pointer) to the location of the data for that key located in the stack. The key file is maintained in some type of sorted order, either by sorting, an ordered linked list, binary tree, etc. The issue of order is not pertinent to the concept of an inverted file. Rather, it is easier to maintain any order by manipulating a small key field than to process the entire data record.

Assume that the following key or index file has been sorted but not the data.



When you create a relational table the RDBMS storage manager creates a data structure similar to a stack. Any new rows are inserted at the end of the stack. The relational table has no sorted order and must be search from the beginning of a table. To provide order and faster searching one creates a Table Index based on a table column. The INDEX creates a separate data structure, which contains the column value and maintains a data structure that provides dynamic ordering and faster look up. This example of a key file is superficial. Never would an order linked list be used to main order. Rather a combination as Binary Tree, BnTree or Hash data structure will be used as appropriate to the design requirements and the type of data stored in the table.

This discussion of the inverted file or data structure clearly indicates that every relational index uses external storage space and slows performance with the index is updated, inserted or deleted. While the good news may be an inverted pointer may use other data structure which permits dynamic ordering and faster inquires there is always an extra storage and performance cost.

## 3.8 Binary Tree

Binary Trees - https://www.cs.cmu.edu/~adamchik/15-121/lectures/Trees/trees.html

Video - Data structures: Introduction to Trees - https://www.youtube.com/watch?v=qH6yxkw0u78

Video - binary tree tutorial Introduction - <https://www.youtube.com/watch?v=IAFNTOLwCgg>

**A Binary Tree** is a hierarchical data structure that each node can have at most 2 children or subtrees, 1 children or no children. The path length of each node is determined by the number of nodes that must be transversed from the root node to that particular node. Each node must have at two pointers. If a new data element is to be inserted at a given node that already has a key value, e.g., GEORGE, any new element less than the node key will be pointed to by the left pointer. Any new element greater the node key will be pointed by the left pointer. If both pointers of a given node (row) are currently being used, the insert algorithm should be repeated for the left and right child. The fact that a binary tree can be defined as a set of binary subtrees with one root node, means that binary tree algorithms are recursive. This means that the algorithm that works for the parent node will work correctly at the root node.

empty tree insert george insert fred ---------------- --------------- ------------- 1 |0| |0| 1 |0| GEORGE |0| 1 |2|GEORGE|0|

---------------- --------------- -------------

---------------

2 |0| FRED |0|

---------------

insert paul insertadam

--------------- -------------

1 |2| GEORGE |3| 1 |2|GEORGE|3|

--------------- -------------

---------------- --------------- --------------- -------------

2 |0| FRED |0| 3 |0| PAUL |0| 2 |4| FRED |0| 3|0| PAUL |0|

---------------- --------------- --------------- -------------

-------------

4 |0| ADAM |0|

-------------

insert irwin and tom

---------------

1 |2| GEORGE |3|

---------------

--------------- ---------------

2 |4| FRED |0| 3 |5| PAUL |6|

--------------- ---------------

---------------- --------------- -------------

4 |0| ADAM |0| 5 |0| IRWIN |0| 6 |0|TOM |0|

---------------- --------------- -------------

### 3.8.1 Advantages and Disadvantages of Binary Tree

**Advantages of Binary Tree**

1. Inserting, deleting or searching for a given data in a binary tree is a factor of

LOG2 (number of elements) +1 if the tree is balanced. The tree is balanced if the length of the longest path to the last the left element (the leaf) is equal to the longest path to the last right

2) An inorder transversal of a tree will visit and process the nodes in lexographical order.

**Disadvantages of a Binary Tree**

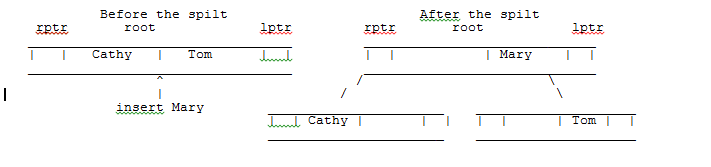
1) Slight extra storage requirements

2) However, do to random arrival of new inserts, the binary tree does not insure a balanced tree without some additional algorithmic overhead. Therefore, a balance tree's performance may degraded to a sequential linked list.

