Function and Procedure

Q.1. Given the relation

Instructor(id, name, dept\_name, salary).

Write a function named salary-status that will take id as input parameter and return the salary status as follows:

If salary is less than 50000, the status will be ‘low income’

If salary is less than 80000 and greater than or equal to 50000, the status is ‘middle income’

Otherwise, the salary status is ‘high income’.

Write an SQL statement to find the id and salary status of all instructors.

Answer:

CREATE FUNCTION salary-status(instructor\_id INT)

RETURNS VARCHAR(50)

BEGIN

Declare i\_status varchar(50)

Declare i\_salary decimal(10, 2)

select salary into i\_salary

from instructor

where id = instructor\_id

if i\_salary < 50000 then

set i\_status = 'low income'

elseif i\_salary < 80000 then

set i\_status = 'middle income'

else

set i\_status = 'high income'

end if

return i\_status;

END

select id, salary-status(id) as 'salary status'

from instructor;

3Q2: Given the relation

student(id, name, dept\_name, tot-credit).

Write a function named credit-status that will take id as input parameter and return the credit status as follows:

If total credit is less than 100, the status will be ‘Junior’

Otherwise, the status is ‘Senior’.

Write an SQL statement to find the id and credit status of all students.

Answer:

create function credit-status (student\_id INT)

returns VARCHAR(50)

begin

declare s\_tot-credit INT

declare s\_status VARCHAR(50)

select tot-credit into s\_tot-credit

from student

if s\_tot-credit < 100 then

set s\_status = ‘Junior’

else

set s\_status = ‘Senior’

end if

return s\_status

end

select id, credit-status(id) as ‘credit status’

from student

Q. 3: Given the parents of students schema as follows:

Student\_parents(F-NID, M-NID, S-id, F-name, M-name, street, city, annual-income)

Write a function named s-p-income-status that will take S-id type integer as input and return the status of the income as follows:

If annual-income is less than or equal to 2,50,000, the status is ‘Poor’. If annual-income is less than or equal to 30,00,000 and greater than 2,50,000, the status is ‘Middle Class’. Others are ‘Rich’.

Write SQL statement to find S-id, F-name and income status of all students of Dhaka city

Answer:

create function s-p-income-status (student\_id INT)

returns VARCHAR(50)

begin

declare i\_status VARCHAR(50)

declare d\_annual-income INT

select annual-income into d\_annual-income

from Student\_parents

if d\_annual-income <= 250000 then

set i\_status = ‘Poor’

elseif d\_annual-income <= 3000000 and d\_annual-income > 250000 then

set i\_status = ‘Middle Class’

else

set i\_status = ‘Rich’

end if

return i\_status

end

select S-id, F-name, s-p-income-status(S-id) as ‘income status’

from Student\_parents

Procedure

Q4. Given the relation

Instructor(id, name, dept\_name, salary).

Write a procedure named salary-update that will take two ids (i-id1 and i-id2) as input parameter and update the salaries as follows:

Consider i-id2 is always greater than i-id1.

For all instructors id less than or equal to i-id1, salary will be increased 20%.

For all instructors id less than or equal to i-id2 and greater than i-id1, salary will be increased 15%.

For all other instructors, salary will be increased 10%.

Call the procedure for updating salaries of instructors as above with i-di1 = 50 and i-id2 =80.

Answer:

create procedure salary-update (in i-id1 integer,

in i-id2 integer)

begin

if i-id2 <= i-id1 then

return

end if

update Instructor

set salary = salary \* 1.2

where id <= i-id1

update Instructor

set salary = salary \* 1.15

where id <= i-id2 and id > i-id1

update Instructor

set salary = salary \* 1.1

where id > i-id2

end

call salary-update(50, 80)

Q. 5: The courses taken by students have been given in the schema as follows:

Takes (id, course-id, semester, year, section-id, credit-hour, grade)

Write a procedure course-count-tot-credit that will have id (integer type) as input variable and two output variables t-course-count (integer type) and tot-credit (integer type). The total number of courses taken by the student id will be assigned to t-course-count and sum of the credit-hour of all courses taken by the student id will be assigned to tot-credit.

