Readings:

* chapter 5: “Physical Database Design and Performance” (Hoffer, Ramesh, & Topi)

Assignment:

* Chapter 5(Hoffer, Ramesh, & Topi)
  + Problems and Exercises 1,5,8,9,13,16,17

1)

1. On what attributes should indexes be defined to speed up this query? Give the reasons for each attribute selected.
2. Write SQL commands to create indexes for each attribute you identified in part a.

Answers:

1. **Student ID ,Course ID, GPA, Student Name,  
   Reasons:   
   StudentID – primary key and indexes of the key.  
   StudentID and CourseID – uniqueness and primarykey, index.  
   StudentName -Non-key cluster to retrieve records in sorted order.  
   GPA – non key cluster to retrieve records faster.**
2. **CREATE UNIQUE INDEX STUDENTID\_PK ON STUDENT (STUDENTID) ,   
   CREATE UNIQUE INDEX STUD\_COURSE\_ID ON REGISTRATION (STUDENTID and COURSEID)  
   CREATE INDEX STUD\_NAME ON STUDENT (STUDENTNAME)  
   CREATE INDEX GPA\_CLUST\_INDX ON STUDENT(GPA)  
   CREATE INDEX STUD\_ID\_CLUST\_INDX ON REGISTRATION (STUDENTID)**

5) Say that you are interested in storing the numeric value 3,456,349.2334. What will be stored, with each of the follow- ing Oracle data types:

a. NUMBER(11)

b. NUMBER(11,1)

c. NUMBER(11,-2)

d. NUMBER(6)

e. NUMBER

**a) NUMBER(11) – Integer field with a length of 11 so it will store max of 11 digit. Hence 3456349  
b) NUMBER(11,1) - It can store decimal field with a precision of 1 , that is on the left side max 11 digits while on the right side max of 1 digit. 3456349.2  
c) NUMBER(11,-2) – it can store max of 11 digits and (-2 ) denotes that last 2 digit will replace the value with 0  
3456300  
d) NUMBER(6) – it cannot store the value as it exceeds the precision  
e) NUMBER – it can store the value as 3,456,349.2334 since no precision is mentioned.**

8)

STORE (StoreID, Region, ManagerID, SquareFeet)   
EMPLOYEE (EmployeeID, WhereWork, EmployeeName,EmployeeAddress)  
DEPARTMENT (DepartmentID, ManagerID, SalesGoal)   
SCHEDULE (DepartmentID, EmployeeID, Date)

What opportunities might exist for denormalizing these relations when defining the physical records for this data- base? Under what circumstances would you consider creat- ing such denormalized records?

**Answer:**

**Denormalizing we do in order to access the data faster, in normalized data there is reference data due to which when we run query each time database look for two or more tables.   
to refine this design we should use many to many relationship, in above design Manager ID and Employee ID are attributes which we need to denormalize**

A screenshot of a cell phone

Description automatically generated

9)

TEAM(TeamID, TeamName, TeamLocation)

PLAYER(PlayerID, PlayerFirstName, PlayerLastName,PlayerDateOfBirth, PlayerSpecialtyCode)

SPECIALTY(SpecialtyCode, SpecialtyDescription)

CONTRACT(TeamID, PlayerID, StartTime, EndTime, Salary)

LOCATION(LocationID, CityName, CityState,CityCountry, CityPopulation)

MANAGER(ManagerID, ManagerName, ManagerTeam)

What recommendations would you make regarding opportunities for denormalization? What additional information would you need to make fully informed denormalization decisions?  
 **Answers:**

**We need to denormalize player and speciality , Team and Manager , because these tables has frequent retrieve information from table. It will also reduce join time of two table.  
we can remove the speciality entity from the table because specialityCode is nothing but PlayerSpecialityCode and this attribute can be added to Player entity. So we are not required to have speciality entity requirement.**  
  
13)  
Assume that a student table in a university database had an index on StudentID (the primary key) and indexes on Major, Age, MaritalStatus, and HomeZipCode (all secondary keys). Further, assume that the university wanted a list of students majoring in MIS or computer science, over age 25, and mar- ried OR students majoring in computer engineering, single, and from the 45462 zip code. How could indexes be used so that only records that satisfy this qualification are accessed?

**Answers: Entity Student Table   
Attributes : Major , Age, Maritial Status, HomeZipCode**

**Index : STUDENTID – UNIQUE KEY INDEX since it is a primary key.**

**Secondary key : Age, Marital Status, HomeZipCode, Major.  
CLUSTERED INDEX – Age , Major, Marital Status , HomeZipCode.**

**If we want to retrieve information about student majoring in MIS or Computer science , over age 25 and married or students majoring in Computer Science , Single from zip code 4562**

**We would need these condition in where clause hence clustered index and unique index on these attriubutes will help retrieve information faster.  
  
WHERE (MAJOR = ‘MIS’ or MAJOR = ‘COMPUTER SCIENCE’**

**AND AGE > 25**

**AND MARITAL STATUS = SINGLE**

**OR MAJOR = ‘COMPUTER ENGINEERING’  
AND MARITAL STATUS = ‘SINGLE’**

**AND HOMEZIPCODE =45462  
- The main memory scans the index and it will list operations without using the secondary memory  
- The qualified records are populated quickly and to process the query the only indexes need secondary memory to access those records.**  
  
  
16. Can clustering of files occur after the files are populated with records? Why or why not?   
Answer :

**No, Clustering of files will not occur after the files are populated with records   
Because, it will be difficult to accomplish records if tables are retrieve information before clustering.**

**Also, the column by which the tables are joined, might need to have same values.  
A Cluster is created using a single column or multiple column by which tables are joined together.**

17. Parallel query processing, as described in this chapter, means that the same query is run on multiple processors and that each processor accesses in parallel a different subset of the database. Another form of parallel query pro- cessing, not discussed in this chapter, would partition the query so that each part of the query runs on a different processor, but that part accesses whatever part of the data- base it needs. Most queries involve a qualification clause that selects the records of interest in the query. In general, this qualification clause is of the following form:

(condition OR condition OR . . .) AND (condition OR

conditionOR. . .)AND. . .

Given this general form, how might a query be broken apart so that each parallel processor handles a subset of the query and then combines the subsets together after each part is processed?

Answer:

**Generally queries includes qualification clause such as (OR.condition .) AND (OR Condition) , parallel query response is based on number of available processors, one set of condition is called as conjuction and can be assigned to separate processor and after each processor process the query than later joined together for final result**

**Below is the example for one of the parallel processing approach.  
SELECT STUDENT\_ID, STUDENT\_NAME  
FROM STUDENT  
WHERE (MAJOR = “MIS” OR MAJOR = “COMPUTER SCIENCE”)**  
**AND AGE > 25A  
AND MARITAL STATUS =”SINGLE”)  
OR (MAJOR =”COMPUTER ENGINEERING”  
AND MARITAL STATUS =”SINGLE”  
AND HOMEZIPCODE =45462)**

**We need to ensure that subsequent scans of this table should run in parallel with not less than 3 processor, we also need to modify the table with the command   
ALTER TABLE STUDENT PARALLEL 3**