3) Additional overhead for a balancing algorithm

## 3.9 Bn-Tree

**A Bn Tree** is a modification of the binary tree algorithm. Each node or record stores two or more keys (where 'n' represents the number of keys per node). When the node is filled, the tree splits in a balanced fashion (see below). One can describe the binary tree algorithm as growing from the root 'down'. On the other hand, the Bn trees are perceived as growing upwards, constantly pushing the key value toward the root.



### 3.9.1 Uses of Bn Tree

1) Used by many databases to provide key or indexed order, i.e., DB2, Btrieve, Oracle, XDB

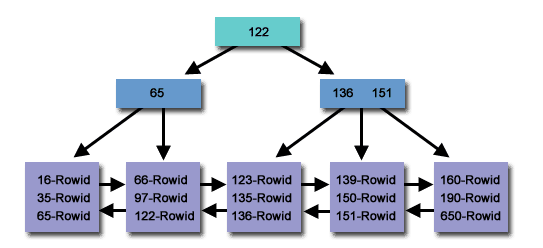
2) Slowly replacing the indexed sequential file access method.

### 3.9.2 Advantages of Bn Tree

1) Same as a Binary Tree, except that there is log 2 + performance without the need of a balancing algorithm.

### 3.9.3 Oracle b-tree index

**Oracle includes numerous data structures to improve the speed of Oracle SQL queries. Taking advantage of the low cost of disk storage, Oracle includes many new indexing algorithms that dramatically increase the speed with which Oracle queries are serviced. This article explores the internals of Oracle indexing; reviews the standard b-tree index, bitmap indexes, function-based indexes, and index-only tables (IOTs); and demonstrates how these indexes may dramatically increase the speed of Oracle SQL queries.  
  
Oracle uses indexes to avoid the need for large-table, full-table scans and disk sorts, which are required when the SQL optimizer cannot find an efficient way to service the SQL query. I begin our look at Oracle indexing with a review of standard Oracle b-tree index methodologies.**  
The oldest and most popular type of Oracle indexing is a standard b-tree index, A B-tree index excels at servicing simple queries. The b-tree index was introduced in the earliest releases of Oracle and remains widely used with Oracle.   
  
B-tree indexes are used to avoid large sorting operations. For example, a SQL query requiring 10,000 rows to be presented in sorted order will often use a b-tree index to avoid the very large sort required to deliver the data to the end user.

  
*An Oracle b-tree index*

Oracle offers several options when creating an index using the default b-tree structure. It allows you to index on multiple columns (concatenated indexes) to improve access speeds. Also, it allows for individual columns to be sorted in different orders. For example, we could create a b-tree index on a column called lastname in ascending order and have a second column within the index that displays the salary column in descending order.

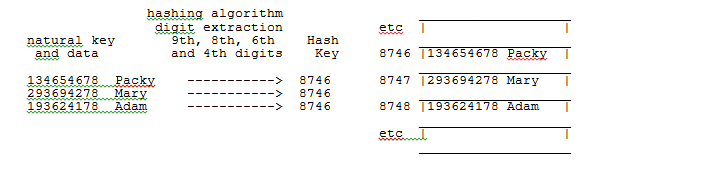
## 3.10 Hash Tables

Basics of Hash Tables - https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/tutorial/

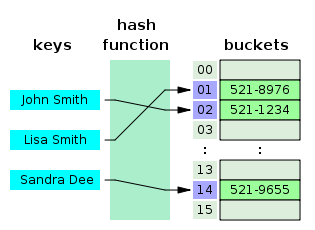
Video - What is a HashTable Data Structure - <https://www.youtube.com/watch?v=MfhjkfocRR0>

**A Hash Table**‑ is a relative file or array that reserves a number of storage positions, i.e., 1000 positions. The natural key, i.e., a social security number, is converted to a relative record number within the limits of the available storage position by a hashing algorithm. Natural keys that "hash" to the same storage locations are located at various overflow storage positions.