Call the procedure with id = 1001 and two variables t-count and t-ctrdit of integer type.

Answer:

create procedure course-count-tot-credit(in student\_id integer,

out t-course-count integer,

out tot-credit integer)

begin

select count(\*) into t-course-count, sum(credit-hour) into tot-credit

from Takes

where id = student\_id

end

decalre t-count integer

declare t-credit integer

call course-count-tot-credit(1001, t-credit, t-count)

Index

|  |
| --- |
| Q. 1:  Give the following relation:  Person (NID, name, DOB, street, city)  Person relation is stored physically sorted order of NID. Answer the following questions.   1. Which type of indexing is created for DOB (primary/secondary)? 2. Which type of indexing is created for NID (primary/secondary)? 3. Why secondary index must be dense index?   Answer:  i. Secondary  ii. Primary  iii. Secondary indexes are often required to be dense indexes because they need to map every possible value of the indexed attribute to its corresponding record in the primary index. This ensures that there are no "gaps" in the index, allowing efficient retrieval of records based on the indexed attribute. |

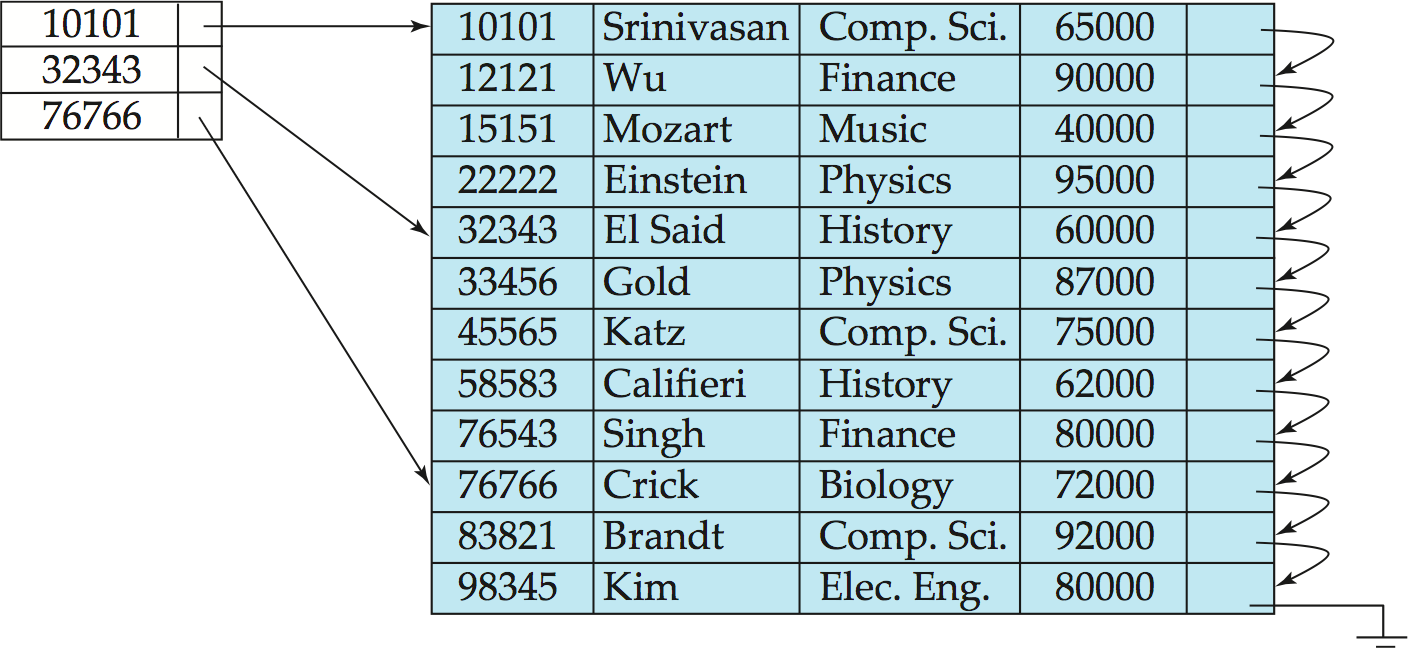
2: A table will be given. You will have to created dense index, sparse index, secondary index with bucket.

SQL will be given. You will have to explain how the SQL will be executed using the index.

Answer:

1. Dense Index – Question 5 given table

2. Sparse Index



3. Secondary Index – Answer of the Question 5

3. Explain how the query SELECT \* FROM instructor WHERE id = 83821 is executed using the given index as follows.

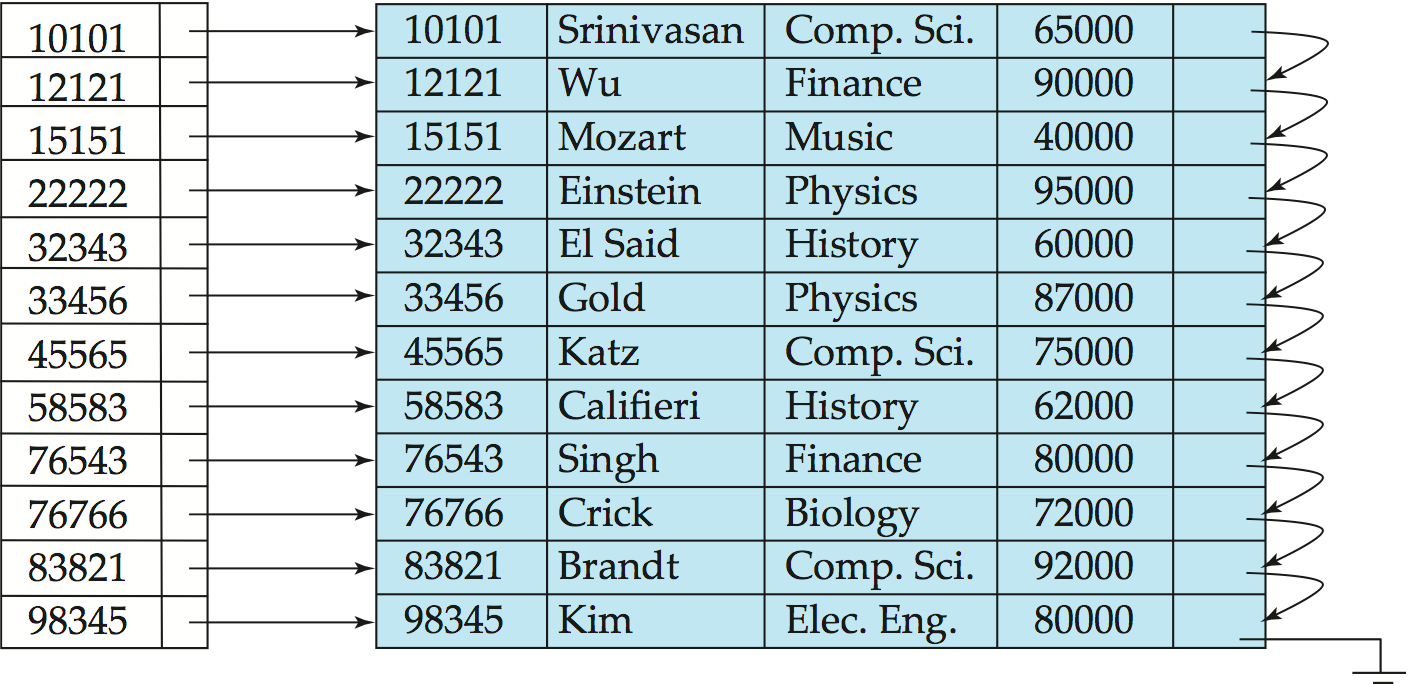
Answer: First, find the physical address (pointer) of id = 83821 by searching the index file

Then, using the physical address of 83821; the record from the disk is fetched into the memory and give the output.