A Hash table is a data structure that can map keys to values. Ideally, the hash function will assign each key to a unique bucket, but it is possible that two keys will generate an identical hash causing both keys to point to the same bucket. Instead, most hash table designs assume that hash collisions—different keys that are assigned by the hash function to the same bucket—will occur and must be accommodated in some way.



A small phone book as a hash table

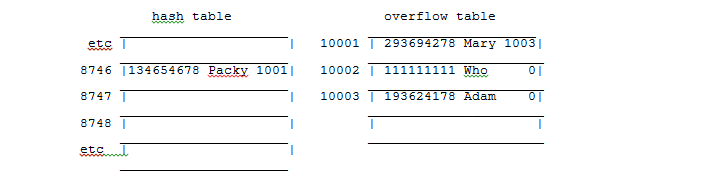


### 3.10.1 Hash Collision Strategies

There are two primary types of hash collision strategies:

1) **Progressive Overflow** ‑ when an original hash address is occupied, a progressive overflow strategy will attempt to use the next higher location until it finds an open record. The previous diagram illustrates a progressive overflow strategy.

2) **Chained Overflow** ‑ will store collision records at a separate locations. Each overflow will be connected to the original hash address by he means of a linked list.



### 3.10.2 Uses of a Hash Table

1) The primary use for hash tables is the storage of primary key is databases, i.e., hierarchical and network.

2) Operating systems frequently hash the file directory table rather than using sequential lists, in order to increase the performance of opening and closing a file.

### 3.10.3 Advantages of a Hash Table

1) Near relative file performance for alphanumeric keys when accessing individual records.

### 3.10.4 Disadvantages of a Hash Table

1) No lexographical order, generally relies of other supplementary data structures, i.e. Bn Trees to provide order.

2) As the 'load factor' increases (the percentage of stored record relative to the reserved positions) above 60%, the performance of hash table begins to degrade into a sequential search.

Primary Clustering ‑ several original hash addresses, i.e., 8746, 8747 and 8748, will use the same overflow address: 8749.

Secondary Clustering ‑ two or more natural keys using the same hash address, i.e., a collision.

3) Since a large percentage of a hash table is left generally unused so as to keep performance near relative file levels, the hash table will waste considerable storage space.

### 3.10.11 Factors affecting the performance of Hashing

1) The Load Factor- How full is the hash table?

2) The Collision strategy3) The performance of the hashing algorithm relative to the data being stored. There is no one best hashing algorithm. Some are best for numeric keys, others are best for alphanumeric keys, and so forth. The objective is hashing algorithm is to hash each natural key to a unique location, without collision. This is impossible. But, a bell shaped normal curve representing hash collisions is unacceptable.

4) The number of keys stored at one hash location.

## 3.11 Indexed Sequential (VSAM)

**Indexed Sequential** (VSAM) is a sequential data file that stores the key information is separate index sets. To access record 4751 one would sequentially search the Master Index. The master index would contain the highest key stored on a disk volume. Since 5456 was the first index greater than record 4751, then the record must be found on the cylinder index for volume 3 (V3). The cylinder indexed contains the highest key stored on a cylinder. Since 4845 is greater than 4751, then the record must be stored on the track indexed for cylinder 4 (C4). The process is repeated for the track index and the disk head is moved to track 48 to sequential search all the records from 4720 until it finds the correct data record.

Master Cylinder Track

Index Index (V3) Index (C4) Data tracks

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|1869 |V1| | 3678 |C1| |4621 |T45| |4432| .. | .. |... | ..|4621 T45

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|3750 |V2| | 3999 |C2| |4670 |T46| |4635| .. | .. |...|....|4670 T46

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|5456 |V3| | 4400 |C3| |4712 |T46| |4675|... | ...| ...|...|4712 T47

|8647 |V4| | 4845 |C4| | 4789|T48| |4720|4726|4750|4751|4769|4789 T48

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 5166 |C5| |4822 |T49| |4790|... |... |... |...|4822 T49