4. Explain how the query SELECT \* FROM instructor WHERE salary < 70000 is executed.

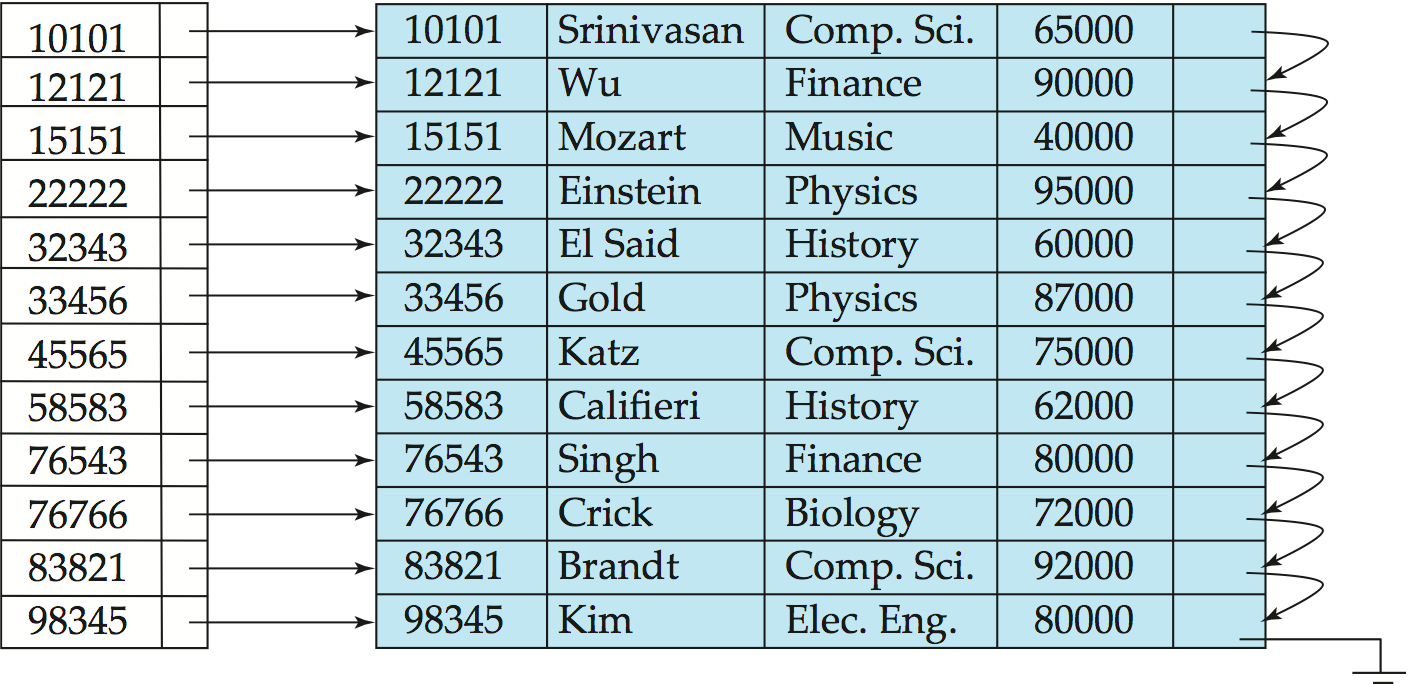
Answer: First, The database will sequentially read each row from the ‘instructor’ table and evaluate the condition salary < 70000 for each row. After scanning the entire table and applying the filter, the database will return all the columns for the rows where the salary is less than 70,000.

**Instructor (id, name, dept\_name, salary)**



5. Construct a secondary index structure on dept\_name of instructor relation as above.

Answer:



|  |  |
| --- | --- |
| Comp.Sci. |  |
| Finance |  |
| Music |  |
| Physics |  |
| History |  |
| Biology |  |
| Elec.Eng. |  |