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 5456 |C6| |4845 |T50| |4823|.. | ...|....|...|4845 T50

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The original ISAM file would be created by loading sorted records from a sequential data file. Therefore, when the ISAM was first created it processed records sequentially as fast as a normal sequential data file. However, when new records were added, these records would be stored in a separate overflow location and connected to the original data track by pointers. The original tremendously degraded in performance with high file volatility a high percentage of adds or deletes). This necessitated frequent reorganizations of the file. A reorganization run copies the data records from an ISAM into a sequential file and then the records are reloaded into a newly created ISAM file. All the records in the overflow area are now stored contiguously in the data area.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

01 03 04 07 12 13 16 19 21 ------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

02 28 31 32 34 37 40 43 47 ---- |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

| |

----------------- --------------| |

\_\_\_\_\_\_\_\_\_\_\_\_V\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_V\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Overflow 56 25 53 <- 52 23 50 |

\_\_^\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_V\_\_^\_\_\_\_\_\_\_^\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_ |

|----------------| | |-----|-----------------|

|------------<|

The VSAM enhancement substantially improved the indexed sequential file's ability to handle high file volatility by leaving extra space open on each data track when the file was created and by using the 'track splitting technique when the track was full.

Before After

13

\_\_\_\_\_V\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 11 15 18 22 1 11 13 13 \_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

2 28 29 2 |>25 26 29 |

\_\_\_\_\_^\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

26 | |

| |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|-- 18 22 <---------

overflow \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 4.0 Questions – Data Structures

Data structure - <https://en.wikipedia.org/wiki/Data_structure>

Hash table - <https://en.wikipedia.org/wiki/Hash_table>

Advantage of B+ trees over BSTs? <http://stackoverflow.com/questions/15485220/advantage-of-b-trees-over-bsts>

ADVANTAGES AND DISADVANTAGES OF HASHING - http://rajaghoshtech2.blogspot.com/2010/03/advantages-and-disadvantages-of-hashing.html

Advantages of Binary Search Trees over Hash Tables - <http://stackoverflow.com/questions/4128546/advantages-of-binary-search-trees-over-hash-tables>

Comparison of B-Tree and Hash Indexes - https://docs.oracle.com/cd/E17952\_01/refman-5.5-en/index-btree-hash.html

1. What are the functions of Data Structures? Answer =>

The term Data Structure is used for arranging and storing of data. Data structure includes various types like array, file, record, table, tree and so on. Data structure is specially designed to organize data in such a way that it can fulfill some purpose.

.Summer 2018

2. What is meant by a FIFO data structure? Answer =>

Queue system of data Structure in which data is server by First In First Out basis. It is kind of data structure where data is accessed and removed in the order in which it was inserted.

.Summer 2018

3. What is meant by a LIFO data structure? Answer =>

LIFO data structure work in a stack like the data which has been added last , is used first.

.Summer 2018

4. List the applications of a Stack data structure in computers. Answer =>

helps to undo some action , or backtracking.

.helps to remember partially completed tasks in computer

helps to execute math operation in computer and calculators.Summer 2018

5. List the applications of a Queue data structure in computers. Answer =>

Serving the request when single resource is share by multiple processes like printers.

.it is useful in handling of interrupts in real-time system.Summer 2018

6. What are the disadvantages of Binary Tree data structure as compared to Bn-trees Answer =>

in binary tree data structure, a can have maximum 2 numbers of children or subtree and bn tree is a modification of binary tree and a node can have two or more numbers of keys. Summer 2018

7. What are the advantages of an Oracle Index built on a hash data structure as compare to an Oracle index based on a BnTree index? Answer =>

.Summer 2018

8. What is a Hash Collision? Answer =>

in a hash table data structure, data is mapped through keys to values. When two different keys hash to the same value, or to the same location in a hash table , then collision happens.

.Summer 2018

9. Which type of data structure is used table data and is created when you execute a SQL CREATE TABLE statement? Answer =>

.Summer 2018

10. Which types of data structure are used table data and are created when you execute a SQL CREATE INDEX statement? Answer =>

.Summer 2018